SAARLAND UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

PROVABLE SECURITY OF SYMMETRIC-KEY CRYPTOGRAPHIC SCHEMES IN CLASSICAL AND QUANTUM FRAMEWORKS

DISSERTATION
ZUR ERLANGUNG DES GRADES
DES DOKTORS DER INGENIEURWISSENSCHAFTEN
DER FAKULTÄT FÜR MATHEMATIK UND INFORMATIK
DER UNIVERSITÄT DES SAARLANDES

VON

JORDAN ETHAN

SAARBRÜCKEN, 2024

Tag des Kolloquiums:	19.12.2024
Dekan:	Prof. Dr. Roland Speicher

Prüfungsausschuss:

Vorsitzender:	Prof. Dr. Markus Blaser
Berichterstattende:	Prof. Dr. Antoine Joux
	Prof. Dr. Cas Cremers
	Dr. Benoît Cogliati
Akademischer Mitarbeiter:	Dr. Rocco Mora

Dedicated to my parents.

Dedicated to my partner.

This page intentionally left blank.

Zusammenfassung

In dieser Dissertation beschäftigen wir uns mit dem Entwurf sicherer symmetrischer Kryptosysteme, indem wir Schwachstellen aufdecken, neue Konstruktionen vorschlagen und Sicherheitsbeweise gegen klassische und Quantenangreifer liefern.

Klassisches Setting: Zunächst entwerfen wir tweakbare Blockchiffren (TBCs), die über die Birthday-Grenze hinaus Sicherheit bieten. Wir schlagen ein tweakbares Verschlüsselungsschema mit einer einzelnen S-Box vor. Außerdem analysieren wir das `TWEAKEY`-Framework und leiten Schranken für IND-CCA-Sicherheit und Schlüsselwahlangriffe ab. Weiterhin untersuchen wir Authenticated Encryption (AE)-Schemes, analysieren das `MTProto`-Protokoll von Telegram, decken einen partiellen Schlüsselwiederherstellungsangriff auf und schlagen eine Lösung vor. Wir betrachten auch AEs für Leakage-Resilienz und Kontextbindung und entwickeln ein Blueprint zur Analyse von Single-Pass-Schemes wie `Triplex`.

Quanten-Setting: Wir analysieren $2n$-Bit-zu-n-Bit komprimierende Funktionen mit einem n-Bit PRF-Aufruf und zeigen, dass die meisten Zwei- oder Dreifach-Aufrufe unsicher sind. Wir identifizieren drei sichere Konstruktionen und beweisen ihre qPRF-Sicherheit mit einem neuen Framework basierend auf Zhandrys komprimiertem Orakel [325]. Zusätzlich entdecken wir die Grenzen des Frameworks für adaptive Angreifer und finden einen Fehler im Vier-Runden-Luby-Rackoff-Beweis [174]. Trotzdem beweisen wir die qPRF-Sicherheit einer Variante von Feistel-Netzwerken, den `Misty`-Konstruktionen.

This page intentionally left blank.

Abstract

In this dissertation, we focus on designing secure symmetric-key schemes by identifying flaws, proposing new constructions, and providing rigorous security proofs against classical and quantum adversaries.

Classical Setting: First, we design tweakable block ciphers (TBCs) that achieve security beyond the birthday bound. We propose a tweakable enciphering scheme with a single S-box. Further, we analyze the `TWEAKEY` framework, deriving bounds for IND-CCA security and chosen-key resistance. Second, we analyze authenticated encryption (AE) schemes, examining Telegram's `MTProto` protocol, revealing a partial key recovery subversion attack, and suggesting a fix. We further explore AEs for leakage resilience and context commitment, proposing a blueprint for analyzing single-pass schemes like `Triplex`.

Quantum Setting: We analyze $2n$-bit to n-bit compressing functions with a single n-bit PRF call, showing that most two-call or three-call functions are vulnerable. We identify three secure constructions and prove their qPRF security using a new framework based on Zhandry's compressed oracle [325]. Additionally, we discover the framework's limitations for adaptive adversaries and identify a flaw in the four-round Luby-Rackoff proof [174]. Nonetheless, we prove the qPRF security of a variant of Feistel networks, the `Misty` constructions.

This page intentionally left blank.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my official advisor, Prof. Dr. Antoine Joux. Your unwavering support and guidance, both academically and personally, have been truly invaluable. From the moment I first approached you for a position in 2020, you have been a constant source of inspiration and a guiding hand. I am profoundly thankful for your immense understanding, patience, and the countless ways you have helped me navigate the challenges along this journey.

I am immensely grateful to my unofficial supervisor, Dr. Benoît Cogliati, the leader of our research team. Our journey together began at the end of 2020, when you welcomed me into the newly formed research group at CISPA. I truly cherish the countless hours we spent with Dr. Ashwin Jha, solving problems on the board and brainstorming new research ideas. Those moments were not only intellectually stimulating but also among the most enjoyable and inspiring periods of my life. I am particularly thankful for your unwavering support and guidance, even after your transition to the private sector. Your personal attention and dedication have left a lasting impact on me, and I deeply appreciate our meaningful discussions and invaluable advice, which extended beyond academic matters to personal ones as well.

A special thanks goes to Dr. Ashwin Jha, my main collaborator and teammate. From the moment we met, you have been a constant source of inspiration and a true friend. I am deeply grateful for the immense effort you've invested in supporting me, both academically and in our research, even during those late-night brainstorming sessions. Your steadfast presence through both the highs and challenges is something I will always cherish and never forget. I also want to express my heartfelt gratitude for the private discussions and the wonderful times we shared during my time in Germany—those moments were truly special and will always be remembered fondly.

I would also like to acknowledge my other co-authors, Prof. Dr. Marine Minier, Dr. Byeonghak Lee, Dr. Jooyoung Lee, Dr. Virginie Lallemand, Dr. Ritam Bhaumik, Mr. Chandranan Dhar, Dr. Ravindra Jejurikar, Dr. Mustafa Khairallah, Dr. Eik List, Mr. Sougata Mandal and Mr. Abishanka Saha. Your contributions and the knowledge I gained through our collaborations have significantly shaped the direction and quality of this thesis.

I extend my thanks to all my CISPA colleagues for the engaging discussions, the coffee

breaks, and the fun times we shared. Your camaraderie made the work environment truly enjoyable.

To my father Samuel, my mother Valerie, and my brother Yonathan, I am profoundly grateful for your support and encouragement. Your belief in me has been a constant source of motivation, and I am deeply thankful for the sacrifices you have made to help me reach this milestone.

Finally, I wish to express my deepest gratitude to my partner, Cristina. Your love, patience, and understanding have been my greatest source of strength. I am forever grateful for your unwavering support throughout this journey.

This page intentionally left blank.

Contents

This page intentionally left blank.

List of Figures

This page intentionally left blank.

List of Tables

This page intentionally left blank.

Acronyms

AE	Authenticated Encryption.
AEAD	Authenticated Encryption with Associated Data.
AES	Advanced Encryption Standard.
ASA	Algorithm Substitution Attack.
AU	Almost Universal.
AXU	Almost XOR Universal.
BC	Block Cipher.
CBC	Cipher Block Chaining Mode.
CCA	Chosen-Ciphertext Attack.
CKA	Chosen-Key Attack.
CPA	Chosen-Plaintext Attack.
CR	Collision Resistance.
DAE	Deterministic Authenticated Encryption.
ECC	Elliptic Curve Cryptography.
EM	Even-Mansour.
ePre	Everywhere Preimage Resistance.
GCM	Galois Counter Mode.
IEM	Iterated Even-Mansour.
IND-CCA	Indistinguishability under Chosen-Ciphertext Attack.
IND-CPA	Indistinguishability under Chosen-Plaintext Attack.
IV	Initialization Vector.
KAC	Key-Alternating Cipher.
KAC	Key-Alternating Cipher.
KKA	Known-Key Attack.
LCR	Left Collision Resistance.

MAC Message Authentication Code.
mu Multi-User.

PRF Pseudorandom Function.
PRP Pseudorandom Permutation.

qPRF Quantum Pseudorandom Function.
qPRP Quantum Pseudorandom Permutation.
QROM Quantum Random Oracle Model.
qTPRP Quantum Tweakable Pseudorandom Permutation.

RCR Right Collision Resistance.
ROM Random Oracle Model.
RSA Rivest Shamir Adleman.

SPN Substitution-Permutation Network.
SPRP Strong Pseudorandom Permutation.
STK Superposition Tweakey Construction.
STPRP Strong Tweakable Pseudorandom Permutation.

TBC Tweakable Block Cipher.
TEM Tweakable Even-Mansour.
TEML Tweakable Even-Mansour with Linear Tweak and Key Mixing.
TPRP Tweakable Pseudorandom Permutation.

wPRF Weak Pseudorandom Function.

XOR Exclusive Or.

This page intentionally left blank.

1 Introduction

1.1 Overview

Cryptography originated as the science of securing communication by concealing information from unauthorized parties. According to Kahn [192], this practice dates back to ancient civilizations, such as Egypt, where non-standard hieroglyphs encoded religious texts, and Mesopotamia, where clay tablets contained encrypted trade secrets. The Hebrews used techniques like Atbash, a substitution cipher, to protect their writings. Over time, cryptography evolved with more sophisticated methods, such as the Caesar Cipher used by Julius Caesar and the Scytale cipher employed by the Spartans for military communications.

The development of cryptography saw significant advancements in the 20th century, marked by two major milestones. The first was the introduction of the mathematical foundations for cryptography by Claude Shannon in 1945, published in 1949 [290]. Shannon's work also led to the creation of information theory [291], which focuses on the quantification, storage, and communication of information. Until the 1970s, cryptography primarily concentrated on confidentiality, relying on symmetric key algorithms, where the same cryptographic key is used by both the sender and recipient, who must keep it secret. The second major milestone occurred in 1976 when Whitfield Diffie and Martin Hellman [114] introduced new security goals for cryptography, such as data integrity and authenticity. Their work also addressed the critical key distribution problem by introducing the concept of public key cryptography, a revolutionary advancement that became foundational to modern cryptographic practices.

Consequently, cryptography is classified into two main categories: symmetric-key cryptography and public-key (or asymmetric-key) cryptography. Symmetric-key cryptography, also referred to as secret-key cryptography, relies on a single key for both encryption and decryption. This approach is prized for its efficiency and speed, making it particularly well-suited for encrypting large volumes of data. However, the primary challenge of symmetric-key cryptography is the secure distribution and management of the key, as both the sender and recipient must share the same key and protect it from unauthorized access. To address this challenge, modern cryptographic protocols

typically begin with the negotiation of a common key using an asymmetric-key scheme. Although asymmetric-key schemes are computationally intensive, they are primarily used only during the initial key exchange.

In this thesis, we assume that the key distribution requirement has been met and focus exclusively on symmetric-key schemes.

1.2 Symmetric-Key Cryptography

Symmetric-key cryptography, where a single secret key is used for both encryption and decryption, is fundamental for secure digital communication. It is especially useful in applications like securing online banking transactions, encrypting data in wireless networks (e.g., Wi-Fi [100]), and protecting stored files on devices. As described in [114], modern symmetric-key schemes aim to achieve several security goals, including:

- **Confidentiality:** Ensures that only those with the correct key can access the encrypted data.
- **Data Integrity:** Verifies that the data has not been changed during transmission or storage.
- **Authenticity:** Confirms the identities of the communicating parties to prevent impersonation.

1.2.1 Cryptographic Primitives

In this thesis, we primarily focus on confidentiality, and also consider authenticity and data integrity. To set the stage for these topics, we will introduce some basic cryptographic primitives that help achieve the security goals, either independently or as components within more complex schemes. Throughout this chapter, we consider a sender A tries to send a message to receiver B through a secure scheme.

1.2.1.1 Pseudorandom Function

A pseudorandom function (PRF) is a keyed function, a function that is indexed by a key, with the property that its output is indistinguishable from a truly random function. The notion was introduced by Goldreich, Goldwasser and Micali, under the name poly-random collections [141] and later popularized with its finite variant by Bellare et al. [32]. For a formal definition see Section 2.3.1.

The concept of pseudorandom functions has become a foundational approach for constructing and analyzing symmetric-key primitives, including secure hash functions, key derivation functions, block ciphers, encryption schemes, and message authentication codes (e.g. [140, 220, 29, 34, 198, 39, 26, 193]).

1.2.1.2 Message Authentication Code

Although not explicitly studied in this work, a Message Authentication Code (MAC) is a cryptographic primitive designed to ensure authenticity and integrity of messages. Informally, a MAC scheme $\mathtt{M}$ consists of two algorithms: a tag generation algorithm $\mathtt{M}_g$ and a tag verification algorithm $\mathtt{M}_v$. When a sender A with a secret key k wants to send a message m to a receiver B, they compute a *tag* $t = \mathtt{M}_g(k, m)$ and send the pair (m, t). Upon receiving a pair (m', t'), the receiver B uses the verification algorithm $\mathtt{M}_v(k, m', t')$ to check the validity of the message. The verification process ensures that $\mathtt{M}_g(k, m')$ is equal to t'.

1.2.1.3 Block Cipher

One of the most widely used and straightforward encryption schemes for ensuring confidentiality is the block cipher. A block cipher (BC) is a symmetric encryption algorithm that processes fixed-size blocks of plaintext and ciphertext, typically consisting of 64, 128, or even 256 bits. Informally, a block cipher $\mathtt{E}$ is a function that encrypts a message m using a key k, producing a ciphertext $c = \mathtt{E}(k, m)$, such that for any key k, the mapping $m \mapsto \mathtt{E}(k, m)$ is a bijection. This means that each message m has a unique corresponding ciphertext c, and vice versa, under the same key k. For a sender A with a message m and key k, the encryption process is simply $c = \mathtt{E}(k, m)$. The decryption process initiated by receiver B, which recovers the original message simply uses the inverse function of $\mathtt{E}$, namely $m = \mathtt{E}^{-1}(k, c)$ (see Section 2.3.2 for formal definition).

Design Approach. The foundational principles of block cipher design, dating back to Shannon's work [291], emphasize constructing ciphers through an iterative process that applies simple operations in each round until desired cryptographic properties are achieved. A block cipher typically employs a round function F_i that, in each round, processes a state x using a round key k_i, derived from a master key k via a key schedule. The output of the final round produces the ciphertext. These product ciphers alternate between substitution operations for non-linearity (confusion), mixing operations for spreading influence across the data (diffusion), and key addition to integrate the key into the encryption process. Despite being abstract concepts, confusion and diffusion guide designers in constructing secure ciphers. Popular design strategies include Feistel networks, Substitution-Permutation Networks (SPNs), and Key-Alternating Ciphers (KAC), each balancing security and efficiency by leveraging these principles in various ways.

Block ciphers are not only fundamental to symmetric encryption but also serve as building blocks for a wide range of cryptographic primitives and protocols. These include schemes designed to ensure confidentiality, such as symmetric encryption schemes (e.g. [28, 277]), those that ensure authenticity, such as message authentication codes (MACs) (e.g. [235, 254]), and schemes that provide both confidentiality and authenticity, like authenticated encryption schemes (e.g.,[314, 129]). Additionally, block

ciphers are utilized to achieve other security goals, such as the construction of universal hash functions (e.g. [29]).

1.2.1.4 Tweakable Block Cipher

While traditional block ciphers are effective, they have limitations in certain applications where flexibility and additional control over the encryption process are required. A Tweakable block cipher (TBC) is a block cipher incorporating an additional public parameter known as a "tweak". This tweak introduces added variability at the message-block level, akin to how a nonce or an initialization vector (IV) brings variability at the message level. The notion of tweakable block ciphers was first formalized by Liskov, Rivest, and Wagner [216]. Informally, a tweakable block cipher $\widetilde{\mathtt{E}}$ takes as input a message m, a key k, and a tweak t, and produces a ciphertext c, such that for any fixed key k and tweak t, the mapping $m \mapsto \widetilde{\mathtt{E}}(k, t, m)$ is a bijection. See Section 2.3.3 for a formal definition.

The introduction of tweakable block ciphers is driven by the need for independent instances of a block cipher. For instance, in Electronic Codebook (ECB) mode of operation, identical ciphertext blocks indicate identical plaintext blocks, which can be exploited by adversaries. A direct solution involves replacing the key for each block with a new random key, but this re-keying process is typically costly. Therefore, tweakable block ciphers efficiently create independent instances of the cipher, without the need for re-keying.

Prior to the formalization of tweakable block ciphers (TBCs) by Liskov et al. [216], a few block ciphers were already designed with tweakability. For example: `HPC` [289], `Mercy` [101], and `Threefish` —which is integral to the Skein hash function [136]. Owing to their versatility, tweakable block ciphers have a broad range of applicability, most notably in authenticated encryption schemes [217, 278, 267], and message authentication codes [249, 183, 90, 149, 82]. Apart from these, TBCs have also been employed to achieve other symmetric-key security goals, including the construction of online ciphers [282, 190], the design of large domain block ciphers [244] and the attainment of strong security notions such as variable- input-length strong pseudorandom permutation (SPRP) [52].

1.2.1.5 Authenticated Encryption

In many scenarios, such as when the receiver needs to verify that the data originates from an authorized sender while also maintaining a secure communication channel, ensuring both confidentiality and authenticity is crucial. A simple way to achieve both requirements is to encrypt the message and authenticate it with a Message Authentication Code (MAC). However, trying to achieve confidentiality and authenticity separately can lead to security issues, which lead to the introduction of *Authenticated Encryption* (AE) schemes [35] that unify both functions. *Authenticated Encryption with Associated Data* (AEAD) [275] extends AE by also protecting additional data that must be authenticated

but not encrypted, such as addresses or routing information, ensuring message integrity while allowing some data to remain in plaintext for processing.

Informally, an authenticated encryption scheme $\mathcal{E}$ consists of two algorithms: an encryption algorithm $\mathcal{E}^+$ and a decryption algorithm $\mathcal{E}^-$. When the sender A wants to send a message m to the receiver B, they produce a pair $(c, t) = \mathcal{E}^+(k, m)$, where c is the ciphertext and t is the *authentication tag*. Upon receiving a pair (c', t'), the receiver B uses the decryption algorithm $\mathcal{E}^-(k, c', t')$ to get a message m', when the pair (c', t') is *valid* and a special symbol $\perp$ otherwise. Moreover, for any key k and message m it always holds that $\mathcal{E}^-(k, \mathcal{E}^+(k, m)) = m$. See Section 2.3.5 for a formal definition

The security of AE schemes requires that generated ciphertexts appear random, even if the same message (and associated data) is encrypted multiple times with the same key. To achieve this, modern AE schemes use a *nonce* (some number) [278] that must be unique for each encrypted message to randomize the output, also known as *nonce-based* AE. Standard security definitions for AE assume that nonces never repeat, placing the responsibility on the implementer. Therefore, in scenarios where ensuring unique nonces is challenging or impossible, users are left without security guarantees. Consequently, robustness against nonce misuse is a critical practical concern for AE. Popular examples of nonce-based AE include: `AES-GCM` [233], `OCB3` [205], `Ascon` [117] etc.

In 2006, Rogaway and Shrimpton [281] addressed this issue by introducing *deterministic authenticated encryption* (DAE), which deterministically transforms a key, message, and associated data into a ciphertext. In the same paper, they formalized the security notion of *nonce-misuse-resistant* authenticated encryption (AE) and demonstrated that a DAE construction following the SIV (Synthetic Initialization Vector) paradigm achieves this security notion. Several popular SIV-based AEAD schemes include: `GCM-SIV` and `GCM-SIV2` which were introduced in [153]. These schemes were further generalized by Iwata and Minematsu with `GCM-SIVr` [182], a lightweight construction known as `SUNDAE` [16] etc. One major drawback of SIV-based constructions is that they require two passes to process the message, making them slower and less efficient than nonce-based AE, which processes the message once.

1.2.1.6 Universal Hash Function

Unlike a single, publicly known cryptographic hash function, a universal hash function family consists of multiple distinct hash functions. Carter and Wegman [71] introduced universal hash function families to enhance hashing performance by minimizing collisions in storage and retrieval operations. Building on this, Wegman and Carter [312] applied these families to information-theoretic message authentication. They introduced the concept of strongly universal hash function families, exemplified by polynomials with bounded degrees, and demonstrated their effectiveness in authenticating single messages.

Following these works, various notions of universality for hash functions were in-

troduced [313, 109, 204, 238, 245, 276, 287]. In this thesis, we focus on the notion of Almost XOR Universal (AXU) hash functions, introduced by Krawczyk [204] and further developed by Rogaway [276]. This notion generalizes the original universal hash function concept by Carter and Wegman [71]. Popular examples using AXUs include: Poly1305 [43], UMAC [57],CBC-MAC [119],PMAC [58] and SipHash [14].

Informally, a universal hash function $\mathcal{H}$ is said to be AXU if for any randomly selected function H out of the family $\mathcal{H}$, the likelihood of a collision in its outputs is universally small (upper bounded by a universal small constant). For more details see Section 2.3.4.

1.3 Provable Security in Symmetric-Key Cryptography

Provable security is a subfield of cryptography focused on designing secure protocols and schemes by reducing their security to the security of atomic components known as *primitives*. The concept was first introduced in the pioneering work of Goldwasser and Micali [142] within the realm of public-key cryptography. This foundational idea rapidly extended to other areas, including pseudorandomness [59, 317, 141] and digital signatures [143].

According to Bellare [22], provable security is a paradigm that involves the following key steps:

- Formally define a *security model* that specifies the desired security goals, such as confidentiality, authenticity, or both.
- Formally define the capabilities of the adversary by establishing an *adversarial model*.
- Provide a proof of the security of the scheme, showing that it remains secure as long as the underlying primitives within the scheme are secure.

In the most general context, also known as the standard model, a primitive is considered secure as long as a specific computational problem is hard to solve. For example, the security of a primitive may rely on the hardness of the factoring problem or the assumption that `AES` [3] is a pseudorandom permutation (PRP), meaning it is computationally indistinguishable from a truly random permutation.

1.3.1 Reduction Proofs in The Symmetric-Key Setting

In the context of this thesis, we define a symmetric key scheme as comprising two main components. The first is a *primitive*, which is typically the smallest building block of the scheme and operates on small or fixed-size inputs. For example, a block cipher like `AES` is a primitive that encrypts fixed-size blocks of data. The second component is a *mode of operation*, which extends the scheme's functionality to handle variable-length inputs. Examples of modes of operation include `CBC` (Cipher Block Chaining) [131] (see also Figure 1.1) and `GCM` (Galois/Counter Mode) [286], which allow the block cipher to

securely encrypt data of arbitrary length while providing additional features such as authentication.

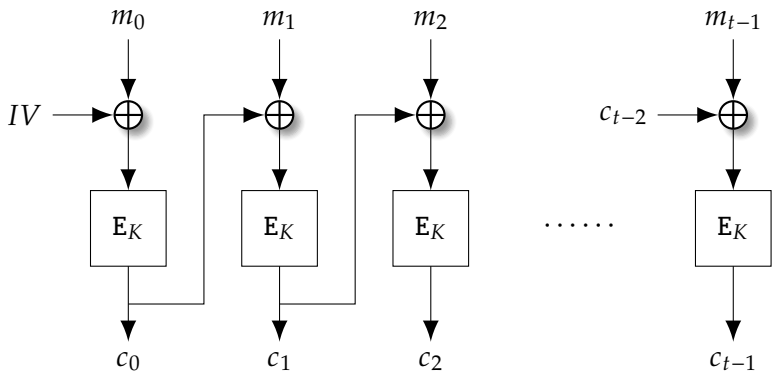

Figure 1.1: *CBC mode of operation [185].*

Security proofs for a mode of operation F built upon a primitive `E` typically involve the following steps:

1. Replace the primitive `E` with an idealized counterpart P according to the security model. For example, in the case of `AES`, the block cipher is often replaced by a uniformly random permutation, as it is widely regarded in the literature as a strong candidate for a pseudorandom permutation (PRP).
2. Prove the security of F when built upon P instead of `E`. This step depends on the specific security goals defined for F. For instance, in the context of `AES-GCM`, the security goal might be show that F is a secure authenticated encryption (AE).

The second step involves an analysis based on the concept of *indistinguishability*. Rather than attempting to break the scheme directly, the adversary's objective is to differentiate between two scenarios: the real world, where they interact with the actual implementation, and an idealized world, where the scheme is replaced by its idealized counterpart as defined by the security model. The security of a scheme is defined by the probability that the adversary can successfully distinguish between the two worlds, also known as the *advantage* of an adversary. Notably, the security proof often holds even against adversaries with unlimited computational power, known as *information-theoretic* adversaries.

We generally assume that the first step has already been established and focus primarily on the second step, which involves providing the actual security proof. To this end, we introduce the various security models used throughout this thesis to abstract the idealized counterparts discussed earlier.

1.3.2 Security Models

In our work, we will use four widespread security models for analyzing cryptographic primitives or constructions. We follow the definitions from Black [56].

The Standard Model. The most common model used in modern cryptography is the so-called “standard model". In this model, we do not use idealized mathematical objects such as infinite random strings or random oracles [38], but rely on complexity-theoretic hardness assumptions. For instance, in public key cryptography, we rely on the hardness of factoring positive integers who are products of large primes, or in the case of symmetric key cryptography, that AES is a pseudorandom permutation [235]. The standard model is usually well-accepted in our community despite the fact that proofs done in this model rest upon unproven assumptions and real world effects: power consumption, error messages and more, have been abstracted away.

The Random Oracle Model. The Random Oracle Model (ROM), formalized by Bellare and Rogaway [38], is an alternative approach when proofs in the standard model are cumbersome, involved or plainly impossible [256]. In the random-oracle model we have a public random function, accessible to all parties, which typically accepts any arbitrary length strings and outputs a n-bit string, which is uniformly random and independent from all other possible outputs. While the Random Oracle Model allows for the construction of provably secure and efficient schemes, it does not imply that the scheme will remain secure when the random oracle is replaced by a concrete real world instance.

The Ideal-Cipher and Random Permutation Models. In certain cases, we cannot prove the security for cryptographic schemes built on top of a block cipher, solely by assuming blockciphers are PRPs in the standard model [297]. In the ideal-cipher model, we think of a block cipher with a k-bit key and n-bit block as been chosen uniformly at random, which is equivalent to having a family of 2^k independent random permutations of $\{0,1\}^n$. The ideal-cipher model differs from the random oracle model in three main points [99, 123]:

- For each key, the ideal-cipher is required to be a permutation, while random oracles do not.
- In the ideal-cipher model, the adversary has access to both the cipher and its inverse.
- In the ideal-cipher the length of the queries are n-bit long whereas for random oracles we allow arbitrary length queries.

Similarly, unlike the random oracle model, which assumes a publicly accessible random function, the random permutation model provides the adversary with access to a random permutation in addition to the construction (or random) oracle(s).

In this work, the primary security model utilized will be the random permutation model. Nevertheless, the security model will be specifically defined by a more precise notion known as the *security game* (see Section 2.2.1 for more details).

1.3.3 Comparing Provable Guarantees

Following Bellare's work [22], we can compare the provable security of cryptographic schemes by evaluating them against two key standards: qualitative assumptions and quantitative assumptions.

Qualitative Assumptions: These assumptions relate to the type and nature of the underlying assumptions that a scheme's security depends on. A qualitative comparison looks at how strong and reliable these assumptions are. For instance:

- *Type of Hard Problem:* Some schemes rely on well-known and trusted problems, like the pseudorandomness of block ciphers such as AES. These are considered strong assumptions because they have been thoroughly studied and are widely believed to be secure. Other schemes might depend on newer or less-tested assumptions, which could be considered weaker or less reliable.
- *Idealized Models:* Some security proofs use idealized models, like the Random Oracle Model (ROM) [38] or the Ideal Cipher Model. While these models offer strong theoretical guarantees, they demonstrate resistance only against generic attacks.

Quantitative Assumptions: These assumptions deal with the numerical parameters and specific bounds provided by the security proof. Quantitative comparisons focus on practical aspects of security, such as:

- *Security Bound:* This refers to the probability that an adversary can successfully attack (or break some security game) the scheme, based on factors like the number of queries, key size, or computational effort. A lower success probability for the adversary means a stronger scheme.
- *Reduction Tightness:* This measures how closely the security of the scheme matches the security of its underlying primitive. A tighter reduction means the scheme's security is nearly as strong as the primitive's, often allowing for more efficient parameter choices (e.g., smaller key sizes for the same level of security).
- *Real-World Resources:* These metrics consider the specific resources (e.g., time, memory) an adversary would need to breach the scheme's security. For instance, a scheme that requires 2^{128} operations to break is more secure than one that needs only 2^{80} operations.

We often present arguments supporting the choice of one security model over another and focus on the quantitative resources available to the adversary, such as the number and length of its queries.

1.3.4 Security Frameworks

Within this work, we use two key frameworks to describe the security of a concrete symmetric-key scheme: *indistinguishability* and *indifferentiability*.

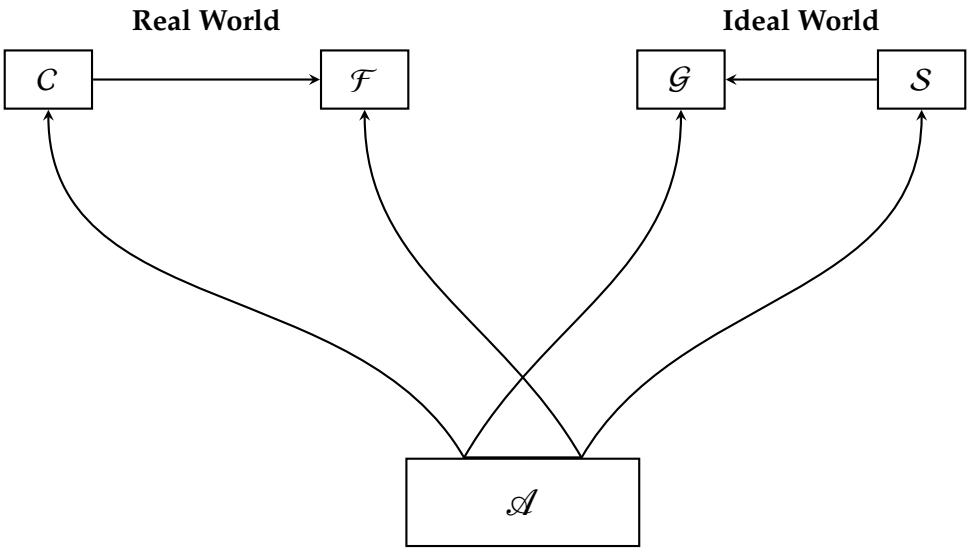

Figure 1.2: *The differentiability setup.*

Indistinguishability. Here, the adversary is tasked with distinguishing between a real-world cryptographic construction and its idealized counterpart. A construction is considered indistinguishable if the adversary's ability to tell them apart is negligible. More broadly, it is argued that if two systems, S_0 and S_1, are indistinguishable, then S_1 can replace S_0 in any application without compromising security, up to some class of adversaries. In Section 2.2.1, we formalize this idea with the notion of a *distinguishing game*.

Indifferentiability. In systems where components are interdependent, an adversary might gain partial information about the randomness or state of these components. For example, in AES, the AES round functions are used as subroutines. The security framework was formalized by Maurer et al. [232] and popularized by Coron et al. [97]. Informally, a construction $\mathcal{C}$ using an ideal primitive $\mathcal{F}$ (e.g. a hash function based on a random compression function) is said to be indifferentiable from another ideal primitive $\mathcal{G}$ (e.g. a random oracle) if there exists a simulator $\mathcal{S}$ accessing $\mathcal{G}$ such that the two systems $(\mathcal{C}^{\mathcal{F}}, \mathcal{F})$ and $(\mathcal{G}, \mathcal{S}^{\mathcal{G}})$ are indistinguishable. Namely, the *differentiability setup* is such that: the adversary $\mathcal{A}$ interacts with the real construction $\mathcal{C}$, which can call the ideal primitive $\mathcal{F}$; or an ideal construction $\mathcal{G}$ and a simulator $\mathcal{S}$, which can query the ideal construction $\mathcal{G}$, as depicted in Figure 1.2. For a formal definition see Section 2.2.2.

1.3.5 (Beyond) Birthday Bound Security

In the context of provable security for symmetric-key schemes, the birthday bound represents a (typically conservative) upper limit on the security of cryptographic constructions, particularly in scenarios involving collision-based attacks. The birthday paradox illustrates that the probability of encountering a collision—where two distinct inputs produce the same output—rises more rapidly than intuition might suggest as the

number of inputs increases. Specifically, for a hash function or block cipher with an output size of n bits, the birthday bound implies that after roughly $2^{n/2}$ operations, the chance of a collision becomes considerable.

This limitation is especially relevant in symmetric-key schemes, where the birthday bound can restrict the effective security of certain primitives. For instance, a mode of operation which uses a block cipher with a 128-bit block size may become vulnerable after approximately 2^{64} queries, regardless of the key size. Hence, the concept of beyond-birthday-bound security—refers to achieving security even against adversaries capable of making more than $2^{n/2}$ queries.

Within this thesis, our objective in the classical setting is to achieve security beyond the birthday bound. However, in the quantum setting, although the quantum birthday bound is reduced to $2^{n/3}$ due to Grover's algorithm [151], achieving beyond-birthday-bound security remains a significant challenge with current techniques for most symmetric-key schemes.

1.4 Leakage-Resilient Cryptography

In the context of modern cryptography, local computations are assumed to be completely private. Adversaries might decrypt a chosen ciphertext, but the decryption processes is typically hidden. Thus, the only information adversaries can access is through well-defined interfaces, like decrypting. Such adversaries are known as "black-box" attackers.

However, real life adversaries do not necessarily adhere to such abstractions. In fact, *side channel attacks* have shown that information about the secret key or the internal state of cryptographic algorithms can be leaked to the adversary. These attacks exploit the fact that cryptographic algorithms are implemented on physical devices, which interact with the environment in measurable ways. Prominent examples of side-channel attacks include exploiting the time taken by an algorithm [202], power consumption [203], and electromagnetic radiation [4]. "Cold boot" attacks [159] recover parts of cryptographic keys from powered-off devices with physical access. The emergence of such attacks led the cryptographic community to rethink the black-box adversary model, resulting in a new field called “leakage-resilient cryptography”.

Due to the physical nature of leakage, the initial countermeasures were typically proposed at low abstraction levels. For example, hardware countermeasures aim to reduce side-channel information by blurring the signal into noise in the time or amplitude domains [84, 225], or by minimizing this signal through special (dual-rail) circuit technologies [308]. These hardware countermeasures can be augmented by implementation-level randomization mechanisms designed to further reduce side-channel leakage. Masking achieves this goal by utilizing data randomization (i.e., secret sharing) [77, 144], while shuffling does so by randomizing the order of operation execution [166, 310]. Significant progress has been made in understanding these different

countermeasures. For instance, masking is supported by a strong theoretical foundation (e.g. [180, 127, 126, 19]). However, securely implementing low-level countermeasures (e.g., masking) remains sensitive to physical defects [226, 258, 98] and is costly in both software and hardware contexts [145, 150].

Given the sensitive and costly nature of hardware and implementation-level side-channel countermeasures, researchers have initiated complementary work focusing on cryptographic primitives that inherently offer improved security against physical leakage. In symmetric cryptography, this trend began with heuristic proposals (e.g. [266, 234]). This concept was formalized by Dziembowski and Pietrzak within the framework of leakage-resilient cryptography [130], inspiring numerous subsequent works and designs. Simple and efficient Pseudorandom generators and stream ciphers were proposed in [268, 322, 321, 299], while PRFs and PRPs can be found in [135, 122, 1].

In this work, we concentrate on the leakage resilience of AEAD schemes. Specifically, we investigate context-committing AEAD schemes in Chapter 6, which are intrinsically linked to leakage resilience. Within this framework, the adversary is allowed to decrypt under multiple contexts, each consisting of a key K, a nonce N and the associated data A (see Section 6.2 for more details).

1.5 Post-Quantum Cryptography

Quantum computers represent a significant leap from classical computers by leveraging the principles of quantum mechanics, such as superposition and entanglement, to perform calculations. Unlike classical computers, quantum computers use quantum bits or qubits, which can exist in multiple states simultaneously. This ability allows quantum computers to solve certain complex problems exponentially faster than classical computers, posing a serious threat to current cryptographic systems.

In 1995, Peter Shor [293] introduced a polynomial-time quantum algorithm for solving prime factorization and discrete logarithm problems, which directly undermines the security of widely-used public-key systems such as RSA [274] and Elliptic-curve cryptography (ECC) [201]. Although the current generation of quantum computers lacks the capability to break popular schemes like RSA with a 2048-bit key, the rapid advancements in quantum technology make it imperative to develop and transition to quantum-resistant alternatives to ensure long-term security.

While public-key cryptography faces the most immediate threat, symmetric-key cryptography is also impacted by quantum computing. Grover's algorithm [151], a quantum algorithm, significantly accelerates brute-force searches by reducing the search time for a database of N entries to $O(\sqrt{N})$. For example, a 128-bit key, which is considered secure against classical attacks, would offer only 64 bits of security in a quantum computing scenario. To maintain adequate security in a post-quantum world, cryptographic best practices recommend doubling the key sizes of symmetric-key algorithms, such as

transitioning from AES-128 to AES-256. This adjustment ensures that symmetric-key cryptography remains robust even in the face of quantum adversaries.

Contrary to the belief that doubling the key is sufficient, a long line of research has shown that (see e.g. [63, 64, 65, 66, 72, 148, 178, 195, 196]) this was not the case, as quantum distinguishers were able to be significantly more efficient than Grover's search for some constructions. This has renewed the interest in formally proving [49, 62, 102, 174, 176, 177, 179, 298, 325] the post-quantum security of symmetric modes of operation or generic constructions.

1.5.1 Post-Quantum Provable Security of Symmetric-Key Schemes

In the context of provable security in the quantum setting two notable attack models are considered when analyzing the security of symmetric-key schemes [323].

1. *The Q1 model [196]:* assumes that the adversary, equipped with a quantum computer, has access to a classical keyed oracle of the target cryptographic scheme. For instance, if the target scheme is a block cipher $\mathtt{E}_k$, the adversary can query the classical encryption oracle with any n-bit string x and receive $\mathtt{E}_k(x)$ in response. The adversary then uses its quantum computer to process these queries and attempt to break the scheme.
2. *The Q2 model:* assumes that the adversary has direct access to a quantum keyed oracle. In this scenario, the quantum oracle allows the adversary to query an arbitrary quantum superposition of $2n$-bit strings. For example, an adversary can query the oracle with a superposition state $\sum_{x,y} \alpha_{x,y}|x\rangle|y\rangle$, and the oracle responds with the state $\sum_{x,y} \alpha_{x,y}|x\rangle|y \oplus \mathtt{E}_k(x)\rangle$.

Although the Q1 model is more realistic as it is up to the oracle to decide whether to accept superposition queries, we believe that symmetric-key schemes should be studied in the Q2 model for a two main reasons. Firstly, as quantum technology advances, the assumption that adversaries could access quantum oracles may become increasingly realistic, particularly in scenarios involving quantum communication or quantum-enhanced cryptographic protocols. Secondly, proving security in the Q2 model provides stronger guarantees; a scheme that is secure in this model is likely to be robust against a broader range of quantum attacks.

Throughout, we analyze security in the quantum setting in the Q2 model.

1.6 Summary of Contributions

This thesis aims to provide improvements in the security and the design of concrete symmetric-key cryptographic schemes. Our contributions can be categorized into three main areas:

- *Improving Security Bounds.* We provide tighter security bounds for symmetric-key schemes, specifically for (tweakable) block ciphers, enhancing their resilience against attacks and improving their practical security.
- *Identifying Design Flaws and Fixing Previous Proofs.* We uncover and address flaws in the design and existing security proofs of certain symmetric-key schemes, highlighting flaws that may have been overlooked in prior analyses.
- *Developing New Proof Frameworks.* We introduce novel frameworks for proving the security of symmetric-key schemes, offering step by step paradigms that improve the clarity and rigor of security proofs.

1.6.1 Analysis of Tweakable Block Ciphers

The first part of this thesis focuses on designing tweakable block ciphers that provide security beyond the birthday bound. In Chapter 3, we discuss how to create a (tweakable) block cipher with a large domain using the substitution permutation network (SPN) design paradigm, from a single smaller permutation. Then, in Chapter 4, we study how to build tweakable block ciphers using public random permutations, with an emphasis on making the key schedule more efficient, using the Tweakable Even-Mansour construction which is a super class of SPN constructions.

1.6.1.1 Beyond Birthday Bound Domain Extenders for (Tweakable) Block Ciphers

In Chapter 3, we study how to design an efficient block cipher with a large domain, say wn-bits, constructed from a few n-bit permutations that achieves beyond birthday bound security. Traditionally, this is accomplished by employing an SPN structure, which involves the following iterative steps:

- Apply a keyed permutation layer to the entire wn-bit state.
- Divide the state into w n-bit blocks.
- Compute an S-box on each block of the state.

After the final step, a keyed permutation layer is applied to the entire wn-bit state. To study the security of such constructions, it usually suffices to prove the soundness of the high-level structure in a relevant security model.

Security of SPNs. The first paper to investigate the security of SPNs models the S-boxes as secret random permutations. Iwata and Kurosawa [181] showed an attack against 2-round SPNs and proved security for 3-round SPNs against non-adaptive adversaries when used with the linear permutation layer from the `SERPENT` block cipher. Miles and Viola [242] studied the security of various SPN-like block ciphers, but their bound gets worse as the number of rounds of the block cipher increases. They also analyzed the security of several SPNs using the AES S-box against various classes of attacks, notably differential and linear attacks. In the *public* permutation model, Dodis et al. [121] proved the birthday-bound security for linear and non-linear SPNs using a single public S-box.

Later, Cogliati et al. [85] studied tweakable SPNs when S-boxes for each round are uniformly random and independent. In particular, they proved that the security of a non-linear tweakable SPN is beyond-birthday-bound when the number of rounds is greater than two, and grows towards optimal security with the number of rounds.

Tweakable Enciphering Schemes. SPNs can be viewed as a domain extender for block ciphers [75, 161]. Indeed, the underlying S-box can be replaced by another block cipher (e.g. with a public random key) or a large permutation in order to obtain a wide block cipher. In other words, (tweakable) SPNs can be seen as (tweakable) enciphering modes of operations. Various such schemes have been proposed with application to disk encryption, where the design principles are classified into three approaches; encrypt-mix-encrypt [163, 164, 160], hash-ECB-hash [75, 161], and hash-CTR-hash [311, 74, 137]. All these constructions typically accept inputs of variable length, and their security is proved up to the birthday bound in the secret permutation model. 2-round SPNs can be viewed as extending the hash-ECB-hash approach, or more precisely, the hash-ECB-hash-ECB-hash approach.

Contributions. The contributions of Chapter 3 are twofold.

- *Beyond-Birthday-Bound Security of Single S-box-based SPNs.* We prove beyond-birthday-bound multi-user security for the 2-round tweakable SPN structure with a single S-box and independent round keys, with the added benefit that the inner linear permutation can be far simpler than the outer linear permutations. More specifically, we rely on the H-coefficient technique [262] (see Section 2.4.2 for more details) and on computational techniques from [80, 158, 171] to prove that the security level of this construction is roughly equivalent to the one of the 2-round SPN structure with two independent S-boxes and a strong inner linear layer; Theorem 3.3.1 indicates that the multi-user advantage of any adversary will be small as long as the number of queries she issues is small in front of $2^{2n/3}$.
- *A New Tweakable Enciphering Scheme.* Our security proof is information-theoretic, and hence it has an inherent limit that the security level cannot go beyond the size of the underlying S-box. In the case of real block ciphers, which are based on very small S-boxes and use many rounds, our results might be too weak to provide any insight. However, when the underlying S-box is instantiated with a secure block cipher such as `AES`, our construction can be viewed as a tweakable enciphering scheme that encrypts wn-bit messages for any integer $w \geq 2$ using $5n$-bit keys (plus one `AES` key) and n-bit tweaks, providing $2n/3$-bit security. We propose an efficient tweakable permutation in the inner permutation layer (as defined in Section 3.2), which might be of independent interest from a practical point of view. To the best of our knowledge, the resulting scheme, dubbed $\texttt{CTET}^{+}$, becomes the first tweakable enciphering scheme that provides beyond-birthday-bound security (with respect to the size of the underlying block cipher) using only a single permutation.

Related Publications. Chapter 3 is derived from our paper [88], which was published in the *IACR Transactions on Symmetric Cryptology* and presented at the *Fast Software Encryption* conference in 2022.

1.6.1.2 Tweakable Even-Mansour with Linear "Tweakey" Mixing

In Chapter 4, we study the design of tweakable block ciphers achieving beyond birthday bound security while still staying efficient. The main issue in the design of tweakable block ciphers on top of a block cipher is that they tend to be either inefficient or only secure up to the birthday bound. Therefore, an alternative design approach is to construct TBCs from low level primitives such as public random permutations. The first work in that direction was done by Goldenberg et al. [139], who showed how to tweak Feistel networks. This was later extended to generalized Feistel ciphers by Iwata and Mitsuda [246].

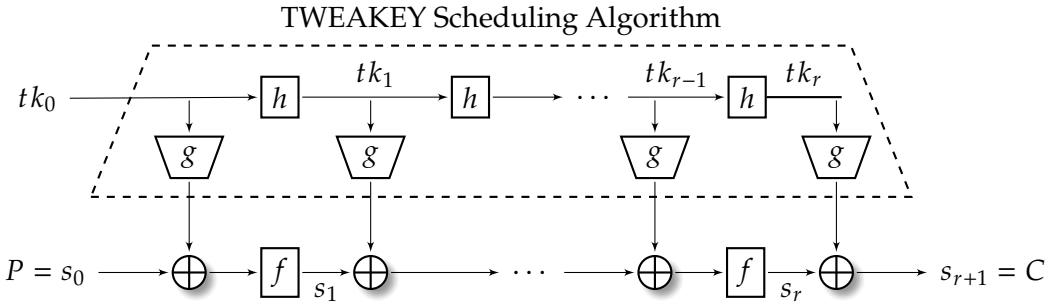

Figure 1.3: *The* `TWEAKEY` *framework [185].*

The `TWEAKEY` Framework. At Asiacrypt 2014, Jean et al. [188] introduced a groundbreaking dedicated design strategy for TBCs called the `TWEAKEY` framework, which revolutionized the design landscape of concrete TBCs. Their idea was to provide a simple framework for designing a tweakable block cipher with any key and tweak sizes. The construction, which is a natural extension of key-alternating ciphers, incorporates a *tweakey* (i.e., a value obtained from the key and tweak inputs) into the internal state at every round of the iterative cipher. One advantage of this framework is that it allows for the creation of a single-key length tweakable block cipher or a double-key length block cipher using the same primitive.

The `TWEAKEY` construction is a framework to build a n-bit tweakable block cipher with τ-bit tweak and κ-bit key. It consists of two states: the n-bit internal state s and the $(\tau + \kappa)$-bit tweakey state tk, and we denote respectively as s_i and tk_i their values throughout the rounds. The state s_0 is initialized with the plaintext P (or ciphertext C for decryption), and h_0 is initialized with the tweak and key material. Then, the cipher is composed of r successive rounds each composed of three steps (see also Figure 1.3):

- a subtweakey extraction function g from the tweakey state, and incorporation of this subtweakey to the internal state;

- an internal state update permutation f;
- a tweakey state update function h.

Namely, $s_{i+1} := f(s_i \oplus g(tk_i))$ and $tk_{i+1} := h(tk_i)$.

From a performance perspective, it is optimal to treat the key and the tweak similarly. Additionally, since the main challenge in designing ad-hoc tweakable block ciphers is the security analysis, this approach would greatly simplify the proof. However, if not done carefully, it could result in an insecure design. Specifically, the main challenge is determining the appropriate number of rounds required to ensure the cipher's security. For large tweak or key sizes, this problem can become intractable. Jean et al. [188] addressed this issue by introducing a subclass of the `TWEAKEY` framework for `AES`-like ciphers, named `STK`.

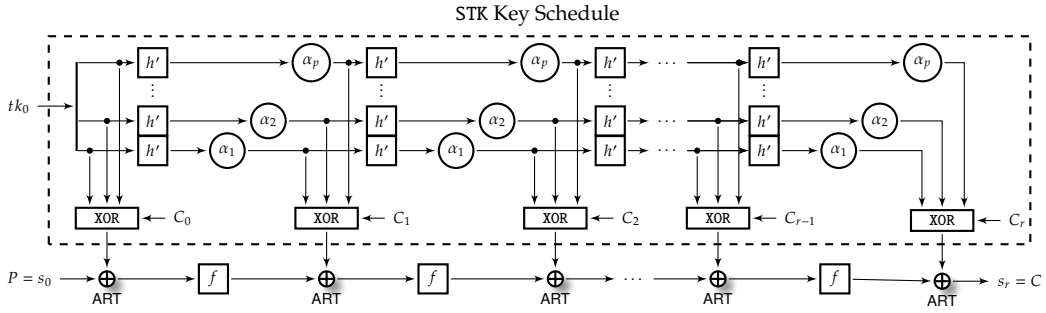

Figure 1.4: *The STK Construction [185].*

The Superposition `TWEAKEY` Construction (`STK`). The `STK` construction is a subclass of the `TWEAKEY` framework for `AES`-like ciphers defined over a finite field $\mathbb{F}_{2^c}$. For $p = (\tau + \kappa)/n$ and assuming the `AES`-like S-Box operates on c-bits, the `STK` construction further specifies the f, g and h functions as follows (see figure 1.4):

- the function g simply XORs all the p n-bit words of the tweakey state to the internal state (denoted ART) and then XORs a round-dependent constant C_i;
- the function h first applies the same nibble position substitution function h' to each of the p n-bit words of the tweakey state, and then multiply each c-bit cell of the j-th n-bit word by a nonzero coefficient α_j in the finite field $\mathbb{F}_{2^c}$;
- the function f is simply a block-cipher round call.

The performance of the `STK` construction is very high due to the simple transformations used in the schedules which are all linear and efficient.

The `TWEAKEY` framework has been the basis of most recent TBCs such as `Deoxys-BC` [186], `Joltik-BC` [186], `Kiasu-BC` [186], and `Skinny` [20]. The success of the TWEAKEY framework also motivated new design frameworks for more specialized usage, such as the Elastic-Tweak framework [73] for TBCs with small tweak size like TweAES and TweGIFT.

Following [186], Cogliati et al. [89] introduced the Tweakable Even-Mansour (TEM) construction which closely follows the high-level design of STK. This construction is based on the key-alternating variant of the famous Even-Mansour construction, known as the Iterated Even-Mansour (IEM). As shown in Figure 1.5, each round involves simply XORing a round key $\gamma_i(k,t) = k_i \in \{0,1\}^n$). Below, we provide a brief overview of the Tweakable Even-Mansour (TEM) construction. For a more comprehensive overview see Section 2.3.2.

Tweakable Even-Mansour. The Even-Mansour (EM) construction [132, 133] is a straightforward method to build a block cipher from a public permutation and two secret keys, achieving birthday-bound security. This approach was later generalized [60] to maintain efficiency while constructing a birthday-bound secure block cipher (see Section 2.3.2 for more details). The first to study how to incorporate tweaks into the IEM construction were Cogliati and Seurin [91], and independently Farshim and Procter [134], who analyzed the simple case with an n-bit key and n-bit tweak. They showed that one can simply XOR the tweak and the key in each round of the IEM construction. However, this approach was found to be insecure for one or two rounds, and for three rounds, it only achieves birthday bound security. Furthermore, based on a result by Bellare and Kohno [33], it is evident that XORing an n-bit key and an n-bit tweak at each round of the IEM construction cannot surpass birthday bound security. Therefore, to achieve beyond the birthday bound security, one should aim for more complex tweak and key mixing functions, at least to prevent the TBC to be of the form $\mathtt{E}(k \oplus t, m)$ for some block cipher E with n-bit key.

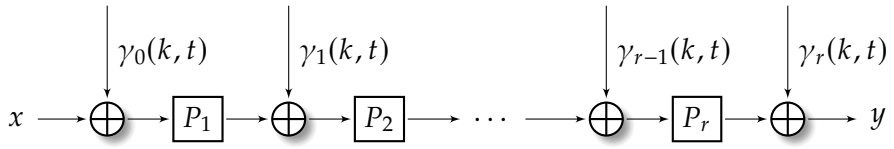

Figure 1.5: *r-rounds TEM based on public permutations $P_1, \ldots, P_r$ and $r+1$ function $\gamma_0, \ldots, \gamma_r$.*

In [89], Cogliati, Lampe and Seurin introduced the Tweakable Even-Mansour (TEM) construction and its cascaded variant. Namely, the r-round Tweakable Even-Mansour construction is built on a tuple of r permutations $\mathbf{P} = (P_1, \ldots, P_r) \in \mathsf{Perm}(n)^r$ and a tuple of $(r+1)$ functions $\gamma = (\gamma_0, \ldots, \gamma_r)$ from $\{0,1\}^{\theta n} \to \{0,1\}^n$, with θn indicating the total size of the combined tweak and key size. It takes as input an θn-bit tweakey (tweak and key) (k,t) and an n-bit input x, and outputs (see also Figure 1.5):

$$\mathtt{TEM}[\gamma,\mathbf{P}]_{k,t}(x) = P_r\left(P_{r-1}\left(\cdots P_1\left(x \oplus \gamma_0(k,t)\right)\cdots\right) \oplus \gamma_{r-1}(k,t)\right) \oplus \gamma_r(k,t).$$

In their work, they considered the non-linear key and tweak mixing function $\gamma_i(k,t) = H_{k_i}(t)$, where H is an AXU hash function and subkeys $k_0, \ldots, k_r \in \{0,1\}^n$ are derived from the master key $k \in (\{0,1\})^{r+1}$. They proved that for $r = 2$ rounds the construction is secure up to $2^{2n/3}$ queries. They also proved that the $2r$-round construction is secure

up to $2^{rn/(r+1)}$ queries. Later, Cogliati and Seurin proved that for $r = 4$ rounds and for linear tweak and key mixing functions the construction, denoted TEML, is secure up to $2^{2n/3}$ queries. Finally, Dutta [128] proved a similar result with a smaller number of independent permutations.

Indifferentiability of (Tweakable) Even-Mansour. As we will see in Section 2.2, proving the (sequential) indifferentiability of a construction is sufficient to assert some degree of resistance against chosen key attacks.

In 2013, Andreeva et al. [8] introduced a variant of IEM, termed EMKD, which employs a random oracle $h : \{0,1\}^{\kappa} \to \{0,1\}^{n}$ to derive the key at each round. They showed the full indifferentiability of this construction for 5 rounds. Concurrently, Lampe and Seurin [210] proposed another IEM variant, known as EMIP, which uses the same key for each round with a trivial key schedule. They proved the full indifferentiability of this construction for 12 rounds.

These results were further refined [106, 156, 155], showing that 3-round EMKD (tight for the sequential indifferentiability) and 5-round EMIP achieve the full indifferentiability. Later, Cogliati et al. [91] showed that 4 rounds of EMIP achieve sequential indifferentiability. Subsequently, Dai et al. [106] showed that it does not achieve full indifferentiability, thereby distinguishing between these two notions.

In a recent study, Xu et al. [316] explored whether the requirement for independent permutations in 4-rounds EMIP (single-key setting) could be relaxed while still achieving sequential indifferentiability. They showed that using a single permutation results in a slide window attack. However, they found that using two permutations in an alternating manner can still achieve sequential indifferentiability.

Interestingly, the works so far, either consider a trivial key schedule, where the round keys are the same as the master key, and hence the master key size is same as the permutation input size, say n, or employ independent random oracles to derive the round keys from the master key. This makes it difficult to employ these results directly to TWEAKEY based block ciphers where the tweakey size is usually bigger than the block size and the tweakey schedule is, in general, linear.

Towards a Theoretical Treatment of The TWEAKEY Framework. While the rationale behind the TWEAKEY framework has been extensively tested through the cryptanalysis of various proposals, it has seen little theoretical analysis. Notably, it is well-known that the high-level design of STK largely follows the Tweakable Even-Mansour construction. However, the only work that provides an asymptotically tight bound for TEM requires the use of AXU hash functions. A typical tweak and key mixing of this type would be the multiplication of the key and the tweak in $\mathbb{F}_{2^n}$, where n denotes the block size. This deviates significantly from the STK construction and presents obvious performance drawbacks. The only works dealing with linear tweak and key mixing currently only consider 4-round constructions. In addition, while STK allows for large-tweak values in

design, there is no suitable theoretical model for analyzing TEML with tweak size larger than n bits. Given the recent push [249, 18, 96] from the community towards leveraging large tweak size, it is necessary to come up with a sound theoretical mechanism for it.

The main goal of Chapter 4 is to study the TEML construction for arbitrary number of rounds and possibly large tweak size. This allows us to give a theoretically sound argument in favor of the resistance of STK-based TBCs against generic attacks in the indistinguishability setup, where the goal is to show indistinguishability from a secret tweakable random permutation assuming a uniform at random and secret key. Moreover, we take the first step by studying the sequential indifferentiability of TWEAKEY block ciphers with arbitrary key size and a special class of linear key scheduling.

Contributions. The main goal of Chapter 4 is to study the TEML construction for an arbitrary number of rounds and potentially large tweak sizes. This allows us to provide a theoretically sound argument supporting the resistance of TBCs based on the TWEAKEY framework against generic attacks in the indistinguishability setup, where the objective is to demonstrate indistinguishability from a secret tweakable random permutation given a uniformly random and secret key. Furthermore, we take the first step in analyzing the sequential indifferentiability, a relaxed variant of indifferentiability (see Section 2.2.2 for a formal definition) of TWEAKEY block ciphers with arbitrary key sizes and a specific class of linear key scheduling, which provides some chosen key resistance as described in Section 2.2.2.

To elaborate, the contributions of Chapter 4 are twofold.

- *Security of TEML with bijective tweakey schedule.* In Section 4.2, using the coupling technique (see Lemma 2.4.2 for more details), we show that a $2r$-round TEML with αn-bit tweak, and weak α-bijective tweak schedule (see Section 4.1.1 for a formal definition), is secure up to $2^{\frac{r-\alpha}{r}n}$ chosen plaintext and chosen ciphertext queries, under the assumption that the round permutations are independent public random permutation and round keys are chosen independently and uniformly at random. Our proof extends the proof strategy used in [212, 89], and introduces, what we call, the *activity patterns* — a succinct string representation of the coupling failure event (see section Section 4.2.3). Basic coupling proofs proceed iteratively by upper-bounding, for each round, the probability to get various collision events using the randomness of the round key. This simple approach can fail if collision events can span several rounds. Looking ahead briefly, a collision between two inputs of the j-th round permutation for the TEML construction with n-bit tweak gives rise to an equation of the form $x_i^j \oplus t_i = x_{i'}^j \oplus t_{i'}$, where the key has been eliminated. This forces us to consider the inputs to the previous round permutation, which may both be involved in another collision, and so on (possibly) down to the first round. In order to solve this issue, we introduce the idea of activity patterns: instead of considering each round individually, we consider the succession of collision events throughout the full evaluation of the construction, and sum over

all possible choices. Our main contribution is to give a fine-grained analysis of the $2r$-TEML construction.

- *Chosen Key Resistance of* ***TEML****.* In Section 4.3, we prove the sequential indifferentiability of (Tweakable) Even-Mansour cipher with rn-bit key (or tweakey) and any weak r-bijective key schedule. Specifically, we show an attack on $r + 2$ rounds and prove the security for $r + 3$ rounds, thereby establishing the necessary and sufficient number of rounds for security in sequential indifferentiability setting. Note that our result directly implies the security of $(r + 3)$-round (Tweakable) Even-Mansour against chosen key attacks.

Related Publications. Chapter 4 is derived from our paper [87], which was published in the *IACR Transactions on Symmetric Cryptology* and presented at the *Fast Software Encryption* conference in 2024.

1.6.2 Analysis of Authenticated Encryption Schemes

The next part deals with the study of authenticated encryption schemes both in real world instances and also more theoretical settings. In Chapter 5, we study the the famous messaging app Telegram and its home-brewed security protocol MTProto, by studying an abstraction of that protocol named MTProto-G, which is a deterministic authenticated encryption (DAE) scheme. While in Chapter 6, we study leakage resilience and context commitment of authenticated encrypted schemes with associated data (AEAD).

1.6.2.1 Subverting Telegram's End-to-End Encryption

Telegram. Over the past two decades, smartphones have become ubiquitous, leading to the widespread adoption of secure messaging services like WhatsApp, Signal, Facebook Messenger, and Telegram. Telegram, in particular, boasts over 900 million active users worldwide and is projected to reach one billion users by the end of 2024 [70]. Its popularity is especially notable among independent journalists and political dissidents, as it asserts independence from governmental control and censorship [222]. Telegram ranks as one of the leading social media platforms globally, following Facebook, YouTube, WhatsApp, Instagram, TikTok, and WeChat [70].

Telegram offers two conversation modes: the cloud chat mode and the secret chat mode. Messages in cloud chats employ client-server/server-client encryption, and are stored on the Telegram server in encrypted form. So, all messages can be read by the server, allowing for chat history accessibility across devices. Messages in the secret chat mode employ client-client or end-to-end encryption for only two parties. In this mode, the messages are sent through the server, but can only be decrypted by the two parties involved in the communication.

MTProto and Its Security. Telegram opted to use a home-brewed original protocol known as MTProto [306], both for cloud chats as well as secret chats. At the heart of this

protocol lies its eponymous encryption scheme `MTProto`. In their online technical FAQ [305], the Telegram team justified the use of an in-house encryption scheme, as opposed to some well-studied and provably secure encryption scheme, as follows:

> *In order to achieve reliability on weak mobile connections as well as speed when dealing with large files (such as photos, large videos and files up to 2 GB each),* `MTProto` *uses an original approach..*

However, the general cryptographic community is still skeptical of Telegram's security claims and justifications. Indeed, their skepticism is not entirely unfounded, as demonstrated by the attacks on `MTProto1.0` [184] by Jakobsen and Orlandi. In response to the attacks in [184], the Telegram team revised the encryption scheme to `MTProto2.0`. In [304], the Telegram team claims that the latest version of `MTProto` achieves IND-CCA [35] security. However, to the best of our knowledge, a formal proof of security was noticeably missing up until 2022 [6]. In fact, the Telegram team goes on to say that security notions like IND-CCA while convenient for theoretical and scientific inquiry, do not directly relate to the actual security of communication [304], and nothing short of a full plaintext recovery or corruption is practically useful. In our opinion, this limited security policy is quite detrimental to the privacy interests of Telegram's users. Indeed, cryptographic literature is filled with examples [48, 301], where theoretical attacks formed the basis for more efficient and practically relevant attacks.

In 2022, Albrecht et al. finally presented a formal IND-CCA security proof [6] for `MTProto2.0`. However, within the proof, they make several non-standard assumptions on the underlying building blocks. To a large extent these assumptions are necessitated by the design choices made in `MTProto2.0`. In addition, Albrecht et al. also propose four attacks on `MTProto2.0` by exploiting some vulnerable behaviors exhibited by Telegram clients and servers in some boundary cases. In response, the Telegram team updated the protocol to mitigate these boundary conditions.

In a recent study, Arx and Paterson [12] demonstrated replay attacks on several popular third-party Telegram clients and identified a theoretical timing attack against a specific client. Their analysis reveals that many third-party clients fail to securely implement `MTProto2.0`. This finding underscores the importance of examining both the implementation and theoretical design when analyzing a security protocol.

Subversion Attacks. The veiled use of mass surveillance and web traffic interception by government agencies became apparent due to the Snowden revelations. Among other things, it revealed that the government agencies do not just apply intensive cryptanalytic techniques, but also subvert cryptosystems to bypass well established cryptographic algorithms. One such mechanism for subversion is to manipulate the algorithms used in implementations by injecting a backdoor into otherwise secure implementations. A formal treatment of such mechanisms predates the Snowden revelations, and was initiated in a line of work by Young and Yung that they named

kleptography [319, 320]. Basically, Young and Yung considered an adversary who designs a subverted cryptographic algorithm whose outputs are computationally indistinguishable from the outputs of an unmodified algorithm. The subverted algorithm should leak the secret key through the output, which was achieved using principles similar to Simmons' *subliminal channels* [295].

The Snowden revelations reignited interest in this kind of subversion attacks, starting with the so-called *Algorithm Substitution Attacks* (ASAs) by Bellare et al. [36] against randomized encryption schemes. Their attack relies on influencing the randomness generated in the course of encryption. Specifically, the attack applies to a sub-class of randomized schemes satisfying a property they call *coin-injectivity*. Degabriele et al. criticized [111] the perfect decryptability condition required from the subverted ciphertext in BPR's model. Bellare et al. improved over the attacks in [36], proposing stateless attacks [31] against all randomized schemes. While previous attacks [36, 31] targeted the encryption algorithm, Armour and Poettering proposed an attack [11] by subverting the decryption algorithm. Hodges and Stebila explored the detectability of ASAs via state resetting [173]. Apart from these attacks on (authenticated) encryption schemes, ASAs have also been proposed on message authentication code [10], signature schemes [13, 15, 218], and key encapsulation mechanisms [78]. Additionally, Russell et al. consider ASAs on (trapdoor) one-way functions and key generation, as well as a generic way to defend randomized algorithms against ASAs [283, 284, 285].

ASAs were conceptualized to model government sponsored eavesdropping on real world protocols with millions of active users. So, it is just natural to explore these attacks against secure messaging services like WhatsApp and Telegram. Recently, Berndt et al. studied [42] the feasibility of ASAs on three popular protocols: `TLS`, `WireGuard`, and most notably `Signal`— the cryptographic protocol used in several messaging apps, including WhatsApp and Signal. To the best of our knowledge, such studies have not been conducted on Telegram's `MTProto` protocol.

Our Motivation. As pointed out by the Telegram security team in a private conversation, the code of all their official apps is open source and their builds are reproducible. This obviously makes massive subversion attacks against the Telegram official clients difficult to roll out. However, targeted attacks at individuals could still be deployed. Moreover, closed-source (or open source without reproducible builds) third-party clients would be easy to subvert. This second scenario is our main motivation in this work: is it possible to mount an efficient subversion attack against the authenticated encryption scheme that is used in Telegram?

Our Contributions. To elaborate, the contributions in Chapter 5 are twofold.

- We propose the first partial key recovery ASAs (see Section 5.4) on the secret chat mode of Telegram. Our attacks are completely passive in nature and incur significantly less latency as compared to previous such attacks on generic

authenticated encryption schemes [31, 11]. Our attack exploits the random length padding used in the `MTProto2.0` encryption. Strangely, each official client (desktop, android, iOS, `tdlib` library) uses different padding algorithms. Our attack can be mounted with the padding algorithm of the desktop client and the tdlib library (which can be used by third-party clients). As per our undetectability proofs (see Theorem 5.4.2), our subverted algorithms are indistinguishable from Telegram's original encryption algorithm from the desktop client or the `tdlib` library (depending on which one has been chosen for the corrupted client).

- We propose a minor change in the definition of `MTProto2.0`, that ensures all the advantages of the existing algorithm, and thwarts the proposed key recovery attack. In fact, we show that the modified algorithm is subversion-resistant in most of the practical scenarios. This is done in three steps.
  1. We show that an abstraction of `MTProto2.0`, called `MTProto-G`, is a secure deterministic authenticated encryption (DAE) scheme.
  2. We make three small changes, mainly in the padding algorithm of `MTProto2.0`, to make the protocol deterministic.
  3. Under the assumptions of perfect decryptability and key-independent messages, we show that the modified protocol, called `MTProto-D`, is subversion-resistant in context of algorithm substitution attacks, resulting in a more secure solution for Telegram.

Related Publications. Chapter 5 is derived from our paper [86], which was published in the *IACR Transactions on Symmetric Cryptology* and presented at the *Fast Software Encryption* conference in 2023.

1.6.2.2 Context-Committing Security of Authenticated Encryption Schemes

As we will see in Section 2.3.5, Authenticated Encryption with Associated Data (AEAD) has become a fundamental component in modern security applications, providing both confidentiality and authenticity. The development of efficient AEAD schemes has led to widespread adoption of constructions such as: `GCM` [286], `Ascon` [118], `Deoxys` [187], `AES-GCM-SIV` [152]. These constructions address diverse security goals, including Nonce-based AEAD [278], Misuse-resistant AEAD, and Deterministic AEAD [280].

However, as AEAD schemes and their analyses mature, attackers continuously seek new ways to exploit their security. Additionally, new applications introduce fresh security challenges. Consequently, two areas of research have gained prominence:

1. *Leakage-resilient* ***AEAD***: This area focuses on security notions and schemes where the adversary can observe different forms of auxiliary leakage that may depend on sensitive or secret information. The objective is to construct schemes that maintain confidentiality and authenticity even in the presence of certain leakages (see also Section 1.4).

2. *Context-committing* **`AEAD`**: This area addresses scenarios where the adversary has access to, and can manipulate, secret keys. For example, it deals with situations where the ciphertext allows for correct decryption under multiple contexts, where a context consists of the key K, the nonce N, and the associated data A.

1.6.2.3 Leakage-resilient and Context-Committing AEAD

Leakage-resilient AEAD. Leakage-resilient AEAD has been a burgeoning research area for nearly two decades. In Chapter 6, we focus on recent developments, particularly on the schemes discussed by Bellizia et al. in [41]. In their work, the authors categorized modern leakage-resilient AEAD schemes into four grades, emphasizing so-called leveled implementations. In these schemes, a few functions are assumed to be either leak-free or heavily protected, while the rest of the construction can leak significant amounts of information. Here, we focus on the so-called `Grade-2` schemes. These schemes typically employ a single-pass AEAD scheme, a hash function, and two heavily protected tweakable block cipher (TBC) calls. They target Ciphertext Integrity with Misuse and Decryption Leakage (CIML2) security and Indistinguishability against Chosen-Ciphertext Adversaries with Misuse Resilience and Encryption Leakage (CCAmL1), see Section 2.3.5 for more details.

Context-Committing AEAD. In recent years, a series of attacks, such as the Facebook message-franking attack [120] and the partitioning-oracle attack [215], have exposed vulnerabilities in the use of conventionally secure AEAD schemes. These attacks share a common root cause: the existence of ciphertexts that can be decrypted correctly under multiple keys. This issue falls outside the scope of conventional AEAD security but is critical for practical security in these specific use cases.

To bridge this gap, Bellare and Hoang introduced the concept of commitment security in [30], which ensures that each ciphertext commits to the key (CMT-1) or the entire context (CMT-4) that generated it. Among the proposed notions, CMT-4 is the strongest and therefore the most desirable for designers. It is formalized through the following game: Given an AEAD scheme $\mathcal{E}$, an adversary is tasked with providing two contexts, i.e., tuples (K, N, A, M) and (K', N', A', M'), consisting of a key, nonce, associated data, and message each. The adversary wins the game if the contexts differ, i.e., $(K, N, A, M) \neq (K', N', A', M')$, but both encrypt to the same ciphertext: $\mathcal{E}^+(K, N, A, M) = \mathcal{E}^+(K', N', A', M')$.

Connecting The Two Notions. At first glance, the overlap between leakage-resilient and context-committing AEAD schemes is unclear, as their potential synergies have not been extensively explored. However, Struck and Weishäupl [303] have begun to illuminate this area by investigating the generic compositions of Encryption and Message Authentication Code (MAC) schemes to develop schemes that are both leakage-resilient and committing. Their study revealed that Encrypt-then-MAC (`EtM`) and

MAC-then-Encrypt (MtE) schemes are generally not committing. Conversely, they demonstrated that Encrypt-and-MAC (EaM) can achieve committing properties under relatively weak assumptions about the underlying schemes. Additionally, they presented a transformation method that converts an AEAD scheme into one that is both leakage-resilient and context-committing.

Our Contributions. While black-box compositions such as EtM, EaM, or MtE are valuable for studying generic constructions and inspiring specific implementations, real-world schemes often deviate from these models. Many leakage-resilient schemes, for example, are based on blueprints that incorporate small, targeted changes to meet specific security goals or to improve efficiency. We introduce a blueprint targeting leveled single-pass implementations. We demonstrate that with a careful selection of underlying primitives, this blueprint can be made committing. This blueprint enables us to establish that leakage-resilient schemes can also be committing. For instance, the single-pass scheme Triplex.

For schemes that adhere to our blueprint, demonstrating their context-committing security reduces to proving the collision resistance of the core components: Key Derivation (KDF), Encryption, and Tag Generation (TGF). Our analysis imposes a few cryptographic assumptions on the components used. For keyed primitives, we operate in the ideal-cipher model, which is necessary in the chosen-key setting of committing security. For hash functions and compression functions, we require either collision and everywhere-preimage resistance, or collision resistance alone. While our assumptions are slightly stronger, they align closely with practical standards. Everywhere-preimage resistance can be viewed as a worst-case scenario analysis of range-oriented preimage resistance, thus it is expected to provide a similar security bound for any robust standard hash function.

Related Publications. Chapter 6 is derived from Sections 3.2,6 and 7 in our paper [113], which was published in the *IACR Transactions on Symmetric Cryptology* and will be presented at the *Fast Software Encryption* conference in 2025.

1.6.3 Post-Quantum Provable Security of Symmetric-Key Schemes

Finally, the last part of this thesis focuses on the analysis of symmetric-key schemes in the quantum setting.

In Chapter 7, we study $2n$-bit to n-bit compression functions that follow an SPN-like structure, characterized by a single non-linear function, or "call," sandwiched between two linear layers. We characterize the security of compression functions with two or three calls, assessing their vulnerabilities to both classical and quantum attacks. Additionally, we show the quantum PRF (qPRF) security for a selected set of efficient three-call constructions, using a new framework for proving the indistinguishability of symmetric-key schemes in the quantum setting.

In Chapter 8, we examine the limitations of the proof framework introduced in Chapter 7. We identify a critical flaw in the security proofs of the Luby-Rackoff construction, a well-known Feistel network, and demonstrate how this issue can be mitigated by focusing on non-adaptive adversaries. On a positive note, we also establish the quantum PRF (qPRF) security of the `Misty` constructions, a variant of Feistel networks, even when considering adaptive adversaries.

1.6.3.1 Post-Quantum Secure Compressing PRFs

One of the most studied primitive in symmetric-key cryptography is the block cipher. Thanks to the classical PRP-PRF Switching Lemma, block ciphers are known to be secure PRFs in the classical setting as long as the number of adversarial queries is small in front of $2^{n/2}$, where n denotes the block-size. In the quantum setting, this bound degrades to $2^{n/3}$ [324], which can be seen as the quantum equivalent of the so-called birthday bound. Block ciphers can also be used to build other primitives, such as authenticated encryption schemes, or message authentication codes (MACs), that are secure in the classical sense. Among these primitives, $2n$-bit-to-n-bit PRFs are a key component in building higher-level optimally-secure (in the classical sense) schemes. Indeed, combining a universal $2n$-bit hash function with a $2n$-bit-to-n-bit PRF yields an n-bit secure variable-input-length PRF, which can be used as it is as a deterministic MAC, or to construct an optimally secure authenticated encryption scheme using the SIV construction [281].

While these composition results do not yet exist in the quantum world, constructing a (quantum secure) contracting PRF from a block cipher is a key component in building more sophisticated algorithms. A first step in this direction has been taken by Hosoyamada and Iwata — after developing a variant of Zhandry's compressed oracle [325] in [174], they proved that the `LRWQ` construction, defined by the mapping

$$(x_1, x_2) \longmapsto \texttt{LRWQ}(x_1, x_2) := f_3(f_1(x_1) \oplus f_2(x_2)),$$

where f_1, f_2, f_3 are random n-bit functions, is a (quantum) secure PRF as long as the number of queries is small in front of $2^{n/4}$ in [177]. Since this construction uses three PRF calls, two natural questions arise from this result:

- can a construction using only two PRF calls be proven secure?
- does there exist any other secure construction using three PRF calls?

It is worth noting that these questions have conclusively affirmative answers (see fixed-length CBC-MAC [32]) in the classical setting. We aim to answer the two questions in the quantum settings.

Our Contributions. To elaborate, our contributions in Chapter 7 are twofold.

- We study all possible $2n$-bit-to-n-bit PRFs that are built using two or three PRF calls, and only linear function, as described above. We prove that all the 2-call

constructions are either classically broken, or vulnerable to a quantum period-finding distinguisher. Furthermore, we identify classes of 3-call constructions that are vulnerable to attacks, and categorize candidates that may be secure.

- We prove the qPRF security of the following select constructions:

$$\texttt{TNT}(x_1, x_2) := f_3(x_2 \oplus f_2(x_2 \oplus f_1(x_1)));$$
$$\texttt{LRWQ}(x_1, x_2) := f_3(f_1(x_1) \oplus f_2(x_2)).$$

  We adapt the rigorous formulation of Zhandry's compressed oracle technique [325] by Chung et al. [83] in the indistinguishability setting. Using this framework, we prove that the two constructions are secure quantum PRFs as long as the number of adversarial queries is small in front of $O(2^{n/5})$. As a byproduct, we infer that by combining our main result and [175, Proposition 5], we can prove that the aforementioned two constructions (including TNT [17]) are quantum-secure TBCs against chosen plaintext attacks as long as the number of adversarial queries is small in front of $O(2^{n/6})$. We note that our combination of Hosoyamada and Iwata's proof strategy and Chung et al. framework leads to compact proofs that look mostly classical in nature. As a comparison, we derive a similar security bound for LRWQ as Hosoyamada and Iwata [177], albeit without the heavy computations from that work.

Related Publications. Chapter 7 is derived from our paper [50], which was published in *Advances in Cryptology* and presented at the *ASIACRYPT* conference in 2023.

1.6.3.2 Flaws in Post-Quantum Security Proofs for The Adaptive Setting

Classically, most of the well-known symmetric cryptographic algorithms are constructed as a mode of operation over fixed length primitives that are instantiated with either a pseudorandom[1] permutation (PRP) or function (PRF).

Some well-known examples of generic PRP constructions include the Luby-Rackoff cipher [221], Lai-Massey [207] and the generic Misty ciphers [228, 248] . Of these the former two constructions can be instantiated by any primitive (function or permutation), while the latter solely works with permutations. In general, PRP-based constructions are preferred as they can be directly instantiated with well-analyzed block ciphers. On the other hand, PRF based constructions are usually easier to analyze in security proofs. Indeed, many security proofs involve the boilerplate switching lemma [37, 294]: replace PRP calls with PRF calls with a factor of $O(q^2/2^n)$ per call, where q and n denote the number of queries and output size, respectively. Thus, all of the above mentioned constructions are classically secure birthday-bound PRFs. On the other hand more recent efforts have focused on building beyond-the-birthday-bound secure PRP-to-PRF constructions, starting with the well-known sum of permutations [34, 165] and the

[1] the fixed-length permutation /function is keyed, efficiently computable, and indistinguishable from a uniform random permutation/function.

truncation of permutation [165] to the more recent encrypted Davis-Meyer [93] and its dual [239]. The analysis of these PRP and PRF constructions lead to a great advancement in the provable security research, mushrooming several new proof techniques such as the H-coefficient technique [262, 172], mirror theory [260, 261, 93] the χ^2-technique [104], and the recent use [116] of Fourier analysis by Dinur to prove the exact security of sum of permutations.

The Compressed Oracle. In the quantum setting, however, the research on the security of these well-known constructions is still in the rudimentary stage. While there are some generic attacks on Luby-Rackoff [206, 174] and Misty [146], on the security proofs front the results are still far from tight even in the quantum birthday-bound (up to $2^{n/3}$ queries). Having said that, the situation has changed in recent years, largely due to Zhandry's compressed oracle technique [325] — an elegant way to lazy sample a random function. Indeed most recent security proofs [174, 176, 177, 50] in symmetric cryptography relied on the compressed oracle [325] and its variants respectively introduced by Hosoyamada and Iwata [174] and Chung et al. [83].

When proving the indistinguishability of a construction C based on PRFs from an actual random function, proofs usually rely on the following steps:

- describe the random function as a construction that bears a similar structure to C, but such that some of its inputs are augmented by the adversarial queries in order to ensure their uniqueness, and hence the uniformity of the output;
- describe "bad events" on the output of intermediate function calls that trigger input collision in later calls;
- upper-bound the probability of triggering such bad events;
- describe a one-to-one mapping between intermediate values in both constructions when no bad event occurred.

We emphasize that it is critical for the proof that these bad events are only described using inputs and outputs that were recorded by the compressed oracle. In particular, some information may be lost, such as the actual adversarial query, or which input-output pairs belonged to the same query.

Our Contributions. To elaborate, our contributions in Chapter 8 are threefold.

1. We revisit identify several flaws in previous works. They relate to the aforementioned one-to-one mapping: as an example, in the 4-round Luby-Rackoff security proof [174], the authors cannot prevent bad collisions without relying on information that is not present in the compressed oracle entries. We spotted similar flaws in other works in [177, 227].
2. We prove the security of `Misty` schemes in the quantum setting using the framework developed in Chapter 7. In more details, we prove that the 4-round `MistyR` (resp. 5-round `MistyL`) constructions are secure up to $2^{n/5}$ (resp. $2^{n/7}$) adversarial queries,

where n denotes the size of the underlying permutation. We note that, in both cases, this corresponds to the minimum number of rounds to achieve an exponential bound in n, since period-finding attacks based on Simon's algorithm exist for the 3-round `MistyR` (resp. 4-round `MistyL`) constructions [146].

3. Finally, we propose a new security proof for the 4-round Luby-Rackoff construction in the non-adaptive setting: the adversary has to prepare all of its queries in advance, and receive the corresponding outputs at once. By using an artificial dummy database call on all the adversary's inputs, this allows us to mitigate the issue from [174], since now the database contains all the necessary information to handle the bad events.

Related Publications. Chapter 8 is derived from our paper [51], which will be published in *Advances in Cryptology* and presented at the *ASIACRYPT* conference in December 2024.

1.7 Thesis Layout

The thesis is organized as follows: Chapter 2 establishes the foundational elements, including the mathematical and cryptographic notations, formalizes key security concepts, introduces essential primitives, and presents the proof techniques utilized throughout the thesis. The core contributions are divided into three main parts: the analysis of tweakable block ciphers, detailed in Chapter 3 and Chapter 4; the analysis of authenticated encryption schemes, presented in Chapter 5 and Chapter 6; and post-quantum provable security for symmetric-key schemes, discussed in Chapter 7 and Chapter 8. Finally, Chapter 9 wraps up our work and offers perspectives on future research directions. Additionally, Appendix A provides supplementary results that, while not directly related, support the theoretical development of the final part. The remainder of this section offers a detailed overview of the previous works followed by our contributions from each part. Finally, we provide a list of our papers upon which this thesis is based.

1.7.1 List of Publications

This thesis is based on the following research papers:

1. Cogliati, B., Ethan, J., Lallemand, V., Lee, B., Lee, J., Minier, M.: CTET+: A beyond-birthday-bound secure tweakable enciphering scheme using a single pseudorandom permutation. IACR Trans. Symm. Cryptol. 2021(4), 1–35 (2021)
2. Cogliati, B., Ethan, J., Jha, A.: Subverting telegram's end-to-end encryption. IACR Trans. Symm. Cryptol. 2023(1), 5–40 (2023)
3. Cogliati, B., Ethan, J., Jha, A., Saha, S.K.: On large tweaks in tweakable Even-Mansour with linear tweak and key mixing. IACR Trans. Symm. Cryptol. 2023(4), 330–364 (2023)

4. Bhaumik, R., Cogliati, B., Ethan, J., Jha, A.: On quantum secure compressing pseudorandom functions. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part III. LNCS, vol. 14440, pp. 34–66. Springer, Singapore (Dec 2023)
5. Dhar, C., Ethan, J., Jejurikar, R., Khairallah, M., List, E., Mandal, S.: Context-committing security of leveled leakage-resilient aead. IACR Transactions on Symmetric Cryptology 2024(2), 348–370 (Jun 2024)
6. Bhaumik, R., Cogliati, B., Ethan, J., Jha, A.: Mind the bad norms: Revisiting compressed oracle-based quantum indistinguishability proofs. Cryptology ePrint Archive, Paper 2024/1478 (2024), `https://eprint.iacr.org/2024/1478`

Additionally, I have published another paper on topics outside the scope of this thesis:

1. Cogliati, B., Ethan, J., Jha, A., Nandi, M., Saha, A.: On the number of restricted solutions to constrained systems and their applications. Cryptology ePrint Archive, Paper 2024/1163 (2024), `https://eprint.iacr.org/2024/1163`

Publication Order. In accordance with the common practice in the field of cryptography, the authors are listed in alphabetical order by their surnames.

2

Preliminaries

2.1 General Definitions and Notions

Integers, Reals and Complex Numbers. Let $\mathbb{N}$ denote the set of all positive integers, and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. Similarly, let $\mathbb{R}$ and $\mathbb{R}_0$ be the set of all reals and non-negative reals respectively. For any two positive reals $a \leq b$, we denote by $[a, b]$ the set of all real numbers x such that $a \leq x \leq b$. Let $\mathbb{C}$ be the set of all complex numbers. For $\alpha = a + bi \in \mathbb{C}$, we denote by $\mathrm{Re}(\alpha) := a$ its real part, $\mathrm{Im}(\alpha) := b$ its ideal part and by $\alpha^* := a - bi$ its complex conjugate. The absolute value of α is a real number defined as $|\alpha| := \sqrt{a^2 + b^2}$.

Functions and Sets. For $i \leq j \in \mathbb{N}_0$, $[i; j]$ denotes the set $\{i, \ldots, j\}$. For $n \geq k \in \mathbb{N}_0$, the falling factorial is defined as $(n)_k := n!/(n-k)!$. For a set $\mathcal{X}$ and an integer $s \geq 1$, we write $\mathcal{X}^{*s}$ the set of all sequences that consist of s pairwise distinct elements of $\mathcal{X}$. Additionally, if $|\mathcal{X}| = r$, then $(r)_s = |R^{*s}|$. For two sets $\mathcal{X}$ and $\mathcal{Y}$, the set $\mathsf{Func}(\mathcal{X}; \mathcal{Y})$ denotes the set of all functions $f : \mathcal{X} \to \mathcal{Y}$ and $\mathsf{Perm}(\mathcal{X})$ denotes the set of all permutations on $\mathcal{X}$.

Alphabet and Strings. An alphabet Γ is a finite non-empty set of symbols. For any integer $n \in \mathbb{N}$ and alphabet Γ, Γ^n denotes the set of all strings $x = x_1 \ldots x_n$ where for any $i \in [1; n]$, $x_i \in \Gamma$. Moreover, we say x has length n, denoted $|x| = n$. Let $\Gamma^0 = \{\varepsilon\}$ and $\Gamma^* = \bigcup_{n \in \mathbb{N}_0} \Gamma^n$. For any string $x \in \Gamma^n$ and any two positive integers $1 \leq i < j \in [1; n]$, the string $x[i, j]$ is defined as $x[i, j] := x_i \ldots x_j$ and $x[i] := (x_1, \ldots, x_i)$. For any two strings, $x, y \in \Gamma^*$, the *concatenation* of x and y is denoted by $x \parallel y$. We say that string $x \in \Gamma^*$ is a *prefix* of $y \in \Gamma^*$ if there exists $z \in \Gamma^*$ such that $x \parallel z = y$ and that x is a *proper prefix* of y if $x \neq y$. For $s \in \mathbb{N}$ and symbol $\sigma \in \Gamma$, σ^s denotes the string $\sigma \parallel \cdots \parallel \sigma \in \Gamma^s$.

Particularly, for any $n \in \mathbb{N}$, we denote by $\{0,1\}^n$ the set of all binary strings of length n. For any $x \in \{0,1\}^n$ we define the hamming weight of x by $HW(x) := |\{i \in [1; n] : x_i = 1\}|$. For any two strings $x, y \in \{0,1\}^n$ we define the *dot product* of $x = x_1 \ldots x_n$ and $y = y_1 \ldots y_n$ as $x \cdot y := (x_1 \wedge y_1) \oplus \ldots \oplus (x_n \wedge y_n)$ where $\wedge$ denotes the logical AND operation and $\oplus$ is the logical XOR operation. For any two positive integers $j, k \in \mathbb{N}$ such that $j \in [0; 2^k - 1]$, we write $\langle j \rangle_k \in \{0,1\}^k$ as the binary representation of j in k bits.

For ease of notations, for $n, m \in \mathbb{N}$, let $\mathsf{Perm}(n) := \mathsf{Perm}(\{0,1\}^n)$ and $\mathsf{Func}(n;m) := \mathsf{Func}(\{0,1\}^n;\{0,1\}^m)$.

Finite Fields and Rings. Throughout, we assume all rings are unitary and commutative. For a positive integer $n \in \mathbb{N}$, we denote by $\mathbb{F}_{2^n}$ the finite field consisting of 2^n elements. Each element $\alpha \in \mathbb{F}_{2^n}$ can be uniquely represented by a polynomial $\alpha(x) = \sum_{i=0}^{n-1} a_i x^i$ where $a_i \in \mathbb{F}_2$. The addition $\alpha \oplus \beta \in \mathbb{F}_{2^n}$ is equivalent to the bit-wise XOR of their coefficients. The product $\alpha \cdot \beta \in \mathbb{F}_{2^n}$ is computed as $\alpha(x) \cdot \beta(x) \bmod p(x)$, where $p(x)$ is some fixed irreducible polynomial. We identify the finite field $\mathbb{F}_{2^n}$ with the set of binary strings $\{0,1\}^n$, unless stated otherwise.

Probability Space. A finite probability space is a pair $(\Omega, \Pr)$ where Ω is a non-empty finite set and $\Pr : \Omega \to \mathbb{R}$ is a function, satisfying two properties:

- For all $x \in \Omega$, $\Pr(x) \geq 0$;
- $\sum_{x \in \Omega} \Pr(x) = 1$.

For simplicity, we usually refer to Ω as the probability space when the probability function is clear from the context. For an event $\mathsf{E} \subseteq \Omega$, let $\overline{\mathsf{E}} := \Omega \setminus \mathsf{E}$ be the complement of E. Finally, we denote by $\Pr(\mathsf{E})$ its probability (resp. $\Pr\left(\overline{\mathsf{E}}\right)$ for the probability of its complement). A random variable is simply a function $\mathtt{X} : \Omega \to \mathbb{R}$. The expectation of $\mathtt{X}$ is denoted by $\mathbb{E}(\mathtt{X})$, and its variance is denoted by $\mathbb{V}(\mathtt{X})$.

Probability Distributions. We say a random variable $\mathtt{X}$ is distributed according to a probability distribution $D = \{p_1, \ldots, p_k\}$, if the support of $\mathtt{X}$, $\mathrm{Sup}(\mathtt{X}) = \{x_1, \ldots, x_k\}$, satisfies $p_{\mathtt{X}}(x_i) = \Pr(\mathtt{X} = x_i) = p_i$ for every $i \in [1;k]$. The function $p_{\mathtt{X}}$ is often called the *probability mass function* of distribution D. For a distribution over domain $\mathcal{X}$ whose probability mass function is p, we define three notions of entropy on p:

- Shannon entropy: $H(p) = -\sum_{x \in \{0,1\}^n} p(x) \log(p(x))$;
- Min-entropy: $H_\infty(p) = -\log\left(\max_{x \in \{0,1\}^n} p(x)\right)$;
- Collision entropy : $H_2(p) = -\log\left(\sum_{x \in \{0,1\}^n} p^2(x)\right)$,

(where log denotes the logarithm function in the binary base). In this thesis, our primary focus will be on min-entropy, which quantifies the unpredictability of a set of outcomes. Moreover, the following distributions will be of particular interest:

- *The Uniform Distribution* is a symmetric probability distribution wherein a finite number of values are equally likely to be observed. Namely, for a finite set X, the uniform distribution $\mathcal{U}$ on X is defined as $\mathcal{U}(x) = \frac{1}{|X|}$, for any $x \in X$ (we omit the set X when it is clear from the context). In this case, we say x is sampled uniformly at random from X and denote $x \twoheadleftarrow_{\$} X$.

- *The Binomial Distribution* with parameters $n \in \mathbb{N}_0$ and $p \in [0,1]$ is the probability distribution of the number of successes in a sequence of n independent experiments, each asking a yes–no question, and each with its own Boolean-valued outcome: success (with probability p) or failure (with probability $1 - p$). Formally, the binomial distribution with parameters n and p as above, denote by $\mathcal{B}_{n,p}$, is defined using the formula,

$$\mathcal{B}_{n,p}(k) = \binom{n}{k} p^k (1-p)^{n-k},$$

  where $k \in [0; n]$ is the number of successes. When $n = 1$ it is often referred to as the *Bernoulli distribution*.
- *The Hypergeometric Distribution* is a probability distribution that describes the probability of k successes (random draws for which the object drawn has a specified feature) in n draws, without replacement, from a finite set of size N that contains K objects with that feature, wherein each draw is either a success or a failure. Formally, the hypergeometric distribution with the parameters as above, denote by $\mathrm{Hyp}_{N,K,n}$ is defined by the formula,

$$\mathrm{Hyp}_{N,K,n}(k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}},$$

  where $\max(0, n + K - n) \leq k \leq \min(n, K)$ and $K, n \in [0; N]$.

2.2 Cryptographic Security

This section outlines the security concepts that are foundational to this thesis. We begin by exploring general security notions such as distinguishing games and adversaries, along with more specific security frameworks or notions. Throughout, we clarify the details for particular scenarios to ensure everything remains straightforward and understandable.

2.2.1 Distinguishing Games and Adversaries

In symmetric cryptography, the security of a particular scheme is frequently analyzed through the framework of security games. Formally, a security game is a an interactive game between an *adversary* $\mathcal{A}$, who is trying to break the scheme and an *oracle* O, who has access to the real scheme or some idealized counterpart.

Adversaries and Oracles. An oracle is simply an interface to some function. An adversary is an interactive Turing machine that has black box access to a set of oracles. For an oracle O, $\mathcal{A}^O$ denotes the output of $\mathcal{A}$ after its interaction with O. Finally, based on $\mathcal{A}$ response to O, the security game ends in success or failure.

Distinguishing Games. In this thesis, we concentrate on distinguishing games with an optional set of extra restrictions to reflect the exact security goal. Namely, some security games will allow bidirectional access to the underlying permutation of a given oracle O and its inverse, denoted $O^{\pm}$. In a *distinguishing game* the adversary $\mathscr{A}$, also known as the *distinguisher*, tries to distinguish between two sets of oracles, O_0 and O_1. It interacts with a set of oracles O_b for some $b \in \{0,1\}$ and outputs a single bit b' as a response. We say an adversary $\mathscr{A}$ wins the distinguishing game if $b = b'$. Formally, we define the distinguishing advantage of $\mathscr{A}$ as

$$\mathbf{Adv}_{O_0;O_1}(\mathscr{A}) = \left|\Pr\left(\mathscr{A}^{O_1} = 1\right) - \Pr\left(\mathscr{A}^{O_0} = 1\right)\right|.$$

Additionally, if a specific distinguishing game G is used,the advantage of $\mathscr{A}$ is denoted by $\mathbf{Adv}^{G}_{O_0;O_1}(\mathscr{A})$.

We assume that the adversary is non-trivial, i.e. it never makes a duplicate query, and it never makes a query for which the response is already known due to some previous query. We note that the adversary might be computationally unbounded. Usually, we measure the adversary's capabilities by a tuple of parameters (q, ℓ, σ, t) where

- q - is an upper bound on the number of queries made by the adversary;
- ℓ - is an upper bound of the length of any query;
- σ - is an upper bound on the total length of all queries;
- t - is an upper bound on the computation time of the adversary.

We denote by $\mathbb{A}(\mathbf{r})$ the set of all adversaries with a tuple of parameters $\mathbf{r}$. These adversaries are also referred to as $\mathbf{r}$-adversaries. Finally, for any security game G and any $\varepsilon > 0$, we say a function F with access to a set of oracles O and set of parameters $\mathbf{r}$ is $(\varepsilon, \mathbf{r})$-$G$ secure if

$$\mathbf{Adv}^{G}_{\mathsf{F}}(\mathbf{r}) := \max_{\mathscr{A} \in \mathbb{A}(\mathbf{r})} \mathbf{Adv}^{G}_{O}(\mathcal{A}) \leq \varepsilon.$$

In general, we consider not only distinguishers but also more general adversaries. For instance, adversaries may aim to recover the key or create a collision on some function. It is worth noting that distinguishers are the strongest type of adversaries, as their goals are the easiest to achieve, in contrast to more challenging objectives such as full key recovery.

2.2.1.1 Adversarial Capabilities

The distinguishing game uniquely defines the capabilities of a given adversary. Following the classification in [235], we categorize different types of adversaries based on their capabilities:

- In a *ciphertext-only attack*, the adversary attempts to deduce the decryption key or plaintext solely by observing ciphertext. Any scheme vulnerable to this type of attack is considered completely insecure.

- In a *known-plaintext attack*, the adversary has access to some plaintexts and their corresponding ciphertexts. This type of attack is typically slightly more difficult than the ciphertext-only attack.
- In a *chosen-plaintext attack* (CPA), the adversary can encrypt plaintexts of their choice by accessing an encryption oracle.
- In a *chosen-ciphertext attack* (CCA), the adversary has access to both decryption and encryption oracles, enabling him to decrypt (and encrypt) a ciphertext of his choice.

If the adversary can adapt its queries based on earlier queries, he is called *adaptive*; otherwise, he is called *non-adaptive*.

2.2.1.2 Multi-User Security Games

The security games defined in this thesis are typically used to distinguish between two worlds. In the real-world, the distinguisher has access to an actual construction, usually instantiated by a single key. In contrast, the ideal-world grants the distinguisher access to an ideal primitive. However, in practical applications, primitives such as block ciphers are generally deployed on a large scale. Attackers often aim to compromise any user among many, highlighting the importance of considering a multi-user (sometimes called multi-key security) security setting.

The notion of multi-user (mu) security was first introduced for pseudorandom functions (PRFs) as a technical tool by Bellare, Canetti, and Krawczyk [27]. It was later defined as a full-fledged security goal for public-key encryption by Bellare, Boldyreva, and Micali [25]. In the mu setting, an attacker can distribute its resources to attack multiple users, each with independent keys, and is deemed successful if it compromises at least one user.

In this thesis, we define each notion separately for the single-user setting and, where necessary, for the multi-user setting. Additionally, we add to the set of parameters for an adversary the number of user.

2.2.2 Indifferentiability

Following the formulation by Gunsing et al. [154], we distinguish between three notions of indifferentiability.

- **Regular Indifferentiability:** In (regular) indifferentiability as formalized by Coron et al. [97], the adversary has full freedom in the order in which it makes the queries and he can do so adaptively;
- **Public Indifferentiability:** A weaker variant of indifferentiability introduced by Yoneyama et al [318] and Dodis et al [124], where the construction oracle is public and known to the simulator;
- **Sequential Indifferentiability:** Another variant that differs from the regular indifferentiability, introduced by Mandal et al [224], requiring that all primitive queries

be made before any construction queries. Interestingly, sequential indifferentiability is equivalent to public indifferentiability for stateless ideal primitives [224], like the sum of permutations.

In this work, we introduce the notion of sequential indifferentiability. Formally, let $\mathscr{A}$ be an adversary that has access to a pair of oracles, denoted generically as $(C, \mathcal{P})$. We call $\mathscr{A}$ *sequential* if, after making its first query to the left oracle C, it refrains from querying the right oracle $\mathcal{P}$. Thus, such an adversary operates in two distinct phases: first, it queries only $\mathcal{P}$, and subsequently, only C. We define the total oracle query cost of $\mathscr{A}$ as the combined number of queries made to the right oracle (either by $\mathscr{A}$ or C) during $\mathscr{A}$'s interaction with the pair $(C^{\mathcal{P}}, \mathcal{P})$. Specifically, if C issues c queries to $\mathcal{P}$ for each query it receives, and if $\mathscr{A}$ makes q_c queries to the left oracle and q_p queries to the right oracle, then the total oracle query cost is at most $q_p + cq_c$.

We consider a simplified version for the indifferentiability of a block cipher $\mathtt{E} : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ that is built upon a tuple of r independent permutation $\mathbf{\Pi} = (\Pi_1, \ldots, \Pi_r)$, denoted by $\mathtt{E}^{\mathbf{\Pi}}$. In this setting, the adversary $\mathscr{A}$ aims to distinguish between two pairs of oracles using adaptive bidirectional queries:

- the *real world* oracle, $(\mathtt{E}_k^{\mathbf{\Pi}}, \mathbf{\Pi})$, where $k \twoheadleftarrow_{\$} \mathcal{K}$ and $\mathbf{\Pi} \twoheadleftarrow_{\$} \mathsf{Perm}(n)^r$ are sometimes referred as the left and right oracle, respectively; and
- the *ideal world* oracle, $(\widetilde{\Pi}, \mathcal{S})$, where $\widetilde{\Pi} \twoheadleftarrow_{\$} \mathsf{BC}(\mathcal{K}; \{0,1\}^n)$ (the left oracle), and $\mathcal{S}$ (the right oracle) is an oracle Turing machine, referred as the simulator, with bidirectional oracle access to $\widetilde{\Pi}$.

Finally, we define the notion of $(q, \sigma, t, \varepsilon)$-sequential indifferentiability from an ideal cipher.

Definition 2.2.1. *Let $q, \sigma, t \in \mathbb{N}$ and $0 < \varepsilon \in \mathbb{R}$ be some security parameters. A block cipher $E^{\mathbf{\Pi}}$ is said to be $(q, \sigma, t, \varepsilon)$-sequential indifferentiable from an ideal cipher if there exists an oracle simulator $\mathcal{S}$ such that*

$$\mathbf{Adv}_E^{\text{seq-indiff}}(q, \sigma, t) := \max_{\mathscr{A} \in \mathbb{A}(q)} \left| \Pr\left(\left[\mathscr{A}^{\widetilde{\Pi}, \mathcal{S}^{\widetilde{\Pi}}} = 1\right]\right) - \Pr\left(\left[\mathscr{A}^{E^{\mathbf{\Pi}}, \mathbf{\Pi}} = 1\right]\right) \right| \leq \varepsilon, \tag{2.1}$$

where $\mathcal{S}$ makes at most σ oracle queries, and runs in time at most t.

2.2.2.1 Chosen Key Setting and Sequential Indifferentiability

In many scenarios, (tweakable) block ciphers are analyzed in stronger security models, such as the known-key (KKA) and chosen-key (CKA) [199, 55] attack models, where the adversary either knows the random key or, in an even more stronger setting, can instantiate the block cipher with its choice of key at each invocation. Knudsen and Rijmen [199] suggested *correlation intractability* [69] notion due to Canetti et al. as a possible theoretical formalization to capture the KKA and CKA models. While it is well-known [69] that a rigorous definition of chosen-key security is impossible in the standard model, the idealized models help us avoid classical impossibility results. This is done using the notion of evasive relations. We adapt the definitions from [92].

Definition 2.2.2. *An m-ary relation $\mathcal{R}$ is said to be (q, ε)-evasive with respect to an ideal cipher $\widetilde{P}$ if for any oracle Turing machine $\mathcal{M}$ making at most q oracle queries, one has*

$$\Pr\left(\widetilde{P} \twoheadleftarrow_{\$} \widetilde{\mathsf{Perm}}(\kappa; n), (\alpha_i)_{i\in[1;m]} \leftarrow \mathcal{M}^{\widetilde{P}} : (\alpha_i, \widetilde{P}(\alpha_i))_{i\in[1;m]} \in \mathcal{R}\right) \leq \varepsilon.$$

Informally, a relation is evasive if it is hard, for any algorithm with oracle access to an ideal cipher, to output an m-tuple of inputs $(\alpha_i)_{i\in[1;m]}$ such that $(\alpha_i, \widetilde{P}(\alpha_i))_{i\in[1;m]}$ satisfies the relation. A similar notion can be defined through correlation intractability, when we consider a block cipher $\mathtt{E}^{\mathbf{P}}$ constructed over a tuple of random permutations $\mathbf{P}$.

Definition 2.2.3. *Let $E^{\mathbf{P}}$ be a block cipher construction over a tuple of independent and uniform random permutations $\mathbf{P}$, and let $\mathcal{R}$ be an m-ary relation. $E^{\mathbf{P}}$ is said to be (q, ε)-correlation intractable with respect to $\mathcal{R}$ if, for any Turing machine $\mathcal{M}$ with oracle access to $\mathbf{P}$ making at most q oracle queries, one has*

$$\Pr\left(\mathbf{P} \twoheadleftarrow_{\$} \mathsf{Perm}(n)^r, (\alpha_i)_{i\in[1;m]} \leftarrow \mathcal{M}^{\mathbf{P}} : (\alpha_i, E^{\mathbf{P}}(\alpha_i))_{i\in[1;m]} \in \mathcal{R}\right) \leq \varepsilon.$$

We will deem a block cipher construction $\mathtt{E}^{\mathbf{P}}$ resistant to chosen-key attacks if, for every relation $\mathcal{R}$ that is (q, ε)-evasive with respect to an ideal cipher, $\mathtt{E}^{\mathbf{P}}$ is (q', ε')-correlation intractable with respect to $\mathcal{R}$, with $q' \approx q$ and $\varepsilon' \approx \varepsilon$. The link between correlation intractability and sequential indifferentiability comes from the following result based on [224, Theorem 3].

Theorem 2.2.1. *[91, Theorem 4] Let $E^{\mathbf{P}}$ be a block cipher constructed over a tuple of independent and uniform random permutations $\mathbf{P}$ such that $E^{\mathbf{P}}$ makes at most c queries to $\mathbf{P}$ on any input. Assume that $E^{\mathbf{P}}$ is $(q + cm, \sigma, t, \varepsilon_{SI})$-sequential indifferentiable from an ideal cipher. Then, for any m-ary relation $\mathcal{R}$, if $\mathcal{R}$ is $(\sigma + m, \varepsilon_{\mathcal{R}})$-evasive with respect to an ideal cipher, then $E^{\mathbf{P}}$ is $(q, \varepsilon_{SI} + \varepsilon_{\mathcal{R}})$-correlation intractable with respect to $\mathcal{R}$.*

Theorem 2.2.1 clearly implies that proving the sequential indifferentiability of $\mathtt{E}^{\mathbf{P}}$ is sufficient to justify some form of resistance to chosen-key attacks.

2.3 Cryptographic Primitives

In this section, we introduce the cryptographic primitives that are foundational to this thesis. We begin with a formal definition of the primitives, followed by design approaches when relevant and the associated security notions.

2.3.1 Pseudorandom Function

For any keyed function $\mathtt{F} : \mathcal{K} \times \mathcal{M} \to \mathcal{M}'$ and any adversary $\mathscr{A}$ with oracle access to either $\mathtt{F}_k$ where $k \twoheadleftarrow_{\$} \mathcal{K}$ or a uniformly random function $\Gamma \twoheadleftarrow_{\$} \mathsf{Func}(\mathcal{M}; \mathcal{M}')$. The PRF advantage of $\mathscr{A}$ against $\mathtt{F}$ is defined as

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathtt{F}}(\mathscr{A}) := \mathbf{Adv}_{\mathtt{F}_k;\Gamma}(\mathcal{A}).$$

The multi-user security PRF (denoted by mu-PRF) advantage is naturally extended the following way:

$$\mathbf{Adv}_{\mathsf{F}}^{\text{mu-prf}}(\mathscr{A}) := \mathbf{Adv}_{\mathsf{F}_{\mathbf{k}},\boldsymbol{\Gamma}}(\mathcal{A}),$$

where $\mathsf{F}_{\mathbf{k}} = (\mathsf{F}_{k_1}, \ldots, \mathsf{F}_{k_u})$, $\boldsymbol{\Gamma} = (\Gamma_1, \ldots, \Gamma_u)$ such that for any $i \in [1;u]$, $k_i \twoheadleftarrow_{\$} \mathcal{K}$ and $\Gamma_i \twoheadleftarrow_{\$} \mathsf{Func}(\mathcal{M};\mathcal{M}')$ are all sampled uniformly and independently at random.

2.3.1.1 Weak Pseudorandom Function

When designing cryptographic schemes, it's prudent to consider weaker properties, as they are more likely to be satisfied by a given function and can lead to more efficient implementations compared to stronger requirements.

One popular example, proposed by Naor and Reingold [252] (see also [253, 5, 108, 251, 269]), is the notion of a *weak* PRF. In this setting, the distinguisher is not allowed to make arbitrary queries. Instead, it receives a uniform and independent random input along with the evaluation of the underlying function on this input whenever it queries the oracle. This notion can also be viewed as a game where the adversary is constrained to sample its inputs uniformly at random.

Formal Definition. We define the wPRF (resp. mu-wPRF) advantage of a adversary $\mathscr{A}$ against a keyed function F (defined above) as

$$\mathbf{Adv}_{\mathsf{F}}^{\text{wprf}}(\mathscr{A}) := \mathbf{Adv}_{\mathsf{F}}^{\text{prf}}\left(\mathscr{A}^{\$}\right), \qquad \mathbf{Adv}_{\mathsf{F}}^{\text{mu-wprf}}(\mathscr{A}) := \mathbf{Adv}_{\mathsf{F}}^{\text{mu-prf}}\left(\mathscr{A}^{\$}\right)$$

where $\mathscr{A}^{\$}$ denotes the same adversary, except that it samples its queries uniformly at random at each turn.

$\mathrm{Re}_{\mathsf{F},\mathsf{chop}}^{\text{prfleak}}(\mathscr{D})$ | $\boxed{\mathrm{Id}_{\mathsf{F},\mathsf{chop}}^{\text{prfleak}}(\mathscr{D})}$

1: $u \leftarrow 0$
2: $b \leftarrow \mathscr{D}^{\textsc{New},\mathsf{O}}$
3: **return** $b =_? 1$

Oracle $\mathsf{O}(m, i)$

1: **if** $i \notin [u]$ **then return** $\perp$
2: **end if**
3: $c \leftarrow \mathsf{F}(K_i, x)$
4: $\boxed{c \leftarrow \Gamma_i(x)}$
5: **return** c

Oracle $\textsc{New}()$

1: $u \leftarrow u + 1$
2: $K_u \twoheadleftarrow_{\$} \mathcal{K}$
3: $\boxed{\Gamma_u \twoheadleftarrow_{\$} \mathsf{Func}(\mathcal{K};\mathcal{M})}$
4: **return** $\mathsf{chop}(K_u)$

Figure 2.1: *PRFLEAK game: the real world is denoted by* $\mathrm{Re}_{F,\mathsf{chop}}^{\text{prfleak}}(\mathscr{D})$ *and the ideal world by* $\mathrm{Id}_{F,\mathsf{chop}}^{\text{prfleak}}(\mathscr{D})$.

2.3.1.2 PRF Under Leakage

In certain scenarios, a construction (keyed function) is designed so that the adversary can obtain some information about the secret key. This is particularly relevant in the context of the augmented MAC (AMAC) construction introduced in Bellare et al. [23]. Specifically, they highlight that an adversary might be able to gather partial information about the secret key through output transforms like truncation, which makes understanding and mitigating such leakage critical to ensuring security.

Formal Definition. We follow the definition of Pseudorandom function under leakage (PRFLEAK) from [23]. Formally, we describe the multi-user security game in Figure 2.1. Note that the game is similar to the mu-PRF notion only that here the adversary is given a part of each key using the leakage function chop. The PRFLEAK advantage of any adversary $\mathcal{D}$ against $\mathtt{F}$ and leakage function chop is defined as

$$\mathbf{Adv}^{\text{prfleak}}_{\mathtt{F},\mathsf{chop}}(\mathscr{D}) := \left|\Pr\left(\mathrm{Re}^{\text{prfleak}}_{\mathtt{F},\mathsf{chop}}(\mathscr{D}) = 1\right) - \Pr\left(\mathrm{Id}^{\text{prfleak}}_{\mathtt{F},\mathsf{chop}}(\mathscr{D}) = 1\right)\right|.$$

2.3.2 Block Cipher

A *block cipher* with key space $\mathcal{K}$ and message space $\mathcal{M}$ is a mapping $\mathtt{E} : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ such that for any key $k \in \mathcal{K}$, the mapping $m \mapsto \mathtt{E}(k, m)$ is a permutation of $\mathcal{M}$. The set of all block ciphers with key space $\mathcal{K}$ and messages space $\mathcal{M}$ will be denoted by $\mathsf{BC}(\mathcal{K}; \mathcal{M})$. Moreover, if $|\mathcal{M}| = n$ and $|\mathcal{K}| = \kappa$, by abuse of notations we denote $\mathsf{BC}(\kappa; n) := \mathsf{BC}(\mathcal{K}; \mathcal{M})$.

2.3.2.1 Design Approaches

The basic principles in the design of block ciphers date back to the seminal work of Shannon [291], where he proposes that ciphers should be constructed through an iterative process, where at each round we use simple operations until the cipher achieves certain properties. More precisely, for each round i, a round function $F_i : \mathcal{K}_i \times \mathcal{M} \to \mathcal{M}$ is used to derive the next state. In this process, F_i takes a round key $k_i \in \mathcal{K}_i$ derived from a master key $k \in \mathcal{K}$, through a process called a *key schedule*, and a state $x_{i-1} \in \mathcal{M}$ and outputs a new state $x_i \in \mathcal{M}$. The ciphertext is defined as the output of the last round. Formally, for a key $k \in \mathcal{K}$, the block cipher $\mathtt{E}_k$ is defined as,

$$\mathtt{E}_k(M) := F_{k_r} \circ F_{k_{r-1}} \circ \ldots \circ F_{k_0}(M).$$

These ciphers are called *product ciphers*.

In the same seminal work, Shannon [291] suggested that the round functions satisfy the properties of *confusion* and *diffusion*. Numerous descriptions of these two concepts have been given throughout the years. In this thesis, we use the notions given by Massey [200]:

- **Confusion**: The ciphertext statistics should depend on the plaintext statistics in a manner too complicated to be exploited by the cryptanalyst.
- **Diffusion**: Each digit of the plaintext and each digit of the secret key should influence many digits of the ciphertext.

Although the properties of confusion and diffusion are not quantifiable, they still serve as intuition to properties the designers should strive for in the design of a block cipher. Interestingly, most block ciphers alternate between three types of operations:

- **Substitution operation.** This involves the application of a non-linear function to achieve the goal of confusion. Typically, this is implemented through a lookup table known as an S-box, where the input is directly substituted with a predefined output.
- **Mixing operation.** This operation is designed to achieve diffusion by applying a linear function. In practical terms, this often involves matrix multiplication, which helps to spread the influence of the plaintext and key bits throughout the ciphertext.
- **Sub-key addition.** This simple operation involves the XOR of a round key with the block cipher's state at each round. It plays a crucial role in integrating the key into the cipher, enhancing the overall diffusion process.

Drawing on the principles described above, three design strategies have garnered significant interest: Feistel networks, substitution-permutation networks (SPNs) and key alternating ciphers (KACs). There are many other strategies such as Lai–Massey ciphers [208] and various generalizations of Feistel networks (e.g. [247, 169, 229]), but in this thesis we keep our focus on these three design strategies.

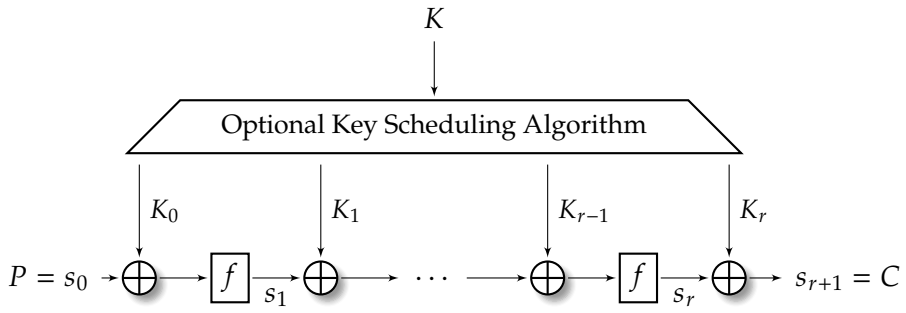

Figure 2.2: *r-round Key Alternating Cipher (KAC) [185].*

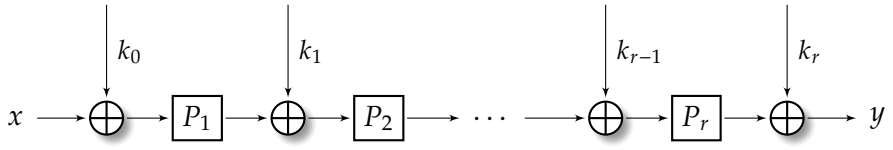

Figure 2.3: *r-round* `IEM`

Key Alternating Cipher (KAC). As mentioned above, mixing layers are often linear to facilitate efficient implementation. A common method for implementing a linear layer is to XOR a sub-key at each round. Consequently, substitution layers are necessary to disrupt the linear relationships in the outputs. Together, these operations create a sufficiently secure round function. A *key-alternating cipher* [103] consists of an alternating sequence of unkeyed round functions (usually permutations) and simple bitwise key additions (see also Figure 2.2).

The simplest construction utilizing this design paradigm which is provably secure in some formal way, dates back to Even and Mansour [132, 133] in 1991. In their paper, they design a block cipher from a public permutation $P : \{0,1\}^n \to \{0,1\}^n$ using two keys k_0, k_1 defined as: for any message $m \in \{0,1\}^n$, $\mathtt{EM}^P_{k_0,k_1}(m) := k_1 \oplus P(k_0 \oplus m)$. This straightforward yet powerful scheme achieves birthday-bound security (up to $2^{n/2}$ queries) in the random-permutation model. Remarkably, it stands as one of the most "minimal" constructions to achieve this level of security, with the elimination of any component leading to a complete breakdown of its security guarantees.

In 2012, Bogdanov et al. [60] generalized this construction to r-rounds called the Iterated Even-Mansour (IEM). Namely, for a tuple of r public permutations, $\mathbf{P} = (P_1, \ldots, P_r) \in \mathsf{Perm}(n)^r$, and a tuple of $r+1$ keys, $\mathbf{k} = (k_0, \ldots, k_r) \in (\{0,1\}^n)^{r+1}$, the r-round Iterated Even-Mansour construction is defines as: for any message $m \in \{0,1\}^n$ (see also Figure 2.3),

$$\mathtt{IEM}^{\mathbf{P}}_{\mathbf{k}}(m) := k_{r+1} \oplus P_r(k_r \oplus P_{r-1}(\ldots P_2(k_2 \oplus P_1(k_1 \oplus m))\ldots)).$$

As far as security guarantees are concerned, Bogdanov et al. [60] showed that, for $r \geq 2$, security is guaranteed up to $2^{2n/3}$ queries and gave an attack requiring $2^{rn/r+1}$ queries. Later on, Steinberger [300] improved that result to up to $2^{3n/4}$ queries for $r \geq 3$ and Lampe et al. [209] proved the security up to $2^{rn/r+2}$ queries for even number of rounds r. Finally, Chen and Steinberger [81] proved a tight security proof of up to $2^{rn/r+1}$ queries.

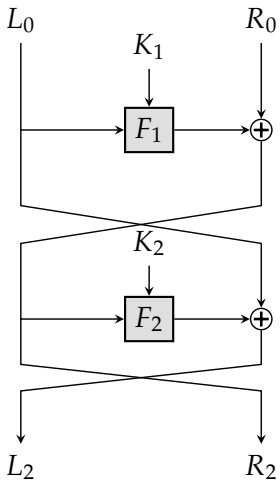

Figure 2.4: *2-round Feistel network*

Generalized Feistel Network. The construction dates back to the seminal work of Luby and Rackoff [219], and at the design of the popular block ciphers: `DES` [112], `Blowfish` [288], `KASUMI` [194], and `Camellia` [9]. At its core, a Feistel network is an r round construction that divides the input into two halves $(L_0, R_0) \in \mathcal{M}$. During each round i, the round function $F_i : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ takes a state (L_{i-1}, R_{r-1}) together with a key $K_i \in \mathcal{K}$ and outputs a new state (L_i, R_i) where

$$L_i := R_{i-1} \oplus F_i(K_i, L_{i-1}), R_i := L_{i-1},$$

and finally the ciphertext is defined as (L_r, R_r) (see also Figure 2.4).

The provable security treatment of Fesitel Networks date back to the original paper by Luby-Rackoff [221]. In that paper, they show that the 3-round Feistel scheme is a PRP when its round functions are modeled as PRFs. This was followed by a paper by Patarin [259] who showed that four rounds yield a SPRP. Consequently, a plethora of subsequent work followed [230, 231, 309, 260, 170, 264].

This balanced two branch construction is also called Type-1 Feistel Network. In the generalized versions, the input is divided into four parts. Zheng et al. [326] categorize the constructions according to how many round functions are used in a single round, called: Type-1, Type-2 and Type-3 Networks. Modern examples of Generalized Feistel Network based block ciphers include: `CAST` -256 [2], `CLEFIA` -128 [197] and `Skipjack` [54].

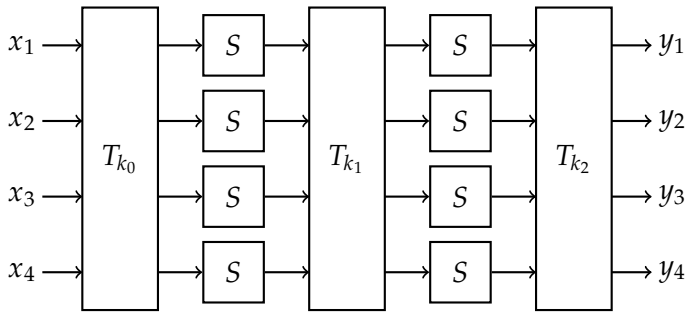

Figure 2.5: *Two round SPN with $b = 4$, where T_k is a linear layer for some key k and S is a non-linear permutation.*

Substitution Permutation Network (SPN). In these constructions, each round function consists of three layers: a round-key addition, a substitution layer and a mixing layer. Variations among different substitution-permutation networks (SPNs) often manifest in the design of the initial and final rounds. For instance, `AES` (Rijndael) [3] omits the last mixing layer to allow for faster decryption. On the other hand, `SERPENT` [53] (an AES contest finalist) enhances its security framework by implementing more complex operations in the initial rounds, setting a robust foundation for the encryption process. Since the substitution and mixing layers are public and invertible operation most SPNs today include an additional sub-key addition at the end of each round. Modern

examples of SPN-based block ciphers include: `AES` [3], `PRESENT` [200], `Skinny` [21], Kuznyechik [125] and `SERPENT` [53].

Formally, let a, b be some positive integers such that $a \cdot b = n$. An SPN is built upon two functions, a non-linear permutation $S : \{0,1\}^a \to \{0,1\}^a$ and a linear function $T_k : \mathcal{M} \to \mathcal{M}$ for any key $k \in \mathcal{K}$. We define the round function F_{k_i}, with key $k_i \in \mathcal{K}_i$ and input $x = (x_1, \ldots, x_b) \in \mathcal{M}$ as $F_{k_i}(x) = T_{k_i}(z_1, \ldots, z_b)$ where for every $i \in [1; b]$, $z_i = S(x_i)$ (see also Figure 2.5).

2.3.2.2 Security Notions

Once the key is fixed, block ciphers are permutations, therefore their security is captured by the ability of a adversary to distinguish between their output and that of a uniformly random permutation. The *Pseudorandom permutation* (PRP) notion captures the indistinguishability when the adversary is given access only to the encryption. Additionally, if the adversary is given access to decryption queries as well, we call it the *Strong Pseudorandom permutation* or SPRP.

Definition 2.3.1 (PRP/SPRP advantage)**.** *Let $\mathtt{E} \in \mathsf{BC}(\mathcal{K}; \mathcal{M})$ be a block cipher and $\mathscr{A}$ be an adversary with oracle access to either one or two functions. Then, for a random key $k \twoheadleftarrow_{\$} \mathcal{K}$ and uniform random permutation $\mathsf{P} \in \mathsf{Perm}(\mathcal{M})$, the PRP/SPRP advantage of $\mathscr{A}$ on $\mathtt{E}$ is defined as*

$$\mathbf{Adv}^{\mathrm{prp}}_{E}(\mathscr{A}) := \mathbf{Adv}_{E_k;\mathsf{P}}(\mathcal{A}), \qquad \mathbf{Adv}^{\mathrm{sprp}}_{E}(\mathscr{A}) := \mathbf{Adv}_{E_k^{\pm};\mathsf{P}^{\pm}}(\mathcal{A}).$$

Finally, the multi-user PRP/SPRP advantage is defined similarly.

Definition 2.3.2 (mu-PRP/mu-SPRP advantage)**.** *For $u \in \mathbb{N}$ users, a block cipher $\mathtt{E} \in \mathsf{BC}(\mathcal{K}; \mathcal{M})$ and an adversary $\mathscr{A}$ with oracle access a set of functions. Then, for uniformly independent keys $\mathbf{k} = (k_1, \ldots, k_u) \twoheadleftarrow_{\$} \mathcal{K}^u$ and uniformly independent random permutations $\mathbf{P} = (\mathsf{P}_1, \ldots, \mathsf{P}_u) \in \mathsf{Perm}(\mathcal{M})^u$, the mu-PRP/mu-SPRP advantage of $\mathscr{A}$ on $\mathtt{E}$ is defined as*

$$\mathbf{Adv}^{\mathrm{mu-prp}}_{E}(\mathscr{A}) := \mathbf{Adv}_{E_{\mathbf{k}};\mathbf{P}}(\mathcal{A}), \qquad \mathbf{Adv}^{\mathrm{mu-sprp}}_{E}(\mathscr{A}) := \mathbf{Adv}_{E_{\mathbf{k}}^{\pm};\mathbf{P}^{\pm}}(\mathcal{A}),$$

where in the mu-SPRP notion the adversary has bidirectional access to a tuple of permutations.

2.3.3 Tweakable Block Cipher

A *tweakable block cipher* with key space $\mathcal{K}$, tweak space $\mathcal{T}$ and message space $\mathcal{M}$ is a mapping $\widetilde{\mathtt{E}} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ such that for any $k \in \mathcal{K}$ and any tweak $t \in \mathcal{T}$, $m \mapsto \widetilde{\mathtt{E}}(k, t, m)$ is a permutation of $\mathcal{M}$. Additionally, for ease of notations, for any key $k \in \mathcal{K}$ and tweak $t \in \mathcal{T}$, we write $\mathtt{E}_{k,t} : \mathcal{M} \to \mathcal{M}$ as the function $\mathtt{E}_{k,t} = \mathtt{E}(k, t, \cdot)$. The set of all tweakable block ciphers with key space $\mathcal{K}$, tweak space $\mathcal{T}$ and message space $\mathcal{M}$ will be denoted by $\widetilde{\mathsf{BC}}(\mathcal{T}; \mathcal{K}; \mathcal{M})$. Moreover, if $|\mathcal{T}| = 2^{\tau}$, $|\mathcal{K}| = 2^{\kappa}$ and $|\mathcal{M}| = 2^{n}$, by abuse of notations we denote by $\widetilde{\mathsf{BC}}(\tau; \kappa; n) := \widetilde{\mathsf{BC}}(\mathcal{T}; \mathcal{K}; \mathcal{M})$.

2.3.3.1 Design Approaches

Tweakable block ciphers are usually built from block ciphers in a black box manner. This design approach was introduced by Liskov et al. [216], where they propose two constructions, LRW1 and LRW2 . The first proposed construction, LRW1 transforms a block cipher into a tweakable block cipher by masking the encryption output of the input message with the given tweak which is again re-encrypted to produce the ciphertext. Namely, for a tweak $t \in \mathcal{T}$ and message $m \in \mathcal{M}$,

$$\texttt{LRW1}_k^{\texttt{E}}(t, m) = \texttt{E}_k(\texttt{E}_k(m) \oplus t).$$

where E is some block cipher. It has been shown that LRW1 achieves CPA security up to $2^{n/2}$ queries. In order to achieve CCA security, Liskov et al. [216] propose the LRW2 construction which requires an additional AXU hash function H and is defined as,

$$\texttt{LRW2}_{k,k'}^{\texttt{E},H}(t, m) := \texttt{E}_k(m \oplus H_{k'}(t)) \oplus H_{k'}(t).$$

In the same work, Liskov et al. [216] show that LRW2 achieves tight CCA security of up to $2^{n/2}$ queries and uses only one block cipher call but requires an almost xor universal hash function. Following this work, several other block cipher-based TBC constructions achieving birthday bound security emerged, such as Rogaway's XEX [278] and its future refinements [76, 243, 147].

Beyond Birthday Bound Secure TBCs. The first block cipher-based TBC achieving beyond birthday bound security was proposed by Minematsu [244]. However, it has limitations in its design: the tweak length cannot exceed half of the block length, and it requires a key change for each new tweak value. In 2012, Landecker et al. [214], proposed a cascaded version of LRW2 , consisting of two rounds, defined as follows:

$$\texttt{CLRW2}_{k,k'}^{\texttt{E},H_1,H_2}(t, m) := \texttt{LRW2}_{k'}^{\texttt{E},H_1}(t, \texttt{LRW2}_k^{\texttt{E},H_2}(t, m)),$$

where H_1, H_2 are AXU hash functions and E a block cipher. In their paper, they proved CLRW2 achieves security up to $2^{2n/3}$ queries. Improving the security of this construction (or cascaded versions of it) has led to a plethora of subsequent works [271, 213, 238, 191]. The constructions mentioned above are proven in the standard model. However, there are also TBC constructions proven in the ideal cipher model, such as $\widetilde{\mathsf{F}}[1]$ and $\widetilde{\mathsf{F}}[2]$ by Mennink [236, 237] and XHX by Jha et al. [189].

Note that all the constructions mentioned above are either inefficient or only secure up to the birthday bound. Therefore, a natural conclusion is to design TBCs from scratch or using low-level primitives. This design strategy will be discussed in more detail in Chapter 4.

2.3.3.2 Security Notions

The security notions are adapted from the ones of block ciphers.

Definition 2.3.3 (TPRP/STPRP advantage)**.** *Let $\widetilde{E} \in \widetilde{BC}(\mathcal{T};\mathcal{K};\mathcal{M})$ be a tweakable block cipher and $\mathscr{A}$ be an adversary with oracle access to either one or two functions. Then, for a random key $k \twoheadleftarrow_{\$} \mathcal{K}$ and uniform random tweakable permutation $\widetilde{\mathsf{P}} \twoheadleftarrow_{\$} \widetilde{\mathsf{Perm}}(\mathcal{T};\mathcal{M})$, the TPRP/STPRP advantage of $\mathscr{A}$ on $\widetilde{E}$ is defined as*

$$\mathbf{Adv}^{\mathrm{tprp}}_{\widetilde{E}}(\mathscr{A}) := \mathbf{Adv}_{\widetilde{E}_k;\widetilde{\Pi}}(\mathcal{A}), \qquad \mathbf{Adv}^{\mathrm{sprp}}_{\widetilde{E}}(\mathscr{A}) := \mathbf{Adv}_{\widetilde{E}^{\pm}_k;\widetilde{\mathsf{P}}^{\pm}}(\mathcal{A}).$$

Definition 2.3.4 (mu-TPRP/mu-STPRP advantage)**.** *For some number of users Let $u \in \mathbb{N}$ be a positive integers corrsponding to the number of user, $\widetilde{E} \in \widetilde{BC}(\mathcal{T};\mathcal{K};\mathcal{M})$ be a tweakable block cipher and $\mathscr{A}$ be an adversary with oracle access to a set of functions. Then, for uniformly independent keys $\mathbf{k} = (k_1,\ldots,k_u) \twoheadleftarrow_{\$} \mathcal{K}^u$ and uniformly independent random tweakable permutations $\widetilde{\mathbf{P}} = \left(\widetilde{\mathsf{P}}_1,\ldots,\widetilde{\mathsf{P}}_u\right) \twoheadleftarrow_{\$} \widetilde{\mathsf{Perm}}(\mathcal{T};\mathcal{M})^u$, the mu-PRP/mu-SPRP advantage of $\mathscr{A}$ on $\mathbf{E}$ is defined as*

$$\mathbf{Adv}^{\mathrm{mu-tprp}}_{\widetilde{E}}(\mathscr{A}) := \mathbf{Adv}_{\widetilde{E}_{\mathbf{k}};\widetilde{\mathbf{P}}}(\mathcal{A}), \qquad \mathbf{Adv}^{\mathrm{mu-tsprp}}_{\widetilde{E}}(\mathscr{A}) := \mathbf{Adv}_{\widetilde{E}^{\pm}_{\mathbf{k}};\widetilde{\mathbf{P}}^{\pm}}(\mathcal{A}),$$

where in the mu-TSPRP notion the adversary has bidirectional access to a tuple of permutations.

IND-CPA and IND-CCA Security Under the Random Permutation Model. Let $\widetilde{\mathrm{E}} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ be a tweakable block cipher that is constructed over a tuple of independent and uniform random permutations $\mathbf{P} = (P_1,\ldots,P_r)$, denoted by $\widetilde{\mathrm{E}}^{\mathbf{P}}$, where $P_i \twoheadleftarrow_{\$} \mathsf{Perm}(n)$ for every $i \in [1;r]$. We consider an adversary $\mathscr{A}$ whose goal is to distinguish between two pairs of oracles:

- the *real world* oracle, where $\mathscr{A}$ can make adaptive bidirectional queries to $(\widetilde{\mathrm{E}}^{\mathbf{P}}_k, \mathbf{P})$;
- the *ideal world* oracle $(\widetilde{P}, \mathbf{P})$, where $\widetilde{P} \twoheadleftarrow_{\$} \widetilde{\mathsf{Perm}}(\mathcal{T};\mathcal{M})$ independent from $\mathbf{P}$.

In this thesis, we consider two types of adversarial settings for tweakable block ciphers. In the first setting, the adversary is adaptive but can only make forward queries to either $(\widetilde{\mathrm{E}}^{\mathbf{P}}_k, \mathbf{P})$ or $(\widetilde{P}, \mathbf{P})$, referred to as *indistinguishability under chosen-plaintext attack* (IND-CPA). In the second setting, the adversary is allowed to make adaptive bidirectional queries in both worlds, referred to as *indistinguishability under chosen-ciphertext attack* (IND-CCA).

2.3.4 Hash Function and Universality

We follow [235, 279] for definitions and security notions. For a key space $\mathcal{K}$ and two non-empty sets $\mathcal{X}$ (message space) and $\mathcal{Y}$ (hash space), a *keyed hash function* $\mathcal{H}$, or simply an *hash function*, takes two inputs: a key $k \in \mathcal{K}$ and a message $x \in \mathcal{X}$, and outputs a short digest $\mathcal{H}(k,x)$ (indicating that $\mathcal{H}(k,\cdot)$ should be a compressing function). The key k acts as a selector for the hash function, defining a specific function H_k that maps messages to digests. For simplicity, $H \twoheadleftarrow_{\$} \mathcal{H}$ denotes sampling a key $k \twoheadleftarrow_{\$} \mathcal{K}$ and setting $H = H_k$.

2.3.4.1 Security Notions

Since hash functions are designed to be compressing, it is typical that $|\mathcal{X}| > |\mathcal{Y}|$, making collisions unavoidable. In many-to-one mappings, collisions are guaranteed,

meaning that the unique association between inputs and hash values can only be ensured in a computational sense. In practice, a hash value should ideally be uniquely identifiable with a single input, and finding collisions should be computationally infeasible (essentially never occurring in practice). This requirement gives rise to the security notion of *collision resistance*.

Definition 2.3.5 (Collision Resistance (CR))**.** *We say that $\mathcal{H}$ is (ε_{CR}, t)-collision-resistant (CR) if for any t-bounded adversary $\mathscr{A}$, one has*

$$\Pr_{H \leftarrow_{\$} \mathcal{H}} \left(\mathscr{A}^H = (X_0, X_1) : X_0 \neq X_1 \wedge H(X_0) = H(X_1)\right) \leq \varepsilon_{CR}.$$

The probability above defines a game in which adversary $\mathscr{A}$ wins if he can find a collision in H_k for some random key $k \leftarrow_{\$} \mathcal{K}$. Another useful variant is the notion of *everywhere preimage resistance*, where in this variant it should be difficult for an adversary to find the preimage of any hash function output.

Definition 2.3.6 (Everywhere Preimage Resistance (ePre))**.** *We say that $\mathcal{H}$ is (ε_{ePre}, t)-everywhere-preimage-resistant (ePre) if for any t-bounded adversary $\mathscr{A}$, one has*

$$\max_{Y \in \mathcal{Y}} \Pr_{H \leftarrow_{\$} \mathcal{H}} \left(\mathscr{A}^H = X : H(X) = Y\right) \leq \varepsilon_{ePre}.$$

If a hash function (or family of hash functions) is collision-resistant for some ε, it is called an *almost universal hash function* (AU).

Definition 2.3.7 (Almost universal hash function)**.** *We say that $\mathcal{H}$ is almost-universal (AU) if there exists $\varepsilon > 0$ such that for any distinct $X \neq X' \in \mathcal{X}$,*

$$\Pr_{H \leftarrow_{\$} \mathcal{H}} (H(X) = H(X')) \leq \varepsilon.$$

A stronger variant of the AU notion is called an *almost XOR universal hash function* (AXU). This means that the differential probability of the hash function is small for any efficient adversary. Specifically, it is difficult to find distinct inputs $X \neq X'$ such that $H(X) \oplus H(X') = \delta$ for any given δ.

Definition 2.3.8 (Almost XOR universal hash function)**.** *We say that $\mathcal{H}$ is Almost-XOR-universal (AXU) if there exists $\varepsilon > 0$ such that for any $X \neq X' \in \mathcal{X}$ and $\delta \in \mathcal{Y}$ we have,*

$$\Pr_{H \leftarrow_{\$} \mathcal{H}} (H(X) \oplus H(X') = \delta) \leq \varepsilon.$$

Multiple Input Hash Function. When our hash function takes two inputs, i.e., $\mathcal{X} = \mathcal{X}_l \times \mathcal{X}_r$, we can relax the notion of collision resistance to *right collision resistance* (denoted by RCR) and *left collision resistance* (denoted by LCR). These definitions will be particularly useful in Chapter 6.

Definition 2.3.9 (Right Collision Resistance (RCR))**.** *We say that $\mathcal{H}$ is (ε_{RCR}, t)-right collision-resistant (RCR) if for any t-bounded adversary $\mathscr{A}$, one has*

$$\Pr_{H \leftarrow_{\$} \mathcal{H}} \left(\mathscr{A}^H = (X, X') : X = (X_0, X_1) \neq (X_0', X_1') \wedge H(X_0) = H(X_0')\right) \leq \varepsilon_{RCR}.$$

Definition 2.3.10 (Left Collision Resistance (LCR))**.** *We say that $\mathcal{H}$ is (ε_{LCR}, t)-left collision-resistant (LCR) if for any t-bounded adversary $\mathscr{A}$, one has*

$$\Pr_{H \leftarrow_{\$} \mathcal{H}} \left(\mathscr{A}^H = (X, X') : X = (X_0, X_1) \neq (X'_0, X'_1) \wedge H(X_1) = H(X'_1) \right) \leq \varepsilon_{LCR}.$$

Further, we extend the definition to a hash function with multiple inputs, where the collision resistance property holds for a subset of the inputs. Let $\mathcal{X}_1, \mathcal{X}_2, \ldots \mathcal{X}_n$ denote nonempty sets or spaces and define $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n$. We define the notion of *partial collision resistance*.

Definition 2.3.11 (Partial CR)**.** *We say that $\mathcal{H}$ (defined over message space $\mathcal{X}$) is called (ε_{CR}, t) $(\mathcal{X}_1 \ldots \mathcal{X}_i)$-collision-resistant if for any t-bounded adversary $\mathscr{A}$, one has*

$$\Pr_{H \leftarrow_{\$} \mathcal{H}} \left(\mathscr{A}^H = (X, X') : X[i] \neq X[i] \wedge H(X) = H(X') \right) \leq \varepsilon_{CR}.$$

2.3.5 Authenticated Encryption

2.3.5.1 Formal Definition

In this work, we require formal definitions for several key cryptographic concepts: `IV`-based encryption, which utilizes a random or pseudo-random initialization vector (`IV`) in conjunction with a secret key to ensure unique ciphertexts for identical plaintexts; Nonce-based Authenticated Encryption with Associated Data (AEAD); and Deterministic Authenticated Encryption with Associated Data (DAE). For clarity and simplicity, we refer to Nonce-based Authenticated Encryption with Associated Data simply as AEAD, and to the deterministic variant as DAE.

`IV`-based Encryption: A $(\mathcal{K}, \mathcal{R}, \mathcal{M})$-encryption scheme E is a tuple of algorithms $(\mathsf{E}^+, \mathsf{E}^-)$, defined over the key space $\mathcal{K}$, IV space $\mathcal{R}$, message and ciphertext space $\mathcal{M}$, where

$$\mathsf{E}^+ : \mathcal{K} \times \mathcal{R} \times \mathcal{M} \to \mathcal{M} \qquad \mathsf{E}^- : \mathcal{K} \times \mathcal{R} \times \mathcal{M} \to \mathcal{M}.$$

For all $(k, r) \in \mathcal{K} \times \mathcal{R}$, $\mathsf{E}^-_{k,r}(\cdot) := \mathsf{E}^-(k, r, \cdot)$, referred as the decryption algorithm, is defined as the inverse of $\mathsf{E}^+_{k,r}(\cdot) := \mathsf{E}^+(k, r, \cdot)$, referred as the encryption algorithm, i.e., for all $m \in \mathcal{M}$, $\mathsf{E}^-_{k,r}(\mathsf{E}^+_{k,r}(m)) = m$. It is not necessary to release the IV along with the ciphertext if the IV can be derived from the sequence number or the traffic secret (e.g., TLS 1.3 [273]). Without loss of generality we assume that the IV is released along with the ciphertext in order to facilitate correct decryption. In most cases, including this thesis, $\mathsf{E}^+_{k,r}(\cdot)$ is a length-preserving permutation for all $(k, r) \in \mathcal{K} \times \mathcal{R}$. In this thesis, we only consider random IV schemes, i.e., the IV is sampled uniformly at random for each execution of the encryption algorithm.

Authenticated Encryption Scheme (AE): A $(\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{T}, \mathcal{C})$-authenticated encryption scheme $\mathcal{E}$ is a tuple of algorithms $(\mathcal{E}^+, \mathcal{E}^-)$ defined over the key space $\mathcal{K}$, nonce space $\mathcal{N}$, associated data space $\mathcal{A}$, message space $\mathcal{M}$, ciphertext space $\mathcal{C}$, and tag space $\mathcal{T}$, where:

$$\mathcal{E}^+ : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M} \to \mathcal{C} \times \mathcal{T} \quad \mathcal{E}^- : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M} \cup \{\perp\},$$

and $\perp$ denotes the error symbol indicating authentication failure. For all keys $k \in \mathcal{K}$, we write $\mathcal{E}_k^+(\cdot,\cdot) := \mathcal{E}^+(k,\cdot,\cdot,\cdot)$, referred as the encryption algorithm, and $\mathcal{E}_k^-(\cdot,\cdot,\cdot) := \mathcal{E}^-(k,\cdot,\cdot,\cdot,\cdot)$, referred as the decryption algorithm. For simplicity, we also write $\mathcal{E}_{k,n,a}^+ := \mathcal{E}^+(k,n,a,\cdot)$ and $\mathcal{E}_{k,n,a}^-(k,n,a,\cdot,\cdot)$ for any $n \in \mathcal{N}$ and $a \in \mathcal{A}$.

We say a scheme is *correct* if

$$\mathcal{E}_k^-(a, \mathcal{E}_k^+(n,a,m)) = m, \qquad \forall (k,n,a,m) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M}$$

and we say it is *tidy* if

$$m = \mathcal{E}_k^-(n,a,c,t) \neq \perp \to \mathcal{E}_k^+(n,a,m) = (c,t).$$

In this thesis, we consider all AEs correct and tidy. Moreover, if $\mathcal{N} = \emptyset$ we call the scheme a Deterministic Authenticated Encryption (DAE) and Nonce-based Authenticated Encryption (AE), otherwise. For simplicity, if we discuss an DAE, we simply omit the nonce space from all definitions.

$\mathrm{Re}_{\mathcal{E}}^{\mathrm{priv\$}}(\mathscr{D})$ | $\boxed{\mathrm{Id}_{\mathcal{E}}^{\mathrm{priv\$}}(\mathscr{D})}$

1: $u \leftarrow 0$
2: $b \leftarrow \mathscr{D}^{\textsc{New},\mathsf{O}}$
3: **return** $b =_? 1$

Oracle $\textsc{New}()$

1: $u \leftarrow u + 1$
2: $K_u \twoheadleftarrow_\$ \mathcal{K}$
3: $\boxed{\Gamma_u \twoheadleftarrow_\$ \mathsf{Func}_{\mathsf{lp}}(\mathcal{R} \times \mathcal{M}; \mathcal{M})}$

Oracle $\mathsf{O}(m, i)$

1: **if** $i \notin [u]$ **then return** $\perp$
2: **end if**
3: $R \twoheadleftarrow_\$ \mathcal{R}$
4: $c \leftarrow \mathsf{E}_{K_i,R}^+(m)$
5: $\boxed{c \leftarrow \Gamma_i(R,m)}$
6: **return** (R,c)

Figure 2.6: *The PRIV\$ game: the real world is denoted by* $\mathrm{Re}_{\mathcal{E}}^{\mathrm{priv\$}}(\mathscr{D})$ *and the ideal world by* $\mathrm{Id}_{\mathcal{E}}^{\mathrm{priv\$}}(\mathscr{D})$.

2.3.5.2 Security Notions

The security of both IV-based encryption and DAE hinges on the concept of multi-user (mu) security for authenticated encryption (AE). To establish concrete security guarantees, we follow the approach in [96] and define two multi-user security games

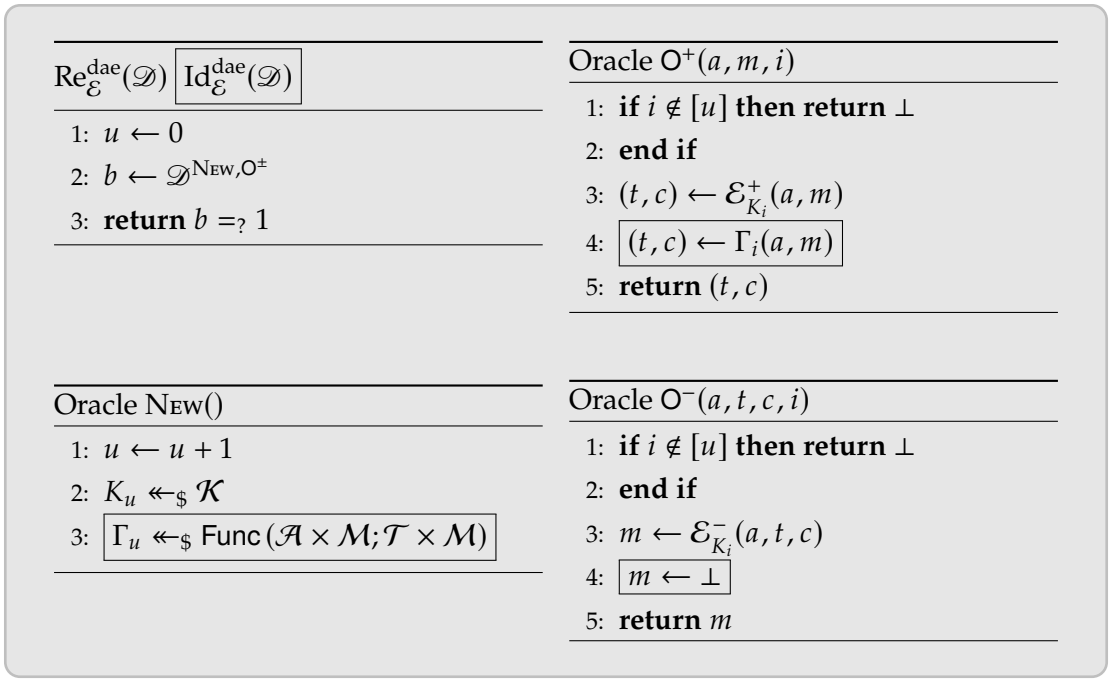

Figure 2.7: *The DAE game: the real world is denoted by* $\mathrm{Re}^{\mathrm{dae}}_{\mathcal{E}}(\mathscr{D})$ *and the ideal world by* $\mathrm{Id}^{\mathrm{dae}}_{\mathcal{E}}(\mathscr{D})$. *In order to avoid trivial wins, the adversary is not allowed decryption queries with an answer he received from an encryption query.*

with an adversary $\mathscr{D}$ that has access to $u \in \mathbb{N}$ number of users. We begin with an informal description of the games, which is then followed by a formal algorithmic definition. The PRIV$ game, which corresponds to the security notion for IV-based encryption, is illustrated in Figure 2.6. Similarly, the DAE game, corresponding to the security notion for DAE, is depicted in Figure 2.7.

PRIV$ game: In this setting, the adversary has access to two sets of functions. In the real world he has access to $\left(\mathsf{E}^{+}_{k_1}, \ldots, \mathsf{E}^{+}_{k_u}\right)$ where the keys $k_1, \ldots, k_u \twoheadleftarrow_{\$} \mathcal{K}$ are drawn uniformly and independently at random and for each execution of $\mathsf{E}^{+}_{k_i}$, a random IV $R \twoheadleftarrow_{\$} \mathcal{R}$ is sampled uniformly at random and independent of everything else. In the ideal world, the adversary has access to $\boldsymbol{\Gamma} = (\Gamma_1, \ldots, \Gamma_u)$ drawn uniformly and independently at random from the set of all functions $f : \mathcal{R} \times \mathcal{M} \to \mathcal{M}$ such that $|f(r, m)| = |m|$ for all $(r, m) \in \mathcal{R} \times \mathcal{M}$, denoted by $\mathsf{Func}_{\mathsf{lp}}(\mathcal{R} \times \mathcal{M}; \mathcal{M})$.

DAE game: In this setting, the adversary has access to two sets of functions. In the real world he has access to $\left(\mathcal{E}^{\pm}_{k_1}, \ldots, \mathcal{E}^{\pm}_{k_u}\right)$ where the keys $k_1, \ldots, k_u \twoheadleftarrow_{\$} \mathcal{K}$ are drawn uniformly and independently at random. In the ideal world, the adversary has access to $\boldsymbol{\Gamma} = (\Gamma_1, \ldots, \Gamma_u)$ drawn uniformly and independently at random from $\mathsf{Func}(\mathcal{A} \times \mathcal{M}; \mathcal{T} \times \mathcal{M})$, while for decryption he simply receives an error symbol $\perp$.

Finally, the advantage of IV-based encryption and DAE for adversary $\mathcal{D}$ are defined as

$$\mathbf{Adv}^{\mathsf{priv\$}}_{\mathscr{D}}(\mathsf{E}) := \left|\Pr\left(\mathrm{Re}^{\mathsf{priv\$}}_{\mathsf{E}}(\mathscr{D}) = 1\right) - \Pr\left(\mathrm{Id}^{\mathsf{priv\$}}_{\mathsf{E}}(\mathscr{D}) = 1\right)\right|,$$

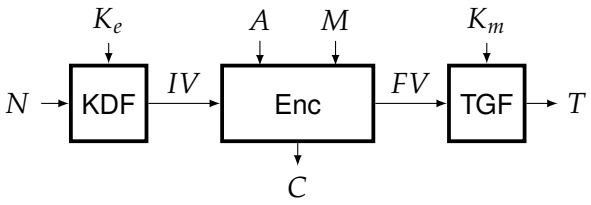

Figure 2.8: *Decomposition for AEAD: Key derivation function (KDF), Encryption scheme (Enc) and Tag derivation function (TGF) .*

$$\mathbf{Adv}^{\mathsf{dae}}_{\mathscr{D}}(\mathcal{E}) := \left|\Pr\left(\mathrm{Re}^{\mathsf{dae}}_{\mathcal{E}}(\mathscr{D}) = 1\right) - \Pr\left(\mathrm{Id}^{\mathsf{dae}}_{\mathcal{E}}(\mathscr{D}) = 1\right)\right|,$$

respectively.

Leakage-Resilient AE. The foundational results on basic cryptographic primitives (PRFs and PRPs) introduced in Section 1.4 prompted the analysis of leakage resilience complete functionalities such as encryption and authentication. This quickly shifted the attention of designers towards AE schemes, which combine both integrity and confidentiality guarantees.

In their seminal paper, Micali and Reyzin demonstrated that indistinguishability-based notions are significantly harder to capture and ensure in the presence of leakage compared to unpredictability-based notions. [241]. Consequently, strong integrity properties with leakage have then been investigated, initially focusing on encryption only [46], and later extending to include both encryption and decryption [47].

As it turns out, weak physical assumptions are needed for most of the computation. For instance, ciphertext integrity—the notion that it is computationally hard for an adversary to produce new ciphertexts that decrypt correctly—can be achieved with full leakage of all the intermediate computations of an Authenticated Encryption scheme, provided that two manipulations of a long-term secret key are performed using a strongly protected block cipher implementation. This approach is obviously insufficient for any type of confidentiality guarantee, as it would leak plaintexts immediately. This conclusion motivated a systematic analysis of composite security definitions, enabling different physical requirements for integrity and confidentiality guarantees in the presence of leakage [157].

Security Notions for AEAD Under Leakage. As seen in [41], investigating the leakage requirements for the components of a scheme often benefits from the following decomposition (see also Figure 2.8):

1. A Key Generation Function (KDF) is used to generate a fresh key K' based on a long-term master key K_e and a nonce N.
2. The message processing part then uses the (optionally fresh) key to encrypt the message blocks.

3. Finally, a Tag Generation Function (TGF) uses the result of the message processing part to output a tag for message authentication. The tag is verified only during decryption.

Following the above decomposition, two modifications have been shown to enhance security under leakage:

- *Key Evolution:* As formalized by Dziembowski and Pietrzak, updating the ephemeral keys of an implementation so that each key is used minimally—and therefore leaks minimally—can improve confidentiality with leakage [130].
- *Strengthened KDF and TGF:* Berti et al. [47] propose that key and tag generating functions make it difficult to compute outputs from inputs and vice versa, can improve security with leakage [46]. For instance, this prevents the recovery of a long-lived master key from temporary keys during the message processing.

In this thesis, we adopt the definitions of integrity and confidentiality with leakage as presented in [41, 47]. An informal description of these notions will suffice for our purposes. In both games, the adversary can perform multiple queries to encryption and decryption oracles enhanced with leakage functions that capture the implementation of an Authenticate Encryption scheme.

- *Ciphertext Integrity with Misuse and Decryption Leakage (CIML2):* In this security game, the adversary's goal is to produce a valid, fresh ciphertext. The implementation is considered secure if the adversary cannot succeed with a good probability.
- *Indistinguishability against Chosen-Ciphertext Adversaries with Misuse Resilience and Encryption Leakage (CCAmL1):* In this security game, the adversary faces a challenge phase in which he picks up two fresh messages, X_0 and X_1, and receives a ciphertext Y_b encrypting X_b for $b \in \{0, 1\}$, with the corresponding leakage. His goal is to guess the bit b and the implementation is considered secure if the adversary cannot succeed with good advantage. In this notion, we assume the adversary has access to encryption only and all nonces are assumed to be fresh (distinct).

For integrity guarantees, it is possible to ensure both misuse-resistance and leakage-resistance simultaneously. However, as discussed in [41], achieving this combination is believed to be impossible under reasonable leakage models for confidentiality guarantees. Consequently, one must choose between Barwell et al.'s CIML2 security [47] and Guo et al.'s CCAmL1 security [157].

2.4 Proof Techniques

This section introduces three key proof techniques that will come in handy when establishing concrete security bounds in Chapter 3, Chapter 4 and Chapter 5.

2.4.1 Statistical Distance and the Coupling Technique

Let Ω be a finite event space and two probability distributions μ and ν are defined on Ω.

Statistical Distance. The *statistical distance* (or total variation) between μ and ν, denoted by $\|\mu - \nu\|$, is defined as:

$$\|\mu - \nu\| := \frac{1}{2} \sum_{x \in \Omega} |\mu(x) - \nu(x)| .$$

It is easy to verify that the statistical distance satisfies the symmetry and *triangle inequality*. Moreover, it is always lying between zero and one. It is one if and only if the support[1] of the probability distributions are disjoint and zero if and only if the distributions are the same. It is well known that the following definitions of statistical distance are all equivalent.

Remark 2.4.1.

$$\|\mu - \nu\| = \max_{S \subset \Omega} \{\mu(S) - \nu(S)\} = \max_{S \subset \Omega} \{\nu(S) - \mu(S)\}$$

In the context of provable security, we can derive a straightforward yet useful result from Remark 2.4.1. Specifically, the advantage of any adversary (including randomized ones) in a distinguishing game is upper bounded by the statistical distance between the corresponding probability distributions of the input/output pairs used during the interaction between the adversary and its corresponding oracle [81, 172, 263]. For completeness, we provide a proof for the case of deterministic adversaries. This can be generalized to randomized adversaries by considering the random coins used during the aforementioned interaction.

Lemma 2.4.1. *For any deterministic distinguisher $\mathcal{A}$ that aims to distinguish between two oracles O_0 and O_1, denote by μ_{O_0} (resp. μ_{O_1}) the distribution induced by the adversary's interaction with O_0 (resp. O_1). Then, one has*

$$\mathbf{Adv}_{O_0;O_1}(\mathcal{A}) \leq \|\mu_{O_0} - \mu_{O_1}\| .$$

Proof. Let Θ be the set of all possible outcomes from an interaction of $\mathcal{A}$ with an oracle. For two random variables $\mathtt{X}_0 \sim \mu_{O_0}$, $\mathtt{X}_1 \sim \mu_{O_1}$ and any bit $b \in \{0,1\}$,

$$\begin{aligned}
\mathbf{Adv}_{O_0;O_1}(\mathcal{A}) &= \left| \Pr\left(\mathcal{A}^{O_1} = b\right) - \Pr\left(\mathcal{A}^{O_0} = b\right) \right| \\
&= \left| \Pr\left(\mathcal{A}(\mathtt{X}_0) = b\right) - \Pr\left(\mathcal{A}(\mathtt{X}_1) = b\right) \right| \\
&= \left| \sum_{\tau \in \Theta} \Pr\left(\mathcal{A}(\tau) = b \mid \mathtt{X}_0 = \tau\right) \Pr\left(\mathtt{X}_0 = \tau\right) - \Pr\left(\mathcal{A}(\tau) = b \mid \mathtt{X}_1 = \tau\right) \Pr\left(\mathtt{X}_1 = \tau\right) \right| \\
&= \left| \sum_{\tau \in \Theta} \Pr\left(\mathcal{A}(\tau) = b \mid \mathtt{X}_0 = \tau\right) \left(\Pr\left(\mathtt{X}_0 = \tau\right) - \Pr\left(\mathtt{X}_1 = \tau\right)\right) \right|
\end{aligned}$$

[1] The set of all elements having positive probability.

$$\leq \sum_{\tau \in \Theta} \Pr\left(\mathscr{A}(\tau) = b \mid \mathtt{X}_0 = \tau\right) \left|\Pr\left(\mathtt{X}_0 = \tau\right) - \Pr\left(\mathtt{X}_1 = \tau\right)\right| ,$$

where the last equality follows from the fact that the distinguisher is indifferent to whether τ came from μ_{O_0} or μ_{O_1}, and the inequality follows from the triangle inequality. Thus, we conclude that

$$\begin{aligned}
\mathbf{Adv}_{O_0;O_1}\left(\mathscr{A}\right) &\leq \frac{1}{2} \sum_{b \in \{0,1\}} \sum_{\tau \in \Theta} \Pr\left(\mathscr{A}(\tau) = b \mid \mathtt{X}_0 = \tau\right) \left|\Pr\left(\mathtt{X}_0 = \tau\right) - \Pr\left(\mathtt{X}_1 = \tau\right)\right| \\
&= \frac{1}{2} \sum_{\tau \in \Theta} \left|\Pr\left(\mathtt{X}_0 = \tau\right) - \Pr\left(\mathtt{X}_1 = \tau\right)\right| \\
&= \left\|\mu_{O_0} - \mu_{O_1}\right\| .
\end{aligned}$$

□

Coupling. A *coupling* of μ and ν is a distribution λ on $\Omega \times \Omega$ such that for all $x \in \Omega$, $\sum_{y \in \Omega} \lambda(x, y) = \mu(x)$ and for all $y \in \Omega$, $\sum_{x \in \Omega} \lambda(x, y) = \nu(y)$. In other words, λ is a joint distribution whose marginal distributions are resp. μ and ν. The following lemma is the main technical ingredient of the well-known coupling technique [7]. A proof of this lemma is available in [209], and restated here for completeness.

Lemma 2.4.2 (Coupling Lemma)**.** *Let μ and ν be probability distributions on a finite event space Ω. Let λ be a coupling of μ and ν, and let $(X, Y) \sim \lambda$ (i.e. (X,Y) is a random variable sampled according to distribution λ). Then $\left\|\mu - \nu\right\| \leq \Pr\left(X \neq Y\right)$.*

Proof. Let λ be the coupling of μ and ν, and $(X, Y) \sim \lambda$. By definition, we have that for any $z \in \Omega$, $\lambda(z, z) \leq \min\left\{\mu(z), \nu(z)\right\}$. Moreover, $\Pr\left(X = Y\right) = \sum_{z \in \Omega} \lambda(z, z)$. Hence we have:

$$\Pr\left(X = Y\right) \leq \sum_{z \in \Omega} \min\left\{\mu(z), \nu(z)\right\} .$$

Therefore from Remark 2.4.1,

$$\begin{aligned}
\Pr\left(X \neq Y\right) &\geq 1 - \sum_{z \in \Omega} \min\left\{\mu(z), \nu(z)\right\} \\
&= \sum_{z \in \Omega} \left(\mu(z) - \min\left\{\mu(z) - \nu(z)\right\}\right) \\
&= \sum_{\substack{z \in \Omega \\ \mu(z) \geq \nu(z)}} \left(\mu(z) - \nu(z)\right) \\
&= \max_{S \subset \Omega} \left\{\mu(S) - \nu(S)\right\} \\
&= \left\|\mu - \nu\right\| .
\end{aligned}$$

□

To provide some intuition for the Coupling Lemma, let's consider a straightforward example involving the coupling of two coins.

Example for Coupling Coins. Let $\mathtt{C}_1$ be a random variable representing a balanced coin and $\mathtt{C}_2$ represents an unbalanced coin that outputs $\mathtt{T}$ with probability $3/4$. In this setting, the sample space is $\Omega = \{\mathtt{T}, \mathtt{H}\}$ and we denote by μ the distribution corresponding to $\mathtt{C}_1$ and ν to $\mathtt{C}_2$. Let us define two couplings of μ and ν over $\Omega \times \Omega$ the following way:

$$\lambda_1(x, y) := \mu(x)\nu(y), \qquad \forall x, y \in \Omega$$
$$\lambda_2(\mathtt{T}, \mathtt{T}) := 1/2, \qquad \lambda_2(\mathtt{H}, \mathtt{T}) := \lambda_2(\mathtt{H}, \mathtt{H}) := 1/4, \qquad \lambda_2(\mathtt{T}, \mathtt{H}) := 0.$$

By definition, λ_1 corresponds to a coupling where the coins are independent and λ_2 where the coins are maximally correlated. As for the statistical distance between μ and ν, one has, $\left\|\mu - \nu\right\| = \frac{1}{2}\left(\frac{2}{4}\right) = 1/4$. Moreover, it is easy to see that

$$(\mathtt{C}_1, \mathtt{C}_2) \sim \lambda_1 \rightarrow \Pr\left(\mathtt{C}_1 \neq \mathtt{C}_2\right) = 1/2,$$
$$(\mathtt{C}_1, \mathtt{C}_2) \sim \lambda_2 \rightarrow \Pr\left(\mathtt{C}_1 \neq \mathtt{C}_2\right) = \left\|\mu - \nu\right\| = 1/4.$$

Thus, here the maximally correlated coupling is clearly better. In general, to make an optimal use of Lemma 2.4.2, we try to minimize $\Pr(X \neq Y)$ to get the best upper bound on $\left\|\mu - \nu\right\|$.

2.4.2 The Expectation Method and H-Coefficient Technique

Let $\mathscr{A}$ be an unbounded distinguisher making some $q \in \mathbb{N}$ queries. Since $\mathscr{A}$ is unbounded we can assume it also deterministic, as an unbounded probabilistic distinguisher can be replaced by a deterministic unbounded distinguisher with at least the same advantage by simulating all the random coins and maximizing the advantage.

Transcripts. The query-response tuple of $\mathscr{A}$ interacting with an oracle O is called a *transcript*. We note that the transcript might include additional information revealed to the distinguisher at the end of the security game. For $b \in \{0, 1\}$, we denote by $\mathtt{T}_b$ the transcript when $\mathscr{A}$ is interacting with O_b. Note that $\mathtt{T}_b$ is a random variable and the probability of realizing that transcript is often called *the interpolation probability* with respect to the oracle O_b. While considering the interpolation probability we assume that the distinguisher is making the queries listed in the transcript, as otherwise the probability would be zero. Thus, we are only concerned with the oracles responses. We say a transcript τ is *attainable* if $\Pr(\mathtt{T}_0 = \tau) > 0$.

The following method called the *expectation method* is particularly powerful tool in producing upper bounds on the distinguishing advantage by partitioning the transcripts to "good" and "bad" and using a simple probabilistic argument.

Lemma 2.4.3 (Expectation Method [171])**.** *Let Θ be the set of all attainable transcripts. Assume that there is a partition, $\Theta_{\text{good}} \cup \Theta_{\text{bad}} = \Theta$, a function $\varepsilon_r : \Omega \rightarrow [0, \infty)$ and a constant $\varepsilon_b > 0$ such that:*

1. $\Pr(\tau_0 \in \Theta_{\text{bad}}) \leq \varepsilon_b$,

2. *For any* $\tau \in \Theta_{\text{good}}$, $1 - \dfrac{\Pr(\mathsf{T}_1 = \tau)}{\Pr(\mathsf{T}_0 = \tau)} \leq \varepsilon_r(\tau)$.

Then for any deterministic distinguisher $\mathscr{A}$,

$$\mathbf{Adv}^{O_0;O_1}(\mathscr{A}) \leq \varepsilon_b + \mathbb{E}(\varepsilon_r(\tau_0)),$$

where $\mathbb{E}(\cdot)$ *is the expectation function of a random variable.*

Proof. For a bit $c \in \{0,1\}$, let Ω_c be the set of all transcripts for which $\mathscr{A}$ returns c. Therefore, if $\mathscr{A}$ is interacting with the oracle $O_{c'}$ for a bit $c' \in \{0,1\}$ returning c, then one has,

$$\Pr\left(\mathscr{A}^{O_c} = c'\right) = \sum_{\tau \in \Omega_{c'}} \Pr(\mathsf{T}_c = \tau)$$

Hence,

$$\mathbf{Adv}^{O_0;O_1}(\mathscr{A}) \leq \sum_{\tau \in \Omega_c} |\Pr(\tau_1 = \tau) - \Pr(\tau_0 = \tau)|$$

So we conclude that,

$$\mathbf{Adv}^{O_0;O_1}(\mathscr{A}) \leq \frac{1}{2} \sum_{\tau \in \Omega} |\Pr(\mathsf{T}_1 = \tau) - \Pr(\mathsf{T}_0 = \tau)| = \|\mathsf{T}_1 - \mathsf{T}_0\|$$

Combining those with Remark 2.4.1 yields,

$$\begin{aligned}
\mathbf{Adv}^{O_0;O_1}(\mathscr{A}) &\leq \sum_{\tau \in \Theta} \max\{0, \Pr(\mathsf{T}_1 = \tau) - \Pr(\mathsf{T}_0 = \tau)\} \\
&\leq \sum_{\tau \in \Theta} \max\left\{0, \left(1 - \frac{\Pr(\mathsf{T}_1 = \tau)}{\Pr(\mathsf{T}_0 = \tau)}\right) \cdot \Pr(\mathsf{T}_0 = \tau)\right\} \\
&\leq \sum_{\tau \in \Theta_{\text{bad}}} \Pr(\mathsf{T}_0 = \tau) + \sum_{\tau \in \Theta_{\text{good}}} \Pr(\mathsf{T}_0 = \tau) \cdot \max\left\{0, 1 - \frac{\Pr(\mathsf{T}_1 = \tau)}{\Pr(\mathsf{T}_0 = \tau)}\right\} \\
&\leq \Pr(\tau \in \Theta_{\text{bad}}) + \sum_{\tau \in \Omega_g} \Pr(\mathsf{T}_0 = \tau)\, \varepsilon_r(\mathsf{T}_0) \\
&\leq \varepsilon_b + \mathbb{E}(\varepsilon_r(\mathsf{T}_0)).
\end{aligned}$$

□

The famous H-coefficient technique developed by Patarain [263] is a corollary of the expectation method when ε_r is a constant.

Corollary 2.4.1 (H-Coefficient Technique [262, 81])**.** *Let* Θ *be the set of all attainable transcripts. Assume that there is a partition of* Θ, $\Theta_{\text{bad}} \cup \Theta_{\text{good}} = \Omega$, *and two constants* $\varepsilon_b, \varepsilon_r > 0$ *such that:*

1. $\Pr(\mathsf{T}_0 \in \Theta_{\text{bad}}) \leq \varepsilon_b$,
2. *For any* $\tau \in \Theta_{\text{good}}$, $1 - \dfrac{\Pr(\mathsf{T}_1 = \tau)}{\Pr(\mathsf{T}_0 = \tau)} \leq \varepsilon_r$.

Then for any deterministic distinguisher $\mathscr{A}$,

$$\mathbf{Adv}^{O_0;O_1}(\mathscr{A}) \leq \varepsilon_b + \varepsilon_r.$$

2.4.3 The Fundamental Lemma of Game-Playing

In this thesis, we formalize the security of cryptographic schemes through the notion of distinguishing games as described in Section 2.2.1. In this setting, a distinguisher $\mathscr{A}$ aims to distinguish between two worlds O_0, dubbed the ideal world, and O_1, dubbed the real world. We denote by G_c the game corresponding to the interaction of $\mathscr{A}$ with O_c for $c \in \{0,1\}$. We view G_0 and G_1 as though they are written as pseudo-code parts and assume that G_0 and G_1 are syntactically the same up to setting some flag to the value *bad*, we call these games *identical up to bad* games.

A particularly powerful result asserts that the advantage of $\mathscr{A}$ is upper bounded by the probability that one of these games sets its flag to bad. A thorough introduction to proofs by game-playing can be found in [40].

Lemma 2.4.4 (Lemma 2 in [40])**.** *Let G and H be identical up to bad games and let $\mathscr{A}$ be an adversary interacting with them. Then, one has*

$$\mathbf{Adv}_{G;H}(\mathscr{A}) \leq \Pr(G \text{ sets to bad}), \qquad \mathbf{Adv}_{G;H}(\mathscr{A}) \leq \Pr(H \text{ sets to bad})$$

As a corollary we show that the famous PRF-PRP switching lemma can be proved using Lemma 2.4.4.

Corollary 2.4.2 (Lemma 1 in [40])**.** *For any $n \in \mathbb{N}$ and adversary $\mathscr{A}$ interacting with a random function $\Gamma \twoheadleftarrow_{\$} \mathsf{Func}(n;n)$ or a random permutation $\mathsf{P} \twoheadleftarrow_{\$} \mathsf{Perm}(n)$ and making at most q queries, one has*

$$\mathbf{Adv}_{\Gamma;\mathsf{P}}(\mathscr{A}) \leq \frac{q(q-1)}{2^{n+1}}.$$

Proof. Let S_0 be the game implementing a random function by sampling an output Y at random and setting a flag to bad if Y collides with a previous output. Let S_1 be identical to S_0 except that when S_0 sets to bad then S_1 resamples the output from the set of authorized values (the values that do not collide with previous outputs). Then, it is easy to see that S_1 implements a random permutation. Moreover, S_0 and S_1 are identical up to bad by definition. Hence, by Lemma 2.4.4, one has

$$\begin{aligned}\mathbf{Adv}_{\Gamma;\mathsf{P}}(\mathscr{A}) &\leq \Pr(S_0 \text{ sets to bad}) \\ &\leq \Pr\left(\exists i \in [2;q] : Y_i \in \{Y_1, \ldots, Y_{i-1}\}\right) \\ &\leq \frac{q(q-1)}{2^{n+1}},\end{aligned}$$

where $Y_1, \ldots, Y_q$ are the outputs in game S_0. □

Part I

Analysis of Tweakable Block Ciphers

3

Single Pseudorandom Permutation Based Tweakable Enciphering Scheme

In this chapter, we study the design of efficient block ciphers with large domains (e.g., wn-bit blocks) that achieve security beyond the birthday bound. This is typically accomplished using an SPN (Substitution-Permutation Network) structure, where a keyed permutation layer is applied to the entire state, which is then divided into w smaller blocks. Each block undergoes an S-box operation, and the process is repeated iteratively. We discuss the security implications of these structures, drawing on prior research that has analyzed SPNs under various conditions, such as the use of secret random permutations and different types of linear layers. Notably, the security of SPNs has been proven to improve with the number of rounds, especially when non-linear or tweakable components are used.

Our contributions in this chapter focus on two main areas: first, we analyze the security of two-round SPNs with a single S-box, proving that they can achieve beyond-birthday-bound multi-user security with independent keys and an inner linear permutation that is simpler compared to previous works [94]. Second, we introduce a new tweakable enciphering scheme ($\texttt{CTET}^{+}$) that extends these security guarantees to practical applications, particularly when instantiated with secure block ciphers like AES. This new scheme is notable for providing beyond-birthday-bound security using a single permutation, making it an efficient and practical solution for secure encryption in various contexts. We refer the reader to Section 1.6.1, for a complete overview of the subject and an elaborate discussion of our contributions.

3.1 Regular Blockwise Universal Tweakable Permutations

Throughout, we fix a parameter $w \in \mathbb{N}$ and consider the message space $\mathcal{X} = \{0,1\}^{wn}$. We extend the notion of permutations to a parameterized permutation called a tweakable permutation, which introduces an additional parameter called a tweak. Namely, for a tweak space $\mathcal{T}$ and message space $\mathcal{X}$, a *tweakable permutation* $\widetilde{\pi}$ is a function

$\widetilde{\pi} : \mathcal{T} \times \mathcal{X} \to \mathcal{X}$ such that for any tweak $t \in \mathcal{T}$, $\widetilde{\pi}(t, \cdot)$ is a permutation of $\mathcal{X}$. We denote the set of all tweakable permutations with tweak space $\mathcal{T}$ by $\widetilde{\mathsf{Perm}}(\mathcal{T}; \mathcal{X})$.

Keyed Tweakable Permutations. A keyed tweakable permutation with key space $\mathcal{K}$, tweak space $\mathcal{T}$ and message space $\mathcal{X}$ is a mapping $T : \mathcal{K} \times \mathcal{T} \times \mathcal{X} \to \mathcal{X}$ such that, for any key $k \in \mathcal{K}$,

$$(t, x) \mapsto T(k, t, x)$$

is a tweakable permutation with tweak space $\mathcal{T}$ and message space $\mathcal{X}$. We sometimes write $T(k, t, x)$ as $T_k(t, x)$ or $T_{k,t}(x)$. In particular, $T_{k,t}$ is a permutation of $\mathcal{X}$ for each $(k, t) \in \mathcal{K} \times \mathcal{T}$.

3.1.1 Blockwise Universality and Regularity

The core part of our security proof is to compute a lower bound on the number of possible intermediate values that map a tuple of plaintexts to a tuple of ciphertexts, given some conditions on the permutation. A key point in such proofs is the ability to control collisions between inputs to the inner primitive. Hence, we are going to need our keyed layers to satisfy a universality property as follows.

Definition 3.1.1. *A keyed tweakable permutation $T : \mathcal{K} \times \mathcal{T} \times \mathcal{X} \longrightarrow \mathcal{X}$ is called (δ, δ')-blockwise universal ((δ, δ')-BU) if the following conditions hold:*

1. *For all distinct $(t, x, i), (t', x', i') \in \mathcal{T} \times \mathcal{X} \times [1; w]$, we have*

$$\Pr_{k \xleftarrow{\$} \mathcal{K}} \left(T_{k,t}(x)_i = T_{k,t'}(x')_{i'} \right) \leq \delta.$$

2. *For all $(t, x, i, c) \in \mathcal{T} \times \mathcal{X} \times [1; w] \times \{0, 1\}^n$, we have*

$$\Pr_{k \xleftarrow{\$} \mathcal{K}} \left(T_{k,t}(x)_i = c \right) \leq \delta'.$$

Let $T^{-1} : \mathcal{K} \times \mathcal{T} \times \mathcal{X} \longrightarrow \mathcal{X}$ be a tweakable block cipher such that $T^{-1}(k, t, x) = (T_{k,t})^{-1}(x)$ for each $(k, t, x) \in \mathcal{K} \times \mathcal{T} \times \mathcal{X}$. If T and T^{-1} are both (δ, δ')-blockwise universal, then T is called (δ, δ')-super blockwise universal ((δ, δ')-SBU).

In our security proof, it will be essential to ensure that, when at most one of the w blocks of an input (resp. output) of the second keyed permutation layer is fixed to an arbitrary value[1], and the remaining blocks are chosen uniformly at random without replacement in some set, the distribution of the output (resp. input) is close enough to uniform. More specifically, we will only be interested in the probability that the w blocks of the output (resp. input) are pairwise distinct and belong to a specific set of authorized values, in order to avoid collisions with previously queried inputs/outputs to the public permutation. More formally, we define the regularity of a permutation as follows.

[1] The remaining cases will be easy to rule out thanks to the previously defined blockwise universal property.

Definition 3.1.2. *A permutation $P \in \mathsf{Perm}(n)$ is called* regular *if for any $A, B \subset \{0,1\}^n$, the following three conditions are satisfied.*

1. *The number of elements $x \in (\{0,1\}^n \setminus A)^{*w}$ such that $P(x) \in (\{0,1\}^n \setminus B)^{*w}$ is at least*
$$(2^n - |A|)_w \left(1 - \frac{w|B| + w(w-1)/2}{2^n - |A| - w + 1}\right).$$
2. *For any $a \in A$ and $i \in [1;w]$, the number of elements $x \in (\{0,1\}^n)^{*w}$ such that $x_i = a$, $x_j \notin A$ for $j \neq i$ and $P(x) \in (\{0,1\}^n \setminus B)^{*w}$ is at least*
$$(2^n - |A|)_{w-1} \left(1 - \frac{w|B| + w(w-1)/2}{2^n - |A| - w + 2}\right).$$
3. *For any $b \in B$ and $i \in [1;w]$, the number of elements $x \in (\{0,1\}^n \setminus A)^{*w}$ such that $P(x) \in (\{0,1\}^n)^{*w}$, $P(x)_i = b$ and $P(x)_j \notin B$ for $j \neq i$ is at least*
$$(2^n - |B|)_{w-1} \left(1 - \frac{w|A| + (w-1)w/2}{2^n - |B| - w + 2}\right).$$

Similarly, we say a keyed tweakable permutation T is *regular* if $T(k,t,\cdot)$ is regular for any $(k,t) \in \mathcal{K} \times \mathcal{T}$. This technical definition is actually rather natural. If we consider the first condition, we want a lower bound on the number of tuples $x \in (\{0,1\}^n \setminus A)^{*w}$ such that $P(x) \in (\{0,1\}^n \setminus B)^{*w}$. There are exactly $(2^n - |A|)_w$ elements in $(\{0,1\}^n \setminus A)^{*w}$, while $P(x)$ should satisfy $w|B| + \frac{w(w-1)}{2}$ conditions since all w blocks of $P(x)$ should be distinct and outside B. Intuitively, the first point of the definition essentially requires that each of the conditions on $P(x)$ removes at most $(2^n - |A|)_{w-1}$ possibilities for x. Other lower bounds can be derived similarly. As we will see, there is a simple sufficient condition for an affine map to be regular.

Lemma 3.1.1. *Let $R = (\{0,1\}^n, +, \times)$ is a unitary ring of characteristic 2. Let $P : \mathcal{X} \to \mathcal{X}$ be an affine map where*

$$\begin{aligned} P : \mathcal{X} &\longrightarrow \mathcal{X} \\ x &\longmapsto Mx + a, \end{aligned}$$

for a $w \times w$ invertible matrix M over the ring R and $a \in \mathcal{X}$, identifying elements in $\mathcal{X}$ with w-dimensional column vectors over R. Then P is regular if M satisfies the following conditions:

1. *Each row of M and M^{-1} contains at least two invertible entries.*
2. *The sum of any two rows of M contains at least two invertible entries.*
3. *The sum of any two rows of M^{-1} contains at least two invertible entries.*

Proof. To lower bound the number of elements $x = (x_1, \ldots, x_w)$ such that $x \in (\{0,1\}^n \setminus A)^{*w}$ and $P(x) \in (\{0,1\}^n \setminus B)^{*w}$, we first fix $i \in [1;w]$ and $b \in B$. Suppose that the j-th entry of the i-th row of M is Invertible. Then we select distinct $w-1$ values $x_1, \ldots, x_w$ except x_j, all from $\{0,1\}^n \setminus A$. The number of possible choices for these values is

$(2^n - |A|)_{w-1}$. Since the equation $P(x)_i = b$ uniquely determines x_j, the number of $x \in (\{0,1\}^n \setminus A)^{*w}$ such that $P(x)_i \in B$ for some $i = 1, \ldots, w$ is at most $w|B|(2^n - |A|)_{w-1}$.

Next, we fix two different indices $i_1, i_2 \in [1; w]$. Suppose that the j-th entry is Invertible in the sum of the i_1-th row and the i_2-th row of M. We select distinct $w-1$ values $x_1, \ldots, x_w$ except x_j, all from $\{0,1\}^n \setminus A$. Then the equation $P(x)_{i_1} = P(x)_{i_2}$ will uniquely determine x_j. So the number of $x \in (\{0,1\}^n \setminus A)^{*w}$ such that $P(x)_{i_1} = P(x)_{i_2}$ for some $1 \le i_1 < i_2 \le w$ is at most $\binom{w}{2}(2^n - |A|)_{w-1}$.

Overall, the number of "bad choices" is at most $w|B|(2^n - |A|)_{w-1} + \binom{w}{2}(2^n - |A|)_{w-1}$, and hence the number of elements $x \in (\{0,1\}^n \setminus A)^{*w}$ such that $P(x) \in (\{0,1\}^n \setminus B)^{*w}$ is at least

$$(2^n - |A|)_w - w|B|(2^n - |A|)_{w-1} - \binom{w}{2}(2^n - |A|)_{w-1} \\ = (2^n - |A|)_w \left(1 - \frac{w|B| + w(w-1)/2}{2^n - |A| - w + 1}\right).$$

To prove the second condition of regularity, we will assume that $x_1 = a$ for some $a \in A$ (without loss of generality), and lower bound the number of element $x = (x_1, \ldots, x_w)$ such that $x \in (\{0,1\}^n)^{*w}$, $x_i \notin A$ for $i \ge 2$ and $P(x) \in (\{0,1\}^n \setminus B)^{*w}$. We first fix $i \in [1; w]$ and $b \in B$. Then we can find an index $j \ge 2$ such that the j-th entry of the i-th row of M is Invertible since each row of M contains at least two invertible entries. We select distinct $w-2$ values $x_2, \ldots, x_w$ except x_j, all from $\{0,1\}^n \setminus A$. The number of possible choices for these values is $(2^n - |A|)_{w-2}$. Since the equation $P(x)_i = b$ uniquely determines x_j, the number of elements $x \in (\{0,1\}^n)^{*w}$ such that $x_1 = a$, $x_j \notin A$ for $j \ge 2$ and $P(x)_i \in B$ for some $i = 1, \ldots, w$ is at most $w|B|(2^n - |A|)_{w-2}$.

Next, we fix two different indices $i_1, i_2 \in [1; w]$. We can find an index $j \ge 2$ such that the j-th entry is Invertible in the sum of the i_1-th row and the i_2-th row of M. We select distinct $w-2$ values $x_2, \ldots, x_w$ except x_j, all from $\{0,1\}^n \setminus A$. Then the equation $P(x)_{i_1} = P(x)_{i_2}$ will uniquely determine x_j. So the number of $x \in (\{0,1\}^n \setminus A)^{*w}$ such that $x_1 = a$, $x_j \notin A$ for $j \ge 2$ and $P(x)_{i_1} = P(x)_{i_2}$ for some $1 \le i_1 < i_2 \le w$ is at most $\binom{w}{2}(2^n - |A|)_{w-2}$.

By discarding at most $w|B|(2^n - |A|)_{w-2} + \binom{w}{2}(2^n - |A|)_{w-2}$ elements, the number of elements $x \in (\{0,1\}^n)^{*w}$ such that $x_1 = a$, $x_j \notin A$ for $j \ge 2$ and $P(x) \in (\{0,1\}^n \setminus B)^{*w}$ is lower bounded by

$$(2^n - |A|)_{w-1} - w|B|(2^n - |A|)_{w-2} - \binom{w}{2}(2^n - |A|)_{w-2} \\ = (2^n - |A|)_{w-1} \left(1 - \frac{w|B| + w(w-1)/2}{2^n - |A| - w + 2}\right).$$

To prove the third condition, we fix $b \in B$ and $i \in [1; w]$. Then the number of elements $x \in (\{0,1\}^n \setminus A)^{*w}$ such that $P(x) \in (\{0,1\}^n)^{*w}$, $P(x)_i = b$ and $P(x)_j \notin B$ for $j \ne i$ is the same as the number of elements $y \in (\{0,1\}^n)^{*w}$ such that $y_i = b$, $y_j \notin B$ for $j \ne i$ and

$P^{-1}(y) \in (\{0,1\}^n \setminus A)^{*w}$. The linear part of P^{-1} is represented by M^{-1}, and in the same way as we proved the second condition, we can prove that this number is lower bounded by

$$(2^n - |B|)_{w-1} \left(1 - \frac{w|A| + w(w-1)/2}{2^n - |B| - w + 2}\right).$$

□

3.1.2 An Efficient Regular SBU Tweakable Permutation

Halevi [162] proposed an efficient xor-blockwise universal construction BPE, and it has been made tweakable by Cogliati et al [94]. In this section, we will show that a slightly generalized version of the tweakable variant is also regular. Let us assume that $n = m \cdot d$ for two positive integers m and d. We begin with the definition of $\mathtt{TBPE}^d$.

Assuming $2^m \geq w + 3$, denote by $\mathbb{F} = \mathbb{F}_{2^m}$ and let R be the ring $\mathbb{F}^d$, where addition and multiplication are simply done component-wise. For each $k \in R$, define a $w \times w$ matrix over R, $M_k := A_k \oplus I$, where I is the identity matrix and

$$A_k = \begin{bmatrix} k & k^2 & & k^w \\ k & k^2 & & k^w \\ & & \ddots & \\ k & k^2 & & k^w \end{bmatrix}. \tag{3.1}$$

Precisely, $(A_k)_{i,j} = k^j$ for $1 \leq i, j \leq w$. Let z be a primitive element of $\mathbb{F}$, and let

$$\mathcal{K} = \left\{k = (k_1, \ldots, k_d) \in R : \forall j \in \{1 \ldots, d\}, \sum_{i=0}^{w} k_j^i \neq 0\right\} \times R$$

and $\mathcal{T} = \{0,1\}^n$ denote the key space and the tweak space, respectively. Then $\mathtt{TBPE}^d$ is defined as follows.

$$\begin{aligned} \mathtt{TBPE}^d : \mathcal{K} \times \mathcal{T} \times \mathcal{X} &\longrightarrow \mathcal{X} \\ ((k, k'), t, x) &\longmapsto M_k(x \oplus b_t) \oplus a_{k'} \oplus b_t, \end{aligned}$$

where we identify $x \in \mathcal{X}$ with an element of R^w, and

$$a_{k'} = \begin{bmatrix} k' \\ zk' \\ \vdots \\ z^{w-1}k' \end{bmatrix}, \qquad b_t = \begin{bmatrix} t \\ t \\ \vdots \\ t \end{bmatrix}.$$

It is easy to check that M_k is invertible if $\sum_{i=0}^{w} k_j^i \neq 0$ for $j = 1, \ldots, d$; precisely,

$$M_k^{-1} = I \oplus \frac{A_k}{k^*},$$

where $k^* := \sum_{i=0}^{w} k^i$. Therefore, each pair of key $(k, k') \in \mathcal{K}$ and tweak $t \in \mathcal{T}$ defines a permutation $\texttt{TBPE}^d_{k,k',t}$ on $\{0,1\}^{wn}$; let

$$(\texttt{TBPE}^d)^{-1} : \mathcal{K} \times \mathcal{T} \times \mathcal{X} \longrightarrow \mathcal{X}$$
$$(k, k', t, x) \longmapsto (\texttt{TBPE}^d_{k,k',t})^{-1}(x).$$

It is easy to see that $\texttt{TBPE}^1$ actually corresponds to the TBPE tweakable permutation as defined in [94]. Moreover, this generalized $\texttt{TBPE}^d$ permutation is actually equivalent to the parallel application of d independent TBPE tweakable permutations over $\mathbb{F}$. Hence, using [94, Lemma 2], we can easily prove the following Lemma.

Lemma 3.1.2. *Let TBPE^d be the keyed tweakable permutation as defined above, and let $(\mathit{TBPE}^d)^{-1}$ be its inverse.*

1. *For all distinct $(t, x, i), (t', x', i') \in \mathcal{T} \times \mathcal{X} \times [1; w]$, we have*

$$\Pr_{(k,k') \xleftarrow{\$} \mathcal{K}} \left(\mathit{TBPE}^d_{k,k',t}(x)_i = \mathit{TBPE}^d_{k,k',t'}(x')_{i'} \right) \leq \frac{w^d}{(2^m - w)^d}.$$

2. *For all $(t, x, i, c) \in \mathcal{T} \times \mathcal{X} \times [1; w] \times \{0,1\}^n$, we have*

$$\Pr_{(k,k') \xleftarrow{\$} \mathcal{K}} \left(\mathit{TBPE}^d_{k,k',t}(x)_i = c \right) \leq \frac{1}{2^n}.$$

3. *For all distinct $(t, x, i), (t', x', i') \in \mathcal{T} \times \mathcal{X} \times [1; w]$, we have*

$$\Pr_{(k,k') \xleftarrow{\$} \mathcal{K}} \left((\mathit{TBPE}^d)^{-1}_{k,k',t}(x)_i = (\mathit{TBPE}^d)^{-1}_{k,k',t'}(x')_{i'} \right) \leq \frac{w^d}{(2^m - w)^d}.$$

4. *For all $(t, x, i, c) \in \mathcal{T} \times \mathcal{X} \times [1; w] \times \{0,1\}^n$, we have*

$$\Pr_{(k,k') \xleftarrow{\$} \mathcal{K}} \left((\mathit{TBPE}^d)^{-1}_{k,k',t}(x)_i = c \right) \leq \frac{(w+1)^d}{(2^m - w)^d}.$$

Using Lemma 3.1.1, we can now prove the regularity of $\texttt{TBPE}^d$.

Lemma 3.1.3. *If $w \geq 3$, then TBPE is regular.*

Proof. For a fixed $(k, t) \in \mathcal{K} \times \mathcal{T}$, the linear part of $\texttt{TBPE}^d$ is represented by $M_k = A_k + I$, and $M_k^{-1} = I \oplus \frac{A_k}{k^*}$, where A_k is defined in Equation (3.1). One can easily see that M_k and M_k^{-1} satisfy the three conditions of Lemma 3.1.1, and hence $\texttt{TBPE}^d$ is regular when $w \geq 3$.[2] □

From Lemma 3.1.2 and Lemma 3.1.3, it follows that $\texttt{TBPE}^d$ is (δ, δ')-super blockwise universal and regular with $\delta = \frac{w^d}{(2^m - w)^d}$, $\delta' = \frac{(w+1)^d}{(2^m - w)^d}$.

3.2 The CTET$^+$ Tweakable Enciphering Scheme

In this section, we define our new tweakable enciphering scheme based on a single S-Box S and three tweakable permutations $T_{k_0}, L_{k_1}, T_{k_2}$, for some keys k_0, k_1, k_2.

[2] When $w = 2$, we can make $\texttt{TBPE}^d$ regular simply by discarding the square roots of 1 from $\mathcal{K}$.

3.2.1 Notional Setup

Let J_w (resp. I_w) denote the $w \times w$ all-ones matrix (resp. identity matrix) over $\mathbb{F}$, and let z be a primitive element of $\mathbb{F}$. We define a keyed tweakable permutation L as follows.

$$\begin{aligned} L : \mathcal{K}' \times \mathcal{T} \times \mathcal{X} &\longrightarrow \mathcal{X} \\ (k, t, x) &\longmapsto (2J_w \oplus I_w)x \oplus a_k \oplus b_t \end{aligned}$$

where $\mathcal{K}' = \mathcal{T} = \{0,1\}^n$, a_k, b_t are defined as in Section 3.1.2, and 2 is a shorthand for the n-bit block $0 \cdots 010$ (similarly, 1 is a shorthand for the block $0 \cdots 01$). For any key k and tweak t, we have

$$L_{k,t}^{-1}(x) = \begin{cases} (2J_w \oplus I_w)(x \oplus a_k \oplus b_t) & \text{if } w \text{ is even,} \\ ((1 \oplus (2 \oplus 1)^{-1})J_w \oplus I_w)(x \oplus a_k \oplus b_t) & \text{if } w \text{ is odd.} \end{cases}$$

So the regularity of L and L^{-1} is immediate from Lemma 3.1.1.

3.2.2 Our Construction

The CTET$^+$ enciphering scheme based on an n-bit S-box S is defined as follows; for a key $\mathbf{k} = (k_0, k_1, k_2) \in \mathcal{K} \times \mathcal{K}' \times \mathcal{K}$, a tweak $t \in \mathcal{T}$ and a plaintext $x \in \mathcal{X}$ (see also Figure 3.1):

$$\text{CTET}^+[S]_{\mathbf{k}}(t, x) = T_{k_2}\left(t, S^{\|}\left(L_{k_1}\left(t, S^{\|}\left(T_{k_0}(t, x)\right)\right)\right)\right),$$

where T denotes TBPE (as defined in Section 3.1), and for $x = (x_1, \ldots, x_w)$, we write

$$S^{\|}(x) = S(x_1) \,\|\, S(x_2) \,\|\, \cdots \,\|\, S(x_w).$$

As we will see in the following section, the middle layer of our construction has to be regular, but does not need to be SBU (although it still has to satisfy a weaker universality constraint).

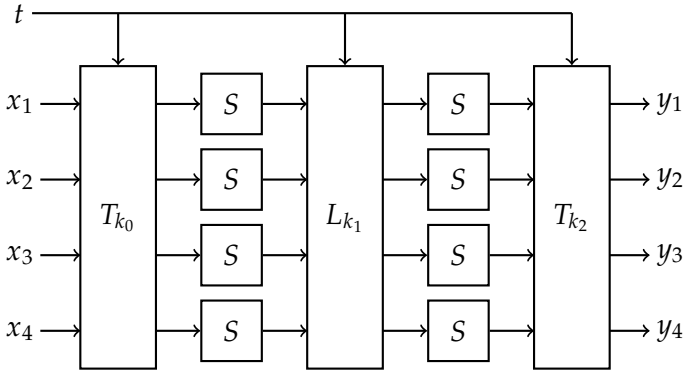

Figure 3.1: *CTET$^+$ with $w = 4$.*

3.3 Security Analysis of CTET$^+$

In this section, we study the security of our new tweakable enciphering scheme CTET$^+$ in the multi-user setting where the S-Box S is assumed to be secret.

3.3.1 Indistinguishability in the Multi-user Setting

Let $\mathsf{C}[S]$ be a keyed tweakable permutation on $\mathcal{M}$ with key space $\mathcal{K}$ and tweak space $\mathcal{T}$ using an n-bit *secret* S-box S as its inner primitive. In the multi-user setting, let ℓ denote the number of users. In the *real* world, ℓ secret keys $\mathbf{k} = (\mathbf{k}_1, \ldots \mathbf{k}_\ell) \in \mathcal{K}^\ell$ are chosen uniformly at random. An ℓ-tuple of S-boxes $\mathbf{S} = (S_1, \ldots, S_\ell)$ is also chosen uniformly at random from $\mathsf{Perm}(n)^\ell$. A distinguisher $\mathscr{D}$ is given oracle access to $\mathsf{C}_{\mathbf{k}}[\mathbf{S}] = (\mathsf{C}_{\mathbf{k}_1}[S_1], \ldots, \mathsf{C}_{\mathbf{k}_1 \ell}[S_\ell])$ in the *real world*. In contrast, in the *ideal world*, $\mathscr{D}$ is given a set of independent random tweakable permutations $\widetilde{\mathsf{P}} = (\widetilde{P}_1, \ldots, \widetilde{P}_\ell) \in \widetilde{\mathsf{Perm}}(\mathcal{T}; \mathcal{X})^\ell$ instead of $\mathsf{C}_{\mathbf{k}}[\mathbf{S}]$.

The adversarial goal is to tell apart the two worlds $(\mathsf{C}_{\mathbf{k}_1}[S_1], \ldots, \mathsf{C}_{\mathbf{k}_\ell}[S_\ell])$ and $(\widetilde{P}_1, \ldots, \widetilde{P}_\ell)$ by adaptively making forward and backward queries to each of the constructions (without access to the S-boxes). Using the notations above, we define $\mathscr{D}$'s distinguishing advantage as

$$\mathbf{Adv}^{\mathrm{mu}}_{\mathsf{C}}(\mathscr{D}) := \left| \Pr\left(\widetilde{\mathsf{P}} \twoheadleftarrow_{\$} \widetilde{\mathsf{Perm}}(\mathcal{T}; \mathcal{X})^\ell : \mathcal{D}^{\widetilde{\mathsf{P}}} = 1\right) - \Pr\left(\mathbf{k} \twoheadleftarrow_{\$} \mathcal{K}^\ell, \mathbf{S} \twoheadleftarrow_{\$} \mathsf{Perm}(n)^\ell : \mathcal{D}^{\mathsf{C}_{\mathbf{k}}[\mathbf{S}]} = 1\right) \right|$$

For $q \geq 0$ adversarial queries, we define,

$$\mathbf{Adv}^{\mathrm{mu}}_{\mathsf{C}}(q) = \max_{\mathscr{D} \in \mathbb{A}(q)} \mathbf{Adv}^{\mathrm{mu}}_{\mathsf{C}}(\mathscr{D})$$

3.3.1.1 H-Coefficient Technique

Let $\mathscr{D}$ be a distinguisher making a total q queries to the construction oracles. The queries made to the λ-th construction oracle, denoted C_λ, are recorded in a query history

$$Q_{C_\lambda} = (\lambda, t_{\lambda,i}, x_{\lambda,i}, y_{\lambda,i})_{1 \leq i \leq q_\lambda}$$

for $\lambda = 1, \ldots, \ell$, where q_λ is the number of queries made to C_λ and $(\lambda, t_{\lambda,i}, x_{\lambda,i}, y_{\lambda,i})$ represents the evaluation obtained by the i-th query to C_λ. So according to our construction, it implies either $\mathsf{C}_{\mathbf{k}_\lambda}[S_\lambda](t_{\lambda,i}, x_{\lambda,i}) = y_{\lambda,i}$ or $\widetilde{P}_\lambda(t_{\lambda,i}, x_{\lambda,i}) = y_{\lambda,i}$. Then Q_C defined by

$$Q_C = Q_{C_1} \cup \cdots \cup Q_{C_\ell}.$$

is called the transcript of $\mathscr{D}$. Since the distinguisher is deterministic and does not make any redundant query, the output of $\mathscr{D}$ can be regarded as a function of Q_C, denoted $\mathscr{D}(Q_C)$.

Fix a transcript Q_C, a key $k \in \mathcal{K}$, a tweakable permutation $\widetilde{P} \in \widetilde{\mathsf{Perm}}(\mathcal{T}; \mathcal{X})$, an S-box $S \in \mathsf{Perm}(n)$ and $\lambda \in [1; \ell]$: if $\mathsf{C}_k[S](t_{\lambda,i}, x_{\lambda,i}) = y_{\lambda,i}$ (resp. $\widetilde{\mathsf{P}}(t_{\lambda,i}, x_{\lambda,i}) = y_{\lambda,i}$) for every $i = 1, \ldots, q_\lambda$, then we will write $\mathsf{C}_k[S] \vdash Q_{C_\lambda}$ (resp. $\widetilde{P} \vdash Q_{C_\lambda}$). Similarly, let $\mathbf{k}_1, \ldots, \mathbf{k}_\ell \in \mathcal{K}$, $S_1, \ldots, S_\ell \in \mathsf{Perm}(n)$ and $\widetilde{P}_1, \ldots \widetilde{P}_\ell \in \widetilde{\mathsf{Perm}}(\mathcal{T}; \mathcal{X})$. If $\mathsf{C}_{\mathbf{k}_\lambda}[S_\lambda] \vdash Q_{C_\lambda}$ (resp. $\widetilde{P}_\lambda \vdash Q_{C_\lambda}$) for every $\lambda = 1, \ldots, \ell$, then we will write $(\mathsf{C}_{\mathbf{k}_\lambda}[S_\lambda])_{\lambda=1,\ldots,\ell} \vdash Q_C$ (resp. $(\widetilde{P}_\lambda)_{\lambda=1,\ldots,\ell} \vdash Q_C$).

For an attainable transcript Q_C, let

$$\mathsf{p}_1(Q_C) = \Pr\left(\widetilde{\mathsf{P}} \twoheadleftarrow_{\$} \widetilde{\mathsf{Perm}}(\mathcal{T}; \mathcal{X})^\ell : (\mathsf{C}_{\mathbf{k}_\lambda}[S_\lambda])_\lambda \vdash Q_C\right),$$

$$\mathsf{p}_2(Q_C) = \Pr\left(\mathbf{k} \xleftarrow{\$} \mathcal{K}^\ell, \mathbf{S} \xleftarrow{\$} \mathsf{Perm}(n)^\ell : (\widetilde{\mathsf{P}} \vdash Q_C\right).$$

In the H-coefficient technique (see Corollary 2.4.1), the lower bound in the second condition is often referred to as the *ε-point-wise proximity* of the transcript Q_C [171]. The point-wise proximity of a transcript in the multi-user setting is guaranteed by the point-wise proximity of (Q_{C_λ}) for each $\lambda = 1, \ldots, \ell$ in the single-user setting. The following lemma is a restatement of Lemma 3 in [171] in the secret permutation setting.

Lemma 3.3.1. *Let $\varepsilon : \mathbb{N} \times \mathbb{N} \to \mathbb{R}^{\geq 0}$ be a function such that $\varepsilon(y) + \varepsilon(z) \leq \varepsilon(y+z)$ for every y, $z \in \mathbb{N}$. Suppose that for any distinguisher $\mathscr{D}$ in the single-user setting that makes q construction queries, and for any attainable transcript Q_C obtained by $\mathscr{D}$, one has*

$$\mathsf{p}_2(Q_C) \geq (1 - \varepsilon(q))\mathsf{p}_1(Q_C).$$

Then for any distinguisher $\mathscr{D}$ in the multi-user setting that makes total q construction queries, and for any attainable transcript Q_C obtained by $\mathscr{D}$, one has

$$\mathsf{p}_2(Q_C) \geq (1 - \varepsilon(q))\mathsf{p}_1(Q_C).$$

3.3.2 Security Proof of CTET$^+$

Next, we establish the security of our construction, CTET$^+$, by proving the following theorem.

Theorem 3.3.1. *Let n, and w be positive integers such that $w \geq 2$ and let ℓ be the number of users. Then for any q such that $3w^2 + 16w^2q \leq 2^n$, one has*

$$\mathbf{Adv}^{\mathrm{mu}}_{CTET^+}(q) \leq \frac{(4w^3 + 31w^2)q}{2^n} + \frac{32w^4q^2 + (4w^6 + 32w^5 + 128w^4)q^3}{2^{2n}} + \frac{12w^6q^4}{2^{3n}}.$$

The proof of Theorem 3.3.1 follows from the Lemma below (with $\delta \leq \frac{2w}{2^n}$), Corollary 2.4.1 and Lemma 3.3.1. Therefore, we are left with the proof of the following Lemma.

Lemma 3.3.2. *Let q be positive integers such that $3w^2 + 16w^2q \leq 2^n$, and let $\delta = \frac{w}{2^n - w}$.*[3] *Let also $\mathcal{D}$ be a distinguisher in the single-user setting that makes q construction queries. Then for any attainable transcript $\tau = Q_C$, one has*

$$\frac{\mathsf{p}_2(Q_C)}{\mathsf{p}_1(Q_C)} \geq 1 - \frac{31w^2q}{2^n} - \frac{28w^4q^2 + 128w^4q^3}{2^{2n}} - \frac{12w^6q^4}{2^{3n}} - 2w^2q\delta - \frac{2w^3q^2\delta}{2^n} - w^4q^3\delta^2 - \frac{16w^4q^3\delta}{2^n}.$$

Outline for the Proof of Lemma 3.3.2. The proof is structured into four parts. First, we provide some preliminary setup for the proof, including the definition of an extension of a transcript and the criteria for what constitutes a bad extended transcript. Second,

[3] The security proof requires only the blockwise universality of TBPE, not the uniformity. So δ' does not appear in our bound.

we establish an upper bound on the number of bad extended transcripts in Lemma 3.3.3. Third, in the most substantial part of the proof, we show in Lemma 3.3.4 that the probability of obtaining any good extension in the real world is sufficiently close to the probability of obtaining it in the ideal world. Finally, we derive Lemma 3.3.2 from Lemma 3.3.3 and Lemma 3.3.4.

3.3.2.1 Preliminary Setup for the Proof of Lemma 3.3.2

First note that, if $\frac{2w^2q}{2^n} + \frac{16w^4q^2+64w^4q^3}{2^{2n}} > 1$, then there is nothing to prove as the r.h.s. of the inequality becomes negative. Thus we are going to focus on the case where this inequality does not hold, as this condition will allow us to prove the positivity of several terms throughout our proof. We fix a distinguisher $\mathscr{A}$ as described in Lemma 3.3.2 and fix an attainable transcript $\tau = Q_C$ obtained by $\mathscr{A}$. Let us also denote q_t the number of queries done using tweak $t \in \mathcal{T}$, and $q = \sum_{t\in\mathcal{T}} q_t$ the total number of queries.

Extension of a transcript. We will extend τ as follows. Let us choose any pair of keys $(k_0, k_2) \in \mathcal{K}^2$. Once these keys have been chosen, some construction queries will become involved in collisions. A *first-order colliding query* is a construction query $(t, x, y) \in Q_C$ such that one of the following conditions holds:

1. there exist a construction query $(t', x', y') \in Q_C$ and two integers $i, j \in [1; w]$ such that $(t, x, i) \neq (t', x', j)$ and $T_{k_0}(t, x)_i = T_{k_0}(t', x')_j$;
2. there exist a construction query $(t', x', y') \in Q_C$ and two integers $i, j \in [1; w]$ such that $(t, y, i) \neq (t', y', j)$ and $T^{-1}_{k_2}(t, y)_i = T^{-1}_{k_2}(t', y')_j$.

A first-order colliding query will be said *forward* (resp. *backward*) if it satisfies Condition 1 (resp. Condition 2) above. As we will see later, no first-order colliding query will be both backward and forward with overwhelming probability. Let us denote FColl^+ (resp. FColl^-) the set of all forward (resp. backward) first-order colliding queries and $\mathrm{FColl} = \mathrm{FColl}^+ \cup \mathrm{FColl}^-$.

We are now going to build a new set Q_S that will play the role of an extension of transcript. For each *forward* (resp. *backward*) *first-order colliding* query $(t, x, y) \in Q_C$, we will add a tuple $(T_{k_0}(t, x)_i, v')_{1\leq i\leq w}$ (resp. $(u', T^{-1}_{k_2}(t, y)_i)_{1\leq i\leq w}$) to Q_S, by lazily sampling a uniformly random (dummy) permutation. In more details, using an arbitrary ordering of the queries, for every first-order forward (resp. backward) colliding query (t, x, y) and every $i = 1, \ldots, w$, if $T_{k_0}(t, x)_i$ (resp. $T^{-1}_{k_2}(t, y)_i$) does not appear in Q_S, we draw uniformly at random a block v' (resp. u') in $\{0, 1\}^n$ that is different from the values that already appear in the second (resp. first) coordinate of a tuple from Q_S. We finally choose a key k_1.

An extended transcript τ' will then be a tuple $\tau' = (Q_C, Q_S, \mathbf{k})$ where $\mathbf{k} = (k_0, k_1, k_2)$. These added values will prove useful in the description of bad extended queries transcript. Indeed, for each first-order colliding query, we will now have complete information about the evaluation of one round of the SPN, which will allow us to define

a condition on the draw of the last key k_1. Note that the addition of a pair $(T_{k_0}(t,x)_i, v')$ or $(u', T_{k_2}^{-1}(t,y)_i)$ could create a new colliding query. Such colliding queries will be referred to as *second-order colliding* queries. As we will see later, this type of collision will only occur with negligible probability as the values u' and v' are chosen uniformly randomly in the set of authorized values. We will denote SColl the set of second-order colliding queries.

Definition of Bad Transcript Extensions. Let

$$U = \{u \in \{0,1\}^n : (u,v) \in Q_S\}, \qquad V = \{v \in \{0,1\}^n : (u,v) \in Q_S\}$$

denote the domain and range of Q_S.

Definition 3.3.1. *We say that an extended transcript τ' is* bad *if at least one of the following conditions is fulfilled:*

(C-1) $\mathrm{FColl}^- \bigcap \mathrm{FColl}^+ \neq \emptyset$;

(C-2) $\mathrm{SColl} \neq \emptyset$

(C-3) there exists $(t,x,y) \in \mathrm{FColl}^+$, $i \in [1;w]$ *such that*

$$L_{k_1}\left(t, S^{\|}\left(T_{k_0}(t,x)\right)\right)_i \in U;$$

(C-4) there exists $(t,x,y) \in \mathrm{FColl}^+$, $(t',x',y') \in Q_C$, $i,j \in [1;w]$ *such that*

$$L_{k_1}\left(t, S^{\|}\left(T_{k_0}(t,x)\right)\right)_i = T_{k_0}(t',x')_j;$$

(C-5) there exists $(t,x,y),(t',x',y') \in \mathrm{FColl}^+$, $i,j \in [1;w]$ *with* $(t,x,i) \neq (t',x',j)$ *such that*

$$L_{k_1}\left(t, S^{\|}\left(T_{k_0}(t,x)\right)\right)_i = L_{k_1}\left(t', S^{\|}\left(T_{k_0}(t',x')\right)\right)_j;$$

(C-6) there exists $(t,x,y) \in \mathrm{FColl}^-$, $i \in [1;w]$ *such that*

$$L_{k_1}^{-1}\left(t, (S^{-1})^{\|}\left(T_{k_2}^{-1}(t,y)\right)\right)_i \in V;$$

(C-7) there exists $(t,x,y) \in \mathrm{FColl}^-$, $(t',x',y') \in Q_C$, $i,j \in [1;w]$ *such that*

$$L_{k_1}^{-1}\left(t, (S^{-1})^{\|}\left(T_{k_2}^{-1}(t,y)\right)\right)_i = T_{k_2}^{-1}(t',y')_j;$$

(C-8) there exists $(t,x,y),(t',x',y') \in \mathrm{FColl}^-$, $i,j \in [1;w]$ *with* $(t,y,i) \neq (t',y',j)$ *such that*

$$L_{k_1}^{-1}(t, (S^{-1})^{\|}(T_{k_2}^{-1}(t,y)))_i = L_{k_1}^{-1}(t', (S^{-1})^{\|}(T_{k_2}^{-1}(t',y')))_j.$$

Otherwise we say that τ' is good. We denote Θ_{good}, resp. Θ_{bad} the set of good, resp. bad extended transcripts and Θ' the set of all extended transcripts.

We are also going to define a probability distribution on the set Θ' as follows. First, the keys k_0, k_2 are chosen independently and uniformly at random in $\mathcal{K}$, then the evaluation Q_S (based on collisions) is chosen uniformly at random (meaning that each possible u' and v' is chosen uniformly at random in the set of its authorized values, beginning by forward first-order colliding queries and choosing an arbitrary ordering of the queries), and finally the key k_1 is chosen uniformly at random from $\{0,1\}^n$, independently from everything else. Thus the exact probability of observing an extended transcript τ' is $1/\left(2^n|\mathcal{K}|^2(2^n)_{|Q_S|}\right)$.

3.3.2.2 Analysis of Bad Transcripts

With the setting now well defined, we begin by establishing an upper bound on the probability of obtaining a bad extended transcript under the previously defined probability distribution, by proving in the following lemma.

Lemma 3.3.3. *One has*

$$\Pr\left(\tau' \in \Theta_{\text{bad}}\right) \leq 2w^2q\delta + \frac{2w^3q^2\delta}{2^n} + w^4q^3\delta^2 + \frac{16w^4q^3\delta}{2^n}.$$

Setup for the Proof of Lemma 3.3.3. Let us fix any construction query $(t,x,y) \in Q_C$. By the blockwise universality of T,

$$\Pr_{k_0 \xleftarrow{\$} \mathcal{K}}\left((t,x,y) \in \text{FColl}^+\right) \leq w^2q\delta, \tag{3.2}$$

and similarly,

$$\Pr_{k_2 \xleftarrow{\$} \mathcal{K}}\left((t,x,y) \in \text{FColl}^-\right) \leq w^2q\delta. \tag{3.3}$$

Also let us define auxiliary events,

- $\mathsf{aux}_1 \Leftrightarrow$ there exists $(t,x,y) \in Q_C$, $(i,j) \in ([1;w])^{*2}$ such that $T_{k_0}(t,x)_i = T_{k_0}(t,x)_j$.
- $\mathsf{aux}_2 \Leftrightarrow$ there exists $(t,x,y) \in Q_C$, $(i,j) \in ([1;w])^{*2}$ such that $T^{-1}_{k_2}(t,y)_i = T^{-1}_{k_2}(t,y)_j$.
- $\mathsf{aux}_3 \Leftrightarrow$ there exists distinct $(t,x,y),(t',x',y') \in Q_C$, and $a,b,c \in [1;w]$ such that $b \neq c$, $T_{k_0}(t,x)_a = T_{k_0}(t',x')_a$, and $T_{k_0}(t,x)_b = T_{k_0}(t',x')_c$.
- $\mathsf{aux}_4 \Leftrightarrow$ there exists distinct $(t,x,y),(t,x,y) \in Q_C$, and $a,b,c \in [1;w]$ such that $b \neq c$, $T^{-1}_{k_2}(t,y)_a = T^{-1}_{k_2}(t',y')_a$, and $T^{-1}_{k_2}(t,y)_b = T^{-1}_{k_2}(t',y')_c$.

One can easily derive $\Pr(\mathsf{aux}_1) \leq w^2q\delta$ and $\Pr(\mathsf{aux}_3) \leq w^3q^2\delta 2^{-n}$. Since aux_2 and aux_4 are symmetric events of aux_1 and aux_3, respectively, we have

$$\Pr(\mathsf{aux}_1 \vee \mathsf{aux}_2 \vee \mathsf{aux}_3 \vee \mathsf{aux}_4) \leq 2w^2q\delta + 2w^3q^2\delta 2^{-n}.$$

We now upper bound the probabilities of the eight conditions in turn. We denote by Θ_i the set of attainable transcripts fulfilling condition (C-i).

Condition (C-1). By definition, one has

$$\Pr\left(\tau' \in \Theta_1\right) \leq \sum_{(t,x,y)\in Q_C} \Pr\left((t,x,y) \in \mathrm{FColl}^+ \cap \mathrm{FColl}^-\right).$$

Since the random draw of k_0 and k_2 are independent, (3.2) and (3.3) give

$$\Pr\left(\tau' \in \Theta_1\right) \leq w^4 q^3 \delta^2.$$

Condition (C-2). A second-order collision can occur in two ways:

- the completion of the information about a query $(t,x,y) \in \mathrm{FColl}^+$ triggers a collision, i.e. there exists $i \in [1;w]$ such that we added the pair $(T_{k_0}(t,x)_i, v')$ where $v' = T_{k_2}^{-1}(t',y')_j$ for some query $(t',x',y') \in Q_C$ and some $j \in [1;w]$.
- the completion of the information about a query $(t,x,y) \in \mathrm{FColl}^-$ triggers a collision, i.e. there exists $i \in [1;w]$ such that we added the pair $(u', T_{k_2}^{-1}(t,y)_i)$ where $u' = T_{k_0}(t',x')_j$ for some query $(t',x',y') \in Q_C$ and some $j \in [1;w]$.

Since the values u' and v' are randomly chosen in a set of size at least $2^n - wq$, one has

$$\Pr\left(\tau' \in \Theta_2\right) \leq \frac{2w^4q^3\delta}{2^n - wq}.$$

Condition (C-3), (C-4), (C-6), and (C-7). Let $h = h_1 \| \cdots \| h_w$, then one has,

$$\Pr\left(L_{k_1}(t,h)_i = c \wedge \neg \mathsf{aux}_1\right) \leq \frac{1}{2^n - wq}$$

since the coefficient of h_1 cannot be 0 and the number of such choices for h_1 is always larger than $2^n - wq$. Thus, by summing over the number of queries that can make each event, we get

$$\Pr\left(\tau' \in \Theta_3 \wedge \neg \mathsf{aux}_1\right) \leq \frac{w^4q^3\delta}{2^n - wq}, \qquad \Pr\left(\tau' \in \Theta_4 \wedge \neg \mathsf{aux}_1\right) \leq \frac{w^4q^3\delta}{2^n - wq}.$$

Similarly, we get

$$\Pr\left(\tau' \in \Theta_6 \wedge \neg \mathsf{aux}_3\right) \leq \frac{w^4q^3\delta}{2^n - wq}, \qquad \Pr\left(\tau' \in \Theta_7 \wedge \neg \mathsf{aux}_3\right) \leq \frac{w^4q^3\delta}{2^n - wq}.$$

Condition (C-5) and (C-8). Let us fix queries $(t,x,y),(t',x',y') \in \mathrm{FColl}^+$ and assume $\neg(\mathsf{aux}_1 \vee \mathsf{aux}_2 \vee \mathsf{aux}_3 \vee \mathsf{aux}_4)$. Let $S^{\|}(T_{k_0}(t,x)) = h = h_1 \| \cdots \| h_w$ and $S^{\|}(T_{k_0}(t',x')) = h' = h'_1 \| \cdots \| h'_w$, then Condition (C-5) holds (by given two queries) when there exists $i,j \in [1;w]$ such that $(t,x,y,i) \neq (t',x',y',j)$ and

$$2\sum_{r=1}^{w}(h_r \oplus h'_r) \oplus h_i \oplus h'_j = (z^i \oplus z^j)k_1 \oplus t \oplus t'. \tag{3.4}$$

The probability of the above equation can then be calculated as follows:

(a-1) If $i \neq j$, (3.4) holds with probability at most 2^{-n} by the randomness of k_1.

(a-2) If $i = j$ and $t \neq t'$, (3.4) holds with probability at most $1/(2^n - wq)$ since the number of choices for $S(\cdot)$ is always larger than $2^n - wq$.

(a-3) If $(t, i) = (t', j)$ and $h_i \neq h'_i$. Then (3.4) holds with probability at most $1/(2^n - wq)$ since the coefficient of h_i cannot be 0.

(a-4) If $(t, i, h_i) = (t', j, h'_j)$, we have $x \neq x'$ and $h \neq h'$ so there exists $a \in [1; w]$ such that $h_a \neq h'_a$ since otherwise we have aux_1 or aux_3. Then, h_a is unique in h and h' so for the same reason above, (3.4) holds with probability at most $1/(2^n - wq)$.

Overall, the probability that Condition (C-5) occurs is smaller than

$$\Pr\left(\mathsf{aux}_1 \vee \mathsf{aux}_3\right) + \frac{w^4 q^3 \delta}{2^n - wq}.$$

Similarly, the probability that Condition (C-8) occurs is smaller than

$$\Pr\left(\mathsf{aux}_2 \vee \mathsf{aux}_4\right) + \frac{w^4 q^3 \delta}{2^n - wq}.$$

Finally, since $wq \leq 2^{n-1}$, the proof follows by taking a union bound over all conditions. □

From Attainable Transcripts to Extended Transcripts. Observe that Lemma 3.3.2 is expressed in terms of transcripts. Hence, it will be advantageous to have a lower bound on the ratio that depends solely on good extended transcripts instead. Let us fix any extended transcript $\tau' = (Q_C, Q_S, \mathbf{k})$ and let

$$\mathsf{p}_{\mathrm{re}}(\tau') = \frac{1}{2^n |\mathcal{K}|^2} \Pr\left((S \vdash Q_S) \wedge (\mathrm{CTET}^+[S]_{\mathbf{k}} \vdash Q_C)\right),$$
$$\mathsf{p}(\tau') = \Pr\left(\mathrm{CTET}^+[S]_{\mathbf{k}} \vdash Q_C \mid S \vdash Q_S\right).$$

Note that one has

$$\mathsf{p}_2(Q_C) \geq \sum_{\tau' \in \Theta_{\mathrm{good}}} \mathsf{p}_{\mathrm{re}}(\tau') = \sum_{\tau' \in \Theta_{\mathrm{good}}} \frac{1}{2^n |\mathcal{K}|^2 (2^n)_{|Q_S|}} \mathsf{p}(\tau'),$$

and

$$\mathsf{p}_1(Q_C) = \frac{1}{\prod_{t \in \mathcal{T}} (2^{wn})_{q_t}}.$$

Thus one has

$$\frac{\mathsf{p}_2(Q_C)}{\mathsf{p}_1(Q_C)} \geq \sum_{\tau' \in \Theta_{\mathrm{good}}} \frac{\prod_{t \in \mathcal{T}} (2^{wn})_{q_t}}{2^n |\mathcal{K}|^2 (2^n)_{|Q_S|}} \mathsf{p}(\tau') \geq \Pr\left[\tau' \in \Theta_{\mathrm{good}}\right] \left(\min_{\tau' \in \Theta_{\mathrm{good}}} \mathsf{p}(\tau') \prod_{t \in \mathcal{T}} (2^{wn})_{q_t} \right), \quad (3.5)$$

where the last line comes from the fact that the exact probability to obtain an extended transcript τ is $\frac{1}{2^n |\mathcal{K}|^2 (2^n)_{|Q_S|}}$. Thus, our final step is the study of good extended transcripts, and especially how close is the term $\mathsf{p}(\tau') \prod_{t \in \mathcal{T}} (2^{wn})_{q_t}$ to 1.

3.3.2.3 Analysis of Good Extended Transcripts

The major part of our analysis will be to prove the following lemma.

Lemma 3.3.4. *For any good extended transcript τ', one has*

$$\mathsf{p}(\tau')\prod_{t\in\mathcal{T}}(2^{wn})_{q_t} \geq 1 - \frac{31w^2q}{2^n} - \frac{28w^4q^2 + 128w^4q^3}{2^{2n}} - \frac{12w^6q^4}{2^{3n}}.$$

Setup for The Proof of Lemma 3.3.4. Fix any good extended transcript $\tau' = (Q_C, Q_S, (k_0, k_1, k_2))$ and let $p = |Q_S|$. Recall that U and V denote respectively the domain and the range of Q_S, which means that $|U| = |V| = p$. We define two quantities characterizing an extended transcript τ', namely

$$\alpha_1 := \left|\{(t,x,y) \in Q_C : T_{k_0}(t,x)_i \in U \text{ for some } i \in [1;w]\}\right|,$$
$$\alpha_2 := \left|\{(t,x,y) \in Q_C : T^{-1}_{k_2}(t,y)_i \in V \text{ for some } i \in [1;w]\}\right|.$$

Put simply, α_1 (resp. α_2) is the number of queries $(t,x,y) \in Q_C$ which "collide" with a query $(u,v) \in Q_S$ in the extended transcript. This corresponds exactly to the number of queries $(t,x,y) \in Q_C$ which collide with a query $(t',x',y') \in Q_C$ at the input of S (resp. at the output of S), once the choice of (k_0,k_2) has been made. Indeed, since τ' is a good extended transcript, there are no second-order collisions. Thus $\alpha_1 = |\mathrm{FColl}^+|$ and $\alpha_2 = |\mathrm{FColl}^-|$.

Our goal is then to prove that $\mathsf{p}(\tau')$ is close enough to $1/\prod_{t\in\mathcal{T}}(2^{wn})_{q_t}$. In order to do so, we are going to successively consider queries belonging to FColl^+, FColl^- and $Q_0 = Q_C \setminus (\mathrm{FColl}^+ \cup \mathrm{FColl}^-)$. Note that, thanks to the additional information from the extended transcript, and since there are no second-order collisions in a good extended transcript, for every query $(t,x,y) \in Q_C$, it holds that

$$\forall i \in [1;w], T_{k_0}(t,x)_i \in U \Leftrightarrow \exists i \in [1;w], T_{k_0}(t,x)_i \in U.$$

Also note that these three sets of queries form a partition of Q_C:

- $Q_0 \bigcap \mathrm{FColl}^+ = \emptyset$ by definition;
- $Q_0 \bigcap \mathrm{FColl}^- = \emptyset$ by definition;
- $\mathrm{FColl}^+ \bigcap \mathrm{FColl}^- = \emptyset$ since otherwise τ' would satisfy Condition (C-1).

Thus, we define the following auxiliary events based on the above partition:

$$\mathsf{E}_+ : \mathrm{CTET}^+[S]_{\mathbf{k}} \vdash \mathrm{FColl}^+; \qquad \mathsf{E}_- : \mathrm{CTET}^+[S]_{\mathbf{k}} \vdash \mathrm{FColl}^-; \qquad \mathsf{E}_0 : \mathrm{CTET}^+[S]_{\mathbf{k}} \vdash Q_0.$$

Hence, the event $\mathrm{CTET}^+[S]_{\mathbf{k}} \vdash Q_C$ is equivalent to $\mathsf{E}_+ \wedge \mathsf{E}_- \wedge \mathsf{E}_0$. Note that, by definition for every $(t,x,y) \in \mathrm{FColl}^+$, $T_{k_0}(t,x)_i \in U$ for each $i \in [1;w]$; this means that the output of S is already fixed by Q_S. A similar reasoning can be made for E_-. Thus we have

$$\begin{aligned}\mathsf{p}(\tau') &= \Pr(\mathsf{E}_+ \wedge \mathsf{E}_- \wedge \mathsf{E}_0 \mid S \vdash Q_S) \\ &= \Pr(\mathsf{E}_+ \wedge \mathsf{E}_- \mid S \vdash Q_S) \cdot \Pr(\mathsf{E}_0 \mid \mathsf{E}_+ \wedge \mathsf{E}_- \wedge S \vdash Q_S) \qquad (3.6)\end{aligned}$$

Evaluation of $\Pr(\mathsf{E}_+ \wedge \mathsf{E}_- \mid S \vdash \mathcal{Q}_S)$**.** First note that, since we condition on the event $S \vdash \mathcal{Q}_S$, S is already fixed on p values. Second, remark that this event is actually equivalent to the following equations.

$$\begin{aligned} S\left(L_{k_1}\left(t, S^{\|}\left(T_{k_0}(t,x)\right)\right)_i\right) &= T_{k_2}^{-1}(t,y)_i \qquad \text{for every } (t,x,y,i) \in \mathrm{FColl}^+ \times [1;w], \\ S\left(T_{k_0}(t,x)_i\right) &= L_{k_1}^{-1}\left(t, (S^{-1})^{\|}\left(T_{k_2}^{-1}(t,y)\right)\right)_i \qquad \text{for every } (t,x,y,i) \in \mathrm{FColl}^- \times [1;w]. \end{aligned}$$

For the first set of equations, note that the values $L_{k_1}\left(t, S^{\|}\left(T_{k_0}(t,x)\right)\right)_i$ are pairwise distinct otherwise τ' would satisfy Condition (C-5). These values are also outside U otherwise τ' would satisfy Condition (C-3). The values $T_{k_2}^{-1}(t,y)_i$ are pairwise distinct (otherwise τ' would satisfy Condition (C-1)) and outside V since otherwise τ' would satisfy Condition (C-1) or (C-2).
Similarly, for the second set of equations,the values $T_{k_0}(t,x)_i$ are all pairwise distinct and outside U, otherwise τ' would satisfy (C-1) or (C-2). The values $L_{k_1}^{-1}\left(t, (S^{-1})^{\|}\left(T_{k_2}^{-1}(t,y)\right)\right)_i$ are also pairwise distinct and outside V or τ' would satisfy Condition (C-6) or (C-8). Moreover, the values $L_{k_1}\left(t, S^{\|}\left(T_{k_0}(t,x)\right)\right)_i$ for every $(t,x,y,i) \in \mathrm{FColl}^+ \times [1;w]$ are distinct from the values $T_{k_0}(t,x)_i$ for every $(t,x,y,i) \in \mathrm{FColl}^- \times [1;w]$ since otherwise τ' would satisfy Condition (C-4). Similarly the values $T_{k_2}^{-1}(t,y)_i$ for every $(t,x,y,i) \in \mathrm{FColl}^+ \times [1;w]$ are pairwise distinct from the values

$$L_{k_1}^{-1}\left(t, (S^{-1})^{\|}\left(T_{k_2}^{-1}(t,y)\right)\right)_i$$

for every $(t,x,y,i) \in \mathrm{FColl}^- \times [1;w]$.
Hence the event $\mathsf{E}_+ \wedge \mathsf{E}_-$ is actually equivalent to $w\alpha_1 + w\alpha_2$ new and distinct equations on S, so that

$$\Pr(\mathsf{E}_+ \wedge \mathsf{E}_- | S \vdash \mathcal{Q}_S) = \frac{1}{(2^n - p)_{w\alpha_1 + w\alpha_2}}. \tag{3.7}$$

Lower bounding $\Pr(\mathsf{E}_0 \mid \mathsf{E}_+ \wedge \mathsf{E}_- \wedge S \vdash \mathcal{Q}_S)$**.** Conditioned on $\mathsf{E}_+ \wedge \mathsf{E}_- \wedge S \vdash \mathcal{Q}_S$, S is fixed on exactly $p + w\alpha_1 + w\alpha_2$ values. Let U' be the set of values on which S is already fixed and $V' = \{S(u) : u \in U'\}$. For the sake of clarity, we denote

$$\mathcal{Q}_0 = \{(t_0, x_0, y_0), \ldots, (t_{q_0-1}, x_{q_0-1}, y_{q_0-1})\},$$

using an arbitrary ordering of the queries, and $q_0 := |\mathcal{Q}_0| = q - \alpha_1 - \alpha_2$.
First note that the values $T_{k_0}(t,x)_i$ for each $(t,x,y) \in \mathcal{Q}_0, i \in [1;w]$ are pairwise distinct by definition of $\mathcal{Q}_0$. On the other hand, we note that U' consists of two different types of values:

- values of the form $T_{k_0}(t',x')_{i'}$ that were either added in U' during the completion of a forward colliding query, or that came from a condition that was introduced by event E_-,
- values of the form $L_{k_1}\left(t', S^{\|}\left(T_{k_0}(t',x')\right)\right)_{i'}$ that were either added during the completion of a backward colliding query, or that came from a condition that was introduced by event E_+.

Therefore, the values $T_{k_0}(t,x)_i$ for each $(t,x,y) \in \mathcal{Q}_0, i \in [1;w]$ are outside U' by definition of $\mathcal{Q}_0$ and since otherwise τ' would satisfy either Conditions (C-2) or (C-4). Similarly, the values $T_{k_2}^{-1}(t,y)_i$ for each $(t,x,y) \in \mathcal{Q}_0, i \in [1;w]$ are pairwise distinct and outside V'. Let us denote

$$U'' = \{T_{k_0}(t,x)_i : (t,x,y) \in \mathcal{Q}_0, i \in [1;w]\}, \quad V'' = \{T_{k_2}^{-1}(t,y)_i : (t,x,y) \in \mathcal{Q}_0, i \in [1;w]\}.$$

Thus,

$$U' \cup U'' = \{T_{k_0}(t,x)_i : (t,x,y) \in \mathcal{Q}_C, i \in [1;w]\},$$
$$V' \cup V'' = \{T_{k_2}^{-1}(t,y)_i : (t,x,y) \in \mathcal{Q}_C, i \in [1;w]\},$$

since the conditions from the queries in FColl were integrated in U' and V'.
In order to lower bound $\Pr(\mathsf{E}_0 \mid \mathsf{E}_+ \wedge \mathsf{E}_- \wedge S \vdash \mathcal{Q}_S)$, we are going to lower bound the number of possible "intermediate" values $S\left(T_{k_0}(t,x)_i\right)$ for each $(t,x,y) \in \mathcal{Q}_0, i \in [1;w]$, such that no new collision will be created. More precisely, the following requirements must be met:

- the values $S\left(T_{k_0}(t,x)_i\right)$ for each $(t,x,y) \in \mathcal{Q}_0, i \in [1;w]$ are pairwise distinct and outside V',
- the values $L_{k_1}\left(t, S^{\|}\left(T_{k_0}(t,x)\right)\right)_i$ for each $(t,x,y) \in \mathcal{Q}_0, i \in [1;w]$ are pairwise distinct and outside U'.

Let N_0 be the number of tuples of distinct values $\left(v_{i,j} : i \in [1;q_0], j \in [1;w]\right)$ in $\{0,1\}^n \setminus V'$ such that the values $\left(L_{k_1}\left(t_i, v_{i,1}\| \ldots \| v_{i,w}\right)\right)_i$ are pairwise distinct and outside U'. Simply lower bounding N_0 will not be sufficient to achieve an accurate lower bound on $\Pr(\mathsf{E}_0 \mid \mathsf{E}_+ \wedge \mathsf{E}_- \wedge S \vdash \mathcal{Q}_S)$. Indeed, if we choose one of these N_0 tuples and condition on S satisfying $S\left(T_{k_0}(t_i,x_i)_j\right) = v_{i,j}$ for each $(i,j) \in [1;q_0] \times [1;w]$, the event E_0 will then be equivalent to a number of new equations on S that will depend on the number of collisions between values $\left(L_{k_1}\left(t_i, v_{i,1}\| \ldots \| v_{i,w}\right)\right)_j$ and $T_{k_0}(t_{i'}, x_{i'})_{j'}$. Thus we will have to be mindful of the occurrence of such collisions. The simplest way to do so would be to only consider the tuples of values that do not create such collisions. However, such a strategy could only lead to a security bound up to the birthday bound. Instead, we are going to fix in advance a small number of collisions and then lower bound the number of tuples that will exactly satisfy these collisions. The result will follow by summing over every possible choice of collisions.

Let $\theta \in [0; \lfloor q_0/2 \rfloor]$. We are going to choose θ pairs of queries that are not involved in a first-order collision, and to force a collision of the type $S(T_{k_0}(t,x)_i) = T_{k_2}^{-1}(t',y')_{i'}$ for each pair of queries. In order to simplify the computations, no query should appear in more than one pair. Note that, in that case, there are exactly $w^{2\theta}(q_0)_{2\theta}/\theta!$ possible sets of θ pairs of the form $((t,x,y,i),(t',x',y',i'))$ where $(t,x,y),(t',x',y') \in \mathcal{Q}_0$ and $i,j \in [1;w]$. Let us fix one of these sets A. We are going to lower bound the number N_θ of possible intermediate values $(v_{i,1},\ldots,v_{i,w})_{i\in[q_0]} \in (\{0,1\}^n)^{wq_0}$ such that:

(D-1) these values are pairwise distinct and outside V';

(D-2) the values $L_{k_1}(t_i, v_{i,1}\| \ldots \|v_{i,w})_j$ for $i \in [1; q_0]$, $j \in [1; w]$ are pairwise distinct and outside U';

(D-3) for every pair of queries $((t_i, x_i, y_i, j), (t_{i'}, x_{i'}, y_{i'}, j')) \in A$, $v_{i,j} = T_{k_2}^{-1}(t_{i'}, y_{i'})_{j'}$ and $L_{k_1}(t_{i'}, v_{i',1}\| \ldots \|v_{i',w})_{j'} = T_{k_0}(t_i, x_i)_j$;

(D-4) every other value $v_{i,j}$ for $i \in [q_0]$, $j \in [1; w]$ should be outside V'' and every other value $L_{k_1}(t_i, v_{i,1}\| \ldots \|v_{i,w})_j$ for $i \in [1; q_0]$, $j \in [1; w]$ should be outside U''.

To finalize our analysis of good extended transcripts, a technical lemma is needed. Namely, we will need a lower bound on the quantity N_θ.

Lemma 3.3.5. *One has*

$$N_\theta \geq (2^n - p - wq)_{wq_0-2\theta} \prod_{i=0}^{q_0-2\theta-1} \left(1 - \frac{w(p + wq + wi) + w(w-1)/2}{2^n - p - wq - w(i+1) + 1}\right) \times \prod_{j=0}^{\theta-1} \left(1 - \frac{2wp + 4w^2q + 4jw^2 + (2w+1)(w-1)}{2^n - p - 4wq - 2w}\right).$$

First we complete the proof of Lemma 3.3.4 assuming Lemma 3.3.5. The proof of Lemma 3.3.5 will be given at the end of the chapter.

3.3.2.4 Deriving Lemma 3.3.4 from Lemma 3.3.5.

Note that, if we fix such intermediate values, the probability that E_0 is satisfied along with the equations $S\left(T_{k_0}(t_i, x_i)_j\right) = v_{i,j}$ for each $(i, j) \in [1; q_0] \times [1; w]$ is exactly $1/(2^n - p - w\alpha_1 - w\alpha_2)_{2wq_0-\theta}$. Indeed, if $\theta = 0$, then no collision occurs and each query in Q_0 adds $2w$ new conditions on S. If $\theta > 0$, then, for each one of the θ pairs of queries in A, the condition $S(T_{k_0}(t, x)_i) = T_{k_2}^{-1}(t', y')_{i'}$ appears twice, thus adding only $4w - 1$ new conditions for these two queries. Hence

$$\Pr\left(\mathsf{E}_0 | \mathsf{E}_+ \wedge \mathsf{E}_- \wedge S \vdash Q_S\right) \geq \sum_{\theta=0}^{\lfloor q_0/2 \rfloor} \frac{w^{2\theta}(q_0)_{2\theta} N_\theta}{\theta!(2^n - p - w\alpha_1 - w\alpha_2)_{2wq_0-\theta}}. \tag{3.8}$$

Combining Equation (3.8) with Equations (3.6) and (3.7) yields

$$\mathsf{p}(\tau') \geq \sum_{\theta=0}^{\lfloor q_0/2 \rfloor} \frac{w^{2\theta}(q_0)_{2\theta} N_\theta}{\theta!(2^n - p)_{wq+wq_0-\theta}}$$

since $q_0 + \alpha_1 + \alpha_2 = q$.

The final part of the proof involves numerous calculations, although they are relatively straightforward. We appeal to a trick used by Chen et al. [79]. For any $0 \leq \theta \leq wq_0$, recall that the probability mass function of $\mathrm{Hyp}_{2^n-p-w\alpha_1-w\alpha_2, wq_0, wq_0}$ is given by

$$\mathrm{Hyp}_{2^n-p-w\alpha_1-w\alpha_2, wq_0, wq_0}(\theta) = \frac{(wq_0)_\theta (wq_0)_\theta (2^n - p - wq)_{wq_0-\theta}}{\theta!(2^n - p - w\alpha_1 - w\alpha_2)_{wq_0}}.$$

Hence, one has

$$\begin{aligned}
\mathsf{p}(\tau')\prod_{t\in\mathcal{T}}(2^{wn})_{q_t} &\geq \sum_{\theta=0}^{\lfloor q_0/2\rfloor}\frac{w^{2\theta}(q_0)_{2\theta}N_\theta\prod_{t\in\mathcal{T}}(2^{wn})_{q_t}}{\theta!(2^n-p)_{wq+wq_0-\theta}}\\
&\geq \sum_{\theta=0}^{\lfloor q_0/2\rfloor}\frac{w^{2\theta}(q_0)_{2\theta}N_\theta(2^{wn})_q}{\theta!(2^n-p)_{wq+wq_0-\theta}}\\
&\geq \sum_{\theta=0}^{\lfloor q_0/2\rfloor}\mathrm{Hyp}_{2^n-p-w\alpha_1-w\alpha_2,wq_0,wq_0}(\theta)A_\theta B_\theta \qquad (3.9)
\end{aligned}$$

where

$$\begin{aligned}
A_\theta &= \frac{w^{2\theta}(q_0)_{2\theta}}{(wq_0)_\theta(wq_0)_\theta},\\
B_\theta &= \frac{N_\theta(2^n-p-w\alpha_1-w\alpha_2)_{wq_0}(2^{wn})_q}{(2^n-p)_{wq+wq_0-\theta}(2^n-p-wq)_{wq_0-\theta}}.
\end{aligned}$$

We will lower bound A_θ and B_θ in turn. Since $\theta \leq q_0/2$ and $w > 1$, one has

$$\begin{aligned}
A_\theta &= \prod_{i=0}^{\theta-1}\frac{w^2(q_0-i)(q_0-\theta-i)}{(wq_0-i)^2}\\
&= \prod_{i=0}^{\theta-1}\left(1-\frac{q_0(w^2\theta+2w^2i-2iw)-i(w^2\theta+w^2i-i)}{(wq_0-i)^2}\right)\\
&\geq 1-\sum_{i=0}^{\theta-1}\frac{q_0(w^2\theta+2w^2i-2iw)-i(w^2\theta+w^2i-i)}{(wq_0-i)^2}\\
&\geq 1-\sum_{i=0}^{\theta-1}\frac{3w^2q_0\theta}{(wq_0-i)^2} \geq 1-\sum_{i=0}^{\theta-1}\frac{12\theta}{q_0} = 1-\frac{12\theta^2}{q_0}. \qquad (3.10)
\end{aligned}$$

Using Lemma 3.3.5, one has

$$\begin{aligned}
B_\theta &= \frac{N_\theta(2^n-p-w\alpha_1-w\alpha_2)_{wq_0}(2^{wn})_q}{(2^n-p)_{wq+wq_0-\theta}(2^n-p-wq)_{wq_0-\theta}}\\
&\geq \frac{N_\theta 2^{wnq}}{(2^n-p)_{w\alpha_1+w\alpha_2}\left((2^n-p-wq)_{wq_0-\theta}\right)^2}\left(1-\frac{q^2}{2^{wn+1}}\right)\\
&\geq \frac{N_\theta 2^{wnq_0}}{(2^n-p-wq)_{wq_0-\theta}(2^n-p-wq)_{wq_0-\theta}}\left(1-\frac{q^2}{2^{wn+1}}\right)\\
&\geq \frac{N_\theta 2^{n(wq_0-2\theta)}}{(2^n-p-wq)_{wq_0-2\theta}(2^n-p-wq)_{wq_0-2\theta}}\left(1-\frac{q^2}{2^{wn+1}}\right)\\
&\geq \frac{2^{n(wq_0-2\theta)}}{(2^n-p-wq)_{wq_0-2\theta}}\prod_{i=0}^{q_0-2\theta-1}\left(1-\frac{w(p+wq_+wi)+w(w-1)/2}{2^n-p-wq-w(i+1)+1}\right)\\
&\times\left(1-\frac{q^2}{2^{wn+1}}\right)\prod_{j=0}^{\theta-1}\left(1-\frac{2wp+4w^2q+4jw^2+(2w+1)(w-1)}{2^n-p-4wq-2w}\right). \qquad (3.11)
\end{aligned}$$

Finally, one has

$$\begin{aligned}
&\frac{2^{n(wq_0-2\theta)}}{(2^n-p-wq)_{wq_0-2\theta}}\prod_{i=0}^{q_0-2\theta-1}\left(1-\frac{w(p+wq+wi)+w(w-1)/2}{2^n-p-wq-w(i+1)+1}\right)\\
&\geq \frac{2^{wn(q_0-2\theta)}}{(2^n-p-wq)_{wq_0-2w\theta}}\prod_{i=0}^{q_0-2\theta-1}\left(1-\frac{w(p+wq+wi)+w(w-1)/2}{2^n-p-wq-w(i+1)+1}\right)\\
&\geq \prod_{i=0}^{q_0-2\theta-1}\left(1-\frac{w(p+wq+wi)+w(w-1)/2}{2^n-p-wq-w(i+1)+1}\right)\left(1+\frac{p+wq+wi}{2^n-p-wq-wi}\right)^w\\
&\geq \prod_{i=0}^{q_0-2\theta-1}\left(1-\frac{w(p+wq+wi)+w(w-1)/2}{2^n-p-wq-w(i+1)+1}\right)\left(1+\frac{wp+w^2q+w^2i}{2^n-p-wq-wi}\right)\\
&\geq \prod_{i=0}^{q_0-2\theta-1}\left(1-\frac{\frac{w^2}{2}(2^n+(w+1)p+(w+w^2)(q+i))+(wp+w^2q+w^2i)^2}{(2^n-p-wq-w(i+1)+1)(2^n-p-wq-wi)}\right)\\
&\geq 1-\frac{2w^2q}{2^n}-\frac{4w^3q\times 4wq+4q(4w^2q)^2}{2^{2n}}.
\end{aligned}\tag{3.12}$$

since $p \leq 2wq$ by definition and we assumed $3w^2 + 16w^2q \leq 2^n$. Hence, by combining (3.10), (3.11) and (3.12), one has

$$\begin{aligned}
A_\theta B_\theta \geq &\left(1-\frac{12\theta^2}{q_0}\right)\left(1-\frac{q^2}{2^{wn+1}}\right)\\
&\times\prod_{j=0}^{\theta-1}\left(1-\frac{2wp+4w^2q+4jw^2+(2w+1)(w-1)}{2^n-p-4wq-2w}\right)\\
&\times\left(1-\frac{2w^2q}{2^n}-\frac{16w^4q^2+64w^4q^3}{2^{2n}}\right).
\end{aligned}\tag{3.13}$$

Thanks to our assumptions on q, each term in the previous product, except the first one, is between 0 and 1. Since both A_θ and B_θ are positive, the previous bound holds whether $q_0 \leq 12\theta^2$ or not. Combining (3.9) and (3.13) with Weierstrass product inequality applied, and using $p \leq 2wq$, one has

$$\begin{aligned}
\mathsf{p}(\tau')\prod_{t\in\mathcal{T}}(2^{wn})_{q_t} &\geq \sum_{\theta=0}^{\lfloor q_0/2\rfloor}\mathrm{Hyp}_{2^n-p-w\alpha_1-w\alpha_2,wq_0,wq_0}(\theta)(1-f(\theta))\\
&= \sum_{\theta=0}^{\lfloor q_0/2\rfloor}\mathrm{Hyp}_{2^n-p-w\alpha_1-w\alpha_2,wq_0,wq_0}(\theta)-\mathbb{E}\left(f(\theta)\right)\\
&\geq 1-\sum_{\theta>\lfloor q_0/2\rfloor}\mathrm{Hyp}_{2^n-p-w\alpha_1-w\alpha_2,wq_0,wq_0}(\theta)-\mathbb{E}\left(f(\theta)\right),
\end{aligned}\tag{3.14}$$

where

$$\begin{aligned}
f(\theta) = &\frac{12\theta^2}{q_0}+\frac{q^2}{2^{wn+1}}+\frac{2w\theta p+4w^2\theta q+\theta^2(2w^2+1)+2(2w+1)(w-1)\theta}{2^n}\\
&+\frac{2w^2q}{2^n}+\frac{16w^4q^2+64w^4q^3}{2^{2n}}
\end{aligned}$$

and the expectation $\mathbb{E}\left(f(\theta)\right)$ is taken over the random variable θ which follows the probability distribution $\mathrm{Hyp}_{2^n-p-w\alpha_1-w\alpha_2,wq_0,wq_0}$. Note that

$$\mathbb{E}\left(f(\theta)\right) = \frac{w^2q_0^2}{2^n-p-w\alpha_1-w\alpha_2} \leq \frac{2w^2q_0^2}{2^n},$$
$$\mathbb{E}[\theta^2] \leq \frac{4w^4q_0^4}{2^{2n}} + \frac{2w^2q_0^2}{2^n}.$$

Hence, using Markov's inequality, one has

$$\sum_{\theta>q_0/2} \mathrm{Hyp}_{2^n-p-w\alpha_1-w\alpha_2,wq_0,wq_0}(\theta) \leq \frac{2\mathbb{E}[\theta]}{q_0} \leq \frac{4w^2q_0}{2^n} \leq \frac{4w^2q}{2^n}. \tag{3.15}$$

Moreover, one has

$$\begin{aligned}\mathbb{E}[f(\theta)] &\leq \frac{26w^2q}{2^n} + \frac{q^2}{2^{wn+1}} + \frac{20w^4q^2+128w^4q^3}{2^{2n}} \\ &\quad + \frac{2q^2w^2\left(2(2w+1)(w-1)+1\right)}{2^{2n}} + \frac{12w^6q^4}{2^{3n}} \\ &\leq \frac{27w^2q}{2^n} + \frac{28w^4q^2+128w^4q^3}{2^{2n}} + \frac{12w^6q^4}{2^{3n}}. \end{aligned} \tag{3.16}$$

Combining (3.14), (3.15) and (3.16) yields the desired result. □

3.3.2.5 Proof of Lemma 3.3.5

Let us fix one sets of pairs of queries A. Recall that we are going to lower bound the number N_θ of possible intermediate values $(v_{i,1},\ldots,v_{i,w})_{i\in[1;q_0]} \in \{0,1\}^n)^{wq_0}$ that satisfy conditions (D-1) to (D-4). In order to do so, we are going to rely on the regularity of L. We are first going to reorder the queries from $\boldsymbol{Q}_0$ so that the queries appearing in A are last, and both queries of a pair are consecutive. We are going to lower bound the number of possible intermediate values for these queries iteratively as follows.

The $q_0-2\theta$ single queries. These queries are not involved in any collision. Let $i \in [0; q_0-2\theta-1]$ and let us assume that $v_{j,1},\ldots,v_{j,w}$ for $0 \leq j < i$ are already chosen according to the conditions (D-1) to (D-4). The values $v_{i,1},\ldots,v_{i,w}$ should be pairwise distinct, outside of $V' \cup V''$ [4], and distinct from $v_{j,1},\ldots,v_{j,w}$ for $j < i$ [5]. This excludes a set of values of size exactly $p+wq+wi$. Similarly, the values $L_{k_1}\left(t_i, v_{i,1}\|\ldots\|v_{i,w}\right)_j$ for $j \in [1;w]$ should be pairwise distinct, outside of $U' \cup U''$ and different from the values $L_{k_1}\left(t_j, v_{j,1}\|\ldots\|v_{j,w}\right)_{j'}$ for $j < i$, $j' \in [1;w]$. This excludes a set of exactly $p+wq+wi$ values.

Using the regularity of L, the number of possibilities for $v_{i,1},\ldots,v_{i,w}$ is greater than

$$(2^n-p-wq-wi)_w\left(1-\frac{w(p+wq+wi)+w(w-1)/2}{2^n-p-wq-w(i+1)+1}\right).$$

[4] Note that $|U' \cup U''| = |V' \cup V''| = p+w\alpha_1+w\alpha_2+wq_0 = p+wq$ since τ' is a good extended transcript.
[5] These values are pairwise distinct and outside of $V' \cup V''$ by construction.

Overall, the number of possible intermediate values for the first $q_0 - 2\theta$ intermediate values is greater than

$$(2^n - p - wq)_{w(q_0-2\theta)} \prod_{i=0}^{q_0-2\theta-1} \left(1 - \frac{w(p + wq + wi) + w(w-1)/2}{2^n - p - wq - w(i+1) + 1}\right). \tag{3.17}$$

where this value is non-negative since we assume $3w^2 + 16w^2q \leq 2^n$.
Let us fix one of these sequences of intermediate values and define the sets

$$U''' = \{L_{k_1}(t_i, v_{i,1} \| \ldots \| v_{i,w})_j : i \in [0; q_0 - 2\theta - 1], j \in [1; w]\},$$
$$V'' = \{v_{i,j} : i \in [0; q_0 - 2\theta - 1], j \in [1; w]\}.$$

By construction, $U''' \cap (U' \cup U'') = \emptyset$, $V''' \cap (V' \cup V'') = \emptyset$, and $|U'''| = |V'''| = w(q_0 - 2\theta)$.
We now have to handle the remaining 2θ queries. Let

$$Q'_0 = \{(t_0, x_0, y_0), \ldots, (t_{2\theta-1}, x_{2\theta-1}, y_{2\theta-1})\}$$

be the set of the remaining queries appearing in A.

The last θ pairs of queries. Let $((t_0, x_0, y_0, i_0), (t_1, x_1, y_1, i_1))$ be the first pair of queries. Let us consider the first query. We want to fix $v_{0,i_0} = T_{k_2}^{-1}(t_1, y_1)_{i_1}$. Moreover, we want $v_{0,i}$ for $i \neq i_0$ to be pairwise distinct and outside of $V' \cup V'' \cup V'''$, and the values $L_{k_1}(t_0, v_{0,1} \| \ldots \| v_{0,w})_j$ for $j \in [1; w]$ to be pairwise distinct and outside of $U' \cup U'' \cup U'''$. Using the regularity of L, there are at least

$$\begin{aligned}(2^n - p - w(q + q_0 - 2\theta))_{w-1} \left(1 - \frac{w(p + w(q + q_0 - 2\theta)) + w(w-1)/2}{2^n - p - w(q + q_0 - 2\theta) - w + 2}\right) \\ \geq (2^n - p - w(q + q_0 - 2\theta))_{w-1} \left(1 - \frac{wp + 2w^2q + w(w-1)/2}{2^n - p - 2wq - 2w}\right)\end{aligned}$$

possibilities for $v_{0,1}, \ldots, v_{0,w}$.
Similarly, for the second query, we want $v_{1,i}$ for $i \in [1; w]$ to be pairwise distinct, outside of $V' \cup V'' \cup V'''$ and different from the values $v_{0,i}$ for $i \neq i_0$, which excludes exactly $p' + w(q + q_0 - 2\theta) + w - 1$ values. Note that the value v_{0,i_0} is automatically excluded since it appears in V''. We also want the values $L_{k_1}(t_1, v_{1,1} \| \ldots \| v_{1,w})_i$ for $i \neq i_1$ to be pairwise distinct, outside of $U' \cup U'' \cup U'''$ and different from the values $L_{k_1}(t_0, v_{0,1} \| \ldots \| v_{0,w})_j$ for $j \in [1; w]$. This excludes exactly $p + w(q + q_0 - 2\theta) + w$ values. Finally, we fix $L_{k_1}(t_1, v_{1,1} \| \ldots \| v_{1,w})_{i_1} = T_{k_0}(t_0, y_0)_{i_0}$. Using the regularity of L, the number of possibilities for $v_{1,1}, \ldots, v_{1,w}$ is lower bounded by

$$\begin{aligned}&(2^n - p - w(q + q_0 - 2\theta) - w)_{w-1} \\ &\times \left(1 - \frac{w(p + w(q + q_0 - 2\theta) + w - 1) + w(w-1)/2}{2^n - p - w(q + q_0 - 2\theta) - 2w + 2}\right) \\ &\geq (2^n - p - w(q + q_0 - 2\theta) - w + 1)_{w-1} \left(1 - \frac{w-1}{2^n - p - w(q + q_0 - 2\theta + 2)}\right)\end{aligned}$$

$$\times \left(1 - \frac{w(p + w(q + q_0 - 2\theta)) + 3w(w-1)/2}{2^n - p - w(q + q_0 - 2\theta) - 2w + 2}\right)$$
$$\geq (2^n - p - w(q + q_0 - 2\theta) - w + 1)_{w-1} \left(1 - \frac{wp + 2w^2q + (3w+2)(w-1)/2}{2^n - p - 2wq - 2w}\right).$$

Overall, the number of possible intermediate values for this pair of queries is at least

$$(2^n - p - w(q + q_0 - 2\theta))_{2w-2} \left(1 - \frac{2wp + 4w^2q + (2w+1)(w-1)}{2^n - p - 2wq - 2w}\right) \geq 0.$$

Let $j \in [1; \theta - 1]$, and let

$$((t_{2j}, x_{2j}, y_{2j}, i_{2j}), (t_{2j+1}, x_{2j+1}, y_{2j+1}, i_{2j+1}))$$

be the $(j+1)$-th pair of queries. Let us consider the first query. We want to fix $v_{2j,i_{2j}} = T_{k_2}^{-1}(t_{2j+1}, y_{2j+1})_{i_{2j+1}}$. Moreover, we want $v_{2j,j'}$ for $j' \neq i_{2j}$ to be pairwise distinct and outside of $V' \cup V'' \cup V'''$, and distinct from the $j(2w-1)$ values $v_{2j',j''}$ and $v_{2j'+1,j'''}$ for $j' < j$, $j'' \neq i_{2j'}$ and $j''' \in [1; w]$. Similarly, we want the values $L_{k_1}\left(t_{2j}, v_{2j,1} \| \ldots \| v_{2j,w}\right)_{j'}$ for $j' \in [1; w]$ to be pairwise distinct and outside of $U' \cup U'' \cup U'''$, and distinct from the $j(2w-1)$ values that were previously fixed for the j previous pairs of queries. Using the regularity of L, the number of possibilities for $v_{2j,1}, \ldots, v_{2j,w}$ is lower bounded by

$$(2^n - p - w(q + q_0 - 2\theta) - j(2w-1))_{w-1}$$
$$\times \left(1 - \frac{w(p + w(q + q_0 - 2\theta) + j(2w-1)) + w(w-1)/2}{2^n - p - w(q + q_0 - 2\theta) - j(2w-1) - w + 2}\right)$$
$$\geq (2^n - p - w(q + q_0 - 2\theta) - j(2w-1))_{w-1}$$
$$\times \left(1 - \frac{wp + 2w^2q + jw(2w-1)) + w(w-1)/2}{2^n - p - 2wq - j(2w-1) - 2w}\right).$$

Similarly, for the second query, we want $v_{2j+1,j'}$ for $j' \in [1; w]$ to be pairwise distinct, outside of $V' \cup V'' \cup V'''$ and different from the values $v_{j',j''}$ for $j' \leq 2j$ and $j'' \neq i_{j'}$ if j' is even, which excludes exactly $p + w(q + q_0 - 2\theta) + j(2w-1) + w - 1$ values. Note that the values $v_{2j',i_{2j'}}$ are automatically excluded since they appear in V''. We also want the values $L_{k_1}\left(t_{2j+1}, v_{2j+1,1} \| \ldots \| v_{2j+1,w}\right)_{j'}$ for $j' \neq i_{2j+1}$ to be pairwise distinct, outside of $U' \cup U'' \cup U'''$ and different from the values $L_{k_1}\left(t_{j'}, v_{j',1} \| \ldots \| v_{j',w}\right)_{j''}$ for $j' \leq 2j$, and $j'' \neq i_{j'}$ if j' is odd. This excludes exactly $p + w(q + q_0 - 2\theta) + j(2w-1) + w$ values. Finally, we fix $L_{k_1}\left(t_{2j+1}, v_{2j+1,1} \| \ldots \| v_{2j+1,w}\right)_{i_{2j+1}} = T_{k_0}(t_{2j}, y_{2j})_{i_{2j}}$. Using the regularity of L, the number of possibilities for $v_{2j+1,1}, \ldots, v_{2j+1,w}$ is lower bounded by

$$(2^n - p - w(q + q_0 - 2\theta) - j(2w-1) - w)_{w-1}$$
$$\times \left(1 - \frac{w(p + w(q + q_0 - 2\theta) + j(2w-1) + w - 1) + w(w-1)/2}{2^n - p - w(q + q_0 - 2\theta) - j(2w-1) - 2w + 2}\right)$$
$$\geq (2^n - p - w(q + q_0 - 2\theta) - j(2w-1) - w + 1)_{w-1}$$
$$\times \left(1 - \frac{w-1}{2^n - p - w(q + q_0 - 2\theta + 2w) - j(2w-1)}\right)$$

$$\times \left(1 - \frac{w(p + w(q + q_0 - 2\theta) + j(2w-1)) + 3w(w-1)/2}{2^n - p - w(q + q_0 - 2\theta + 2) - j(2w-1)}\right)$$
$$\geq (2^n - p - w(q + q_0 - 2\theta) - j(2w-1) - w + 1)_{w-1}$$
$$\times \left(1 - \frac{wp + 2w^2q + jw(2w-1) + (3w+2)(w-1)/2}{2^n - p - 2wq - j(2w-1) - 2w}\right).$$

Overall, the number of possible intermediate values for this pair of queries is at least

$$(2^n - p - w(q + q_0 - 2\theta) - j(2w-1))_{2w-2}$$
$$\times \left(1 - \frac{2wp + 4w^2q + 4jw^2 + (2w+1)(w-1)}{2^n - p - 2wq - j(2w-1) - 2w}\right)$$
$$\geq (2^n - p - w(q + q_0 - 2\theta) - j(2w-2))_{2w-2}$$
$$\times \left(1 - \frac{2wp + 4w^2q + 4jw^2 + (2w+1)(w-1)}{2^n - p - 4wq - 2w}\right) \geq 0.$$

Hence, the number of possible intermediate values for the last θ pairs of queries is lower bounded by

$$(2^n - p - w(q + q_0 - 2\theta))_{\theta(2w-2)}$$
$$\times \prod_{j=0}^{\theta-1} \left(1 - \frac{2wp + 4w^2q + 4jw^2 + (2w+1)(w-1)}{2^n - p - 4wq - 2w}\right) \geq 0. \tag{3.18}$$

Combining (3.17) and (3.18) yields the result. □

4

Tweakable Even-Mansour with Linear Tweak and Key Mixing

Traditional designs of TBCs often struggle to balance efficiency with security, particularly when constructed atop block ciphers. To address this, alternative approaches, such as using public random permutations, have been explored. Notably, the introduction of the TWEAKEY framework by Jean et al. [188], marked a significant advancement in TBC design. This framework allows for flexible tweak and key sizes and integrates a "tweakey" into the cipher's internal state at each round, enabling the construction of highly secure and efficient TBCs.

Further developments have focused on specialized constructions within the TWEAKEY framework, such as the STK construction, which optimizes performance while maintaining strong security properties. However, challenges remain in extending these designs to scenarios with large tweak or key sizes, particularly in ensuring their security through rigorous theoretical analysis. This chapter addresses these challenges by exploring the TEML construction for various rounds and tweak sizes, establishing the necessary conditions for achieving robust security in both indistinguishability and sequential indifferentiability settings. Our contributions include proving the security of TEML constructions under certain conditions and showing the sequential indifferentiability of Even-Mansour, which are crucial for defending against chosen key attacks. See Section 1.6.1, for a complete overview of the subject.

4.1 TEML: TEM with Linear Tweak-Key Mixing

Throughout, we fix $r \in \mathbb{N}$ as the number of rounds. In addition, we set $\eta = \alpha n$ as the tweak size, $\kappa = \beta n$ as the key size, and define $\theta := \alpha + \beta$. Moreover, let $N = 2^n$. Recall that the TEM construction is defined as:

$$\mathtt{TEM}_{k,t}^{\gamma,\mathbf{P}}(x) = P_r\left(P_{r-1}\left(\cdots P_1\left(x \oplus \gamma_0(k,t)\right)\cdots\right) \oplus \gamma_{r-1}(k,t)\right) \oplus \gamma_r(k,t). \tag{4.1}$$

where $\mathbf{P} = (P_1, \ldots, P_r) \in \mathsf{Perm}(n)^r$ is a tuples of r permutations and $\gamma = (\gamma_0, \ldots, \gamma_r)$ is a tuple of $r+1$ functions defined over $\{0,1\}^{\theta n} \to \{0,1\}^n$.

4.1.1 The TEML Construction

In [91], Cogliati and Seurin provided the first result on Tweakable Even-Mansour with linear tweak and key mixing (TEM with $\gamma_i(k,t) = k \oplus t$ for any $i \in [0;r]$), henceforth referred as TEML. They proved beyond-the-birthday bound security for a 4-round TEM with $2n$-bit key and n-bit tweak., i.e., $\alpha = 1$ and $\beta = 2$, in (4.1). The main goal of this chapter is to generalize TEML for all $r \geq 1$ and $\alpha \geq 1$, i.e., we also consider tweaks larger than n bits. This is particularly the case for several TBCs based on the TWEAKEY framework. For instance, Skinny-128-384 can be used with 128-bit block and key size and 256-bit tweak size.

Bijective Tweakey Schedules. When γ is linear, then there exists a tuple of linear functions $\lambda = (\lambda_0, \ldots, \lambda_r)$ and $\delta = (\delta_0, \ldots, \delta_r)$, such that for all $i \in [0;r]$

$$\gamma_i(k,t) = \lambda_i(k) \oplus \delta_i(t) \tag{4.2}$$

In other words, we can always view the key and tweak scheduling as separate linear functions, whenever the tweakey schedule is linear. We refer to λ and δ as the key and tweak schedule corresponding the tweakey schedule γ, respectively.

Ideally, one would aim to minimize the number of rounds on account of a larger tweak, to obtain similar security bounds as in the case of $\alpha = 1$. From Equation (4.1), it is clear that an r-round TEM construction uses $r+1$ round tweakeys. So, r must be at least $\alpha - 1$, otherwise, it is easy to see that the adversary can choose two distinct tweaks t and t', such that $\delta_i(t) = \delta_i(t')$ for all $i \in [0;r]$, resulting in a simple collision distinguisher. The case where $\alpha - 1 \leq r \leq \alpha$ does not fare well either. Specifically, the adversary can always choose distinct tweaks and block input pairs (t,x) and (t',x') such that $t \neq t'$, $\delta_0(t) \oplus x = \delta_0(t') \oplus x'$, and $\delta_i(t) = \delta_i(t')$ for all $i \in [1; r-1]$ (since $r - 1 \leq \alpha$). Clearly, the XOR of the outputs corresponding to (t,x) and (t',x') equals $\delta_r(t) \oplus \delta_r(t')$.

The above discussion clearly shows that $r = \alpha + 1$ rounds are necessary to securely absorb an η-bit tweak using a linear tweak schedule. However, just having $r > \alpha$ rounds is not sufficient for security. Indeed, one can come up with some pathological linear tweak(ey) schedule that makes the resulting construction completely insecure. For instance, assume $\alpha = 2$, and let $\delta_i(t_1,t_2) = t_1$ for all $i \in [0; r-1]$ and $\delta_r(t_1,t_2) = t_2$. This tweak schedule is obviously insecure irrespective of the number of rounds. So, some care has to be taken while deciding on a tweak(ey) schedule. In fact, similar concerns were raised in the core discussion behind the rationale of the STK construction in [186]. Indeed, their main observation requires a one-to-one relation between the input tweakey (k,t) and any θ-subset of the $(r+1)$ round tweakeys. Formally, we introduce the following definitions.

Definition 4.1.1 (Strong s-bijectivity)**.** *Let $s \in \mathbb{N}$. A strong s-bijective schedule $\gamma := (\gamma_0, \ldots, \gamma_r)$ is a tuple of $r \geq s$ linear functions $\gamma_i : \{0,1\}^{sn} \to \{0,1\}^n$ such that for any s-subtuple, $\gamma' = (\gamma_{i_1}, \ldots, \gamma_{i_s})$ of γ, the mapping*

$$(k,t) \mapsto (\gamma_{i_1}(k,t), \ldots, \gamma_{i_s}(k,t))$$

is a bijection.

Definition 4.1.2 (Weak s-bijectivity). *Let $s \in \mathbb{N}$. A weak s-bijective schedule $\gamma := (\gamma_0, \ldots, \gamma_r)$ is a tuple of $r \geq s$ linear functions $\gamma_i : \{0,1\}^{sn} \to \{0,1\}^n$ such that for any contiguous s-subtuple, $\gamma' = (\gamma_i, \ldots, \gamma_{i+s-1})$ of γ, the mapping*

$$(k, t) \mapsto (\gamma_i(k, t), \ldots, \gamma_{i+s-1}(k, t))$$

is a bijection.

It is obvious to see that strong s-bijectivity implies weak s-bijectivity. However, the converse may not be true. By definition, a strong s-bijective schedule cannot collide on more than $(s-1)$ round tweakeys for any two distinct tweaks. On the contrary, a weak s-bijective schedule only requires at least one distinct round tweakey for every consecutive s rounds. In the following results, we show that weak s-bijectivity of the public[1] part of the tweak(ey) schedule is sufficient for desired security with minimal number of rounds. In particular, we will not employ the strong bijectivity property in this chapter, but it remains of independent interest for achieving better security in other schemes.

Thus, in the indistinguishability setting, we consider the TEML construction with a weak α-bijective tweakey schedule. For simplicity, we denote by r-TEML, the TEML construction consisting of r rounds.

4.1.2 Security of TEML

The security of TEML is analyzed in two main parts. First, we establish IND-CCA security for r rounds TEML. Second, we show sequential indifferentiability for TEML by presenting an attack on $r+2$ rounds and proving that $r+3$ rounds are sufficient to achieve sequential indifferentiability.

IND-CCA Security of TEML. In the indistinguishability framework, the underlying key is secret. Additionally, it is quite common to consider independent and uniform at random keys at each round. We will also employ this assumption. More specifically, we assume that the key is an $(r+1)$ tuple $\mathbf{k} = (k_0, \ldots, k_r)$, where $k_i \xleftarrow{\$} \{0,1\}^n$, and k_i is independent of k_j, for all $i \neq j \in [0; r]$. In addition, we take $\lambda_i(\mathbf{k}) = k_i$, i.e., we ignore the key schedule λ, and simply XOR the i-th component of $\mathbf{k}$ as the i-th round key. The following result establishes the IND-CCA security of r-TEML for any $r \geq 2$.

Theorem 4.1.1 (IND-CCA Security of r-TEML). *Let $r \geq \alpha + 1$ be an even integer and $r' = r/2$. Let $q_c, q_p, q_{\max}$ be positive integers such that $q_{\max} = \max\{q_c, q_p\}$ and $q_c + q_p < N/2$. Then, for any weak α-bijective tweak schedule δ, we have*

$$\mathbf{Adv}^{\mathit{IND\text{-}CCA}}_{r\text{-}\mathtt{TEML}^{\delta},\mathbf{P}}(q_c, q_p) \leq \sqrt{2^{4+3r'} q_c \left(\frac{2q_{\max}}{N}\right)^{\lceil \frac{r'}{\alpha} \rceil - 1}}.$$

[1] In the indistinguishability setting, this is the tweak part of the tweakey, whereas in the indifferentiability setting the entire tweakey is controlled by the adversary.

For odd $r \geq 3$, we have:

$$\mathbf{Adv}^{\mathit{IND\text{-}CCA}}_{r\text{-}\mathtt{TEML}^{\delta,\mathsf{P}}}(q_c, q_p) \leq \mathbf{Adv}^{\mathit{IND\text{-}CCA}}_{(r-1)\text{-}\mathtt{TEML}^{\delta,\mathsf{P}}}(q_c, q_p)$$

More concretely, r-TEML achieves IND-CCA security up to $O(N^{\frac{r-2\alpha}{r}})$ queries. Note that, we can use the upper bound of Theorem 4.1.1 for both r and $r-1$ (if r is odd). Hence, in the following, we always assume that r is even.

Sequential Indifferentiability of TEML. In the sequential indifferentiability setting, we are concerned with resistance against chosen-key attacks. In this case, since the adversary will always be allowed to choose its own keys, there will be functionally no difference between the tweak and key bits. In other words, the full tweakey is public and controlled by the adversary. Consequently, we need weak bijectivity property for the entire tweakey input. In the following results we take the tweakey size to be rn bits.

We provide two results in this direction. We start off with a simple attack (see Section 4.3.1) on $(r+2)$-TEML with a r-bijective tweakey schedule δ. This clearly establishes that $r+3$ rounds are necessary for security.

Lemma 4.1.1 (Seq. Indiff. Attack on $r+2$ Rounds)**.** *For any efficient simulator* Sim *making at most σ oracle queries to the ideal cipher* P*, there exists a sequential distinguisher $\mathcal{D}$ with at most $2r+6$ total query cost such that:*

$$\left|\Pr\left(\mathcal{D}^{\mathsf{P},\mathsf{Sim}^{\overline{\mathsf{P}}}} = 1\right) - \Pr\left(\mathcal{D}^{\mathtt{TEML}^{\delta,\mathbf{P}},\mathbf{P}} = 1\right)\right| \geq 1 - \frac{1}{N-1} - \frac{q'^4}{2N},$$

where $q' = 2r + \sigma + 6$ is the total calls to P *from $\mathcal{D}$ and* Sim *combined.*

The proof of this lemma mostly follows the strategy used in a similar attack on Even-Mansour cipher [91]. For completeness, we provide the proof in Section 4.3.2.

Next, in Theorem 4.1.2, we show that $r+3$ rounds are also sufficient for sequential indifferentiability.

Theorem 4.1.2 (Seq. Indiff. on $r+3$ Rounds)**.** *Let $q, \sigma, t \in \mathbb{N}, \varepsilon \in [0,1]$. Suppose $q^{r+1} \leq N/4$. Then, the $(r+3)$-round* TEML *construction with a weak r-bijective tweakey schedule γ is $(q, \sigma, T, \varepsilon)$-sequentially indifferentiable from an ideal cipher, where $\sigma = q^{r+1}$, $T = O(q^{r+1})$, and*

$$\mathbf{Adv}^{\mathsf{seq\text{-}indiff}}_{r\text{-}\mathtt{TEML}^{\gamma,\mathbf{P}}}(q, \sigma, T) \leq \varepsilon = \frac{\left((r+5)^2 + 32\right) q^{2r+2}}{N}.$$

4.2 Proof of IND-CCA Security of TEML

4.2.1 Setup for The Proof of Theorem 4.1.1

Fix a computationally unbounded and deterministic adversary $\mathscr{A}$ that maximizes the advantage. Let $\mathcal{T} = \{0,1\}^\eta$. Given a tuple $\mathbf{t} = (t_1, \ldots, t_{q_c}) \in \mathcal{T}^{q_c}$, we denote by $\Omega_{\mathbf{t}} \subseteq (\{0,1\}^n)^{q_c}$ the set of all inputs $\mathbf{x} = (x_1, \ldots, x_{q_c})$ such that all pairs (t_i, x_i) are pairwise distinct, i.e.,

$$\Omega_{\mathbf{t}} := \{\mathbf{x} := (x_1, \ldots, x_{q_c}) \in (\{0,1\}^n)^{q_c} : \forall i \neq j, (x_i, t_i) \neq (x_j, t_j)\}.$$

Transcripts. The interaction of $\mathscr{A}$ with its oracles can be summarized in a *query transcript* $(Q_C, Q_{P_1}, \ldots, Q_{P_r})$ of the attack. Here Q_C records the queries to the construction oracle which contains all triples $(t, x, y) \in \mathcal{T} \times \{0,1\}^n \times \{0,1\}^n$ such that $\mathscr{A}$ either made the direct query (t, x) to the construction oracle and received answer y, or made the inverse query (t, y) and received answer x. Similarly, for each $i \in [1; r]$, Q_{P_i} contains the queries to the round permutation P_i in the form of pairs $(u, v) \in \{0,1\}^n \times \{0,1\}^n$ such that $\mathscr{A}$ either made the direct query u to permutation P_i and received answer v, or made the inverse query v and received answer u. Note that the queries are recorded in a directionless and unordered fashion, but by our assumption that $\mathscr{A}$ is deterministic, there is a one-to-one mapping between this representation and the raw transcript of the interaction of $\mathscr{A}$ with oracles (see [81, 89] for more details). Recall, that by our assumption $\mathscr{A}$ never makes pointless queries. So, each query to the construction oracle results in a distinct triple in Q_C, and each query to P_i results in a distinct pair in Q_{P_i}, so that $|Q_C| = q_c$ and $\left|Q_{P_i}\right| = q_p$ for each $i \in [1; r]$ since we assume that $\mathscr{A}$ always makes the maximal number of allowed queries to each oracle. Let m denote the number of distinct tweaks appearing in Q_C, and q_i the number of queries for the i-th tweak, $i \in [1; m]$, using an arbitrary ordering of tweaks. Note that m may depend on the answer received from the oracles, yet we have $\sum_{i=1}^{m} q_i = q_c$.

Let $\tau' = (Q_C, Q_{P_1}, \ldots, Q_{P_r})$ be the resulting transcript. We say that τ' is *attainable* (with respect to some fixed adversary $\mathscr{A}$) if the probability to realize τ' in an interaction of $\mathscr{A}$ with $(\widetilde{\mathsf{P}}, \mathbf{P})$ (the ideal world) is non-zero. Let Θ denote the set of all attainable transcripts. We denote by μ_{re} (resp. μ_{id}), the probability distribution of the transcript induced in the real world (resp. the ideal world). Note that these two probability distributions depend on the adversary. By a slight abuse of notations, we reuse the same notations to denote the random variables distributed according to these distributions.

Given a permutation queries transcript Q and a permutation P, $P \vdash Q$ (referred as P *extends* Q) denotes the event $P(u) = v$ for all $(u, v) \in Q$. By extension, given a tuple of permutation queries transcript $Q_{\mathbf{P}} = (Q_{P_1}, \ldots, Q_{P_r})$ and a tuple of permutations $\mathbf{P} = (P_1, \ldots, P_r)$, $\mathbf{P} \vdash Q_{\mathbf{P}}$ (referred as $\mathbf{P}$ *extends* $Q_{\mathbf{P}}$) denotes the event $\wedge_{i=1}^{r}(P_i \vdash Q_{P_i})$. Note that for a permutation transcript of size q_p, we have:

$$\Pr_{P \xleftarrow{\$} \mathsf{Perm}(n)}(P \vdash Q) = \frac{1}{(N)_{q_p}}.$$

Thus, it follows that,

$$\Pr_{\mathbf{P} \xleftarrow{\$} \mathsf{Perm}(n)^r}(\mathbf{P} \vdash Q_{\mathbf{P}}) = \frac{1}{(N)_{q_p}^r},$$

as the permutations $\mathbf{P} = (P_1, \ldots, P_{r'})$ are uniformly random and independent.

Similarly, given a tweakable permutation transcript $\widetilde{Q}$ and a tweakable permutation $\widetilde{P}$, we say that $\widetilde{P} \vdash \widetilde{Q}$, if $\wp(t, x) = y$ for all $(t, x, y) \in \widetilde{Q}$. For a tweakable permutation transcript $\widetilde{Q}$ with m distinct tweaks and q_i queries corresponding to the i-th tweak, we

have:

$$\Pr_{\widetilde{P} \twoheadleftarrow_{\$} \widetilde{\mathsf{Perm}}(\eta;n)} \left(\widetilde{P} \vdash \widetilde{Q}\right) = \prod_{i=1}^{m} \frac{1}{(N)_{q_i}}.$$

So, the probability of getting any attainable transcript $\tau' = (Q_C, Q_{\mathbf{P}})$ in the ideal world is

$$\Pr\left(\mu_{\mathsf{id}} = \tau'\right) = \left(\frac{1}{(N)_{q_p}}\right)^r \times \prod_{i=1}^{m} \frac{1}{(N)_{q_i}}.$$

In the real world, the probability to obtain τ' is

$$\Pr\left(\mu_{\mathsf{re}} = \tau'\right) = \left(\frac{1}{(N)_{q_p}}\right)^r \times p(\tau'),$$

where $p(\tau') := \Pr\left(\mathbf{P} \twoheadleftarrow_{\$} \mathsf{Perm}\,(n)^r : \mathtt{TEML}_{\mathbf{k}}^{\mathbf{P}} \vdash Q_C \mid \mathbf{P} \vdash Q_{\mathbf{P}}\right)$.

Proof Overview. Let us fix an IND-CCA adversary $\mathscr{A}$ against the r-TEML construction. The first part of the proof relies on the famous H-coefficient technique introduced in Section 2.4. Our aim will be to use Corollary 2.4.1 with $\Theta_{\text{good}} = \emptyset$. To do so we lower bound the pointwise proximity of the transcripts in both worlds. We start by dividing the r-TEML construction into two r'-TEML constructions as follows. For any $\mathbf{k} = (k_0, \ldots, k_r) \in (\{0,1\}^n)^{r+1}$, and tweak schedule $\delta = (\delta_0, \ldots, \delta_r)$, any permutation tuple $\mathbf{P} = (P_1, \ldots, P_r)$, any $t \in \mathcal{T}$, and any $x \in \{0,1\}^n$, one has

$$r\text{-}\mathtt{TEML}_{\mathbf{k}}^{\delta,\mathbf{P}}(t,x) = \left(r'\text{-}\mathtt{TEML}_{\mathbf{k}_2}^{\delta^2,\mathbf{P}_2}\right)^{-1} \left(t, r'\text{-}\mathtt{TEML}_{\mathbf{k}_1}^{\delta^1,\mathbf{P}_1}(t,x) \oplus \delta_{r'}(t)\right), \tag{4.3}$$

where

$$\begin{aligned}
&\mathbf{P}_1 = (P_1, \ldots, P_{r'}), \mathbf{P}_2 = (P_r, \ldots, P_{r'+1}),\\
&\mathbf{k}_1 = (k_0, \ldots, k_{r'-1}, k_{r'} \oplus k'), \mathbf{k}_2 = (k_r, \ldots, k_{r'+1}, k'),\\
&\delta^1 = (\delta_0, \ldots, \delta_{r'}), \delta^2 = (\delta_r, \ldots, \delta_{r'+1}),
\end{aligned}$$

for any $k' \in \{0,1\}^n$. Hence, the r-TEML construction with uniformly random keys and round permutations can be seen as the composition (up to a shift) of two independent instances of the r'-TEML construction, also with uniformly random keys and round permutations. The crucial point of this proof will be to upper bound the statistical distance between the distribution of the outputs of r'-TEML *conditioned on partial information on the permutations* (namely $P_i \vdash Q_{P_i}$ for $i = 1, \ldots, r'$) and the uniform distribution on $\Omega_{\mathbf{t}}$.

Fix $\mathbf{t} = (t_1, \ldots, t_{q_c})$ and $\mathbf{x} = (x_1, \ldots, x_{q_c}) \in \Omega_{\mathbf{t}}$. We define a distribution of the outputs $r'\text{-}\mathtt{TEML}_{\mathbf{k}}^{\mathbf{P}}$ conditioned on the partial information on the permutations.

Definition 4.2.1. *Let $\mu_{\mathbf{t},\mathbf{x},Q_{\mathbf{P}}}$ be the distribution of the tuple*

$$r'\text{-}\mathtt{TEML}_{\mathbf{k}}^{\mathbf{P}}(\mathbf{t},\mathbf{x}) := \left(r'\text{-}\mathtt{TEML}_{\mathbf{k}}^{\mathbf{P}}(t_1,x_1), \ldots, r'\text{-}\mathtt{TEML}_{\mathbf{k}}^{\mathbf{P}}(t_{q_c},x_{q_c})\right)$$

conditioned on the event $\mathbf{P} \vdash Q_{\mathbf{P}}$ (i.e. when the key $\mathbf{k} = (k_0, \ldots, k_{r'})$ is uniformly random and the permutation $\mathbf{P} = (P_1, \ldots, P_r')$ are uniformly random among permutation satisfying $\wedge_{i=1}^{r'}(P_i \vdash Q_{P_i})$).

We denote by $\mu^*_{\mathbf{t}}$ the uniform distribution on $\Omega_{\mathbf{t}}$. The following lemma, establishes an appropriate upper bound on $\left\|\mu_{\mathbf{t},\mathbf{x},\mathcal{Q}_\mathbf{P}} - \mu^*_{\mathbf{t}}\right\|$, which is the main technical part of this section.

Lemma 4.2.1. *Let $q_c, q_p \in \mathbb{N}$ such that $q_c + q_p < N/2$, and $q_{\max} = \max\{q_c, q_p\}$. Fix any attainable permutation transcript $\mathcal{Q}_\mathbf{P}$ and any $\mathbf{t} \in \mathcal{T}^{q_c}$, $\mathbf{x} \in \Omega_{\mathbf{t}}$. Then, we have*

$$\left\|\mu_{\mathbf{t},\mathbf{x},\mathcal{Q}_\mathbf{P}} - \mu^*_{\mathbf{t}}\right\| \leq 8^{r'} q_c \left(\frac{2q_{\max}}{N}\right)^{\lceil \frac{r'}{\alpha} \rceil - 1}.$$

4.2.2 Proof of Lemma 4.2.1

Fix any attainable permutation queries transcript $\mathcal{Q}_\mathbf{P} = (\mathcal{Q}_{P_1}, \ldots, \mathcal{Q}_{P_{r'}})$ and $\mathbf{t} = (t_1, \ldots, t_{q_c}) \in \mathcal{T}^{q_c}$, $\mathbf{x} = (x_1, \ldots, x_{q_c}) \in \Omega_{\mathbf{t}}$. Our main task is to upper bound $\left\|\mu_{\mathbf{t},\mathbf{x},\mathcal{Q}_\mathbf{P}} - \mu^*_{\mathbf{t}}\right\|$.

As a first step, we can split the computation of $\left\|\mu_{\mathbf{t},\mathbf{x},\mathcal{Q}_\mathbf{P}} - \mu^*_{\mathbf{t}}\right\|$ into q_c simpler computations. The idea is to construct a distribution ν_l for every $l \in [0; q_c]$ such that ν_l is the distribution of the outputs of a random instance of r'-$\mathtt{TEML}^{\mathbf{P}}_{\mathbf{k}}$ queried with (t_i, x_i) for $i \in [1; l]$ and the last $q_c - l$ queries keep the same tweak t_i as in adversarial queries, but their block inputs z_i are chosen uniformly at random among the values that were not queried. More precisely, for each $l \in [0; q_c]$, let $\mathbf{z} = (z_1, \ldots, z_{q_c})$ be a tuple of queries such that:

$$\begin{cases} z_i = x_i, \forall i \in [1; l], \\ z_i \twoheadleftarrow_{\$} \{0,1\}^n \setminus \left\{z_j | t_j = t_i, j < i\right\}, \forall i > l. \end{cases}$$

This means that the first l queries are the adversary's queries and the remaining z_i are chosen uniformly at random among all the possible values (all queries have to be pairwise distinct). Denote by ν_l the distribution of r'-$\mathtt{TEML}^{\mathbf{P}}_{\mathbf{k}}(\mathbf{t}, \mathbf{z})$, conditioned on $\mathbf{P} \vdash \mathcal{Q}_\mathbf{P}$. Hence we have:

$$\left\|\mu_{\mathbf{t},\mathbf{x},\mathcal{Q}_\mathbf{P}} - \mu^*_{\mathbf{t}}\right\| = \left\|\nu_{q_c} - \nu_0\right\| \leq \sum_{l=0}^{q_c-1} \left\|\nu_{l+1} - \nu_l\right\|. \tag{4.4}$$

Note that for $l = q_c$, $z_i = x_i$, $\forall i \in [1; q_c]$ and hence r'-$\mathtt{TEML}^{\mathbf{P}}_{\mathbf{k}}(\mathbf{t}, \mathbf{z}) = r'$-$\mathtt{TEML}^{\mathbf{P}}_{\mathbf{k}}(\mathbf{t}, \mathbf{x})$ leads to $\nu_{q_c} = \mu_{\mathbf{t},\mathbf{x},\mathcal{Q}_\mathbf{P}}$. It is easy to see that ν_0 is identical to $\mu^*_{\mathbf{t}}$.

In Lemma 4.2.2, we upper bound the total variation distance $\|\nu_{l+1} - \nu_l\|$.

Lemma 4.2.2 (Hybrids-Distance)**.** *Let $q_c, q_p \in \mathbb{N}$ such that $q_c + q_p < N/2$ and $q_{\max} = \max\{q_c, q_p\}$. For any $l \in [0; q_c - 1]$, we have*

$$\left\|\nu_{l+1} - \nu_l\right\| \leq 8^{r'} \left(\frac{2q_{\max}}{N}\right)^{\lceil \frac{r'}{\alpha} \rceil - 1}.$$

The proof of Lemma 4.2.1 follows from (4.4) and Lemma 4.2.2. Before moving to the proof of Lemma 4.2.2 we first conclude the proof of Theorem 4.1.1.

Concluding the proof of Theorem 4.1.1. The final result in Theorem 4.1.1 can be obtained by relying on the following composition lemma, whose proof is identical to [89, Lemma 11].

Lemma 4.2.3. *Let r be an even integer and $r' = r/2$. Let $q_c, q_p \in \mathbb{N}$ and $q_{\max} = \max\{q_c, q_p\}$. Assume that there exists an ε such that, for any attainable queries transcript $\boldsymbol{Q}_{\mathbf{P}}$ and every $\mathbf{t} \in \mathcal{T}^{q_c}, \mathbf{x} \in \Omega_{\mathbf{t}}$, we have*

$$\left\|\mu_{\mathbf{t},\mathbf{x},\boldsymbol{Q}_{\mathbf{P}}} - \mu^*_{\mathbf{t}}\right\| \leq \varepsilon.$$

Then, for any attainable transcript queries τ', one has

$$\Pr\left(\mu_{\mathsf{re}} = \tau'\right) \geq (1 - 4\sqrt{\varepsilon})\Pr\left(\mu_{\mathsf{id}} = \tau'\right), \tag{4.5}$$

The proof of Theorem 4.1.1 follows from Corollary 2.4.1, Lemma 4.2.1 Lemma 4.2.3.

4.2.3 Proof of Hybrids-Distance Lemma

In order to prove Lemma 4.2.2, we remain with the task of giving an upper bound on the total variation distance between ν_{l+1} and ν_l, for each $l \in [0; q_c - 1]$. For the rest of this proof, since we are considering a single instance of r'-TEML, we drop the number of rounds r' and simply denote it by TEML in order to lighten the notation.

Note that we only consider the first $l+1$ elements of the two tuples of outputs since for both distributions, the i-th input for $i > l+1$ is sampled at random. In other words,

$$\|\nu_{l+1} - \nu_l\| = \left\|\nu'_{l+1} - \nu'_l\right\|, \tag{4.6}$$

where ν'_{l+1} and ν'_l are the respective distributions of the $l+1$ first outputs of the cipher as defined in Section 4.1. We will construct a suitable coupling of the two distributions, ν'_{l+1} and ν'_l, and apply the coupling lemma (see Lemma 2.4.2) to bound the distance.

4.2.3.1 Coupling of ν'_{l+1} and ν'_l

To define the coupling of ν'_{l+1} and ν'_l, we consider the construction $\mathtt{TEML}^{\mathbf{P}}_{\mathbf{k}}$, where $\mathbf{P}$ satisfies $\mathbf{P} \vdash \boldsymbol{Q}_{\mathbf{P}}$. Namely, where the key $\mathbf{k} = (k_1, \ldots, k_{r'})$ is uniformly random and the permutations $\mathbf{P} = (P_1, \ldots, P_{r'})$ are uniformly random among permutation satisfying $\wedge_{i=1}^{r'}(P_i \vdash \boldsymbol{Q}_{P_i})$). It receives inputs $\mathbf{x}' = (x_1, \ldots, x_{l+1})$ and $\mathbf{t}' = (t_1, \ldots, t_{l+1})$, so that $\mathtt{TEML}^{\mathbf{P}}_{\mathbf{k}}(\mathbf{t}', \mathbf{x}')$ is distributed according to ν'_{l+1}.

We will now construct a second instance of $\mathtt{TEML}^{\mathbf{P}'}_{\mathbf{k}'}$ with inputs $\mathbf{z}' = (z_1, \ldots, z_{l+1})$ and $\mathbf{t}' = (t_1, \ldots, t_{l+1})$, satisfying the following properties:

(P-1) $\mathbf{P}' = \left(P'_1, \ldots, P'_{r'}\right) \in \mathsf{Perm}(n)^{r'}$ are uniformly random among permutation tuples satisfying $\mathbf{P}' \vdash \boldsymbol{Q}_{\mathbf{P}}$ and $\mathbf{k}' \twoheadleftarrow_{\$} (\{0,1\}^n)^{r'}$.

(P-2) $z_i = x_i$ for every $i \in [1; l]$, and $z_{l+1} \twoheadleftarrow_{\$} \{0,1\}^n \setminus \left\{x_j | t_j = t_{l+1}, j < l+1\right\}$;

(P-3) for each $i \in [1; l+1]$, if the outputs of the j-th round permutation in the computations of $\mathtt{TEML}^{\mathbf{P}}_{\mathbf{k}}(t_i, x_i)$ and $\mathtt{TEML}^{\mathbf{P}'}_{\mathbf{k}'}(t_i, z_i)$ are equal, then this also holds for all subsequent inner permutations.

Note that, the same tweaks are used for both ciphers. So, Properties (P-1) and (P-2) will ensure that $\texttt{TEML}^{\mathbf{P}'}_{\mathbf{k}'}(\mathbf{t}', \mathbf{z}')$ is distributed according to ν'_l.

Notations. For $i \in [1; r']$, we denote:

$$U_i = \{u_i | (u_i, v_i) \in \mathcal{Q}_{P_i}\}, \qquad V_i = \{v_i | (u_i, v_i) \in \mathcal{Q}_{P_i}\}.$$

For $i \in [1; l+1]$ and $j \in [1; r']$, we also define x_i^j (resp. y_i^j) as the output (rep. input) of the j-th round permutation, P_j when computing $\texttt{TEML}^{\mathbf{P}}_{\mathbf{k}}(t_i, x_i)$, and similarly z_i^j (resp. w_i^j) as the output (rep. input) of the j-th round permutation, P'_j when computing $\texttt{TEML}^{\mathbf{P}'}_{\mathbf{k}'}(t_i, z_i)$, i.e.,

$$\begin{cases} x_i^0 = x_i, & z_i^0 = z_i, \\ y_i^j = x_i^{j-1} \oplus k_j \oplus \delta_j(\mathbf{t}_j), & w_i^j = z_i^{j-1} \oplus k'_j \oplus \delta_j(\mathbf{t}_i) \\ x_i^j = P_j(y_i^j), & z_i^j = P'_j(w_i^j). \end{cases} \tag{4.7}$$

View of A Transcript. We refer to $\tau = ((x_i^j, y_i^j, z_i^j, w_i^j, \mathbf{k}, \mathbf{t}, U_j) : i \in [1; l+1],\ j \in [1; r'],\ \mathbf{k} = (k_1, \ldots, k_{r'}),\ \mathbf{t} = (t_1, \ldots, t_{l+1})))$, as an extension of the transcript $(\mathcal{Q}_{\mathbf{C}}, \mathcal{Q}_{\mathbf{P}})$, and call it the *view* of τ. Note that for a view, we must have

$$\left(x_i^j = x_{i'}^j\right) \iff \left(y_i^j = y_{i'}^j\right), \left(z_i^j = z_{i'}^j\right) \iff \left(w_i^j = w_{i'}^j\right).$$

In order to apply the coupling lemma, we have to find how to correlate $(\mathbf{P}, \mathbf{k})$ and $(\mathbf{P}', \mathbf{k}')$ so that the outputs, $\left(x_1^{r'}, \ldots, x_{l+1}^{r'}\right)$ and $\left(z_1^{r'}, \ldots, z_{l+1}^{r'}\right)$, are equal with high probability. We choose $(\mathbf{P}, \mathbf{k})$ uniformly at random and we construct $(\mathbf{P}', \mathbf{k}')$ as a function of $(\mathbf{P}, \mathbf{k})$, i.e., $(\mathbf{P}', \mathbf{k}')$ will not be independent from $(\mathbf{P}, \mathbf{k})$. The only requirement is that both $(\mathbf{P}, \mathbf{k})$ and $(\mathbf{P}', \mathbf{k}')$ have the correct marginal distributions. We have to pay attention that the distribution of $(\mathbf{P}', \mathbf{k}')$ remains uniform in order for $\left(z_1^{r'}, \ldots, z_{l+1}^{r'}\right)$ to be distributed according to ν'_l.

We now describe how $(\mathbf{P}', \mathbf{k}')$ is constructed using $(\mathbf{P}, \mathbf{k})$. First, choose the key $\mathbf{k}' = \mathbf{k} = (k_1, \ldots, k_r)$. Next, we want both tuples $\mathbf{P}$ and $\mathbf{P}'$ to agree on the permutation queries, i.e., for any $i \in [1; r']$ and $(y, x) \in \mathcal{Q}_{P_i}$, we want $P'_i(y) = x$. Moreover, in order to obtain Property (P-3), we will want that for every $(i, j) \in [1; l] \times [1; r']$, $z_i^j = x_i^j$ and $w_i^j = y_i^j$. For the $(l+1)$-th query, we will try to make the outputs of the two corresponding permutations equal, at some round j, as long as it does not interfere with the previous rules, i.e., Properties (P-1) to (P-3). If it succeeds, by Property (P-3), the outputs of all the subsequent round permutations must be equal. Formally, we describe the following sampling.

Coupling the first l queries. For every $i \in [1; l]$, the i-th queries x_i^0 and z_i^0 are equal by definition. Considering the system (4.7), we set $P'_j\left(w_i^j\right) = P'_j\left(y_i^j\right) = P_j\left(y_i^j\right)$ for every $i \in [1; l]$ and $j \in [1; r']$. This implies that the first l outputs $\left(x_1^{r'}, \ldots, x_l^{r'}\right)$ and $\left(z_1^{r'}, \ldots, z_l^{r'}\right)$ are equal.

Coupling the $(l+1)$-th query. For every $j \in [1; r']$ we define the coupling for the $(l+1)$-th query according to the following conditions:

(A-1) If $w_{l+1}^j \in U_j$ or there exists $i \in [1; l]$ such that $w_{l+1}^j = w_i^j = y_i^j$, then $z_{l+1}^j = P_j'\left(w_{l+1}^j\right)$ is already determined; unless we have coupled z_{l+1}^{j-1} and x_{l+1}^{j-1} in a previous round, we cannot couple z_{l+1}^j and x_{l+1}^j at this round.

(A-2) else, if $w_{l+1}^j \notin U_j$ and $w_{l+1}^j \neq w_i^j$ for all $i \in [1; l]$, then;

(a-1) If $y_{l+1}^j \in U_j$ or there exists $i \in [1; l]$ such that $y_{l+1}^j = y_i^j$, then we choose $z_{l+1}^j = P_j'\left(w_{l+1}^j\right)$ uniformly at random in $\{0,1\}^n \setminus (V_j \cup \{P_j'\left(w_i^j\right), i \leq l\})$, and, so we cannot couple z_{l+1}^j and x_{l+1}^j at this round.

(a-2) else, we define $P_j'\left(w_{l+1}^j\right) = P_j\left(y_{l+1}^j\right)$. This implies that $z_{l+1}^j = x_{l+1}^j$.

Note that, for the first l construction queries we define $\mathbf{P'}$ to be exactly same as $\mathbf{P}$ and for the $(l+1)$-th query we have defined $\mathbf{P'}$ by the above rules. Hence, using the fact that the keys and the tweaks are the same for both the ciphers, we can conclude that Property (P-3) is satisfied. So, once $z_{l+1}^j = x_{l+1}^j$, we must have $z_{l+1}^{j'} = x_{l+1}^{j'}$ for any subsequent round $j' \geq j$. In particular, for $j' = r'$, $z_{l+1}^{r'} = x_{l+1}^{r'}$. So, the coupling succeeds.

Uniformly random $(\mathbf{P'}, \mathbf{k'})$*:* Since $\mathbf{k'} = \mathbf{k}$ and $\mathbf{k}$ is uniformly random, $\mathbf{k'}$ is also uniformly random. During the coupling of the first l queries , we set $P_j'\left(w_i^j\right) = P_j\left(y_i^j\right)$ for every $i \in [1; l]$, $j \in [1; r']$ and $P_j\left(y_i^j\right)$ is uniformly random among possible values, thus $P_j'\left(w_i^j\right)$ is uniformly random among possible values.

Condition (A-1), says that if there is a collision with a previous input of P_j', we cannot choose the value of $P_j'(w_{l+1}^j)$ so this does not change anything to the distribution of P_j'. When the conditions of Condition (a-1) are met, we have:

- for some $i \in [1; l]$:

$$\begin{cases} P_j\left(y_{l+1}^j\right) = P_j\left(y_i^j\right) = P_j'\left(w_i^j\right) \\ w_{l+1}^j \neq w_i^j, \end{cases}$$

- or for some $(u_j, v_j) \in \mathcal{Q}_{P_j}$:

$$\begin{cases} P_j\left(y_{l+1}^j\right) = P_j\left(u_j\right) = P_j'\left(u_j\right) \\ w_{l+1}^j \neq u_j. \end{cases}$$

The two cases imply that $P_j'\left(w_{l+1}^j\right)$ is chosen uniformly random among possible values to keep P_j' uniformly distributed and distinct from $P_j'\left(w_i^j\right)$. Finally, in Condition (a-2), both $P_j(y_{l+1}^j)$ and $P_j'\left(w_{l+1}^j\right)$ are set to a uniformly at random chosen value excluding V_j. In conclusion, the permutations P_j' are uniformly random and independent as desired, whence $\left(z_1^{r'}, \ldots, z_{l+1}^{r'}\right)$ is distributed according to v_l'. Hence, the joint distribution,

$$\left(\mathtt{TEML}_{\mathbf{k}}^{\mathbf{P}}\left(\mathbf{t'}, \mathbf{x'}\right), \mathtt{TEML}_{\mathbf{k'}}^{\mathbf{P'}}\left(\mathbf{t'}, \mathbf{z'}\right)\right),$$

is created in such a way that the marginal distribution $\texttt{TEML}^{\mathbf{P}}_{\mathbf{k}}(\mathbf{t}', \mathbf{x}')$ and $\texttt{TEML}^{\mathbf{P}'}_{\mathbf{k}'}(\mathbf{t}', \mathbf{z}')$ are v'_{l+1} and v'_l, respectively. We can now apply Lemma 2.4.2 to obtain:

$$\left\|v'_{l+1} - v'_l\right\| \leq \Pr\left(\left(z_1^{r'}, \ldots, z_{l+1}^{r'}\right) \neq \left(x_1^{r'}, \ldots, x_{l+1}^{r'}\right)\right). \tag{4.8}$$

4.2.3.2 Coupling Failure

According to (4.8), we need to upper bound

$$\Pr\left(\left(z_1^{r'}, \ldots, z_{l+1}^{r'}\right) \neq \left(x_1^{r'}, \ldots, x_{l+1}^{r'}\right)\right).$$

Define the coupling failure event fail as

$$\mathsf{fail} : \left(z_1^{r'}, \ldots, z_{l+1}^{r'}\right) \neq \left(x_1^{r'}, \ldots, x_{l+1}^{r'}\right),$$

and note that since $z_i^{r'} = x_i^{r'}$ for all $i \in [1; l]$, we have:

$$\Pr(\mathsf{fail}) \leq \Pr\left(z_{l+1}^{r'} \neq x_{l+1}^{r'}\right). \tag{4.9}$$

Thus, from now on we will focus on the scenario where the final outputs in both instances of TEML are distinct, i.e., $z_{l+1}^{r'} \neq x_{l+1}^{r'}$.

In earlier works [209, 211, 89], each round was analyzed independently. However, for our purposes, this approach tends to provide loose upper bounds. The key innovation of our work is to consider the global event of failure across all rounds simultaneously, rather than bounding the probability for each round individually. We will briefly discuss the motivation behind this change.

Motivation for Our New Approach. Consider the following collision events on round $j \in [1; r']$,

$$\mathsf{F}_0^j : (y_{l+1}^j \in U_j), \qquad \mathsf{F}_2^j : (w_{l+1}^j \in U_j),$$
$$\mathsf{F}_1^j : \left(\exists i \in [1; r'] : y_{l+1}^j = y_i^j\right), \qquad \mathsf{F}_3^j : \left(\exists i \in [1; r'] : w_{l+1}^j = w_i^j\right).$$

Then, it is easy to see that, if $x_{l+1}^{r'} \neq z_{l+1}^{r'}$ then each round must have incurred a collision event. More precisely,

$$\mathsf{fail} \subseteq \bigcap_{j=1}^{r'} \left(\mathsf{F}_0^j \cup \mathsf{F}_1^j \cup \mathsf{F}_2^j \cup \mathsf{F}_3^j\right) := \mathsf{Col}(\tau).$$

From now on we say τ leads to a coupling failure if and only if $\mathsf{Col}(\tau)$ occurs. Fix a round $j \in [1; r']$ and suppose F_1^j occurs, then without loss of generality, there exists $i \in [1; l]$, such that $y_{l+1}^j = y_i^j$. In the proof of r-TEML ($\mathcal{H}$) [89] and CLRW2 [211], this gives an equation of the form

$$\mathcal{H}_{k_j}(t_{l+1}) \oplus \mathcal{H}_{k_j}(t_i) = x_{l+1}^{j-1} \oplus x_i^{j-1},$$

where $\mathcal{H}$ is an AXU hash function. Since, $t_{l+1} \neq t_i$, the event can be easily bounded by the AXU property (see Definition 2.3.8) of $\mathcal{H}$. However, in our case the same event gives rise to the following equation:

$$x_{l+1}^{j-1} \oplus \delta_j(\mathbf{t}_{l+1}) \oplus k_j = x_i^{j-1} \oplus \delta_j(\mathbf{t}_i) \oplus k_j. \tag{4.10}$$

Note that in the equation above the key k_j cancels out. As a result, we can no longer use the AXU property.

To analyze the probability of (4.10) occurring, we distinguish between the possible values of the function $d(\mathbf{t}_{l+1}, \mathbf{t}_i) = \mathbf{t}_{l+1} \oplus \mathbf{t}_i$:

- if $d(\mathbf{t}_{l+1}, \mathbf{t}_i) = 0$, it implies that $x_{l+1}^{j-1} = x_i^{j-1}$, i.e., the current collision is implied by a similar collision in the previous round. Luckily, the number of consecutive implied collisions is bounded by $\alpha - 1$, as otherwise it would violate the weak α-bijectivity property of δ;
- Otherwise, $d(\mathbf{t}_{l+1}, \mathbf{t}_i) \neq 0$. Distinguish between two cases.
  1. if $y_{l+1}^{j-1} \notin U_{j-1}$ or $y_i^{j-1} \notin U_{j-1}$ holds, then we can simply use the randomness of P_{j-1}, since this value is not known to the adversary.
  2. Otherwise, the values of y_{l+1}^{j-1}, y_i^{j-1} are known to the adversary, whence we cannot use the randomness of P_{j-1} on these inputs. However, $y_{l+1}^{j-1}, y_i^{j-1} \in U_{j-1}$ is still an event over the randomness of the round key k_{j-1}. Therefore, it holds with probability at most $q_p N^{-1}$. Using the union bound, we obtain that F_1^j occurs with probability at most $\left(q_p \cdot q_c\right) N^{-1}$. Looking ahead momentarily, this is far from a desirable upper bound. Interestingly, we can actually extend this same argument to previous rounds until we reach a round $j' < j$, where $y_{l+1}^{j'} \notin U_{j'}$ or $y_i^{j'} \notin U_{j'}$, at which point we can terminate the argument. Considering such an extension might actually be useful in getting a better bounds for F_1^j (res. F_3^j).

Note that the argument discussed in the last case above creates a chain structure for calculating the probability that F_1^j (res. F_3^j) it starts at round j with (4.10) and stops once a source of randomness has been found at round $j' < j$. The following definition gives a concrete formulation of this idea.

Definition 4.2.2 (Chain). *For symbols $(C, c) \in \{(Y, y), (W, w)\}$, and indices $i \in [1; l]$, $j \in [2; r']$, $p \in [0; j-1]$, we say a (i, j, p)-chain, denoted $C(i, j, p)$, occurs in the view τ if the following conditions occur:*

(C-1) $c_{l+1}^j = c_i^j$;
(C-2) $d(\mathbf{t}_{l+1}, \mathbf{t}_i) \neq 0$;
(C-3) $\forall j' \in [j-p; j-1]$, $c_{l+1}^{j'}, c_i^{j'} \in U_{j'}$, $\qquad j > p+1 \rightarrow \left|\left\{c_{l+1}^{j-p-1}, c_i^{j-p-1}\right\} \cap U_{j-p-1}\right| < 2$.

If $p = 0$, we refer to the special case where Condition (C-3) does not occur, we call it an *empty chain*. Otherwise, if $p = j - 1$, then we call $C(i, j, p)$ a *complete chain*, and a *partial chain* in any other case. For a symbol $C \in \{\mathcal{Y}, \mathcal{W}\}$ and query $i \in [1; l]$ we denote by $\mathcal{C}(j, p)$ the set of all chains $C(i, j, p)$ where $p \in [0; j-1]$ is called the size of the chain.

4.2.3.3 Activity Pattern

More importantly, it is clear from the preceding discussion that we may have to consider a joint event on some consecutive rounds, as the earlier approach of bounding failure probability locally at each round will be tedious and loose. In our new approach, we associate each view τ with a string $\mathsf{S}(\tau) = s_1 \dots s_{r'}$ over the alphabet $\Gamma = \{\top, \bot, 0, 1, 2, 3, 4, 5, 6, 7\}$, representing the failure at every round. Here, the symbol $\top$ corresponds to an empty symbol, while $\bot$ denotes a failing event that we disregard because its probability cannot be bounded.

Description of $\mathsf{S}(\tau)$. We give a description of the mapping $\tau \mapsto \mathsf{S}(\tau)$:

- We start with a string $\mathsf{S}(\tau) = (\top)^{r'}$ (our representation of an empty string);
- For any round $j \in [1; r']$, we assign a symbol to s_j the following way,
  - If F_0^j (collision with a permutation query) occurs we assign $s_j \leftarrow 0$. Otherwise, if F_2^j occurs we assign $s_j \leftarrow 4$. The randomness can be drawn from the key k_j;
  - Else if $\mathsf{F}_1^j \cup \mathsf{F}_3^j$ (collision between internal variables) occurs but $d(\mathbf{t}_{l+1}, \mathbf{t}_i) = 0$ then we assign $s_j \leftarrow \bot$, as this represents an implied collision;
  - Else, if $\mathcal{Y}(j, 0)$ is non empty (there is an empty chain) then we assign $s_j \leftarrow 1$, otherwise if $\mathcal{W}(j, 0)$ is non empty then we assign $s_j \leftarrow 5$, in this case the probability of the collision between the variables can be bounded with the randomness of the previous round;
- Once the first loop is done we start searching for chains. At this point any chain will be of size at least one. For any $j \in [2; r']$ we assign the following symbols,
  - Let p be maximal such that $\mathcal{Y}(j, p)$ is non empty, if $p > 0$, we assign $s_j \leftarrow 3$ and $s_{j'} \leftarrow 2$ for every $j' \in [j - p; j - 1]$;
  - Otherwise, let p' be maximal such that $\mathcal{W}(j, p)$ is non empty, if $p' > 0$, we assign $s_j \leftarrow 7$ and $s_{j'} \leftarrow 6$ for every $j' \in [j - p'; j - 1]$;

We call $\mathsf{S}(\tau)$ the *activity pattern*, or simply the pattern corresponding to τ. In a sense, $\mathsf{S}(\tau)$ gives a necessary local view of the activity at each round during the computation of both $\mathtt{TEML}_{\mathbf{k}}^{\mathbf{P}}(t', x')$ and $\mathtt{TEML}_{\mathbf{k}'}^{\mathbf{P}'}(t', z')$. It is easy to see that the following set captures the various patterns we can produce, i.e., let

$$\mathcal{P} = \left\{ S_1 \| \cdots \| S_d \ : \ \forall i \in [1; d],\ (S_i \in \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3) \wedge \sum_{i=1}^{d} |S_i| = r' \right\},$$

where $\mathcal{S}_1 = \cup_{i=1}^{r'} \{0, 1, 4, 5\}^i$, $\mathcal{S}_2 = \{2^i \| 3, 6^i \| 7 : i \in [1; r'-1]\}$, and $\mathcal{S}_3 = \{\bot^i \ : \ i \in [1; \alpha-1]\}$. The following lemma gives a complete the characterization of $\mathsf{S}(\tau)$.

Lemma 4.2.4. *For any view τ, if $\mathsf{Col}(\tau)$ occurred then $\mathsf{S}(\tau) \in \mathcal{P}$. Moreover, $\mathsf{S}(\tau)$ consists of at least $\lceil r'/\alpha \rceil$ non $\bot$ symbols.*

Proof. Let $\mathsf{S}(\tau) = s_1 \dots s_{r'}$ where for every $j \in [1; r']$, $s_j \in \Gamma$. The first part of the lemma is easy to see by the construction of $\mathsf{S}(\tau)$, the definition of $\mathcal{P}$, and α-bijectivity of δ. As for

the second part, note that, $s_1 \neq \perp$ (by definition). Divide the remaining $r' - 1$ symbols in contiguous substrings of length α, except the last which could be of length less than α. Then, using the α-bijectivity of δ, we have at least $1 + \lfloor \frac{r'-1}{\alpha} \rfloor \geq \lceil r'/\alpha \rceil$ non-$\perp$ symbols in $\mathsf{S}(\tau)$. □

We will be interested in the probability that the view τ produces a pattern $S \in \mathcal{P}$, i.e, $\Pr(\mathsf{S}(\tau) = S)$, which essentially covers the global failure event. For any string $S \in \mathcal{P}$, fix a representation $S = S_1 \| \ldots \| S_d$. Then, we can write $\mathsf{S}(\tau) := E_1(\tau) \| \cdots \| E_d(\tau)$ such that $|E_i(\tau)| = |S_i|$. Let d' be the number of strings such that $S_i \notin \bigcup_{\ell<\alpha} \perp^\ell$, and let $m_1 < m_2 < \ldots < m_{d'}$ be the indices corresponding to these sets.

Let E_j denote the event $(E_{m_1}(\tau) = S_{m_1}, \ldots, E_{m_j}(\tau) = S_{m_j})$ for any $j \in [1; d' - 1]$. We are interested in the following conditional probability

$$\Pr_{\tau}\left(E_{m_i}(\tau) = S_{m_i} \mid \mathsf{E}_{i-1}\right) \tag{4.11}$$

for all $i \in [1; d']$ and where E_0 is simply defined as the entire sample space (as to always hold). Note that, the above conditional event is well-defined, and non-trivial. This can be argued as follows: for distinct $i, i' \in [1; d']$, $E_{m_i}(\tau)$ and $E_{m_{i'}}(\tau)$ involve different rounds. Further, at round $j \in [1; r']$, we either use the randomness of the key k_j or that of the permutation P_{j-1}, which means that no two rounds share the same source of randomness.

In Lemma 4.2.5, we upper bound the conditional probability (4.11) depending on the type of string S_{m_i} for all $i \in [1; d']$.

Lemma 4.2.5. *Suppose $|S_{m_i}| = s$. Then*

1. *for $i = 1$, we have*
$$\Pr_{\tau}\left(E_{m_i}(\tau) = S_{m_i} \mid \mathsf{E}_{i-1}\right) \leq \left(\frac{2q_{\max}}{N}\right)^{s-1}.$$
2. *for $i > 1$, we have*
$$\Pr_{\tau}\left(E_{m_i}(\tau) = S_{m_i} \mid \mathsf{E}_{i-1}\right) \leq \left(\frac{2q_{\max}}{N}\right)^{s}.$$

Proof. Distinguish between two cases.

Case A. $S_{m_i} \in \mathcal{S}_1$**:** Suppose $E_{m_i}(\tau) = e_j \cdots e_{j+s-1}$ for some consecutive rounds $(j, \ldots, j + s - 1)$ such that $j \in [1; r' - s + 1]$. First, assume that $i > 1$, i.e., $E_{m_i}(\tau)$ is not a prefix of $\mathsf{S}(\tau)$. Let $\mathsf{D}_0 = \mathsf{E}_{i-1}$, and for all $s' \in [1; s - 1]$, let $\mathsf{D}_{s'}$ denote the event $(e_j = f_j, \ldots, e_{j+s'-1} = f_{j+s'-1}, \mathsf{E}_{i-1})$. Our goal is to compute

$$\Pr\left(e_{j+s'} = f_{j+s'} \mid \mathsf{D}_{s'}\right), \quad \forall s' \in [0; s - 2]. \tag{4.12}$$

Differentiate between two cases based on the value of f_j:

- If $f_{j+s'} \in \{0, 4\}$ then since $k_{j+s'}$ is uniform and independent of $\{k_1, \ldots, k_{j+s'-1}\}$, we have
$$\Pr\left(e_{j+s'} = f_{j+s'} | \mathsf{D}_{s'}\right) \leq \frac{q_p}{N}. \tag{4.13}$$

– Otherwise, $f_{j+s'} \in \{1,5\}$. Without loss of generality assume that $f_{j+s'} = 1$. Then by definition there exists $i' \in [1;l]$ such that $y_{l+1}^{i_j} = y_{i'}^{i_j}$. This gives rise to the following equation,

$$P_{j+s'-1}\left(y_{l+1}^{j+s'-1}\right) \oplus P_{j+s'-1}\left(y_{i'}^{j+s'-1}\right) = \delta_{j+s'}(t_{l+1} \oplus t_{i'}) \neq 0 \qquad (\mathrm{Eq}_{i'})$$

Further, since $\mathsf{F}_0^j \cup \mathsf{F}_2^j = \emptyset$, at least one of $y_{l+1}^{j+s'-1}$ or $y_{i'}^{j+s'-1}$ is fresh (does not belong to $U_{j+s'-1}$), whence we have,

$$\Pr\left(e_{j+s'} = f_{j+s'} | \mathsf{E}_{s'}\right) \leq \sum_{i' \leq l} \Pr\left((\mathrm{Eq}_{i'})\right) \leq \sum_{i' \leq l} \frac{1}{N - q_c - q_p} \leq \frac{2q_c}{N}. \qquad (4.14)$$

Combining (4.12), (4.13) and (4.14), one has,

$$\Pr\left(e_{j+s'} = f_{j+s'} | \mathsf{D}_{s'}\right) \leq \frac{2q_{\max}}{N}.$$

Finally, by applying the chain rule, we get

$$\Pr_{\tau}\left(E_{m_i}(\tau) = S_{m_i} \mid \mathsf{E}_{i-1}\right) = \prod_{s'=1}^{s} \Pr\left(e_{j+s'} = f_{j+s'} \mid \mathsf{D}_{s'}\right) \leq \left(\frac{2q_{\max}}{N}\right)^s. \qquad (4.15)$$

This proves the second part of the lemma for **Case A**. Now, if $i = 1$, then $E_i(\tau)$ is a prefix of $\mathsf{S}(\tau)$, i.e., $j = 1$, then we can view it as $E_{m_i}(\tau) := e_1 \| E'(\tau)$. Note that, the adversary can easily choose inputs $(x_1, \ldots, x_{l+1})$ such that

$$y_{l+1}^1 = x_{l+1}^0 \oplus \delta_1(t_{l+1}) \oplus k_1 = x_i^0 \oplus \delta_1(t_i) \oplus k_1 = y_i^1.$$

Therefore, the event $e_1 = f_1$ is trivially upper bounded (by 1). Applying the chain rule again gives

$$\begin{aligned}\Pr_{\tau}\left(E_{i_1}(\tau) = S_{i_1} \mid \mathsf{E}_0\right) &= \Pr\left(e_1 = f_1\right) \times \Pr\left(E'(\tau) = f_2 \cdots f_s \mid \mathsf{D}_1\right) \\ &\leq \left(\frac{2q_{\max}}{N}\right)^{s-1},\end{aligned}$$

where the last inequality follows from (4.15). This completes **Case A**.

Case B. $S_{m_i} \in \mathcal{S}_2$**:** Assume without loss of generality, $S_{m_i} = 2^{s-1} \| 3$ (the proof for the other type of chain is identical). Suppose $E_{m_i}(\tau) = e_j \cdots e_{j+s-1}$ for some consecutive rounds $(j, \ldots, j+s-1)$ such that $1 \leq i_1 < \ldots < i_s \leq r$. Our goal is to compute

$$\Pr\left(e_{i_1} = 2, \ldots, e_{i_{s-1}} = 2, e_{i_s} = 3 \mid \mathsf{E}_{i-1}\right). \qquad (4.16)$$

So there exists a chain $\Upsilon(i', j, p)$ chain with length $p = s - 1 \geq 1$ for some $i' \in [1;l]$. According to the definition of a chain given in Definition 4.2.2, the following conditions hold:

$$y_{l+1}^{i_s} = y_{i'}^{i_s}, \delta_j(t_{l+1} \oplus t_{i'}) \neq 0; \qquad (\mathrm{Ch}_{i'}^1)$$

$$\forall m \in [1; s-1], y_{l+1}^{i_m} \in U_{i_m}; \qquad (\mathrm{Ch}^2)$$

$$\forall m \in [1; s-1], y_{i'}^{i_m} \in U_{i_m}; \tag{$\mathrm{Ch}^3_{i'}$}$$
$$y_{l+1}^{i_1-1} \notin U_{i_1-1}; \tag{Ch^4}$$
$$y_{i'}^{i_1-1} \notin U_{i_1-1}, . \tag{$\mathrm{Ch}^5_{i'}$}$$

where only if $i_1 \geq 2$, (Ch^4) or ($\mathrm{Ch}^5_{i'}$) hold. Now, we can upper bound (4.16) using the probability of the conditions mentioned above. Namely,

$$\begin{aligned}
&\Pr\left(e_{i_1} = 2, \ldots, e_{i_{s-1}} = 2, e_{i_s} = 3 \mid \mathsf{E}_{i-1}\right) \\
&\leq \sum_{i' \leq l} \Pr\left((\mathrm{Ch}^1_{i'}) \wedge (\mathrm{Ch}^2) \wedge (\mathrm{Ch}^3_{i'}) \wedge \left((\mathrm{Ch}^4) \vee (\mathrm{Ch}^5_{i'})\right)\right) \\
&\leq \sum_{i' \leq l} \Pr\left((\mathrm{Ch}^1_{i'}) \wedge (\mathrm{Ch}^2) \wedge (\mathrm{Ch}^3_{i'}) \wedge (\mathrm{Ch}^4)\right) \\
&\qquad + \Pr\left((\mathrm{Ch}^1_{i'}) \wedge (\mathrm{Ch}^2) \wedge (\mathrm{Ch}^3_{i'}) \wedge (\mathrm{Ch}^5_{i'})\right) \\
&\leq \sum_{i' \leq l} \Pr\left((\mathrm{Ch}^1_{i'}) \wedge (\mathrm{Ch}^2) \wedge (\mathrm{Ch}^3_{i'}) \mid (\mathrm{Ch}^4)\right) \\
&\qquad + \Pr\left((\mathrm{Ch}^1_{i'}) \wedge (\mathrm{Ch}^2) \wedge (\mathrm{Ch}^3_{i'}) \mid (\mathrm{Ch}^5_{i'})\right). \qquad (4.17)
\end{aligned}$$

To analyze we the probabilities in the last inequality assume without loss of generality that ($\mathrm{Ch}^5_{i'}$) occurs. Then, for a fixed query $i \in [1; l]$ consider the event,

$$\mathsf{Y} - \mathsf{chain}_{i'} = \left((\mathrm{Ch}^1_{i'}) \wedge (\mathrm{Ch}^2) \wedge (\mathrm{Ch}^3_{i'}) \mid (\mathrm{Ch}^5_{i'})\right).$$

To analyze this event we will need to split the conditions (Ch^2) and ($\mathrm{Ch}^3_{i'}$) into sub-events. Our strategy will consist of upper bounding the probability of $y_{l+1}^{i_m} \in U_{i_m}$ for any $m \in [1; s-1]$, conditioned on $y_{l+1}^{i_s} = y_{i'}^{i_s}$. Assuming those, it will be possible to give a sharper upper bound on the probability of $y_{i'}^{i_m} \in U_{i_m}$ for any $m \in [1; s-1]$ (starting from $m = s-1$ down to 1). More precisely, for any $j \in [1; s-1]$, we define the events

$$\mathsf{Y}_{l+1}^{j} : \bigwedge_{m=1}^{j} \left(y_{l+1}^{i_m} \in U_{i_m}\right), \qquad \mathsf{Y}_{i'}^{j} : \bigwedge_{m=j}^{s-1} \left(y_{i'}^{i_m} \in U_{i_m}\right).$$

Additionally, we define $\mathsf{Y}_{l+1}^{0}, \mathsf{Y}_{i'}^{0}$ to be the entire sample space, so that they always hold. Using the chain rule, one has

$$\begin{aligned}
\Pr\left(\mathsf{Y} - \mathsf{chain}_{i'}\right) &= \prod_{m=1}^{s-1} \Pr\left(y_{l+1}^{i_m} \in U_{i_m} \mid \mathsf{Y}_{l+1}^{m-1} \wedge (\mathrm{Ch}^5_{i'})\right) \\
&\quad \times \Pr\left((\mathrm{Ch}^1_{i'}) \mid \mathsf{Y}_{l+1}^{s+1} \wedge (\mathrm{Ch}^5_{i'})\right) \\
&\quad \times \prod_{m=s-1}^{1} \Pr\left(y_{i'}^{i_m} \in U_{i_m} \mid \mathsf{Y}_{l+1}^{s-1} \wedge \wedge \mathsf{Y}_{i'}^{m+1} (\mathrm{Ch}^1_{i'}) \wedge (\mathrm{Ch}^5_{i'})\right) \\
&\leq \left(\frac{q_p}{N}\right)^{s-1} \times \prod_{m=s-1}^{1} \Pr\left(y_{i'}^{i_m} \in U_{i_m} \mid \mathsf{Y}_{l+1}^{s-1} \wedge \mathsf{Y}_{i'}^{m+1} \wedge (\mathrm{Ch}^1_{i'}) \wedge (\mathrm{Ch}^5_{i'})\right). \qquad (4.18)
\end{aligned}$$

where the reverse product simply means the indices are iterated from $m = s-1$ down to 1, the last inequality holds since all the round keys are uniform and independent, and $\Pr\left((\mathsf{Ch}^1_{i'}) \mid \mathsf{Y}^{s+1}_{l+1} \wedge (\mathsf{Ch}^5_{i'})\right) \leq 1$. The intricate part of the proof will be to upper bound

$$\prod_{m=s-1}^{1} \Pr\left(y^{i_m}_{i'} \in U_{i_m} \mid \mathsf{Y}^{s-1}_{l+1} \wedge \wedge \mathsf{Y}^{m+1}_{i'} \wedge (\mathsf{Ch}^1_{i'}) \wedge (\mathsf{Ch}^5_{i'})\right).$$

We start with the case where $E_i(\tau)$ is not a prefix of $\mathsf{S}(\tau)$. In this case, we have $i_1 \geq 2$. Note that

$$\prod_{m=s-1}^{1} \Pr\left(y^{i_m}_{i'} \in U_{i_m} \mid \mathsf{Y}^{s-1}_{l+1} \wedge \mathsf{Y}^{m+1}_{i'} \wedge (\mathsf{Ch}^1_{i'}) \wedge (\mathsf{Ch}^5_{i'})\right) \leq \Pr\left(y^{i_1}_{i'} \in U_{i_1} | \mathsf{U}_0\right), \quad (4.19)$$

where $\mathsf{U}_0 = \left(\mathsf{Y}^{s-1}_{l+1} \wedge \mathsf{Y}^2_{i'} \wedge (\mathsf{Ch}^1_{i'}) \wedge (\mathsf{Ch}^5_{i'})\right)$. To finalize the proof, we will give an upper bound on the last probability as in (4.19). Indeed, the main claim is that conditioned on U_0, there exists at most one $u_{i_1} \in U_{i_1}$ such that $y^{i_1}_{i'} = u_{i_1}$. We proceed by reverse recursion on $1 \leq m \leq s-1$. First for $m = s-1$, note that since U_0 occurs this implies the variable $x^{i_s-1}_{l+1}$ is fixed, since the round keys involved in the event Y^{s-1}_{l+1} are fixed and as a consequence $x^{i_s-1}_{i'}$ is fixed to,

$$x^{i_s-1}_{i'} = x^{i_s-1}_{l+1} \oplus \delta_{i_s-1}(\mathbf{t}_{l+1} \oplus \mathbf{t}i') \neq x^{i_s-1}_{l+1}.$$

Since the adversary never repeats a primitive query, this gives at most one choice of $v_{i_s-1} \in V_{i_s-1}$, such that $x^{i_s-1}_{i'} = v_{i_s-1}$. In other words, there is at most one $(u_{i_s-1}, v_{i_s-1}) \in Q_{P_{i_s-1}}$ such that $y^{i_s-1}_{i'} = u_{i_s-1}$. Applying the same argumentation, it is easy to show that there is at most one $(u_{i_m}, v_{i_m}) \in Q_{P_{i_m}}$, such that $y^{i_m}_{i'} = u_{i_m}$ for any $1 \leq m < s-1$, which proves our claim. Now, $y^{i_1}_{i'} = u_{i_1}$ can be rewritten as $x^{i_1-1}_{i'} = u_{i_1} \oplus k_{i_1} \oplus \delta_{i_1-1}(\mathbf{t}_{i'})$. Since, $y^{i_1-1}_{i'} \notin U_{i_1-1}$ and $x^{i_1-1}_{i'} = P_{i_1-1}(y^{i_1-1}_{i'})$ then by using the randomness of the permutation P_{i_1-1}, the value $y^{i_1-1}_{i'}$ is chosen uniformly at random from a set of size at least $N - q_c - q_p$. In conclusion, we have the following upper bound,

$$\Pr\left(y^{i_1}_{i'} \in U_{i_1} \mid \mathsf{U}_0\right) \leq \frac{1}{N - q_c - q_p} \leq \frac{2}{N}, \quad (4.20)$$

(see also Figure 4.1 for more intuition). Using (4.17), (4.18), (4.19), and (4.20), we have:

$$\Pr\left(e_{i_1} = 2, \ldots, e_{i_{s-1}} = 2, e_{i_s} = 3 \mid \mathsf{E}_{i-1}\right) \leq 4\left(\frac{q_{\max}}{N}\right)^s \leq \left(\frac{2q_{\max}}{N}\right)^s.$$

This proves the second part of the lemma for **Case B**. Now, assume that $E_i(\tau)$ is a prefix of $\mathsf{S}(\tau)$. In this case, we have

$$\Pr\left(e_{i_1} = 2, \ldots, e_{i_{s-1}} = 2, e_{i_s} = 3 \mid \mathsf{E}_{i-1}\right) \leq \Pr\left(\forall m \in [1; s-1], y^{i_m}_{l+1} \in U_{i_m}\right)$$

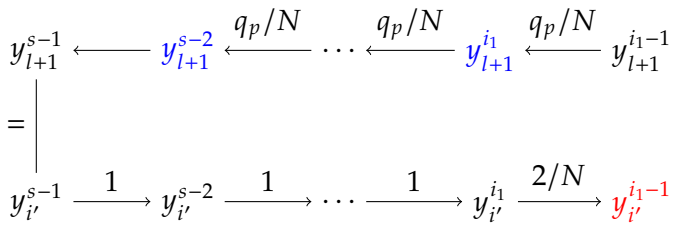

Figure 4.1: *Chain probability computation: each node represents a variable. Blue nodes denote variables with key randomness, while red nodes signify variables with permutation randomness. Each directed edge (y, z) indicates the probability of the equation induced by y given the randomness from z; WLOG: $y_i^{j'} \notin U_{j'}$.*

$$\leq \left(\frac{2q_p}{2^n}\right)^{s-1} \leq \left(\frac{2q_{\max}}{2^n}\right)^{s-1},$$

where the second inequality follows from the independence of key tuple. This proves the first part of the lemma for **Case B**, whence the proof is complete.

□

Finalizing The Proof. The following lemma gives an upper bound on the probability of an activity pattern to belong to $\mathcal{P}$.

Lemma 4.2.6. *Let q_c, q_p be positive integers, $q_{\max} = \max\{q_c, q_p\}$, and $q_c + q_p < N/2$. For any pattern $S \in \mathcal{P}$, we have*

$$\Pr_{\tau}(\mathsf{S}(\tau) = S) \leq \left(\frac{2q_{\max}}{N}\right)^{\lceil \frac{r'}{\alpha} \rceil - 1}.$$

Proof. By repeated application of Lemma 4.2.5, we have

$$\begin{aligned}
\Pr_{\tau}(\mathsf{S}(\tau) = S) &\leq \prod_{i=1}^{k'} \Pr_{\tau}\left(E_{m_i}(\tau) = S_{m_i} \mid \mathsf{E}_{i-1}\right) \\
&\leq \left(\frac{2q_{\max}}{N}\right)^{s_{m_1}-1} \times \prod_{i=2}^{k'} \left(\frac{2q_{\max}}{N}\right)^{s_{m_i}} \\
&\leq \left(\frac{2q_{\max}}{N}\right)^{\sum_{i=1}^{k'} s_{m_i} - 1} \\
&\leq \left(\frac{2q_{\max}}{N}\right)^{\lceil \frac{r'}{\alpha} \rceil - 1}
\end{aligned}$$

where $\sum_{i=1}^{k'} s_{m_i} \geq \lceil \frac{r'}{\alpha} \rceil$ comes from Lemma 4.2.4. This completes the proof. □

Now, we return to our main problem, i.e., (4.9). We have,

$$\Pr(\mathsf{fail}) \leq \sum_{\tau} \Pr(\mathsf{Col}(\tau))$$

$$\leq \sum_{S\in\mathcal{P}} \Pr\left(\mathsf{Col}(\tau) \wedge \mathsf{S}(\tau) = S\right)$$
$$\leq \sum_{S\in\mathcal{P}} \Pr\left(\mathsf{S}(\tau) = S\right)$$
$$\leq 8^{r'} q_c \left(\frac{2q_{\max}}{N}\right)^{\lceil r'/\alpha\rceil - 1} \tag{4.21}$$

The Hybrids-Distance lemma follows from (4.9) and (4.21).

4.3 Sequential Indifferentiability of TEML

We begin by introducing some notations relevant to both the attack and the security proof.

Notations. In this section, we have access to a weak r-bijective tweakey schedule γ. For any contiguous r-tuple, $I = [i; i+r-1]$ we denote by $\gamma_I^{-1} : (\{0,1\}^n)^r \to \{0,1\}^{rn}$ the inverse mapping of the bijection

$$k \mapsto (\gamma_i(k), \ldots, \gamma_{i+r-1}(k)).$$

4.3.1 Sequential Indifferentiability Attack of $(r+2)$-TEML

In this part, we prove Lemma 4.1.1. To achieve this, we define a distinguisher $\mathscr{D}$ that has access to a cipher $\widetilde{\mathsf{P}}$ and a tuple of permutations $\mathbf{P}$, as described in Algorithm 4.3.1.

4.3.1.1 Interaction with The Real World

We claim that $\mathcal{D}$ outputs 1 with overwhelming probability when interacting with $(\texttt{TEML}^{\mathbf{P}}, \mathbf{P})$. As a first step, notice that K, K', K'', K''' are not pairwise distinct in only two scenarios, either $x_2 = x_2'$ or $x_2 \oplus x_2' = \gamma_1(K \oplus K')$. Note that the first scenario is impossible as if $x_2 = x_2'$ it implies that $y_2 = y_2'$, which in turn implies that $x_3 = x_3'$. Continuing this process we must have $y_{r+1} = y_{r+1'}$, hence the two round keys k_{r+1} and $k_{r+1'}$ must be equal in contradiction to our assumption. In the second scenario, note that the probability the equation $x_2 \oplus x_2' = \gamma_1(K \oplus K') \neq 0$ holds, where $x_2 = P_2^{-1}(y_2)$, $x_2' = P_2^{-1}(y_2')$, is satisfied with probability at most $1/(N-1)$, since P_2 is a random permutation. Moreover, note that since the schedule is linear, $\gamma_1(K + K') \oplus k_1'' \oplus k_1''' = 0$. Therefore, since the schedule is also r-bijective we get that $K \oplus K' \oplus K'' \oplus K''' = 0$.

Next, we show that conditioned on $\mathcal{D}$ not outputting 0 in Line 15, it always outputs 1. Consider the computational paths of inputs (K, x), (K', x'), (K'', x''), (K''', x''') and note that they are well defined from our last two observations. Following both computational paths of (K, x) and (K', x') inside the TEML cipher, it is easy to see that the input to P_{r+2} in both paths is x_{r+2}. Similarly, in both computational paths of (K'', X'') and (K''', x''') the input to the permutation P_{r+2} is $x_{r+2}' = y_{r+1} \oplus k_{r+1}' = y_{r+1}' \oplus k_{r+1}$. It implies that,

$$y \oplus y' = (y_{r+2} \oplus \gamma_{r+2}(K)) \oplus (y_{r+2}' \oplus \gamma_{r+2}(K')),$$

Algorithm 4.3.1 The sequential differentiator, $\mathcal{D}^{(\widetilde{\mathsf{P}},\mathbf{P})}$.

1: Choose $x_{r+2}, k_{r+1}, k'_{r+1} \twoheadleftarrow_{\$} \{0,1\}^n$ at random such that $k_{r+1} \neq k'_{r+1}$.
2: Compute $y_{r+1} \leftarrow x_{r+2} \oplus k_{r+1}, y'_{r+1} \leftarrow x_{r+2} \oplus k'_{r+1}$.
3: Query $x_{r+1} \leftarrow P_{r+1}^{-1}(y_{r+1}), x'_{r+1} \leftarrow P_{r+1}^{-1}(y'_{r+1})$.
4: **for** $i' \in [2; r]$ **do**
5: Let $i = (r-2) - i'$.
6: Choose $k_i \twoheadleftarrow_{\$} \{0,1\}^n$ at random.
7: Compute $y_i \leftarrow k_i \oplus x_{i+1}, y'_i \leftarrow k_i \oplus x'_{i+1}$.
8: Query $x_i \leftarrow P_i^{-1}(y_i), x'_i \leftarrow P_i^{-1}(y'_i)$.
9: **end for**
10: Let $I = [2; r+1]$, compute the master keys:

$$K \leftarrow \gamma_I^{-1}(k_2, \ldots, k_{r+1}), K' \leftarrow \gamma_I^{-1}(k_2, \ldots, k'_{r+1}).$$

11: Compute $y_1 \leftarrow x_2 \oplus \gamma_1(K), y'_1 \leftarrow x'_2 \oplus \gamma_1(K')$.
12: Query $x_1 \leftarrow P_1^{-1}(y_1), x'_1 \leftarrow P_1^{-1}(y'_1)$.
13: Compute the round keys $k''_1 \leftarrow y_1 \oplus x'_2, k'''_1 \leftarrow y'_1 \oplus x_2$.
14: Let $I' = [1; r]$, compute the master keys:

$$K'' \leftarrow \gamma_{I'}^{-1}(k''_1, k_2, \ldots, k_r), K''' \leftarrow \gamma_{I'}^{-1}(k'''_1, k_2, \ldots, k_r).$$

15: **if** K, K', K'', K'' are not pairwise distinct **then** return 0 **else** continue
16: Compute the inputs:

$$x \leftarrow x_1 \oplus \gamma_0(K), x' \leftarrow x'_1 \oplus \gamma_0(K'),$$
$$x'' \leftarrow x_1 \oplus \gamma_0(K''), x''' \leftarrow x'_1 \oplus \gamma_0(K''').$$

17: Query $y \leftarrow \widetilde{\mathsf{P}}(K, x), y' \leftarrow \widetilde{\mathsf{P}}(K', x'), y'' \leftarrow \widetilde{\mathsf{P}}(K'', x''), y''' \leftarrow \widetilde{\mathsf{P}}(k''', x''')$.
18: **if** $y \oplus y' \oplus y'' \oplus y''' = 0$ **then** return 1 **else** return 0

$$y'' \oplus y''' = (y'_{r+2} \oplus \gamma_{r+2}(K'')) \oplus (y_{r+2} \oplus \gamma_{r+2}(K''')). \tag{4.22}$$

where $y_{r+2} = P_{r+2}(x_{r+2})$ and $y'_{r+2} = P_{r+2}(x'_{i+2})$. Finally, one has,

$$y \oplus y' \oplus y'' \oplus y''' = \gamma_{r+2}(K \oplus K' \oplus K'' \oplus K''') = \gamma_{r+2}(0) = 0,$$

where the last equation comes from XORing the equations in (4.22), the definition of the master keys in Line 10 and the fact that γ_{r+2} is linear. Hence, we get the following upper bound,

$$\Pr\left(\mathcal{D}^{\mathtt{TEML}^{\mathbf{P}},\mathbf{P}} = 1\right) \geq 1 - \frac{1}{N-1}.$$

4.3.1.2 Interaction with The Ideal World

Consider now what happens when $\mathcal{D}$ interacts with $(\widetilde{\mathsf{P}}, \mathsf{Sim}^{\widetilde{\mathsf{P}}})$ for some efficient simulator Sim which makes at most σ queries when $\mathcal{D}$ makes at most $2r+6$ queries. Denote

by $\mathcal{R}$ the Turing machine which runs both $\mathcal{D}$ and Sim together and makes at most $q' = 2r + 6$ queries to $\widetilde{\mathsf{P}}$. Whenever $\mathcal{D}$ outputs 1, we see that $\mathcal{R}$ has successfully found four inputs $(K, x), (K', x'), (K'', x''), (K''', x''') \in \{0,1\}^{rn} \times \{0,1\}^n$ such that K, K', K'', K''' are pairwise distinct and satisfy the following system of equations:

$$\begin{aligned} K \oplus K' \oplus K'' \oplus K''' &= 0, \\ x \oplus x' \oplus x'' \oplus x''' &= 0, \\ y \oplus y' \oplus y'' \oplus y''' &= 0. \end{aligned}$$

where $y = \widetilde{\mathsf{P}}(K, x), y' = \widetilde{\mathsf{P}}(K', x'), y'' = \widetilde{\mathsf{P}}(K'', x''), y''' = \widetilde{\mathsf{P}}(K''', x''')$. Note that the first two equations occur with probability 1 according to the definition of the distinguisher. Consider the q' queries of $\mathcal{R}$ to $\widetilde{\mathsf{P}}$ sequentially, and denote by BAD the event where such values can be found among the q' queries. For any $i \in [1; q']$, let BAD_i be the event where such values can be found among the first i queries. Hence, by the union bound,

$$\Pr(\mathsf{BAD}) \leq \sum_{i=1}^{q'} \Pr\left(\mathsf{BAD}_i | \overline{\mathsf{BAD}_{i-1}}\right). \tag{4.23}$$

Let $i \in [1; q']$ and consider the i-th encryption query (a similar argument can made about decryption), $y_i = \widetilde{\mathsf{P}}(K_i, x_i)$. Assume without loss of generality that K_i is distinct from the previous keys used up to now (otherwise it does not help to find pairwise distinct keys in the system above). Hence, y_i is chosen uniformly at random from the set $\{0,1\}^n$. Therefore, BAD_i occurs only if y_i takes some value from a set of size at most $\binom{i-1}{3} \leq i^3$. Using (4.23), we obtain

$$\Pr(\mathsf{BAD}) \leq \sum_{i=1}^{q'} \Pr\left(\mathsf{BAD}_i | \overline{\mathsf{BAD}_{i-1}}\right) \leq \sum_{i=1}^{q'} \frac{i^3}{N} \leq \frac{q'^4}{2N}. \tag{4.24}$$

In conclusion,

$$\Pr\left(\mathcal{D}^{\widetilde{\mathsf{P}}, \mathsf{Sim}^{\widetilde{\mathsf{P}}}} = 1\right) \leq \Pr(\mathsf{BAD}) \leq \frac{q'^4}{2N}.$$

This concludes the proof. □

4.3.2 Proof of Sequential Indifferentiability of $(r+3)$-TEML

In this part, we prove Theorem 4.1.2. To accomplish this, we introduce an efficient simulator designed to replicate the behavior of the internal permutations utilized in TEML.

4.3.2.1 Informal Description of The Simulator

We start with an informal description of our simulator (for a formal pseudo-code see Algorithm 4.3.2). The simulator offers an interface $\mathsf{Query}(i, \delta, w)$ to the distinguisher for querying the internal permutations, where $i \in [1; r+3]$ indicates the index of the permutation, and $\delta \in \{+, -\}$ the direction of the query (direct or inverse). For

Algorithm 4.3.2 Formal Description of the Simulator, $\mathsf{Sim}(\mathbf{P})$

Variables:
  tables $(\Pi_i : i \in [1; r+3])$ initially empty.
procedure $\textsc{Query}(i, \delta, w)$
  if $(\delta, w) \notin \Pi_i$ **then**
    $w' = P_i(\delta, w)$
    $\Pi_i(\delta, w) := w', \Pi_i(\overline{\delta}, w') := w$ ▷ may overwrite an entry
    // complete $(v_2, \mathbf{a}, u_{r+2})$ chains if exists
    if $i = 2$ **then**
      if $\delta = +$ **then** $v_i := w'$ **else** $v_i := w$
      for all $\mathbf{a} \in A_r, u_{r+3} \in \Pi^+_{r+3}$ **do**
        CompleteChain $(v_2, \mathbf{a}, u_{r+2}, 1)$
      end for
    else if $i = r + 2$ **then**
      if $\delta = +$ **then** $u_i := w$ **else** $u_i := w'$
      for all $\mathbf{a} \in A_r, v_2 \in \Pi^-_2$ **do**
        CompleteChain $(v_2, \mathbf{a}, u_{r+2}, r + 3)$
      end for
    end if
  end if
  return $\Pi_i(\delta, w)$
end procedure
procedure ForceVal (u_i, v_i, i)
  $\Pi_i(+, u_i) := v_i$ ▷ may overwrite an entry
  $\Pi_i(-, v_i) := u_i$ ▷ may overwrite an entry
end procedure
procedure CompleteChain $(v_2, \mathbf{a}, u_{r+2}, \ell)$
  for $i \in [2; r + 1]$ **do**
    $k_i := v_i \oplus u_{i+1}$
  end for
  $K = \gamma_I^{-1}(k_2, \ldots, k_{r+1})$, $I = [2; r + 1]$

  case $\ell = 1$:
    // evaluate backwards to v_1
    $u_2 := \Pi_2(-, v_2)$
    $v_1 := u_2 \oplus \gamma_2(K)$
    // evaluate forwards to u_1
    $v_{r+2} := \Pi_{r+2}(+, u_{r+2})$
    $u_{r+3} = v_{r+2} \oplus \gamma_{r+2}(K)$
    $v_{r+3} =$ Query $(r + 3, +, u_{r+3})$
    $x := \widetilde{\mathsf{P}}(-, K, v_{r+3} \oplus \gamma_{r+3}(K))$
    $u_1 := x \oplus \gamma_0(K)$
    // adapt the chain
    ForceVal $(u_1, v_1, 1)$

  case $\ell = r + 3$:
    // evaluate forwards to u_{r+3}
    $v_{r+2} = \Pi_{r+2}(+, u_{r+2})$
    $u_{r+3} = v_{r+2} \oplus \gamma_{r+2}(K)$
    // evaluate backwards to v_{r+3}
    $u_2 := \Pi_2(-, v_2)$
    $v_1 := u_2 \oplus \gamma_1(K)$
    $u_1 :=$ Query $(1, -, v_1)$
    $y = \widetilde{\mathsf{P}}(+, K, u_1 \oplus \gamma_0(K))$
    $v_{r+3} = y \oplus \gamma_{r+3}(K)$
    // adapt the chain
    ForceVal $(u_{r+3}, v_{r+3}, r + 3)$

  end procedure

each $i \in [1; r+3]$ the simulator maintains (internally) a table Π_i mapping entries $(\delta, w) \in \{+,-\} \times \{0,1\}^n$ to a value $w' \in \{0,1\}^n$, initially undefined for all entries (defined by the symbol $\perp$). We denote by Π_i^+, respectively Π_i^-, the time dependent sets of strings $w \in \{0,1\}^n$ such that Π_i^+, respectively Π_i^-, is defined (not a $\perp$ symbol). When the simulator receives a query (i, δ, w), it looks in table Π_i to see whether the corresponding answer $\Pi_i(\delta, w)$ is already defined. When this is the case, it outputs the answer and waits for the next query. Otherwise, it randomly draws an answer $w' \in \{0,1\}^n$ and defines $\Pi_i(\delta, w) := w'$, as well as the opposite direction table entry, $\Pi_i(\overline{\delta}, w') := w$, where $\overline{\delta}$ defined as $-$ if δ is $+$, and $+$ otherwise. In order to handily describe how the answer w', we make the randomness used by the simulator explicit through a tuple of random permutations $\mathbf{P} = (P_1, \ldots, P_{r+3})$.

After this random choice of the answer w', and before returning it to the distinguisher, the simulator takes additional steps to ensure consistency with the ideal cipher by running a chain completion mechanism. For that we define the set of all intermediate value in the middle layer of the cipher. Formally, let A_r be the set of all r tuples, $\mathbf{a} = ((u_i, v_i) : i \in [3; r+1])$, where for each $i \in [3; r+1]$, we have that $\Pi_i^+(u_i) = v_i$ and $\Pi_i^-(v_i) = u_i$. Then, if the distinguisher called $\mathsf{Query}(i, \delta, w)$ for $i = 2$ or $i = r+2$, the simulator completes all newly created "chains" $(v_2, \mathbf{a}, u_{r+2})$ where $v_2 \in \Pi_2^-$, $u_{r+2} \in \Pi_{r+2}^+$ and $\mathbf{a} \in A_r$, by executing a procedure $\mathsf{CompleteChain}(v_2, \mathbf{a}, u_{r+}, \ell)$ where ℓ indicates at which endpoint the chain will be "adapted".

For example, assume that the distinguisher called $\mathsf{Query}(2, +, u_2)$ and that the answer randomly chosen by the simulator was u_2 (or backwards where the random value is u_2). Then for each $\mathbf{a} \in A_r$ and endpoints $u_{r+2} \in \Pi_{r+2}^-$, the simulator computes the corresponding round keys $k_i = v_i \oplus u_{i+1}$ for $i \in [2; r+1]$, and defines the master key $K = \gamma_{[2;r+1]}(k_2, \ldots, k_{r+1})$. The simulator then can adapt at round 1. Indeed, we can compute the value $v_1 = k_2 \oplus \Pi_2(-, v_2)$. Moreover, looking at the other endpoint of the cipher we can retrieve u_1 by applying the following steps:

$$
\begin{aligned}
&v_{r+2} = \Pi_{r+2}(+, u_{r+2}), u_{r+3} = v_{r+2} \oplus \gamma_{r+2}(K),\\
&v_{r+3} = \Pi_{r+3}(+, u_{r+3}), x = \widetilde{\mathsf{P}}(-, K, v_{r+3} \oplus \gamma_{r+3}(K)),\\
&u_1 = x \oplus \gamma_0(K),
\end{aligned}
$$

where v_{r+3} is drawn at random if it is not in Π_{r+3}^+. Now we can force the pair of input/output (u_1, v_1) to the table Π_1, in order to ensure consistency of the simulated TEML construction with $\widetilde{\mathsf{P}}$. For the case of $\mathsf{Query}(r+3, \cdot, \cdot)$ the behavior of the simulator will be symmetrical, namely adaptation of the chain takes place in Π_{r+3} instead.

4.3.2.2 Analysis of The Simulator

To prove Theorem 4.1.2, we show that Sim runs in polynomial time and makes only a polynomial number of queries. Subsequently, we show that the two systems $\Sigma_1 = (\widetilde{\mathsf{P}}, \mathsf{Sim}^{\widetilde{\mathsf{P}}})$ and $\Sigma_3 = (\mathtt{TEML}^{\mathbf{P}}, \mathbf{P})$ are sequentially indistinguishable, by using an intermediate system Σ_2. The first part of the proof is established by the following lemma.

Lemma 4.3.1. *Consider the execution of the simulator* $\mathsf{Sim}^{\widetilde{\mathsf{P}}}$ *which makes* q *queries in total. Then:*

1. *the size of* $\Pi_2, \ldots, \Pi_{r+2}$ *is at most* q*;*
2. *the size of* Π_1, Π_{r+3} *is that most* $q^{r+1} + q$*;*
3. *the simulator executes* CompleteChain *at most* q^{r+1} *times and makes at most* q^{r+1} *queries to* $\widetilde{\mathsf{P}}$*;*
4. *the total runtime of the simulator is* $O(q^{r+1})$*.*

Proof. Notice that for $i \in [2; r+2]$, the table Π_i can only increase in a call to the procedure Query(i, δ, w). Therefore the size of Π_i is bounded by the number of the distinguisher's queries q. CompleteChain is called once for at most every tuple of permutation queries, $((u_i, \Pi_i(+, u_i)) : i \in [2; r+2])$, hence at most q^{r+1} in total. Since CompleteChain makes at most one query to $\widetilde{\mathsf{P}}$, the simulator cannot make more than q^{r+1} queries to $\widetilde{\mathsf{P}}$. Note that tables Π_1 and Π_{r+3} are only increased by one for the calls to Query $(1, \delta, w)$ or Query $(5, \delta, w)$, which happens only once in the procedure CompleteChain , therefore the size of those tables is bounded by $q^{r+1} + q$. In conclusion, since CompleteChain runs in constant runtime, the total runtime of the simulator is $O(q^{r+1})$. □

We will denote by $\mathsf{Sim}(\widetilde{\mathsf{P}}, \mathbf{P})$ the simulator with oracle access to the ideal cipher $\widetilde{\mathsf{P}}$ and the randomness coming from $\mathbf{P}$. In order to prove the indistinguishability of the two systems $(\widetilde{\mathsf{P}}, \mathsf{Sim}(\widetilde{\mathsf{P}}, \mathbf{P}))$ and $(\mathtt{TEML}^{\mathbf{P}}, \mathbf{P})$, we will use an intermediate system $\Sigma_2 = (\mathtt{TEML}^{\mathsf{Sim}(\widetilde{\mathsf{P}}, \mathbf{P})}, \mathsf{Sim}(\widetilde{\mathsf{P}}, \mathbf{P}))$. In other words, the right oracle is the simulator $\mathsf{Sim}(\widetilde{\mathsf{P}}, \mathbf{P})$, with oracle access to an ideal cipher $\widetilde{\mathsf{P}}$ as in Σ_1, but now the left oracle is the $r+3$-round TEML construction with oracle access to $\mathsf{Sim}(\widetilde{\mathsf{P}}, \mathbf{P})$ instead of independent random permutations.

Transition from Σ_1 to Σ_2. The strategy behind the proof is to first define what constitutes a good pair $(\widetilde{\mathsf{P}}, \mathbf{P})$, and then show that the probability of transitioning from Σ_1 to Σ_2 is upper bounded by the probability of encountering a bad pair.

Definition 4.3.1. *A pair* $(\widetilde{\mathsf{P}}, \mathbf{P})$ *is said to be good if the simulator* Sim *never overwrites an entry of its tables* $(\Pi_i : i \in [1; r+3])$ *during an execution of* $\mathcal{D}^{\Sigma_2(\widetilde{\mathsf{P}}, \mathbf{P})}$*, otherwise the pair is called bad.*

Note that an overwrite may only happen during a random assignment in Line (6) or when adapting a chain in Lines (23) and (24). Moreover, whether a pair is good depends on the queries of the distinguisher $\mathcal{D}$. We first upper bound the probability that a random pair $(\widetilde{\mathsf{P}}, \mathbf{P})$ is bad.

Lemma 4.3.2. *Consider a distinguisher* $\mathcal{D}$ *with total oracle query cost at most* q*, with* $q^{r+1} \leq N/4$*. Then a uniformly random pair* $\widetilde{\mathsf{P}} \twoheadleftarrow_{\$} \widetilde{\mathit{Perm}}(rn; n)$ *and* $\mathbf{P} \in \mathit{Perm}(n)^{r+3}$ *is bad, with respect to* $\mathcal{D}$*, with probability at most* $16q^{2r+2}/N$*.*

Proof. First, note that the total number of queries received by the simulator in Σ_2 is exactly q. Since $(\Pi_2, \ldots, \Pi_{r+2})$ are never adapted, they can never be overwritten either.

Therefore, we only consider the probability of the tables Π_1 and Π_{r+3}. Let BadRand be the event that an overwrite occurs during a random assignment, at line (6), and BadAdapt be the event that an overwrite occurs when adapting a chain $(v_2, \mathbf{a}, u_{r+2})$ at lines (23) and (24). Distinguish between the two cases.

- *Probability of* BadRand*:* Let $i \in [1; r+3]$ and consider the assignments $\Pi_i(\delta, w) := w'$ and $\mathsf{P}\delta, w') := w$ where $w' := P_i(\delta, w)$ and P_i is some random permutation. By Lemma 4.3.1 (2), there are at most $q^{r+1} + q$ random assignments in Π_1 and Π_{r+3}, so that w' is sampled out of a set of size at least $N - q^{r+1} - q$. Moreover, this assignment cannot overwrite a value that was previously added during a random assignment, but only a value that was added by ForceVal, when adapting a chain, therefore by Lemma 4.3.1 (3) there are at most q^{r+1} such values. In conclusion, the probability w' hits a previously added value in table Π_i by a call to ForceVal is at most $\frac{q^{r+1}}{N-q^{r+1}-q}$. Summing over all possible random assignments in Π_1 and Π_{r+3}, we obtain the following upper bound,

$$\Pr(\mathsf{BadRand}) \leq 2\left(q^{r+1} + q\right) \cdot \frac{q^{r+1}}{N - q^{r+1} - q} \leq \frac{8q^{2r+2}}{N}.$$

- *Probability of* BadAdapt*:* Consider the probability of BadAdapt, conditioned on BadRand not occurring. Let $\mathsf{BadAdapt}_i$ be the event where a value is overwritten by the i-th call to ForceVal. We will be interested in the probability

$$\Pr\left(\mathsf{BadAdapt}_i \,\middle|\, \overline{\mathsf{BadRand}} \wedge \left(\bigwedge_{j=1}^{i-1} \overline{\mathsf{BadAdapt}_j}\right)\right).$$

  Consider the i-th execution of CompleteChain $(v_2, \mathbf{a}, u_{r+2}, \ell)$ and assume that no value was overwritten before this i-th call to CompleteChain. More precisely, consider the query Query $(j, \delta, \cdot)$ that was triggered during the chain completion and the call to ForceVal (u_ℓ, v_ℓ, ℓ). We must show that with high probability the entries of the tables $\Pi_\ell(+, u_\ell)$ and $\Pi_\ell(-, v_\ell)$ are undefined previously to this call. Distinguish between two main cases.

  - Assume $j = 2$, $\ell = 1$, (the case $i = r+2$, $\ell = r+3$ is symmetrical) and consider the value of v_1 given by $v_1 = u_2 \oplus \gamma_1(K)$ where $K = \gamma^{-1}_{[2;r+1]}(v_2 \oplus u_3, \ldots, v_{r+1} \oplus u_{r+2})$. Note that since $j = 2$, then either u_2 or v_2 is a random value. Consider when K is created and note that if v_2 is a random value and since the other values involved are fixed, then the value $\gamma_1(K)$ is a random variable that depends solely on the sampling of v_2. Therefore, v_1 takes a random value from a set of size at least $N - q$ (from u_2 or v_2). Hence, by Lemma 4.3.1 (2), the probability that v_1 takes a values from a defined value in table Π_1 is at most $\frac{q^{r+1}+q}{N-q}$.
  - Next, we show that simulator never made the query $\widetilde{\mathsf{P}}(-, K, v_{r+3} \oplus \gamma_{r+3}(K))$ before nor received the value from a previous query to $\widetilde{\mathsf{P}}(+, K, u_1 \oplus \gamma_0(K))$. Assume otherwise, then there exists a chain $(v_2', \mathbf{a}', u_{r+2}')$ such that the query

$\widetilde{\mathsf{P}}(-, K, v_{r+3} \oplus \gamma_{r+3}(K))$ appears during it's completion. Then, since both chains use the same master key (there is only one query to $\widetilde{\mathsf{P}}$ for each chain completion), then $v_{r+3} = v'_{r+3}$. Hence, $u_{r+3} = u'_{r+3}$, which implies that $v_{r+2} = v'_{r+2}$, since again they share the same master key. By going down with this recursive process we conclude that the chains are equal. Therefore, by Lemma 4.3.1 (3) and (2), the probability that $u_1 = x \oplus \gamma_0(K)$ hits one of the values in the table Π_1 is at most $\frac{q^{r+1}+q}{N-q^{r+1}}$, since that are at most q^{r+1} calls to $\widetilde{\mathsf{P}}$.

Summing over all at most q^{r+1} calls to $\mathsf{CompleteChain}$, we conclude that,

$$\begin{aligned}\Pr\left(\mathsf{BadAdapt} \mid \overline{\mathsf{BadRand}}\right) &\leq \sum_{i=1}^{q^{r+1}} \Pr\left(\mathsf{BadAdapt}_i \middle| \overline{\mathsf{BadRand}} \wedge \left(\bigwedge_{j=1}^{i-1} \overline{\mathsf{BadAdapt}_j}\right)\right) \\ &\leq q^{r+1}\left(\frac{q^{r+1}+q}{N-q} + \frac{q^{r+1}+q}{N-q^{r+1}}\right) \leq \frac{8q^{2r+2}}{N}.\end{aligned}$$

Combining both upper bounds yields the result. □

Finally, the following lemma establishes the probability of transitioning from Σ_1 to Σ_2.

Lemma 4.3.3. *For any distinguisher $\mathcal{D}$ of total oracle query cost at most q, one has,*

$$\left|\Pr\left(\mathcal{D}^{\Sigma_1(\widetilde{\mathsf{P}},\mathbf{P})} = 1\right) - \Pr\left(\mathcal{D}^{\Sigma_2(\widetilde{\mathsf{P}},\mathbf{P})} = 1\right)\right| \leq \frac{16q^{2r+2}}{N}.$$

where both probabilities are taken over $\widetilde{\mathsf{P}} \twoheadleftarrow_{\$} \widetilde{\mathsf{Perm}}(rn; n)$, $\mathbf{P} \twoheadleftarrow_{\$} \mathsf{Perm}(n)^{r+3}$.

Proof. We show that for any good pair $(\widetilde{\mathsf{P}}, \mathbf{P})$, the transcript of the interaction of $\mathcal{D}$ with $\Sigma_1(\widetilde{\mathsf{P}}, \mathbf{P})$ and $\Sigma_2(\widetilde{\mathsf{P}}, \mathbf{P})$ is identical. Since the distinguisher is sequential and they both share the same right oracle it is clear it is the same interaction during the first phase. For the second phase of the interaction, since the simulator never overwrites the tables Π_i for $i \in [1; r+3]$, it follows that, for any $\delta \in \{+,-\}$ and $z \in \{0,1\}^n$, $\mathtt{TEML}^{\mathsf{Sim}(\widetilde{\mathsf{P}},\mathbf{P})}(\delta, K, z) = \widetilde{\mathsf{P}}(\delta, K, z)$. Therefore, the interaction of $\mathcal{D}$ with $\Sigma_1(\widetilde{\mathsf{P}}, \mathbf{P})$ and $\Sigma_2(\widetilde{\mathsf{P}}, \mathbf{P})$ is identical in both phases. Hence,

$$\left|\Pr\left(\mathcal{D}^{\Sigma_1(\widetilde{\mathsf{P}},\mathbf{P})} = 1\right) - \Pr\left(\mathcal{D}^{\Sigma_2(\widetilde{\mathsf{P}},\mathbf{P})} = 1\right)\right| \leq \Pr\left((\widetilde{\mathsf{P}}, \mathbf{P}) \text{ is bad}\right),$$

from which the result follows by Lemma 4.3.2. □

Transition From Σ_2 to Σ_3 and Randomness Mappings. Next, we consider the transition from Σ_2 to Σ_3, using the randomness mapping argument introduced in [92]. To achieve this, we will introduce the concept of a partial permutation.

Definition 4.3.2. *A partial permutation is a function $P'_i : \{+,-\} \times \{0,1\}^n \to \{0,1\}^n \cup \{\perp\}$ such that for all $u, v \in \{0,1\}^n$, $P'_i(+, u) = v \neq \perp \Leftrightarrow P'_i(-, v) = u$.*

We define a map $\mathcal{A}$ mapping pairs $(\widetilde{\mathsf{P}}, \mathbf{P})$ either to the special symbol $\perp$ when $(\widetilde{\mathsf{P}}, \mathbf{P})$ is bad, or to a tuple of partial permutations as follows: run $\mathscr{D}^{\Sigma_2(\widetilde{\mathsf{P}},\mathbf{P})}$, and consider the

tables ($\Pi_i : i \in [1; r+3]$) at the end of the simulation, then fill all undefined entries of the tables with the special symbol $\perp$. Then define $\mathcal{A}(\widetilde{\mathsf{P}}, \mathbf{P}) = (\Pi_1, \ldots, \Pi_{r+3})$. Note that since $(\widetilde{\mathsf{P}}, \mathbf{P})$ is a good pair, the simulator never overwrites an entry in its tables, which implies that $\mathcal{A}(\widetilde{\mathsf{P}}, \mathbf{P})$ is indeed a partial permutation and $\mathcal{A}$ is well defined.

Good Partial Permutations: We say a tuple of partial permutations $\mathbf{P}' = (P'_1, \ldots, P'_{r+3})$ is good if there exists an ideal cipher $\widetilde{\mathsf{P}}$ and a tuple of permutations $\mathbf{P} = (P_1, \ldots, P_{r+3})$, such that $\mathcal{A}(\widetilde{\mathsf{P}}, \mathbf{P}) = \mathbf{P}'$. We say that a tuple of permutations $\mathbf{P} = (P_1, \ldots, P_{r+3})$ extends a tuple of partial permutations $\mathbf{P}' = (P'_1, \ldots, P'_{r+3})$, denoted by $\mathbf{P} \vdash \mathbf{P}'$, if for any $i \in [1; r+3]$, P_i and P'_i agree on all entries that are already defined in P'_i (where $P'_i(\delta, w) \neq \perp$). The following establishes the probabaility to transition from Σ_2 to Σ_3.

Lemma 4.3.4. *For any distinguisher $\mathcal{D}$ of total oracle query cost at most q, one has,*

$$\left|\Pr\left(\mathcal{D}^{\Sigma_2(\widetilde{\mathsf{P}},\mathbf{P})} = 1\right) - \Pr\left(\mathcal{D}^{\Sigma_3(\mathbf{P})} = 1\right)\right| \leq \frac{\left((r+5)^2 + 16\right) q^{2r+2}}{N}.$$

where the first probability is take over $\widetilde{\mathsf{P}} \twoheadleftarrow_{\$} \widetilde{\mathsf{Perm}}(rn; n)$, $\mathbf{P} \twoheadleftarrow_{\$} \mathsf{Perm}(n)^{r+3}$, and the second only over $\mathbf{P} \twoheadleftarrow_{\$} \mathsf{Perm}(n)^{r+3}$.

Proof. Let

$$\varepsilon := \left|\Pr\left(\mathcal{D}^{\Sigma_2(\widetilde{\mathsf{P}},\mathbf{P})} = 1\right) - \Pr\left(\mathcal{D}^{\Sigma_3(\mathbf{P})} = 1\right)\right|$$

and assume without loss of generality that $\Pr\left(\mathcal{D}^{\Sigma_2(\widetilde{\mathsf{P}},\mathbf{P})} = 1\right) \geq \Pr\left(\mathcal{D}^{\Sigma_3(\mathbf{P})} = 1\right)$. By the definition of the map $\mathcal{A}$, for any good tuple of partial permutations $\mathbf{P}'$, the outputs of $\mathcal{D}^{\Sigma_2(\widetilde{\mathsf{P}},\mathbf{P})}$ and $\mathcal{D}^{\Sigma_3(\mathbf{P})}$ are equal for any pair $(\widetilde{\mathsf{P}}, \mathbf{P})$ such that $\mathcal{A}(\widetilde{\mathsf{P}}, \mathbf{P}) = \mathbf{P}'$, and any tuple of permutations $\mathbf{P}$ such that $\mathbf{P} \vdash \mathbf{P}'$. Let Θ_1 be the set of tuples of partial permutations $\mathbf{P}'$ such that $\mathcal{D}^{\Sigma_2(\widetilde{\mathsf{P}},\mathbf{P})}$ outputs 1 for any pair $(\widetilde{\mathsf{P}}, \mathbf{P})$ such that $\mathcal{A}(\widetilde{\mathsf{P}}, \mathbf{P}) = \mathbf{P}'$. Then, we can conclude that,

$$\varepsilon \leq \Pr\left((\widetilde{\mathsf{P}}, \mathbf{P}) \text{ is bad}\right) + \sum_{\mathbf{P}' \in \Theta_1} \left(\Pr\left(\mathcal{A}(\widetilde{\mathsf{P}}, \mathbf{P}) = \mathbf{P}'\right) - \Pr\left(\mathbf{P} \vdash \mathbf{P}'\right)\right). \tag{4.25}$$

Fix any good tuple of partial permutations $\mathbf{P}'$ and for any $i \in [1; r+3]$ let

$$|P'_i| = |\{u \in \{0,1\}^n : P'_i(+, u) \neq \perp\}| = |\{v \in \{0,1\}^n : P'_i(-, v) \neq \perp\}|.$$

Then by definition of a partial permutation, one has,

$$\Pr\left(\mathbf{P} \twoheadleftarrow_{\$} \mathsf{Perm}(n)^{r+3} : \mathbf{P} \vdash \mathbf{P}'\right) = \frac{1}{\prod_{i=1}^{r+3} (N)_{|P'_i|}}.$$

Fix now any good pre-image $(\widetilde{\mathsf{P}}, \mathbf{P})$ of $\mathbf{P}'$, where $\mathbf{P} = (P_1, \ldots, P_{r+3})$ and let q_e and let $(q_i : i \in [1; r+3])$ be the number of queries made by the simulator to $\widetilde{\mathsf{P}}$ and P_i respectively in the execution of $\mathcal{D}^{\Sigma_2(\widetilde{\mathsf{P}},\mathbf{P})}$. Then we conclude that,

$$\Pr\left(\widetilde{\mathsf{P}} \twoheadleftarrow_{\$} \widetilde{\mathsf{Perm}}(rn; n), \mathbf{P} \twoheadleftarrow_{\$} \mathsf{Perm}(n)^{r+3} : \mathcal{A}(\widetilde{\mathsf{P}}, \mathbf{P}) = \mathbf{P}'\right) \leq \frac{1}{(N)_{q_e} \prod_{i=1}^{r+3} (N)_{q_i}},$$

since the probability is maximized when the same master key is used for all q_e queries to $\widetilde{\mathsf{P}}$. Moreover, since the number of executions of ForceVal made by the simulator, i.e., the number of chain adaptions, is exactly the number of queries made by the simulator to $\widetilde{\mathsf{P}}$, one has,

$$q_e + \sum_{i=1}^{r+3} q_i = \sum_{i=1}^{r+3} |P'_i| \leq 2q^{r+1} + (r+3)q. \tag{4.26}$$

where the last inequality follows by Lemma 4.3.1 (1) and (2). In conclusion, by (4.26) we have that,

$$\begin{aligned}
\frac{\Pr\left(\mathbf{P} \vdash \mathbf{P}'\right)}{\Pr\left(\mathcal{A}(\widetilde{\mathsf{P}}, \mathbf{P}) = \mathbf{P}'\right)} &= \frac{(N)_{q_e} \prod_{i=1}^{r+3} (N)_{q_i}}{\prod_{i=1}^{r+3} (N)_{|P'_i|}} \\
&\geq \frac{N^{q_e + \sum_{i=1}^{r+3} q_i}}{N^{\sum_{i=1}^{r+3} |P'_i|}} \prod_{j=1}^{q_e - 1} \left(1 - \frac{j}{N}\right) \prod_{i=1}^{r+3} \prod_{j=1}^{q_i - 1} \left(1 - \frac{j}{N}\right) \\
&\geq 1 - \frac{q_e^2 + \sum_{i=1}^{r+3} q_i^2}{N} \geq 1 - \frac{(2q^{r+1} + (r+3)q)^2}{N} \\
&\geq 1 - \frac{(r+5)^2 q^{2r+2}}{N}.
\end{aligned}$$

Combining (4.25) with (4.26), we obtain,

$$\begin{aligned}
\varepsilon &\leq \Pr\left((\widetilde{\mathsf{P}}, \mathbf{P}) \text{ is bad}\right) + \sum_{\mathbf{P}' \in \Theta_1} \Pr\left(\mathcal{A}(\widetilde{\mathsf{P}}, \mathbf{P}) = \mathbf{P}'\right) \left(\frac{\Pr\left(\mathbf{P} \vdash \mathbf{P}'\right)}{\Pr\left(\mathcal{A}(\widetilde{\mathsf{P}}, \mathbf{P}) = \mathbf{P}'\right)} \right) \\
&\leq \Pr\left((\widetilde{\mathsf{P}}, \mathbf{P}) \text{ is bad}\right) + \frac{(r+5)^2 q^{2r+2}}{N} \sum_{\mathbf{P}' \in \Theta_1} \Pr\left(\mathcal{A}(\widetilde{\mathsf{P}}, \mathbf{P}) = \mathbf{P}'\right) \\
&\leq \Pr\left((\widetilde{\mathsf{P}}, \mathbf{P}) \text{ is bad}\right) + \frac{(r+5)^2 q^{2r+2}}{N} \leq \frac{(r+5)^2 q^{2r+2}}{N} + \frac{16 q^{2r+2}}{N} \\
&= \frac{\left((r+5)^2 + 16\right) q^{2r+2}}{N}
\end{aligned}$$

□

Theorem 4.1.2 then follows from Lemma 4.3.3 and Lemma 4.3.4.

Part II

Analysis of Authenticated Encryption Schemes

5

Subverting Telegram's End-to-End Encryption

In this chapter, we analyze the popular messaging app Telegram and its custom-built security protocol, `MTProto`. With over 900 million users, Telegram has become a crucial platform for independent journalists and political dissidents, emphasizing its independence from government censorship. Telegram offers two types of chat modes: cloud chats, which use client-server encryption and store messages on Telegram's servers, and secret chats, which use end-to-end encryption and ensure that messages are only readable by the communicating parties. The security of `MTProto`, particularly in its latest version, `MTProto2.0`, has been a topic of debate, with various attacks revealing vulnerabilities in its design, despite the Telegram team's claims of achieving IND-CCA security. See Section 1.6.2 for a complete overview on this chapter.

The chapter further explores subversion attacks, particularly Algorithm Substitution Attacks (ASAs), which have gained attention following the Snowden revelations. These attacks involve tampering with cryptographic algorithms to leak secret keys, even when outputs appear normal. While Telegram's open-source nature makes mass subversion challenging, targeted attacks, especially on third-party clients, remain feasible. The chapter's contributions include proposing the first partial key recovery ASAs on Telegram's secret chat mode and suggesting modifications to `MTProto2.0` to enhance its resistance to such attacks. The proposed changes ensure the protocol's security while maintaining its efficiency, providing a more robust solution against potential subversion efforts.

5.1 Presentation of the Full `MTProto2.0` Protocol

Telegram clients rely on the `MTProto` protocol to secure communications. A message that is typed by a user, or any application-defined message, first has to be *enriched* to include additional information, along with a padding and some random bits; we refer to these as protocol-enriched messages. Then, these plaintexts are encrypted using a DAE scheme that is also dubbed `MTProto`.

We start with the full description of the MTProto2.0 protocol. Telegram offers two types of protocols: the client-server communication protocol for cloud chat mode and the end-to-end communication protocol for secret chat mode.

5.1.1 Client-Server Encrypted Communication

Similar to the first version of MTProto [307], the communication begins with a Diffie-Hellman key exchange [115] (a famous key exchange protocol), which is used to generate a shared key between the sender and the receiver. After the key exchange, the sender and the receiver share a 2048-bit symmetric key denoted by K and an additional key fingerprint f defined as the last 64 bits of SHA-1 on K. This fingerprint is used as a sanity check for the key exchange procedure to detect bugs in the software implementation. Moreover, in order to keep past communications safe, the secret key is regenerated once a key has been used for more than 100 messages or more than a week. Next, we define the *protocol enriched message* X as (see also Figure 5.1):

$$\begin{aligned} \texttt{X} := \textbf{salt}\|\textbf{session\_id}\|\textbf{message\_id}\|\textbf{seq\_no}\|\textbf{message\_data\_length} \\ \|\textbf{message\_data}, \end{aligned} \tag{5.1}$$

where

- **salt**(64-bit) - Changed every 30 minutes (separately for each session) at the request of the server. All subsequent messages must contain the new salt (although, messages with the old salt are still accepted for a further 30 minutes). Required to protect against replay attacks and certain tricks associated with adjusting the client clock to a moment in the distant future.
- **session_id**(64-bit) - Generated by the client to distinguish between individual sessions (for example, between different instances of the application, created with the same authorization key). The session in conjunction with the key identifier corresponds to an application instance.
- **message_id**(64-bit) - Time-dependent number used uniquely to identify a message within a session. Client message identifiers are divisible by 4, server message identifiers modulo 4 yield 1 if the message is a response to a client message, and 3 otherwise. Client message identifiers must increase monotonically (within a single session), the same as server message identifiers, and must approximately equal $unixtime * 2^{32}$. This way, a message identifier points to the approximate moment in time the message was created. A message is rejected over 300 seconds after it is created or 30 seconds before it is created (this is needed to protect from replay attacks). In this situation, it must be re-sent with a different identifier (or placed in a container with a higher identifier). The identifier of a message container must be strictly greater than those of its nested messages.
- **seq_no**(32-bit) - Equal to twice the number of "content-related" messages (those requiring acknowledgment, and in particular those that are not containers) created

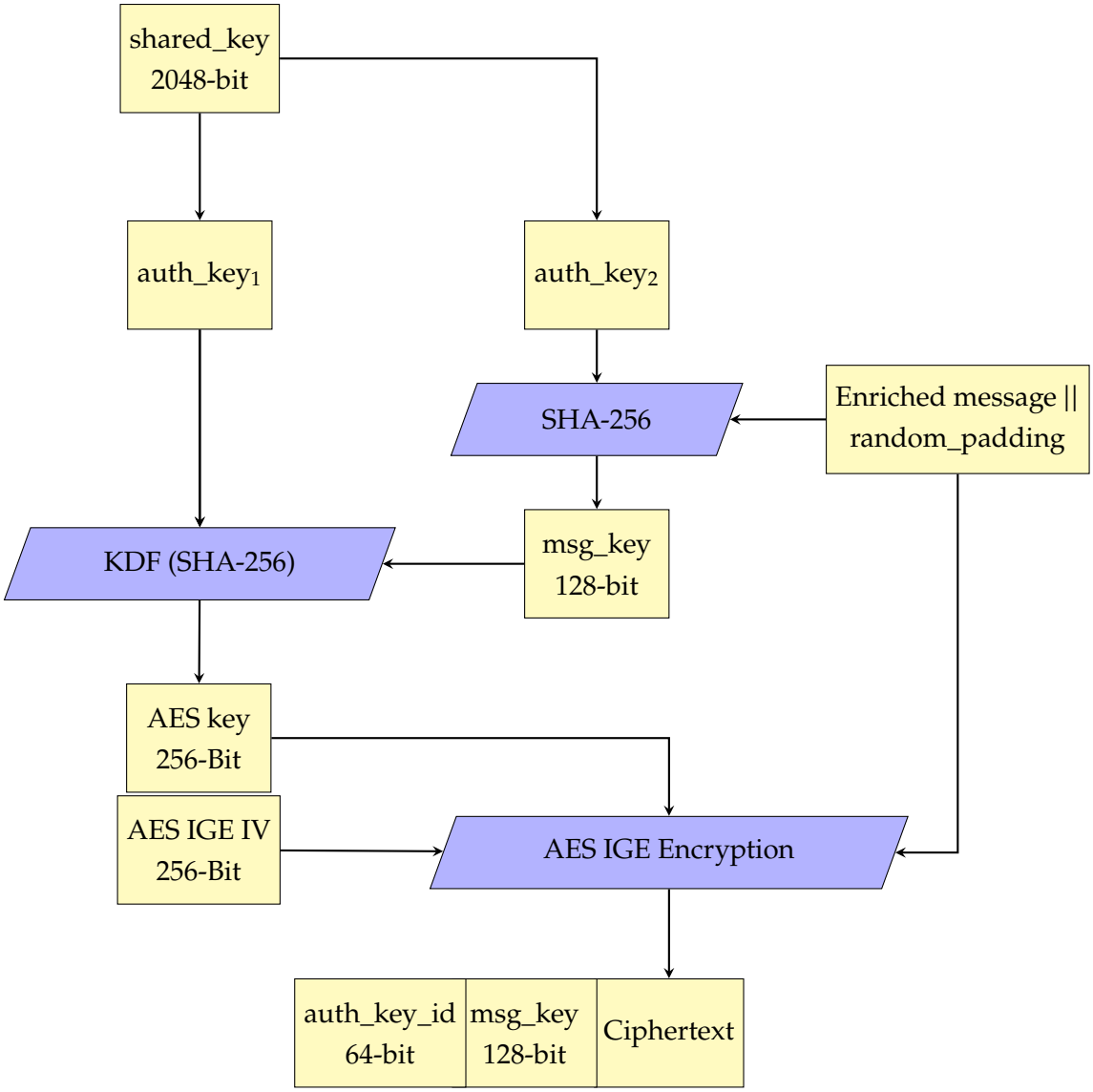

Figure 5.1: *`MTProto2.0`: client-server encryption*

by the sender prior to this message and subsequently incremented by one if the current message is a content-related message. A container is always generated after its entire contents; therefore, its sequence number is greater than or equal to the sequence numbers of the messages contained in it.

- **message_data_length**(32-bit).
- **message_data**.

In addition, we also have the random padding value, **random_padding**, that consists of 12 to 1024 bytes to make its length divisible by 16 bytes (the sampling is described in detail in Section 5.1.3). Finally, we define the *fully encoded message* for encryption $\mathtt{X}'$ as

$$\mathtt{X}' := \mathtt{X} \| \textbf{random\_padding} \tag{5.2}$$

Let y be a chat parameter defined as $y = 0$ for messages from client to server and $y = 8$ otherwise. We define the authentication tag t, first let $k_1 = K[88 + y, \ldots, 119 + y]$

be some middle bytes of the shared key then we define $t = \mathsf{F}_{k_1}(\mathtt{X}')$. Notice that t is also used for deriving keys in the encryption of E. In order to use the encryption scheme E, we first generate the key and iv for the encryption scheme. For that let $k_2 = K[y, \ldots, 35 + y], k_3 = K[40 + y, \ldots, 75 + y]$ be some part of the shared key then for $k = (k_2, k_3)$ we have that $\mathsf{G}_k(t) = (l, \mathrm{iv})$.

Finally, the encryption scheme (uses AES-256 with IGE mode) is defined as, $\mathsf{E}^{+}_{l,\mathrm{iv}}(\mathtt{X}') = c$, where the final cipher-text returned is defined as the string, $(f||t||c)$.

5.1.2 End to End Encrypted Communication Protocol

Encoding of a Message. The encoding of a message is almost identical to the one in the client-server encryption described above. The difference lies in the definition of protocol enriched message X (see Eq. (5.1)) and the chat parameter (y is equal to 0 if the current user is the chat creator and otherwise $y = 8$). In this setting, X is defined as

$$\begin{aligned}\mathtt{X} := \mathbf{length}||\mathbf{payload\_type}||\mathbf{random\_bytes}||\mathbf{layer}||\mathbf{in\_seq\_no} \\ ||\mathbf{out\_seq\_no}||\mathbf{message\_type}||\mathbf{message\_data},\end{aligned} \tag{5.3}$$

The auxiliary information can be summarized using the following fields (see also Figure 5.2).

- **length** (32-bit) - Length of the payload.
- **payload_type** (32-bit).
- **random_bytes** ($\geq$ 128-bit) - Set of random bytes to prevent content recognition in short encrypted messages. Clients are required to check that there are at least 15 random bytes included in each message. Messages with less than 15 random bytes must be ignored.
- **layer** (32-bit) - Layer number.
- **in_seq_no** (32-bit) - Twice the number of messages in the sender's inbox (including deleted and service messages), incremented by 1 if current user was not the chat creator.
- **out_seq_no** (32-bit) - Twice the number of messages in the recipient's inbox (including deleted and service messages), incremented by 1 if current user was the chat creator.
- **message_type** (32-bit).
- **message_data**.

Note that, the sequence numbers are especially important for our subversion attack presented in Section 5.4. Finally, the fully encoded message X′ in secret chat setting is generated by appending a random padding in exactly the same fashion (see Equation (5.2)) as in the client-server chat. Besides, some Telegram clients do check that the random padding is at least 12 bytes long (notably the iOS client).

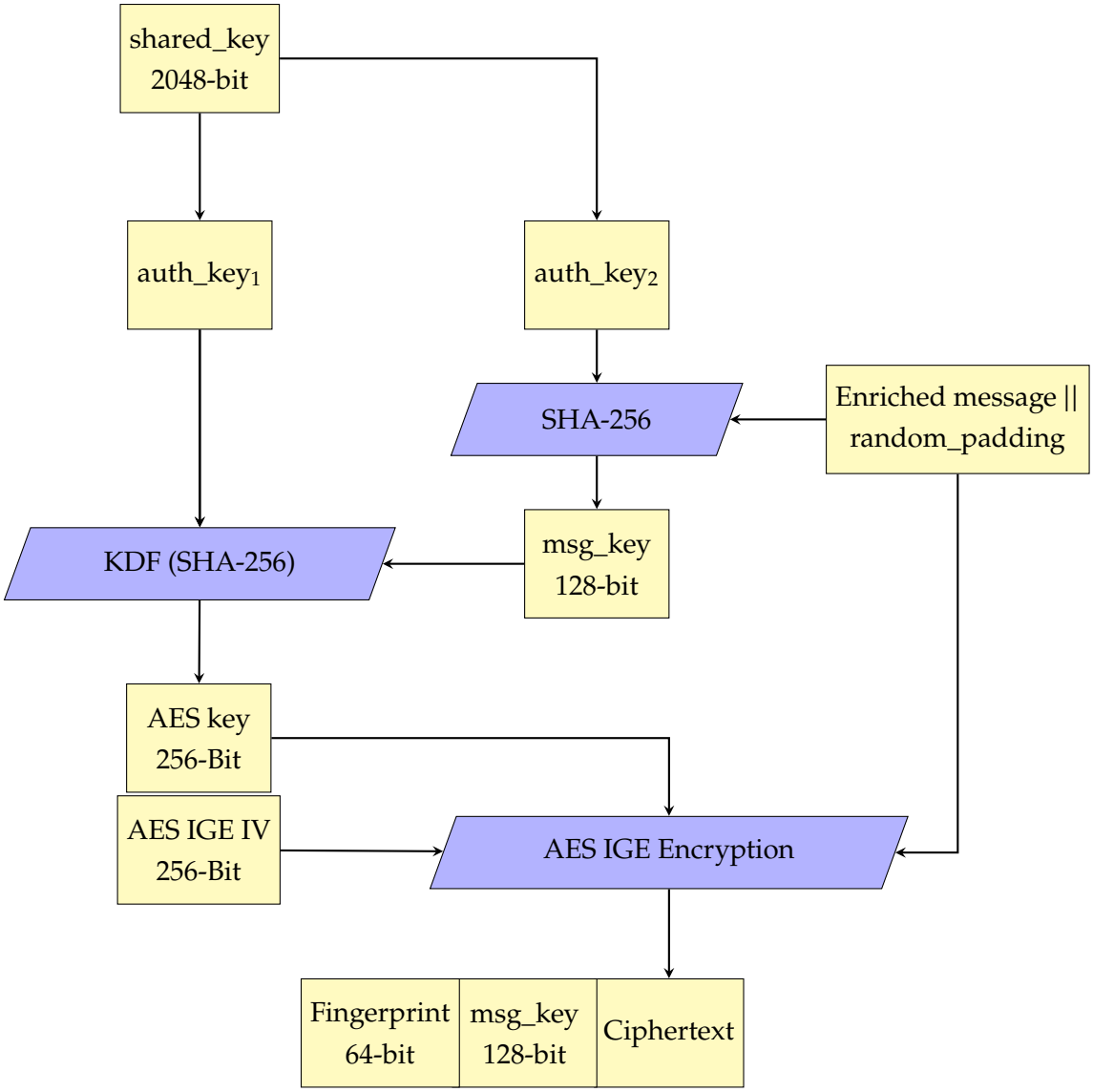

Figure 5.2: *`MTProto2.0`: end-to-end encryption*

5.1.3 Sampling of a Random Padding

The padding algorithm of `MTProto2.0` is uncommon, and deserves to be described in more details. Indeed, the padding is filled with random bits, and its length is also chosen at random[1]. It is worth noting that each official client seems to use a different algorithm to randomize padding length. Since our goal is to focus on building a hypothetical malicious client, we will present the two length randomization algorithms that are best suited to our attack. We start with the algorithm of the desktop client. Let us assume that we want to encrypt a σ-byte message M. Let us write $\sigma = 16q + 4r + s$, where $0 \leq r < 4$, and $0 \leq s < 4$. The padded message will consist $\sigma' = 4q + r + f(r) + 4\mathbf{Rand}(0, 15)$ 32-bit blocks, where $f(0) = 4$, $f(1) = 3$ $f(2) = 6$, $f(3) = 5$, and $\mathbf{Rand}(0, 15)$ denotes an integer

[1] Although it seems optional in the source code of Telegram, we focus on the randomized-length padding scheme, as the official documentation presents this one.

that is chosen uniformly at random between 0 and 15. In particular, this means that, after padding, the length of the plaintext in 32-bit blocks will be

$$\sigma' = 4\left(q + \lfloor r/2 \rfloor + 1 + \mathbf{Rand}(0, 15)\right).$$

Let us define the function $g : \mathbb{N} \to \mathbb{N}$ as the mapping

$$16q + 4r + s \longmapsto q + \lfloor r/2 \rfloor + 1.$$

Then, one clearly has $\sigma' = 4(g(\sigma) + \mathbf{Rand}(0, 15))$, and thus the size of the padded data in 128-bit blocks will be $\sigma_{128} = g(\sigma) + \mathbf{Rand}(0, 15)$. In particular, it means that $\sigma_{128} \bmod 16$ is uniformly random. An equivalent way of sampling this value would be to generate a random integer $v \in \{0, \ldots, 15\}$, to write $g(\sigma) = 16q' + r'$, and then to choose $\sigma_{128} = 16q' + v$ if $v \geq r'$, or $\sigma_{128} = 16(q' + 1) + v$ otherwise. This second sampling mechanism will prove useful in the following section. We will denote this alternative padding rule $\mathsf{pad}(M, v)$, where M is padded to a message whose length in 16-byte blocks is equal to v modulo 16.

Similarly, we discuss the length randomization algorithm from the `tdlib` library, that can be used to develop third-party clients. Let σ be the byte-length of the message to be encrypted. The length of the padded message will be

$$\sigma' = 16 \times \left\lfloor \frac{\sigma + 27 + \mathbf{Rand}(0, 255)}{16} \right\rfloor.$$

Like in the previous case, $\sigma'/16 \mod 16$ will be uniformly random, and we can similarly define a reverse padding mechanism with the exact same probability distribution, where we generate the target $\sigma'/16 \mod 16$ value ℓ uniformly at random, and then the padded message length as

$$\sigma' = 16 \times \left\lfloor \frac{\sigma + 27 + (\ell \times 16 + \mathbf{Rand}(0, 15) - \sigma - 27 \mod 256)}{16} \right\rfloor.$$

5.2 Abstraction of The MTProto Protocol

This section focuses on the DAE scheme corresponding to the `MTProto` protocol and provides concrete security guarantees for that scheme.

5.2.1 Generic View of MTProto: MTProto-G

`MTProto` can be viewed as a somewhat generic deterministic authenticated encryption (DAE) scheme, referred here as `MTProto-G`, that utilizes two independently keyed functions, $\mathsf{F} : \mathcal{K}_1 \times \{0,1\}^* \to \{0,1\}^\tau$ and $\mathsf{G} : \mathcal{K}_2 \times \{0,1\}^\tau \to \{0,1\}^{\kappa+n}$, and a $(\{0,1\}^\kappa, \{0,1\}^n, \mathcal{M})$-encryption scheme E. As illustrated in Figure 5.3, the output of F serves two purposes:

1. obviously it acts as the authentication tag;

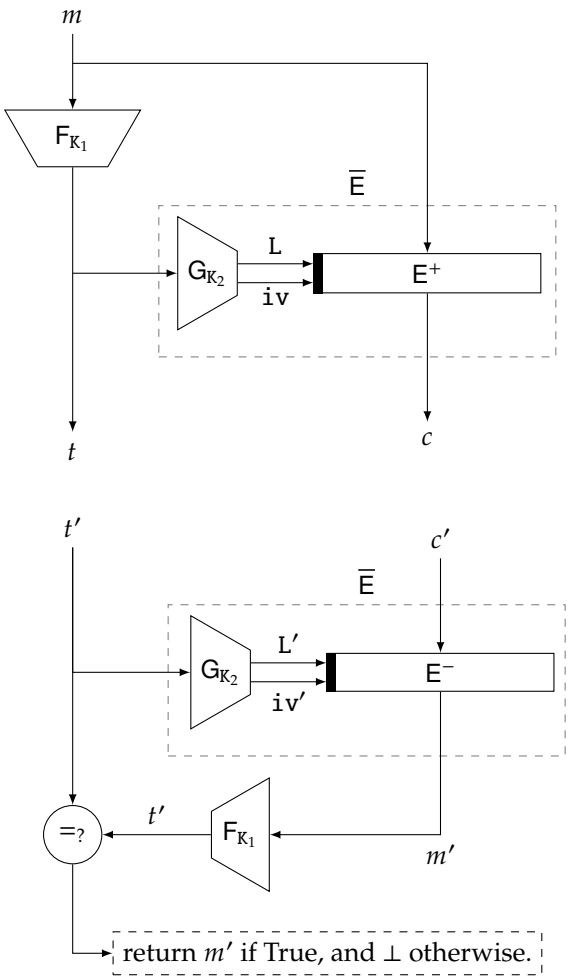

Figure 5.3: *Encryption (**top**) and decryption (**bottom**) algorithms in* `MTProto-G`*. The dashed rectangle represents the* iv*-based encryption scheme* $\overline{\mathsf{E}}$*.*

2. it acts as the input for deriving the keys and initialization values for the encryption scheme E.

The following lemma reduces the security of MTProto-G, to the security of it's main components: F, G and E as described above.

Lemma 5.2.1. *For $\mu, q_{\max}, q, \ell, \sigma, t > 0$, let $\mathscr{A}$ be a $(\mu, q_{\max}, q, \ell, \sigma, t)$-distinguisher against MTProto-G that runs in time at most t, and issues at most q queries, of length at most ℓ n-bit blocks, for a total queries length μ, over at most μ users, and such that each user is queried at most $q_{\max}$. Then, there exist $(\mu, q_{\max}, q, \ell, \sigma, t')$, $(\mu, q_{\max}, q, \hat{t})$, and $(q, 1, q, \ell, \sigma, \dot{t})$ distinguishers $\mathscr{B}$, $\mathscr{C}$, and $\mathscr{D}$, respectively, such that*

$$\mathbf{Adv}^{\mathsf{dae}}_{\mathit{MTProto\text{-}G}}(\mathscr{A}) \le \mathbf{Adv}^{\mathrm{mu-prf}}_{\mathsf{F}}(\mathscr{B}) + \mathbf{Adv}^{\mathrm{mu-wprf}}_{\mathsf{G}}(\mathscr{C}) + \mathbf{Adv}^{\mathsf{priv\$}}_{\mathsf{E}}(\mathscr{D}) + \frac{q}{2^{\tau}} + \frac{q^2}{2^{\tau}}, \quad (5.4)$$

where $t' = O(t + qt_{\mathsf{F}})$, $\hat{t} = O(t + qt_{\mathsf{G}})$, and $\dot{t} = O(t + qt_{\mathsf{E}})$.

Proof. First of all, we view MTProto-G as an instance of the SIV paradigm [280], where F is used to generate the tag (also acts as the synthetic IV), and the combination of G and E is viewed as an IV-based encryption scheme. More formally, we define a $(\mathcal{K}, \{0,1\}^{\tau}, \mathcal{M})$-encryption scheme $\overline{\mathsf{E}}$ (see Figure 5.3) as follows: for all $k, t, m \in \mathcal{K} \times \{0,1\}^{\tau} \times \mathcal{M}$, we have

$$(l, \mathrm{iv}) := \mathsf{G}_k(t), \qquad c := \mathsf{E}^{+}_{l,\mathrm{iv}}(m), \qquad \overline{\mathsf{E}}[\mathsf{G},\mathsf{E}]^{+}_{k,t}(m) := c$$

Then, using the SIV composition result by Rogaway and Shrimpton [280, Theorem 2], we have

$$\mathbf{Adv}^{\mathsf{dae}}_{\mathtt{MTProto\text{-}G}}(\mathscr{A}) \le \mathbf{Adv}^{\mathrm{mu-prf}}_{\mathsf{F}}(\mathscr{B}) + \mathbf{Adv}^{\mathsf{priv\$}}_{\mathsf{E}}(\mathscr{A}') + \frac{q}{2^{\tau}},$$

where $\mathscr{A}'$ is a $(\mu, q_{\max}, q, \ell, \sigma, t'')$-distinguisher for $t'' = O(t + qt_{\overline{\mathsf{E}}})$. Note that, the generic reduction result in [280] is proved in single-key setting. However, exactly the same approach generalizes to the multi-user setting as well. Next, by definition, one has

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{priv\$}}_{\mathsf{E}}(\mathscr{A}') &= \left| \Pr_{(\mathtt{K}_i)_{i\in[1;\mu]} \xleftarrow{\$} \mathcal{K}} \left(\mathscr{A}'^{(\overline{\mathsf{E}}[\mathsf{G}_{\mathtt{K}_i},\mathsf{E}]^{+})_{i\in[1;\mu]}} = 1 \right) - \Pr_{\$} \left(\mathscr{A}'^{\$} = 1 \right) \right| \\
&\le \left| \Pr_{(\mathtt{K}_i)_{i\in[1;\mu]}} \left(\mathscr{A}'^{(\overline{\mathsf{E}}[\mathsf{G}_{\mathtt{K}_i},\mathsf{E}]^{+})_{i\in[1;\mu]}} = 1 \right) \right. \\
&\quad \left. - \Pr_{(\Gamma_i)_{i\in[1;\mu]} \xleftarrow{\$} \mathsf{Func}(\tau;\kappa+n)} \left(\mathscr{A}'^{(\overline{\mathsf{E}}[\Gamma_i,\mathsf{E}]^{+})_{i\in[1;\mu]}} = 1 \right) \right| \\
&\quad + \left| \Pr_{(\Gamma_i)_{i\in[1;\mu]} \xleftarrow{\$} \mathsf{Func}(\tau;\kappa+n)} \left(\mathscr{A}'^{(\overline{\mathsf{E}}[\Gamma_i,\mathsf{E}]^{+})_{i\in[1;\mu]}} = 1 \right) - \Pr_{\$} \left(\mathscr{A}'^{\$} = 1 \right) \right|, \qquad (5.5)
\end{aligned}$$

where \$ denotes the uniform distribution sampler from the corresponding set.

Now, all that remains is to show that there exists a $(\mu, q_{\max}, q, \hat{t})$-distinguisher $\mathscr{C}$ and a $(q, 1, q, \ell, \sigma, \dot{t})$-distinguisher $\mathscr{D}$ such that the first absolute difference on the right hand side is bounded by $\mathbf{Adv}^{\mathrm{mu-wprf}}_{\mathsf{G}}(\mathscr{C})$ and the second difference is bounded by $\mathbf{Adv}^{\mathsf{priv\$}}_{\mathsf{E}}(\mathscr{D}) + q^2/2^{\tau}$.

Constructing a Distinguisher $\mathscr{C}$: We construct the distinguisher $\mathscr{C}$, which is trying to distinguish between $(\mathsf{G}_{\mathsf{K}_i}(\$))_{i\in[1;\mu]}$ and $(\Gamma_i(\$))_{i\in[1;\mu]}$, where $\$$ is the uniform distribution sampler implemented via a uniform random function from $\mathcal{M}$ to $\{0,1\}^\tau$ (courtesy of $\mathbf{Adv}^{\text{mu-prf}}_{\mathsf{F}}(\mathscr{B})$). We simply define $\mathscr{C}$ as the distinguisher that runs $\mathscr{A}'$ in a black box manner, answering all its queries by applying E^+ (keyed with the answers given by its own oracle on uniform at random inputs) and outputs the same value as $\mathscr{A}'$. Then, clearly, $\mathscr{C}$ correctly simulates $(\overline{\mathsf{E}}[\mathsf{G}_{\mathsf{K}_i}(\$),\mathsf{E}]^+)_{i\in[1;\mu]}$ when its oracle is $(\mathsf{G}_{\mathsf{K}_i}(\$))_{i\in[1;\mu]}$, and it correctly simulates $(\overline{\mathsf{E}}[\Gamma_i,\mathsf{E}]^+)_{i\in[1;\mu]}$ when its oracle is $(\Gamma_i(\$))_{i\in[1;\mu]}$. Moreover, $\mathscr{C}$ makes at most q queries to its oracle and runs in time $\hat{t} = O(t + qt_{\mathsf{G}})$. Thus, we have

$$\begin{aligned}\mathbf{Adv}^{\text{mu-wprf}}_{\mathsf{G}}(\mathscr{C}) &= \left|\Pr_{\$,(\mathsf{K}_i)_{i\in[1;\mu]}\twoheadleftarrow\$\mathcal{K}}\left(\mathscr{C}^{(\mathsf{G}_{\mathsf{K}_i}(\$))_{i\in[1;\mu]}=1}\right) - \Pr_{\$,(\Gamma_i)_{i\in[1;\mu]}\twoheadleftarrow\$\mathsf{Func}(\tau;\kappa+n)}\left(\mathscr{C}^{(\Gamma_i(\$))_{i\in[1;\mu]}}=1\right)\right| \\ &\geq \left|\Pr_{\substack{(\mathsf{K}_i)_{i\in[1;\mu]}\\ \twoheadleftarrow\$\mathcal{K}}}\left(\mathscr{A}'^{(\overline{\mathsf{E}}[\mathsf{G}_{\mathsf{K}_i},\mathsf{E}]^+)_{i\in[1;\mu]}}=1\right) - \Pr_{(\Gamma_i)_{i\in[1;\mu]}\twoheadleftarrow\$\mathsf{Func}(\tau;\kappa+n)}\left(\mathscr{A}'^{(\overline{\mathsf{E}}[\Gamma_i,\mathsf{E}]^+)_{i\in[1;\mu]}=1}\right)\right|.\end{aligned} \tag{5.6}$$

Before we move on to constructing the distinguisher $\mathscr{D}$, we introduce a small change in the game: instead of sampling the IVs for $\mathscr{A}'$'s oracle ($(\overline{\mathsf{E}}[\Gamma_i,\mathsf{E}]^+)_{i\in[1;\mu]}$ or $\$$) in a with replacement fashion, we sample the IVs in a without replacement manner, i.e., all the IVs will be distinct. Let the appropriately modified oracles be $(\widetilde{\mathsf{E}}[\Gamma_i,\mathsf{E}])_{i\in[1;\mu]}$ and $\widetilde{\$}$. This switching is possible at the cost of two times the statistical distance between with and without replacement samples of size q, i.e. $q^2/2^\tau$. Formally, we have

$$\begin{aligned}&\left|\Pr\left(\mathscr{A}'^{(\overline{\mathsf{E}}[\Gamma_i,\mathsf{E}]^+)_{i\in[1;\mu]}}=1\right) - \Pr\left(\mathscr{A}'^{\$}=1\right)\right| \\ &\leq \left|\Pr\left(\mathscr{A}'^{(\overline{\mathsf{E}}[\Gamma_i,\mathsf{E}]^+)_{i\in[1;\mu]}}=1\right) - \Pr\left(\mathscr{A}'^{(\widetilde{\mathsf{E}}[\Gamma_i,\mathsf{E}]^+)_{i\in[1;\mu]}}=1\right)\right| \\ &+ \left|\Pr\left(\mathscr{A}'^{(\widetilde{\mathsf{E}}[\Gamma_i,\mathsf{E}]^+)_{i\in[1;\mu]}}=1\right) - \Pr\left(\mathscr{A}'^{\widetilde{\$}}=1\right)\right| \\ &+ \left|\Pr\left(\mathscr{A}'^{\widetilde{\$}}=1\right) - \Pr\left(\mathscr{A}'^{\$}=1\right)\right| \\ &\leq \left|\Pr\left(\mathscr{A}'^{(\widetilde{\mathsf{E}}[\Gamma_i,\mathsf{E}]^+)_{i\in[1;\mu]}}=1\right) - \Pr\left(\mathscr{A}'^{\widetilde{\$}}=1\right)\right| + \frac{q^2}{2^\tau}.\end{aligned} \tag{5.7}$$

Constructing a Distinguisher $\mathscr{D}$: Now, we define $\mathscr{D}$ as a $(q,1,q,\ell,\sigma,\dot{t})$-PRIV$ distinguisher that runs $\mathscr{A}'$ in a black box manner. For each query (m,i) from $\mathscr{A}'$, $\mathscr{D}$ chooses a fresh user ID, t from the set $\{0,1\}^\tau$ in a without replacement manner. It then queries (m,t) to its own oracle (either $(\mathsf{E}^+_{\Gamma_i(t)}(m))_{i\in[1;\mu]}$ or $(\overline{\Gamma}_j(t,m))_{j\in[1;q]}$), where $(\overline{\Gamma}_j)_{j\in[q]} \twoheadleftarrow_{\$} \mathsf{Func}_{\mathsf{lp}}(\{0,1\}^\tau\times\mathcal{M};\mathcal{M})$, and returns (t,c) to $\mathscr{A}'$, where c is the corresponding response of $\mathscr{D}$'s oracle. At the end $\mathscr{D}$ outputs the same value as $\mathscr{A}'$. It is obvious to see that $\mathscr{D}$ correctly simulates $\widetilde{\$}$ when it is interacting with $(\overline{\Gamma}_j)_{j\in[1;q]}$. Also, since Γ is a random function, $\mathscr{D}$ correctly simulates $(\widetilde{\mathsf{E}}[\Gamma_i,\mathsf{E}]^+)_{i\in[1;\mu]}$ when it is interacting with $(\mathsf{E}^+_{\Gamma_i(\cdot)})_{i\in[1;\mu]}$. Moreover, $\mathscr{D}$ makes at most q queries to its oracle and runs in time

$\dot{t} = O(t + qt_{\mathsf{E}})$. Thus, we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{priv\$}}_{\mathsf{E}}(\mathscr{D}) = &\left| \Pr_{\substack{(\Gamma_i)_{i\in[1;\mu]} \twoheadleftarrow\$ \\ \mathsf{Func}(\tau;\kappa+n)}} \left(\begin{smallmatrix}(\Gamma_i)_{i\in[1;\mu]} \\ \twoheadleftarrow\$\mathsf{Func}(\tau;\kappa+n)\end{smallmatrix} : \mathscr{D}^{(\mathsf{E}^+_{\Gamma_i(\cdot)})_{i\in[1;\mu]}} = 1 \right) \right. \\ &\left. - \Pr_{\substack{(\bar{\Gamma}_j)_{j\in[1;q]} \twoheadleftarrow\$ \\ \mathsf{Func}_{\mathsf{lp}}(\{0,1\}^\tau\times\mathcal{M};\mathcal{M})}} \left(\mathscr{D}^{(\bar{\Gamma}_j)_{j\in[1;q]}} = 1 \right) \right| \\ &\geq \left| \Pr\left(\mathscr{A}'^{(\widetilde{\mathsf{E}}[\Gamma_i,\mathsf{E}]^+)_{i\in[1;\mu]}} = 1 \right) - \Pr\left(\mathscr{A}'^{\widetilde{\$}} = 1 \right) \right|. \end{aligned} \tag{5.8}$$

The result follows from (5.5)-(5.8). □

5.2.2 Abstraction of MTProto2.0

The protocol MTProto2.0 can be seen as an instantiation of MTProto-G, where the three underlying functions: F, G, and E, are constructed using the hash function SHA-256 [255] and the IV-based encryption mode of operation Infinite Garble Extension (IGE) [68].

SHA-256: The full construction uses a Merkle-Damgård paradigm [240, 107] with a Davies-Meyer compression function [270] and length-strengthened padding. Let $r = 512$, $c = 256$, $\ell = 64$ and a function $f \in \mathsf{Func}(r + c; c)$. We define the length-strengthened padding function $\mathsf{pad}_r : \{0,1\}^{<2^\ell} \to \{0,1\}^{r+}$, where $\{0,1\}^{r+}$ (resp. $\{0,1\}^{<2^\ell}$) denotes the set of (non-empty) bit strings whose length is a multiple of r (resp. of length smaller than 2^ℓ), by the mapping

$$m \mapsto m\|10^d\|\langle|m|\rangle_\ell,$$

where $d = \min\{i \geq 0 \ : \ |m| + 1 + i + \ell \pmod r \equiv 0\}$ and $\langle|m|\rangle_\ell$ denotes the 64-bit unsigned binary representation of $|m|$. Let $\mathrm{iv} \in \{0,1\}^c$ be some application constant. Formally, the SHA-256 algorithm based on compression function f is defined as follows (see figure for a simplified version): for all $m \in \{0,1\}^{<2^\ell}$ we write,

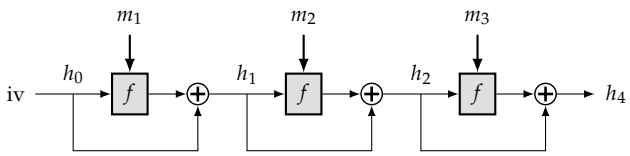

Figure 5.4: *SHA-256 hash computation over a 3-block padded message* $m_1\|m_2\|m_3 = \mathsf{pad}_r(m)$.

$$\begin{aligned}(m_1,\ldots,m_l) &:= \mathsf{pad}_r(m), \qquad & h_0 &:= \mathrm{iv},\\ h_i &:= f(m_i, h_{i-1}) \oplus h_{i-1}, \qquad & \forall i &\in [1;\ell]\end{aligned}$$

and finally, $\mathtt{SHA\text{-}256}(m) := h_\ell$. We refer to h_i values as *compression input* and the r-bit inputs as *compression key*.

Note that the mapping $(m_i, h_{i-1}) \mapsto h_i$ applies the well-known Davies-Meyer transformation using f as the underlying primitive. In `MTProto2.0`, `SHA-256` is exclusively used to construct two hash based PRFs, F and G. Before we describe these functions, we first digress a little to discuss the security assumption on f vis-à-vis the security analysis of F and G.

Security Assumption on f: It is worth noting that, f can actually be viewed as a block cipher with an r-bit key and a c-bit block. Indeed, later we assume that the underlying block cipher in `SHA-256`, i.e. the f function, is a TBC following the TWEAKEY framework by Jean et al. [186] described thoroughly in Chapter 4. Thus, f is a TBC in $\widetilde{\mathsf{BC}}\left(r - \frac{\kappa}{2}; \frac{\kappa}{2}; \{0,1\}^c\right)$ can be identified with a TBC in $\widetilde{\mathsf{BC}}\left(\mathcal{B} \times \{0,1\}^{r-\frac{\kappa}{2}}; \frac{\kappa}{2}; \{0,1\}^c\right)$ where

$$\mathcal{B} \subseteq \{x \in \{0,1\}^r : HW(x) = r - \frac{\kappa}{2}\}.$$

Further, each $b = b_0 \| \cdots \| b_{r-1} \in \mathcal{B}$ pinpoints the placement of tweak bits ($t \in \{0,1\}^{r-\frac{\kappa}{2}}$) in the actual tweakey. For $i \in [0; r-1]$, if $b_i = 1$ then the i-th bit of the tweakey holds the next-in-line (starting from the first) bit of tweak t. While there is no explicit analysis of this feature, the framework itself does not distinguish between key and tweak, or their respective placement in the tweakey. Each choice of placement gives a new TBC, which justifies our assumption. Indeed practical examples such as `Skinny` and `Mantis` [20] satisfy this criteria, i.e., one can choose multiple b vectors freely to create multiple instances. Specifically, in case of f, we define

$$\mathcal{B} = \{B_a := 1^\tau \| 0^{\frac{\kappa}{2}} \| 1^{r-\frac{\kappa}{2}-\tau} \| 1^{r-\frac{\kappa}{2}}, B_b := 0^{\frac{\kappa}{2}} \| 1^{r-\frac{\kappa}{2}}\}.$$

G function: For simplicity we assume a one-way communication from clients to the server. Let $\kappa = 576$ and $\tau = 128$. The G function takes a κ-bit key $k = (k_0, k_1)$, where $|k_b| = \kappa/2$, and a τ-bit input x and produces a $2c$-bit output $y = (y_0, y_1)$, where $|y_b| = c$. Internally, G can be viewed as two parallel invocations of `SHA-256` with independent keys. Formally, for key (k_0, k_1) and input t, we have $y = \mathsf{G}_k(x) := (\overline{\mathsf{G}}_{k_0}(x), \widetilde{\mathsf{G}}_{k_1}(x))$, where

$$\overline{\mathsf{G}}_{k_0}(x) := a[0, \ldots, 7] \| b[8, \ldots, 23] \| a[24, \ldots, 31]$$
$$\widetilde{\mathsf{G}}_{k_1}(x) := b[0, \ldots, 7] \| a[8, \ldots, 23] \| b[24, \ldots, 31]$$

and $a = \texttt{SHA-256}(x \| k_0), b = \texttt{SHA-256}(k_1 \| x)$.

In Lemma 5.2.2, under the assumption that f is a TPRP, we show that G is a secure wPRF as is required in Lemma 5.2.1.

Lemma 5.2.2. *Let $\mathscr{D}$ be a (μ, q, t) wPRF distinguisher against G that issues at most q queries over at most μ users, and runs in time at most t. Then there exists a $(\mu, 2q, t')$ TPRP distinguisher $\mathscr{D}'$ against f such that we have*

$$\mathbf{Adv}^{\text{mu-wprf}}_{\mathsf{G}}(\mathscr{D}) \leq \mathbf{Adv}^{\text{mu-tprp}}_{f}(\mathscr{D}').$$

Proof. First, using the tweakey block cipher description of f, we can redefine a and b as:

$$a := f_{k_0}((B_a, x\|10^{31}\|\langle\tau\rangle_{64}), \text{iv}) \oplus \text{iv},$$
$$b := f_{k_1}((B_b, x\|10^{31}\|\langle\tau\rangle_{64}), \text{iv}) \oplus \text{iv},$$

where

$$B_a = 1^{\tau}\|0^{\frac{\kappa}{2}}\|1^{r-\frac{\kappa}{2}-\tau}, \qquad B_b = 0^{\frac{\kappa}{2}}\|1^{r-\frac{\kappa}{2}}.$$

Now, using a simple hybrid argument we can replace all the instances of f with tweakable random permutations, which incur a cost of at most $\mathbf{Adv}^{\text{tprp}}_{f}(\mathscr{D}')$. The remainder of the proof follows from the fact that the output distribution of a tweakable random permutation is identical to a random function, given that it is always invoked with distinct tweaks. □

F function: This function is simply defined as $\mathsf{F}_k(m) := \mathsf{chop}_\tau(\texttt{SHA-256}(k\|m))$ for all keys $k \in \{0,1\}^{r/2}$ and messages $m \in \{0,1\}^{\ell}$, where $\mathsf{chop}_\tau(\cdot)$ returns a substring of its output of length τ bits. In essence, F is nothing but the popular hash-based MAC construction called AMAC [44, 23]. In [23], Bellare et al. showed that AMAC is a multi-user secure PRF under the assumption that the underlying compression function is a secure PRF under the presence of leakage. Additionally, in the same paper, they show that within the ideal cipher model, a Davies-Meyer style compression function remains a secure PRF, even in the presence of leakage through truncation. Here, we restate their result in our setting and for F. First, we define two keyed functions $\texttt{DM}[f] : \{0,1\}^r \times \{0,1\}^c \to \{0,1\}^c$ and $\texttt{DMD}[f] : \{0,1\}^c \times \{0,1\}^r \to \{0,1\}^c$ with r-bit and c-bit keys respectively, as follows

$$\texttt{DM}_f(k,x) := f_k(x) \oplus x,$$
$$\texttt{DMD}_f(k,x) := \texttt{DM}_f(x,k).$$

Lemma 5.2.3 (Theorem 5.3 in [23]). *Let $\mathscr{D}$ be a $(\mu, q, \ell, \sigma, t)$ PRF distinguisher against F, that issues at most q queries of length at most ℓ n-bit blocks, over at most u users, for a total queries length of at most σ n-bit blocks. Then, there exists (μ, q, t'') and $(\mu, q, \ell, \sigma, t')$ distinguishers $\mathscr{A}$ and $\mathscr{B}$, respectively, such that*

$$\mathbf{Adv}^{\text{mu-prf}}_{\mathsf{F}}(\mathscr{D}) \le 2\mathbf{Adv}^{\text{mu-prf}}_{DM_f}(\mathscr{A}) + \ell\mathbf{Adv}^{\text{prfleak}}_{DMD_f,\mathsf{chop}_\tau}(\mathscr{B}).$$

Infinite Garble Extension (IGE): The mode is an extension of the well-known Cipher Block Chaining (CBC) mode [131], where each plaintext block is XORed with the previous ciphertext block before encryption, thereby enhancing security by chaining each block's encryption to the previous one. In IGE mode, alongside the CBC-like ciphertext feed-forward to the next block cipher input, plaintext feed-forward is also employed to the next block cipher output, as illustrated in Figure 5.5.

Formally, we define the IGE construction as follows: for a positive integer $n \in \mathbb{N}$ let $\mathtt{E} \in \mathsf{BC}(\mathcal{K}; \{0,1\}^n)$ be some block cipher. Let $\mathcal{I} = \{0,1\}^{2n}$ be the nonce space and

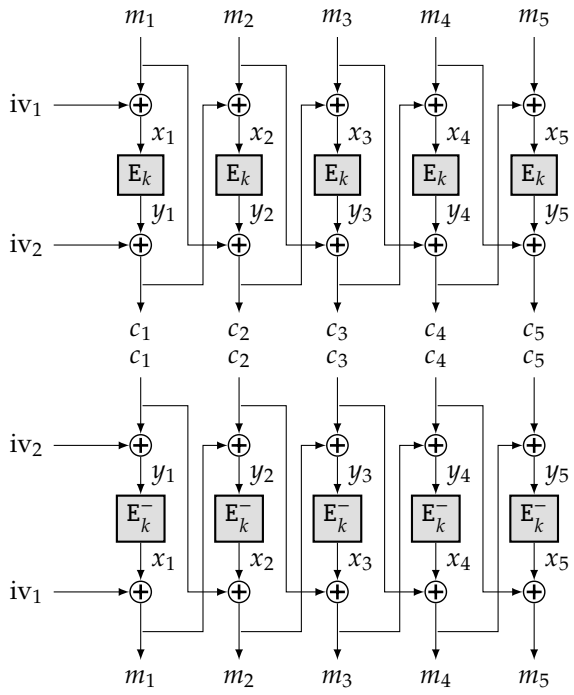

Figure 5.5: *IGE encryption (**top**) and decryption (**bottom**) algorithms.*

$\mathcal{M} = \{0,1\}^{\ell n}$ for some integer ℓ be the message space. For every $m = (m_1, \ldots, m_\ell) \in \mathcal{M}$ and $(\mathrm{iv}_1, \mathrm{iv}_2) \in \mathcal{I}$ the encryption function is defined as,

$$\mathsf{E}^+_{k,\mathrm{iv}_1,\mathrm{iv}_2}(m) =: c = (c_1, \ldots, c_\ell),$$

where

$$c_i = \begin{cases} \mathrm{E}_k(\mathrm{iv}_1 \oplus m_1) \oplus \mathrm{iv}_2, & i = 1 \\ \mathrm{E}_k(c_{i-1} \oplus m_i) \oplus m_{i-1}, & i > 1 \end{cases}.$$

Similarly, we define the decryption function as follows. For every $c = (c_1, \ldots, c_\ell) \in \mathcal{M}$ and $(\mathrm{iv}_1, \mathrm{iv}_2) \in \mathcal{I}$,

$$\mathsf{E}^-_{k,\mathrm{iv}_1,\mathrm{iv}_2}(c) = m = (m_1, \ldots, m_\ell),$$

where

$$m_i = \begin{cases} \mathrm{E}_k^{-1}(\mathrm{iv}_2 \oplus c_1) \oplus \mathrm{iv}_1, & i = 1 \\ \mathrm{E}_k^{-1}(m_{i-1} \oplus c_i) \oplus c_{i-1}, & i > 1 \end{cases}.$$

For most part of our analysis, only the privacy security of `IGE` will suffice. Accordingly, we bound the advantage of a distinguisher that tries to distinguish between `IGE` and a uniform random string generator in Lemma 5.2.4.

Lemma 5.2.4. *Let $\mathscr{A}$ be a $(\mu, q_{\max}, q, \ell, \sigma, t)$ multi-user distinguisher against `IGE`, that runs in time at most t, and such that each user $u \in [1;\mu]$ makes q_u queries (the maximal number of queries for a single user is denoted by q_{max}), each of length (in n-bit blocks) at most ℓ, of total queries length at most σ. Then there exists a multi-user distinguisher $\mathscr{A}'$ against PRP with u*

users, at most $q_u \cdot \ell$ queries per user, and total σ queries across all users such that,

$$\mathbf{Adv}^{\mathsf{priv\$}}_{IGE}(\mathscr{A}) \leq \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathscr{A}') + \frac{2\mu q_{max}\ell^2}{2^n}.$$

Proof. First, using a simple straightforward hybrid argument, for our adversary $\mathscr{A}$ there exists a multi-user distinguisher $\mathscr{A}'$ against PRP as described above such that,

$$\mathbf{Adv}^{\mathsf{priv\$}}_{\mathtt{IGE}}(\mathscr{A}) \leq \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathscr{A}') + \delta(\mathscr{A}),$$

where $\delta(\mathscr{A})$ is the advantage of $\mathscr{A}$ against IGE where the permutation for each user $\mathtt{E}_{k_u}$ is replaced by a uniform random permutation of $\{0,1\}^n$. Note that all these random permutations are independent from one another.

For the rest of this proof we employ the H-coefficient technique described in Section 2.4.2. In this context, an adversary $\mathscr{A}$ prompts a user $u \in [1;\mu]$ to make a query $m^{i,u}$. The adversary then receives a pair of uniformly random IVs $\mathrm{iv}_1^{i,u}, \mathrm{iv}_2^{i,u} \twoheadleftarrow_{\$} \mathcal{I}$, and one of two possible responses:

- in the real world, a ciphertext $c^{i,u} = \mathsf{E}^{+}_{k_u,\mathrm{iv}_1^{i,u},\mathrm{iv}_2^{i,u}}(m^{i,u})$, or
- in the ideal world, a uniformly random string $c^{i,u} \twoheadleftarrow_{\$} \{0,1\}^{n\ell}$, where the function $\Gamma_u \twoheadleftarrow_{\$} \mathsf{Func}(\mathcal{M};\mathcal{M})$.

Thus, the transcript of an adversary $\mathscr{A}$, denoted $\tau(\mathscr{A})$, is defined as

$$\tau(\mathscr{A}) = \{(m^{i,u}, c^{i,u}, \mathrm{iv}_1^{i,u}, \mathrm{iv}_2^{i,u}) : u \in [1;\mu], i \in [1;q_u]\},$$

where for every $i \in [q_u]$, $m^{i,u}, c^{i,u} \in \mathcal{M}$ and $\mathrm{iv} = (\mathrm{iv}_{i,1}, \mathrm{iv}_{i,2}) \in \mathcal{I}$ are chosen uniformly and independently from $\mathcal{I}$. Further, we define some intermediate values. For every user $u \in [1;\mu]$, $i \in [1;q_u]$, $j \in [1;\ell]$ let

$$x^u_{i,j} = \begin{cases} m^{i,u}_1 \oplus \mathrm{iv}^{i,u}_1, & j = 1 \\ m^{i,u}_j \oplus c^{i,u}_{j-1}, & j > 1 \end{cases}, \qquad y^u_{i,j} = \begin{cases} c^{i,u}_1 \oplus \mathrm{iv}^{i,u}_2, & j = 1, \\ c^{i,u}_j \oplus m^{i,u}_{j-1}, & j > 1 \end{cases}.$$

Note that if the adversary $\mathscr{A}$ interacts with the real world, since $\mathtt{E}_{k_u}$ is a permutation for any key k_u, he can leverage the following property to gain an advantage:

$$x^u_{i,j} = x^u_{i',j'} \Leftrightarrow y^u_{i,j} = y^u_{i',j'}, \qquad \forall (i,j) \neq (i',j') \in [1;q_u] \times [1;\ell].$$

To avoid such scenarios, we say a transcript τ is bad if one of two events occur

$$\exists u \in [1;\mu], (i,j) \neq (i',j') \in [1;q_u] \times [1;\ell] : x^u_{i,j} = x^u_{i',j'}, \quad y^u_{i,j} \neq y^u_{i',j'}, \tag{B_1}$$

$$\exists u \in [1;\mu], (i,j) \neq (i',j') \in [1;q_u] \times [1;\ell] : y^u_{i,j} = y^u_{i',j'}, \quad x^u_{i,j} \neq x^u_{i',j'}. \tag{B_2}$$

Finally, we denote by Θ_{bad} the set of all possible bad transcripts, and by Θ_{good} the set of all remaining transcripts.

On the other hand in the ideal world for every $(i,j) \neq (i',j')$ the random variables $c_j^{i,u}$ and $c_{j'}^{i',u}$ are independent uniform random variables over $\{0,1\}^n$. Thus,

$$\Pr\left(x_{i,j}^u = x_{i',j'}^u\right) = \Pr\left(c_{j-1}^{i,u} \oplus c_{j'-1}^{i',u} = m_j^{i,u} \oplus m_{j'}^{i',u}\right) = \frac{1}{2^n}.$$

Similarly, $\Pr\left(y_{i,j}^u = y_{i',j'}^u\right) = \frac{1}{2^n}$. In conclusion, the probability of a bad transcript can be upper bounded the following way.

$$\begin{aligned}\Pr\left(\tau(\mathscr{A}) \in \Theta_{\text{bad}}\right) &= \Pr_{\tau(\mathscr{A})}(\mathsf{B}_1) + \Pr_{\tau(\mathscr{A})}(\mathsf{B}_2) \\ &\leq \sum_{u\in[1;\mu]} \sum_{(i,j)\neq(i',j')\in[1;q_u]\times[1;\ell]} \Pr\left(x_{i,j}^u = x_{i',j'}^u, y_{i,j}^u \neq y_{i',j'}^u\right) \\ &+ \Pr\left(y_{i,j}^u = y_{i',j'}^u, x_{i,j}^u \neq x_{i',j'}^u\right) \leq \frac{2\mu q_{max}\ell^2}{2^n}.\end{aligned}$$

Finally, let T_{re} be the random variable corresponding to the real world and T_{id} to the ideal world. Then, for a good transcript $\tau(\mathscr{A})$, and using the randomness of the IVs and permutations (block ciphers), one has

$$\Pr\left(\mathsf{T}_{\text{re}} = \tau(\mathscr{A})\right) = \prod_{u=1}^{\mu} \frac{1}{2^{2q_u n} \cdot (2^n)_{q_u\ell}}, \qquad \Pr\left(\mathsf{T}_{\text{id}} = \tau(\mathscr{A})\right) = \prod_{u=1}^{\mu} \frac{1}{2^{q_u n(\ell+2)}}. \tag{5.9}$$

Moreover, It is easy to see that for every user $u \in [1;\mu]$

$$\frac{2^{q_u n\ell}}{(2^n)_{q_u\ell}} \geq 1 \tag{5.10}$$

Using equations (5.9) and (5.10) together with Corollary 2.4.1, one has

$$\mathbf{Adv}_{\text{IGE}}^{\text{priv\$}}(\mathscr{A}) \leq \mathbf{Adv}_{\text{E}}^{\text{prp}}(\mathscr{A}') + \frac{2\ell^2\mu q_{max}\ell^2}{2^n}.$$

□

5.3 Subversion Attacks

This section introduces the notion of subversion attacks, providing a formal definition and discussing several key examples from previous research, on which we will build upon in the following section.

5.3.1 Algorithm Substitution Attack (ASA)

We formalize subversion attacks via the notion of algorithm substitution attacks (ASAs) by following the definitions from [31]. From a high level, a subversion attack aims to replace an encryption scheme with a different keyed algorithm, with the following two goals:

- the subversion should be difficult to distinguish from the actual encryption scheme for someone who does not know the adversary's key;

- the subversion should break the security of the subverted encryption scheme in some way.

In this chapter, as in [31], we focus on key-recovery attacks.

Formal Definition. Let $\mathcal{E} = (\mathcal{E}^+, \mathcal{E}^-)$ be an AEAD with key space $\mathcal{K}$. A subversion of $\mathcal{E}$ is a tuple $\widetilde{\mathcal{E}} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}^+}, \widetilde{\mathcal{E}^{\text{ext}}})$, where the master-key space $\widetilde{\mathcal{K}}$ is a non-empty set, such that:

- the subverted encryption algorithm $\widetilde{\mathcal{E}^+}$ maps a tuple (K_A, K_E, A, M, σ) to a pair (C, σ'), where $A, M \in \{0,1\}^*$, $K_A \in \widetilde{\mathcal{K}}$, $K_E \in \mathcal{K}$, C is a ciphertext and σ' corresponds to the update of the state σ;
- the key-recovery algorithm $\widetilde{\mathcal{E}^{\text{ext}}}$ takes as input a master key $\widetilde{\mathcal{K}}$, a vector of associated data $\mathbf{A}$, a vector of ciphertexts $\mathbf{C}$, and produces a key guess $K \in \mathcal{K}$.

We say that $\widetilde{\mathcal{E}}$ is *decryptable* (with respect to $\mathcal{E}$) if for every plaintext M, every associated data A, every key tuple $(K_E, K_A) \in \mathcal{K} \times \widetilde{\mathcal{K}}$ and every state σ, one has

$$\mathcal{E}^-_{K_E}(A, \widetilde{\mathcal{E}^+}_{K_A,K_E}(A, M, \sigma)) = M.$$

Besides, if the state σ is never updated by the encryption algorithm, we say that $\widetilde{\mathcal{E}}$ is *stateless*. Otherwise, it is said to be *stateful*.

Undetectability. It is clear that a subversion attack can only be effective as long as it is hard to detect. In this chapter, we concentrate on computational detection. Namely, the output of the attacker's encryption scheme should be indistinguishable from the output of the subverted scheme, even from the point of view of the decryption algorithm. We formalize this notion with the (multi-user) detection games presented in Figure 5.6. Our undetectability definition differs from the strong undetectability notion from [31] in two ways:

- we allow the subverted algorithm to also be stateful: looking ahead momentarily, our goal is to model the behavior of secret chats in the `MTProto` protocol, which maintain a state σ that counts the number of sent and received messages for each key (see Section 5.1.2);
- in [31], the attacker is allowed to choose the key of the encryption scheme $\mathcal{E}$, while, in our game, the key is generated uniformly at random.

Informally, we can think of this notion as the strong undetectability where the adversary is assumed to honestly generate encryption keys (the honest setting). We believe this assumption is natural. In this context, the attacker could be Telegram servers, a Telegram client, or an external observer attempting to detect an ASA. All these actors have an interest in maintaining secure communications by generating encryption keys uniformly at random, rather than intentionally creating weak keys (for instance, by using small order elements in a Diffie-Hellman key exchange to cause a lot of key collisions).

We define the advantage of an adversary $\mathscr{D}$ trying to distinguish between the genuine encryption scheme $\mathcal{E}$ and its subversion $\widetilde{\mathcal{E}}$ as follows:

$$\mathbf{Adv}^{\mathsf{det}}_{\mathcal{E},\widetilde{\mathcal{E}}}(\mathscr{D}) := \left|\Pr\left(\mathrm{Re}^{\mathsf{det}}_{\mathcal{E},\widetilde{\mathcal{E}}}(\mathscr{D}) = 1\right) - \Pr\left(\mathrm{Sub}^{\mathsf{det}}_{\mathcal{E},\widetilde{\mathcal{E}}}(\mathscr{D}) = 1\right)\right|.$$

$\mathrm{Re}^{\mathsf{dae}}_{\mathcal{E}}(\mathscr{D})$ $\boxed{\mathrm{Id}^{\mathsf{dae}}_{\mathcal{E}}(\mathscr{D})}$

1: $u \leftarrow 0$
2: $K_A \twoheadleftarrow_{\$} \widetilde{\mathcal{K}}$
3: $b \leftarrow \mathscr{D}^{\textsc{New},\textsc{Enc}}$
4: **return** $b =_? 1$

Oracle $\textsc{New}()$

1: $u \leftarrow u + 1$
2: $\sigma_u \leftarrow \epsilon$
3: $K_u \twoheadleftarrow_{\$} \mathcal{K}$
4: **return** (K_u, σ_u)

Oracle $\textsc{Enc}(A, M, i)$

1: **if** $i \notin [u]$ **then return** $\perp$
2: **end if**
3: $(C, \sigma_i) \leftarrow \mathcal{E}^{+}_{K_i}(A, M, \sigma_i)$
4: $\boxed{(C, \sigma_i) \leftarrow \widetilde{\mathcal{E}^{+}}_{K_A,K_i}(A, M, \sigma_i)}$
5: **return** (C, σ_i)

Figure 5.6: *Detection Game for subversion* $\widetilde{\mathcal{E}}$

Key Recovery. The main goal of a subversion attack is to break the security of the original encryption scheme $\mathcal{E}$ in some way. The weakest possible goal would be to allow the attacker to distinguish between $\widetilde{\mathcal{E}}$ and an ideal encryption scheme. However, the practical consequences of such an attack are minimal. Therefore, we focus on attacks that enable the recovery of part of the key of $\mathcal{E}$, which would allow the decryption of all ciphertexts.

We formalize the key recovery experiment in Figure 5.7. Note that the game is parameterized by an algorithm $\mathfrak{M}$ that samples new message queries when given the current state σ' (which may be different from the state maintained by the protocol), and a number of queries q. We stress that our attack will work independently of the choice of $\mathfrak{M}$, and its success will only depend on the number of encryption queries. The subversion attack is successful if $\widetilde{\mathcal{E}^{\mathsf{ext}}}$ recovers the key K_E from the ciphertexts produced by $\widetilde{\mathcal{E}^{+}}$ on messages produced by $\mathfrak{M}$, and its advantage is defined as:

$$\mathbf{Adv}^{\mathsf{kr}}_{\mathfrak{M},q}\left(\widetilde{\mathcal{E}}\right) := \Pr\left(\mathrm{Kr}_{\widetilde{\mathcal{E}}}(\mathfrak{M}, q) = 1\right).$$

5.3.2 Previous Subversion Attacks for Encryption Schemes

In [36], Bellare et al. present a very simple subversion attack again IV-based encryption schemes such that the IV is public in ciphertexts (see Algorithm 5.3.6). The attack simply

$\mathrm{Kr}_{\widetilde{\mathcal{E}}}(\mathfrak{M}, q)$

1: $(K_E, K_A) \twoheadleftarrow_{\$} \mathcal{K} \times \widetilde{\mathcal{K}}$
2: $K \leftarrow \widetilde{\mathcal{E}^{\mathrm{ext}}}(K_A, \mathcal{A}, \textsc{Enc}(\mathfrak{M}, q))$
3: **return** $K =_? K_E$

Oracle $\textsc{Enc}(\mathfrak{M}, q)$

1: $\sigma' \leftarrow \varepsilon$
2: **for** $i \in [1; q]$ **do**
3: $\quad (A_i, M_i, \sigma') \leftarrow \mathfrak{M}(i, \sigma')$
4: $\quad (C_i, \sigma') \leftarrow \widetilde{\mathcal{E}^{+}}_{K_A, K_E}(A_i, M_i, \sigma')$
5: **end for**
6: $\mathbf{A} = (A_1, \ldots, A_q)$
7: $\mathbf{C} = (C_1, \ldots, C_q)$
8: **return** $(\mathbf{A}, \mathbf{C})$

Figure 5.7: *Key recovery game for subversion* $\widetilde{\mathcal{E}}$

encrypts the target key K_E using the adversarial key K_A, and uses this value as IV for the first encryption query. The main drawback of this attack is that it is inherently stateful: a simple state reset allows the detection of the subversion, as it triggers an IV repetition.

In order to avoid such simple countermeasures and to make the attack usable against any randomized encryption scheme, Bellare et al. introduce a new stateless substitution attack in [31] (see Algorithm 5.3.7 for a description of the algorithm). The key idea is to rely on a second PRF F with output space $\{0,1\} \times [1; n]$ where $|K_E| = n$, and to sample IVs until the corresponding ciphertext C satisfies $\mathtt{F}_{K_A}(C) = (b, i)$, where $K_E[i] = b$ (the i-th bit of K_E is b). The new attack is then stateless in nature but comes at a cost: the attack is now randomized and can fail to transmit a specific key bit.

Later on, more general subversion attacks have been introduced. For example, Armour and Poettering [11] proposed another stateless attack that targets the decryption algorithm of any AE scheme. It operates similarly as the previous attack: when the decryption algorithm is given a ciphertext such that $F_{K_A}(C) = (b, i)$, where $K_E[i] = b$, it will reject the ciphertext instead of decrypting. The main difference with previous attacks is that this subversion comes at a functionality cost: some valid ciphertexts get rejected. In order to avoid the easy detection of the attack, the subverted algorithm will only test a small fraction of all ciphertexts.[2]

5.4 Subverting Secret Chats in MTProto2.0

This section introduces our subversion attack against the MTProto2.0 protocol, exploiting a vulnerability in the unique structure of the random padding detailed in Section 5.1.3.

[2] This is done at random by sampling a Bernoulli random variable in order to decide whether to attack a particular ciphertext or not.

Algorithm 5.3.6 Pseudocode of the subversion attack from [36]. Here $\mathcal{E}$ denotes an IV-based encryption scheme, and E is a length-preserving deterministic encryption scheme.

function $\widetilde{\mathcal{E}^+}(K_A, K, A, M, \sigma)$
  if $\sigma = 0$ **then**
    $\text{iv} \leftarrow E(K_A, K)$
  else
    $\text{iv} \twoheadleftarrow_{\$} \{0,1\}^n$
  end if
  $C \leftarrow \mathcal{E}^+(K, A, M, \text{iv})$
  $\sigma \leftarrow \sigma + 1$
  return (C, σ)
end function

function $\widetilde{\mathcal{E}^{\text{EXT}}}((K_A, \mathbf{A}, \mathbf{C}, i))$
  $\text{iv} \leftarrow \text{iv}(\mathbf{C}[1])$
  $K \leftarrow E^-(K_A, \text{iv})$
  return K
end function

5.4.1 Simple Attack on `MTProto2.0`

Intuition Behind The Attack. As previously discussed in Section 5.3, state reset is a simple countermeasure that can make stateful subversion attacks easy to detect. In the general case, it is thus important to design stateless subversion attacks. When attacking encryption schemes used in complex protocols, this requirement can sometimes be alleviated. For example, if the protocol maintains an internal counter that is given to the encryption scheme via associated data or plaintext, then a subversion attack can simply rely on this external counter in order to act as a stateful attack, even if it does not directly maintain its state. This is exactly what happens during secret chats, as the `MTProto2.0` protocol relies on sequence counters in order to uniquely identify each message. In more details, from any encryption query, it is possible to extract a monotonically increasing counter that only depends on the number of encryption queries issued by the client, and the only time where these counters are reset is during the rekeying of the authenticated encryption scheme. We can thus rely on this counter in order to mount a stateful substitution attack.[3]

Description of The Attack. For the remainder of this section, we focus on the `MTProto` authenticated encryption scheme (without associated data). Moreover, we fix a padding scheme that generates ciphertexts whose length in 128-bit blocks is uniformly distributed modulo 16.[4] In order to create a subversion attack, we have to find a way to exploit the randomization of the encryption scheme in order to ex filtrate key bits. As described in Section 5.1.1, although the random values used as input are encrypted and authenticated, part of the randomized length of the padding is still visible by

[3] Even though the state is maintained by the protocol and not by the encryption scheme, we still make it explicit in the pseudocode of our attacks.
[4] This is the case for the desktop client and the `tdlib` library.

Algorithm 5.3.7 Pseudocode of the subversion attack from [31]. Here $\mathcal{E}$ denotes encryption scheme that uses n-bit IVs, E is a length-preserving deterministic encryption scheme, and F a PRF with range $\{0,1\} \times [1;\kappa]$, where κ denotes the key size of $\mathcal{E}$. The state σ is constant ($\sigma = \epsilon$), meaning that the attack is stateless.

function $\widetilde{\mathcal{E}^+}_s(K_A, K_E, M, A, \sigma)$
  $j \leftarrow 0$
  do
    $j \leftarrow j + 1$
    $r \twoheadleftarrow_\$ \{0,1\}^n$
    $C \leftarrow \mathcal{E}^+(K_E, A, M, r)$
    $(v, t) \leftarrow F(K_A, C)$
  while $(j < s)$ and $K_E[t] \neq v$
  return (C, ϵ)
end function

function $\widetilde{\mathcal{E}^{\text{EXT}}}((K_A, \mathbf{C}, \mathbf{A}, i))$
  $K \leftarrow 0^\kappa$
  for $i = 1, \ldots, |\mathbf{C}|$ **do**
    $(v, t) \leftarrow F(K_A, C)$
    $K[t] = v$
  end for
  return K
end function

an adversary. It is thus possible to modify Algorithm 5.3.6 in order to exploit these characteristics to transmit 4 bits of key material for each encrypted message, as seen in Algorithm 5.4.1.

Algorithm 5.4.1 Pseudocode of our subversion attack. Here σ denotes the internal counter of the MTProto protocol corresponding to key K that appears in the header of each message (it is not updated by the encryption algorithm), E is a length-preserving deterministic encryption scheme, $\mathcal{E}$ is the MTProto AE scheme, and pad denotes its padding algorithm, as presented in Section 5.1.1.

function $\widetilde{\mathcal{E}^+}(K_A, K, M, \sigma)$
  $Y \leftarrow \mathtt{E}(K_A, K)$
  if $\sigma \leq \lceil |K|/4 \rceil$ **then**
    $\text{len} \leftarrow Y[4\sigma, 4\sigma + 3]$
  else
    $\text{len} \twoheadleftarrow_\$ [0;15]$
  end if
  $M \leftarrow \mathsf{pad}(M, \text{len})$
  $(C, T) \leftarrow \mathcal{E}^+(K, M)$
  return (C, T)
end function

function $\widetilde{\mathcal{E}^{\text{EXT}}}(K_A, \mathbf{C}, \sigma)$
  $r \leftarrow \lceil |K|/4 \rceil$
  $Y \leftarrow ||_{j=0}^{r} \langle |\mathbf{C}[j]|/16 \mod 16 \rangle_4$
  $Y \leftarrow \langle Y \rangle_{|K|}$
  $K \leftarrow \mathtt{E}^-(K_A, Y)$
  return K
end function

The following result provides the strong undetectability and key recovery of the attack above.

Theorem 5.4.1. *Let $q_k \geq \lceil |K|/4 \rceil$, and let $\mathcal{D}$ be an adversary against the strong detectability of*

$\widetilde{\mathcal{E}}$, as defined in Algorithm 5.4.1, that uses at most q queries to at most u users, for a total of at most l bits, and runs in time at most t. Then there exists a distinguisher $\mathscr{D}'$ against the security of $\mathtt{E}$ that uses at most u queries, and runs in time $t + O(l)$, such that:

$$\mathbf{Adv}^{\mathsf{det}}_{\mathcal{E},\widetilde{\mathcal{E}}}(\mathscr{D}) \leq \mathbf{Adv}^{\mathrm{mu-wprf}}_{E}(\mathscr{D}'),$$
$$\mathbf{Adv}^{\mathsf{kr}}_{\mathfrak{M},q_k}\left(\widetilde{\mathcal{E}}\right) = 1.$$

Proof. The only difference between $\mathcal{E}$ and $\widetilde{\mathcal{E}}$ lies in the padding length len used. Consequently, as long as `E` is secure, $\widetilde{\mathcal{E}^+}$ is indistinguishable from $\mathcal{E}^+$ during the execution of `MTProto2.0`. Regarding key recovery, it always succeeds given a sufficient number of ciphertexts, as described in Algorithm 5.4.1. □

5.4.2 Improved Subversion Attack on `MTProto2.0`

Capabilities of Algorithm 5.4.1. A state actor that has subverted the encryption algorithm of a client, and has access to the server-side (encrypted and ordered) transcript of the conversation, will thus be able to recover 4 key bits per message sent between two rekeying steps with probability 1. Unfortunately, this is not sufficient to break the security of `MTProto`. Indeed, each key can only encrypt or decrypt at most 100 messages (as described in Section 5.1.2). This limits the number of key bits that can be targeted to a maximum of 400, and more realistically to around 200 bits (assuming each party sends around 50 messages each). Since `MTProto` keys are much longer, this is not sufficient to allow a realistic guess of the remaining bits. Fortunately, we have only leveraged the randomness length of the padded data, not the randomness used in generating the padding bits. Therefore, we can now mount a variant of the attack from Algorithm 5.4.1, allowing us to send more key bits, albeit at a higher computational cost.

Description of The Improved Attack. The pseudocode for this new attack is presented in Algorithm 5.4.2. Given that the `MTProto` protocol is an authenticated encryption scheme using two passes, we have optimized our attack by applying a PRF `F` with a new adversarial key K'_A to the authentication tag instead of the entire ciphertext. This approach allows us to defer the encryption pass until after the main loop of $\widetilde{\mathcal{E}^+}_{\delta,s}$ has terminated, requiring it to be computed only once, even if the authentication step is repeated s times.

Implications of The Improved Attack. To analyze this new attack, we follow the approach of [31] and introduce the min-entropy $\mathbf{H}_\infty(\mathcal{E}.\mathrm{Auth})$ of the randomized tag generation algorithm $\mathcal{E}.\mathrm{Auth}$.[5] Formally, we define

$$2^{-\mathbf{H}_\infty(\mathcal{E}.\mathrm{Auth})} = \max \Pr\left(\mathcal{E}.\mathrm{Auth}_K(\mathsf{pad}(M, \mathrm{len})) = T\right),$$

where the maximum is taken over all possible keys K, plaintexts M, tag values T, and padding lengths len, and the probability is taken over the uniformly random draw of

[5] This corresponds to the sampling of the padding, and the computation of `F`.

Algorithm 5.4.2 Pseudocode of the updated subversion attack $\widetilde{\mathcal{E}^{+}}_{\delta,s}$. We use the same notation as in Algorithm 5.4.1, and F a PRF with range $\{0,1\}^{\delta}$. Ciphertexts are denoted (C,T) where T refers to the authentication tag. Denote by $\delta' = 4+\delta$.

function $\widetilde{\mathcal{E}^{+}}_{\delta,s}(K_A, K'_A, K, M, \sigma)$
  $Y \leftarrow \mathtt{E}(K_A, K)$
  if $\sigma \leq \lceil |K|/\delta' \rceil$ **then**
    $\text{len} \leftarrow Y[\delta'\sigma, \delta'\sigma+3]$
  else
    $\text{len} \twoheadleftarrow_{\$} [0;15]$
  end if
  $i \leftarrow 0$
  do
    $M' \leftarrow \mathsf{pad}(M, \text{len})$
    $T \leftarrow \mathsf{F}(K, M')$
    $X \leftarrow \mathtt{F}_{K'_A}(T)$
    $b \leftarrow X =_? Y[\delta'\sigma+3, \delta'(\sigma+1)-1]$
    $i \leftarrow i+1$
  while (not b) and $(i < s)$
  $(L, \text{iv}) \leftarrow \mathsf{G}(K, T)$
  $C \leftarrow \mathsf{E}^{+}(L, \text{iv}, M')$
  return (C, T)
end function

function $\widetilde{\mathcal{E}^{\text{EXT}}}_{\delta,s}(K_A, \mathbf{C}, \sigma)$
  $Y \leftarrow \epsilon$
  for $i \in \{1, \ldots, |K|/\delta'\}$ **do**
    $Y \leftarrow Y || \langle |\mathbf{C}[i]|/16 \mod 16 \rangle_4$
    $Y \leftarrow Y || F_{K'_A}(T)$
  end for
  $K \leftarrow \mathtt{E}^{-}(K_A, \langle Y \rangle_{|K|})$
  return K
end function

the (at least 96) padding bits. Our results assume that $2^{-\mathbf{H}_{\infty}(\mathcal{E}.\text{Auth})}$ is small. One has the following result with respect to the detectability and key recovery success of this algorithm.

Theorem 5.4.2. *Let $q \geq \lceil |K_E|/(4+\delta) \rceil$, and let $\mathscr{D}$ be an adversary against the strong detectability of $\widetilde{\mathcal{E}}$, as defined in Algorithm 5.4.2, that uses at most q queries to at most u users, for a total of at most l bits, and runs in time at most t. Then there exists $\mathscr{D}_E$, and $\mathscr{D}_F$, such that*

- *$\mathscr{D}_E$ makes at most 1 query per user to $\mathtt{E}$, u queries in total, and runs in time $t + O(sql)$;*
- *$\mathscr{D}_F$ makes at most sq queries per user to $\mathtt{F}$, and runs in time $t + O(sql)$,*

and

$$\mathbf{Adv}^{\text{det}}_{\mathcal{E},\widetilde{\mathcal{E}}_{\delta,s}}(\mathscr{D}) \leq \mathbf{Adv}^{\text{mu-wprf}}_{E}(\mathscr{D}_E) + \mathbf{Adv}^{\text{mu-prf}}_{F}(\mathscr{D}_F) + \frac{u^2}{2|\mathcal{K}|} + q^2 s^2 2^{-\mathbf{H}_{\infty}(\mathcal{E}.\text{Auth})-1}.$$

Further, there exists $\mathscr{D}'_F$ such that $\mathscr{D}'_F$ makes at most sq queries per user to $\mathtt{F}$, and runs in time $t + O(sql)$, and

$$\mathbf{Adv}^{\text{kr}}_{\mathcal{M},q}\left(\widetilde{\mathcal{E}}_{\delta,s}\right) \geq 1 - \mathbf{Adv}^{\text{mu-prf}}_{F}(\mathscr{D}'_F)$$

$$- q\left(1 - \frac{1}{2^{\delta}}\right)^{s} - q^2 s^2 2^{-\mathbf{H}_{\infty}(\mathcal{E}.\mathrm{Auth})-1}.$$

The proof is very similar to the proofs of [31, Theorems 4.1 and 4.2]. We provide it in two parts for the sake of completeness.

5.4.3 Proving the Strong Undetectability of The Subversion Attack

$\mathrm{Sub}^{\mathrm{det}}_{\mathcal{E},\widetilde{\mathcal{E}}_{\delta,s}}(\mathscr{D})$

1: $u \leftarrow 0$
2: $(K_A, K'_A) \twoheadleftarrow_{\$} \widetilde{\mathcal{K}}$
3: $b \leftarrow \mathscr{D}^{\textsc{Enc}_{\mathrm{Sub}},\textsc{Enc}}$
4: **return** $b = 1$

Oracle $\textsc{New}()$

1: $u \leftarrow u + 1$
2: $\sigma_u \leftarrow 0$
3: $K_u \twoheadleftarrow_{\$} \mathcal{K}$

Oracle $\textsc{Enc}_{\mathrm{Sub}}(M, A, i)$

1: **if** $i \notin [u]$ **then**
2: **return** $\perp$
3: **end if**
4: $Y \leftarrow \mathrm{E}_{K_A}(K_i)$
5: **if** $\sigma_i \leq \lceil |K_i|/\delta' \rceil$ **then** len $\leftarrow Y[\delta'\sigma_i, \delta'\sigma_i + 3]$ **else** len $\twoheadleftarrow_{\$} [0; 15]$
6: $j \leftarrow 0$
7: **do**
8: $M' \leftarrow \mathsf{pad}(M, \mathrm{len})$
9: $(C, T) \leftarrow \mathcal{E}^{+}_{K_i}(M')$
10: $X \leftarrow \mathrm{F}_{K'_A}(T)$
11: $b \leftarrow X = Y[\delta'\sigma_i + 3, (\delta' + 1)\sigma_i - 1]$
12: $j \leftarrow j + 1$
13: **while** $b = 0 \wedge j < s$
14: $\sigma_i \leftarrow \sigma_i + 1$
15: **return** (C, T)

$\boxed{\mathsf{I}_0}$ I_1

1: $u \leftarrow 0$
2: $(K_A, K'_A) \twoheadleftarrow_{\$} \widetilde{\mathcal{K}}$
3: $S \leftarrow \emptyset$
4: $b \leftarrow \mathscr{D}^{\textsc{New},\textsc{Enc}}$
5: **return** $b = 1$

Oracle $\textsc{New}()$

1: $u \leftarrow u + 1$
2: $\sigma_u \leftarrow 0$
3: $K_u \twoheadleftarrow_{\$} \mathcal{K}$
4: $Y_u \twoheadleftarrow_{\$} \{0,1\}^{|K_u|}$
5: **if** $\exists y : (K_u, y) \in S$ **then**
6: $\mathsf{bad} \leftarrow \mathsf{true}$
7: $\boxed{Y_u \leftarrow y}$
8: **end if**
9: $S \leftarrow S \cup \{(K_u, Y_u)\}$

Oracle $\textsc{Enc}(M, A, i)$

1: **return** $\textsc{Enc}_{\mathrm{Sub}}(M, A, i)$

Figure 5.8: *Games* $\mathrm{Sub}^{\mathrm{det}}_{\mathcal{E},\widetilde{\mathcal{E}}_{\delta,s}}(\mathscr{D})$, I_0, *and* I_1 *used in the proof from Section 5.4.3.*

$\boxed{\mathsf{I}_2}$ I_3

$u \leftarrow 0$
$(K_A, K'_A) \twoheadleftarrow_\$ \widetilde{\mathcal{K}}$
$L \leftarrow \emptyset$
$b \leftarrow \mathscr{D}^{\text{New},\text{Enc}}$
return $b = 1$

Oracle New()

$u \leftarrow u + 1$
$\sigma_u \leftarrow 0$
$K_u \twoheadleftarrow_\$ \mathcal{K}$
$Y_u \twoheadleftarrow_\$ \{0,1\}^{|K_u|}$

Oracle Enc(M, A, i)

if $i \notin [u]$ **then**
  return $\perp$
end if
if $\sigma_i \leq \lceil |K_i|/\delta' \rceil$ **then** len $\leftarrow$ $Y[\delta'\sigma_i, \delta'\sigma_i + 3]$ **else** len $\twoheadleftarrow_\$ [0; 15]$
$j \leftarrow 0$
do
  $M' \leftarrow \mathsf{pad}(M, \text{len})$
  $(C, T) \leftarrow \mathcal{E}^+_{K_i}(M')$
  if $\exists x : (T, x) \in L$ **then**
    bad $\leftarrow$ true
    $\boxed{X \leftarrow x}$
  end if
  $L \leftarrow L \cup \{(T, x)\}$
  $b \leftarrow X = Y[\delta'\sigma_i + 3, (\delta' + 1)\sigma_i - 1]$
  $j \leftarrow j + 1$
while $b = 0 \wedge j < s$
$\sigma_i \leftarrow \sigma_i + 1$
return (C, T)

$\text{Re}^{\text{det}}_{\mathcal{E},\widetilde{\mathcal{E}}_{\delta,s}}(\mathscr{D})$

$u \leftarrow 0$
$b \leftarrow \mathscr{D}^{\text{EncSub},\text{Enc}}$
return $b = 1$

Oracle New()

$u \leftarrow u + 1$
$\sigma_u \leftarrow 0$
$K_u \twoheadleftarrow_\$ \mathcal{K}$

Oracle Enc(M, A, i)

if $i \notin [u]$ **then**
  return $\perp$
end if
$(C, T) \leftarrow \mathcal{E}^+_{K_i}(M, \sigma_i)$
$\sigma_i \leftarrow \sigma_i + 1$
return (C, T)

Figure 5.9: *Games* I_2, I_3, *and* $\text{Re}^{\text{det}}_{\mathcal{E},\widetilde{\mathcal{E}}_{\delta,s}}(\mathscr{D})$ *used in the proof from Section 5.4.3.*

The proof begins by introducing intermediate games, as shown in Figure 5.8 and Figure 5.9, that serve as a bridge between the real and ideal world games. By definition of strong undetectability described in Section 5.3, one has

$$\begin{aligned}\mathbf{Adv}^{\mathrm{det}}_{\mathcal{E},\widetilde{\mathcal{E}}}(\mathscr{D}) &\leq \left|\Pr\left(\mathrm{Sub}^{\mathrm{det}}_{\mathcal{E},\widetilde{\mathcal{E}}}(\mathscr{D})\right) - \Pr\left(\mathrm{Re}^{\mathrm{det}}_{\mathcal{E},\widetilde{\mathcal{E}}}(\mathscr{D})\right)\right| \\ &\leq \left|\Pr\left(\mathrm{Sub}^{\mathrm{det}}_{\mathcal{E},\widetilde{\mathcal{E}}}(\mathscr{D})\right) - \Pr\left(\mathsf{I}_0\right)\right| \\ &\quad + \left|\Pr\left(\mathsf{I}_0\right) - \Pr\left(\mathsf{I}_1\right)\right| + \left|\Pr\left(\mathsf{I}_1\right) - \Pr\left(\mathsf{I}_2\right)\right| \\ &\quad + \left|\Pr\left(\mathsf{I}_2\right) - \Pr\left(\mathsf{I}_3\right)\right| \\ &\quad + \left|\Pr\left(\mathsf{I}_3\right) - \Pr\left(\mathrm{Re}^{\mathrm{det}}_{\mathcal{E},\widetilde{\mathcal{E}}}(\mathscr{D})\right)\right|.\end{aligned}$$

The proof consists of upper bounding the probability to transition from one game to the next.

Transition from $\mathrm{Sub}^{\mathrm{det}}_{\mathcal{E},\widetilde{\mathcal{E}}}(\mathscr{D})$ to I_0. The only difference between the two games is that Y is sampled using $\mathtt{E}_{K_A}$ in $\mathrm{Sub}^{\mathrm{det}}_{\mathcal{E},\widetilde{\mathcal{E}}}(\mathscr{D})$, whereas it is sampled using a lazily-sampled random function in I_0. Thus one has

$$\left|\Pr\left(\mathrm{Sub}^{\mathrm{det}}_{\mathcal{E},\widetilde{\mathcal{E}}}(\mathscr{D})\right) - \Pr\left(\mathsf{I}_0\right)\right| \leq \mathbf{Adv}^{\mathrm{mu-wprf}}_{\mathtt{E}}\left(\mathscr{D}_{\mathtt{E}}\right).$$

Transition from I_0 to I_1. Games I_0 and I_1 are identical until there is a collision between the keys of two users. Hence we have

$$\left|\Pr\left(\mathsf{I}_0\right) - \Pr\left(\mathsf{I}_1\right)\right| \leq \Pr\left(\mathsf{I}_1 \text{ sets } \mathsf{bad}\right) \leq \frac{u^2}{2|\mathcal{K}|}.$$

Transition from I_1 to I_2. In game I_2, $\mathtt{F}_{K'_A}$ is replaced with a lazily sampled uniformly random function, which gives

$$\left|\Pr\left(\mathsf{I}_1\right) - \Pr\left(\mathsf{I}_2\right)\right| \leq \mathbf{Adv}^{\mathrm{mu-prf}}_{\mathtt{F}}\left(\mathscr{D}_{\mathtt{F}}\right).$$

Transition from I_2 to I_3. Games I_2 and I_3 are identical until there is a tag collision. Thus

$$\left|\Pr\left(\mathsf{I}_2\right) - \Pr\left(\mathsf{I}_3\right)\right| \leq \Pr\left(\mathsf{I}_3 \text{ sets } \mathsf{bad}\right) \leq q^2 s^2 2^{-\mathbf{H}_\infty(\mathcal{E}.\mathrm{Auth})-1},$$

where $\mathcal{E}.\mathrm{Auth}$ denotes the tag generation algorithm of $\mathcal{E}^+$.

Transition from I_3 to $\mathrm{Re}^{\mathrm{det}}_{\mathcal{E},\widetilde{\mathcal{E}}}(\mathscr{D})$. Finally, the games I_3 and $\mathrm{Re}^{\mathrm{det}}_{\mathcal{E},\widetilde{\mathcal{E}}}(\mathscr{D})$ are identical. Indeed, in game I_3, the condition that stops the while loop is completely independent from the generated ciphertext, which means that the distribution of the outputs of both encryption oracles are completely identical, and

$$\Pr\left(\mathsf{I}_3\right) = \Pr\left(\mathrm{Re}^{\mathrm{det}}_{\mathcal{E},\widetilde{\mathcal{E}}}(\mathscr{D})\right).$$

H_0 $\boxed{\mathsf{H}_1}$

1: $(K, (K_A, K'_A)) \twoheadleftarrow_{\$} \mathcal{K} \times \widetilde{\mathcal{K}}$
2: $\boxed{\Gamma \twoheadleftarrow_{\$} \mathsf{Func}\left(\{0,1\}^*; \{0,1\}^{\delta}\right)}$
3: $\sigma' \leftarrow \epsilon$
4: $r \leftarrow \lceil |K|/\delta' \rceil$
5: $Y \leftarrow \mathtt{E}(K_A, K)$
6: **for** $\sigma \in [1; r]$ **do**
7: $\quad i \leftarrow 0, b \leftarrow 0$
8: $\quad (M, \sigma') \leftarrow \mathfrak{M}(\sigma')$
9: $\quad j \leftarrow 0$
10: $\quad$ **while** $b = 0 \wedge j < \delta$ **do**
11: $\quad\quad \mathsf{len} \twoheadleftarrow_{\$} [0; 15]$
12: $\quad\quad M' \leftarrow \mathsf{pad}(M, \mathsf{len})$
13: $\quad\quad (C, T) \leftarrow \mathcal{E}^+(K_E, M')$
14: $\quad\quad X \leftarrow \mathtt{F}(K'_A, T)$
15: $\quad\quad \boxed{X \leftarrow \Gamma(T)}$
16: $\quad\quad b \leftarrow X = Y[\delta'\sigma + 3, \delta'(\sigma + 1) - 1]$
17: $\quad\quad j \leftarrow j + 1$
18: $\quad$ **end while**
19: **end for**
20: **return** $b = 0$

H_2 $\boxed{\mathsf{H}_3}$

1: $\mathsf{bad} \leftarrow \mathsf{false}$
2: $(K, (K_A, K'_A)) \twoheadleftarrow_{\$} \mathcal{K} \times \widetilde{\mathcal{K}}$
3: $\sigma' \leftarrow \epsilon$
4: $L \leftarrow \emptyset$
5: $r \leftarrow \lceil |K|/\delta' \rceil$
6: $Y \leftarrow \mathtt{E}(K_A, K)$
7: **for** $\sigma \in [1; r]$ **do**
8: $\quad i \leftarrow 0, b \leftarrow 0$
9: $\quad (M, \sigma') \leftarrow \mathfrak{M}(\sigma')$
10: $\quad j \leftarrow 0$
11: $\quad$ **while** $b = 0 \wedge j < \delta$ **do**
12: $\quad\quad \mathsf{len} \twoheadleftarrow_{\$} [0; 15]$
13: $\quad\quad M' \leftarrow \mathsf{pad}(M, \mathsf{len})$
14: $\quad\quad (C, T) \leftarrow \mathcal{E}^+(K_E, M')$
15: $\quad\quad$ **if** $\exists x : (T, x) \in L$ **then**
16: $\quad\quad\quad \mathsf{bad} \leftarrow \mathsf{true}$
17: $\quad\quad\quad \boxed{X \leftarrow x}$
18: $\quad\quad$ **end if**
19: $\quad\quad L \leftarrow L \cup \{(T, X)\}$
20: $\quad\quad b \leftarrow X = Y[\delta'\sigma + 3, \delta'(\sigma + 1) - 1]$
21: $\quad\quad j \leftarrow j + 1$
22: $\quad$ **end while**
23: **end for**
24: **return** $b = 0$

Figure 5.10: *Games* H_0 *to* H_3 *used in the proof from Section 5.4.4.*

5.4.4 Lower Bounding the Probability of Key Recovery

To complete the proof of Theorem 5.4.2, we will establish a lower bound on the success probability of $\widetilde{\mathcal{E}}_{s,\delta}$. To achieve this, we define intermediate games, which are detailed in Figure 5.10. In this proof, we shift our perspective to consider the failure probability instead. The only source of failure arises from the new sampling mechanism for the content of the padding. Therefore, we will replace the sampling of the padding length with a uniformly random draw[6]. Similarly, as in the previous proof we consider the transition probability from one game to the next.

Transition from H_0 to H_1. We start by replacing $\mathtt{F}$ in the pseudocode of $\widetilde{\mathcal{E}}$ by its ideal counterpart. Thus, one has

$$1 - \mathbf{Adv}^{\mathsf{kr}}_{\mathfrak{M},q}\left(\widetilde{\mathcal{E}}_{s,\delta}\right) = \Pr\left(\mathsf{H}_0\right) \leq \Pr\left(\mathsf{H}_1\right) + \mathbf{Adv}^{\mathrm{mu-prf}}_{\mathtt{F}}\left(\mathscr{D}'_{\mathtt{F}}\right),$$

where $\mathscr{D}'_{\mathtt{F}}$ is an adversary against the PRF-security of $\mathtt{F}$ that runs I_0, and replaces the calls to $\mathtt{F}$ by calls to its oracle. Hence, $\mathscr{D}'_F$ issues at most sq queries to its oracle, and runs in time $t + O(sql)$, where l is an upper bound on the number of bits that $\mathfrak{M}(q)$ can output.

Transition from H_1 to H_3. Game H_2 is identical to game H_1 since Γ has been replaced by its equivalent lazy sampling. Finally, game H_3 is identical to game H_2 until there exists a tag repetition, in which case game H_3 breaks the consistency with a simulated random function by sampling a new random value every time. We rely on a fundamental result by Bellare and Rogaway [40], known as the fundamental lemma of game playing, as described in Section 2.4.3. Thus, by Lemma 2.4.4, one has

$$\begin{aligned}
1 - \mathbf{Adv}^{\mathsf{kr}}_{\mathfrak{M},q}\left(\widetilde{\mathcal{E}}_{s,\delta}\right) &= \Pr\left(\mathsf{H}_1\right) + \mathbf{Adv}^{\mathrm{mu-prf}}_{\mathtt{F}}\left(\mathscr{D}'_{\mathtt{F}}\right) \\
&\leq \Pr\left(\mathsf{H}_2\right) + \mathbf{Adv}^{\mathrm{mu-prf}}_{\mathtt{F}}\left(\mathscr{D}_{\mathtt{F}}\right) \\
&\leq \Pr\left(\mathsf{H}_3\right) + \Pr\left(\mathsf{H}_3 \text{ sets } \mathsf{bad}\right) \\
&\quad + \mathbf{Adv}^{\mathrm{mu-prf}}_{\mathtt{F}}\left(\mathscr{D}'_{\mathtt{F}}\right).
\end{aligned}$$

Clearly, the event H_3 sets bad implies that the game has created some tag collision in its at most sq encryption queries. Since $\mathfrak{M}$ simulates a run of the $\mathtt{MTProto}$ protocol, then the inputs to $\mathcal{E}$ are nonce-respecting. Thus, one has

$$\Pr\left(\mathsf{H}_3 \text{ sets } \mathsf{bad}\right) \leq q^2 s^2 2^{-\mathbf{H}_\infty(\mathcal{E}.\mathrm{Auth})-1},$$

where $\mathcal{E}.\mathrm{Auth}$ denotes the tag generation algorithm of $\mathcal{E}^+$. The last step of the proof is to upper bound the probability that H_3 returns 1. In that case, for every σ, the only way for H_3 to return 1 is if every draw of X is not equal to the corresponding bits of Y. Since

[6] We emphasize that this approach is valid because the part of the attack that exploits the randomized length of the padding cannot fail. However, this simplification will not be applicable when considering the strong undetectability of the attack.

a fresh uniformly random X is drawn at every execution of the loop, the probability that the loop continues till $j = \delta$ is equal to $(1 - 2^{-\delta})^s$. Overall, one has

$$1 - \mathbf{Adv}^{\mathsf{kr}}_{\mathfrak{M},q}\left(\widetilde{\mathcal{E}}_{s,\delta}\right) \leq q\left(1 - \frac{1}{2^\delta}\right)^s + q^2 s^2 2^{-\mathbf{H}_\infty(\mathcal{E}.\mathrm{Auth})-1} + \mathbf{Adv}^{\mathrm{mu-prf}}_{\mathrm{F}}\left(\mathscr{D}'_{\mathrm{F}}\right).$$

5.5 Averting Subversion of MTProto2.0

In this section, we discuss three aspects of our subversion attacks described in Section 5.4. First, we analyze the impact of the attack, focusing on the number of bits that can be extracted and the probability of successful extraction. Second, we propose specific instantiations for the components required in our attack. Finally, we demonstrate how MTProto2.0 can be made subversion-resistant through minor modifications.

5.5.1 Impact of our Attack

The attack described in the previous section targets the MTProto AE scheme. However, in the full MTProto protocol, a single key is used for the encryption of at most 100 messages. Thus, the dominating term in the success probability bound is clearly

$$q\left(1 - \frac{1}{2^\delta}\right)^s,$$

where $q \leq 100$. Table 5.1 presents several choices for the parameters δ and s, and provides the associated probability to recover k key bits given the number of victim queries.

Given the huge key size of the MTProto AE scheme and the information provided by Table 5.1, targeting a full key recovery does not feasible. Instead, if the goal of the adversary is to allow decryption of a high percentage of sent messages, one possible choice is to target the part of the MTProto key that is used during the encryption pass, which is 576 bits long. Assuming that the victim sends around 50 messages per key, choosing $\delta = 8$ and $s = 1485$ allows the complete recovery of the encryption key with a probability around 0.85. Of course, this comes at a computational cost: the subverted client will have to repeat the authentication step at most 1485 times. The increased energy consumption may become noticeable. A more modest choice ($\delta = 6$ and $s = 369$) will allow the recovery of most key bits with a high probability, while being computationally cheaper. Note that, even though the authentication pass has to be evaluated at most s times, it is still possible to save the internal state of the SHA-256 hash function after the absorption of the message, and to start the authentication pass from this value for every choice of padding values. This reduces the computational overhead to its minimal value.

Table 5.1: *This table presents an approximated lower bound on the probability to recover k bits of key material with the subversion attack $\widetilde{\mathcal{E}}_{\delta,s}$, under the assumption that the adversary does at least the specified number of queries.*

δ	s	num. of queries	k	success probability
2	21	10	60	≥ 0.97
2	21	50	300	≥ 0.88
2	21	100	600	≥ 0.76
4	91	10	80	≥ 0.97
4	91	50	400	≥ 0.85
4	91	100	800	≥ 0.71
6	369	10	100	≥ 0.97
6	369	50	500	≥ 0.85
6	369	100	1000	≥ 0.70
8	1485	10	120	≥ 0.97
8	1485	50	600	≥ 0.85
8	1485	100	1200	≥ 0.70
10	5946	10	140	≥ 0.97
10	5946	50	700	≥ 0.85
10	5946	100	1400	≥ 0.70

Our attack requires access to a reliable counter, which is provided by the `MTProto` protocol in the case of secret chats. However, client-server chats do not offer such a convenient counter. To address this issue, we propose the following workarounds:

- Our algorithm can be made stateful, at the cost of making it detectable through a simple state reset.
- A randomized encryption scheme can be used instead of the length-preserving scheme `E` when computing Y in Algorithm 5.4.2. After each state reset, a new IV would be generated uniformly at random and stored as a state, along with a counter. Although this approach would reduce the number of sent key bits (due to the need to send the IV bits and the fact that after resetting, the attacker will start sending the same key bits a second time), it would also mitigate the impact of state resets.
- Since the client-server key is long-lived, we can afford to transmit key bits very slowly; hence, Algorithm 5.3.7 can be used.

5.5.2 Instantiating `F` and `E`

Our subverted algorithm(s) require a length-preserving encryption scheme and a PRF `F`. Here, we briefly discuss possible instantiation for these components.

Choice of `F`: From Table 5.1, we observe that efficient instances of our attacks set $\delta \leq 10$. So, we can simply reuse the `AES-256` block cipher and truncate the output to

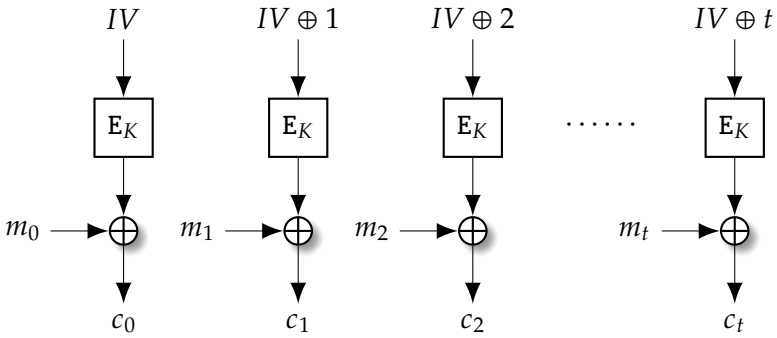

Figure 5.11: *Counter mode (CTR) [185].*

δ bits. Since the output size δ is quite small, for all practical purposes, we can simply assume that truncated `AES-256` is a perfect random function.

Choice of `E`: One could consider using a wide-block cipher or format-preserving encryption, where the output and input maintain the same format, to meet the wPRF security requirement. However, these schemes lack flexibility as they require the entire ciphertext (encrypted K) for correct decryption. Upon closer examination, it becomes clear that all we need is a sufficiently long key stream to mask K. Therefore, an efficient stream cipher or block key stream generator, such as the Counter mode (`CTR`), is sufficient. `CTR` mode turns a block cipher into a stream cipher by encrypting successive counter values (see also Figure 5.11). Unfortunately, these methods require a seed, making them susceptible to detection via state reset. Instead, we instantiate E with an efficient *online encryption* scheme [24].

An online encryption scheme `E'` is a length-preserving encryption scheme that satisfies the online property: for any key K_A, an input x is a prefix of another input x' if and only if $\texttt{E}'K_A(x)$ is a prefix of $\texttt{E}'K_A(x')$. Essentially, the online property implies that encryption and decryption can be performed on-the-fly, i.e., as soon as a full block[7] of ciphertext is available, it can be decrypted based on the previous ciphertext blocks. Thus, the online cipher offers an efficient trade-off between a format-preserving cipher and a stream cipher. Furthermore, `E'` behaves similarly to a uniform random function as long as its inputs do not share any prefixes. Consequently, $\texttt{E}'(K)$ will be uniformly random and independent across different sessions, provided the first block of K does not collide across sessions.

Fortunately, one of the components in `MTProto` satisfies the online property. `IGE` with a fixed IV value acts as a secure online cipher in the known plaintext setting. Within `E`, `IGE` can be instantiated with `AES-256` to directly reuse the `MTProto` components, or with a smaller block cipher, such as a 64-bit block, to minimize the amount of ciphertext needed for correct decryption of any ciphertext bit. Let E be the block cipher used within `E`. It can be shown that `E` can be instantiated with `IGE` at the cost of $O\left(u^2/2^n\right)$ plus the mu-PRP of E, where n denotes the block size of $\widetilde{\texttt{E}}$. Indeed, the proof for this is

[7] Here, block refers to the underlying block cipher of `E'`.

straightforward, with the bound $O\left(u^2/2^n\right)$ arising from two cases. First, the term $u^2/2^n$ accounts for the probability that the first n bits of keys across two distinct sessions (or users) collide. Second, `IGE` behaves as a secure online cipher in the known plaintext setting, up to a cost of $c^2u^2/2^n$, where $c = \lceil |K|/n \rceil$ is a small constant.

5.5.3 Subversion-Resistant `MTProto`: `MTProto-D`

The primary vulnerability of the protocol lies in its randomized padding algorithm. However, merely altering the padding length randomization is insufficient, as the padding value can still be used to transmit δ bits of data per ciphertext (instead of $4 + \delta$ bits in Algorithm 5.4.2). A potential countermeasure is to derive the padding values deterministically from the secret key (and possibly the encrypted message) and verify these values as the final step of the decryption process. This approach would eliminate the ability of a subverted algorithm to manipulate the content and length of the padding, thereby preventing our attack. In this section, we describe such a method and demonstrate that the modified padding algorithm is indeed subversion-resistant.

Unique Ciphertexts. It is well-known [11, 111] that perfect decryptability is a necessary condition for any symmetric-key encryption scheme to be subversion-resistant. While perfect decryptability is a theoretical requirement that is challenging to achieve in practical settings, attacks targeting decryption algorithms are generally inefficient in practice. Thus, we assume perfect decryptability in our discussion. Additionally, Degabriele et al. highlighted input-triggered subversion [111], where adversaries exploit ambiguities in message language to construct attacks. Therefore, it is crucial to ensure independence between the keys and the message distribution. We assume that messages (including protocol parameters) are sampled independently of the keys.

Formally, we define the notion of *unique ciphertexts* for a DAE scheme.

Definition 5.5.1 (Unique Ciphertexts). *A $(\mathcal{K}, \mathcal{A}, \mathcal{M}, \mathcal{T})$-deterministic authenticated encryption scheme $\mathcal{E}$ has unique ciphertexts if for any tuple $(K, A, M) \in \mathcal{K} \times \mathcal{A} \times \mathcal{M}$ the set of all ciphertexts (C, T) such that $M = \mathcal{E}^{-}(K, A, C, T)$ is of size at most one.*

Note that Definition 5.5.1 is closely related to the context commitment notion `CMT-4` defined in Section 6.2. The unique ciphertext property implies that each context is committing. In [36], Bellare et al. showed that the unique ciphertext property is sufficient for subversion resistance in the context of algorithm substitution attacks. Therefore, to achieve subversion resistance for `MTProto2.0`, we need to ensure it possesses the unique ciphertext property.

Changes in `MTProto2.0`: `MTProto-D`. We reuse the notations Section 5.1. We define a modified protocol called `MTProto-D`, by making four small changes in the definition of `MTProto2.0`:

1. First, we redefine $\mathsf{F}_{k_1}(x) := \mathsf{chop}_{\tau+4}(\texttt{SHA-256}(k_1 \| x))$. Specifically, we extract an extra 4 bits at the tag generation stage which will be used later in padding algorithm.
2. Second, we redefine the protocol enriched message **random_bytes** $= 0^{128}$. This is done specifically to avoid adversary's control on the payload.
3. Third, we define $(t, \ell) = \mathsf{F}_{k_1}(\mathtt{X})$, where $|t| = \tau$ and $|\ell| = 4$.
4. Fourth, we change the padding algorithm by
   (a) first, redefining **Rand**$(\mathtt{X}) := \ell$, where ℓ is viewed as a 4-bit integer value, i.e., $\ell \in \{0, \ldots, 15\}$. Since, F is a secure PRF, we can assume that for all practical purposes ℓ is uniform at random. So all we have done is eliminate the adversary's influence over the padding length, under the assumption of independence of message distribution.
   (b) second, sampling the **random_padding** in either one of the following way:
      - Set **random_padding** $:= 10^{d-1}$, where d is the length of the padding; or
      - Set **random_padding** $:= \mathsf{F}_{K'}(\mathtt{X})$ for some secure PRF F and a key K' independent of the other keys.

In terms of security, MTProto-D loses at most 4 bits of security since we now release an additional 4 bits via F. Additionally, although the tag generation is performed only over the protocol-enriched message, i.e., $\mathtt{X}$, this does not compromise the security of the protocol. This is because the random padding is either set to a constant value or is fully dependent on $\mathtt{X}$. In both scenarios, the decryption of ciphertext blocks corresponding to the padding bits must either conform to a specific form or follow an exact deterministic relation with $\mathtt{X}$, which serves as a verification step for the padding value. For simplicity, we assume that padding follows **random_padding** $:= 10^{d-1}$.

In Theorem 5.5.1, we show that MTProto-D is a subversion-resistant algorithm under the assumption of perfect decryptability and an independent message distribution.

Theorem 5.5.1. *Suppose the message distribution $\mathcal{M}$ is independent of keys and perfect decryptability holds. Then,* MTProto-D *is subversion-resistant in context of algorithm substitution attacks.*

Proof. First, note that once a message is fixed, the padding length is also fixed (since MTProto-D is deterministic), and consequently, the corresponding ciphertext length is fixed. Thus, for any triple (K, A, M), the corresponding ciphertexts (C, T), as defined in Definition 5.5.1, will all have equal lengths (At this point, it is not yet evident that there is a unique such ciphertext). Next, we demonstrate that there is only one such ciphertext. Indeed, for each message M, the ciphertext is generated over $M|10^{d-1}$ for a fixed d (which depends only on K and M). The bijectivity property of IGE ensures that only one such ciphertext can exist. □

6

Context-committing Security of Authenticated Encryption Schemes

In this chapter, we explore the development and security challenges of Authenticated Encryption with Associated Data (AEAD) schemes, which are crucial for modern security applications. AEAD schemes like `GCM` , `Ascon` , and `Deoxys` address various security goals, including Nonce-based AEAD, Misuse-resistant AEAD, and Deterministic AEAD. However, as these schemes evolve, new attack vectors emerge, necessitating ongoing research in areas such as leakage-resilient AEAD, which focuses on maintaining security despite potential leaks of sensitive information, and context-committing AEAD, which addresses vulnerabilities related to ciphertexts that can be decrypted under multiple contexts.

The intersection of leakage-resilient and context-committing AEAD schemes is particularly intriguing. Recent studies have shown that while traditional AEAD constructions like Encrypt-then-MAC or MAC-then-Encrypt may not inherently offer commitment properties, approaches like Encrypt-and-MAC can achieve them under certain conditions. This chapter introduces a blueprint for designing AEAD schemes that are both leakage-resilient and context-committing, focusing on single-pass implementations. The blueprint demonstrates that by carefully selecting cryptographic primitives and proving the collision resistance of key components, it's possible to create schemes that meet both security goals, providing a robust framework for future AEAD development. For a more comprehensive overview of the topic, we refer the reader to Section 1.6.2.

6.1 Context Committing Blueprint for Single-pass Schemes

In this section, we describe a paradigm for designing leakage-resilient schemes based on leveled implementations. In this approach, different components of the scheme operate under varying assumptions regarding their implementation and the associated leakage functions. Given our focus on the relationship to context commitment, we examine only integrity and Ciphertext Integrity with Misuse and Leakage under decryption leakage (CIML2). Additionally, we consider a common leakage model where the adversary can

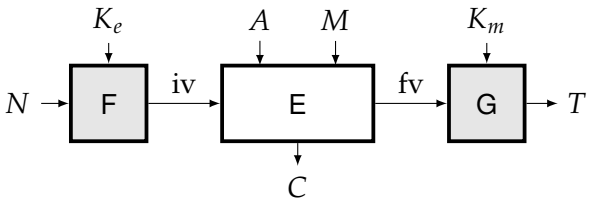

Figure 6.1: *The* KET *blueprint for single-pass leveled leakage-resilient context-committing AEAD. The gray components are assumed to be strongly protected.*

obtain unlimited leakage. In this model, most parts of the scheme are unprotected, with only specific components assumed to be leak-free.

We introduce a single-pass blueprint similar to the Encrypt-then-MAC (EtM) scheme (for more details see the N2 construction from [302]), where the design includes leak-free Key Derivation (KDF) and Tag Generation (TGF) functions, alongside an encryption function that allows for unlimited leakage. We denote this blueprint as Key Derivation, Encryption, and Tag Generation (KET), which is defined as follows (see also Figure 6.1).

Definition 6.1.1 (KET Blueprint)**.** *A nonce-based AEAD, denoted by $\mathcal{E}$, is said to follow the* KET *blueprint if there exists four component functions:*

1. *Key Generation Function* $\mathsf{Gen} : \mathcal{K} \to \mathcal{K}_\mathsf{F} \times \mathcal{K}_\mathsf{G}$;
2. *Key Derivation Function* $\mathsf{F} : \mathcal{K}_\mathsf{F} \times \mathcal{N} \to \mathcal{R}$;
3. $(\mathcal{R}, \mathcal{A}, \mathcal{M})$*-encryption scheme* E *with ciphertext space* $\mathcal{C} \times \mathcal{F}$;
4. *Tag Generation Function* $\mathsf{G} : \mathcal{K}_\mathsf{G} \times \mathcal{C} \to \mathcal{T}$,

such that for any master key $K \in \mathcal{K}$ *and context* $(N, A, M) \in \mathcal{N} \times \mathcal{A} \times \mathcal{M}$*, one has*

$$(K_e, K_m) = \mathsf{Gen}(K), \qquad \mathrm{iv} = \mathsf{F}(K_e, N)$$
$$(C, \mathrm{fv}) = \mathsf{E}^+(\mathrm{iv}, A, M), \qquad T = \mathsf{G}(K_m, \mathrm{fv}),$$

and additionally

$$\mathcal{E}^+_{(K_e,K_m)}(N, A, M) = (C, T), \qquad \mathcal{E}^-_{(K_e,K_m)}(N, A, C, T) = M.$$

Security Requirements. We can identify several requirements on the components. CIML2 security requires that the F is collision-resistant for a given key K. Similarly as EtM, this means that the scheme commits to (N, A, M) for K. As with EtM, further issues arise when considering the keys. If the F is not collision-resistant, then we can find $K_{m_1} \neq K_{m_2}$ such that $T_1 = T_2$ for the same (N, A, M). The commitment will break if K_m is independent of K_e. However, if K_m depends on K_e, the success of the attack depends on the properties of the F and the interaction between the F and the Efunction. Alternatively, it may be possible to relax the requirements on the F if the G is collision-resistant.

6.2 Context Committing Security of KET Schemes

In this section, we establish the security for the KET blueprint. KET can be seen as the paradigm underlying single-pass leveled leakage-resilient schemes such as Triplex [292] or Multiplex [265]. We start with the formal definition for context commitment.

CMT-4 Security. Recall that the two prevalent notions of committing security in the literature are CMT-1 security: A commitment to only the key K and CMT-4 security: A commitment to the complete context (K, N, A, M). In this chapter, we consider CMT-4 security and define it formally. Note that Bellare and Hoang [30] demonstrated that incorporating the message M into the context is unnecessary, as committing to (K, N, A) is equivalent to committing to (K, N, A, M). The CMT-4 game against an AEAD scheme $\mathcal{E}$ is defined as follows:

- An adversary $\mathscr{A}$ outputs a pair of commitments $(K_1, N_1, A_1, M_1) \neq (K_2, N_2, A_2, M_2)$;
- $\mathscr{A}$ wins $\Leftrightarrow \mathcal{E}^+(K_1, N_1, A_1, M_1) = \mathcal{E}^+(K_2, N_2, A_2, M_2)$.

We write $\mathbf{Adv}^{\text{cmt}-4}_{\mathcal{E}}(\mathscr{A})$ to denote the probability that $\mathscr{A}$ wins the CMT-4 game where $\mathscr{A}$ has access to the ideal primitives and hash keys used by $\mathcal{E}$.

Overview of Security Proofs. In this section, we establish three goals. First, we show that the KET composition is CMT-4 secure when each component satisfies a specific set of collision-resistance properties. Second, we show that it can achieve compact commitment, where verifying the tag is sufficient to verify the commitment. Finally, we prove that if the keys used in the first and last components are identical (or generated by an algorithm with specific collision resistance properties), we can relax the collision-resistance requirements for certain components.

Notional Setup. From this point onward, we assume that an adversary $\mathscr{A}$ outputs challenge values (N, A, M, K) and (N', A', M', K'), where $(K_{m_1}, K_{e_1}) = \mathtt{Gen}(K)$ and $(K'_{m_1}, K'_{e_1}) = \mathtt{Gen}(K')$, resulting in the corresponding ciphertexts (C, T) and (C', T'). Furthermore, we denote the intermediate values for each invocation by $(\text{iv}_1, \text{fv}_1)$ and $(\text{iv}_2, \text{fv}_2)$. Our focus will be on the probability that $(C, T) = (C', T')$.

6.2.1 CMT-4 Security of the Generic KET scheme

We begin with the generic KET scheme wherein the keys in F and G are independent, i.e. no constraints are imposed on their keys. For such schemes, we show that achieving CMT-4 security requires collision resistance in all three components, F, E, and G, with the minor relaxation that we require only right collision resistance for E, i.e. collision resistance for the part of its outputs that are used in the G. We study four relevant variants of KET which differ in their assumptions posed on their individual components. The variants KET-1 and KET-1a follow the KET blueprint as defined in Definition 6.1.1,

	Component				
Scheme	Gen	F	E	G	Result
KET-1	-	CR	RCR	CR	
KET-1a	LCR	CR	RCR	partial CR	
KET-2	RCR	-	RCR	CR	
KET-2a	RCR	CR	partial RCR	CR	

Table 6.1: *Different variants of KET and the requirements on their components for CMT-4 security. RCR/CR = (right) collision resistance.*

while the variants KET-2 and KET-2a will follow a tweaked definition of KET, denoted by KET' and defined in Definition 6.2.1. Table 6.1 summarizes their properties.

Theorem 6.2.1. *Let $\mathcal{E}$ be a nonce-based AEAD based on the KET blueprint then there exists $\varepsilon_1, \varepsilon_2, \varepsilon_3 \in \mathbb{R}$ and positive integers $t_1, t_2, t_3 \in \mathbb{N}$ such that: F is (ε_1, t_1)-CR, E is (ε_2, t_2)-RCR and G is (ε_3, t_3)-CR. Then, for any t-bounded adversary $\mathcal{A}$ against $\mathcal{E}$, one has*

$$\mathbf{Adv}^{\text{cmt}-4}_{\mathcal{E}}(\mathcal{A}) \leq \varepsilon_1 + \varepsilon_2 + \varepsilon_3, \tag{6.1}$$

where $t_1, t_2, t_3 = O(t)$.

Proof. Since there is no restriction on Gen, the CMT-4 security for the context (K, N, A, M) follows from the CMT-4 security of the context (K_m, K_e, N, A, M). We consider the following disjoint cases that cover all possibilities. First note that if $K_{m_1} \neq K_{m_2}$, there must be a collision in G. Distinguish between two cases.

1. If $K_{m_1} = K_{m_2}$, $(A_1, M_1) \neq (A_2, M_2)$, then we have two sub cases:
   (a) If $\text{fv}_1 \neq \text{fv}_2$, there must be a collision in G.
   (b) Otherwise, there must be a right-collision in E.
2. Otherwise, $(K_{e_1}, N_1) \neq (K_{e_2}, N_2)$. Then there can be three sub cases:
   (a) If $\text{fv}_1 \neq \text{fv}_2$, there must be a collision in G.
   (b) If $\text{fv}_1 = \text{fv}_2$ and $\text{iv}_1 \neq \text{iv}_2$, there must be a right-collision in E.
   (c) Otherwise, there must be a collision in F.

For each of the cases above, the adversary's advantage is bounded by the collision resistance properties of the individual components, as detailed in Equation 6.1. □

6.2.2 CMT-4 Security of KET-1a, KET-2 and KET-2a

CMT-4 Security of KET-1a. Theorem 6.2.1 does not require any restriction on Gen and holds even when the keys K_e and K_m are independent. However, if Gen is left-collision-resistant, a left collision resistance on the values fv of G will suffice.

Theorem 6.2.2. *Let $\mathcal{E}$ be a nonce-based AEAD based on the* KET *blueprint then there exists $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4 \in \mathbb{R}$ and positive integers $t_1, t_2, t_3, t_4 \in \mathbb{N}$ such that:* Gen *is (ε_1, t_1)-LCR,* F *is (ε_2, t_2)-CR,* E *is (ε_3, t_3)-CR and* G *is (ε_4, t_4)-partial-CR only restricted to input* fv. *Then, for any t-bounded adversary $\mathcal{A}$ against $\mathcal{E}$, one has*

$$\mathbf{Adv}^{\mathrm{cmt}-4}_{\mathcal{E}}(\mathcal{A}) \leq \varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4, \tag{6.2}$$

where $t_1, t_2, t_3, t_4 = \mathcal{O}(t)$.

Proof. The case-based analysis is similar to Theorem 6.2.1, except for the case where $(A_1, M_1) = (A_2, M_2)$ and $(K_1, N_1) \neq (K_2, N_2)$. Since Gen is LCR, the probability of $K_1 \neq K_2$ and $K_{e_1} = K_{e_2}$ is bounded by ε_1. Using a standard hybrid argument to account for ε_1, we can safely assume $K_1 \neq K_2$ and therefore $K_{e_1} \neq K_{e_2}$ in the remainder. Thus, this case reduces to $(K_{e_1}, N_1) \neq (K_{e_2}, N_2)$, and the rest of the proof follows the same steps as in Theorem 6.2.1. □

CMT-4 Security of KET-2. Next, we consider variants of KET that use the nonce as an additional input of the encryption function. For those variants, collision resistance of the F is not necessary. This is intuitive since we can view the next scheme as KET-1a where N is appended to the output of the E.

Definition 6.2.1 (KET′ Blueprint)**.** *A nonce-based AEAD, denoted $\mathcal{E}$, is said to follow the* KET′ *blueprint if there exists four component functions:*

1. *Key Generation Function* $\mathsf{Gen} : \mathcal{K} \to \mathcal{K}_\mathsf{F} \times \mathcal{K}_\mathsf{G}$;
2. *Key Derivation Function* $\mathsf{F} : \mathcal{K}_\mathsf{F} \times \mathcal{N} \to \mathcal{R}$;
3. $(\mathcal{R}, \mathcal{A}, \mathcal{N}, \mathcal{M})$*-encryption scheme* E *with ciphertext space* $\mathcal{C} \times \mathcal{F}$;
4. *Tag Generation Function* $\mathsf{G} : \mathcal{K}_\mathsf{G} \times \mathcal{C} \to \mathcal{T}$,

such that for any master key $K \in \mathcal{K}$ and context $(N, A, M) \in \mathcal{N} \times \mathcal{A} \times \mathcal{M}$, one has

$$(K_e, K_m) = \mathsf{Gen}(K), \qquad \mathrm{iv} = \mathsf{F}(K_e, N)$$
$$(C, \mathrm{fv}) = \mathsf{E}^+(\mathrm{iv}, A, N, M), \qquad T = \mathsf{G}(K_m, \mathrm{fv}),$$

and additionally

$$\mathcal{E}^+_{(K_e,K_m)}(N, A, M) = (C, T), \qquad \mathcal{E}^-_{(K_e,K_m)}(N, A, C, T) = M.$$

Theorem 6.2.3. *Let $\mathcal{E}$ be a nonce-based AEAD based on the* KET′ *blueprint then there exists $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4 \in \mathbb{R}$ and positive integers $t_1, t_2, t_3, t_4 \in \mathbb{N}$ such that:* Gen *is (ε_1, t_1)-RCR,* E *is (ε_2, t_2)-RCR and* G *is (ε_3, t_3)-CR. Then, for any t-bounded adversary $\mathcal{A}$ against $\mathcal{E}$, one has*

$$\mathbf{Adv}^{\mathrm{cmt}-4}_{\mathcal{E}}(\mathcal{A}) \leq \varepsilon_1 + \varepsilon_2 + \varepsilon_3, \tag{6.3}$$

where $t_1, t_2, t_3 = \mathcal{O}(t)$.

Proof. This proof follows the same argumentation as Theorem 6.2.1 and Theorem 6.2.2. Distinguish between a few cases.

1. If $K_1 \neq K_2$, there are two sub cases. If $K_{m_1} = K_{m_2}$ then Gen has a a right-output collision. Otherwise, there is a collision in G.
2. If the keys are distinct then $(N_1, A_1, M_1) \neq (N_2, A_2, M_2)$ and here too there are two sub cases. If $\mathrm{fv}_1 \neq \mathrm{fv}_2$ then there is a collision in G and otherwise there is a right-output collision in E.

□

CMT-4 Security of KET-2a. Finally, we consider a special case of KET-2 that we call KET-2a, where E is only collision-resistant when iv, A, or C change, i.e. it may be easy to find (iv, A, C, N_1) and (iv, A, C, N_2) such that $\mathrm{fv}_1 = \mathrm{fv}_2$. However, if $(\mathrm{iv}_1, A_1) \neq (\mathrm{iv}_2, A_2)$, then collisions are hard to find. The following theorem demonstrates that, despite this restrictive assumption on the collision resistance of E, we can still attain CMT-4 security by imposing a milder condition. In this case, it is essential to also assume that F is collision-resistant.

Theorem 6.2.4. *Let $\mathcal{E}$ be a nonce-based AEAD based on the KET′ blueprint then there exists $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4 \in \mathbb{R}$ and positive integers $t_1, t_2, t_3, t_4 \in \mathbb{N}$ such that:* Gen *is (ε_1, t_1)-RCR,* F *is (ε_2, t_2)-CR,* E *is (ε_3, t_3)-RCR, i.e,* E *is right collision resistant only on input* (iv, A)*, and* G *is (ε_4, t_4)-CR. Then, for any t-bounded adversary $\mathscr{A}$ against $\mathcal{E}$, one has*

$$\mathbf{Adv}^{\mathrm{cmt}-4}_{\mathcal{E}}(\mathscr{A}) \leq \varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4, \tag{6.4}$$

where $t_1, t_2, t_3, t_4 = O(t)$.

Proof. Distinguish between a few cases.

1. If $K_1 \neq K_2$, there are two sub cases. If $K_{m_1} = K_{m_2}$ then Gen has a a right-output collision. Otherwise, there is a collision in F.
2. If the keys are distinct then $(N_1, A_1, M_1) \neq (N_2, A_2, M_2)$ and If $\mathrm{fv}_1 \neq \mathrm{fv}_2$ then there is a collision in G. Otherwise, $\mathrm{fv}_1 = \mathrm{fv}_2$ and we distinguish between three sub cases.
   (a) If $\mathrm{iv}_1 \neq \mathrm{iv}_2$ there is a (iv, A)-right-output collision on E.
   (b) Else if $\mathrm{iv}_1 = \mathrm{iv}_2$ but $N_1 \neq N_2$ then there is collision on F.
   (c) Otherwise, since $(\mathrm{iv}_1, \mathrm{iv}_2) = (N_1, N_2)$ then by by injectivity of Eover the message space we must have $A_1 \neq A_2$. This implies that there is a right-output collisions on inputs (iv, A) of E.

□

6.3 Triplex as an Instantiation of KET-2

In this section, we illustrate the practicality of the KET blueprint by demonstrating that the recent single-pass scheme Triplex [292] can be interpreted as a specific case of KET-2. We will use the same notations for consistency: F for the Key Derivation Function, E for encryption, and G for the Tag Generation function.

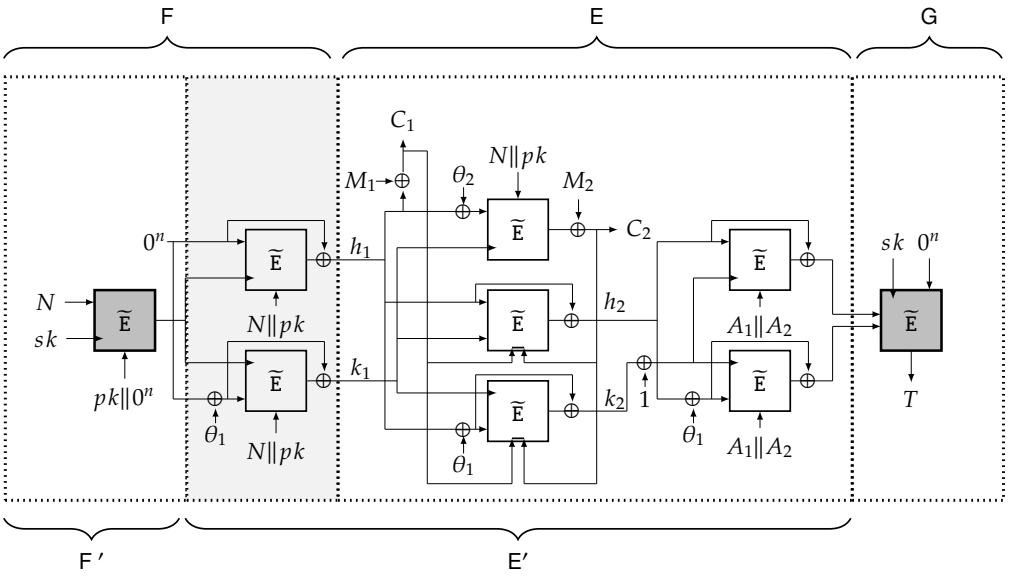

Figure 6.2: *Encryption with* `Triplex`.

`Triplex`. In `Triplex`, the function F consists of three TBCs, a protected call followed by two parallel calls to an unprotected TBC. It takes a key $K = sk\|pk$ – that combines a secret part sk with a public part pk for higher multi-user security – and a nonce N and produces a $2n$-bit output $\text{iv} = h_1\|k_1$. The encryption function of `Triplex` takes various inputs including pk, N, A, M, and $\text{iv} = h_1\|k_1$, and outputs a ciphertext C along with $\text{fv} = V\|W$. Its Tag Generation Function is essentially a single TBC call. It takes sk as the key, $V|W$ as the tweak, and a fixed input 0^n to generate a tag T. Note that both F and Etake N as input and both the F and the G use the same key sk.

There are multiple ways to view `Triplex`, and each leads to the application of a different theorem. We will view the CMT-4 security of `Triplex` as an application of Theorem 6.2.3. We can consider pk as part of the nonce instead of the key since it is not utilized as a key anywhere. This simplification allows us to view `Triplex` as a specific instance of the generic `KET-2` construction. According to Theorem 6.2.3, for achieving CMT-4 security, we need to demonstrate collision resistance of G and right-output collision resistance of E.

Security of `Triplex`. Before presenting the security analysis of `Triplex`, we first highlight that its Tag Generation Function bears a resemblance to that of the MAC `LRMAC1` [45]. The latter is defined as follows: let $\mathtt{H} : \mathcal{M} \to \mathcal{T}$ be a hash function and $\widetilde{\mathtt{E}} : \mathcal{K}_m \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ be a tweakable block cipher (TBC). Given $K_m \in \mathcal{K}_m$ and a message $M \in \mathcal{M}$, `LRMAC1` computes the authentication tag using as

$$T \leftarrow \widetilde{\mathtt{E}}(K_m, \mathtt{H}(M), 0^n).$$

The collision-resistance game for `LRMAC1` is defined as follows: the adversary $\mathscr{A}$ receives the hash-function key K_h at the start of the game. Throughout the game, $\mathscr{A}$ makes q_e

chosen-key queries to the ideal-cipher oracle $\widetilde{\mathrm{E}}$ and receives the corresponding outputs. If a query (K_i, X_i) is in the forward direction, $\mathscr{A}$ receives $Y_i \leftarrow \widetilde{\mathrm{E}}_{K_i}(X_i, 0^n)$. If a query (K_i, Y_i) is in the backward direction, $\mathscr{A}$ receives $X_i \leftarrow \widetilde{\mathrm{E}}^{-1}K_i(Y_i, 0^n)$. After completing its interactions, $\mathscr{A}$ outputs two distinct pairs $(Km_1, M_1) \neq (K_{m_2}, M_2)$. The adversary wins if and only if $\mathtt{LRMAC1}[\mathtt{H}, \widetilde{\mathrm{E}}](K_{m_1}, M_1) = \mathtt{LRMAC1}[\mathtt{H}, \widetilde{\mathrm{E}}](K_{m_2}, M_2)$.

The security of the Tag Generation Function of LRMAC1 within the collision-resistance game defined above is established by Lemma 6.3.1. This lemma will be utilized in the security analysis of Triplex.

Lemma 6.3.1. *Let $\widetilde{E} : \mathcal{K}_m \times \mathcal{T} \times \{0,1\}^n \rightarrow \{0,1\}^n$ be an ideal cipher and $H : \mathcal{M} \rightarrow \mathcal{T}$ be a (ε_1, t_1)-collision-resistant hash function. Then, for any t-bounded adversary $\mathscr{A}$ making at most $q_e \leq 2^{n-1}$ queries to the ideal cipher $\widetilde{E}$ as defined in the game above, where $t_1 = O(t + q_e)$, $\mathtt{LRMAC1}[H, \widetilde{E}]$ is (ε, t)-collision-resistant, with*

$$\varepsilon \leq \varepsilon_1 + \frac{q_e^2 + 2q_e + 5}{2^n}. \tag{6.5}$$

Proof. Suppose $\mathscr{A}$ outputs $(K_{m_1}, M_1) \neq (K_{m_2}, M_2)$ such that $\mathtt{LRMAC1}[\mathtt{H}, \widetilde{\mathrm{E}}](K_{m_1}, M_1) = \mathtt{LRMAC1}[\mathtt{H}, \widetilde{\mathrm{E}}](K_{m_2}, M_2)$. We define a sequence of hybrid games and denote by E_i the event corresponding to when the adversary wins in game G_i.

- *Game G_0:* The real-world game.
- *Game G_1:* The game terminates if one of the following events happens during the ideal-cipher queries.
  - Two forward queries with different keys produce the same output. This probability is bounded by $q_e^2/2^n$.
  - A backward query with input T outputs 0^n. This probability is bounded by $2q_e/2^n$.

  It follows that
  $$|\Pr(\mathsf{E}_0) - \Pr(\mathsf{E}_1)| \leq \frac{q_e^2 + 2q_e}{2^n}.$$
- *Game G_2:* Game G_2 is almost identical to G_1 but adds the fact that G_2 will terminate if $M_1 \neq M_2$ and $\mathtt{H}(M_1) = V_1 = V_2 = \mathtt{H}(M_2)$. The probability of this event is bounded by
  $$|\Pr(\mathsf{E}_1) - \Pr(\mathsf{E}_2)| \leq \varepsilon_1.$$

  Finally, we study the probability that $\mathscr{A}$ wins in G_2. It suffices to consider the case where $V_1 \neq V_2$ as if $V_1 = V_2$ it leads to the termination. Distinguish between a few sub cases.
  - If $(K_{m_1}, V_1, 0^n, T)$ appeared in an ideal-cipher query then
    $$\Pr\left(\widetilde{\mathrm{E}}(K_{m_2}, V_2, 0^n) = T\right) \leq \frac{1}{2^n - q_e} \leq \frac{2}{2^n},$$
  - Else if $(K_{m_2}, V_2, 0^n, T)$ appeared in an ideal-cipher query then
    $$\Pr\left(\widetilde{\mathrm{E}}(K_{m_1}, V_1, 0^n) = T\right) \leq \frac{1}{2^n - q_e} \leq \frac{2}{2^n},$$

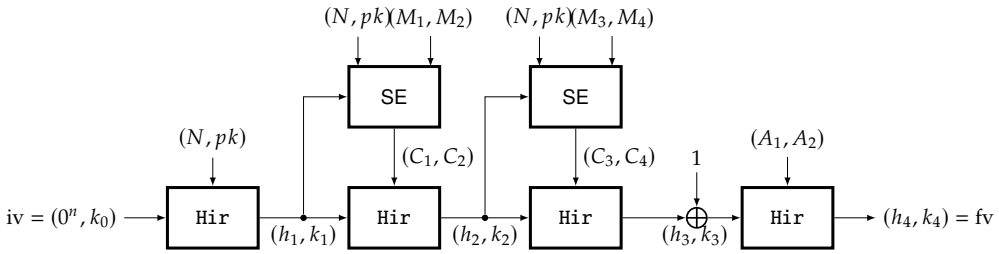

Figure 6.3: *Alternative visualization of the modified encryption function* E′ *of* `Triplex`.

– Otherwise,

$$\Pr\left(\widetilde{\mathsf{E}}(K_{m_1}, V_1, 0^n) = \widetilde{\mathsf{E}}(K_{m_2}, V_2, 0^n)\right) \leq \frac{1}{2^n}.$$

Thus, $\Pr(\mathsf{E}_2) \leq 5/2^n$. Finally, the upper bound in (6.5) is given by adding all the transition probabilities.

□

Now we are ready for the security of `Triplex` given in Corollary 6.3.1.

Corollary 6.3.1. *Let $\mathcal{E}$ be a nonce-based AEAD based on the* `KET′` *blueprint then there exists an (ε, t_1)-collision-resistant hash function* `H` *such that for any t-bounded adversary $\mathscr{A}$ making at most $q_e \leq 2^{n-1}$ queries to an ideal cipher $\widetilde{E}$ as in Lemma 6.3.1. The CMT-4 security of $\mathscr{A}$ against* `Triplex` *is upper bounded by*

$$\mathbf{Adv}^{\text{cmt-4}}_{\texttt{Triplex}}(\mathscr{A}) \leq \frac{q_e^2 + 2q_e + 5}{2^n} + \varepsilon,$$

where $t_1 = O(t)$.

Proof. First, we will redefine the structure of `Triplex`. This is done by moving the two parallel TBC calls from out of F into the Efunction. We denote the modified functions as F′ and E′, respectively, as illustrated in Figure 6.2. This change does not affect the scheme's security as the entire scheme is still identical. In this representation, F′ is not collision-resistant, and we apply Theorem 6.2.3. First, since sk is used as both K_e and K_m, then the Key Generating Function contributes zero to the total upper bound on the advantage. Assume G is (ε_1, t_1)-CR then by Lemma 6.3.1 we have that

$$\varepsilon_1 \leq \frac{q_e^2 + 2q_e + 5}{2^n}.$$

Finally, let E′ be (ε_2, t_2)-RCR, we are left with the task of giving an upper bound on ε_2. Note that the function E′ can be visualized as shown in Figure 6.3, wherein the bottom part is the Triplex hash function of the input $\mathsf{pad}(\text{iv}, N, pk, A, C)$ for some injective padding function. The top symmetric encryption component computes ciphertext C being input to the hash function. Thus, if $N_1 = N_2$, the top part (SE) is Bijective and fv-collision-resistant. If $N_1 \neq N_2$, it is still fv-collision-resistant since N is part of the input to the hash function. Then,

$$\varepsilon_2 \leq \varepsilon\,.$$

Finally, the hash function H employed in Triplex is the Merkle-Damgård with Permutation (MDP) hash function [168] instantiated with Hir, Hirose's double-block-length function [167]. Due to the indifferentiability of this MDPH hash function, which combines the MDP domain extender and Hirose's compression function [250], ε is negligible. This completes the proof. □

Part III

Post Quantum Provable Security of Symmetric-Key Schemes

7

Post-Quantum Secure Compressing PRFs

Block ciphers are a fundamental primitive in symmetric-key cryptography, commonly regarded as secure pseudorandom functions (PRFs) in the classical setting, with their security holding up to about $2^{n/2}$ adversarial queries, where n is the block size. However, in the quantum setting, this security threshold degrades to $2^{n/3}$, as shown by Zhandry. Block ciphers also serve as building blocks for other cryptographic primitives like authenticated encryption schemes and message authentication codes (MACs). Notably, $2n$-bit-to-n-bit PRFs are crucial for constructing higher-level, optimally secure schemes in the classical sense, such as deterministic MACs and authenticated encryption schemes using the SIV construction.

In the quantum context, constructing a secure $2n$-bit-to-n-bit PRF is essential for developing more sophisticated quantum-resistant constructions. While classical composition results do not directly translate to the quantum world, Hosoyamada and Iwata took a significant step by proving the quantum security of the `LRWQ` construction, which uses three PRF calls. Their work raises further questions about the potential security of constructions using fewer or more PRF calls. This chapter investigates these questions, categorizing the security of various $2n$-bit-to-n-bit PRFs based on their structure and providing quantum security proofs for selected constructions using a combination of techniques from Zhandry's compressed oracle framework and recent advancements in the field.

7.1 Characterizing $2n$-Bit to n-Bit Functions

In this section, we study the security of compressing functions in $\mathsf{Func}(2n;n)$, built upon components in $\mathsf{Func}(n;n)$. Specifically, we will construct compressing PRFs by employing an SPN like structure that involves substitution layers with a single PRF in each round. We start with the following definition (see also Figure 7.1).

Definition 7.1.1. *A function $\mathbf{F}:\{0,1\}^{2n}\to\{0,1\}^n$ will be called an r-call function if there exists component functions $f_1,\ldots,f_r\in\mathsf{Func}(n;n)$ and linear functions $L_1,\ldots,L_r$ such that for any $i\in[1;r]$, $L_i:\{0,1\}^{2n}\times Im(f_1)\times\ldots\times Im(f_{i-1})\to\{0,1\}^n$ and $\mathbf{F}=L_r\circ f_r\ldots\circ f_1\circ L_1$.*

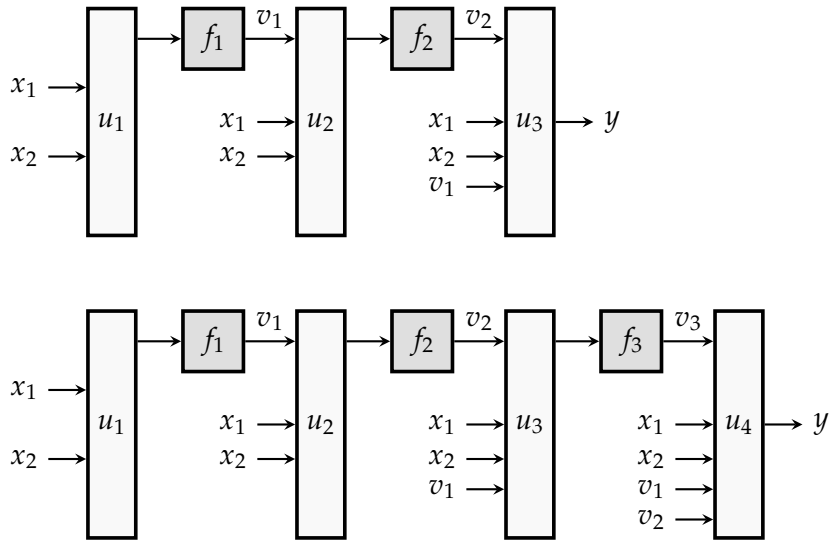

Figure 7.1: *Graphical representation of the generic $2n$-bit-to-n-bit PRF construction with two (top) and three (bottom) n-bit-to-n-bit PRF calls and linear functions. In this figure f_1, f_2, and f_3 are n-bit-to-n-bit PRFs, u_1, u_2, u_3, and u_4 are linear functions, and all wires are n-bit wide.*

In other words, an r-call function alternates between a linear layer taking the original inputs and the outputs of the previous component functions followed by a substitution layer consisting of one component function.

Our primary objective is to determine the minimum number of secret component functions and arbitrary linear functions necessary to construct a secure $2n$-to-n-bit PRF. Additionally, we characterize all secure PRFs that achieve this minimum number of calls. Given that `LRWQ` [177] by Hosoyamada and Iwata can be viewed as a three-call compressing PRF, we restrict our focus to constructions with no more than three calls. Initially, we present attacks on all two-call constructions. Furthermore, we explore attacks on several three-call constructions, identifying promising compressing PRF candidates that can be proved to be post-quantum secure. The proofs are detailed in Section 7.6.

7.1.1 Useful Attack Strategies

Throughout this section, we employ the following attack strategies to construct generic distinguishers against various constructions. Our attacks will be both in the classical and quantum world.

Zero Sum Attack. We start with a classical attack that creates a cycle of four input pairs summing to zero.

Lemma 7.1.1 (Zero-Sum Four-Cycle)**.** *Let $f_1, f_2, f_3 : \{0,1\}^n \to \{0,1\}^n$ be three length preserving functions and let (α_1, α_2), (β_1, β_2), and (γ_1, γ_2) be three arbitrary two dimensional vectors over $\mathbb{F}_{2^n}$. Consider the function $F : \{0,1\}^{2n} \to \{0,1\}^n$ defined by the mapping*

$$(x_1, x_2) \mapsto f_1(\alpha_1 x_1 \oplus \alpha_2 x_2) \oplus f_2(\beta_1 x_1 \oplus \beta_2 x_2) \oplus f_3(\gamma_1 x_1 \oplus \gamma_2 x_2).$$

Then, there exists four distinct pairs $(x_1^1, x_2^1), \ldots, (x_1^4, x_2^4) \in \{0,1\}^{2n}$ *such that,*

$$F(x_1^1, x_2^1) \oplus F(x_1^2, x_2^2) \oplus F(x_1^3, x_2^3) \oplus F(x_1^4, x_2^4) = 0.$$

Proof. The proof involves a case-by-case analysis of the rank of the following matrix:

$$A = \begin{pmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \\ \gamma_1 & \gamma_2 \end{pmatrix}$$

We skip the case where rank is 0, since the lemma is easy to see in that case. We start with the case where the rank of A is 1. Without loss of generality, let (α_1, α_2) be a non-zero vector. One can always find four distinct pairs $(x_1^1, x_2^1), (x_1^2, x_2^2), (x_1^3, x_2^3), (x_1^4, x_2^4) \in \{0,1\}^{2n}$ such that

$$y_1 := \alpha_1 x_1^1 \oplus \alpha_2 x_2^1 = \alpha_1 x_1^2 \oplus \alpha_2 x_2^2, \qquad y_1' := \alpha_1 x_1^3 \oplus \alpha_2 x_2^3 = \alpha_1 x_1^4 \oplus \alpha_2 x_2^4.$$

Since rank of A is 1, for (β_1, β_2) and (γ_1, γ_2) it holds that either they are $(0,0)$ or a non-zero scalar multiple of (α_1, α_2). Hence, one has

$$\begin{aligned} y_2 &:= \beta_1 x_1^1 \oplus \beta_2 x_2^1 = \beta_1 x_1^2 \oplus \beta_2 x_2^2, & y_2' &:= \beta_1 x_1^3 \oplus \beta_2 x_2^3 = \beta_1 x_1^4 \oplus \beta_2 x_2^4, \\ y_3 &:= \gamma_1 x_1^1 \oplus \gamma_2 x_2^1 = \gamma_1 x_1^2 \oplus \gamma_2 x_2^2, & y_3' &:= \gamma_1 x_1^3 \oplus \gamma_2 x_2^3 = \gamma_1 x_1^4 \oplus \gamma_2 x_2^4, \end{aligned}$$

whence we get $\mathrm{F}(x_1^1, x_2^1) = f_1(y_1) \oplus f_2(y_2) \oplus f_3(y_3) = \mathrm{F}(x_1^2, x_2^2)$, and $\mathrm{F}(x_1^3, x_2^3) = f_1(y_1') \oplus f_2(y_2') \oplus f_3(y_3') = \mathrm{F}(x_1^4, x_2^4)$, which completes the proof for this case. Now, assume that rank of A is 2. Without loss of generality, let (α_1, α_2) and (β_1, β_2) be two arbitrary independent vectors. Then, since the rank of A is 2, (γ_1, γ_2) is either $(0,0)$ or a non-zero linear combination of (α_1, α_2) and (β_1, β_2). In other words

$$(\gamma_1, \gamma_2) = (a\alpha_1 \oplus b\beta_1, a\alpha_2 \oplus b\beta_2) \tag{7.1}$$

for some $a, b \in \mathbb{F}_{2^n}$. In any case, we can always fix some $(y_1, y_2) \neq (y_1', y_2') \in \mathbb{F}_{2^n} \times \mathbb{F}_{2^n}$, such that

$$ay_1 \oplus by_2 = ay_1' \oplus by_2'. \tag{7.2}$$

Since, (α_1, α_2) is independent of (β_1, β_2), the mapping φ defined by the mapping $(x_1, x_2) \mapsto \alpha_1 x_1 \oplus \alpha_2 x_2, \beta_1 x_1 \oplus \beta_2 x_2$ is bijective. Let $(x_1^1, x_2^1) = \varphi^{-1}(y_1, y_2)$, $(x_1^2, x_2^2) = \varphi^{-1}(y_1', y_2)$, $(x_1^3, x_2^3) = \varphi^{-1}(y_1', y_2')$, $(x_1^4, x_2^4) = \varphi^{-1}(y_1, y_2')$. From (7.1) and (7.2), we have

$$y_3 := \gamma_1 x_1^1 \oplus \gamma_2 x_2^1 = \gamma_1 x_1^3 \oplus \gamma_2 x_2^3, \qquad y_3' := \gamma_1 x_1^2 \oplus \gamma_2 x_2^2 = \gamma_1 x_1^4 \oplus \gamma_2 x_2^4.$$

Thus, we have $\mathrm{F}(x_1^1, x_2^1) = f_1(y_1) \oplus f_2(y_2) \oplus f_3(y_3)$, $\mathrm{F}(x_1^2, x_2^2) = f_1(y_1') \oplus f_2(y_2) \oplus f_3(y_3')$, $F(x_1^3, x_2^3) = f_1(y_1') \oplus f_2(y_2') \oplus f_3(y_3)$, $\mathrm{F}(x_1^4, x_2^4) = f_1(y_1) \oplus f_2(y_2') \oplus f_3(y_3')$, which completes the proof. □

In our analysis of two call constructions, we often employ the following corollary of Lemma 7.1.1.

Corollary 7.1.1. *Let $f_1, f_2 : \{0,1\}^n \to \{0,1\}^n$ be two length preserving functions and let (α_1, α_2) and (β_1, β_2) be two arbitrary two dimensional vectors over $\mathbb{F}_{2^n}$. Consider the function $F : \{0,1\}^{2n} \to \{0,1\}^n$ defined by the mapping $(x_1, x_2) \mapsto f_1(\alpha_1 x_1 \oplus \alpha_2 x_2) \oplus f_2(\beta_1 x_1 \oplus \beta_2 x_2)$. Then, there exists four distinct pairs $(x_1^1, x_2^1), \ldots, (x_1^4, x_2^4) \in \{0,1\}^{2n}$ such that, $F(x_1^1, x_2^1) \oplus F(x_1^2, x_2^2) \oplus F(x_1^3, x_2^3) \oplus F(x_1^4, x_2^4) = 0$.*

A proof of this result follows from the proof of Lemma 7.1.1 by setting f_3 to be a constant function evaluating to zero.

Remark 7.1.1. *Both Lemma 7.1.1 and Corollary 7.1.1 hold independent of the nature of the underlying functions f_1, f_2, and f_3. Furthermore, the proofs are constructive in nature, which can be utilized by an adversary whose goal is to distinguish F from a uniform random function $\Gamma : \{0,1\}^{2n} \to \{0,1\}^n$. Specifically, finding four distinct inputs $x^1, \ldots, x^4$ such that $\Gamma(x^1) \oplus \Gamma(x^2) \oplus \Gamma(x^3) \oplus \Gamma(x^4) = 0$ is a low probability event. On the other hand, the above results show that such quadruples can be easily derived for a class of functions F, thereby, making them easily distinguishable from a uniform random function.*

Period Finding Attack. Next, we introduce a quantum period finding attack that utilizes the famous Simon's algorithm first introduced in [296]. We start with the definition of a periodic function.

Definition 7.1.2 (Periodic Function)**.** *A function $f : \{0,1\}^n \to \{0,1\}^n$ is said to be periodic if there exists a constant $s \in \{0,1\}^n$ such that for every $x \in \{0,1\}^n$, $f(x \oplus s) = f(x)$. In this case, we say s is a period of f.*

Lemma 7.1.2 (Period Finding)**.** *For any $f_1, f_2, f_3 : \{0,1\}^n \to \{0,1\}^n$, suppose $F : \{0,1\}^{2n} \to \{0,1\}^n$ is defined by the mapping $(x_1, x_2) \mapsto f_3(x_2 \oplus f_1(x_1)) \oplus f_2(x_1)$. Then, for any $x_1^0 \neq x_1^1 \in \{0,1\}^n$, the function $G_{x_1^0, x_1^1} : \{0,1\}^n \to \{0,1\}^n$ defined by the mapping $x_2 \mapsto F(x_1^0, x_2) \oplus F(x_1^1, x_2)$ is periodic with the period $s(x_1^0, x_1^1) = f_1(x_1^0) \oplus f_1(x_1^1)$.*

Proof. The proof of this lemma is straightforward. For any $x_2 \in \{0,1\}^n$, we have

$$\begin{aligned}
\mathsf{G}_{x_1^0, x_1^1}(x_2 \oplus s(x_1^0, x_1^1)) &= \mathsf{F}(x_1^0, x_2 \oplus s(x_1^0, x_1^1)) \oplus \mathsf{F}(x_1^1, x_2 \oplus s(x_1^0, x_1^1)) \\
&= f_3(x_2 \oplus f_1(x_1^0) \oplus f_1(x_1^1) \oplus f_1(x_1^0)) \\
&\quad \oplus f_3(x_2 \oplus f_1(x_1^0) \oplus f_1(x_1^1) \oplus f_1(x_1^1)) \\
&= \mathsf{F}(x_1^0, x_2) \oplus \mathsf{F}(x_1^1, x_2) = \mathsf{G}_{x_1^0, x_1^1}(x_2).
\end{aligned}$$

□

While the first two Lemmas are interesting even in the classical setting, Lemma 7.1.2 is mainly useful in the quantum setting. Specifically, it facilitates the application of Simon's algorithm (see [257] for more details). We often employ Lemma 7.1.2 in conjunction with the following useful result [195] due to Kaplan et al. which greatly extends the scope of Simon's algorithm. Let $f : \{0,1\}^n \to \{0,1\}^n$ be a function with some period $s \neq 0$. In [195], Kaplan et al. define

$$\epsilon(f,s) := \max_{t\in\{0,1\}^n\setminus\{0,s\}} \Pr_x\left(f(x) = f(x\oplus t)\right) \tag{7.3}$$

Theorem 7.1.1 ([195], Theorem 1). *If $\epsilon(f,s) \le p_0 < 1$, then Simon's algorithm returns s with cn queries, with probability at least $1 - \left(2\left[(1+p_0)/2\right]^c\right)^n$.*

Note that by choosing $c > 3/(1-p_0)$ we ensure that the error decreases exponentially with n. Thus, it is sufficient to show that $\epsilon(f,s) < 1$. Specifically, it is well-known that $\epsilon(f,s) = \Theta(n2^{-n})$ when f is a random function. Then, Simon's algorithm returns the period with probability close to 1. Thus, we can utilize this result to mount a quantum attack using Lemma 7.1.2. We give an informal explanation below. A formal treatment of quantum algorithms and distinguishers is given in Section 7.2.

Remark 7.1.2. *Since a uniform random function is not periodic with very high probability, Lemma 7.1.2 can be utilized by an adversary whose goal is to distinguish a periodic random function $\mathbf{F}$ from a uniform random function $\Gamma : \{0,1\}^{2n} \to \{0,1\}^n$. Specifically, the adversary can first apply Simon's period finding algorithm in conjunction with Lemma 7.1.2 to get a candidate period s in $O(n)$ queries. Followed by this, it can simply make two queries x and $x\oplus s$, and look for a collision at the outputs for these two queries. In a uniform random function this happens with roughly 2^{-n} probability, while for a periodic $\mathbf{F}$, this will happen with probability 1.*

Throughout this chapter, when declaring a candidate construction as insecure, we frequently reference Lemma 7.1.1 and Lemma 7.1.2 as well as Corollary 7.1.1 as the basis for the attacks. We omit a formal description of these attacks and their advantage computation, as they involve at most a polynomial number of queries and achieve nearly full advantage. Nevertheless, it is important to note that such attacks can be easily formalized using the concise strategies outlined in Remark 7.1.1 and Remark 7.1.2.

Next, we analyze the constructions based on two and three calls. Throughout this section, let $f_1, f_2, f_3 : \{0,1\}^n \to \{0,1\}^n$ be three independent secret random functions. Let $\alpha = (\alpha_1,\alpha_2) \in \{0,1\}^{2n}$, $\beta = (\beta_1,\beta_2,\beta_3) \in \{0,1\}^{3n}$, $\gamma = (\gamma_1,\gamma_2,\gamma_3,\gamma_4) \in \{0,1\}^{4n}$, $\delta = (\delta_1,\delta_2,\delta_3,\delta_4,\delta_5) \in \{0,1\}^{5n}$ be some public parameters.

7.1.2 Constructions Based on Two Calls

Any compressing function based on two calls $\mathtt{F}$ can be represented by a matrix A and two component functions f_1, f_2, where

$$A = \begin{pmatrix} \alpha_1 & \alpha_2 & 0 & 0 \\ \beta_1 & \beta_2 & \beta_3 & 0 \\ \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \end{pmatrix}$$

and for any two inputs $(x_1,x_2) \in \{0,1\}^{2n}$, one has

1. $u_1(x_1,x_2) = \alpha_1 x_1 \oplus \alpha_2 x_2$;
2. $v_1(x_1,x_2) = f_1(u_1(x_1,x_2))$;

3. $u_2(x_1, x_2, v_1) = \beta_1 x_1 \oplus \beta_2 x_2 \oplus \beta_3 v_1$;
4. $v_2(x_1, x_2) = f_2(u_2(x_1, x_2, v_1))$;
5. $u_3(x_1, x_2, v_1, v_2) = \gamma_1 x_1 \oplus \gamma_2 x_2 \oplus \gamma_3 v_1 \oplus \gamma_4 v_2$;
6. $\mathtt{F}(x_1, x_2) = y = u_3(x_1, x_2, v_1, v_2)$.

With a slight abuse of notation, we simply write u_i and v_j to denote $u_i(\cdot)$ and $v_j(\cdot)$ for all $i \in [1;3]$ and $j \in [1;2]$, whenever the input is known from the context, or the stated fact is independent of the inputs. With this slight simplification, we can represent the entire function using the following system of equations:

$$A \cdot \begin{pmatrix} x_1 \\ x_2 \\ v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}.$$

Simplifications. First, note that some simple simplifications can be done with respect to the matrix A:

- Without loss of generality, we assume that $\gamma_1 = \gamma_2 = 0$, since the adversary can easily create $u_3' = u_3 \oplus \gamma_1 x_1 \oplus \gamma_2 x_2$ for any pair of inputs $(x_1, x_2) \in \{0,1\}^{2n}$.
- We assume that each row of A is non-zero. Otherwise, there exists $i \in [1;3]$ such that $u_i = 0$, whence either $\mathtt{F}$ is independent of f_1 or f_2, or it is a constant.
- We assume that each column of A is non-zero as well. Otherwise, for all $i \in [1;3]$, u_i is independent of one of x_1, x_2, v_1, and v_2, whence $\mathtt{F}$ is independent of f_1 or f_2 or it is independent of one of its inputs.
- We can multiply any row by a non-zero constant. Indeed, for the first two rows, multiplying the input of a uniformly random function by a non-zero constant does not change the distribution of the outputs. For the final row, the adversary can multiply the outputs of the construction by any constant.

Using the above simplifications, from now on we can assume that $\gamma_4 = 1$ by normalizing the final row by γ_4^{-1}. This gives a matrix of the form:

$$A = \begin{pmatrix} \alpha_1 & \alpha_2 & 0 & 0 \\ \beta_1 & \beta_2 & \beta_3 & 0 \\ 0 & 0 & \gamma_3 & 1 \end{pmatrix}$$

where each row and column of A is assumed to be non-zero.

Characterization of Two-call Constructions. The full characterization of $\mathtt{F}$ is given by the following three cases.

1. First assume $\beta_1 = \beta_2 = 0$. Then, according to our simplification $\beta_3 = 1$. Therefore,

   $$\mathtt{F}(x_1, x_2) = (\gamma_3 f_1(u_1)) \oplus (f_2(f_1(u_1))).$$

   Using Corollary 7.1.1, we can find $(x_1, x_2) \neq (x_1', x_2')$ such that $\mathtt{F}(u_1(x_1, x_2)) \oplus \mathtt{F}(u_1(x_1', x_2')) = 0$. That gives a classical collision attack.

2. Next, assume $\beta_1 \neq 0$ or $\beta_2 \neq 0$, and $\alpha_1\beta_2 = \alpha_2\beta_1$. Then, there exists a non-zero $c \in \mathbb{F}_{2^n}$, such that $(\beta_1, \beta_2) = (c\alpha_1, c\alpha_2)$. So for every pair of inputs $(x_1, x_2) \neq (x_1', x_2')$, such that $\alpha_1 x_1 \oplus \alpha_2 x_2 = \alpha_1 x_1' \oplus \alpha_2 x_2'$, we must have $\beta_1 x_1 \oplus \beta_2 x_2 = \beta_1 x_1' \oplus \beta_2 x_2'$. Therefore, $u_1(x_1, x_2) = u_1(x_1', x_2')$ and $u_2(x_1, x_2, v_1) = u_2(x_1', x_2', v_1)$ which implies that $u_3(x_1, x_2, v_1, v_2) = u_3(x_1', x_2', v_1, v_2)$. This clearly gives a collision attack on the construction for inputs (x_1, x_2) and (x_1', x_2').
3. Otherwise, $\beta_1 \neq 0$ or $\beta_2 \neq 0$, and $\alpha_1\beta_2 \neq \alpha_2\beta_1$. Then the construction is reduced to,

$$\mathtt{F}(x_1, x_2) = \gamma_3 f_1(\alpha_1 x_1 \oplus \alpha_2 x_2) \oplus f_2\left(\beta_1 x_1 \oplus \beta_2 x_2 \oplus \beta_3 f_1(\alpha_1 x_1 \oplus \alpha_2 x_2)\right).$$

Let $f_1' = \gamma_3 f_1$, and $f_1'' = \beta_3 f_1$, and $u_2'(x_1, x_2) = \beta_1 x_1 \oplus \beta_2 x_2$. Then, the above construction reduces to

$$\mathtt{F}(x_1, x_2) = f_1'(u_1(x_1, x_2)) \oplus f_2\left(u_2'(x_1, x_2) \oplus f_1''(u_1(x_1, x_2))\right).$$

Using Lemma 7.1.2, we can come up with a periodic function, and hence using Theorem 7.1.1, we can find the period in polynomial number of queries.

This concludes the complete characterization of constructions based on two calls. Thus, the analysis above establishes that two calls are not sufficient to construct a $2n$-bit-to-n-bit quantum secure PRF.

7.1.3 Constructions Based on Three Calls

Any compressing function based on three calls $\mathtt{F}$ can be represented by a matrix A and three component functions f_1, f_2, f_3, where

$$A = \begin{pmatrix} \alpha_1 & \alpha_2 & 0 & 0 & 0 \\ \beta_1 & \beta_2 & \beta_3 & 0 & 0 \\ \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & 0 \\ \delta_1 & \delta_2 & \delta_3 & \delta_4 & \delta_5, \end{pmatrix}$$

where for any input $(x_1, x_2) \in \{0,1\}^{2n}$, one has

- $u_1(x_1, x_2) = \alpha_1 x_1 \oplus \alpha_2 x_2$;
- $v_1(x_1, x_2) = f_1(u_1(x_1, x_2))$;
- $u_2(x_1, x_2, v_1) = \beta_1 x_1 \oplus \beta_2 x_2 \oplus \beta_3 v_1$;
- $v_2(x_1, x_2) = f_2(u_2(x_1, x_2, v_1))$;
- $u_3(x_1, x_2, v_1, v_2) = \gamma_1 x_1 \oplus \gamma_2 x_2 \oplus \gamma_3 v_1 \oplus \gamma_4 v_2$;
- $v_3(x_1, x_2) = f_3(u_3(x_1, x_2, v_1, v_2))$;
- $u_4(x_1, x_2, v_1, v_2, v_3) = \delta_1 x_1 \oplus \delta_2 x_2 \oplus \delta_3 v_1 \oplus \delta_4 v_2 \oplus \delta_5 v_3$;
- $\mathtt{F}(x_1, x_2) = y = u_4(x_1, x_2, v_1, v_2, v_3)$.

Thus, we can represent a three call construction using the following system of equations:

$$A \cdot \begin{pmatrix} x_1 \\ x_2 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix}. \tag{7.4}$$

Simplifications and Preconditions. We make some simplification as in the previous analysis. Namely,

- $\delta_1 = \delta_2 = 0$;
- each row of the matrix is non-zero; and
- each column of the matrix is non-zero; and
- without loss of generality $\delta_5 = 1$.

Further, we identity the following preconditions as necessary to to get a secure construction:

Precondition 1: (α_1, α_2) is independent of (β_1, β_2);

Precondition 2: Either $\gamma_4 \neq 0$, or

(a) (α_1, α_2) is independent of (γ_1, γ_2), and
(b) $(\beta_1, \beta_2, \beta_3)$ should be independent of $(\gamma_1, \gamma_2, \gamma_3)$;

Precondition 3: $\begin{pmatrix} \beta_3 & \gamma_3 \\ \gamma_4 & 0 \end{pmatrix} \neq \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$.

Specifically, the following lemma demonstrates that a violation of any of the preconditions renders the construction vulnerable to a (quantum) attack.

Lemma 7.1.3. *Preconditions 1, 2, and 3 above are necessary for* $\mathtt{F}$ *to be a quantum secure PRF.*

Proof. We start with Precondition 1. Distinguish between two cases.

- If $\alpha_1\gamma_2 = \alpha_2\gamma_1$, then we can construct a collision attack on $\mathtt{F}$ using a similar argument as used in the second case in the full classification of two call constructions described in Section 7.1.2.
- Otherwise, the function $(x_1, x_2) \mapsto (\alpha_1 x_1 \oplus \alpha_2 x_2, \gamma_1 x_1 \oplus \gamma_2 x_2)$ is a bijection. Moreover, there exists $c \neq 0$ such that, $(\alpha_1, \alpha_2) = (c\beta_1, c\beta_2)$. Let $u_3'(x_1, x_2) = \gamma_1 x_1 \oplus \gamma_2 x_2$. Then we can rewrite $\mathtt{F}$ as

$$\begin{aligned} \mathtt{F}(x_1, x_2) = \delta_3 f_1(u_1) \oplus \delta_4 f_2(cu_1 \oplus \beta_3 f_1(u_1)) \\ \oplus f_3\left(u_3' \oplus \gamma_3 f_1(u_1) \oplus \gamma_4 f_2(cu_1 \oplus \beta_3 f_1(u_1))\right). \end{aligned}$$

We define $\mathtt{F}_1, \mathtt{F}_2 : \{0,1\}^n \to \{0,1\}^n$ by

$$\mathtt{F}_1(u_1) = \delta_3 f_1(u_1) \oplus \delta_4 f_2(cu_1 \oplus \beta_3 f_1(u_1)),$$

$$\mathtt{F}_2(u_1) = \gamma_3 f_1(u_1) \oplus \gamma_4 f_2(cu_1 \oplus \beta_3 f_1(u_1)).$$

This reduces $\mathtt{F}(x_1, x_2)$ to $F_1(x_1) \oplus f_3(x_2 \oplus F_2(x_1))$, which, as we show in Lemma 7.1.2, is susceptible to period finding, and hence distinguishable in polynomial number of queries using Theorem 7.1.1.

Next, we analyze Precondition 2 assuming that Precondition 1 holds. Otherwise, a similar attack can be mounted (irrespective of whether $\gamma_4 = 0$ or not). We start with the case where (α_1, α_2) and (γ_1, γ_2) are dependent. Then there exists $c \neq 0$ such that $(c\alpha_1, c\alpha_2) = (\gamma_1, \gamma_2)$. Write $u_2' = \beta_1 x_1 \oplus \beta_2 x_2$, then we can rewrite $\mathtt{F}(x_1, x_2)$ as

$$\delta_3 f_1(u_1) \oplus \delta_4 f_2(u_2' \oplus \beta_3 f_1(u_1)) \oplus f_3\left(cu_1 \oplus \gamma_3 f_1(u_1)\right).$$

We define $\mathtt{F}_1, \mathtt{F}_2 : \{0,1\}^n \to \{0,1\}^n$ as

$$\mathtt{F}_1(u_1) = \delta_3 f_1(u_1) \oplus f_3\left(cu_1 \oplus \gamma_3 f_1(u_1)\right), \qquad \mathtt{F}_2(u_1) = \beta_3 f_1(u_1).$$

Then $\mathtt{F}(x_1, x_2)$ is reduced to $\mathtt{F}_1(u_1) \oplus \delta_3 f_2(u_2' \oplus \mathtt{F}_2(u_1))$, which similarly as before is susceptible to period finding (using Lemma 7.1.2 and Theorem 7.1.1). For the case when $(\beta_1, \beta_2, \beta_3)$ and $(\gamma_1, \gamma_2, \gamma_3)$ are dependent, we can argue similarly that the resulting construction is susceptible to period finding.

Finally, consider Precondition 3, in this case the adversary can deduce and to some extent manipulate u_1, u_2, u_3 (since he knows the parameters). Namely, we can write

$$\mathtt{F}(x_1, x_2) = \delta_3 f_1(\alpha_1 x_1 \oplus \alpha_2 x_2) \oplus \delta_4 f_2(\beta_1 x_1 \oplus \beta_2 x_2) \oplus \delta_5 f_3(\gamma_1 x_1 \oplus \gamma_2 x_2).$$

Using Lemma 7.1.1, we can find four queries whose outputs sum to 0. This gives a simple classical distinguisher. □

Utilizing the aforementioned simplifications and preconditions, we can reduce (7.4) to the following system of equations:

$$\begin{pmatrix} \alpha_1 & \alpha_2 & 0 & 0 & 0 \\ \beta_1 & \beta_2 & \beta_3 & 0 & 0 \\ \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & 0 \\ 0 & 0 & \delta_3 & \delta_4 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix}. \tag{7.5}$$

Characterization of Three-call Constructions. In the following discussion, we analyze the majority of the three-call constructions. First, we demonstrate that some of these constructions are susceptible to the previously described attacks. Second, we identify promising candidates (mainly characterized by efficiency and resistance to the previous attacks) for post-quantum secure compressing PRFs. Our analysis is divided into two cases:

1. Without loss of generality assume $\delta_4 = 1$, and consider the three sub cases below:

(a) $\beta_3 = 0$. By Precondition 3, we must have $\gamma_3 \neq 0$. For simplicity assume $\gamma_3 = 1$. Moreover, notice that Precondition 1 implies that without loss of generality,

$$\begin{pmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Next, note that $\gamma_2 \neq 0$, otherwise this violate Precondition 2. Therefore, the parameter matrix is reduced to

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \gamma_1 & \gamma_2 & 1 & 0 & 0 \\ 0 & 0 & \delta_3 & 1 & 1 \end{pmatrix}, \tag{7.6}$$

where the blue elements indicate strictly non-zero values. (We stick to this colour code in the rest of this section.) We further simplify the above matrix by setting $\gamma_1 = \delta_3 = 0$, and $\gamma_2 = 1$, this simplification stems from the point of view of efficiency: a simple XOR is always preferable to a finite field multiplication followed by an XOR. Finally, we get the following simplified matrix:

$$A_{\mathrm{LRQ}} := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \tag{7.7}$$

and the resulting construction is defined as

$$\mathtt{LRQ}(x_1, x_2) := f_2(x_2) \oplus f_3(x_2 \oplus f_1(x_1)). \tag{7.8}$$

(b) $\gamma_3 = 0$. By Precondition 3, we must have $\beta_3 \neq 0$. For simplicity, assume $\beta_3 = 1$. Moreover, note that Precondition 2 implies that without loss of generality,

$$\begin{pmatrix} \alpha_1 & \alpha_2 \\ \gamma_1 & \gamma_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Further, note that we must have $\beta_2 \neq 0$, otherwise this violates Precondition 1. Therefore, the parameters matrix is reduced to

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \beta_1 & \beta_2 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \delta_3 & 1 & 1 \end{pmatrix}. \tag{7.9}$$

Simplifying the matrix by setting $\beta_1 = \delta_3 = 0$ and $\beta_2 = 1$, we observe that this corresponds to the same construction as (7.7) up to a relabeling of functions.

(c) $\beta_3, \gamma_3 \neq 0$. Without loss of generality assume that $\beta_3 = 1$. The resulting matrix is given by

$$\begin{pmatrix} \alpha_1 & \alpha_2 & 0 & 0 & 0 \\ \beta_1 & \beta_2 & 1 & 0 & 0 \\ \gamma_1 & \gamma_2 & \gamma_3 & 0 & 0 \\ 0 & 0 & \delta_3 & 1 & 1 \end{pmatrix}, \tag{7.10}$$

where the red submatrix represents the fact that it satisfies Precondition 1 and 2, i.e., we must have (α_1, α_2) independent of (β_1, β_2) and (γ_1, γ_2), and $(\beta_1, \beta_2, 1)$ independent of $(\gamma_1, \gamma_2, \gamma_3)$. Using similar simplifying arguments as before, and preserving isomorphism up to a relabeling of functions, we get the following interesting matrices:

$$A_{\texttt{CSUMQ}} := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \qquad A_{\texttt{LMQ}} := \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \tag{7.11}$$

The resulting constructions are defined as

$$\texttt{CSUMQ}(x_1, x_2) := f_2(x_2 \oplus f_1(x_1)) \oplus f_3(x_2 \oplus x_1 \oplus f_1(x_1)), \tag{7.12}$$

$$\texttt{LMQ}(x_1, x_2) := f_2(x_2 \oplus f_1(x_1 \oplus x_2)) \oplus f_3(x_1 \oplus f_1(x_1 \oplus x_2)). \tag{7.13}$$

2. $\gamma_4 \neq 0$. Without loss of generality, assume that $\gamma_4 = 1$. Consider the following three sub-cases:

(a) $\beta_3 = \gamma_3 = 0$. Then, using Precondition 1, the resulting parameters matrix is given by

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \gamma_1 & \gamma_2 & 0 & 1 & 0 \\ 0 & 0 & \delta_3 & \delta_4 & 1 \end{pmatrix}. \tag{7.14}$$

The condition $\gamma_1 \neq 0$ can be easily argued as follows: Suppose, $\gamma_1 = 0$. Then, using Corollary 7.1.1, one can find four queries such that the outputs sum to 0, resulting in a classical distinguishing attack. Similarly, $\delta_3 \neq 0$, since each column must have one non-zero entry. Further, by setting $\gamma_2 = \delta_4 = 0$ and $\gamma_1 = \delta_3 = 1$, we arrive at the same construction as in (7.7) up to a relabeling of functions and input variables.

(b) $\beta_3 = 0$ and $\gamma_3 \neq 0$. Then, using Precondition 1, the parameters matrix is given by

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \gamma_1 & \gamma_2 & \gamma_3 & 1 & 0 \\ 0 & 0 & \delta_3 & \delta_4 & 1 \end{pmatrix}, \tag{7.15}$$

By setting $\gamma_1 = \gamma_2 = \delta_3 = \delta_4 = 0$ and $\gamma_3 = 1$, we get at the following matrix:

$$A_{\texttt{LRWQ}} := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{7.16}$$

which corresponds to the LRWQ construction [177] by Hosoyamada and Iwata, defined as

$$\texttt{LRWQ}(x_1, x_2) := f_3\left(f_1(x_1) \oplus f_2(x_2)\right). \tag{7.17}$$

(c) $\gamma_3 = 0$ and $\beta_3 \neq 0$. Without loss of generality, we assume that $\beta_3 = 1$. Then, using Precondition 1, we are left with the general matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ \gamma_1 & \gamma_2 & 0 & 1 & 0 \\ 0 & 0 & \delta_3 & \delta_4 & 1 \end{pmatrix}, \tag{7.18}$$

where red elements indicate that they cannot all be 0. This can be easily argued by looking at the resulting construction. Suppose, $\gamma_1 = \gamma_2 = 0$. Then, the second and third calls can be clubbed together (since the output of the second call is directly fed into the third call), resulting in a reduction to an equivalent two-call construction, which is already shown to be insecure. Now, using the simplification steps, we have the following two matrices:

$$A_{\texttt{EDMQ}} := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \qquad A_{\texttt{TNT}} := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{7.19}$$

where the second matrix, i.e., $A_{\texttt{TNT}}$ corresponds to the TNT construction [17] by Bao et al. The corresponding constructions are defined as follows:

$$\texttt{EDMQ}(x_1, x_2) := f_3(x_1 \oplus f_2(x_2 \oplus f_1(x_1))), \tag{7.20}$$

$$\texttt{TNT}(x_1, x_2) := f_3(x_2 \oplus f_2(x_2 \oplus f_1(x_1))). \tag{7.21}$$

(d) $\beta_3, \gamma_3 \neq 0$. In this case, using Precondition 1, the parameters matrix is given by

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & \beta_3 & 0 & 0 \\ \gamma_1 & \gamma_2 & \gamma_3 & 1 & 0 \\ 0 & 0 & \delta_3 & \delta_4 & 1 \end{pmatrix}. \tag{7.22}$$

Further, by setting $\gamma_1 = \gamma_2 = \delta_3 = \delta_4 = 0$, and $\beta_3 = \gamma_3 = 1$, we get

$$A_{\texttt{EDMDQ}} := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{7.23}$$

and the corresponding construction is defined as

$$\texttt{EDMDQ}(x_1, x_2) := f_3(f_1(x_1) \oplus f_2(x_2 \oplus f_1(x_1))). \tag{7.24}$$

Table 7.1: *Summary of the possibly secure PRF candidates with minimum number of random function calls.*

Candidate	Definition	Memory	XORs	Invertible	Parallel
LRQ	$f_2(x_2) \oplus f_3(x_2 \oplus f_1(x_1))$	$2n$	2	✓	✓
CSUMQ	$f_2(x_2 \oplus f_1(x_1)) \oplus f_3(x_2 \oplus x_1 \oplus f_1(x_1))$	$2n$	3	×	✓
LMQ	$f_2(x_2 \oplus f_1(x_1 \oplus x_2)) \oplus f_3(x_1 \oplus f_1(x_1 \oplus x_2))$	$2n$	4	×	✓
LRWQ [177]	$f_3(f_1(x_1) \oplus f_2(x_2))$	$2n$	1	✓	✓
EDMQ	$f_3(x_1 \oplus f_2(x_2 \oplus f_1(x_1)))$	n	2	×	×
TNT [17]	$f_3(x_2 \oplus f_2(x_2 \oplus f_1(x_1)))$	n	2	✓	×
EDMDQ	$f_3(f_1(x_1) \oplus f_2(x_2 \oplus f_1(x_1)))$	n	2	×	×

Summary of Interesting Quantum PRF Candidates. In the discussion above we gave a full classification of three-call constructions, where we focused on constructions that are both efficient and potentially post quantum secure. In Table 7.1, we summarize the features of the seven candidate qPRF constructions found above. In particular, we note that three of these constructions, LRQ , LRWQ [177], and TNT [17], are special as they can act as a tweakable permutation when the underlying primitives are permutations. Furthermore, they are also among the most favorable candidates in terms of desirable implementation features like XOR counts, parallelization, and state size. Therefore, for the rest of this chapter, we concentrate on proving the security of these three candidates. We will show that these constructions are qPRF secure. Furthermore, as we will see in Section 7.7, similar results can be derived for the qTPRP security using a well-known switching result [174, 175] due to Hosoyamada and Iwata.[1]

To prove the qPRF security of these constructions, we will extend Zhandry's compressed oracle technique [325] into a more general proof framework (see Section 7.4 for more details). However, as discussed in Section 8.3, this framework faces challenges in the adaptive setting. While we can still prove the security of TNT and LRWQ with this framework—at a cost of looser bound—the situation is more fundamental for LRQ , where the limitations appear to be inherent, making it unprovable within the current framework.

[1] We remark that the TPRP security would only hold against unidirectional quantum distinguishers.

7.2 Quantum Computation and Security

Throughout this chapter, we assume the reader is familiar with the fundamental concepts of Quantum Computation and Linear Algebra as outlined in [257, 138]. For Quantum Computation, we follow the definitions in [257], adapting them to suit our specific needs. Additionally, Appendix A provides basic Linear Algebra results that will be useful in this chapter. We use the standard Dirac notation, as is customary in Quantum computation.

In classical computational complexity theory, information and data are described in terms of bits, represented as elements of $\{0,1\}^n$ for some $n \in \mathbb{N}$. Conversely, in quantum computation theory, information and data used by quantum algorithms are described by quantum bits, or *qubits*. To introduce the concept of qubits and *Quantum systems*, it is essential to be familiar with the notions of Hilbert spaces and Density operators. Therefore, we first need to cover some basic definitions and concepts from linear algebra.

7.2.1 Hilbert Space, Operator and Norm

Inner Product Space and Outer Product. For a vector space $\mathbb{V}$ over a field $\mathbb{F}$ that is either the real numbers $\mathbb{R}$ or the complex numbers $\mathbb{C}$, an inner product is a map $\langle|\rangle : \mathbb{V} \times \mathbb{V} \to \mathbb{F}$ that satisfies the following properties for all vectors $|x\rangle, |y\rangle, |z\rangle \in \mathbb{V}$ and scalars $a, b \in \mathbb{F}$:

- *Linearity:* $\langle ax + by|z\rangle = a\langle x|z\rangle + b\langle y|z\rangle$,
- *Conjugate symmetry:* $\langle x|y\rangle = \langle y|x\rangle^*$,
- *Positive-definiteness:* $\langle x|x\rangle > 0$ for all non-zero $|x\rangle$.

A vector space $\mathbb{V}$ equipped with an inner product is called an inner product space. It is well known that every inner product induces a norm $\|\cdot\|$, called the canonical norm, which is defined by $\|x\| = \sqrt{\langle x|x\rangle}$ for every $x \in \mathbb{V}$.

For two inner product spaces $\mathbb{V}$ and $\mathbb{W}$ with the same inner product and any two vectors $|x\rangle \in \mathbb{V}$ and $|y\rangle \in \mathbb{W}$, the *outer product* $|x\rangle\langle y|$ is a linear operator from $\mathbb{V}$ to $\mathbb{W}$ defined as

$$|x\rangle\langle y|(|x'\rangle) = \langle x|x'\rangle|y\rangle,$$

for any $x' \in \mathbb{V}$.

Hilbert Space and Orthonormal Basis. In this thesis, we focus exclusively on Hilbert spaces with finite dimensions, which are equivalent to inner product spaces over the field of complex numbers. Therefore, we use the following simplified definition of a Hilbert space. For any positive integer $k \in \mathbb{N}$, a k-dimensional *complex Hilbert space* $\mathcal{H}$ is simply the vector space $\mathbb{C}^k$ over the complex field $\mathbb{C}$ with the natural choice of inner product $\langle\cdot|\cdot\rangle$ defined as follows: for any $|\phi\rangle, |\psi\rangle \in \mathcal{H}$,

$$\langle\phi|\psi\rangle = \sum_{i,j\in[1;k]} \alpha_i^* \beta_j,$$

where $|\phi\rangle$ and $|\psi\rangle$ are uniquely represented by some arbitrary basis such that:

$$|\phi\rangle = \sum_{i=1}^{k} \alpha_i |\gamma_i\rangle,$$
$$|\psi\rangle = \sum_{j=1}^{k} \beta_j |\gamma_j\rangle.$$

Throughout this work, we consider only the canonical norm, which we refer to simply as the norm of $\mathcal{H}$.

A basis $\mathtt{B} = \{|\gamma_1\rangle, \ldots, |\gamma_k\rangle\}$ of an inner product space $\mathbb{V}$ is called orthonormal if for every $i, j \in [1; k]$, $\langle\gamma_i|\gamma_j\rangle = 1$ if $i = j$ and is equal zero otherwise. Given an orthonormal basis $\mathtt{B}$ of an Hilbert space $\mathcal{H}$, we sometimes write $\mathbb{C}[\mathtt{B}]$ to emphasize the basis representation of $\mathcal{H}$.

Tensor Product. For any two finite-dimensional complex Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ of dimensions k_1 and k_2, respectively, with the same inner product, the tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$ is another complex Hilbert space of dimension $k_1 k_2$, where the inner product is defined as:

$$\langle\phi_1 \otimes \phi_2|\psi_1 \otimes \psi_2\rangle = \langle\phi_1|\psi_1\rangle\langle\phi_2|\psi_2\rangle.$$

It is also well-known that $\mathcal{H}_1 \otimes \mathcal{H}_2$ is isomorphic to the canonical $k_1 k_2$-dimensional complex Hilbert space. We often use simplified notations for elements in $\mathcal{H}_1 \otimes \mathcal{H}_2$. Specifically, we write:

$$|\phi, \psi\rangle := |\phi\rangle|\psi\rangle := |\phi\rangle \otimes |\psi\rangle.$$

Operator Norm. For a linear operator $A : \mathbb{C}[\mathcal{X}_0] \longrightarrow \mathbb{C}[\mathcal{X}_1]$, we define the *operator norm* of A as

$$\|A\| = \sup_{|\psi\rangle \in \mathbb{C}[\mathcal{X}_0], \||\psi\rangle\| = 1} \|A|\psi\rangle\|,$$

here the norm on the right hand side is the norm over the Hilbert space $\mathbb{C}[\mathcal{X}_1]$. If

$$A = \sum_{i=1}^{r} \sigma_i |x_i\rangle\langle y_i|$$

is the singular value decomposition of A (where r is the rank of A and $x_1, \ldots, x_r \in \mathcal{X}_1, y_1, \ldots, y_r \in \mathcal{X}_0$), then we have

$$\|A\| = \max_i \sigma_i.$$

For four finite sets $\mathcal{X}_0$, $\mathcal{X}_1$, $\mathcal{X}_0'$, and $\mathcal{X}_1'$, let $A : \mathbb{C}[\mathcal{X}_0] \longrightarrow \mathbb{C}[\mathcal{X}_1]$ and $A' : \mathbb{C}[\mathcal{X}_0'] \longrightarrow \mathbb{C}[\mathcal{X}_1']$ be linear operators with singular value decomposition

$$A = \sum_{i=1}^{r} \sigma_i |x_i\rangle\langle y_i|, \qquad A' = \sum_{i'=1}^{r'} \sigma'_{i'} |x'_{i'}\rangle\langle y'_{i'}|.$$

Unitary Operators and Density Operators. A linear operator $\mathbf{U}$ on $\mathcal{H}$ is said to be unitary if $\mathbf{U}^\dagger\mathbf{U} = \mathbf{I}_\mathcal{H}$, where $\mathbf{U}^\dagger$ is the adjoint[2] of $\mathbf{U}$ and $\mathbf{I}_\mathcal{H}$ denotes the identity operator on $\mathcal{H}$. Let $\mathrm{U}(\mathcal{H})$ denote the set of all unitaries on $\mathcal{H}$.

A linear operator $\mathbf{D}$ on $\mathcal{H}$ is said to be a *density operator* if it satisfies the following properties:

- *Hermitian*: $\mathbf{D}^\dagger = \mathbf{D}$,
- *Positive Semi-definite*: $\langle\phi \mid \mathbf{D} \mid \phi\rangle \geq 0$, for every non-zero $|\phi\rangle \in \mathcal{H}$,
- *Trace-1*: $\mathsf{Tr}(\mathbf{D}) = 1$.

Let $\mathrm{D}(\mathcal{H})$ denote the set of all density operators of $\mathcal{H}$.

Trace and Partial Trace. For any linear operator $\mathbf{L}$ on $\mathcal{H}$ we define the *trace* as the sum of diagonal elements of $\mathbf{L}$, i.e.

$$\mathsf{Tr}(\mathbf{L}) := \sum_i \mathbf{L}_{ii}, \tag{7.25}$$

where $\mathbf{L}_{ii}$ denotes the (i,i)-th element of $\mathbf{L}$.

For any linear operator $\mathbf{L}$ on $\mathcal{H}_1(\mathtt{B}_1) \otimes \mathcal{H}_2(\mathtt{B}_2)$, we define the *partial trace* of $\mathbf{L}$ on $\mathcal{H}_1$ as a linear operator from $\mathcal{H}_1(\mathtt{B}_1) \otimes \mathcal{H}_2(\mathtt{B}_2)$ to $\mathcal{H}_2(\mathtt{B}_2)$,

$$\mathsf{Tr}_{\mathcal{H}_1}(\mathbf{L}) := \sum_{|b_1'\rangle \in \mathtt{B}_1} \left(\langle b_1'| \otimes \mathbf{I}_{\mathcal{H}_2}\right) \mathbf{L} \left(|b_1'\rangle \otimes \mathbf{I}_{\mathcal{H}_2}\right), \tag{7.26}$$

where $\mathbf{I}_{\mathcal{H}_2}$ denotes the identity operator on $\mathcal{H}_2$.

Trace Norm. For any linear operator $\mathbf{L}$ on some finite-dimensional complex Hilbert space $\mathcal{H}$, we define the *trace norm* of $\mathbf{L}$ as

$$\|\mathbf{L}\|_1 = \mathsf{Tr}\left(\sqrt{\mathbf{L}^\dagger\mathbf{L}}\right) = \sum_{i=1}^{r} \sigma_i, \tag{7.27}$$

where $\mathbf{L}^\dagger$ denotes the conjugate transpose of $\mathbf{L}$, and $\sigma_1, \ldots, \sigma_r$ denote the singular values of $\mathbf{L}$, where r denotes the rank of $\mathbf{L}$. Note that, $\mathbf{L}^\dagger\mathbf{L}$ is a positive semi-definite matrix, and thus, its square root is well-defined.

7.2.2 Quantum System, State and Quantum Algorithm

In this thesis, a n-qubit quantum system Q is represented by the 2^n-dimensional Hilbert space $\mathcal{H} = \mathbb{C}^{2^n}$ with the inner product $\langle\cdot|\cdot\rangle$ defined above.

Quantum State. The state of a quantum system Q is given by a density operator ρ_Q of $\mathcal{H}$. A state ρ is said to be *pure* if it can be expressed as $|\psi\rangle\langle\psi|$ for some $|\psi\rangle \in \mathcal{H}$ of unit norm (i.e., $\||\psi\rangle\| = 1$), and *mixed* otherwise. Therefore, a pure state can represented by a unit vector $|\psi\rangle_Q$, where the subscript Q is used to make the concerned quantum register explicit. We will also prefer this latter (simplified) representation whenever possible.

[2] This is equivalent to the conjugate transpose of the $2^n \times 2^n$ complex matrix $\mathbf{U}$ of $\mathbb{C}[\mathtt{B}]$ for some orthonormal basis $\mathtt{B}$.

Computational and Fourier Bases. For any finite set $\mathcal{X} = \{x_1, \ldots, x_k\}$, let $\mathtt{C}_\mathcal{X} = \{|x_1\rangle, \ldots, |x_k\rangle\}$ denote an arbitrarily fixed basis of the k-dimensional complex Hilbert space $\mathcal{H}$ that we refer as the canonical *computational basis* of $\mathcal{H}$ with respect to $\mathcal{X}$. Since the mapping $x \mapsto |x\rangle$ is an obvious bijection from $\mathcal{X}$ to $\mathtt{C}_\mathcal{X}$, we simply write $\mathbb{C}[\mathcal{X}]$ to mean $\mathbb{C}[\mathtt{C}_\mathcal{X}]$. Furthermore we often simplify this to $\mathbb{C}^{|\mathcal{X}|}$ since it is isomorphic to $\mathbb{C}[\mathcal{X}]$. Unless stated otherwise, we always assume a computational basis representation of the underlying space, where the computational basis will be clear from the context.

For any finite abelian group $\mathcal{Y}$ of cardinality M let $\widehat{\mathcal{Y}}$ be its dual group, i.e., the set of all group homomorphisms $\mathcal{Y} \to \{\omega \in C : |\omega| = 1\}$. It is well known that $\mathcal{Y}$ and $\widehat{\mathcal{Y}}$ are isomorphic as groups. The *Fourier basis* $\{\widehat{y} : \widehat{y} \in \widehat{\mathcal{Y}}\}$ of $\mathcal{H}$ is defined by the maps:

$$\widehat{y} \mapsto \frac{1}{\sqrt{M}} \sum_x \widehat{y}(x)^* |x\rangle, \qquad x \mapsto \frac{1}{\sqrt{M}} \sum_{\widehat{y}} \widehat{y}(x) |x\rangle.$$

In general, the function that maps $y \mapsto \widehat{y}$ is called the *quantum Fourier transform* (QFT).

Measurement. Given a pure quantum state $|\psi\rangle_Q$ and an orthonormal basis $\mathtt{B} = \{|b_0\rangle, \ldots, |b_{2^n-1}\rangle\}$ of $\mathcal{H}$, a measurement of $|\psi\rangle_Q$ in the basis $\mathtt{B}$ collapses the state to $|b_i\rangle$ (or simply the label $b_i \in \mathtt{B}$) with probability $|\langle b_i|\psi\rangle|^2$. Although we do not explicitly use it in this thesis, we remark that the probabilistic behavior of measurements can be analogously extended to mixed states using the notion of positive operator-valued measurements.

Joint Quantum State. Given two quantum systems $\mathcal{H}_1$ and $\mathcal{H}_2$, the joint quantum system is given by the tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$. Given $\rho_1 \in \mathrm{D}(\mathcal{H}_1)$ (res. $|\psi_1\rangle \in \mathcal{H}_1$) and $\rho_2 \in \mathrm{D}(\mathcal{H}_2)$ (res. $|\psi_2\rangle \in \mathcal{H}_2$), the product state is given by the density operator $\rho_1 \otimes \rho_2$ (res. $|\psi_1, \psi_2\rangle = |\psi_1\rangle|\psi_2\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ when the state is pure). A quantum system that cannot be written as a product state is called *entangled*.

Example For Small Matrices. We provide an example of a product state in matrix form for more intuition. Assume ρ_1 and ρ_2 are density operators for two quantum systems given by:

$$\rho_1 = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}, \quad \rho_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

The tensor product $\rho_{12} = \rho_1 \otimes \rho_2$ is calculated as follows:

$$\rho_{12} = \rho_1 \otimes \rho_2 = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & 0\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \\ 0\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \end{pmatrix}$$

Simplifying the expression, we get:

$$\rho_{12} = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

7.2.3 (Oracle-Aided) Quantum Algorithm

In the theory of quantum computation, it is assumed that all quantum operations, except for measurements, are unitary operators. Thus, a quantum operation $\mathbf{U}$ is invertible and its inverse is the adjoint $\mathbf{U}^\dagger$.

The Hadamard Transform. Quantum gates generalize classical logic gates and are implemented using unitary operators. In this thesis, the *Hadamard Transform* will be of particular interest. For a single bit the *Hadamard gate* is given by

$$\mathtt{H}|0\rangle := \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \qquad \mathtt{H}|1\rangle := \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

It is well known that the Hadamard transform acting on n-qubits is given by the formula

$$\mathtt{H}^{\otimes n} := \sum_{z \in \{0,1\}^n} (-1)^{x \cdot z} |z\rangle.$$

Thus, the Hadamard transform can be seen as special case of QFT when the underlying group is $\{0,1\}^n$.

Quantum Algorithm. A quantum algorithm $\mathcal{A}$ of depth q is represented as a sequence of unitary operators $\mathbf{U}_1, \ldots, \mathbf{U}_q$ on the space $\mathcal{H}_{in} \otimes \mathcal{H}_{out} \times \mathcal{H}_{work}$, followed by an optional measurement in the computational[3] basis. Here $\mathcal{H}_{in}$, $\mathcal{H}_{out}$, and $\mathcal{H}_{work}$ denote the input space, output space and workspace of $\mathcal{A}$. If the algorithm is initialized in the state ρ_0 then the final state (before measurement), say ρ_q, is given by

$$\rho_q = \mathbf{U}_q \ldots \mathbf{U}_1 \rho_0 \mathbf{U}_1^\dagger \ldots \mathbf{U}_q^\dagger.$$

At this stage, ρ_q is measured and by convention the output is written in the register corresponding to $\mathcal{H}_{out}$.

Oracle Aided Quantum Algorithm. Throughout this chapter, we define $\mathcal{H}_{in} := \mathbb{C}^{2^m}$, $\mathcal{H}_{out} := \mathbb{C}^{2^n}$. Let $\mathcal{H}_{work}$ and $\mathcal{H}_{db}$ be two finite dimensional complex Hilbert spaces. To define the interaction between an algorithm or adversary and an oracle, we need a standard representation of the function $f : \{0,1\}^m \to \{0,1\}^n$. Specifically, f can be realized by the unitary mapping $\mathbf{O}_f$, which is defined by

$$\mathbf{O}_f|x, y\rangle \mapsto |x, y \oplus f(x)\rangle,$$

[3] By our convention, the computational basis can be fixed arbitrarily to suit the measurement basis of the algorithm.

for any $|x, y\rangle \in \mathcal{H}_{in} \otimes \mathcal{H}_{out}$. To represent a stateful oracle, we simply bestow additional qubits to represent the oracle state. Formally, we define

$$\mathbf{O}_f|x, y, d\rangle \mapsto |x, y + f(x), d'\rangle,$$

on the product space $\mathcal{H}_{\mathbf{O}_f} := \mathcal{H}_{in} \otimes \mathcal{H}_{out} \otimes \mathcal{H}_{db}$, where $\{|x, y, d\rangle\}$ denotes the computational basis of $\mathcal{H}_{\mathbf{O}_f}$. Note that, by state of the oracle, we exclusively mean the internal state which is (possibly temporary) and persistent across queries, and ignore any ancillary qubits needed to compute the function itself.

For any quantum *oracle-aided algorithm* $\mathcal{A}$ that makes q black-box queries to a (possibly stateful) oracle $\mathbf{O}_f$, we define the interactive game $\mathcal{A}^{\mathbf{O}_f}$ to be the sequence of $2q$ unitaries: $\mathbf{U}_q\mathbf{O}_f \dots \mathbf{U}_1\mathbf{O}_f$ over the product space $\mathcal{H}_{\mathcal{A}^{\mathbf{O}_f}} = \mathcal{H}_{in} \otimes \mathcal{H}_{out} \otimes \mathcal{H}_{work} \otimes \mathcal{H}_{db}$, where it is implicitly understood that $\mathbf{U}_i$'s operate on $\mathcal{H}_{\mathcal{A}} = \mathcal{H}_{in} \otimes \mathcal{H}_{out} \otimes \mathcal{H}_{work}$ and $\mathbf{O}_f$ operates on $\mathcal{H}_{\mathbf{O}_f}$. We write $\mathcal{A}^{\mathbf{O}_f}[\rho_{\mathcal{A}} \otimes \rho_{\mathbf{O}_f}] = b$ to denote the event that the oracle-aided algorithm $\mathcal{A}$ outputs b after making q queries to oracle $\mathbf{O}_f$, where $\mathcal{A}$ and $\mathbf{O}_f$ are initialized in $\rho_{\mathcal{A}} \in \mathrm{D}(\mathcal{H}_{\mathcal{A}})$ and $\rho_{\mathbf{O}_f} \in \mathrm{D}(\mathcal{H}_{db})$, or jointly as $\rho_{\mathcal{A},\mathbf{O}_f} = \rho_{\mathcal{A}} \otimes \rho_{\mathbf{O}_f}$.

7.2.4 Indistinguishability In The Quantum Setting

As in the classical case, the security of cryptographic schemes can be viewed through the concept of distinguishing games, as discussed in Section 2.2.1. In this section, we adapt the notions to indistinguishability for the quantum setting.

Quantum Distinguishing Advantage. For any two quantum oracles $\mathbf{I}$ and $\mathbf{R}$, we define the distinguishing advantage of any quantum distinguisher[4] $\mathscr{A}$ by

$$\mathbf{Adv}_{\mathbf{I},\mathbf{R}}(\mathscr{A}) := \left|\Pr\left(\mathscr{A}^{\mathbf{I}}[\rho_{\mathscr{A},\mathbf{I}}] = 1\right) - \Pr\left(\mathscr{A}^{\mathbf{R}}[\rho_{\mathscr{A},\mathbf{R}}] = 1\right)\right|,$$

where $\rho_{\mathscr{A},\mathbf{I}}$ and $\rho_{\mathscr{A},\mathbf{R}}$ denote the initial state of $\mathscr{A}^{\mathbf{I}}$ and $\mathscr{A}^{\mathbf{R}}$, respectively.

Computationally Unbounded Adversaries. For any computationally-unbounded $\mathscr{A}$, it is well known that

$$\mathbf{Adv}_{\mathbf{I},\mathbf{R}}(\mathscr{A}) \leq \frac{1}{2}\|\mathrm{Tr}_{\mathcal{H}_{\mathbf{I}_{db}}}(\rho^q_{\mathscr{A},\mathbf{I}}) - \mathrm{Tr}_{\mathcal{H}_{\mathbf{R}_{db}}}(\rho^q_{\mathscr{A},\mathbf{R}})\|_1,$$

where $\rho^q_{\mathscr{A},\mathbf{O}} := \mathscr{A}^{\mathbf{O}}\rho_{\mathscr{A},\mathbf{O}}A^{\mathbf{O}\dagger}$ is the state after q queries to the oracle at-hand $\mathbf{O} \in \{\mathbf{I}, \mathbf{R}\}$. This is result is an adaption of Lemma 2.4.1 to the quantum setting and can easily be derived from the definition of the partial trace.
In addition, without loss of generality, we can assume $\mathscr{A}$ to be deterministic, and thus, define the initial state of $\mathscr{A}$, $\rho_{\mathscr{A}} = |\psi_{\mathscr{A}}\rangle\langle\psi_{\mathscr{A}}|$ for some fixed unit vector $|\psi_{\mathscr{A}}\rangle \in \mathcal{H}_{\mathscr{A}}$.

[4] An oracle-algorithm with binary output.

The Quantum IND-CPA Game. In this thesis, we focus on the notion of quantum IND-CPA (IND-qCPA) which involves a CPA quantum adversary trying to distinguish between a keyed function and a function chosen uniformly at random. This notion is equivalent to the quantum PRF (qPRF) as in [174]. Thus, we will use to two notions interchangeably. As we will discuss later in this chapter, establishing upper bounds for adversarial advantage in the quantum setting, even up to the quantum birthday bound, poses significant challenges.

Definition 7.2.1 (IND-qCPA). *Let $F : \mathcal{K} \times \{0,1\}^m \to \{0,1\}^n$ be a keyed function. The IND-qCPA advantage of some distinguisher $\mathscr{A}$ against F is defined as*

$$\mathbf{Adv}_{F}^{\mathrm{qcpa}}(\mathscr{A}) := \mathbf{Adv}_{\mathbf{O}_{F_k};\mathbf{O}_\Gamma}^{\mathrm{dist}}(\mathscr{A}), \tag{7.28}$$

where $k \twoheadleftarrow_{\$} \mathcal{K}$, and $\Gamma \twoheadleftarrow_{\$} \mathsf{Func}(m;n)$ is a uniform random function.

As we will see in Chapter 8, an additional version of IND-qCPA will be required for non-adaptive quantum adversaries. Specifically, for a non-adaptive distinguisher $\mathscr{A}$, the non-adaptive IND-qCPA advantage is defined analogously as follows:

$$\mathbf{Adv}_{F}^{\mathrm{qncpa}}(\mathscr{A}) := \mathbf{Adv}_{\mathbf{O}_{F_k}^{\otimes q};\mathbf{O}_\Gamma^{\otimes q}}^{\mathrm{dist}}(\mathscr{A}), \tag{7.29}$$

The Quantum PRP and TPRP Notions. Analogously, we can introduce the quantum PRP and quantum TPRP notions.

Definition 7.2.2 (Quantum PRP). *Let $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be some block cipher. The quantum PRP (or qPRP) advantage of some distinguisher $\mathscr{A}$ against E is defined as*

$$\mathbf{Adv}_{E}^{\mathrm{qprp}}(\mathscr{A}) := \mathbf{Adv}_{E_k;\mathsf{P}}^{\mathrm{dist}}(\mathscr{A}), \tag{7.30}$$

where $k \twoheadleftarrow_{\$} \mathcal{K}$, and $\mathsf{P} \twoheadleftarrow_{\$} \mathsf{Perm}(n)$.

Definition 7.2.3 (Quantum TPRP). *Let $\widetilde{E} : \mathcal{T} \times \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be a tweakable block cipher. The quantum TPRP (or qTPRP) advantage of some distinguisher $\mathscr{A}$ against $\widetilde{E}$ is defined as*

$$\mathbf{Adv}_{\widetilde{E}}^{\mathrm{qtprp}}(\mathscr{A}) := \mathbf{Adv}_{\widetilde{E}_k;\widetilde{\mathsf{P}}}^{\mathrm{dist}}(\mathscr{A}), \tag{7.31}$$

where $k \twoheadleftarrow_{\$} \mathcal{K}$, and $\widetilde{\mathsf{P}} \twoheadleftarrow_{\$} \widetilde{\mathsf{Perm}}(\mathcal{K};\{0,1\}^n)$.

The following results by Zhandry [324] and Hosoyamada and Iwata [174] are equivalent to the PRP-PRF and TPRP-PRF switching lemmas for the quantum settings. This will give limited qPRP/qTPRP security bounds for the post-quantum proofs introduced in Section 7.6.

Lemma 7.2.1 (Theorem 7 in [324]). *Let Γ and P denote quantum oracles corresponding to a uniform random function and a uniform random permutation from $\{0,1\}^n$ to $\{0,1\}^n$, respectively. Then, for any q-query quantum adversary $\mathscr{A}$, we have $\mathbf{Adv}_{\Gamma;\mathsf{P}}^{\mathrm{dist}}(\mathscr{A}) \leq O\left(q^3/2^n\right)$.*

Lemma 7.2.2 (Proposition 5 in [174]). *Let Γ denote a uniform random function from $\{0,1\}^{2n}$ to $\{0,1\}^n$, and $\widetilde{\mathsf{P}}$ denote a uniform random permutation of $\{0,1\}^n$ with n-bit tweaks. Then, for any q-query quantum adversary $\mathscr{A}$, we have* $\mathbf{Adv}^{\mathsf{dist}}_{\Gamma;\widetilde{\mathsf{P}}}(\mathscr{A}) \leq O\left(\sqrt{q^6/2^n}\right)$.

7.3 The Compressed Oracle

The random oracle model (ROM) has been a useful heuristic tool for proving the security of cryptographic schemes, especially when standard model proofs are challenging or infeasible in classical contexts. To accurately represent oracle-aided quantum adversaries, it is necessary to develop an equivalent formulation of the ROM for the quantum setting. Boneh et al. emphasize that the proper approach is to permit quantum adversaries to make superposition queries to the oracle, a model known as the quantum random oracle model (QROM) [61]. The idea behind such a model is that quantum adversaries that have access to the implementation of a function f can evaluate it in superposition to extract information, by using quantum algorithms such as: quantum search (Grover's algorithm) [151], collision finding (BHT algorithm) [67], and period finding (Simon's algorithm) [296].

7.3.1 The Recording Barrier

As can be seen in the previous chapters, many classical security proofs rely on the idea that one can copy the adversary's query-response pairs into a recorded *transcript* without the adversary's detection. Unfortunately, one cannot copy a quantum state as this would violate a fundamental result in quantum mechanics called the *No Cloning Theorem* [315]. One could try to record information regarding the adversary's queries into a new register. However, Zhandry [325] showed that such a naive strategy will not work. Indeed, he showed that a naive implementation of lazy sampling will not work as the adversary could distinguish between a quantum random oracle (implementing naive lazy sampling) or with an oracle implementing a classical random function. We adapt the example from Zhandry [325] to our notions.

Example: Naive Quantum Lazy Sampling. Consider a random function f, its implementation $\mathbf{O}_f$ and let $\mathbf{O}$ be a quantum random oracle that tries to record information by using the map $|x,u\rangle \mapsto |x,u\oplus y\rangle \otimes |x,y\rangle$, where y is a uniform random string. Consider now an adversary that queries $\mathbf{O}_f$ with the uniform superposition

$$\sum x,u \in \{0,1\}^n |x,u\rangle \xrightarrow{\mathbf{O}_f} \sum_{x,u\in\{0,1\}^n} |x,u\oplus f(x)\rangle = \sum_{x,u\in\{0,1\}^n} |x,u\rangle. \tag{7.32}$$

In contrast, when the adversary makes the query to $\mathbf{O}$ one has

$$\sum_{x,u\in\{0,1\}^n} |x,u\rangle \xrightarrow{\mathbf{O}} \sum_{x,u\in\{0,1\}^n} |x,u\oplus y\rangle|x,y\rangle. \tag{7.33}$$

Therefore, in the latter, the adversary's register becomes entangled with the oracle's register. Note that by applying the Quantum Fourier Transform (QFT) to (7.32) and measuring the adversary's register will always result in the state $|0,0\rangle$. In contrast, applying QFT to (7.33) and measuring results in a random string (the state will collapse to $|0,0\rangle$ with probability $1/2^n$).

7.3.2 The Original Compressed Oracle

Luckily, Zhandry [325] proposes an original technique to implement a restricted lazy sampling for the quantum random oracle. In this section, we give a brief overview of the original method.

Intuition Behind the Technique Zhandry [325] made a few observations that made it easier to handle a uniform random function $f : \{0,1\}^m \to \{0,1\}^n$ in the quantum setting:

- The unitary $\mathbf{O}_f$ is equivalent to the corresponding phase oracle defined by the map $|x,u\rangle \mapsto (-1)^{u\cdot f(x)}|x,u\rangle$.
- From the adversary's point of view, a uniform random function is equivalent to measuring the uniform superposition over all functions in $\mathsf{Func}(m;n)$. Namely, measuring the state $\sum_f |f\rangle$, which is outside of the adversary's view.

Now, assume the adversary makes a superposition query $\sum_{x,u} \alpha_{x,u}|x,u\rangle$, then the joint system of the oracle and the adversary after the query is given by

$$\sum_{x,u} \alpha_{x,u}|x,u\rangle \otimes \sum_f (-1)^{u\cdot f(x)}|f\rangle.$$

Thus, the query introduces a phase term $(-1)^{u\cdot f(x)}$ to the oracle's state. We consider an alternative way to represent f by encoding its truth table in a vector of length $n\cdot 2^m$. Namely, for the function f, $|f\rangle := |f(0)\rangle|f(1)\rangle\ldots|f(2^m-1)\rangle$. Using this representation we can write $u\cdot f(x)$ as $f\cdot P_{x,u}$ where $P_{x,u}$ is a function that maps x to u and maps to zero otherwise. Thus, the post-query state can be rewritten as

$$\sum_{x,u} \alpha_{x,u}|x,u\rangle \otimes \sum_f (-1)^{f\cdot P_{x,u}}|f\rangle.$$

Generalizing this approach we conclude that q queries introduce the term

$$(-1)^{f\left(P_{x_1,u_1}+\ldots+P_{x_q,u_q}\right)},$$

to the oracle's state. We conclude that working in the Fourier basis each query adds a term of $P_{x,u} \mod 2$ to the oracle state, who is initialized to zero in the Fourier basis. Concluding this approach Zhandry [325] showed that the adversary cannot distinguish between interacting with an oracle that implements a real random function f or if it is simulated as follows:

- The oracle maintains a database $D \in \{0,1\}^{n2^m}$, initially set to zero.
- An oracle query is represented by the map $|x,u\rangle \otimes |D\rangle \mapsto |x,u\rangle \otimes |D \oplus P_{x,u}\rangle$.

Furthermore, this approach can be efficiently simulated by storing the database in a compact manner. Indeed, the main observation is that if we look at the database after q queries, there will be at most q positions where the output register is zero in the Fourier basis. Thus, we can compress the database in the following way: we write down all these positions, which we name set points, and keep track of these positions in truth table component, and ignore the rest as we know the entire truth table, we indicate this by the symbol $\perp$. Finally, Zhandry [325] shows that the oracle representing a random function, referred to as the *standard oracle*, is *perfectly indistinguishable* from the compact version, named the *compressed oracle*.

The Standard and Phase Oracles. For a function $f : \{0,1\}^m \to \{0,1\}^n$ the standard oracle **stO** acting on $n + m + n2^m$ qubits is defined by the map

$$\mathbf{stO} : |x,y\rangle \otimes |f\rangle \mapsto |x, y \oplus f(x)\rangle \otimes |f\rangle$$

Similarly, the phase oracle **phO** is defined as

$$\mathbf{phO} : |x,y\rangle \otimes |f\rangle \mapsto (-1)^{y \cdot f(x)} |x,y\rangle \otimes |f\rangle$$

It is easy to see that **stO** is perfectly indistinguishable from a uniform random function when initialized with the uniform superposition over all functions in $\mathsf{Func}(m;n)$.

The Compressed Oracle. Next, we provide a brief informal overview of how the compressed oracle operates. We consider a database as consisting of pairs (x,y), where x is the input and y, along with pairs $(\perp, 0^n)$ where the database entries are undefined. For $x \in \{0,1\}^m$, we define the cell compression unitary $\mathbf{comp}_x$ which acts on databases the following way:

- For a database D where the number of set points is at most t and $D(x) = \perp$ it is defined as
$$\mathbf{comp}_x |D\rangle := \frac{1}{\sqrt{2^n}} \sum_y |D \cup (x,y)\rangle,$$
where $D \cup (x,y)$ is a new database that adds (x,y) to D.
- For a database D where the number of set points is at exactly t and $D(x) = \perp$ then $\mathbf{comp}_x |D\rangle = |D\rangle$.
- For a database D' where the number of set points is at most t but $D'(x) \neq \perp$ distinguish between two cases.
  - For $z \neq 0$ we define
$$\mathbf{comp}_x \left(\frac{1}{\sqrt{2^n}} \sum_y (-1)^{z \cdot y} |D' \cup (x,y)\rangle \right) := \frac{1}{\sqrt{2^n}} \sum_y (-1)^{z \cdot y} |D' \cup (x,y)\rangle.$$

– Otherwise

$$\mathbf{comp}_x\left(\frac{1}{\sqrt{2^n}}\sum_y |D' \cup (x,y)\rangle\right) := |D' \cup (x,y)\rangle$$

Informally, the last case can be explained as follows. If D is defined on x and the y register does not contain zero in the Fourier basis, there is no need to compress and it operates as the identity. Otherwise, we remove the pair (x,y) from the superposition over all corresponding databases. The last point is important as when the database has no more information on an input x it must be removed. Now we can defined the compressed oracle in the computational basis in its standard or phase form. Following the notations from Zhandry [325], let **CStO′** and **CPhsO′** be the unitaries

$$\begin{aligned}
&\mathbf{comp} : |x,y\rangle \otimes |D\rangle \mapsto |x,y\rangle \otimes \mathbf{comp}_x|D\rangle,\\
&\mathbf{CStO'} : |x,y\rangle \otimes |D\rangle \mapsto |x,y \oplus D(x)\rangle \otimes |D\rangle,\\
&\mathbf{CPhsO'} : |x,y\rangle \otimes |D\rangle \mapsto (-1)^{y\cdot D(x)}|x,y\rangle \otimes |D\rangle.
\end{aligned}$$

Finally, the compressed oracles are defined as follows:

$$\begin{aligned}
\mathbf{CStO} &:= \mathbf{comp} \circ \mathbf{CStO'} \circ \mathbf{comp} \circ \mathbf{Increase},\\
\mathbf{CPhsO} &:= \mathbf{comp} \circ \mathbf{CPhsO'} \circ \mathbf{comp} \circ \mathbf{Increase},
\end{aligned}$$

where **Increase** is defined by $|x,y\rangle \otimes |D\rangle \mapsto |x,y\rangle \otimes |D\rangle|(\perp, 0^n)\rangle$.

The following Lemma finalizes the original compressed oracle technique by Zhandry [325].

Lemma 7.3.1 (Lemma 4 in [325]). **CStO** *and* **stO** *are perfectly indistinguishable and* **CPhsO** *and* **phO** *are perfectly indistinguishable. In other words, for any adversary $\mathscr{A}$, one has*

$$\Pr\left(\mathscr{A}^{\mathbf{CStO}} = 1\right) = \Pr\left(\mathscr{A}^{\mathbf{stO}} = 1\right), \qquad \Pr\left(\mathscr{A}^{\mathbf{CPhsO}} = 1\right) = \Pr\left(\mathscr{A}^{\mathbf{phO}} = 1\right).$$

7.3.3 A Refinement of The Compressed Oracle

Although Zhandry's [325] compressed oracle is a powerful tool for analyzing quantum algorithms in the QROM model, it lacks mathematical rigor and is challenging to apply to concrete schemes. In 2020, Chung et al. [83] proposed a reformulation of the compressed oracle technique that is both mathematically rigorous and sufficiently abstract. In this thesis, we present a refinement of Chung et al.'s formulation, which will serve as a foundation for providing post-quantum proofs for specific symmetric schemes. We adopt the notations from [83] and adapt them to our specific needs.

Notations. Let $\mathcal{Y}$ denote $\{0,1\}^n$ and define $\mathtt{C}_{\mathcal{Y}}$ to be the computational basis of the n-qubit space $\mathbb{C}^{2^n}$. Let $\widehat{\mathcal{Y}}$ denote the dual group of $\mathcal{Y}$. It is known that the group homomorphisms in the dual group are of the type: $\widehat{y}(z)(y) := (-1)^{y\cdot z}$. We assume $\widehat{\mathcal{Y}}$ to be an additive group with the group operation $\widehat{y} + \widehat{z} := \widehat{y \oplus z}$. Naturally, $\widehat{0}$ denotes the identity.

Recall that $\mathtt{F}_{\mathcal{Y}} := \{|\widehat{y}\rangle\}$ is referred as the Fourier basis of $\mathbb{C}^{2^n}$. Recall that the Hadamard transform maps the computational basis to the Fourier basis. For any $\widehat{y} \in \widehat{\mathcal{Y}}$, the map is given by the formula

$$|\widehat{y}\rangle := \frac{1}{2^{n/2}} \sum_{z \in \mathcal{Y}} \widehat{y}(z)|z\rangle = \frac{1}{2^{n/2}} \sum_{z \in \mathcal{Y}} (-1)^{y \cdot z}|z\rangle,$$

and for any $y \in \mathcal{Y}$ the reverse map is given by

$$|y\rangle := \frac{1}{2^{n/2}} \sum_{\widehat{z} \in \widehat{\mathcal{Y}}} \widehat{z}(y)|\widehat{z}\rangle = \frac{1}{2^{n/2}} \sum_{\widehat{z} \in \widehat{\mathcal{Y}}} (-1)^{z \cdot y}|\widehat{z}\rangle.$$

Next, let $\mathcal{Z}$ denote the set $\mathcal{Y} \cup \{\perp\}$ for a special symbol $\perp$; similarly $\widehat{\mathcal{Z}}$ will denote $\widehat{\mathcal{Y}} \cup \{\perp\}$. We also choose a corresponding norm-1 vector $|\perp\rangle$ orthogonal to $\mathbb{C}^{2^n}$, so that the span of both $\mathtt{C}_{\mathcal{Z}} := \{|y\rangle \mid y \in \mathcal{Z}\}$ and $\mathtt{F}_{\mathcal{Z}} := \{|\widehat{y}\rangle \mid \widehat{y} \in \widehat{\mathcal{Z}}\}$ is $\mathbb{C}^{2^{n}+1}$; we'll call $\mathtt{C}_{\mathcal{Z}}$ and $\mathtt{F}_{\mathcal{Z}}$ the computational basis and Fourier basis respectively of the extended space $\mathbb{C}^{2^{n}+1}$.

Functions, Partial Functions and Databases. Let $\mathcal{X}$ denote $\{0,1\}^m$ for some arbitrary m. The *quantum truth table* of $f \in \mathsf{Func}(m;n)$ is defined as

$$|f\rangle := \bigotimes_{x \in \mathcal{X}} |x\rangle|f(x)\rangle.$$

Let $\widehat{\mathcal{F}}$ denote the set of *Fourier* functions $\widehat{f} : \mathcal{X} \longrightarrow \widehat{\mathcal{Y}}$. The quantum truth table of $\mathtt{H}$ is defined similarly as

$$|\widehat{f}\rangle := \bigotimes_{x \in \mathcal{X}} |x\rangle|\widehat{f}(x)\rangle.$$

For a subset $\mathcal{S} \subseteq \mathcal{X}$, a function $f : \mathcal{S} \longrightarrow \mathcal{Y}$ will be called a *partial function* from $\mathcal{X}$ to $\mathcal{Y}$. A partial function f can be extended to a function $d_f : \mathcal{X} \longrightarrow \mathcal{Z}$ by defining $d_f(y) = \perp$ for all $y \in \mathcal{X} \setminus \mathcal{S}$. We call d_f the *database* representing f, with $\perp$ denoting the cells where f is not defined. (When f is a full function, d_f coincides with f.) The database will also be represented as a quantum truth table

$$|d_f\rangle := \bigotimes_{x \in \mathcal{X}} |x\rangle|d_f(x)\rangle.$$

Similarly, we define partial Fourier functions $\widehat{f} : \mathcal{S} \longrightarrow \widehat{\mathcal{Y}}$, databases $d_{\widehat{f}} : \mathcal{X} \longrightarrow \widehat{\mathcal{Z}}$ representing partial Fourier functions, and their quantum truth tables $|d_{\widehat{f}}\rangle$. When f and $\widehat{f}$ are clear from context, we'll find it convenient to drop the subscripts and write d_f and $d_{\widehat{f}}$ simply as d and $\widehat{d}$ respectively. We'll write $\mathcal{D}$ (resp. $\widehat{\mathcal{D}}$) to denote the set of all databases $d : \mathcal{X} \longrightarrow \mathcal{Z}$ (resp. all Fourier databases $\widehat{d} : \mathcal{X} \longrightarrow \widehat{\mathcal{Z}}$). When convenient we will treat a database d as a relation on $\mathcal{X} \times \mathcal{Y}$ and write $(x, y) \in \mathcal{D}$ to denote $d(x) = y$; $|d|$ will then denote the size of this relation, i.e., the size of $\{x \in \mathcal{X} \mid d(x) \in \mathcal{Y}\}$.

For any function $f \in \mathsf{Func}(m;n)$, let $\widehat{f} \in \widehat{\mathcal{F}}$ be defined as the map $x \mapsto \widehat{f(x)}$. The following technical lemma proves that $\{|f\rangle \mid f \in \mathsf{Func}(m;n)\}$ and $\{|f\rangle \mid \widehat{f} \in \widehat{\mathcal{F}}\}$ span the same space (isomorphic to $\mathbb{C}^{2^{n2^m}}$).

Lemma 7.3.2. *One has*

$$|\widehat{f}\rangle = \frac{1}{2^{n2^m/2}} \sum_g (-1)^{f\cdot g}|g\rangle,$$

where $f \cdot g$ is defined as $\sum_{x\in\mathcal{X}} f(x)\cdot g(x)$.

Proof. From the definition of $|\widehat{f}\rangle$, one has

$$\begin{aligned}
|\widehat{f}\rangle &= \bigotimes_{x\in\mathcal{X}} |x\rangle|\widehat{f}(x)\rangle = \bigotimes_{x\in\mathcal{X}} |x\rangle|\widehat{f(x)}\rangle \\
&= \bigotimes_{x\in\mathcal{X}} \left(\frac{1}{2^{n/2}} \sum_{y\in\mathcal{Y}} (-1)^{f(x)\cdot y}|x\rangle|y\rangle \right) \\
&= \frac{1}{2^{n2^m/2}} \sum_{y_0,\ldots,y_{2^n-1}\in\mathcal{Y}} \left[\bigotimes_{x\in\mathcal{X}} (-1)^{f(x)\cdot y_x}|x\rangle|y_x\rangle \right] \\
&= \frac{1}{2^{n2^m/2}} \sum_g \left[\bigotimes_{x\in\mathcal{X}} (-1)^{f(x)\cdot g(x)}|x\rangle|g(x)\rangle \right] \\
&= \frac{1}{2^{n2^m/2}} \sum_g (-1)^{f\cdot g}|g\rangle.
\end{aligned}$$

□

Similarly we can show that $\{|d\rangle \mid d \in \mathcal{D}\}$ and $\{|\widehat{d}\rangle \mid \widehat{d} \in \widehat{\mathcal{D}}\}$ span the same space isomorphic to $\mathbb{C}^{(2^n+1)^{2^m}}$; we call this space the *database space* $\mathbb{D}$. Letting $\mathbf{0}$ denote the constant 0^n function and observing that $\mathbf{0}\cdot g = 0$ for any $g \in \mathsf{Func}(m;n)$, we have

$$|\widehat{\mathbf{0}}\rangle = \frac{1}{2^{n2^m/2}} \sum_g |g\rangle,$$

is the uniform superposition over all functions in $\mathsf{Func}(m;n)$.

The Fourier Oracle. Note that, according to our notations, the standard oracle is a stateful oracle with Hilbert space $\mathcal{H}_{db} = \mathbb{C}[\mathsf{Func}(m;n)]$. Given a truth-table representation $|f\rangle$ of a function $f \in \mathsf{Func}(m;n)$, it acts on the adversary registers $|x\rangle|y\rangle$ and the truth-table registers $|f\rangle$ as

$$\mathbf{stO}|x\rangle|y\rangle \otimes |f\rangle = |x\rangle|y \oplus f(x)\rangle \otimes |f\rangle. \tag{7.34}$$

Thus, it follows from the definition of **stO** that it is perfectly indistinguishable from a uniform random function when the truth table register is initialized in $|\widehat{\mathbf{0}}\rangle$. The following Lemma demonstrates the behavior of the standard oracle **stO** when the adversary's response register and the truth table register are in the Fourier basis.

Lemma 7.3.3. *For any $x \in \mathcal{X}$, $y \in \mathcal{Y}$ and $f \in \mathsf{Func}(m;n)$ one has*

$$\mathbf{stO}|x\rangle|\widehat{y}\rangle \otimes |\widehat{f}\rangle = |x\rangle|\widehat{y}\rangle \otimes |\widehat{f} + \widehat{\delta}_{xy}\rangle,$$

where δ_{xy} is the function in $\mathsf{Func}(m;n)$ defined as

$$\delta_{xy}(z) = \begin{cases} y & \text{when } z = x, \\ 0 & \text{otherwise}, \end{cases}$$

and the operations $\oplus$ in $\mathsf{Func}(m;n)$ and $+$ in $\widehat{\mathcal{F}}$ are defined point-wise.

Proof. Substituting the definitions of $|\widehat{y}\rangle$ and $|\widehat{f}\rangle$ in the oracle equation of **stO** gives

$$\begin{aligned}
&\mathbf{stO}|x\rangle|\widehat{y}\rangle \otimes |\widehat{f}\rangle \\
&= \mathbf{stO}|x\rangle \frac{1}{2^{n/2}} \left(\sum_{z\in\mathcal{Y}} (-1)^{y\cdot z}|z\rangle \right) \otimes \left[\frac{1}{2^{n2^m/2}} \sum_g (-1)^{f\cdot g}|g\rangle \right] \\
&= \frac{1}{2^{n(2^m+1)/2}} \sum_{z\in\mathcal{Y}} \sum_g (-1)^{y\cdot z\oplus f\cdot g} \left(\mathbf{stO}|x\rangle|z\rangle \otimes |g\rangle \right) \\
&= \frac{1}{2^{n(2^m+1)/2}} \sum_{z\in\mathcal{Y}} \sum_g (-1)^{y\cdot z\oplus f\cdot g}|z\rangle|z\oplus g(x)\rangle \otimes |g\rangle \\
&= \frac{1}{2^{n(2^m+1)/2}} \sum_{z'\in\mathcal{Y}} \sum_g (-1)^{y\cdot (z'\oplus g(x))\oplus f\cdot g}|x\rangle|z'\rangle \otimes |g\rangle \\
&= \frac{1}{2^{n(2^m+1)/2}} \sum_{z'\in\mathcal{Y}} \sum_g (-1)^{y\cdot z'\oplus (f\oplus\delta_{xy})\cdot g}|x\rangle|z'\rangle \otimes |g\rangle \\
&= |x\rangle \frac{1}{2^{n/2}} \left(\sum_{z'\in\mathcal{Y}} (-1)^{y\cdot z'}|z'\rangle \right) \otimes \left[\frac{1}{2^{n2^m/2}} \sum_g (-1)^{(f\oplus\delta_{xy})\cdot g}|g\rangle \right] \\
&= |x\rangle|\widehat{y}\rangle \otimes |\widehat{f\oplus\delta_{xy}}\rangle = |x\rangle|\widehat{y}\rangle \otimes |\widehat{f}+\widehat{\delta_{xy}}\rangle.
\end{aligned}$$

□

We define the operator $\mathbf{O}_{x\widehat{y}}$ on the truth-table register as $\mathbf{O}_{x\widehat{y}}|\widehat{f}\rangle := |\widehat{f}+\widehat{\delta}_{xy}\rangle$. Finally, the standard oracle can by written as

$$\mathbf{stO}|x\rangle|\widehat{y}\rangle \otimes |\widehat{f}\rangle = |x\rangle|\widehat{y}\rangle \otimes \mathbf{O}_{x\widehat{y}}|\widehat{f}\rangle.$$

The Compressed Oracle In The Fourier Basis. For any $x\ in\mathcal{X}$, the *cell compression* unitary $\mathbf{comp}_x$ on $\mathbb{C}^{2^n+1}$ is defined on the basis $\mathtt{F}_{\mathcal{Z}}$ as

$$\mathbf{comp}_x := |\perp\rangle\langle\widehat{0}| + |\widehat{0}\rangle\langle\perp| + \sum_{\widehat{y}\in\widehat{\mathcal{Y}}\setminus\{\widehat{0}\}} |\widehat{y}\rangle\langle\widehat{y}|.$$

Consequently, for any $\widehat{y} \in \widehat{\mathcal{Y}}$, the action of $\mathbf{comp}_0$ on the Fourier basis elements is given by

$$\mathbf{comp}_x|\widehat{y}\rangle = \begin{cases} |\perp\rangle, & \widehat{y} = \widehat{0} \\ |\widehat{0}\rangle, & \widehat{y} = \perp \\ |\widehat{y}\rangle, & \text{otherwise.} \end{cases}$$

The *database compression* unitary **comp** on $\mathbb{D}$ is defined as

$$\mathbf{comp} := \bigotimes_{x \in \mathcal{X}} \mathbf{comp}_x.$$

The *compressed oracle* **cO** is a stateful oracle with $\mathcal{H}_{db} = \mathbb{D}$. It acts on the adversary's registers and the oracle's database registers as

$$\mathbf{cO} := (\mathbf{I}_{\mathbb{C}[\mathcal{X}]\otimes\mathbb{C}[\widehat{\mathcal{Y}}]} \otimes \mathbf{comp}) \circ \mathbf{stO} \circ (\mathbf{I}_{\mathbb{C}[\mathcal{X}]\otimes\mathbb{C}[\widehat{\mathcal{Y}}]} \otimes \mathbf{comp}).$$

Thus, for a database $\widehat{d}$ we have

$$\mathbf{cO}|x\rangle|\widehat{y}\rangle \otimes |\widehat{d}\rangle = |x\rangle|\widehat{y}\rangle \otimes \mathbf{cO}_{x\widehat{y}}|\widehat{d}\rangle,$$

where $\mathbf{cO}_{x\widehat{y}} := \mathbf{comp}_x \circ \mathbf{O}_{x\widehat{y}} \circ \mathbf{comp}_x$.

7.4 The Two-Domain Distance Technique

In 2019, Hosoyamada and Iwata [174] showed that the 4 round Luby-Rackoff construction, a Feistel network, is a quantum pseudorandom permutation. They introduced a framework for proving post-quantum security schemes, which is a variant of Zhandry's original compressed oracle [325]. While this framework is inherently interesting, it is cumbersome and involves complex, lengthy calculations, making it difficult to generalize to other schemes.

In this section, we build on the interpretation of Chung et al. [83] for the indistinguishability setting and propose a generic framework to represent both ideal and real-world oracles using a single compressed oracle. By combining this framework with the ideas from Hosoyamada and Iwata [174], we develop a quantum analog of the "identical-up-to-bad" technique, referred to as the two-domain distance lemma.

7.4.1 The Transition Capacity Bound

Domain-Restricted Databases. For a subset $\widetilde{\mathcal{X}}$ of $\mathcal{X}$ we will write $\mathcal{D}|_{\widetilde{\mathcal{X}}}$ to denote the set of databases restricted to $\widetilde{\mathcal{X}}$, defined equivalently as $\{d|_{\widetilde{\mathcal{X}}} \mid d \in \mathcal{D}\}$ or the set of databases $d : \widetilde{\mathcal{X}} \longrightarrow \mathcal{Z}$. Since $\mathcal{D}$ is a basis of the database space $\mathbb{D}$, a domain-restricted database space will span a subspace of $\mathbb{D}$ isomorphic to $mplex^{(2^n+1)^{|\widetilde{\mathcal{X}}|}}$. We continue to represent elements of $\widetilde{\mathcal{X}}$ as m-bit numbers.

Transition Capacity. For a domain-restricted database-set $\mathcal{D}|_{\widetilde{\mathcal{X}}}$, a subset $\mathcal{P} \subseteq \mathcal{D}|_{\widetilde{\mathcal{X}}}$ will be called a *database property* on $\mathcal{D}|_{\widetilde{\mathcal{X}}}$, with a corresponding projection defined as

$$\Pi_{\mathcal{P}} := \sum_{d \in \mathcal{P}} |d\rangle\langle d|.$$

For a database $d \in \mathcal{D}|_{\widetilde{\mathcal{X}}}$ and an $x \in \widetilde{\mathcal{X}}$ define

$$d|^x := \{d' \in \mathcal{D}|_{\widetilde{\mathcal{X}}} : d'(x') = d(x'), \forall x' \in \widetilde{\mathcal{X}} \setminus \{x\}\}.$$

In other words, $d|^x$ is the set of databases in $\mathcal{D}|_{\widetilde{\mathcal{X}}}$ which are identical to d except (possibly) at x. Note that since d (resp. x) is also in $\mathcal{D}$ (resp. $\mathcal{X}$), $d|^x$ is only well-defined when we specify $\mathcal{D}|_{\widetilde{\mathcal{X}}}$ as well; however, since $\mathcal{D}|_{\widetilde{\mathcal{X}}}$ will usually be clear from the context, for convenience we leave the dependence of $d|^x$ on $\mathcal{D}|_{\widetilde{\mathcal{X}}}$ implicit.

Definition 7.4.1 (Transition Capacity)**.** *For two properties $\mathcal{P}$ and $\mathcal{P}'$, the* transition capacity *from $\mathcal{P}$ to $\mathcal{P}'$ is defined as*

$$[\![\mathcal{P} \hookrightarrow \mathcal{P}']\!] := \max_{x\in\widetilde{\mathcal{X}},\widehat{y}\in\widehat{\mathcal{Y}},d\in\mathcal{D}|_{\widetilde{\mathcal{X}}}} \|\Pi_{\mathcal{P}'\cap d|^x} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}\cap d|^x}\|.$$

The transition capacity $[\![\mathcal{P} \hookrightarrow \mathcal{P}']\!]$ is roughly a measure of an upper bound for how likely it can be that a database in $\mathcal{P}$ will transition into a database in $\mathcal{P}'$ after a single query to $\mathbf{cO}$.

For any property $\mathcal{P}$ let $\bar{\Pi}_{\mathcal{P}} := \mathbf{I}_{m+n} \otimes \Pi_{\mathcal{P}}$. We adapt the following useful proposition from an intermediate result in [83, Proof of Lemma 5.6].

Lemma 7.4.1. *For any pair of properties $\mathcal{P}$ and $\mathcal{P}'$,*

$$[\![\mathcal{P} \hookrightarrow \mathcal{P}']\!] \geq \|\bar{\Pi}_{\mathcal{P}'} \circ \mathbf{cO} \circ \bar{\Pi}_{\mathcal{P}}\|.$$

Proof. Using Lemma A.3.1 one has

$$\|\bar{\Pi}_{\mathcal{P}'} \circ \mathbf{cO} \circ \bar{\Pi}_{\mathcal{P}}\| \leq \max_{x\in\widetilde{\mathcal{X}},\widehat{y}\in\widehat{\mathcal{Y}}} \|\Pi_{\mathcal{P}'} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}}\|. \tag{7.35}$$

Fix any $x \in \widetilde{\mathcal{X}}$, $\widehat{y} \in \widehat{\mathcal{Y}}$, and $d \in \mathcal{D}|_{\widetilde{\mathcal{X}}}$. Then, by the definition of $d|^x$, for any $|\Delta\rangle \in \mathbb{C}[\mathcal{D}|_{\widetilde{\mathcal{X}}}]$, we have $\mathbf{cO}_{x\widehat{y}}|\Delta\rangle \in \mathbb{C}[d|^x]$, i.e., $\mathbf{cO}_{x\widehat{y}}$ is a unitary on $\mathbb{C}[\mathcal{D}|_{\widetilde{\mathcal{X}}}]$. Thus, for any $|\Delta\rangle \in \mathbb{C}[\mathcal{D}|_{\widetilde{\mathcal{X}}}]$,

$$\begin{aligned}\Pi_{\mathcal{P}'} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}}|\Delta\rangle &= \Pi_{\mathcal{P}'} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}\cap d|^x}|\Delta\rangle \\ &= \Pi_{\mathcal{P}'\cap d|^x} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}\cap d|^x}|\Delta\rangle,\end{aligned}$$

where for the last equality we use the fact that $\Pi_{\mathcal{P}\cap d|^x}|\Delta\rangle \in \mathbb{C}[\mathcal{D}|_{\widetilde{\mathcal{X}}}]$, and thus $\mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}\cap d|^x}|\Delta\rangle \in \mathbb{C}[\mathcal{D}|_{\widetilde{\mathcal{X}}}]$. Thus, for any $x, \widehat{y}$, we have

$$\begin{aligned}\|\Pi_{\mathcal{P}'} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}}\| &= \sup_{|\Delta\rangle\in\mathbb{C}[\mathcal{D}|_{\widetilde{\mathcal{X}}}]} \|\Pi_{\mathcal{P}'} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}}|\Delta\rangle\| \\ &= \max_{d\in\mathcal{D}|_{\widetilde{\mathcal{X}}}} \sup_{|\Delta\rangle\in\mathbb{C}[\mathcal{D}|_{\widetilde{\mathcal{X}}}]} \|\Pi_{\mathcal{P}'} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}}|\Delta\rangle\| \\ &= \max_{d\in\mathcal{D}|_{\widetilde{\mathcal{X}}}} \sup_{|\Delta\rangle\in\mathbb{C}[\mathcal{D}|_{\widetilde{\mathcal{X}}}]} \|\Pi_{\mathcal{P}'\cap d|^x} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}\cap d|^x}|\Delta\rangle\| \\ &= \max_{d\in\mathcal{D}|_{\widetilde{\mathcal{X}}}} \|\Pi_{\mathcal{P}'\cap d|^x} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}\cap d|^x}\|,\end{aligned} \tag{7.36}$$

where for the last equality we observe that $\Pi_{\mathcal{P}'\cap d|^x} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}\cap d|^x}$ takes any state orthogonal to $\mathbb{C}[\mathcal{D}|_{\widetilde{\mathcal{X}}}]$ to 0, so for any $|\Delta\rangle \in \mathbb{C}[\mathcal{D}|_{\widetilde{\mathcal{X}}}]$ we have $|\Delta'\rangle := \Pi_{d|^x}|\Delta\rangle \in \mathbb{C}[\mathcal{D}|_{\widetilde{\mathcal{X}}}]$ such that

$$\|\Pi_{\mathcal{P}'\cap d|^x} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}\cap d|^x}|\Delta\rangle\| \leq \|\Pi_{\mathcal{P}'\cap d|^x} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}\cap d|^x}|\Delta'\rangle\|.$$

Plugging (7.36) in (7.35) gives

$$\|\bar{\Pi}_{\mathcal{P}'} \circ \mathbf{cO} \circ \bar{\Pi}_{\mathcal{P}}\| \leq \max_{x \in \widetilde{\mathcal{X}}, \widehat{y} \in \widehat{\mathcal{Y}}, d \in \mathcal{D}|_{\widetilde{\mathcal{X}}}} \|\Pi_{\mathcal{P}' \cap d|^x} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P} \cap d|^x}\| = [\![\mathcal{P} \hookrightarrow \mathcal{P}']\!],$$

thus establishing the proposition. □

For a property $\mathcal{P} \subseteq \mathcal{D}|_{\widetilde{\mathcal{X}}}$, let $\mathcal{P}^c$ denote its complement, i.e., $\mathcal{D}|_{\widetilde{\mathcal{X}}} \setminus \mathcal{P}$. The following lemma establishes a relation between the transition capacity from $\mathcal{P}^c$ to $\mathcal{P}$ and the existence of an intermediate property (who's size will be easy to upper bound) that lies between these two properties. This lemma named *Transition Capacity Bound* is adapted from [83, Theorem 5.17].

Lemma 7.4.2 (Transition Capacity Bound)**.** *Let $\mathcal{P}, \mathcal{P}'$ be properties on $\mathcal{D}|_{\widetilde{\mathcal{X}}}$ such that for every $x \in \widetilde{\mathcal{X}}$ and $d \in \mathcal{D}|_{\widetilde{\mathcal{X}}}$, we can find a set $\mathcal{S}_{x,d}^{\mathcal{P}^c \hookrightarrow \mathcal{P}'} \subseteq \mathcal{Y}$ satisfying*

$$\mathcal{P}' \cap d|^x \subseteq \{d' \in d|^x \mid d'(x) \in \mathcal{S}_{x,d}^{\mathcal{P}^c \hookrightarrow \mathcal{P}'}\} \subseteq \mathcal{P} \cap d|^x. \tag{7.37}$$

In other words, for any database $d' \in d|^x$,

$$d' \in \mathcal{P}' \Rightarrow d'(x) \in \mathcal{S}_{x,d}^{\mathcal{P}^c \hookrightarrow \mathcal{P}'} \Rightarrow d' \in \mathcal{P}.$$

Then we have

$$[\![\mathcal{P}^c \hookrightarrow \mathcal{P}']\!] \leq \max_{x \in \widetilde{\mathcal{X}}, d \in \mathcal{D}|_{\widetilde{\mathcal{X}}}} \sqrt{\frac{10|\mathcal{S}_{x,d}^{\mathcal{P}^c \hookrightarrow \mathcal{P}'}|}{2^n}}.$$

Setup for The Proof of Lemma 7.4.2. In order to prove Lemma 7.4.2, we require some more setup and a counting argument borrowed from [83]. We start by defining the unitary that acts on the cell $|d(x)\rangle$ when $\mathbf{cO}_{x\widehat{y}}$ acts on $|d\rangle$. Let $\mathbf{V}_{\widehat{y}}$ be the unitary defined on the Fourier basis $\mathtt{F}_{\mathcal{Y}}$ as

$$\mathbf{V}_{\widehat{y}}|\widehat{z}\rangle := |\widehat{z} + \widehat{y}\rangle = \left|\widehat{z \oplus y}\right\rangle.$$

Then we can write

$$\mathbf{O}_{x\widehat{y}} = \bigotimes_{\widetilde{\mathcal{X}}} \left[|x\rangle\langle x| \otimes \mathbf{V}_{\widehat{y}} + (\mathbf{I}_m - |x\rangle\langle x|) \otimes \mathbf{I}_n\right],$$

which applies the same cell unitary $|x\rangle\langle x| \otimes \mathbf{V}_{\widehat{y}} + (\mathbf{I}_m - |x\rangle\langle x|) \otimes \mathbf{I}_n$ to every cell. For the cell $|x\rangle|d(x)\rangle$, this cell unitary is identical to $I_m \otimes \mathbf{V}_{\widehat{y}}$, while for all other cells it is identical to $\mathbf{I}_{m+n}$. Thus we can more simply write

$$\mathbf{O}_{x\widehat{y}} = \mathbf{I}_{m+n} \otimes \ldots \otimes \mathbf{I}_{m+n} \otimes (\mathbf{I}_m \otimes \mathbf{V}_{\widehat{y}}) \otimes \mathbf{I}_{m+n} \otimes \ldots \otimes \mathbf{I}_{m+n}.$$

We extend $\mathbf{V}_{\widehat{y}}$ to $\mathtt{F}_{\mathcal{Z}}$ by defining

$$\mathbf{V}_{\widehat{y}}|\perp\rangle = |\perp\rangle.$$

Next we define

$$\mathbf{cV}_{\widehat{y}} := \mathbf{comp}_x \circ \mathbf{V}_{\widehat{y}} \circ \mathbf{comp}_x.$$

Recalling that

$$\mathbf{comp} = \bigotimes_{x \in \widetilde{\mathcal{X}}} \mathbf{comp}_x,$$

we have

$$\begin{aligned}
\mathbf{cO}_{x\widehat{y}} &= \mathbf{comp} \circ \mathbf{O}_{x\widehat{y}} \circ \mathbf{comp} \\
&= \bigotimes_{\widetilde{\mathcal{X}}} \left[|x\rangle\langle x| \otimes \mathbf{cV}_{\widehat{y}} + (\mathbf{I}_m - |x\rangle\langle x|) \otimes \mathbf{I}_n \right] \\
&= \mathbf{I}_{m+n} \otimes \ldots \otimes \mathbf{I}_{m+n} \otimes (\mathbf{I}_m \otimes \mathbf{cV}_{\widehat{y}}) \otimes \mathbf{I}_{m+n} \otimes \ldots \otimes \mathbf{I}_{m+n}.
\end{aligned}$$

Note that even though $\mathbf{O}_{x\widehat{y}}$ and $\mathbf{cO}_{x\widehat{y}}$ are defined on the entire $\mathbb{C}[\mathcal{D}]$ and not just $\mathbb{C}[\mathcal{D}|_{\widetilde{\mathcal{X}}}]$, in these calculations we continue to ignore the cells with labels outside $\widetilde{\mathcal{X}}$; since we are only dealing with databases restricted to $\widetilde{\mathcal{X}}$, the other cells will always remain empty at the beginning of each oracle call and will get set back to empty at the end of each oracle call, and hence won't affect our computations.

The transition matrix of $\mathbf{cV}_{\widehat{y}}$ is described in detail in [83, Lemma 4.3] (and is in fact also implicitly derived in [174, Proposition 2]). For our purposes it will be sufficient to borrow a combinatorial result from [83], which states

Lemma 7.4.3 (Equation 8 in [83])**.** *For any subset $\mathcal{S} \subseteq \mathcal{Y}$,*

$$\sum_{w \in \mathcal{S}, z \in \mathcal{Z}, z \neq w} |\langle w \,|\, \mathbf{cV}_{\widehat{y}} \,|\, z \rangle|^2 \leq \frac{10|\mathcal{S}|}{2^n}.$$

Note that the condition $\mathcal{S} \subseteq \mathcal{Y}$ is important, as this result may not hold when $\perp \in \mathcal{S}$. Using this result, we can now proceed to prove Lemma 7.4.2.

Proof. Fix $x \in \widetilde{\mathcal{X}}$ and $d \in \mathcal{D}|_{\widetilde{\mathcal{X}}}$. Let $\mathcal{S}$ denote $\mathcal{S}_{x,d}^{\mathcal{P}^c \hookrightarrow \mathcal{P}'}$, and $\Pi_{\mathcal{S}}$ denote the projection onto v, defined by

$$\Pi_{\mathcal{S}} := \sum_{y \in \mathcal{S}} |y\rangle\langle y|.$$

Let $\mathcal{P}_\dagger$ denote the property $\{d' \in d|^x \mid d'(x) \in \mathcal{S}\}$. Then we have

$$\Pi_{\mathcal{P}_\dagger} = \sum_{d \in \mathcal{P}_\dagger} |d\rangle\langle d| = \bigotimes_{x' \in \widetilde{\mathcal{X}}} \left(|x\rangle\langle x| \otimes \Pi_{\mathcal{S}} + |x'\rangle\langle x'| \otimes |d(x')\rangle\langle d(x')| \right).$$

Since $\mathcal{P}' \cap d|^x \subseteq \mathcal{P}_\dagger$, we have $\Pi_{\mathcal{P}' \cap d|^x} \circ \Pi_{\mathcal{P}_\dagger} = \Pi_{\mathcal{P}' \cap d|^x}$. Moreover, since $\mathcal{P}^c \cap d|^x \subseteq \mathcal{P}_\dagger^c$, we have $\Pi_{\mathcal{P}_\dagger^c} \circ \Pi_{\mathcal{P}^c \cap d|^x} = \Pi_{\mathcal{P}^c \cap d|^x}$. Then for any $\widehat{y} \in \widehat{\mathcal{Y}}$ we have

$$\begin{aligned}
\| \Pi_{\mathcal{P}' \cap d|^x} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}^c \cap d|^x} \| &= \| \Pi_{\mathcal{P}' \cap d|^x} \circ \Pi_{\mathcal{P}_\dagger} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}_\dagger^c} \circ \Pi_{\mathcal{P}^c \cap d|^x} \| \\
&\leq \| \Pi_{\mathcal{P}_\dagger} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}_\dagger^c} \|.
\end{aligned}$$

The last inequality comes from two simple facts: $\|\Pi\| \leq 1$ for any projection Π, and that the matrix norm is sub-multiplicative. Applying $\Pi_{\mathcal{P}_\dagger} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}_\dagger^c}$ to a database is

equivalent to applying $\Pi_{\mathcal{S}} \circ \mathbf{cV}_{\widehat{y}} \circ (\mathbf{I}_n - \Pi_{\mathcal{S}})$ to the cell labeled x and $\mathbf{I}_{m+n}$ to all other cells. Thus,

$$\begin{aligned}
\|\Pi_{\mathcal{P}' \cap d|^x} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}^c \cap d|^x}\| &\leq \|\Pi_{\mathcal{S}} \circ \mathbf{cV}_{\widehat{y}} \circ (\mathbf{I}_n - \Pi_{\mathcal{S}})\| \\
&\leq \|\Pi_{\mathcal{S}} \circ \mathbf{cV}_{\widehat{y}} \circ (\mathbf{I}_n - \Pi_{\mathcal{S}})\|_F \\
&= \sqrt{\sum_{w,z \in \overline{\mathcal{Y}}} \left|\langle w \,|\, \Pi_{\mathcal{S}} \circ \mathbf{cV}_{\widehat{y}} \circ (\mathbf{I}_n - \Pi_{\mathcal{S}}) \,|\, z\rangle\right|^2} \\
&= \sqrt{\sum_{w \in \mathcal{S}, z \notin \mathcal{S}} \left|\langle w \,|\, \mathbf{cV}_{\widehat{y}} \,|\, z\rangle\right|^2} \\
&\leq \sqrt{\sum_{w \in \mathcal{S}, z \in \mathcal{Z}, z \neq w} \left|\langle w \,|\, \mathbf{cV}_{\widehat{y}} \,|\, z\rangle\right|^2} \leq \sqrt{\frac{10|\mathcal{S}|}{2^n}},
\end{aligned}$$

where the first inequality follows from Lemma A.2.1 and the second inequality from Lemma 7.4.3 and the fact that $\mathcal{S} \subseteq \mathcal{Y}$. In conclusion, one has

$$\begin{aligned}
[\![\mathcal{P}^c \hookrightarrow \mathcal{P}']\!] &= \max_{x \in \widetilde{\mathcal{X}}, \widehat{y} \in \widehat{\mathcal{Y}}, d \in \mathcal{D}|_{\widetilde{\mathcal{X}}}} \|\Pi_{\mathcal{P}' \cap d|^x} \circ \mathbf{cO}_{x\widehat{y}} \circ \Pi_{\mathcal{P}^c \cap d|^x}\| \\
&\leq \max_{x \in \widetilde{\mathcal{X}}, d \in \mathcal{D}|_{\widetilde{\mathcal{X}}}} \sqrt{\frac{10|\mathcal{S}_{x,d}^{\mathcal{P}^c \hookrightarrow \mathcal{P}'}|}{2^n}}.
\end{aligned}$$

This completes our proof. □

7.4.2 The Two-Domain Distance Lemma

Size-restricted Properties. For a domain-restricted database-set $\mathcal{D}|_{\widetilde{\mathcal{X}}}$, a property $\mathcal{P} \subseteq \mathcal{D}|_{\widetilde{\mathcal{X}}}$, and some $i \leq |\widetilde{\mathcal{X}}|$, we define

$$\mathcal{P}_{[\leq i]} := \{d \in \mathcal{P} \mid |d| \leq i\}.$$

Then the transition capacity $[\![\mathcal{P}^c_{[\leq i-1]} \hookrightarrow \mathcal{P}_{[\leq i]}]\!]$ is a measure of the maximum probability of a database outside $\mathcal{P}$ with at most $i-1$ entries changing to a database in $\mathcal{P}$ after a single application $\mathbf{cO}_{x\widehat{y}}$. (Note that $\mathcal{P}^c_{[\leq i-1]}$ denotes the size-restriction of $\mathcal{P}^c$, and not the complement of $\mathcal{P}_{[\leq i-1]}$.)

Let $\perp := \{d_\perp\}$ denote the *empty* property (where $d_\perp$ is the empty database, i.e., the constant-$\perp$ function). Then for $\mathcal{P}$ such that $d_\perp \notin \mathcal{P}$, $\perp = \mathcal{P}^c_{[\leq 0]}$. We define

$$\left(\perp \overset{q}{\rightsquigarrow} \mathcal{P}\right) := \sum_{i=1}^{q} [\![\mathcal{P}^c_{[\leq i-1]} \hookrightarrow \mathcal{P}_{[\leq i]}]\!],$$

the *q-query transition bound* from $\perp$ to $\mathcal{P}$. In other words, $\left(\perp \overset{q}{\rightsquigarrow} \mathcal{P}\right)$ is a measure of the probability that the empty database changes into a database in $\mathcal{P}$ *at any point* during q successive queries.

Prefixed Oracle. Fix some $t < m$ and write $\mathcal{X} = \mathcal{T} \times \mathcal{I}$, where $\mathcal{T} = \{0,1\}^{\lceil \log_2 t \rceil}$ and $\mathcal{I} = \{0,1\}^{m-\lceil \log_2 t \rceil}$. Any family of functions $\mathbf{p} = (\mathbf{p}_k : \mathcal{I} \to \mathcal{X})_{k\in[t]}$ is said to be a (t,m)-*domain-separator* if for all $(k,x) \in [1;t] \times \mathcal{I}$, $\mathbf{p}_k(x) \in \mathcal{I}_k$, where $\mathcal{I}_k = \{[k]_2 \parallel x : x \in \mathcal{I}\}$. We write $\mathbf{p}_k(\mathcal{I})$ and $\mathbf{p}(\mathcal{I})$ to denote $\{\mathbf{p}_k(x) : x \in \mathcal{I}\}$ and $\cup_{k\in[t]}\mathbf{p}_k(\mathcal{I})$, respectively.

For any (t,m)-domain-separator $\mathbf{p}$, the *prefixed-compressed oracle* $\mathbf{cO}^{\mathbf{p}}$ is defined as a family of oracles $\{\mathbf{cO}^{\mathbf{p}_k}\}_{k\in[t]}$, where $\mathbf{cO}^{\mathbf{p}_k}$ denotes the restriction of $\mathbf{cO}$ to inputs from $\widetilde{\mathcal{X}} := \mathbf{p}(\mathcal{I}) \subset \mathcal{X}$, i.e., for some $k \in [1;t]$, $x \in \mathcal{I}$, $\widehat{y} \in \widehat{\mathcal{Y}}$ and $\widehat{d} \in \widehat{\mathcal{D}}$, we have

$$\mathbf{cO}^{\mathbf{p}_k}|x\rangle|\widehat{y}\rangle \otimes |\widehat{d}\rangle = |\mathbf{p}_k(x)\rangle|\widehat{y}\rangle \otimes \mathbf{cO}^{\mathbf{p}_k}_{x\widehat{y}}|\widehat{d}\rangle,$$

where $\mathbf{cO}^{\mathbf{p}_k}_{x\widehat{y}} := \mathbf{comp}_{\mathbf{p}_k(x)} \circ \mathbf{O}_{\mathbf{p}_k(x)\widehat{y}} \circ \mathbf{comp}_{\mathbf{p}_k(x)}$.

Two-Domain Systems. Let $\mathbf{I}$ and $\mathbf{R}$ be two stateful oracles with $\mathcal{H}_{in} = \mathbb{C}[\mathcal{I}]$, $\mathcal{H}_{out} = \mathbb{C}[\mathcal{Z}]$, $\mathcal{H}_{db} = \mathbb{D}$, defined by the sequences of unitaries:

$$\mathbf{I} := \mathbf{F}_t\mathbf{cO}^{\mathbf{I}_t} \dots \mathbf{cO}^{\mathbf{I}_1}\mathbf{F}_0, \qquad \mathbf{R} := \mathbf{F}_t\mathbf{cO}^{\mathbf{R}_t} \dots \mathbf{cO}^{\mathbf{R}_1}\mathbf{F}_0,$$

where with a slight abuse of notations we reuse $\mathbf{I}$ and $\mathbf{R}$ to also denote the corresponding (t,m)-domain-separators, and the unitaries $\mathbf{F}_0, \dots, \mathbf{F}_t$ only operate on the input, output and ancillary qubits, if any, needed to compute the function itself. However, we continue ignoring the ancillary qubits whenever convenient.

Consider a q-query interactive game where a computationally unbounded and deterministic distinguisher A aims to distinguish $\mathbf{R}$ from $\mathbf{I}$. We emphasize that in such an interactive game with $\mathbf{I}$ or $\mathbf{R}$, the compressed oracle $\mathbf{cO}$ is invoked a total of $q' := kq$ times. Fix two domains $\widetilde{\mathcal{X}}_{\mathbf{I}} = \mathbf{I}(\mathcal{I})$, $\widetilde{\mathcal{X}}_{\mathbf{R}} = \mathbf{R}(\mathcal{I})$, and define $\mathcal{D}_{\mathbf{I}} := \mathcal{D}|_{\widetilde{\mathcal{X}}_{\mathbf{I}}}$ and $\mathcal{D}_{\mathbf{R}} := \mathcal{D}|_{\widetilde{\mathcal{X}}_{\mathbf{R}}}$. Consider properties $\mathcal{B}_{\mathbf{I}} \subseteq \mathcal{D}_{\mathbf{I}} \setminus \perp$ and $\mathcal{B}_{\mathbf{R}} \subseteq \mathcal{D}_{\mathbf{R}} \setminus \perp$, and define $\mathcal{G}_{\mathbf{I}} := \mathcal{D}_{\mathbf{I}} \setminus \mathcal{B}_{\mathbf{I}}$ and $\mathcal{G}_{\mathbf{R}} := \mathcal{D}_{\mathbf{R}} \setminus \mathcal{B}_{\mathbf{R}}$. The central tool of our proof technique will be the following result named the Two-Domain Distance Lemma, largely adapted from [177, Proposition 3].

Lemma 7.4.4 (Two-Domain Distance Lemma)**.** *Suppose we can find a map $h : \mathcal{G}_{\mathbf{I}} \longrightarrow \mathcal{G}_{\mathbf{R}}$ such that the following hold:*

- *h is a bijection from $\mathcal{G}_{\mathbf{I}}$ to $\mathcal{G}_{\mathbf{R}}$ (and hence $|\mathcal{G}_{\mathbf{I}}| = |\mathcal{G}_{\mathbf{R}}|$);*
- *For every $i \in [0;q']$, $h|_{\mathcal{G}_{\mathbf{I}[\le i]}}$ is a bijection from $\mathcal{G}_{\mathbf{I}[\le i]}$ to $\mathcal{G}_{\mathbf{R}[\le i]}$ (and hence $|\mathcal{G}_{\mathbf{I}[\le i]}| = |\mathcal{G}_{\mathbf{R}[\le i]}|$);*
- *For every $i \in [1;q']$, $x \in \mathcal{I}$, $\widehat{y} \in \widehat{\mathcal{Y}}$, $d \in \mathcal{G}_{\mathbf{I}[\le i-1]}$, and $d' \in \mathcal{G}_{\mathbf{I}[\le i]}$,*

$$\langle d' \,|\, \mathbf{cO}^{\mathbf{I}_k}_{x\widehat{y}} \,|\, d\rangle = \langle h(d') \,|\, \mathbf{cO}^{\mathbf{R}_k}_{x\widehat{y}} \,|\, h(d)\rangle.$$

where

$$k = \begin{cases} t & \text{if } i = 0 \bmod t, \\ i \bmod t & \text{otherwise.} \end{cases}.$$

Then, for any computationally unbounded and deterministic distinguisher $\mathscr{A}$ we have

$$\|\mathsf{Tr}_{\mathbb{D}}(\rho^{q}_{\mathscr{A},\mathbf{I}}) - \mathsf{Tr}_{\mathbb{D}}(\rho^{q}_{\mathscr{A},\mathbf{R}})\|_1 \leq 3\big(\bot \overset{q'}{\rightsquigarrow} \mathcal{B}_{\mathbf{I}}\big)_{\mathbf{I}} + 3\big(\bot \overset{q'}{\rightsquigarrow} \mathcal{B}_{\mathbf{R}}\big)_{\mathbf{R}},$$

where $\rho^{q}_{\mathscr{A},\mathbf{p}} := \mathscr{A}^{\mathbf{p}}|\psi_{\mathscr{A}}, d_\bot\rangle\langle\psi_{\mathscr{A}}, d_\bot|\mathscr{A}^{\mathbf{p}\dagger}$ is the state after q queries to the oracle at-hand $\mathbf{p} \in \{\mathbf{I}, \mathbf{R}\}$ for some norm-1 vector $|\psi_{\mathscr{A}}\rangle$ and the empty database $|d_\bot\rangle$. The transition bounds $\big(\bot \overset{q'}{\rightsquigarrow} \cdot\big)_{\mathbf{I}}$ and $\big(\bot \overset{q'}{\rightsquigarrow} \cdot\big)_{\mathbf{R}}$ are computed for queries to $\mathbf{cO}^{\mathbf{I}}$ and $\mathbf{cO}^{\mathbf{R}}$, respectively.

When the oracle in use is clear from the context, we will drop the subscripts for the transition bounds and simply write both as $\big(\bot \overset{q}{\rightsquigarrow} \cdot\big)$. We'll also keep the input-domain separator maps implicit when there's no scope for ambiguity.

Setup for The Proof of Lemma 7.4.4. Let $\mathbf{U}_0, \ldots, \mathbf{U}_q$ denote $\mathscr{A}$'s unitaries. Define:

- $\mathbf{V}_0 := \mathbf{F}_0 \circ \mathbf{U}_0$,
- $\mathbf{V}_{it+j} := \mathbf{F}_j$, for all $i \in [0; q'-1]$, $j \in [1; t-1]$,
- $\mathbf{V}_{it} := \mathbf{U}_0 \circ \mathbf{U}_i \circ \mathbf{F}_t$, for all $i \in [1; q-1]$,
- $\mathbf{V}_{qt} := \mathbf{U}_q \circ \mathbf{F}_t$.

This defines a sequence of $q'+1$ unitaries, $\mathbf{V}_0, \ldots, \mathbf{V}_{q'}$, where $q' = qt$. For each $i \in [1; q']$, $\mathbf{p} \in \{\mathbf{I}, \mathbf{R}\}$, define $\mathbf{W}_{i,\mathbf{p}} := \mathbf{cO}^{\mathbf{p}_{i_t}} \circ \mathbf{V}_{i-1}$, where

$$i_t := \begin{cases} t & \text{if } i = 0 \bmod t, \\ i \bmod t & \text{otherwise.} \end{cases}$$

Let $|\psi_\bot\rangle = |\psi_{\mathscr{A}}\rangle \otimes |d_\bot\rangle$. Then, for all $\mathbf{p} \in \{\mathbf{I}, \mathbf{R}\}$, we can write $\rho^{q}_{A,\mathbf{p}} = |\psi_{q',\mathbf{p}}\rangle\langle\psi_{q',\mathbf{p}}|$, where

$$|\psi_{q',\mathbf{p}}\rangle = \mathbf{V}_{q'} \circ \mathbf{W}_{q',\mathbf{p}} \circ \mathbf{W}_{q'-1,\mathbf{p}} \circ \ldots \circ \mathbf{W}_{1,\mathbf{p}}|\psi_\bot\rangle.$$

Let $\mathbf{W}^{b}_{i,\mathbf{p}} := \Pi_{\mathcal{B}_{\mathbf{p}_{\leq i}}} \circ \mathbf{W}_{i,\mathbf{p}}$ and $\mathbf{W}^{g}_{i,\mathbf{p}} := \Pi_{\mathcal{G}_{\mathbf{p}_{\leq i}}} \circ \mathbf{W}_{i,\mathbf{p}}$. Then we have $\mathbf{W}_{i,\mathbf{p}} = \mathbf{W}^{b}_{i,\mathbf{p}} + \mathbf{W}^{g}_{i,\mathbf{p}}$. Further, let $|\psi_{i,\mathbf{p}}\rangle := \mathbf{W}_{i,\mathbf{p}} \circ \ldots \circ \mathbf{W}_{1,\mathbf{p}}|\psi_\bot\rangle$, and $|\psi^{g}_{i,\mathbf{p}}\rangle := \mathbf{W}^{g}_{i,\mathbf{p}} \circ \ldots \circ \mathbf{W}^{g}_{1,\mathbf{p}}|\psi_\bot\rangle$.

Proposition 7.4.1. *For every $i \in [1; q']$ and each $\mathbf{p} \in \{\mathbf{I}, \mathbf{R}\}$:*

$$\||\psi_{i,\mathbf{p}}\rangle - |\psi^{g}_{i,\mathbf{p}}\rangle\| \leq \big(\bot \overset{i}{\rightsquigarrow} \mathcal{B}_{\mathbf{p}}\big)_{\mathbf{p}}. \tag{7.38}$$

Proof of Proposition 7.4.1. We show this by induction on $i \in [1; q']$. Fix some $\mathbf{p} \in \{\mathbf{I}, \mathbf{R}\}$. First consider $i = 1$, we have

$$\||\psi_{1,\mathbf{p}}\rangle - |\psi^{g}_{1,\mathbf{p}}\rangle\| = \|\mathbf{W}_{1,\mathbf{p}}|\psi_\bot\rangle - \mathbf{W}^{g}_{1,\mathbf{p}}|\psi_\bot\rangle\| = \|\mathbf{W}^{b}_{1,\mathbf{p}}|\psi_\bot\rangle\|.$$

Since $d_\bot \in \mathcal{G}_{\mathbf{p}}$ and $\mathbf{V}_0$ commutes with $\Pi_{\mathcal{G}_{\mathbf{p}_{\leq 0}}}$, we have

$$\begin{aligned} \|\mathbf{W}^{b}_{1,\mathbf{p}}|\psi_\bot\rangle\| &= \|\Pi_{\mathcal{B}_{\mathbf{p}_{\leq 1}}} \circ \mathbf{W}_{1,\mathbf{p}} \circ \Pi_{\mathcal{G}_{\mathbf{p}_{\leq 0}}}|\psi_\bot\rangle\| \\ &= \|\Pi_{\mathcal{B}_{\mathbf{p}_{\leq 1}}} \circ \mathbf{cO}^{\mathbf{p}_1} \circ \mathbf{V}_0 \circ \Pi_{\mathcal{G}_{\mathbf{p}_{\leq 0}}}|\psi_\bot\rangle\| \\ &= \|\Pi_{\mathcal{B}_{\mathbf{p}_{\leq 1}}} \circ \mathbf{cO}^{\mathbf{p}_1} \circ \Pi_{\mathcal{G}_{\mathbf{p}_{\leq 0}}} \circ \mathbf{V}_0|\psi_\bot\rangle\| \end{aligned}$$

$$\leq \|\Pi_{\mathcal{B}_{\mathbf{p}_{\leq 1}}} \circ \mathbf{cO}^{\mathbf{p}_1} \circ \Pi_{\mathcal{G}_{\mathbf{p}_{\leq 0}}}\|$$
$$\leq [\![\mathcal{G}_{\mathbf{p}_{\leq 0}} \hookrightarrow \mathcal{B}_{\mathbf{p}_{\leq 1}}]\!]_{\mathbf{p}} = \left(\bot \overset{1}{\rightsquigarrow} \mathcal{B}_{\mathbf{p}}\right)_{\mathbf{p}},$$

where the last inequality follows from Lemma 7.4.1. This proves the $i = 1$ case. Suppose now $i \geq 2$:

$$\||\psi_{i-1,\mathbf{p}}\rangle - |\psi^{g}_{i-1,\mathbf{p}}\rangle\| \leq \left(\bot \overset{i-1}{\rightsquigarrow} \mathcal{B}_{\mathbf{p}}\right)_{\mathbf{p}}.$$

Then we have

$$\begin{aligned}
\||\psi_{i,\mathbf{p}}\rangle - |\psi^{g}_{i,\mathbf{p}}\rangle\| &= \|\mathbf{W}_{i,\mathbf{p}}|\psi_{i-1,\mathbf{p}}\rangle - \mathbf{W}^{g}_{i,\mathbf{p}}|\psi^{g}_{i-1,\mathbf{p}}\rangle\| \\
&= \|\mathbf{W}_{i,\mathbf{p}}|\psi_{i-1,\mathbf{p}}\rangle - \mathbf{W}_{i,\mathbf{p}}|\psi^{g}_{i-1,\mathbf{p}}\rangle + \mathbf{W}_{i,\mathbf{p}}|\psi^{g}_{i-1,\mathbf{p}}\rangle - \mathbf{W}^{g}_{i,\mathbf{p}}|\psi^{g}_{i-1,\mathbf{p}}\rangle\| \\
&= \|\mathbf{W}_{i,\mathbf{p}}(|\psi_{i-1,\mathbf{p}}\rangle - |\psi^{g}_{i-1,\mathbf{p}}\rangle) + (\mathbf{W}_{i,\mathbf{p}} - \mathbf{W}^{g}_{i,\mathbf{p}})|\psi^{g}_{i-1,\mathbf{p}}\rangle\| \\
&\leq \|\mathbf{W}_{i,\mathbf{p}}(|\psi_{i-1,\mathbf{p}}\rangle - |\psi^{g}_{i-1,\mathbf{p}}\rangle)\| + \|\mathbf{W}^{b}_{i,\mathbf{p}}|\psi^{g}_{i-1,\mathbf{p}}\rangle\| \\
&\leq \||\psi_{i-1,\mathbf{p}}\rangle - |\psi^{g}_{i-1,\mathbf{p}}\rangle\| + \|\Pi_{\mathcal{B}_{\mathbf{p}_{\leq i}}} \circ \mathbf{W}_{i,\mathbf{p}}|\psi^{g}_{i-1,\mathbf{p}}\rangle\|.
\end{aligned}$$

Since $|\psi^{g}_{i-1,\mathbf{p}}\rangle$ is in the column space of $\Pi_{\mathcal{G}_{\mathbf{p}_{\leq i-1}}}$. Thus, by reasoning as in the $i = 1$ case, we have

$$\begin{aligned}
\|\Pi_{\mathcal{B}_{\mathbf{p}_{\leq i}}} \circ \mathbf{W}_{i,\mathbf{p}}|\psi^{g}_{i-1,\mathbf{p}}\rangle\| &\leq \|\Pi_{\mathcal{B}_{\mathbf{p}_{\leq i}}} \circ \mathbf{cO}^{\mathbf{p}_{i_t}} \circ \Pi_{\mathcal{G}_{\mathbf{p}_{\leq i-1}}}\| \\
&\leq [\![\mathcal{G}_{\mathbf{p}_{\leq i-1}} \hookrightarrow \mathcal{B}_{\mathbf{p}_{\leq i}}]\!]_{\mathbf{p}}.
\end{aligned}$$

Using induction on $i \in [1; q']$ we get

$$\begin{aligned}
\||\psi_{i,\mathbf{p}}\rangle - |\psi^{g}_{i,\mathbf{p}}\rangle\| &\leq \||\psi_{i-1,\mathbf{p}}\rangle - |\psi^{g}_{i-1,\mathbf{p}}\rangle\| + \|\Pi_{\mathcal{B}_{\mathbf{p}_{\leq i}}} \circ \mathbf{W}_{i,\mathbf{p}}|\psi^{g}_{i-1,\mathbf{p}}\rangle\| \\
&\leq \left(\bot \overset{i-1}{\rightsquigarrow} \mathcal{B}_{\mathbf{p}}\right)_{\mathbf{p}} + [\![\mathcal{G}_{\mathbf{p}_{\leq i-1}} \hookrightarrow \mathcal{B}_{\mathbf{p}_{\leq i}}]\!]_{\mathbf{p}} = \left(\bot \overset{i}{\rightsquigarrow} \mathcal{B}_{\mathbf{p}}\right)_{\mathbf{p}},
\end{aligned}$$

thus completing the proof.

□

The next step in our proof will be to show that the partial trace on the good databases remains equal across all queries, whether in the ideal world or the real world. For that we will need the following intermediate result.

Proposition 7.4.2. *For any $x \in \mathcal{I}$, $\widehat{y} \in \widehat{\mathcal{Y}}$, any $i \in [1; q']$, and any $d \in \mathcal{G}_{\mathbf{I}_{\leq i}}$,*

$$\langle x, \widehat{y}, d|\psi^{g}_{i,\mathbf{I}}\rangle = \langle x, \widehat{y}, h(d)|\psi^{g}_{i,\mathbf{R}}\rangle. \tag{7.39}$$

Proof of Proposition 7.4.2. For the case of $i = 1$, considering some $d \in \mathcal{G}_{\mathbf{I}_{\leq 1}}$, we have

$$|\psi^{g}_{1,\mathbf{I}}\rangle = \mathbf{W}^{g}_{1,\mathbf{I}}|\psi_{\perp}\rangle = \Pi_{\mathcal{G}_{\mathbf{I}_{\leq 1}}} \circ \mathbf{cO}^{\mathbf{I}_1} \circ \mathbf{V}_0|\psi_{\perp}\rangle.$$

Let $|\gamma_{x,\widehat{y}}\rangle$ denote the basis state $|x\rangle|\widehat{y}\rangle$. Then we have

$$\begin{aligned}
\mathbf{cO}^{\mathbf{I}_1} \circ \mathbf{V}_0|\psi_{\perp}\rangle &= \mathbf{cO}^{\mathbf{I}_1} \circ \mathbf{V}_0|\psi_{\mathscr{A}}\rangle \otimes |d_{\perp}\rangle \\
&= \sum_{x,\widehat{y}} \langle \gamma_{x,\widehat{y}}| \mathbf{V}_0 |\psi_{\mathscr{A}}\rangle \, \mathbf{cO}^{\mathbf{I}_1}|\gamma_{x,\widehat{y}}\rangle \otimes |d_{\perp}\rangle
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{x,\widehat{y}} \langle \gamma_{x,\widehat{y}} | \, \mathbf{V}_0 \, | \, \psi_{\mathscr{A}} \rangle \left(|\gamma_{x,\widehat{y}}\rangle \otimes \mathbf{cO}^{\mathbf{I}_1}_{x\widehat{y}} |d_\perp\rangle \right) \\
&= \sum_{\substack{x,\widehat{y} \\ d \in \mathcal{D}_{\mathbf{I}}}} \langle \gamma_{x,\widehat{y}} | \, \mathbf{V}_0 \, | \, \psi_{\mathscr{A}} \rangle \, \langle d \, | \, \mathbf{cO}^{\mathbf{I}_1}_{x\widehat{y}} \, | \, d_\perp \rangle \, |\gamma_{x,\widehat{y}}\rangle \otimes |d\rangle \\
&= \sum_{\substack{x,\widehat{y} \\ d \in \mathcal{D}_{\mathbf{I}}}} \langle \gamma_{x,\widehat{y}} | \, \mathbf{V}_0 \, | \, \psi_{\mathscr{A}} \rangle \, \langle d \, | \, \mathbf{cO}^{\mathbf{I}_1}_{x\widehat{y}} \, | \, d_\perp \rangle \, |\gamma_{x,\widehat{y}}\rangle \otimes |d\rangle,
\end{aligned}
$$

where $x \in \mathcal{I}$, and $\widehat{y} \in \widehat{\mathcal{Y}}$ in all the sums. Thus,

$$
\Pi_{\mathcal{G}_{\mathbf{I}_{\leq 1}}} \circ \mathbf{cO}^{\mathbf{I}_1} \circ \mathbf{V}_0 |\psi_\perp\rangle = \sum_{\substack{x,\widehat{y} \\ d \in \mathcal{G}_{\mathbf{I}_{\leq 1}}}} \langle \gamma_{x,\widehat{y}} | \, \mathbf{V}_0 \, | \, \psi_{\mathscr{A}} \rangle \, \langle d \, | \, \mathbf{cO}^{\mathbf{I}_1}_{x\widehat{y}} \, | \, d_\perp \rangle \, |\varphi_{x,\widehat{y},d}\rangle,
$$

where $\varphi_{x,\widehat{y},d}$ denotes the basis state $|x, \widehat{y}, d\rangle$. This gives, for any $x \in \mathcal{I}$, $\widehat{y} \in \widehat{\mathcal{Y}}$, and $d \in \mathcal{G}_{\mathbf{I}_{\leq 1}}$,

$$
\langle \varphi_{x,\widehat{y},d} | \psi^g_{1,\mathbf{I}} \rangle = \langle \gamma_{x,\widehat{y}} | \, \mathbf{V}_0 \, | \, \psi_{\mathscr{A}} \rangle \, \langle d \, | \, \mathbf{cO}^{\mathbf{I}_1}_{x\widehat{y}} \, | \, d_\perp \rangle.
$$

Similarly, we can show that

$$
\langle \varphi_{x,\widehat{y},h(d)} | \psi^g_{1,\mathbf{R}} \rangle = \langle \gamma_{x,\widehat{y}} | \, \mathbf{V}_0 \, | \, \psi_{\mathscr{A}} \rangle \, \langle h(d) \, | \, \mathbf{cO}^{\mathbf{R}_1}_{x\widehat{y}} \, | \, d_\perp \rangle.
$$

Since $\mathcal{G}_{\mathbf{I}_{\leq 0}} = \mathcal{G}_{\mathbf{R}_{\leq 0}} = \{d_\perp\}$, we have $d_\perp = h(d_\perp)$, and the third condition Lemma 7.4.4 gives us $\langle \varphi_{x,\widehat{y},d} | \psi^g_{1,\mathbf{I}} \rangle = \langle \varphi_{x,\widehat{y},h(d)} | \psi^g_{1,\mathbf{R}} \rangle$, thus establishing the $i = 1$ case. For some $i \geq 2$, for all $x, \in \mathcal{I}$, $\widehat{y} \in \widehat{\mathcal{Y}}$, and $d \in \mathcal{G}_{\mathbf{I}_{\leq i-1}}$, suppose

$$
\alpha_{x,\widehat{y},d} = \langle \varphi_{x,\widehat{y},d} | \psi^g_{i-1,\mathbf{I}} \rangle = \langle \varphi_{x,\widehat{y},h(d)} | \psi^g_{i-1,\mathbf{R}} \rangle.
$$

Then, since $h|_{\mathcal{G}_{\mathbf{I}_{\leq i-1}}}$ is bijective, we have

$$
\begin{aligned}
|\psi^g_{i-1,\mathbf{I}}\rangle &= \sum_{\substack{x,\widehat{y} \\ d \in \mathcal{G}_{\mathbf{I}_{\leq i-1}}}} \alpha_{x,\widehat{y},d} |\gamma_{x,\widehat{y}}\rangle \otimes |d\rangle, \\
|\psi^g_{i-1,\mathbf{R}}\rangle &= \sum_{\substack{x,\widehat{y} \\ d \in \mathcal{G}_{\mathbf{I}_{\leq i-1}}}} \alpha_{x,\widehat{y},d} |\gamma_{x,\widehat{y}}\rangle \otimes |h(d)\rangle.
\end{aligned}
$$

This gives

$$
\begin{aligned}
|\psi^g_{i,\mathbf{I}}\rangle &= \mathbf{W}^g_{i,\mathbf{I}} |\psi^g_{i-1,\mathbf{I}}\rangle \\
&= \Pi_{\mathcal{G}_{\mathbf{I}_{\leq i}}} \circ \mathbf{cO}^{\mathbf{I}_{i_t}} \circ \mathbf{V}_{i-1} |\psi^g_{i-1,\mathbf{I}}\rangle \\
&= \sum_{\substack{x,\widehat{y} \\ d \in \mathcal{G}_{\mathbf{I}_{\leq i-1}}}} \alpha_{x,\widehat{y},d} \, \Pi_{\mathcal{G}_{\mathbf{I}_{\leq i}}} \circ \mathbf{cO}^{\mathbf{I}_{i_t}} \circ \mathbf{V}_{i-1} |\gamma_{x,\widehat{y}}\rangle \otimes |d\rangle \\
&= \sum_{\substack{x,x' \\ \widehat{y},\widehat{y}' \\ d \in \mathcal{G}_{\mathbf{I}_{\leq i-1}}}} \alpha_{x,\widehat{y},d} \, \langle \gamma_{x',\widehat{y}'} | \, \mathbf{V}_{i-1} \, | \, \gamma_{x,\widehat{y}} \rangle \, \Pi_{\mathcal{G}_{\mathbf{I}_{\leq i}}} \circ \mathbf{cO}^{\mathbf{I}_{i_t}} |\gamma_{x',\widehat{y}'}\rangle \otimes |d\rangle
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{\substack{x,x'\\ \widehat{y},\widehat{y}'\\ d\in\mathcal{G}_{\mathbf{I}_{\leq i-1}}}} \alpha_{x,\widehat{y},d}\, \langle \gamma_{x',\widehat{y}'} \,|\, \mathbf{V}_{i-1} \,|\, \gamma_{x,\widehat{y}} \rangle\, \Pi_{\mathcal{G}_{\mathbf{I}_{\leq i}}} \left(|\gamma_{x',\widehat{y}'}\rangle \otimes \mathbf{cO}^{\mathbf{I}_{i_t}}_{x'\widehat{y}'} |d\rangle \right) \\
&= \sum_{\substack{x,x'\\ \widehat{y},\widehat{y}'\\ d\in\mathcal{G}_{\mathbf{I}_{\leq i-1}}\\ d'\in\mathcal{D}_{\mathbf{I}}}} \alpha_{x,\widehat{y},d}\, \langle \gamma_{x',\widehat{y}'} \,|\, \mathbf{V}_{i-1} \,|\, \gamma_{x,\widehat{y}} \rangle\, \langle d' \,|\, \mathbf{cO}^{\mathbf{I}_{i_t}}_{x'\widehat{y}'} \,|\, d \rangle\, \Pi_{\mathcal{G}_{\mathbf{I}_{\leq i}}} \left(|\gamma_{x',\widehat{y}'}\rangle \otimes |d'\rangle \right) \\
&= \sum_{\substack{x,x'\\ \widehat{y},\widehat{y}'\\ d\in\mathcal{G}_{\mathbf{I}_{\leq i-1}}\\ d'\in\mathcal{G}_{\mathbf{I}_{\leq i}}}} \alpha_{x,\widehat{y},d}\, \langle \gamma_{x',\widehat{y}'} \,|\, \mathbf{V}_{i-1} \,|\, \gamma_{x,\widehat{y}} \rangle\, \langle d' \,|\, \mathbf{cO}^{\mathbf{I}_{i_t}}_{x'\widehat{y}'} \,|\, d \rangle\, |\varphi_{x',\widehat{y}',d'}\rangle,
\end{aligned}
$$

so that for any $x' \in \mathcal{I}$, $\widehat{y}' \in \widehat{\mathcal{Y}}$, and $d' \in \mathcal{G}_{\mathbf{I}_{\leq i}}$, we have

$$
\langle \varphi_{x',\widehat{y}',d'} | \psi^{g}_{i,\mathbf{I}} \rangle = \sum_{\substack{x,\widehat{y}\\ d\in\mathcal{G}_{\mathbf{I}_{\leq i-1}}}} \alpha_{x,\widehat{y},d}\, \langle \gamma_{x',\widehat{y}'} \,|\, \mathbf{V}_{i-1} \,|\, \gamma_{x,\widehat{y}} \rangle\, \langle d' \,|\, \mathbf{cO}^{\mathbf{I}_{i_t}}_{x\widehat{y}} \,|\, d \rangle.
$$

Similarly, we can show that

$$
\langle \varphi_{x',\widehat{y}',h(d')} | \psi^{g}_{i,\mathbf{R}} \rangle = \sum_{\substack{x,\widehat{y}\\ d\in\mathcal{G}_{\mathbf{I}_{\leq i-1}}}} \alpha_{x,\widehat{y},h(d)}\, \langle \gamma_{x',\widehat{y}'} \,|\, \mathbf{V}_{i-1} \,|\, \gamma_{x,\widehat{y}} \rangle\, \langle h(d') \,|\, \mathbf{cO}^{\mathbf{R}_{i_t}}_{x\widehat{y}} \,|\, h(d) \rangle.
$$

Then the third condition of Lemma 7.4.4 concludes this proof. □

Proposition 7.4.2 gives the following Corollary.

Corollary 7.4.1. *For any $i \in [1; q']$,*

$$
\mathsf{Tr}_{\mathbb{D}} \left(|\psi^{g}_{i,\mathbf{I}}\rangle\langle\psi^{g}_{i,\mathbf{I}}| \right) = \mathsf{Tr}_{\mathbb{D}} \left(|\psi^{g}_{i,\mathbf{R}}\rangle\langle\psi^{g}_{i,\mathbf{R}}| \right).
$$

Proof of Corollary 7.4.1.

$$
\begin{aligned}
\mathsf{Tr}_{\mathbb{D}} \left(|\psi^{g}_{i,\mathbf{I}}\rangle\langle\psi^{g}_{i,\mathbf{I}}| \right) &= \sum_{d\in\mathcal{D}} \langle d \,|\, \psi^{g}_{i,\mathbf{I}} \rangle\langle\, \psi^{g}_{i,\mathbf{I}} \,|\, d \rangle \\
&= \sum_{d\in\mathcal{G}_{\mathbf{I}_{\leq i}}} \sum_{\substack{x,x'\\ \widehat{y},\widehat{y}'}} \alpha_{x,\widehat{y},d}\alpha_{x',\widehat{y}',d} |x,\widehat{y}\rangle\langle x',\widehat{y}'| \\
&= \sum_{\substack{x,x'\\ \widehat{y},\widehat{y}'}} \left(\sum_{d\in\mathcal{G}_{\mathbf{I}_{\leq i}}} \alpha_{x,\widehat{y},d}\alpha_{x',\widehat{y}',d} \right) |x,\widehat{y}\rangle\langle x',\widehat{y}'| \\
&= \sum_{\substack{x,x'\\ \widehat{y},\widehat{y}'}} \left(\sum_{d\in\mathcal{G}_{\mathbf{I}_{\leq i}}} \alpha_{x,\widehat{y},h(d)}\alpha_{x',\widehat{y}',h(d)} \right) |x,\widehat{y}\rangle\langle x',\widehat{y}'| \\
&= \sum_{\substack{x,x'\\ \widehat{y},\widehat{y}'}} \left(\sum_{h(d)\in\mathcal{G}_{\mathbf{R}_{\leq i}}} \alpha_{x,\widehat{y},h(d)}\alpha_{x',\widehat{y}',h(d)} \right) |x,\widehat{y}\rangle\langle x',\widehat{y}'|
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{\substack{x,x' \\ \widehat{y},\widehat{y}'}} \left(\sum_{d' \in \mathcal{G}_{\mathbf{R}_{\leq i}}} \alpha_{x,\widehat{y},d'} \alpha_{x',\widehat{y}',d'} \right) |x,\widehat{y}\rangle\langle x',\widehat{y}'| \\
&= \sum_{d' \in \mathcal{G}_{\mathbf{R}_{\leq i}}} \sum_{\substack{x,x' \\ \widehat{y},\widehat{y}'}} \alpha_{x,\widehat{y},d'} \alpha_{x',\widehat{y}',d'} |x,\widehat{y}\rangle\langle x',\widehat{y}'| \\
&= \sum_{d' \in \mathcal{D}} \langle d' \,|\, \psi^{g}_{i,\mathbf{R}} \rangle\langle\, \psi^{g}_{i,\mathbf{R}} \,|\, d' \rangle \\
&= \mathsf{Tr}_{\mathbb{D}} \left(|\psi^{g}_{i,\mathbf{R}}\rangle\langle\psi^{g}_{i,\mathbf{R}}| \right). \qquad (7.40)
\end{aligned}
$$

□

With the setup complete, we now proceed to the actual proof of Lemma 7.4.4.

Proof of Lemma 7.4.4. For each $\mathbf{p} \in \{\mathbf{I}, \mathbf{R}\}$, let $|\psi^{b}_{q',\mathbf{p}}\rangle := |\psi_{q',\mathbf{p}}\rangle - |\psi^{g}_{q',\mathbf{p}}\rangle$. Then, we have

$$
\begin{aligned}
\|\mathsf{Tr}_{\mathbb{D}}(\rho^{q}_{A,\mathbf{I}}) - \mathsf{Tr}_{\mathbb{D}}(\rho^{q}_{A,\mathbf{R}})\|_1 &= \|\mathsf{Tr}_{\mathbb{D}}(|\psi_{q',\mathbf{I}}\rangle\langle\psi_{q',\mathbf{I}}|) - \mathsf{Tr}_{\mathbb{D}}(|\psi_{q',\mathbf{R}}\rangle\langle\psi_{q',\mathbf{R}}|)\|_1 \\
&= \sum_{\mathbf{p}\in\{\mathbf{I},\mathbf{R}\},w,w'\in\{b,g\}} \|\mathsf{Tr}_{\mathbb{D}}(|\psi^{w}_{q',\mathbf{p}}\rangle\langle\psi^{w'}_{q',\mathbf{p}}|)\|_1 \qquad (7.41) \\
&\leq \sum_{\mathbf{p}\in\{\mathbf{I},\mathbf{R}\},w,w'\in\{b,g\}} \||\psi^{w}_{q',\mathbf{p}}\rangle\langle\psi^{w'}_{q',\mathbf{p}}|\|_1 \qquad (7.42) \\
&\leq 3\||\psi^{b}_{q',\mathbf{I}}\rangle\| + 3\||\psi^{b}_{q',\mathbf{R}}\rangle\| \qquad (7.43) \\
&\leq 3\big(\perp \overset{q'}{\rightsquigarrow} \mathcal{B}_{\mathbf{I}}\big)_{\mathbf{I}} + 3\big(\perp \overset{q'}{\rightsquigarrow} \mathcal{B}_{\mathbf{R}}\big)_{\mathbf{R}}, \qquad (7.44)
\end{aligned}
$$

where

- (7.41) follows from the linearity of the partial trace map, Corollary 7.4.1, and the triangle inequality.
- (7.42) follows from the fact that partial trace is a completely positive and trace-preserving map;
- (7.43) follows from repeated applications of Lemma A.4.1; and
- (7.44) follows from Lemma 7.3.2.

This completes the proof of The Two-Domain Distance Lemma. □

7.5 Blueprint for Post-Quantum PRF Security Proofs

Our main objective in this chapter is to leverage the Two-Distance Lemma developed in Section 7.3, to show the post-quantum PRF security of the selected constructions `TNT` and `LRWQ` . To achieve this goal we start by describing a blueprint that illustrates how to use our framework to reduce the post-quantum PRF security of a compressing function $g : \{0,1\}^{tn} \to \{0,1\}^{n}$ to its component functions $f_0, \ldots, f_t \in \mathsf{Func}(n;n)$, while employing an "identical up to bad" classical reasoning. While this blueprint does not include proofs for every compression function g that is built upon $t+1$ component functions, it encapsulates the fundamental capabilities of our framework.

Notional Setup. Let $\mathscr{A}$ be a distinguisher aiming to distinguish between a construction $g_{\mathbf{R}} : \{0,1\}^{tn} \to \{0,1\}^n$ that is built on top of component functions $f_0, \ldots, f_t \in \mathsf{Func}(n;n)$ (the *real world*) and a random function $g_{\mathbf{I}} \twoheadleftarrow_{\$} \mathsf{Func}(tn;n)$ (the *ideal world*). Throughout this section, we assume $f_0, \ldots, f_t$ are ordered according to their composition in the implementation of $g_{\mathbf{R}}$, where f_0 is the first function applied to the input (or parts of it) and f_t is the last function applied. In a concrete implementation of $g_{\mathbf{R}}$, this order will be clear from the context. We employ the following proof strategy.

7.5.1 Modifying The Distinguishing Game

We slightly modify the distinguishing game by implementing the following changes. Let $\ell_t = \lceil \log(t+1) \rceil$, $t' = (t+1)n + \ell_t$ be some positive integers and define the spaces $\mathcal{X} = \{0,1\}^{t'}$ and $\mathcal{Y} = \{0,1\}^n$. We show that there exists a random function $f : \mathcal{X} \to \mathcal{Y}$ that satisfies two properties:

- for any $x \in \mathcal{Y}$ and $i \in [0;t]$ there exists $\mathbf{x} \in \{0,1\}^{tn}$ such that

$$f_i(x) := f(\langle i \rangle_{\ell_t} \| x \| \mathbf{x}) = f([x]_i)$$

  where $\langle i \rangle_{\ell_t}$ denotes the binary representation of i and $[x]_i$ is a notation for the corresponding input; and
- for any $\mathbf{x} \in \{0,1\}^{tn}$ there exists two functions $g, g' : \{0,1\}^{tn} \to \{0,1\}^n$ such that $g_{\mathbf{R}} = f_t(g + g')$ and we have that

$$g^*_{\mathbf{I}}(\mathbf{x}) := f(\langle t \rangle_{\ell_t} \| \mathbf{x} \| g(\mathbf{x})),$$

  where $g^*_{\mathbf{I}}$ is a random function, the functions g and g' are independent (probability wise), and the distribution of $g^*_{\mathbf{I}} + g'$ and $g_{\mathbf{I}}$ are identical.

The last condition implies that we can replace $g_{\mathbf{I}}$ with $g^*_{\mathbf{I}}$. This setup allows us to use a single database $d_f : \mathcal{X} \longrightarrow \mathcal{Z}$ to keep track of $f_0, \ldots, f_t$, and $g^*_{\mathbf{I}}$; we refer to this database as $d_{\mathbf{R}}$ in the real world (tracking $f_0, \ldots, f_t$) and $d_{\mathbf{I}}$ in the ideal world (tracking $f_0, \ldots, f_{t-1}$ and $g^*_{\mathbf{I}}$). Let $\mathcal{D}_{\mathbf{R}}$ (resp. $\mathcal{D}_{\mathbf{I}}$) be the set of all possible choices for $d_{\mathbf{R}}$ (resp. $d_{\mathbf{I}}$). We define the input spaces

$$\widetilde{\mathcal{X}}_{\mathbf{R}} := \{([x]_0, \ldots, [x]_t) \mid x \in \mathcal{Y}\},$$
$$\widetilde{\mathcal{X}}_{\mathbf{I}} := \{([x]_0, \ldots, [x]_{t-1}), \langle t \rangle_{\ell_t} \| \mathbf{x} \| g(\mathbf{x}) \mid x \in \mathcal{Y}, \mathbf{x} \in \{0,1\}^{tn}\}.$$

Then it is easy to see that $\mathcal{D}_{\mathbf{R}} = \mathcal{D}|_{\widetilde{\mathcal{X}}_{\mathbf{R}}}$ and $\mathcal{D}_{\mathbf{I}} = \mathcal{D}|_{\widetilde{\mathcal{X}}_{\mathbf{I}}}$. Moreover, one has

$$\mathbf{Adv}^{\mathrm{qcpa}}_{g_{\mathbf{R}}}(\mathscr{A}) \leq \| \mathsf{Tr}_{\mathbb{D}}(\rho^q_{\mathscr{A},\mathbf{I}}) - \mathsf{Tr}_{\mathbb{D}}(\rho^q_{\mathscr{A},\mathbf{R}}) \|_1,$$

where there are tq calls to f (and hence to $\mathbf{cO}$) during the distinguishing game. The notations are borrowed from Section 7.2.4.

7.5.2 Bad and Good Databases

The next step is the definition of the so called *bad databases*. In this step, we show that there exists sets $\mathcal{B}_{\mathbf{R}}$ and $\mathcal{B}_{\mathbf{I}}$ such that for $\mathcal{G}_{\mathbf{R}} := \mathcal{D}_{\mathbf{R}} \setminus \mathcal{B}_{\mathbf{R}}$ and $\mathcal{G}_{\mathbf{I}} := \mathcal{D}_{\mathbf{I}} \setminus \mathcal{B}_{\mathbf{I}}$ we can find a bijection $h : \mathcal{G}_{\mathbf{R}} \longrightarrow \mathcal{G}_{\mathbf{I}}$ such that:

- for any $i \in [0;t]$ and any input $u_i \in \mathcal{Y}$ to the function f_i we have, $d_{\mathbf{I}}([u_i]_i) = d_{\mathbf{R}}([u_i]_i)$;
- for any $\mathbf{x} \in \mathcal{X}$, the map $\mathbf{x} \mapsto g(\mathbf{x})$ is a bijection, thus satisfying $d_{\mathbf{I}}(\langle t \rangle_{\ell_t} \| \mathbf{x} \| g(\mathbf{x})) = d_{\mathbf{R}}([g(\mathbf{x})]_t)$.

Thus, all conditions for Lemma 7.4.4 are met.

7.5.3 Sequence of Actions

To complete the security analysis we show that there exists an upper bound $\varepsilon(q)$ such that

$$(t+1)\big(\perp \overset{(t+1)q}{\rightsquigarrow} \mathcal{B}_{\mathbf{I}}\big) + (t+1)\big(\perp \overset{(t+1)q}{\rightsquigarrow} \mathcal{B}_{\mathbf{R}}\big) \leq \varepsilon(q)$$

where $\varepsilon(q) := c(g_{\mathbf{R}}) \cdot \sqrt{\frac{10|\mathcal{S}|}{2^n}}$ for some constant $c(g_{\mathbf{R}})$ and $|\mathcal{S}|$ is easy to upper bound with simple counting arguments. The security bound $\varepsilon(q)$ will be derived by the following analysis.

Note that the prefixed oracle is defined so that each query made by the adversary triggers a sequence of $t+1$ queries to f, which we call *actions*. The main focus of the proof is to analyze the transition capacity at each action. Specifically, for any query $i \in [1;q']$, let $i' = i \mod (t+1)$. We then examine the transition capacity $[\![\mathcal{B}^c_{\mathbf{R}[\leq i'-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\leq i']}]\!]$ and use the size of $\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d}$ to establish an upper bound on this capacity. Summing all these derived upper bounds yields the value of $\varepsilon(q)$.

7.6 Post-Quantum PRF Security of TNT and LRWQ

In this section we analyze the post-quantum security of the three selected constructions from Section 7.1 using the blueprint described in Section 7.5.

7.6.1 Post-Quantum PRF Security of TNT

Here, we analyze the post-quantum security of TNT (see Figure 7.2), defined as

$$g^{\mathtt{TNT}}_{\mathbf{R}}(x_1, x_2) := f_3(f_2(f_1(x_1) \oplus x_2) \oplus x_2)$$

for three n-bit-to-n-bit random functions f_1, f_2, f_3. We want to bound the distinguishing advantage between $g^{\mathtt{TNT}}_{\mathbf{R}}$ and a $2n$-bit-to-n-bit random function $g_{\mathbf{I}}$. For that we will follow the blueprint above.

Theorem 7.6.1. *Let $\mathscr{A}$ be a (q,t)-quantum adversary, i.e., making at q queries and running in t time, distinguishing $g_{\mathbf{R}}^{TNT}$ from $g_{\mathbf{I}}$. Then there exists $(O(q),t_i)$-quantum distinguishers $\mathscr{B}_i$ against f_i, such that*

$$\mathbf{Adv}_{TNT}^{\text{qcpa}}(\mathscr{A}) \leq \sum_{i=1}^{3} \mathbf{Adv}_{f_i}^{\text{qcpa}}(\mathscr{B}_i) + 6\sqrt{\frac{10q^5}{2^n}},$$

where $\tau_i \in \widetilde{O}(t+q^2)$, for all $i \in [1;3]$.

We follow the blueprint in Section 7.5 step by step and write down the details specific to the TNT construction.

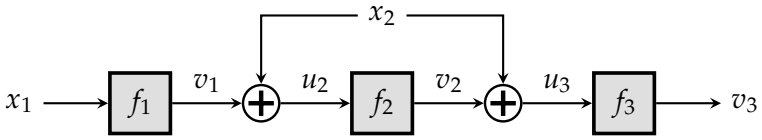

Figure 7.2: *The TNT construction by Bao et al. [17].*

7.6.1.1 Notional Setup

Let $\mathcal{X} := \{0,1\}^{3n+2}$, and let $f : \mathcal{X} \longrightarrow \mathcal{Y}$ be a $(3n+2)$-bit-to-n-bit random function, such that for each $x_1, x_2 \in \mathcal{Y}$,

$$f_1(x_1) = f(00\|x_1\|0^{2n}), \qquad f_2(x_1) = f(01\|x_1\|0^{2n}),$$
$$f_3(x_1) = f(10\|x_1\|0^{2n}), \qquad g_{\mathbf{I}}(x_1,x_2) = f(11\|x_1\|x_2\|0^{n}).$$

The distinctness of the first two bits ensures that $f_1, f_2, f_3, g_{\mathbf{I}}$ are all independent. Thus, this game is identical to the one we began with. Next, we replace $g_{\mathbf{I}}$ by $g_{\mathbf{I}}^*$, defined as

$$g_{\mathbf{I}}^*(x_1,x_2) := f(11\|x_1\|x_2\|f_2(f_1(x_1)\oplus x_2)\oplus x_2),$$

where we also call f_1 and f_2 in the ideal world. Since $f_2(f_1(x_1)\oplus x_2)\oplus x_2$ is a function of x_1 and x_2, $g_{\mathbf{I}}^*$ is still a random function of $x_1\|x_2$, making this game to behave identically with the one we started with.

For any $x \in \mathcal{Y}$ we denote

$$[x]_1 := 00\|x\|0^{2n}, \quad [x]_2 := 01\|x\|0^{2n}, \quad [x]_3 := 10\|x\|0^{2n}.$$

Thus the input spaces are defined as

$$\widetilde{\mathcal{X}}_{\mathbf{R}} := \{[x]_1, [x]_2, [x]_3 \mid x \in \mathcal{Y}\},$$
$$\widetilde{\mathcal{X}}_{\mathbf{I}} := \{[x]_1, [x]_2, 11\|x\|x'\|y \mid x, x', y \in \mathcal{Y}\}.$$

7.6.1.2 Bad Databases

Let $\mathcal{B}_{\mathbf{R}}$ be the set of databases $d_{\mathbf{R}}$ satisfying the following condition: we can find $u_1, v_1, u_1', v_1', x_2, v_2, x_2', v_2' \in \mathcal{Y}$ such that

- $([u_1]_1, v_1), ([u_1']_1, v_1'), ([v_1 \oplus x_2]_2, v_2), ([v_1' \oplus x_2']_2, v_2') \in d_{\mathbf{R}}$;
- $v_2 \oplus x_2 = v_2' \oplus x_2'$.

Next, let $\mathcal{B}_{\mathbf{I}}$ be the set of databases $d_{\mathbf{I}}$ satisfying the following condition: we can find $u_1, v_1, u_1', v_1', x_2, v_2, x_2', v_2' \in \mathcal{Y}$ such that

- $([u_1]_1, v_1), ([u_1']_1, v_1'), ([v_1 \oplus x_2]_2, v_2), ([v_1' \oplus x_2']_2, v_2') \in d_{\mathbf{I}}$;
- $v_2 \oplus x_2 = v_2' \oplus x_2'$.

Let $\mathcal{G}_{\mathbf{R}} := \mathcal{D}_{\mathbf{R}} \setminus \mathcal{B}_{\mathbf{R}}$ and $\mathcal{G}_{\mathbf{I}} := \mathcal{D}_{\mathbf{I}} \setminus \mathcal{B}_{\mathbf{I}}$. Thus the above definitions mean that in both $\mathcal{G}_{\mathbf{R}}$ and $\mathcal{G}_{\mathbf{I}}$, each $u_3 := v_2 \oplus x_2$ is associated with a unique pair (x_1, x_2). Then we can define the bijection $h : \mathcal{G}_{\mathbf{R}} \longrightarrow \mathcal{G}_{\mathbf{I}}$ as follows: for each $d_{\mathbf{R}}$ we define $d_{\mathbf{I}} := h(d_{\mathbf{R}})$ such that

- for each $u_1 \in \mathcal{Y}$, $d_{\mathbf{I}}([u_1]_1) = d_{\mathbf{R}}([u_1]_1)$;
- for each $u_2 \in \mathcal{Y}$, $d_{\mathbf{I}}([u_2]_2) = d_{\mathbf{R}}([u_2]_2)$;
- for each $x_1, x_2 \in \mathcal{Y}$ and the associated u_3, $d_{\mathbf{I}}(11 \| x_1 \| x_2 \| u_3) = d_{\mathbf{R}}([u_3]_3)$.

Using Lemma 7.4.4, we can complete the proof by showing that

$$\left(\perp \overset{3q}{\rightsquigarrow} \mathcal{B}_{\mathbf{I}}\right) + \left(\perp \overset{3q}{\rightsquigarrow} \mathcal{B}_{\mathbf{R}}\right) \leq 4\sqrt{\frac{10q^5}{2^n}}.$$

7.6.1.3 Sequence of Actions

Each query by the adversary to its oracle results in a sequence of three queries to f, one each to f_1, f_2, and one to f_3 in the real world or $g_{\mathbf{I}}^*$ in the ideal world, in that order. We view the query response phase as a sequence of $3q$ (possibly duplicate) *actions* and analyze the transition capacity at each action.

Action of f_1: For $i \in \{3k+1 : 0 \leq k \leq q-1\}$, for any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \leq i-1$ and any $x \in \mathcal{Y}$, since the property $\mathcal{B}_{\mathbf{R}}$ does not depend on $d_{\mathbf{R}}([x]_1)$, we have $\mathcal{S}_{x,d}^{\mathcal{B}_{\mathbf{R}}^c \hookrightarrow \mathcal{B}_{\mathbf{R}}} = \emptyset$. Thus,

$$[\![\mathcal{B}_{\mathbf{R}[\leq i-1]}^c \hookrightarrow \mathcal{B}_{\mathbf{R}[\leq i]}]\!] = 0, \qquad \forall\, i \in \{3k+1 : 0 \leq k \leq q-1\}. \tag{7.45}$$

By the same arguments

$$[\![\mathcal{B}_{\mathbf{I}[\leq i-1]}^c \hookrightarrow \mathcal{B}_{\mathbf{I}[\leq i]}]\!] = 0, \qquad \forall\, i \in \{3k+1 : 0 \leq k \leq q-1\}. \tag{7.46}$$

Action of f_2: Next we look at the transition capacity $[\![\mathcal{B}^c_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!]$ for $i \in \{3k+2 : 0 \le k \le q-1\}$. For any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \le i-1$ and any $x \in \mathcal{Y}$, we have

$$\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} := \{d_{\mathbf{R}}([u_2]_2) \oplus x_2 \oplus x_2' \mid d_{\mathbf{R}}([u_2]_2) \neq \perp\}.$$

Again, there are at most $\lceil (i-1)/3 \rceil^3$ choices for the pair (x_2, x_2', u_2), and arguing as before we have

$$[\![\mathcal{B}^c_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] \le \sqrt{\frac{10q^3}{2^n}}, \qquad \forall\, i \in \{3k+2 : 0 \le k \le q-1\}. \tag{7.47}$$

By the same arguments we can also show that

$$[\![\mathcal{B}^c_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] \le \sqrt{\frac{10q^3}{2^n}}, \qquad \forall\, i \in \{3k+2 : 0 \le k \le q-1\}. \tag{7.48}$$

Action of f_3 (resp. $g^*_{\mathbf{I}}$): Finally, for $i \in \{3k : 1 \le k \le q\}$, for any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \le i-1$ (resp. any $d_{\mathbf{I}}$ with $|d_{\mathbf{I}}| \le i-1$) and any $x \in \mathcal{Y}$, since the property $\mathcal{B}_{\mathbf{R}}$ (resp. $\mathcal{B}_{\mathbf{I}}$) does not depend on $d_{\mathbf{R}}([x]_3)$ (resp. $d_{\mathbf{I}}(11\|x_1\|x_2\|x)$), we have $\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} = \emptyset$ (resp. $\mathcal{S}^{\mathcal{B}^c_{\mathbf{I}} \hookrightarrow \mathcal{B}_{\mathbf{I}}}_{x,d} = \emptyset$). Thus,

$$[\![\mathcal{B}^c_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] = 0, \qquad \forall\, i \in \{3k : 1 \le k \le q\}, \tag{7.49}$$

and also,

$$[\![\mathcal{B}^c_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] = 0, \qquad \forall\, i \in \{3k : 1 \le k \le q\}. \tag{7.50}$$

Summing over the $3q$ actions using (7.45)-(7.50) gives

$$\left(\perp \overset{3q}{\rightsquigarrow} \mathcal{B}_{\mathbf{I}}\right) \le \sqrt{\frac{10q^5}{2^n}}, \qquad \left(\perp \overset{3q}{\rightsquigarrow} \mathcal{B}_{\mathbf{R}}\right) \le \sqrt{\frac{10q^5}{2^n}}. \tag{7.51}$$

Adding the two inequalities completes the proof of Theorem 7.6.1. □

7.6.2 Post-Quantum PRF Security of LRWQ

Finally, we analyze the post-quantum security of LRWQ (see Figure 7.3), defined as

$$g^{\mathtt{LRWQ}}_{\mathbf{R}}(x_1, x_2) := f_3(f_1(x_1) \oplus f_2(x_2)).$$

Theorem 7.6.2. *Let $\mathscr{A}$ be a (q,t)-quantum adversary distinguishing $g^{LRWQ}_{\mathbf{R}}$ from $g_{\mathbf{I}}$. Then there exists $(O(q), t_i)$-quantum distinguishers $\mathscr{B}_i$ against f_i, such that*

$$\mathbf{Adv}^{\mathrm{qcpa}}_{LRWQ}(\mathscr{A}) \le \sum_{i=1}^{3} \mathbf{Adv}^{\mathrm{qcpa}}_{f_i}(\mathscr{B}_i) + 12\sqrt{\frac{10q^5}{2^n}},$$

where $t_i \in \widetilde{O}(t + q^2)$, for all $i \in [1;3]$.

We follow the blueprint in Section 7.5 step by step and write down the details specific to the LRWQ construction. Due to the similarity of the analysis to the TNT construction, we omit redundant details where necessary.

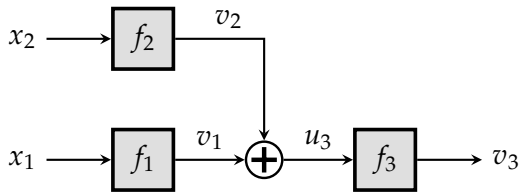

Figure 7.3: *The* `LRWQ` *construction by Hosoyamada et al. [177].*

7.6.2.1 Notional Setup

As before we will simulate all the random functions using a single random function $f : \{0,1\}^{3n+2} \to \{0,1\}^n$. For each $x_1, x_2 \in \mathcal{Y}$,

$$f_1(x_1) = f(00\|x_1\|0^{2n}), \qquad f_2(x_1) = f(01\|x_1\|0^{2n}),$$
$$f_3(x_1) = f(10\|x_1\|0^{2n}), \qquad g^*_{\mathbf{I}}(x_1, x_2) = f(11\|x_1\|x_2\|f_1(x_1) \oplus f_2(x_2)).$$

Using a similar argument as before we can conclude that this game behaves identical with the standard PRF game. Let $\mathcal{D}_{\mathbf{R}}, \mathcal{D}_{\mathbf{I}}, \widetilde{\mathcal{X}}_{\mathbf{R}}, \widetilde{\mathcal{X}}_{\mathbf{I}}$ be as before.

7.6.2.2 Bad Databases

Let $\mathcal{B}_{\mathbf{R}}$ be the set of databases $d_{\mathbf{R}}$ satisfying the following condition: we can find $x_1, v_1, x_1', v_1', x_2, v_2, x_2', v_2' \in \mathcal{Y}$ such that

- $([u_1]_1, v_1), ([u_1']_1, v_1'), ([u_2]_2, v_2), ([u_2']_2, v_2') \in d_{\mathbf{R}}$;
- $v_1 \oplus v_2 = v_1' \oplus v_2'$.

Next, let $\mathcal{B}_{\mathbf{I}}$ be the set of databases $d_{\mathbf{I}}$ satisfying the following condition: we can find $x_1, v_1, x_1', v_1', x_2, v_2, x_2', v_2' \in \mathcal{Y}$ such that

- $([u_1]_1, v_1), ([u_1']_1, v_1'), ([u_2]_2, v_2), ([u_2']_2, v_2') \in d_{\mathbf{R}}$;
- $v_1 \oplus v_2 = v_1' \oplus v_2'$.

As before let $\mathcal{G}_{\mathbf{R}} := \mathcal{D}_{\mathbf{R}} \setminus \mathcal{B}_{\mathbf{R}}$ and $\mathcal{G}_{\mathbf{I}} := \mathcal{D}_{\mathbf{I}} \setminus \mathcal{B}_{\mathbf{I}}$. Thus the above definitions mean that in both $\mathcal{G}_{\mathbf{R}}$ and $\mathcal{G}_{\mathbf{I}}$, each $u_3 := v_1 \oplus v_2$ is associated with a unique pair (x_1, x_2). Then we can define the bijection $h : \mathcal{G}_{\mathbf{R}} \longrightarrow \mathcal{G}_{\mathbf{I}}$ as follows: for each $d_{\mathbf{R}}$ we define $d_{\mathbf{I}} := h(d_{\mathbf{R}})$ such that

- for each $u_1 \in \mathcal{Y}$, $d_{\mathbf{I}}([u_1]_1) = d_{\mathbf{R}}([u_1]_1)$;
- for each $u_2 \in \mathcal{Y}$, $d_{\mathbf{I}}([u_2]_2) = d_{\mathbf{R}}([u_2]_2)$;
- for each $x_1, x_2 \in \mathcal{Y}$ and the associated u_3, $d_{\mathbf{I}}(11\|x_1\|x_2\|u_3) = d_{\mathbf{R}}([u_3]_3)$.

To complete the proof of Theorem 7.6.2, we just need to show that

$$\left(\perp \overset{3q}{\rightsquigarrow} \mathcal{B}_{\mathbf{I}}\right) + \left(\perp \overset{3q}{\rightsquigarrow} \mathcal{B}_{\mathbf{R}}\right) \leq 4\sqrt{\frac{10q^5}{2^n}}.$$

7.6.2.3 Sequence of Actions

As before, we analyze the actions on the component functions, each corresponding to f_1, f_2, and f_3 or $g^*_{\mathbf{I}}$.

Action of f_1: For $i \in \{3k+1 : 0 \le k \le q-1\}$, for any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \le i-1$ and any $x \in \mathcal{Y}$, we have

$$\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} := \{d_{\mathbf{R}}([u_1]_1) \oplus d_{\mathbf{R}}([u_1']_1) \oplus d_{\mathbf{R}}([u_2]_2) \,|\, d_{\mathbf{R}}([u_1]_1) \neq \perp, d_{\mathbf{R}}([u_1']_1) \neq \perp, d_{\mathbf{R}}([u_2]_2) \neq \perp\}.$$

There are at most $\lceil (i-1)/3 \rceil^3$ choices for the pair (u_1, u_1', u_2), so from Lemma 7.4.2 we have

$$[\![\mathcal{B}^c_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] = \sqrt{\frac{10q^3}{2^n}}, \qquad \forall\, i \in \{3k+1 : 0 \le k \le q-1\}. \tag{7.52}$$

By the same arguments

$$[\![\mathcal{B}^c_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] = \sqrt{\frac{10q^3}{2^n}}, \qquad \forall\, i \in \{3k+1 : 0 \le k \le q-1\}. \tag{7.53}$$

Action of f_2: Next, we look at the transition capacity $[\![\mathcal{B}^c_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!]$ for $i \in \{3k+2 : 0 \le k \le q-1\}$. For any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \le i-1$ and any $x \in \mathcal{Y}$, we have

$$\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} := \{d_{\mathbf{R}}([u_2]_2) \oplus d_{\mathbf{R}}([u_2']_2) \oplus d_{\mathbf{R}}([u_1]_1) \,|\, d_{\mathbf{R}}([u_1]_1) \neq \perp, d_{\mathbf{R}}([u_1']_1) \neq \perp, d_{\mathbf{R}}([u_2]_2) \neq \perp\}.$$

There are at most $\lceil (i-1)/3 \rceil^3$ choices for the pair (u_1, u_1', u_2), so from Lemma 7.4.2 we have

$$[\![\mathcal{B}^c_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] \le \sqrt{\frac{10q^3}{2^n}}, \qquad \forall\, i \in \{3k+2 : 0 \le k \le q-1\}. \tag{7.54}$$

By the same arguments

$$[\![\mathcal{B}^c_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] \le \sqrt{\frac{10q^3}{2^n}}, \qquad \forall\, i \in \{3k+2 : 0 \le k \le q-1\}. \tag{7.55}$$

Action of f_3 (resp. $g^*_{\mathbf{I}}$): Finally, for $i \in \{3k : 1 \le k \le q\}$, for any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \le i-1$ (resp. any $d_{\mathbf{I}}$ with $|d_{\mathbf{I}}| \le i-1$) and any $x \in \mathcal{Y}$, since the property $\mathcal{B}_{\mathbf{R}}$ (resp. $\mathcal{B}_{\mathbf{I}}$) does not depend on $d_{\mathbf{R}}([x]_3)$ (resp. $d_{\mathbf{I}}(11\|x_1\|x_2\|x)$), we have $\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} = \emptyset$ (resp. $\mathcal{S}^{\mathcal{B}^c_{\mathbf{I}} \hookrightarrow \mathcal{B}_{\mathbf{I}}}_{x,d} = \emptyset$). Thus,

$$[\![\mathcal{B}^c_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] = 0, \qquad \forall\, i \in \{3k : 1 \le k \le q\}, \tag{7.56}$$

and also,

$$[\![\mathcal{B}^c_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] = 0, \qquad \forall\, i \in \{3k : 1 \le k \le q\}. \tag{7.57}$$

Summing over the $3q$ actions using (7.52)-(7.57) gives

$$\left(\perp \overset{3q'}{\rightsquigarrow} \mathcal{B}_{\mathbf{I}}\right) \le 2\sqrt{\frac{10q^5}{2^n}}, \qquad \left(\perp \overset{3q'}{\rightsquigarrow} \mathcal{B}_{\mathbf{R}}\right) \le 2\sqrt{\frac{10q^5}{2^n}}. \tag{7.58}$$

Adding the two inequalities completes the proof of Theorem 7.6.2. □

7.7 Post-Quantum TPRP Security of TNT and LRWQ

To establish the TPRP security for the aforementioned schemes, we note that TNT and LRWQ can be viewed as tweakable block ciphers by instantiating f_1, f_2, f_3 with keyed permutations and using the second input, x_2, as the tweak value. By applying Lemma 7.2.1 and Lemma 7.2.2, together with Theorem 7.6.1, Theorem 7.6.2 and Theorem 7.6.2, in a specific order, we get the following corollary on the TPRP security of TNT and LRWQ .

Corollary 7.7.1. *For any $\widetilde{E} \in \{TNT, LRWQ\}$, let $\mathscr{A}$ be a (q, τ)-quantum adversary distinguishing $\widetilde{E}$ from $\widetilde{\mathsf{P}}$, a uniform random tweakable permutation of $\{0,1\}^n$ with n-bit tweaks. Then, there exists $(O(q), \tau_i)$-quantum distinguishers $\mathscr{B}_i$ against f_i, such that*

$$\mathbf{Adv}^{\mathrm{qtprp}}_{\widetilde{E}}(\mathscr{A}) \leq \sum_{i=1}^{3} \mathbf{Adv}^{\mathrm{qprp}}_{f_i}(\mathscr{B}_i) + O\left(\sqrt{\frac{q^5}{2^n}} + \sqrt{\frac{q^6}{2^n}} + \frac{q^3}{2^n}\right),$$

where $\tau_i \in \widetilde{O}(\tau + q^2)$, for all $i \in [1;3]$.

8

Flaws in Post-Quantum Security Proofs for The Adaptive Setting

Classically, symmetric cryptographic algorithms are often constructed as modes of operation over fixed-length primitives like pseudorandom permutations (PRPs) or functions (PRFs). Notable examples include the Luby-Rackoff cipher, Lai-Massey, and Misty ciphers. These constructions are usually secure up to the birthday bound, meaning their security holds until the adversary makes around $2^{n/2}$ queries, where n is the output size. Recent efforts have sought to build beyond-birthday-bound secure constructions, such as the sum of permutations and truncation of permutations, advancing the field of provable security with new proof techniques like the H-coefficient technique, mirror theory, and Fourier analysis.

In the quantum setting, research on the security of these well-known constructions is still developing. While there are some known attacks, the security proofs for many constructions remain less tight, even within the quantum birthday bound (up to $2^{n/3}$ queries). The landscape has shifted with Zhandry's compressed oracle technique, which has become a cornerstone for recent quantum security proofs. This technique helps to structure security proofs by ensuring that critical "bad events" are handled correctly, a necessary step in proving the indistinguishability of a construction from a random function. Our contributions build on this framework by identifying flaws in previous security proofs, proving the security of Misty schemes in the quantum setting, and proposing a new proof for the Luby-Rackoff construction under non-adaptive adversaries.

8.1 Revising The Post-Quantum Security of Luby-Rackoff

To understand the post-quantum proof for the qPRP security of four-round Luby-Rackoff, we first introduce the compressed oracle interpretation by Hosoyamada and Iwata as presented in [174]. In their work, they propose a variant of the standard oracle designed to characterize and analyze databases explicitly in the computational basis. This is

achieved with the help of an ancillary flag bit that indicates whether a database entry is defined or not.

8.1.1 The Recording Standard Oracle With Errors

Let $\mathcal{S} \subseteq \mathcal{X}$ and $\mathcal{Z} = \{0,1\} \times \mathcal{Y}$. For any partial function $f : \mathcal{S} \to \mathcal{Y}$, we associate the database function $d_f : \mathcal{X} \to \mathcal{Z}$ defined as:

$$d_f(x) := \begin{cases} (1, y) & \text{when } f(x) = y \in \mathcal{Y}, \\ (0, 0^n) & \text{if } f(x) \text{ is undefined.} \end{cases}$$

We note that $\perp$ in the original interpretation by Zhandry [325] corresponds to $(0, 0^n)$ in HI interpretation. As before, we drop the subscripts when f is clear from context.

Databases. We define the database space as the $2^{(n+1)2^m}$-dimensional complex Hilbert space $\mathcal{H}_{db} = \mathbb{C}[\mathcal{Z}]$, which is isomorphic to $\mathbb{C}^{2^{(n+1)2^m}}$. Note that in this interpretation, not all databases $d \in \mathcal{Z}$ can be associated with a partial function f. Therefore, we say a database $d = ((b_0, \beta_0), \ldots, (b_{2^m-1}, \beta_{2^m-1}))$ is *valid* if for any $i \in [0; 2^m - 1]$, if $b_i = 0$, then β_i must be 0^n. Consequently, any valid database $((b_0, \beta_0), \ldots, (b_{2^m-1}, \beta_{2^m-1}))$ is identified with the set $\{(i, \beta_i) | b_i = 1\}$, which corresponds to the truth table of a partially-defined function from $\{0,1\}^m$ to $\{0,1\}^n$. Accordingly, let Π_{valid} be the orthogonal projection onto the vector space spanned by valid databases.

The Recording Standard Oracle. Any database $|d\rangle \in \mathbb{C}[\mathcal{Z}]$ can be equivalently viewed as an array of 2^m cells $|d[0]\rangle \ldots |d[2^m - 1]\rangle$. Writing $|d[i]\rangle$ as $|b_i, \beta_i\rangle$ for each $i \in \{0, 1, \ldots, 2^m - 1\}$ (where b_i and β_i are respectively the control qubit and the response register of the i-th cell $|d[i]\rangle$ of $|d\rangle$), the standard oracle **stO** is now defined as:

$$\mathbf{stO}|i, y\rangle|d\rangle := |i, y + \beta_i\rangle|d\rangle$$

for each $|i, y, d\rangle \in \mathcal{H}_{in} \times \mathcal{H}_{out} \times \mathcal{H}_{db}$. For a database $|d\rangle$ such that $|d[i]\rangle = |0, 0^n\rangle$, we define $|d \cup (i, \beta)\rangle$ to be the database with $|1, \beta\rangle$ as its i-th cell and identical to $|d\rangle$ in all other cells.

Next, we define the following unitaries on the database cells:

$$\mathbf{IH}_0 := \mathbf{I}_1 \otimes \mathbf{H}^{\otimes n}, \qquad \mathbf{Tg}_0 := \mathbf{I}_1 \otimes |0^n\rangle\langle 0^n| + \mathbf{X}(\mathbf{I}_{2^n} - |0^n\rangle\langle 0^n|),$$
$$\mathbf{cH}_0 := |0\rangle\langle 0| \otimes \mathbf{I}_{2^n} + |1\rangle\langle 1| \otimes \mathbf{H}^{\otimes n},$$

where $\mathbf{I}_k$ is the identity map on some k qubits and databases:

$$\mathbf{IH} := \mathbf{IH}_0^{\otimes 2^m}, \qquad \mathbf{Tg} := \mathbf{Tg}_0^{\otimes 2^m}, \qquad \mathbf{cH} := \mathbf{cH}_0^{\otimes 2^m},$$

where $\mathbf{X}$ and $\mathbf{H}$ are the well-known flip and Hadamard operators on $\mathbb{C}$, i.e. in the computational basis:

$$\mathbf{X} := |0\rangle\langle 1| + |1\rangle\langle 0| \qquad \mathbf{H} := \frac{1}{2}\left(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|\right).$$

A straightforward verification shows that all these operators are indeed Hermitian. Following these, we can now define the encoding (resp. decoding) operator **enc** (resp. **dec**), which acts on databases as follows:

$$\mathbf{enc} := \mathbf{cH} \circ \mathbf{Tg} \circ \mathbf{IH};$$
$$\mathbf{dec} := \mathbf{enc}^{\dagger} = \mathbf{IH} \circ \mathbf{Tg} \circ \mathbf{cH};$$

The recording standard oracle **RStOE**, due to Hosoyamada and Iwata [174], is defined as:

$$\mathbf{RStOE} := (\mathbf{I}_{2^{m+n}} \otimes \mathbf{enc})\mathbf{stO}(\mathbf{I}_{2^{m+n}} \otimes \mathbf{dec})$$

Hence, **RStOE** first decodes the database, then applies **stO** on the adversary's registers and the decoded database, and then encodes the database again. Let $|\mathbf{0}\rangle$ denote the valid empty database.

Post-Quantum Security with RStOE. In the following propositions, Hosoyamada and Iwata demonstrate how the recording standard oracle can be utilized to prove the post-quantum PRF security of symmetric schemes [174, 177].

Proposition 8.1.1 (Proposition 1 in [177]). *Suppose that the oracle state is initialized in $|\mathbf{0}\rangle$. For any $i \geq 1$, if the oracle state register is measured after i queries, then the resulting database d is valid, and contains at most i entries.*

Proposition 8.1.2 (Proposition 2 in [177]). *For any valid database d satisfying $d[i] = |0, 0^n\rangle$, we have*

$$\mathbf{RStOE}|i, y\rangle|d \cup (i, \beta)\rangle = |i, y \oplus \beta\rangle|d \cup (i, \beta)\rangle + |\epsilon_1\rangle; \tag{8.1}$$
$$\mathbf{RStOE}|i, y\rangle|d\rangle = \sum_{\beta \in \{0,1\}^n} \frac{1}{2^{n/2}}|i, y \oplus \beta\rangle|d \cup (i, \beta)\rangle + |\epsilon_2\rangle; \tag{8.2}$$

for some $|\epsilon_1\rangle$ and $|\epsilon_2\rangle$ such that $\||\epsilon_1\rangle\|, \||\epsilon_2\rangle\| \in O(1/\sqrt{2^n})$.

We note that Hosoyamada and Iwata provide an explicit description of $|\epsilon_1\rangle$ and $|\epsilon_2\rangle$ in [174]. Intuitively, these vectors can be interpreted as error terms introduced during the lazy sampling of a quantum random function due to interference, which also explains the name of the framework.

Finally, the main technical result used to study the indistinguishability game and bound the advantage is given below.

Proposition 8.1.3 (Proposition 3 in [177]). *For each $j \in [0; q]$, let $|\mathbb{R}_j\rangle$ and $|\mathbb{I}_j\rangle$ denote the state vector corresponding to the real and ideal worlds after the j-th query, respectively. Suppose, there exist vectors $|\mathbb{R}_j^{\mathsf{g}}\rangle$, $|\mathbb{R}_j^{\mathsf{b}}\rangle$, $|\mathbb{I}_j^{\mathsf{g}}\rangle$, $|\mathbb{I}_j^{\mathsf{b}}\rangle$ and non-negative reals $\epsilon_{\mathbf{I}}^{(j)}$ and $\epsilon_{\mathbf{R}}^{(j)}$ such that*

1. $|\mathbb{R}_j\rangle = |\mathbb{R}_j^{\mathsf{g}}\rangle + |\mathbb{R}_j^{\mathsf{b}}\rangle$, $|\mathbb{I}_j\rangle = |\mathbb{I}_j^{\mathsf{g}}\rangle + |\mathbb{I}_j^{\mathsf{b}}\rangle$;
2. $|\mathbb{R}_j^{\mathsf{g}}\rangle\langle\mathbb{R}_j^{\mathsf{g}}| = |\mathbb{I}_j^{\mathsf{g}}\rangle\langle\mathbb{I}_j^{\mathsf{g}}|$;
3. $\||\mathbb{I}_j^{\mathsf{b}}\rangle\| \leq \||\mathbb{I}_{j-1}^{\mathsf{b}}\rangle\| + \epsilon_{\mathbf{I}}^{(j)}$, $\||\mathbb{R}_j^{\mathsf{b}}\rangle\| \leq \||\mathbb{R}_{j-1}^{\mathsf{b}}\rangle\| + \epsilon_{\mathbf{R}}^{(j)}$.

Then, for any computationally unbounded and deterministic distinguisher A we have

$$\|\mathsf{Tr}_{\mathcal{H}_{\mathbf{I}_{db}}}(\rho^q_{A,\mathbf{I}}) - \mathsf{Tr}_{\mathcal{H}_{\mathbf{R}_{db}}}(\rho^q_{A,\mathbf{R}})\|_1 \leq \sum_{i=1}^{q} \epsilon^{(j)}_{\mathbf{I}} + \sum_{i=1}^{q} \epsilon^{(j)}_{\mathbf{R}},$$

where

$$\rho^q_{A,\mathbf{R}} = |\psi_A\rangle\langle\psi_A| \otimes |\mathbf{0_R}\rangle\langle\mathbf{0_R}|, \qquad \rho^q_{A,\mathbf{I}} = |\psi'_A\rangle\langle\psi'_A| \otimes |\mathbf{0_I}\rangle\langle\mathbf{0_I}|,$$

for some norm-1 vector $\psi_A, \psi'_A \in \mathcal{H}_A$ and $|\mathbf{0_R}\rangle$ and $|\mathbf{0_I}\rangle$ denote the all zero database states in the real and ideal worlds respectively.

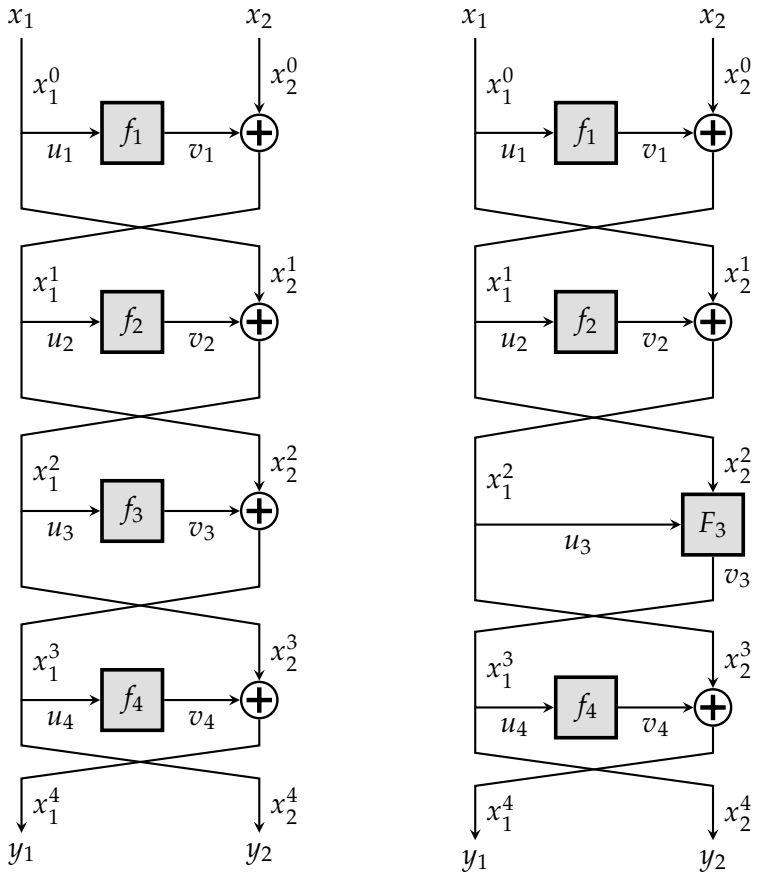

Figure 8.1: 4*-round Luby-Rackoff (left) and* 4*-round Luby-Rackoff with a BIG function (right).*

8.1.2 Flaws in The qPRP Proof of Luby-Rackoff

Recall the definition of the Luby-Rackoff construction for r rounds, following the Feistel network paradigm.

The Luby-Rackoff Construction. For some $r \geq 1$ and $f_1, \ldots, f_r : \{0,1\}^n \to \{0,1\}^n$, we define $g : [1;r] \times \{0,1\}^{2n} \to \{0,1\}^{2n}$ by the mapping:

$$(i, x_1, x_2) \longmapsto (x_2 \oplus f_i(x_1), x_1),$$

and write $g_i(\cdot,\cdot) := g(i,\cdot,\cdot)$. The r-round Luby-Rackoff construction, denoted $\mathtt{LR}_r$ is defined as:

$$(x_1, x_2) \longmapsto g_r \circ \cdots \circ g_1(x_1, x_2). \tag{8.3}$$

For all $i \in [1; r]$, we write (also see Figure 8.1):

- $x^{i-1} := (x_1^{i-1}, x_2^{i-1})$ to denote the input to g_i, where $x^0 := x = (x_1, x_2)$, denotes the input to $\mathtt{LR}_r$.
- (u_i, v_i) to denote the input-output tuple corresponding to f_i.
- $y = (y_1, y_2) := (x_1^r, x_2^r)$ to denote the output of $\mathtt{LR}_r$.

The following IND-qCPA security bound for $\mathtt{LR}_4$ was given by Hosoyamada and Iwata in [174].

Theorem 8.1.1 (Theorem 3 in [174]). *Suppose $f_1, f_2, f_3, f_4 : \{0,1\}^n \to \{0,1\}^n$ are four mutually independent uniform random functions. Then, for any $q \geq 0$, and any quantum adversary $\mathscr{A}$ that makes at most q CPA queries, we have*

$$\mathbf{Adv}^{\mathrm{qcpa}}_{LR_4}(\mathscr{A}) = O\left(\sqrt{\frac{q^3}{2^n}}\right).$$

Proof Sketch of Theorem 8.1.1. The high level proof approach of Theorem 8.1.1 proceeds as follows. First, we simulate the random functions f_1, f_2, f_3, f_4 using independent instances of **RStOE** with the corresponding databases, d_1, d_2, $d_{\mathbf{R}}$, d_4, respectively.

Next, the authors apply a series of hybrids, introducing intermediate constructions between the real construction $\mathtt{LR}_4$, and the ideal construction, a uniform random function $\Gamma : \{0,1\}^{2n} \to \{0,1\}^{2n}$. The first of these intermediate constructions is a length-preserving function, that we refer as $\widetilde{\mathtt{LR}_4}$, defined by the mapping (see also Fig.):

$$(x_1, x_2) \mapsto g_4 \circ G_3 \circ g_2 \circ g_1(x_1, x_2), \tag{8.4}$$

where $G_3(x_1', x_2') := (F_3(x_1', x_2'), x_1')$ for all $(x_1', x_2') \in \{0,1\}^{2n}$. The function $F_3 : \{0,1\}^{2n} \to \{0,1\}^n$ is a uniform random function, to be implemented by an appropriate **RStOE**, named $d_{\mathbf{I}}$.

To explain the flaw in the proof we solely focus on the distance between $\mathtt{LR}_4$ and $\widetilde{\mathtt{LR}_4}$. Indeed, showing that this distance is negligible is the core technical core of the proof. To further simplify the explanation we drop the the application of f_4, and consider the chopped output x_3^1, as denoted in Fig. Finally, we write $d^{\mathbf{R}} = (d_1, d_2, d_{\mathbf{R}})$ and $d^{\mathbf{I}} = (d_1, d_2, d_{\mathbf{I}})$.

In essence, their proof reduces to utilizing Proposition 8.1.3 by applying Proposition 8.1.2 iteratively to examine the actions of the functions f_1, f_2, and f_3 (only in the real world), and F_3 (only in the ideal world) in sequence. This is followed by the respective uncomputation steps for f_2 and f_1, in that order.

They key observation of the proof is that $\mathtt{LR}_4$ and $\widetilde{\mathtt{LR}_4}$ are indistinguishable as long as the inputs to f_3 (resp. F_3 in the ideal world) are pairwise distinct across all queries. Hence, we say a database triple $d^{\mathbf{R}} = (d_1, d_2, d_{\mathbf{R}})$ (res. $d^{\mathbf{I}} = (d_1, d_2, d_{\mathbf{I}})$ in the ideal world) is *good* if the following condition holds: there are no $(u_1, v_1), (u_1', v_1') \in d_1$, $(u_2, v_2), (u_2', v_2') \in d_2$, and $(u_3, v_3) \in d_{\mathbf{R}}$ (res. $(u_3, x_2^2, v_3) \in d_{\mathbf{I}}$ in the real world) such that $u_1 \oplus v_2 = u_1' \oplus v_2' = u_3'$. All other triples are considered *bad*. Let Π_{bad} be the projection onto the space spanned by bad databases. As seen in earlier proofs, a key observation about good databases is that they defined a one to one mapping from $d^{\mathbf{R}} \mapsto [d^{\mathbf{R}}]_{\mathbf{I}}$ between the real and ideal databases. Namely, the two worlds can be easily shown to behave identically when the databases remain good throughout the execution. Therefore, by setting $|\mathbb{R}_j^{\mathsf{b}}\rangle = \Pi_{bad}|\mathbb{R}_j\rangle$, $|\mathbb{R}_j^{\mathsf{g}}\rangle = |\mathbb{R}_j\rangle - |\mathbb{R}_j^{\mathsf{b}}\rangle$, $|\mathbb{I}_j^{\mathsf{b}}\rangle = \Pi_{bad}|\mathbb{I}_j\rangle$, and $|\mathbb{I}_j^{\mathsf{g}}\rangle = |\mathbb{I}_j\rangle - |\mathbb{I}_j^{\mathsf{b}}\rangle$, we satisfy the first two conditions in Proposition 8.1.3.

Following this proof sketch we are left with the task of studying the action of each function call, and bound the norm of the bad vectors after each application, assuming that the state is spanned by good databases before the action. Particularly, we concentrate on the action of the first function, f_1, and point out exactly where argumentation of the proof fails.

Action of f_1 and the Trivialization of Norm. The main flaw we observe in the proof of Theorem 8.1.1 is that one of the norms can only be trivially bounded, i.e., the norm is at most 1. More generally, for any unit vector $|\psi\rangle$ and an arbitrary projection operator Π, we say that $\|\Pi|\psi\rangle\|$ is *trivially bounded* when we simply use the fact that $\|\Pi|\psi\rangle\| \leq 1$.

To explain the flaw in the proof we study the action of f_1 in the ideal world, although the same issue lies in the real world application as well. For brevity we assume that the output of f_1 is written on some ancillary register to be used in later actions. By applying repeated applications of Proposition 8.1.2 recursively, we deduce that there exists two vectors $|\epsilon_1\rangle$ and $|\epsilon_2\rangle$ such that:

$$\begin{aligned}\mathbf{O}_{f_1}|\mathbb{I}_{j-1}^{\mathsf{g}}\rangle := &\sum_{\substack{x,y,z,d^{\mathbf{I}} \\ d^{\mathbf{I}}:\text{good} \\ d_1(x_1)\neq\perp}} \alpha_{x,y,z,d^{\mathbf{I}}}^{(j-1)}|x,y,z\rangle \otimes |d_1(x_1)\rangle \otimes |d^{\mathbf{I}}\rangle \\ &+ \frac{1}{2^{n/2}} \sum_{\substack{x,y,z,\beta,d^{\mathbf{I}} \\ d^{\mathbf{I}}:\text{good} \\ d_1(x_1)=\perp}} \alpha_{x,y,z,d^{\mathbf{I}}}^{(j-1)}|x,y,z\rangle \otimes |\beta\rangle \otimes |d^{\mathbf{I}} \cup (x_1,\beta)_1\rangle \\ &+ |\epsilon_1\rangle + |\epsilon_2\rangle,\end{aligned}$$

where $|d^{\mathbf{I}} \cup (x_1,\beta)_1\rangle = |d_1 \cup (x_1,\beta)\rangle \otimes |d_2\rangle \otimes |d_{\mathbf{I}}\rangle$ denotes the database that is same as $|d^{\mathbf{I}}\rangle$ except for $d_1(x_1)$ which has been newly defined as β. For the sake of argument, we are only concerned on the second summand denoted $|\mathbb{I}_j^{g,1}\rangle$, which gives the state transition on a fresh input to f_1 starting with a good state. Intuitively, a new entry (x_1,β) is recorded in d_1 at the cost of an amplitude factor of $2^{-n/2}$.

Formally, we are interested in the norm below, which is a reformulation of [174, (51)]:

$$\|\Pi_{bad}|\mathbb{I}_j^{g,1}\rangle\|^2 = \| \sum_{\substack{x,y,z,\beta,d^{\mathbf{I}} \\ d^{\mathbf{I}}:\text{good} \\ d_1(x_1)=\perp \\ d^{\mathbf{I}}\cup(x_1,\beta)_1:\text{bad}}} \frac{\alpha^{(j-1)}_{x,y,z,d^{\mathbf{I}}}}{2^{n/2}}|x,y,z\rangle \otimes |\beta\rangle \otimes |d^{\mathbf{I}} \cup (x_1,\beta)_1\rangle\|^2$$

$$= \sum_{\substack{x,y,z,\beta,d^{\mathbf{I}} \\ d^{\mathbf{I}}:\text{good} \\ d_1(x_1)=\perp \\ d^{\mathbf{I}}\cup(x_1,\beta)_1:\text{bad}}} \left|\frac{\alpha^{(j-1)}_{x,y,z,d^{\mathbf{I}}}}{2^{n/2}}\right|^2 \tag{8.5}$$

$$= \sum_{\substack{x,y,z,d^{\mathbf{I}} \\ d^{\mathbf{I}}:\text{good} \\ d_1(x_1)=\perp}} \left|\alpha^{(j-1)}_{x,y,z,d^{\mathbf{I}}}\right|^2 \sum_{\substack{\beta \\ d^{\mathbf{I}}\cup(x_1,\beta)_1:\text{bad}}} \frac{1}{2^n} \tag{8.6}$$

$$\leq O\left(\frac{j}{2^n}\right) \sum_{\substack{x,y,z,d^{\mathbf{I}} \\ d^{\mathbf{I}}:\text{good} \\ d_1(x_1)=\perp}} \left|\alpha^{(j-1)}_{x,y,z,d^{\mathbf{I}}}\right|^2 \tag{8.7}$$

$$\leq O\left(\frac{j}{2^n}\right), \tag{8.8}$$

where (8.7) to (8.8) follows from the fact that $\||\mathbb{I}_{j-1}\rangle\| \leq 1$. However, there is no supporting argument in [174], that would explain why inequality from (8.6) to (8.7) is correct. In fact, for this to be correct, the authors must show that $|\{\beta : d^{\mathbf{I}} \cup (x_1,\beta)_1 \text{ is bad}\}| \leq j$. We claim that this argument is wrong, and prove the following claim to support our counter argument.

Claim 8.1.1. *In the ideal world, one has*

$$\sum_{\substack{\beta \\ d^{\mathbf{I}}\cup(x_1,\beta)_1:\text{bad}}} \frac{1}{2^n} = O(1).$$

Proof. We note that $d^{\mathbf{I}} \cup (x_1,\beta)_1$ is bad if and only if there exists distinct database entries $(u_1',v_1') \in d_1$, $(u_2,v_2),(u_2',v_2') \in d_2$, and $(u_1' \oplus v_2', u_2', v_3') \in d_{\mathbf{I}}$ such that: $x_1 \oplus v_2 = u_1' \oplus v_2'$. However, this condition is independent of β. Hence, in the worst case, this condition can be true for all possible β. □

This completely breaks the security proof, as this revised bound leads to a trivial bound of $O(1)$ on the corresponding advantage.

Does Increasing the Number of Rounds Help? One might assume that increasing the number of rounds to more than three could help avoid the flaw described above. Unfortunately, as we will see below, the "trivialization of norm" appears to be a fundamental issue in the "identical up to bad" proof strategy for the current quantum

proofs. In fact, we argue that this issue persists for an input collision at f_i for any odd $i \in [1; r]$. A similar argument can be made for an even i.

Consider the database snapshot after $j \geq 2$ queries. Suppose, the adversary makes a query (x_1, x_2), such that $d_1(x_1) = \perp$, i.e., the database entry corresponding to x_1 is empty, and a new entry (x_1, β) is to be created. Now, if we have distinct

$$(u'_1, v'_1) \in d_1, (u_2, v_2), (u'_2, v'_2) \in d_2, \ldots, (u_{i-1}, v_{i-1}), (u'_{i-1}, v'_{i-1}) \in d_{i-1}, (u'_i, v'_i) \in d_i,$$

such that $u'_i = u'_1 \oplus \left(v'_2 \oplus \cdots \oplus v'_{i-1}\right)$ and $x_1 \oplus (v_2 \oplus \cdots \oplus v_{i-1}) = u'_1 \oplus \left(v'_2 \oplus \cdots \oplus v'_{i-1}\right)$. Then, there is a possibility[1] that this query leads to a collision at the input of f_i. Moreover, this condition is independent[2] of β, leading to a similar trivialization of norms as in Claim 8.1.1. Thus, this line of argumentation becomes effectively useless.

8.2 The Non-Adaptive IND-qCPA Security of $\mathtt{LR}_4$

The primary reason the existing Luby-Rackoff proof fails is the lack of global knowledge regarding the adversarial query pattern. At any given moment, the compressed oracle only has access to the information recorded in the database and the current input. Consequently, it must consider every possible combination of global inputs, as demonstrated in Section 8.1.2, which leads to a trivialization of norms in the case of $\mathtt{LR}_4$. However, for several other constructions, such as TNT and LRWQ, it is still possible to reconstruct a moderately global view to achieve a meaningful security bound, as demonstrated in Section 7.6.

8.2.1 The Dummy Call Idea

In the non-adaptive setting, the adversary makes a single query of the form $x^q = (x_1, \ldots, x_q)$. We can employ a single dummy compressed oracle call to record x^q, and then implement the oracle at-hand. Note that the compressed oracle in both the dummy call and actual oracle evaluation can be implemented by a single compressed oracle using the prefixed oracle technique, introduced in Section 7.4.2. More formally, fix some $t \in [1; m]$ and suppose $\mathbf{O}_f$ denote the stateful oracle corresponding to the function $f : \{0,1\}^m \to \{0,1\}^n$, defined as follows:

$$\mathbf{O}_f := \mathbf{F}_{t-1}\mathbf{cO}^{\mathbf{p}_{t-1}} \ldots \mathbf{cO}^{\mathbf{p}_1}\mathbf{F}_0,$$

where $\mathbf{p}$ is a (t, ℓ)-domain-separator, with $\ell \gg m$, and the unitaries $\mathbf{F}_0, \ldots, \mathbf{F}_{t-1}$ only operate on the input, output and ancillary qubits, if any. Then, the q-query variant of

[1] We are obviously over counting by considering all possible combinations of queries. In fact, most of these combinations are never queried by the adversary. However, as of now, there is no effective way to determine the query ordering from database entries.

[2] This independence only holds in relation to the badness condition. In a typical execution of $\mathtt{LR}_r$, these variables will obviously depend on β. However, due to the badness condition and the ignorance of query ordering (see the above point), this dependence is lost.

$\mathbf{O}_f$ with dummy call is defined to be the sequence

$$(\mathbf{cO}^{\mathbf{p}_t})^\dagger \circ \mathbf{O}_f^{\otimes q} \circ \mathbf{cO}^{\mathbf{p}_t},$$

where the database space is $\mathbb{D} = \mathbb{C}^{(2^n+1)^{2^\ell}}$, with $\ell \geq mq + \lceil \log_2 t \rceil$. In other words, we enclose the original non-adaptive oracle between two compressed oracle calls, which record and erase the global input $(x^q, \widehat{y^q})$. Note that erasing the dummy call entries is crucial; otherwise, this perturbs the state.

In what follows, we assume the actions of the dummy call are implicit and do not analyze them explicitly. Consequently, we will often focus only on the relevant subspace of the database used in the other actions. We prove the following IND-qNCPA bound for LR_4.

Theorem 8.2.1. *Suppose $f_1, f_2, f_3, f_4 : \{0,1\}^n \to \{0,1\}^n$ are three mutually independent uniform random functions. Then, for any $q \geq 0$, and any quantum adversary $\mathscr{A}$ that makes at most q quantum non-adaptive CPA queries, we have*

$$\mathbf{Adv}^{\mathrm{qncpa}}_{LR_4}(\mathscr{A}) \leq 3\sqrt{\frac{q^6}{2^n}} + 6\sqrt{\frac{q^5}{2^n}}.$$

Our goal is to bound the distinguishing advantage for any non-adaptive adversary trying to distinguish LR_4 from a uniform random function. We start with some notional setup.

8.2.2 Notional Setup

Let $F_3, F_4 : \{0,1\}^{3n} \to \{0,1\}^n$ be two uniform random functions. For $i \in \{3,4\}$, define

$$G_i(x_1, x_2, x_1', x_2') := (x_2' \oplus F_i(x_1, x_2, x_1'), x_1'),$$

for any $(x_1, x_2, x_1', x_2') \in \{0,1\}^{4n}$. We define the hybrid random function $\widetilde{\mathsf{LR}}_4$ as (see also Figure 8.2):

$$\widetilde{\mathsf{LR}}_4(x_1, x_2) := G_4(x_1, x_2, G_3(x_1, x_2, \mathsf{LR}_2(x_1, x_2))).$$

Then, it is easy to see that $\widetilde{\mathsf{LR}}_4$ is indistinguishable to a uniform random function $\Gamma \twoheadleftarrow_{\$} \mathsf{Func}(2n; 2n)$. So, it is sufficient to bound the distance between LR_4 and $\widetilde{\mathsf{LR}}_4$.

Let $\mathcal{X} = \{0,1\}^{4+2nq}$, $\mathcal{Y} = \{0,1\}^n$ and $\Gamma : \mathcal{X} \to \mathcal{Y}$ be a uniform random function. For each $x_1, x_2, x_3 \in \{0,1\}^n$, we define

$$\begin{aligned}
f_1(x_1) &:= \Gamma(1001 \| x_1 \| 0^{2nq-n}),\\
f_2(x_1) &:= \Gamma(1010 \| x_1 \| 0^{2nq-n}),\\
f_3(x_1) &:= \Gamma(1011 \| x_1 \| 0^{2nq-n}),\\
f_4(x_1) &:= \Gamma(1100 \| x_1 \| 0^{2nq-n}),\\
F_3(x_1, x_2, x_3) &:= \Gamma(1101 \| x_1 \| x_2 \| x_3 \| 0^{2nq-3n}),\\
F_4(x_1, x_2, x_3) &:= \Gamma(1110 \| x_1 \| x_2 \| x_3 \| 0^{2nq-3n}).
\end{aligned}$$

In addition, we implicitly define the dummy call, denoted dummy, to operate over a disjoint[3] subspace of the database, mapping $2qn$-bit inputs to n-bit outputs. The

[3] Disjoint from the other functions due to the first bit.

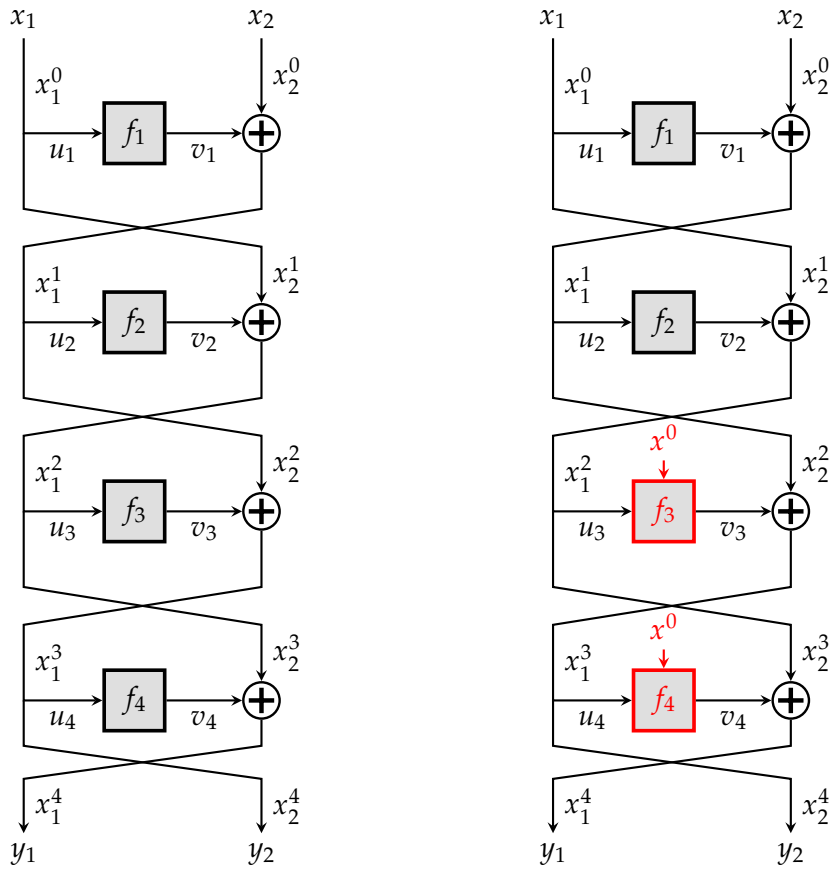

Figure 8.2: *LR_4 (left) vs the hybrid random function, $\widetilde{LR}_4$ (right).*

exact description of the dummy call is not necessary as the output is never used. The distinctness of the first four bits ensures that $f_1, f_2, f_3, f_4, F_3, F_4$ are all independent, and they are independent of dummy by definition.

The database in the real world is denoted $d_{\mathbf{R}}$ (tracking dummy, f_1, f_2, f_3, f_4) and $d_{\mathbf{I}}$ in the ideal world (tracking dummy, f_1, f_2, F_3, F_4). Let $\mathcal{D}_{\mathbf{R}}$ (resp. $\mathcal{D}_{\mathbf{I}}$) be the set of all possible choices for $d_{\mathbf{R}}$ (resp. $d_{\mathbf{I}}$). For some $x = (x_1, x_2, \ldots, x_{2q}) \in \mathcal{Y}^{2q}$, let

$$
\begin{aligned}
[x]_0 &:= 0000\|x, & [x_1]_1 &:= 1001\|x_1\|0^{2nq-n},\\
[x_1, x_2, x_3]_5 &:= 1101\|x_1\|x_2\|x_3\|0^{2nq-3n}, & [x_1]_2 &:= 1010\|x_1\|0^{2nq-n}\\
[x_1, x_2, x_3]_6 &:= 1110\|x_1\|x_2\|x_3\|0^{2nq-3n}, & [x_1]_3 &:= 1011\|x_1\|0^{2nq-n},\\
& & [x_1]_4 &:= 1100\|x_1\|0^{2nq-n}.
\end{aligned}
$$

In addition, for all $k \in [1; q]$, we write $[x_{2k-1}, x_{2k}]_{0\|k}$ to denote the k-th diblock (two-block) coordinate (x_{2k-1}, x_{2k}) of x. We use this notion and view the $2qn$-bit entry as q separate entries of size $2n$-bit each, and thus, $d_{\mathbf{R}}([x_{2k-1}, x_{2k}]_{0\|k}) \neq \perp$ (or $d_{\mathbf{I}}([x_{2k-1}, x_{2k}]_{0\|k}) \neq \perp$) is well-defined as long as $d_{\mathbf{R}}([x]_0) \neq \perp$ (res. $d_{\mathbf{I}}([x]_0) \neq \perp$, for some $x = (z, (x_{2k-1}, x_{2k}), z')$, where z and z' are $2(k-1)n$-bit and $2(q-k)n$-bit strings). Finally, we define the sets

$$
\begin{aligned}
\widetilde{\mathcal{X}}_{\mathbf{R}} &:= \{[x]_0, [x_1]_1, [x_1]_2, [x_1]_3, [x_1]_4 : x = (x_1, \ldots, x_{2q}) \in \mathcal{Y}^{2q}\},\\
\widetilde{\mathcal{X}}_{\mathbf{I}} &:= \{[x]_0, [x_1]_1, [x_1]_2, [x_1, x_2, x_3]_5, [x_1, x_2, x_3]_6 : x = (x_1, \ldots, x_{2q}) \in \mathcal{Y}^{2q}\}.
\end{aligned}
$$

Then, it is easy to see that $\mathcal{D}_{\mathbf{R}} = \mathcal{D}|_{\widetilde{\mathcal{X}}_{\mathbf{R}}}$ and $\mathcal{D}_{\mathbf{I}} = \mathcal{D}|_{\widetilde{\mathcal{X}}_{\mathbf{I}}}$.

8.2.3 Bad Databases Definition

We follow a similar strategy as in Section 7.6 for defining the bad database sets. However, in this case, we can establish a more precise relation, as the information on past queries needed to define a collision is already present in the database, due to the non-adaptive nature of the adversary.

Formally, let $\mathcal{B}_{\mathbf{R}}$ be the set of databases $d_{\mathbf{R}}$ satisfying one of the following condition: we can find $(x_1, x_2) \neq (x_1', x_2') \in \mathcal{Y}^2$ and , $v_1, v_2, v_1', v_2' \in \mathcal{Y}$ such that:

- for some $k \neq k' \in [1; q]$, $d_{\mathbf{R}}([x_1, x_2]_{0\|k}) \neq \perp, d_{\mathbf{R}}([x_1', x_2']_{0\|k'}) \neq \perp$;
- $([x_1]_1, v_1), ([x_1']_1, v_1') \in d_{\mathbf{R}}$;
- $([x_2 \oplus v_1]_2, v_2), ([x_2' \oplus v_1']_2, v_2') \in d_{\mathbf{R}}$;
- $x_1 \oplus v_2 = x_1' \oplus v_2'$;

or we can find $(x_1, x_2) \neq (x_1', x_2') \in \mathcal{Y}^2$ and $v_1, v_2, v_3, v_1', v_2', v_3' \in \mathcal{Y}$ such that

- for some $k \neq k' \in [1; q]$, $d_{\mathbf{R}}([x_1, x_2]_{0\|k}) \neq \perp, d_{\mathbf{R}}([x_1', x_2']_{0\|k'}) \neq \perp$;
- $([x_1]_1, v_1), ([x_1']_1, v_1) \in d_{\mathbf{R}}$;
- $([x_2 \oplus v_1]_2, v_2), ([x_2' \oplus v_1']_2, v_2') \in d_{\mathbf{R}}$;
- $([x_1 \oplus v_2]_3, v_3), ([x_1' \oplus v_2']_3, v_3') \in d_{\mathbf{R}}$;
- $x_2 \oplus v_1 \oplus v_3 = x_2' \oplus v_1' \oplus v_3'$.

Next, let $\mathcal{B}_{\mathbf{I}}$ be the set of databases $d_{\mathbf{I}}$ satisfying one of the the following condition: we can find $(x_1, x_2) \neq (x_1', x_2') \in \mathcal{Y}^2$ and , $v_1, v_2, v_1', v_2' \in \mathcal{Y}$

- for some $k \neq k' \in [1; q]$, $d_{\mathbf{I}}([x_1, x_2]_{0\|k}) \neq \perp, d_{\mathbf{I}}([x_1', x_2']_{0\|k'}) \neq \perp$;
- $([x_1]_1, v_1), ([x_1']_1, v_1') \in d_{\mathbf{I}}$;
- $([x_2 \oplus v_1]_2, v_2), ([x_2' \oplus v_1']_2, v_2') \in d_{\mathbf{I}}$;
- $x_1 \oplus v_2 = x_1' \oplus v_2'$;

or we can find $(x_1, x_2) \neq (x_1', x_2') \in \mathcal{Y}^2$ and $v_1, v_2, v_3, v_1', v_2', v_3' \in \mathcal{Y}$ such that

- for some $k \neq k' \in [1; q]$, $d_{\mathbf{I}}([x_1, x_2]_{0\|k}) \neq \perp, d_{\mathbf{I}}([x_1', x_2']_{0\|k'}) \neq \perp$;
- $([x_1]_1, v_1), ([x_1']_1, v_1') \in d_{\mathbf{I}}$;
- $([x_2 \oplus v_1]_2, v_2), ([x_2' \oplus v_1']_2, v_2') \in d_{\mathbf{I}}$;
- $([x_1, x_2, x_1 \oplus v_2]_5, v_3), ([x_1', x_2', x_1' \oplus v_2']_5, v_3') \in d_{\mathbf{I}}$;
- $x_2 \oplus v_1 \oplus v_3 = x_2' \oplus v_1' \oplus v_3'$;

Let $\mathcal{G}_{\mathbf{R}} := \mathcal{D}_{\mathbf{R}} \setminus \mathcal{B}_{\mathbf{R}}$ and $\mathcal{G}_{\mathbf{I}} := \mathcal{D}_{\mathbf{I}} \setminus \mathcal{B}_{\mathbf{I}}$. Using the definitions above we can infer that, in both $\mathcal{G}_{\mathbf{R}}$ and $\mathcal{G}_{\mathbf{I}}$, each u_3 and u_4 is associated with a unique pair (x_1, x_2). Then it is easy to construct a natural bijection $h : \mathcal{G}_{\mathbf{R}} \longrightarrow \mathcal{G}_{\mathbf{I}}$ that satisfies the following conditions. Namely, for each $d_{\mathbf{R}}$, we define $d_{\mathbf{I}} := h(d_{\mathbf{R}})$ such that:

- for each $x \in \mathcal{Y}^{2q}$, $d_{\mathbf{I}}([x]_0) = d_{\mathbf{R}}([x]_0)$. Note that, by definition of the oracle, there will be only one entry of this type in both the worlds;

- for each $u_1 \in \mathcal{Y}$, $d_{\mathbf{I}}([u_1]_1) = d_{\mathbf{R}}([u_1]_1)$;
- for each $u_2 \in \mathcal{Y}$, $d_{\mathbf{I}}([u_2]_2) = d_{\mathbf{R}}([u_2]_2)$;
- for each $u_3, u_4 \in \mathcal{Y}$ such that $d_{\mathbf{R}}([u_3]_3) \neq \perp$ and $d_{\mathbf{R}}([u_4]_4) \neq \perp$, find the unique $(x_1, x_2) \in \mathcal{Y}^2$, and define $d_{\mathbf{I}}([x_1, x_2, u_3]_5) = d_{\mathbf{R}}([u_3]_3)$ and $d_{\mathbf{I}}([x_1, x_2, u_4]_6) = d_{\mathbf{R}}([u_4]_4)$.

Thus, h satisfies the conditions of Lemma 7.4.4. Consequently to complete the proof of Theorem 8.2.1, it is enough to show that

$$\left(\perp \overset{4q+2}{\rightsquigarrow} \mathcal{B}_{\mathbf{R}}\right) + \left(\perp \overset{4q+2}{\rightsquigarrow} \mathcal{B}_{\mathbf{I}}\right) \leq 2\sqrt{\frac{q^6}{2^n}} + 4\sqrt{\frac{q^5}{2^n}}.$$

8.2.4 Sequence of Actions

From now on we ignore the dummy call actions, as the transition from a good to bad database is independent of this operator. Recall that the q non-adaptive queries can be represented by a single q-fold query to be evaluated sequentially. From now on, we follow the blueprint for proofs as in Section 7.5.

Action of f_1. For $i \in \{4k+2 : 0 \leq k \leq q-1\}$, we bound the the transition capacity $[\![\mathcal{B}^c_{\mathbf{R}[\leq i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\leq i]}]\!]$. For any database $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \leq i-1$ and any $x \in \mathcal{Y}$, we have

$$\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} = \left\{ d_{\mathbf{R}}([x'_1]_1) \oplus d_{\mathbf{R}}([u'_3]_3) \oplus d_{\mathbf{R}}([u_3]_3) \oplus x_2 \oplus x'_2 \mid \mathsf{E}_1 \right\},$$

where E_1 is the condition

$$d_{\mathbf{R}}([u_3]_3) \neq \perp, d_{\mathbf{R}}([u'_3]_3) \neq \perp, d_{\mathbf{R}}([x_1, x_2]_{0\|*}) \neq \perp, d_{\mathbf{R}}([x'_1, x'_2]_{0\|*}) \neq \perp,$$

and $*$ is a symbol representing some element in $[1; q]$. Since there at most q choices for (x'_1, x'_2), at most $\lceil (i-1)/4 \rceil$ choices for each of u_3 and u'_3, and at most q choices for x_2, then we must have $|\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d}| \leq q^2 \lceil (i-1)/3 \rceil^2 \leq q^4$. Thus, by using Lemma 7.4.2, one has

$$[\![\mathcal{B}^c_{\mathbf{R}[\leq i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\leq i]}]\!] \leq \sqrt{\frac{10q^4}{2^n}}, \qquad \forall\, i \in \{4k+2 : 0 \leq k \leq q-1\}. \tag{8.9}$$

By the same arguments we can also show that

$$[\![\mathcal{B}^c_{\mathbf{I}[\leq i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\leq i]}]\!] \leq \sqrt{\frac{10q^4}{2^n}}, \qquad \forall\, i \in \{4k+2 : 0 \leq k \leq q-1\}. \tag{8.10}$$

Action of f_2. Next, consider the transition capacity $[\![\mathcal{B}^c_{\mathbf{R}[\leq i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\leq i]}]\!]$ for $i \in \{4k+3 : 0 \leq k \leq q-1\}$. For any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \leq i-1$ and any $x \in \mathcal{Y}$, we have

$$\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} = \left\{ d_{\mathbf{R}}([u'_2]_2) \oplus x_1 \oplus x'_1 \mid \mathsf{E}_2 \right\},$$

where E_2 is the condition

$$d_{\mathbf{R}}([u'_2]_2) \neq \perp, d_{\mathbf{R}}([x_1, x_2]_{0\|*}) \neq \perp, d_{\mathbf{R}}([x'_1, x'_2]_{0\|*}) \neq \perp.$$

Again, there are at most $\lceil (i-1)/4 \rceil$ choices for u_2' and at most q^2 choices for (x_1, x_1'). Thus, from Lemma 7.4.2, one has

$$[\![\mathcal{B}^c_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] \le \sqrt{\frac{10q^3}{2^n}}, \qquad \forall\, i \in \{4k+3 : 0 \le k \le q-1\}. \tag{8.11}$$

Similarly

$$[\![\mathcal{B}^c_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] \le \sqrt{\frac{10q^3}{2^n}}, \qquad \forall\, i \in \{4k+3 : 0 \le k \le q-1\}. \tag{8.12}$$

Action of f_3 (resp. F_3): For $i \in \{4k : 1 \le k \le q\}$, for any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \le i-1$ and any $x \in \mathcal{Y}$, we have

$$\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} = \left\{ d_{\mathbf{R}}([x_1]_1) \oplus d_{\mathbf{R}}([x_1']_1) \oplus d_{\mathbf{R}}([u_3']_3) \oplus x_2 \oplus x_2' \mid \mathsf{E}_3 \right\},$$

where E_3 is the condition

$$d_{\mathbf{R}}([x_1]_1), d_{\mathbf{R}}([x_1']_1), d_{\mathbf{R}}([u_3]_3) \neq \perp, d_{\mathbf{R}}([x_1, x_2]_{0\|*}) \neq \perp, d_{\mathbf{R}}([x_1', x_2']_{0\|*}) \neq \perp.$$

There are at most $\lceil (i-1)/4 \rceil$ choices for u_3', and at most q^2 choices for $((x_1, x_2), (x_1', x_2'))$. Since the analysis is identical in both the worlds, by using Lemma 7.4.2, we have

$$[\![\mathcal{B}^c_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] \le \sqrt{\frac{10q^3}{2^n}}, \qquad \forall\, i \in \{4k : 1 \le k \le q\}, \tag{8.13}$$

$$[\![\mathcal{B}^c_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] \le \sqrt{\frac{10q^3}{2^n}}, \qquad \forall\, i \in \{4k : 1 \le k \le q\}. \tag{8.14}$$

Action of f_4 (resp. F_4): Since the property $\mathcal{B}_{\mathbf{R}}$ (resp. $\mathcal{B}_{\mathbf{I}}$) is independent of the output of f_4 (resp. F_4) and the database is good right before the action, we have $\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} = \emptyset$. Thus,

$$[\![\mathcal{B}^c_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] = 0, \qquad \forall\, i \in \{4k+1 : 1 \le k \le q\}, \tag{8.15}$$

$$[\![\mathcal{B}^c_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] = 0, \qquad \forall\, i \in \{4k+1 : 1 \le k \le q\}. \tag{8.16}$$

Summing over the $4q+2$ actions using (8.9)-(8.16) gives

$$\left(\perp \overset{4q+2}{\rightsquigarrow} \mathcal{B}_{\mathbf{R}}\right) \le 2\sqrt{\frac{10q^5}{2^n}} + \sqrt{\frac{10q^6}{2^n}}, \qquad \left(\perp \overset{4q+2}{\rightsquigarrow} \mathcal{B}_{\mathbf{I}}\right) \le 2\sqrt{\frac{10q^5}{2^n}} + \sqrt{\frac{10q^6}{2^n}}. \tag{8.17}$$

Adding the two inequalities completes the proof of Theorem 8.2.1. □

8.3 Limitations of the Adaptive Setting

Taking a closer look at the non-adaptive proof shows why it's tricky to get the same result in the adaptive case. The dummy call is used to record all the q non-adaptive queries of the adversary in the database, before $\mathtt{LR}_4$ is applied to each of them sequentially. This

way, the oracle already knows all q queries when it handles each step (f_1, f_2, f_3, etc.), which helps proof the bad norm can be upper bounded by a non trivial value.

The proof relies on identifying any database as "bad" if a collision occurs on the f input in either of the last two rounds—specifically, if there are collisions on $x_1 \oplus v_2$ or $x_2 \oplus v_1 \oplus v_3$ across different database entries. This means that certain later values of x_1 or x_2 can always cause the database to become bad, regardless of the earlier choices for v_1, v_2, or v_3. For instance, recall that in Section 8.2.3 a database is considered bad if:

1. for some $k \neq k \in [1;q]$, $d_{\mathbf{R}}([x_1, x_2]_{0\|k})$, $d_{\mathbf{R}}([x'_1, x'_2]_{0\|k'}) \neq \perp$ (i.e. the adversary has made these two queries).
2. $([x_1]_1, v_1), ([x'_1]_1, v'_1) \in d_{\mathbf{R}}$; ($f_1$ has been evaluated over x_1 and x'_1)
3. $([x_2 \oplus v_1]_2, v_2), ([x'_2 \oplus v'_1]_2, v'_2) \in d_{\mathbf{R}}$; ($f_2$ has been evaluated over $x_2 \oplus v_1$ and $x'_2 \oplus v'_1$)
4. $x_1 \oplus v_2 = x'_1 \oplus v'_2$; (there is an input-collision on f_3)

In the context of f_1's action, comparing the above definition with previous proofs (see the discussion in Section 8.1.2 and especially Claim 8.1.1), it becomes clear that the first and third conditions were missing in earlier proofs. This occurs because the oracle cannot detect the adversary's queries—it only has access to the database entries at any given moment, nothing more. Consequently, the norm bound in those cases is trivial. In contrast, in our case, since condition 1 can always be checked after the dummy call is executed, condition 3 becomes well-defined as well. As a result, as shown in (8.9) and (8.10), the norm bound is non-trivial.

At the same time, the dummy call must be erased before the oracle returns an output to the adversary, or it will perturb the state in a way the adversary could detect. This method only works in non-adaptive scenarios, where the game can be modeled as the adversary making a single "big" query (comprising q regular queries) to the oracle, which responds with a single "big" output (comprising q regular outputs). In contrast, an adaptive game does not follow such simplifications. Specifically, because future values of x_1 and x_2 are controlled by the adversary and unknown to the oracle ahead of time, the amplitude of such events cannot be bounded using known techniques. In the Hosoyamada and Iwata framework, described in Section 8.1.1, this issue manifests as a trivialization of the norm (see Section 8.1.2). In the our framework, this indicates that databases can go bad between two actions, a case not accounted for in the framework. However, in the non-adaptive setting, the oracle knows the future values of x_1 and x_2 in advance, allowing the outputs of f to be classified as 'bad' and bounded at the time of f's action.

Lastly, we note that this issue is not unique to Luby-Rackoff; it is inherent to any proof that defines bad databases based on an input the adversary can choose adaptively. We've identified similar errors in other proofs as well. For instance, the earlier versions of the proofs of `TNT` , `LRQ` and `LRWQ` found in [50], all face this challenge and do not hold in the adaptive setting. Luckily, the issue appears to be largely definitional, as bad events can be described directly in terms of the database entries, albeit possibly leading to slightly

weaker bounds, as presented in Section 7.6 of this thesis. However, for the LRQ proof, this seems to be a more fundamental problem that lacks a straightforward solution. We've observed similar flaws in other works, such as the proof for LRWQ [177] in and the tight security proof for TNT [227]. While the first issue seems fixable, the latter presents a fundamental challenge.

8.4 Post-Quantum Security of The Misty Constructions

On a positive note, another family of Feistel networks, known as the Misty structure, does not suffer from the flaw described in Section 8.1.2. We begin with the description of the Misty constructions for an arbitrary number of rounds.

8.4.1 The Misty Constructions

For some $r \geq 1$ and $f_1, \ldots, f_r : \{0,1\}^n \to \{0,1\}^n$, we define

- $g^L : [1;r] \times \{0,1\}^{2n} \to \{0,1\}^{2n}$ by the mapping:

$$(i, x_1, x_2) \longmapsto (x_2, x_2 \oplus f_i(x_1)),$$

- $g^R : [1;r] \times \{0,1\}^{2n} \to \{0,1\}^{2n}$ by the mapping:

$$(i, x_1, x_2) \longmapsto (x_2 \oplus f_i(x_1), f_i(x_1)),$$

and write $g_i^L(\cdot,\cdot) := g^L(i,\cdot,\cdot)$ and $g_i^R(\cdot,\cdot) := g^R(i,\cdot,\cdot)$.

The r-round MistyL, denoted $\texttt{MistyL}_r$ is defined as:

$$(x_1, x_2) \longmapsto g_r^L \circ \cdots \circ g_1^L(x_1, x_2). \tag{8.18}$$

The MistyR Construction. The r-round MistyR construction, denoted $\texttt{MistyR}_r$ is defined as:

$$(x_1, x_2) \longmapsto g_r^R \circ \cdots \circ g_1^R(x_1, x_2). \tag{8.19}$$

Notations. Throughout this section, we will use the following notations. For all $i \in [1;r]$, we write:

- $x^{i-1} := (x_1^{i-1}, x_2^{i-1})$ to denote the input to g_i, where $x^0 := x = (x_1, x_2)$, denotes the input to $\texttt{Misty}\{\mathsf{L}|\mathsf{R}\}_r$.
- (u_i, v_i) to denote the input-output tuple corresponding to f_i.
- $y = (y_1, y_2) := (x_1^r, x_2^r)$ to denote the output of $\texttt{Misty}\{\mathsf{L}|\mathsf{R}\}_r$.

8.4.2 Post-Quantum Security of $\texttt{MistyR}_4$

In this section, we prove the IND-qCPA security of $\texttt{MistyR}_4$.

Theorem 8.4.1. *Suppose $f_1, f_2, f_3, f_4 : \{0,1\}^n \to \{0,1\}^n$ are four mutually independent uniform random functions. Then, for any $q \geq 0$, and any quantum adversary $\mathscr{A}$ that makes at most q queries, we have*

$$\mathbf{Adv}^{\text{qcpa}}_{\texttt{MistyR}_4}(\mathscr{A}) = O\left(\sqrt{\frac{q^5}{2^n}}\right).$$

We follow the proof blueprint as described in Section 7.5.

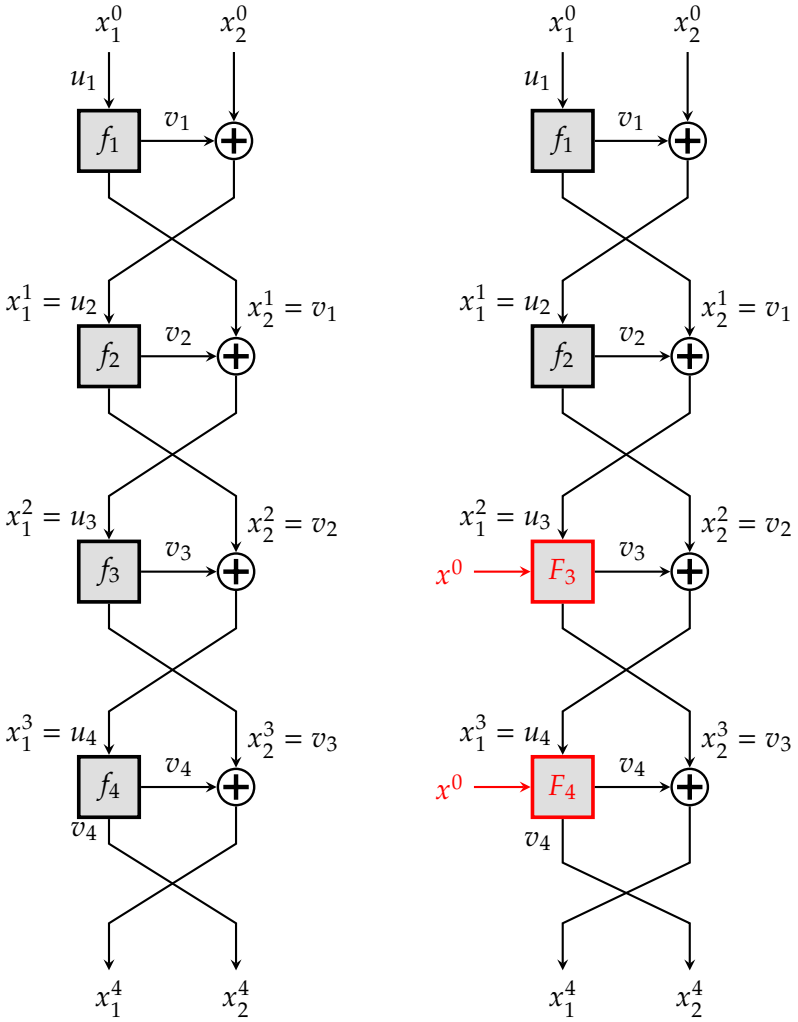

Figure 8.3: $\texttt{MistyR}_4$ *(left) vs the hybrid random function,* $\widetilde{\texttt{MistyR}_4}$ *(right).*

8.4.2.1 Notional Setup

Let $F_3, F_4 : \{0,1\}^{3n} \to \{0,1\}^n$ be two uniform random functions. Define

$$G_3^R(x_1, x_2, x_1', x_2') := (x_2' \oplus F_3(x_1, x_2, x_1'), F_3(x_1, x_2, x_1'))$$
$$G_4^R(x_1, x_2, x_1', x_2') := (x_2' \oplus F_4(x_1, x_2, x_1'), F_4(x_1, x_2, x_1'))$$

for any $(x_1, x_2, x_1', x_2') \in \{0,1\}^{4n}$. We define the hybrid random function $\widetilde{\texttt{MistyR}_4}$ as (see also Figure 8.3):

$$\widetilde{\texttt{MistyR}_4}(x_1, x_2) := G_4^L(x_1, x_2, G_3^L(x_1, x_2, \texttt{MistyR}_2(x_1, x_2))).$$

Therefore, it is easy to see that $\widetilde{\texttt{MistyR}_4}$ is indistinguishable to a uniform random function $\Gamma : \{0,1\}^{2n} \to \{0,1\}^{2n}$. So, it is sufficient to bound the distance between $\texttt{MistyR}_4$ and $\widetilde{\texttt{MistyR}_4}$. Let $\mathcal{X} := \{0,1\}^{3n+3}$, and let $f : \mathcal{X} \longrightarrow \mathcal{Y}$ be a $(3n+3)$-bit-to-n-bit uniform random function. We implement f through **cO** defined over $\mathbb{C}[\mathcal{X}] \otimes \mathbb{C}[\mathcal{Y}] \otimes \mathbb{D}$. For each $x, y, z \in \mathcal{Y}$,

$$\begin{aligned} f_1(x) &= f(000\|x\|0^{2n}) & f_4(x) &= f(011\|x\|0^{2n}) \\ f_2(x) &= f(001\|x\|0^{2n}) & F_3(x,y,z) &= f(100\|x\|y\|z) \\ f_3(x) &= f(010\|x\|0^{2n}) & F_4(x,y,z) &= f(101\|x\|y\|z). \end{aligned}$$

The distinctness of the first three bits ensures that $f_1, f_2, f_3, f_4, F_3, F_4$ are all independent, and they can be implemented by the prefix oracle. This setup allows us to use a single database $d_f : \mathcal{X} \longrightarrow \mathcal{Z}$ to keep track of $f_1, f_1, f_2, f_3, f_4, F_3$ and F_4; we refer to this database as $d_{\mathbf{R}}$ in the real world (tracking f_1, f_2, f_3 and f_4) and $d_{\mathbf{I}}$ in the ideal world (tracking f_1, f_2, F_3 and F_4). Let $\mathcal{D}_{\mathbf{R}}$ (resp. $\mathcal{D}_{\mathbf{I}}$) be the set of all possible choices for $d_{\mathbf{R}}$ (resp. $d_{\mathbf{I}}$). Let

$$\begin{aligned} [x]_1 &:= 000\|x\|0^{2n}, [x]_2 := 001\|x\|0^{2n}, \\ [x]_3 &:= 010\|x\|0^{2n}, [x]_4 := 011\|x\|0^{2n}. \end{aligned}$$

and define the sets

$$\begin{aligned} \widetilde{\mathcal{X}}_{\mathbf{R}} &:= \{[x]_1, [x]_2, [x]_3, [x]_4 \mid x \in \mathcal{Y}\}, \\ \widetilde{\mathcal{X}}_{\mathbf{I}} &:= \{[x]_1, [x]_2, \left(100\|x\|x'\|y\right), \left(101\|x\|x'\|y\right) \mid x, x', y \in \mathcal{Y}\}. \end{aligned}$$

Then, it is easy to see that, $\mathcal{D}_{\mathbf{R}} = \mathcal{D}|_{\widetilde{\mathcal{X}}_{\mathbf{R}}}$ and $\mathcal{D}_{\mathbf{I}} = \mathcal{D}|_{\widetilde{\mathcal{X}}_{\mathbf{I}}}$.

8.4.2.2 Bad Databases

Let $\mathcal{B}_{\mathbf{R}}$ be the set of databases $d_{\mathbf{R}}$ satisfying one of the two following conditions: we can find $u_1, u_1', u_2, u_2', v_1, v_1', v_2, v_2' \in \mathcal{Y}$ such that

1. $([u_1]_1, v_1), ([u_1']_1, v_1'), ([u_2]_2, v_2), ([u_2']_2, v_2') \in d_{\mathbf{R}}$;
2. $v_2 \oplus v_1 = v_2' \oplus v_1'$;

or we can find $u_1, u_1', u_2, u_2', v_1, v_1', v_2, v_2', v_3, v_3' \in \mathcal{Y}$ such that

1. $([u_1]_1, v_1), ([u_1']_1, v_1'), ([u_2]_2, v_2), ([u_2']_2, v_2')$, $[v_2 \oplus v_1]_3, v_3), ([v_2' \oplus v_1']_3, v_3') \in d_{\mathbf{R}}$;
2. $v_3 \oplus v_2 = v_3' \oplus v_2'$;

Next, let $\mathcal{B}_{\mathbf{I}}$ be the set of databases $d_{\mathbf{I}}$ satisfying one of the two following conditions: we can find $u_1, u_1', u_2, u_2', v_1, v_1', v_2, v_2' \in \mathcal{Y}$ such that

1. $([u_1]_1, v_1), ([u_1']_1, v_1'), ([u_2]_2, v_2), ([u_2']_2, v_2') \in d_{\mathbf{I}}$;
2. $v_2 \oplus v_1 = v_2' \oplus v_1'$;

or we can find $u_1, u_1', u_2, u_2', v_1, v_1', v_2, v_2', v_3, v_3' \in \mathcal{Y}$ such that

1. $([u_1]_1, v_1), ([u_1']_1, v_1'), ([u_2]_2, v_2), ([u_2']_2, v_2')$, $(100\|u_1\|v_1 \oplus u_2\|v_2 \oplus v_1, v_3), (100\|u_1'\|v_1' \oplus u_2'\|v_2' \oplus v_1', v_3') \in d_{\mathbf{I}}$;
2. $v_3 \oplus v_2 = v_3' \oplus v_2'$;

Let $\mathcal{G}_{\mathbf{R}} := \mathcal{D}_{\mathbf{R}} \setminus \mathcal{B}_{\mathbf{R}}$ and $\mathcal{G}_{\mathbf{I}} := \mathcal{D}_{\mathbf{I}} \setminus \mathcal{B}_{\mathbf{I}}$. Suppose $d_{\mathbf{R}} \in \mathcal{G}_{\mathbf{R}}$ and $d_{\mathbf{I}} \in \mathcal{G}_{\mathbf{I}}$. Then each u_3 for which there exists v_3 such that $([u_3]_3, v_3) \in d_{\mathbf{R}}$ is associated with a unique pair $([u_1]_1, v_1), ([u_2]_2, v_2) \in d_{\mathbf{R}}$ such that $u_3 = v_1 \oplus v_2$, and each u_4 for which there exists v_4 such that $([u_4]_4, v_4) \in d_{\mathbf{R}}$ is associated with a unique triple $([u_1]_1, v_1), ([u_2]_2, v_2), ([u_3]_3, v_3) \in d_{\mathbf{R}}$ such that $u_3 = v_1 \oplus v_2$ and $u_4 = v_2 \oplus v_3$. Thus, we can define the bijection $h : \mathcal{G}_{\mathbf{R}} \longrightarrow \mathcal{G}_{\mathbf{I}}$ as follows: for each $d_{\mathbf{R}}$ we define $d_{\mathbf{I}} := h(d_{\mathbf{R}})$ such that

- for each $u_1 \in \mathcal{Y}$, $d_{\mathbf{I}}([u_1]_1) = d_{\mathbf{R}}([u_1]_1)$;
- for each $u_2 \in \mathcal{Y}$, $d_{\mathbf{I}}([u_2]_2) = d_{\mathbf{R}}([u_2]_2)$;
- for each $x_1, x_2 \in \mathcal{Y}$ and the associated (u_3, u_4),

$$d_{\mathbf{I}}(100\|x_1\|x_2\|u_3) = d_{\mathbf{R}}([u_3]_3), \qquad d_{\mathbf{I}}(101\|x_1\|x_2\|u_4) = d_{\mathbf{R}}([u_4]_4).$$

Hence, h satisfies the conditions of Lemma 7.4.4. To complete the proof of Theorem 8.4.1, we show that

$$\left(\perp \overset{4q}{\rightsquigarrow} \mathcal{B}_{\mathbf{R}}\right) + \left(\perp \overset{4q}{\rightsquigarrow} \mathcal{B}_{\mathbf{I}}\right) \leq (4 + 2\sqrt{2})\sqrt{10q^5/2^n}.$$

8.4.2.3 Sequence of Actions

The prefix oracle is defined so that each query made by the adversary to the oracle triggers a sequence of four queries to the function f: one each to f_1, f_2, and f_3 followed by f_4 in the real world, or F_3 followed by F_4 in the ideal world, in that specific order. We conceptualize the query-response phase as a sequence of $4q$ (potentially duplicate) *actions* and proceed to analyze the transition capacity at each action.

Action of f_1: For $i \in \{4k + 1 : 0 \leq k \leq q - 1\}$, we first look at the transition capacity $[\![\mathcal{B}^c_{\mathbf{R}[\leq i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\leq i]}]\!]$. For any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \leq i - 1$ and any $x \in \mathcal{Y}$, we have

$$\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} = \{d_{\mathbf{R}}([u_1]_1) \oplus d_{\mathbf{R}}([u_2]_2) \oplus d_{\mathbf{R}}([u_2']_2) : \mathsf{E}_1\},$$

where the condition E_1 is defined as

$$d_{\mathbf{R}}([u_1]_1) \neq \perp, d_{\mathbf{R}}([u_2]_2) \neq \perp, d_{\mathbf{R}}([u_2']_2) \neq \perp.$$

Note that, there are at most $\lceil (i-1)/4 \rceil^3$ choices for the triple (u_2, u_1', u_2'), so $|\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d}| \leq \lceil (i-1)/4 \rceil^3 \leq q^3$, and by using Lemma 7.4.2, one has

$$[\![\mathcal{B}^c_{\mathbf{R}[\leq i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\leq i]}]\!] \leq \sqrt{\frac{10q^3}{2^n}}, \qquad \forall\, i \in \{4k + 1 : 0 \leq k \leq q - 1\}. \tag{8.20}$$

By the same arguments we can also show that

$$[\![\mathcal{B}^c_{\mathbf{I}[\leq i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\leq i]}]\!] \leq \sqrt{\frac{10q^3}{2^n}}, \qquad \forall\, i \in \{4k + 1 : 0 \leq k \leq q - 1\}. \tag{8.21}$$

Action of f_2: Next, we look at the transition capacity $[\![\mathcal{B}^c_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!]$ for $i \in \{4k+2 : 0 \le k \le q-1\}$. For any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \le i-1$ and any $x \in \mathcal{Y}$, we have

$$\begin{aligned}\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} &= \{d_{\mathbf{R}}([u_1]_1) \oplus d_{\mathbf{R}}([u'_1]_1) \oplus d_{\mathbf{R}}([u'_2]_2) \mid \mathsf{E}_{2,1}\} \\ &\cup \{d_{\mathbf{R}}([u_3]_3) \oplus d_{\mathbf{R}}([u'_3]_3) \oplus d_{\mathbf{R}}([u'_2]_2) \mid \mathsf{E}_{2,2}\}\end{aligned}$$

where the conditions $\mathsf{E}_{2,1}$ and $\mathsf{E}_{2,2}$ are defined as

$$\begin{aligned}\mathsf{E}_{2,1} &: d_{\mathbf{R}}([u_1]_1) \ne \perp, d_{\mathbf{R}}([u'_1]_1) \ne \perp, d_{\mathbf{R}}([u'_2]_2) \ne \perp, \\ \mathsf{E}_{2,2} &: d_{\mathbf{R}}([u_3]_3) \ne \perp, d_{\mathbf{R}}([u'_3]_3) \ne \perp, d_{\mathbf{R}}([u'_2]_2) \ne \perp.\end{aligned}$$

Again, there are at most $\lceil (i-1)/4 \rceil^3$ choices for each of the triples (u_2, u'_1, u'_2) and (u_3, u'_2, u'_3), and similarly as before we have

$$[\![\mathcal{B}^c_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] \le \sqrt{\frac{20q^3}{2^n}}, \qquad \forall\, i \in \{4k+2 : 0 \le k \le q-1\}. \tag{8.22}$$

By the same arguments we can also show that

$$[\![\mathcal{B}^c_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] \le \sqrt{\frac{20q^3}{2^n}}, \qquad \forall\, i \in \{4k+2 : 0 \le k \le q-1\}. \tag{8.23}$$

Action of f_3 (resp. F_3): Next, we look at the transition capacity $[\![\mathcal{B}^c_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!]$ for $i \in \{4k+3 : 0 \le k \le q-1\}$. For any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \le i-1$ and any $x \in \mathcal{Y}$, we have

$$\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} = \{d_{\mathbf{R}}([u_2]_2) \oplus d_{\mathbf{R}}([u'_2]_2) \oplus d_{\mathbf{R}}([u'_3]_3) \mid \mathsf{E}_3\},$$

where the condition E_3 is defined as

$$d_{\mathbf{R}}([u_2]_2) \ne \perp, d_{\mathbf{R}}([u'_2]_2) \ne \perp, d_{\mathbf{R}}([u'_3]_3) \ne \perp.$$

Again, there are at most $\lceil (i-1)/4 \rceil^3$ choices for the pair (u_2, u'_2, u'_3), and similarly as before we have

$$[\![\mathcal{B}^c_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] \le \sqrt{\frac{10q^3}{2^n}}, \qquad \forall\, i \in \{4k+3 : 0 \le k \le q-1\}. \tag{8.24}$$

By the same arguments we can also show that

$$[\![\mathcal{B}^c_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] \le \sqrt{\frac{10q^3}{2^n}}, \qquad \forall\, i \in \{4k+3 : 0 \le k \le q-1\}. \tag{8.25}$$

Action of f_4 (resp. F_4): Finally, for $i \in \{4k : 1 \le k \le q\}$, for any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \le i-1$ (resp. any $d_{\mathbf{I}}$ with $|d_{\mathbf{I}}| \le i-1$) and any $x \in \mathcal{Y}$, since the property $\mathcal{B}_{\mathbf{R}}$ (resp. $\mathcal{B}_{\mathbf{I}}$) does not depend on $d_{\mathbf{R}}([x]_4)$ (resp. $d_{\mathbf{I}}(101\|x_1\|x_2\|x)$), we have $\mathcal{S}^{\mathcal{B}^c_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} = \emptyset$ (resp. $\mathcal{S}^{\mathcal{B}^c_{\mathbf{I}} \hookrightarrow \mathcal{B}_{\mathbf{I}}}_{x,d} = \emptyset$). Thus,

$$[\![\mathcal{B}^c_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] = 0, \qquad \forall\, i \in \{4k : 1 \le k \le q\}, \tag{8.26}$$

and also,

$$[\![\mathcal{B}^c_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] = 0, \qquad \forall\, i \in \{4k : 1 \le k \le q\}. \tag{8.27}$$

Summing over the $4q$ actions using (8.20)-(8.27) gives

$$\left(\perp \overset{4q}{\rightsquigarrow} \mathcal{B}_{\mathbf{R}}\right) \le (2+\sqrt{2})\sqrt{\frac{10q^5}{2^n}}, \qquad \left(\perp \overset{4q}{\rightsquigarrow} \mathcal{B}_{\mathbf{I}}\right) \le (2+\sqrt{2})\sqrt{\frac{10q^5}{2^n}}. \tag{8.28}$$

Adding the two inequalities completes the proof of Theorem 8.4.1. □

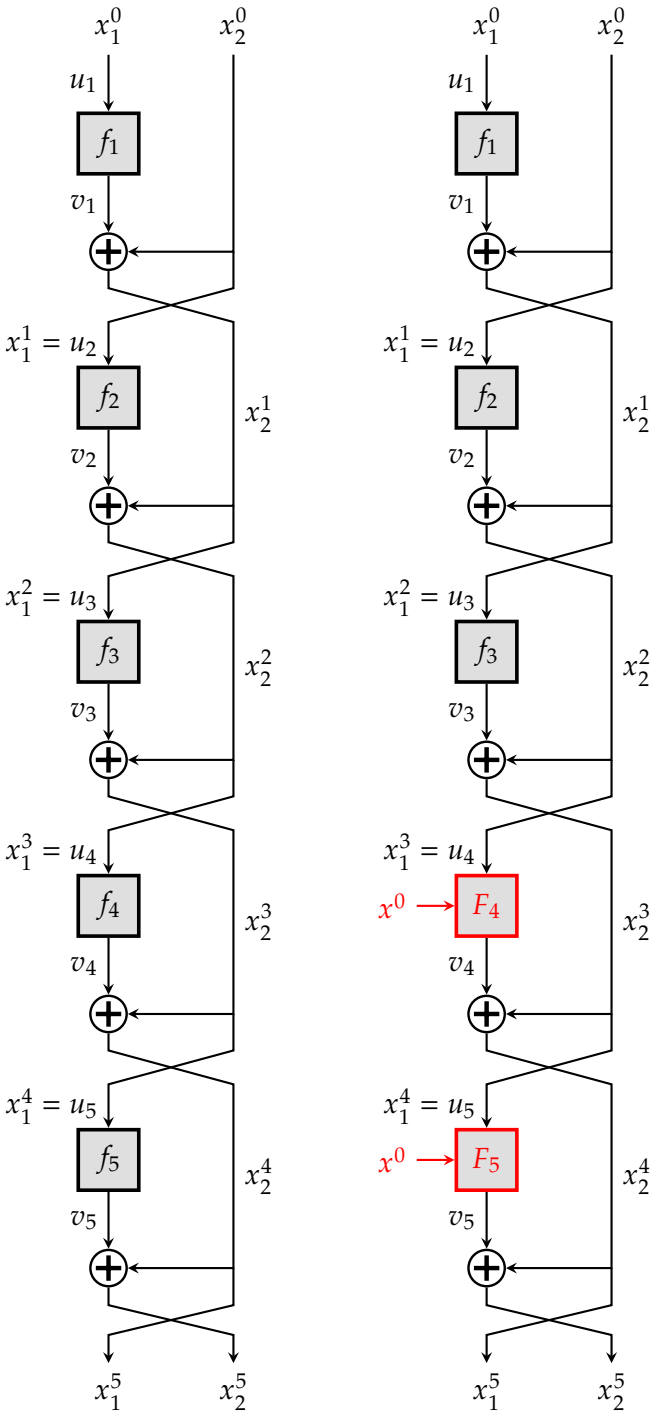

Figure 8.4: $\texttt{MistyL}_5$ *(left) vs the hybrid random function,* $\widetilde{\texttt{MistyL}_5}$ *(right).*

8.4.3 Post-Quantum Security of $\texttt{MistyL}_5$

In this section, we prove the IND-qCPA security of $\texttt{MistyL}_5$.

Theorem 8.4.2. *Suppose* $f_1, f_2, f_3, f_4, f_5 : \{0,1\}^n \to \{0,1\}^n$ *are five mutually independent uniform random functions. Then, for any* $q \geq 0$*, and any quantum adversary* $\mathscr{A}$ *that makes at*

most q queries, we have

$$\mathbf{Adv}^{\text{qcpa}}_{\texttt{MistyL}_5}(\mathscr{A}) = O\left(\sqrt{\frac{q^7}{2^n}}\right).$$

We follow the proof blueprint as described in Section 7.5.

8.4.3.1 Notional Setup

Let $F_4, F_5 : \{0,1\}^{3n} \to \{0,1\}^n$ be two uniform random functions. Define

$$G^L_4(x_1, x_2, x'_1, x'_2) := (x'_2, x'_2 \oplus F_4(x_1, x_2, x'_1)),$$
$$G^L_5(x_1, x_2, x'_1, x'_2) := (x'_2, x'_2 \oplus F_5(x_1, x_2, x'_1)),$$

for any $(x_1, x_2, x'_1, x'_2) \in \{0,1\}^{4n}$. We define the hybrid random function $\widetilde{\texttt{MistyL}_5}$ as (see also Figure 8.4):

$$\widetilde{\texttt{MistyL}_5}(x_1, x_2) := G^L_5(x_1, x_2, G^L_4(x_1, x_2, \texttt{MistyL}_3(x_1, x_2))).$$

Then, it is easy to see that $\widetilde{\texttt{MistyL}_5}$ is indistinguishable to a uniform random function $\Gamma : \{0,1\}^{2n} \to \{0,1\}^{2n}$. So, it is sufficient to bound the distance between $\texttt{MistyL}_5$ and $\widetilde{\texttt{MistyL}_5}$. Let $\mathcal{X} := \{0,1\}^{3n+3}$, and let $f : \mathcal{X} \longrightarrow \mathcal{Y}$ be a $(3n+3)$-bit-to-n-bit uniform random function. We implement f through $\mathbf{cO}$ defined over $\mathbb{C}[\mathcal{X}] \otimes \mathbb{C}[\mathcal{Y}] \otimes \mathbb{D}$. For each $x, y, z \in \mathcal{Y}$,

$$\begin{aligned}
f_1(x) &= f(000\|x\|0^{2n}),\\
f_2(x) &= f(001\|x\|0^{2n}),\\
f_3(x) &= f(010\|x\|0^{2n}),\\
f_4(x) &= f(011\|x\|0^{2n}),\\
f_5(x) &= f(100\|x\|0^{2n}),\\
F_4(x,y,z) &= f(101\|x\|y\|z),\\
F_5(x,y,z) &= f(110\|x\|y\|z).
\end{aligned}$$

The distinctness of the first three bits ensures that $f_1, f_2, f_3, f_4, f_5, F_4, F_5$ are all independent, and they can be implemented by the prefix oracle. This setup allows us to use a single database $d_f : \mathcal{X} \longrightarrow \mathcal{Z}$ to keep track of f_1, $f_1, f_2, f_3, f_4, f_5F_4$ and F_5; we refer to this database as $d_{\mathbf{R}}$ in the real world (tracking f_1, f_2, f_3, f_4 and f_5) and $d_{\mathbf{I}}$ in the ideal world (tracking f_1, f_2, f_3, F_4 and F_5). Let $\mathcal{D}_{\mathbf{R}}$ (resp. $\mathcal{D}_{\mathbf{I}}$) be the set of all possible choices for $d_{\mathbf{R}}$ (resp. $d_{\mathbf{I}}$). Let

$$\begin{aligned}
[x]_1 &:= 000\|x\|0^{2n}, [x]_2 := 001\|x\|0^{2n},\\
[x]_3 &:= 010\|x\|0^{2n}, [x]_4 := 011\|x\|0^{2n},\\
[x]_5 &:= 100\|x\|0^{2n},
\end{aligned}$$

and define the sets

$$\widetilde{\mathcal{X}}_{\mathbf{R}} := \{[x]_1, [x]_2, [x]_3, [x]_4, [x]_5 \mid x \in \mathcal{Y}\},$$

$$\widetilde{\mathcal{X}}_{\mathbf{I}} := \{[x]_1, [x]_2, [x]_3 \left(101\|x\|x'\|y\right), \left(110\|x\|x'\|y\right) \mid x, x', y \in \mathcal{Y}\}.$$

Then it is easy to see that $\mathcal{D}_{\mathbf{R}} = \mathcal{D}|_{\widetilde{\mathcal{X}}_{\mathbf{R}}}$ and $\mathcal{D}_{\mathbf{I}} = \mathcal{D}|_{\widetilde{\mathcal{X}}_{\mathbf{I}}}$.

8.4.3.2 Bad Databases

Let $\mathcal{B}_{\mathbf{R}}$ be the set of databases $d_{\mathbf{R}}$ satisfying one of the two following conditions: we can find $u_1, u_1', u_2, u_2', v_1, v_1', v_2, v_2' \in \mathcal{Y}$ such that

1. $([u_1]_1, v_1), ([u_1']_1, v_1'), ([u_2]_2, v_2), ([u_2']_2, v_2') \in d_{\mathbf{R}}$;
2. $v_2 \oplus v_1 \oplus u_2 = v_2' \oplus v_1' \oplus u_2'$;

or we can find $u_1, u_1', u_2, u_2', v_1, v_1', v_2, v_2', v_3, v_3' \in \mathcal{Y}$ such that

1. $([u_1]_1, v_1), ([u_1']_1, v_1'), ([u_2]_2, v_2), ([u_2']_2, v_2')$,
   $([v_1 \oplus u_2]_3, v_3), ([v_1' \oplus u_2']_3, v_3') \in d_{\mathbf{R}}$;
2. $v_3 \oplus v_2 \oplus v_1 \oplus u_2 = v_3' \oplus v_2' \oplus v_1' \oplus u_2$;

Next, let $\mathcal{B}_{\mathbf{I}}$ be the set of databases $d_{\mathbf{I}}$ satisfying one of the two following conditions: we can find $u_1, u_1', u_2, u_2', v_1, v_1', v_2, v_2' \in \mathcal{Y}$ such that

1. $([u_1]_1, v_1), ([u_1']_1, v_1'), ([u_2]_2, v_2), ([u_2']_2, v_2') \in d_{\mathbf{I}}$;
2. $v_2 \oplus v_1 \oplus u_2 = v_2' \oplus v_1' \oplus u_2'$;

or we can find $u_1, u_1', u_2, u_2', v_1, v_1', v_2, v_2', v_3, v_3' \in \mathcal{Y}$ such that

1. $([u_1]_1, v_1), ([u_1']_1, v_1'), ([u_2]_2, v_2), ([u_2']_2, v_2')$,
   $([v_1 \oplus u_2]_3, v_3), ([v_1' \oplus u_2']_3, v_3') \in d_{\mathbf{I}}$;
2. $v_3 \oplus v_2 \oplus v_1 \oplus u_2 = v_3' \oplus v_2' \oplus v_1' \oplus u_2$;

Let $\mathcal{G}_{\mathbf{R}} := \mathcal{D}_{\mathbf{R}} \setminus \mathcal{B}_{\mathbf{R}}$ and $\mathcal{G}_{\mathbf{I}} := \mathcal{D}_{\mathbf{I}} \setminus \mathcal{B}_{\mathbf{I}}$. Thus the above definitions mean that in both $\mathcal{G}_{\mathbf{R}}$ and $\mathcal{G}_{\mathbf{I}}$, each pair of values $(u_4 := v_2 \oplus v_1 \oplus u_2, u_5 := v_3 \oplus v_2 \oplus v_1 \oplus u_2)$ is associated with a unique pair (x_1, x_2). Then we can define the bijection $h : \mathcal{G}_{\mathbf{R}} \longrightarrow \mathcal{G}_{\mathbf{I}}$ as follows: for each $d_{\mathbf{R}}$ we define $d_{\mathbf{I}} := h(d_{\mathbf{R}})$ such that

- for each $x_L \in \mathcal{Y}$, $d_{\mathbf{I}}([u_1]_1) = d_{\mathbf{R}}([u_1]_1)$;
- for each $u_2 \in \mathcal{Y}$, $d_{\mathbf{I}}([u_2]_2) = d_{\mathbf{R}}([u_2]_2)$;
- for each $u_3 \in \mathcal{Y}$, $d_{\mathbf{I}}([u_3]_3) = d_{\mathbf{R}}([u_3]_3)$;
- for each $x_1, x_2 \in \mathcal{Y}$ and the associated (u_4, u_5),

$$d_{\mathbf{I}}(101\|x_1\|x_2\|u_4) = d_{\mathbf{R}}([u_4]_4), \qquad d_{\mathbf{I}}(110\|x_1\|x_2\|u_5) = d_{\mathbf{R}}([u_5]_5.$$

Hence, h satisfies the conditions of Lemma 7.4.4. To complete the proof of Theorem 8.4.2, we show that

$$\left(\perp \overset{5q}{\rightsquigarrow} \mathcal{B}_{\mathbf{R}}\right) + \left(\perp \overset{5q}{\rightsquigarrow} \mathcal{B}_{\mathbf{I}}\right) \leq (2 + 4\sqrt{2})\sqrt{\frac{10q^5}{2^n}}.$$

8.4.3.3 Sequence of Actions

Similarly, each query made by the adversary to the oracle triggers a sequence of four queries to the function f: one each to f_1, f_2, f_3, and f_4 followed by f_5 in the real world, or F_4 followed by F_5 in the ideal world, in that specific order. We conceptualize the query-response phase as a sequence of $5q$ (potentially duplicate) *actions* and proceed to analyze the transition capacity at each action.

Action of f_1: For $i \in \{5k+1 : 0 \le k \le q-1\}$, we first look at the transition capacity $[\![\mathcal{B}^{c}_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!]$. Note that any two consecutive rounds of MistyL are independent (can be executed in parallel). So, without loss of generality, we assume that f_2 is applied first followed by f_1. Hence, for any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \le i-1$ and any $x \in \mathcal{Y}$, we have

$$\begin{aligned}\mathcal{S}^{\mathcal{B}^{c}_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} &= \{u_2 \oplus u_2' \oplus d_{\mathbf{R}}([u_1]_1) \oplus d_{\mathbf{R}}([u_2]_2) \oplus d_{\mathbf{R}}([u_2']_2) \mid \mathsf{E}_{1,1}\} \\ &\cup \{u_2 \oplus u_2' \oplus d_{\mathbf{R}}([u_1]_1) \oplus d_{\mathbf{R}}([u_2]_2) \oplus d_{\mathbf{R}}([u_2']_2) \oplus d_{\mathbf{R}}([u_3]_3) \oplus d_{\mathbf{R}}([u_3']_3) \mid \mathsf{E}_{1,2}\},\end{aligned}$$

where the conditions $\mathsf{E}_{1,1}$ and $\mathsf{E}_{1,2}$ are defined as

$$\begin{aligned}\mathsf{E}_{1,1} &: d_{\mathbf{R}}([u_1]_1) \ne \perp, d_{\mathbf{R}}([u_2]_2) \ne \perp, d_{\mathbf{R}}([u_2']_2) \ne \perp, \\ \mathsf{E}_{1,2} &: d_{\mathbf{R}}([u_1]_1) \ne \perp, d_{\mathbf{R}}([u_2]_2) \ne \perp, d_{\mathbf{R}}([u_2']_2) \ne \perp, d_{\mathbf{R}}([u_3]_3) \ne \perp, d_{\mathbf{R}}([u_3']_3) \ne \perp.\end{aligned}$$

There are respectively at most $\lceil (i-1)/5 \rceil^3$ and $\lceil (i-1)/5 \rceil^5$ choices for the tuples (u_1, u_2, u_2') and $(u_1, u_2, u_2', u_3, u_3')$, so $|\mathcal{S}^{\mathcal{B}^{c}_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d}| \le 2\lceil (i-1)/5 \rceil^5 \le 2q^5$, and by using Lemma 7.4.2 we have

$$[\![\mathcal{B}^{c}_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] \le \sqrt{\frac{20q^5}{2^n}}, \qquad \forall\, i \in \{5k+1 : 0 \le k \le q-1\}. \tag{8.29}$$

By the same arguments we can also show that

$$[\![\mathcal{B}^{c}_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] \le \sqrt{\frac{20q^5}{2^n}}, \qquad \forall\, i \in \{5k+1 : 0 \le k \le q-1\}. \tag{8.30}$$

Action of f_2: For $i \in \{5k+2 : 1 \le k \le q\}$, for any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \le i-1$ (resp. any $d_{\mathbf{I}}$ with $|d_{\mathbf{I}}| \le i-1$) and any $x \in \mathcal{Y}$, we have

$$\begin{aligned}\mathcal{S}^{\mathcal{B}^{c}_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} &= \{x \oplus u_2 \oplus d_{\mathbf{R}}([u_1]_1) \oplus d_{\mathbf{R}}([u_1']_1) \oplus d_{\mathbf{R}}([u_2]_2) \mid \mathsf{E}_{2,1}\} \\ &\cup \{x \oplus u_2 \oplus d_{\mathbf{R}}([u_1]_1) \oplus d_{\mathbf{R}}([u_1']_1) \oplus d_{\mathbf{R}}([u_2]_2) \oplus d_{\mathbf{R}}([u_3]_3) \oplus d_{\mathbf{R}}([u_3']_3) \mid \mathsf{E}_{2,2}\},\end{aligned}$$

where the conditions $\mathsf{E}_{2,1}$ and $\mathsf{E}_{2,2}$ are defined as

$$\begin{aligned}\mathsf{E}_{2,1} &: d_{\mathbf{R}}([u_1]_1) \ne \perp, d_{\mathbf{R}}([u_1']_1) \ne \perp, d_{\mathbf{R}}([u_2]_2) \ne \perp, \\ \mathsf{E}_{2,2} &: d_{\mathbf{R}}([u_1]_1) \ne \perp, d_{\mathbf{R}}([u_1']_1) \ne \perp, d_{\mathbf{R}}([u_2]_2) \ne \perp, d_{\mathbf{R}}([u_3]_3) \ne \perp, d_{\mathbf{R}}([u_3']_3) \ne \perp\end{aligned}$$

There are respectively at most $\lceil (i-1)/5 \rceil^3$ and $\lceil (i-1)/5 \rceil^5$ choices for the tuples (u_1, u_2, u_1') and $(u_1, u_1', u_2, u_3, u_3')$, so $|\mathcal{S}^{\mathcal{B}^{c}_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d}| \le 2q^5$, and by using Lemma 7.4.2 we have

$$[\![\mathcal{B}^{c}_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] = \sqrt{\frac{20q^5}{2^n}}, \qquad \forall\, i \in \{5k+2 : 1 \le k \le q\}, \tag{8.31}$$

and also,

$$[\![\mathcal{B}^{c}_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] = \sqrt{\frac{20q^5}{2^n}}, \qquad \forall\, i \in \{5k+2 : 1 \le k \le q\}. \tag{8.32}$$

Action of f_3: Next, we look at the transition capacity $[\![\mathcal{B}^{c}_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!]$ for $i \in \{5k+3 : 0 \le k \le q-1\}$. For any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \le i-1$ and any $x \in \mathcal{Y}$, we have

$$\mathcal{S}^{\mathcal{B}^{c}_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} = \{u_2 \oplus u_2' \oplus d_{\mathbf{R}}([u_1]_1) \oplus d_{\mathbf{R}}([u_1']_1) \oplus d_{\mathbf{R}}([u_2]_2) \oplus d_{\mathbf{R}}([u_2]_2) \oplus d_{\mathbf{R}}([u_3]_3) | \mathsf{E}_3\},$$

where the condition E_3 is defined as

$$d_{\mathbf{R}}([u_1]_1) \neq \perp, d_{\mathbf{R}}([u_1']_1) \neq \perp, d_{\mathbf{R}}([u_2]_2) \neq \perp, d_{\mathbf{R}}([u_2']_2) \neq \perp, d_{\mathbf{R}}([u_3]_3) \neq \perp.$$

There are at most $\lceil (i-1)/5 \rceil^5$ choices for the tuple $(u_1, u_1', u_2, u_2', u_3)$, so $|\mathcal{S}^{\mathcal{B}^{c}_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d}| \le \lceil (i-1)/5 \rceil^5 \le q^5$, and by using Lemma 7.4.2 we have

$$[\![\mathcal{B}^{c}_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] \le \sqrt{\frac{10q^5}{2^n}}, \qquad \forall\, i \in \{5k+3 : 0 \le k \le q-1\}. \tag{8.33}$$

By the same arguments we can also show that

$$[\![\mathcal{B}^{c}_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] \le \sqrt{\frac{10q^5}{2^n}}, \qquad \forall\, i \in \{5k+3 : 0 \le k \le q-1\}. \tag{8.34}$$

Action of f_4 (resp. F_4): Finally, for $i \in \{5k : 1 \le k \le q\}$, for any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \le i-1$ (resp. any $d_{\mathbf{I}}$ with $|d_{\mathbf{I}}| \le i-1$) and any $x \in \mathcal{Y}$, since the property $\mathcal{B}_{\mathbf{R}}$ (resp. $\mathcal{B}_{\mathbf{I}}$) does not depend on $d_{\mathbf{R}}([x]_4)$ (resp. $d_{\mathbf{I}}(101\|x_1\|x_2\|x)$), we have $\mathcal{S}^{\mathcal{B}^{c}_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} = \emptyset$ (resp. $\mathcal{S}^{\mathcal{B}^{c}_{\mathbf{I}} \hookrightarrow \mathcal{B}_{\mathbf{I}}}_{x,d} = \emptyset$). Thus,

$$[\![\mathcal{B}^{c}_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] = 0, \qquad \forall\, i \in \{5k+4 : 0 \le k \le q-1\}, \tag{8.35}$$

and also,

$$[\![\mathcal{B}^{c}_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] = 0, \qquad \forall\, i \in \{5k+4 : 0 \le k \le q-1\}. \tag{8.36}$$

Action of f_5 (resp. F_5): Finally, for $i \in \{5k : 1 \le k \le q\}$, for any $d_{\mathbf{R}}$ with $|d_{\mathbf{R}}| \le i-1$ (resp. any $d_{\mathbf{I}}$ with $|d_{\mathbf{I}}| \le i-1$) and any $x \in \mathcal{Y}$, since the property $\mathcal{B}_{\mathbf{R}}$ (resp. $\mathcal{B}_{\mathbf{I}}$) does not depend on $d_{\mathbf{R}}([x]_5)$ (resp. $d_{\mathbf{I}}(110\|x_1\|x_2\|x)$), we have $\mathcal{S}^{\mathcal{B}^{c}_{\mathbf{R}} \hookrightarrow \mathcal{B}_{\mathbf{R}}}_{x,d} = \emptyset$ (resp. $\mathcal{S}^{\mathcal{B}^{c}_{\mathbf{I}} \hookrightarrow \mathcal{B}_{\mathbf{I}}}_{x,d} = \emptyset$). Thus,

$$[\![\mathcal{B}^{c}_{\mathbf{R}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{R}[\le i]}]\!] = 0, \qquad \forall\, i \in \{5k : 1 \le k \le q\}, \tag{8.37}$$

and also,

$$[\![\mathcal{B}^{c}_{\mathbf{I}[\le i-1]} \hookrightarrow \mathcal{B}_{\mathbf{I}[\le i]}]\!] = 0, \qquad \forall\, i \in \{5k : 1 \le k \le q\}. \tag{8.38}$$

Summing over the $5q$ actions using (8.29)-(8.38) gives

$$\left(\perp \overset{5q}{\rightsquigarrow} \mathcal{B}_{\mathbf{R}}\right) \le (1+2\sqrt{2})\sqrt{\frac{10q^7}{2^n}}, \qquad \left(\perp \overset{5q}{\rightsquigarrow} \mathcal{B}_{\mathbf{I}}\right) \le (1+2\sqrt{2})\sqrt{\frac{10q^7}{2^n}}. \tag{8.39}$$

Adding the two inequalities completes the proof of Theorem 8.4.2. □

Part IV

Conclusion

9

Conclusions and Future Work

This final chapter brings together the key insights from our research and explores the potential for future developments. In the following sections, we discuss the applications and implications of the findings from each chapter of this thesis. Finally, we highlight some questions and challenges that could serve as a foundation for future research.

9.1 Analysis of Tweakable Block Ciphers

9.1.1 The Tweakable Enciphering Scheme CTET$^+$

Applicability. In 2008, an actor like Google processed tens of petabytes of data every day [110], and this number must have significantly increased during the last decade. Indeed, as of 2018, [272] estimated that 33 zettabytes of data has been processed worldwide, and predicted this number would grow to 175 zettabytes by 2025. This amounts to a huge amount of data being stored and encrypted at rest (most likely exabytes of data, which is close to the birthday bound of 2^{64} blocks), under many different keys (i.e. in a multi-user scenario). Besides, the current standard disk encryption algorithm, AES-XTS, only offers security up to the birthday bound, and suffers from the fact that, as a sector is rewritten, an adversary can learn information on the plaintexts with a granularity of 16 bytes. Given such information, it seems reasonable to upgrade the underlying encryption scheme to an algorithm that offers worse granularity to the adversary, as well as being beyond birthday bound secure.

Comparison to Previous Constructions. In Chapter 3, we aim to construct an enciphering scheme based on an underlying block cipher. Hence, we consider our SPN construction in the secret permutation model, whereas [85] focused on the use of public permutations. Besides, CTET$^+$ improves on the generic 2-round SPN construction from [85] in two distinct ways:

1. we use the same S-box for both rounds of the SPN, which can help reducing the memory requirements of our scheme;

2. we remark that the middle linear layer can actually have weaker properties than the other two linear layers, which allows the use of a significantly more efficient key-independent matrix multiplication.

Open Questions and Future Work. In this work, we build upon the results on tweakable SPNs with independent round keys and permutations from Cogliati et al. [85] by constructing a 2-round tweakable SPN, based on a single secret permutation, that uses a significantly more efficient middle linear layer, while still offering the same level of beyond-birthday-bound security. We list a few open questions for future research:

1. **Key Material Efficiency:** Is there a way to reduce the amount of key material required by $\texttt{CTET}^{+}$ without diminishing its security guarantees?
2. **Improved SBU Linear Layer:** Can a more efficient linear layer be developed that still meets the security criteria?
3. **Tightness of Security Bounds:** Is Theorem 3.3.1 truly tight, or is there room for refinement in the security bounds it establishes?

9.1.2 Tweakable Even-Mansour and Tweakey Mixing

Why Use The Coupling Technique? In Chapter 4, we employ the coupling technique [7] to establish the IND-CCA bound for $2r$-round TEML. One can argue that the H-coefficients technique [262] might have the potential to derive very tight security bound, as has been demonstrated [81] in the case of key alternating ciphers. However, we note that, in the tight security analysis of key alternating ciphers for an arbitrary number of rounds, the main technical step is actually a combinatorial result (see [81, Lemma 1]) that gives a very sharp lower bound on the number of permutations that can realize a given transcript. Indeed, all the existing tight security analyses of key alternating cipher, be it [81] or a subsequent work by Hoang and Tessaro [171], employ this key result.

The proof utilizes two crucial observations. Firstly, the secret round keys or masks (which are independent of the queries) can be simply subsumed within the permutation calls. Second, and somewhat more importantly, there are no internal input (corresponding to the internal permutation calls) collisions between any two distinct queries. These two facts together help in deriving a conditional lower bound for the current query based on the lower bound for the previous queries. Unfortunately, in the case of TEML, unless the tweak is a constant (equivalent to a key alternating cipher), these two observations no longer apply. First, the secret round masks are tweak-dependent, and thus, depend on adversarial queries. Second, for two distinct tweaks, there can be internal input collisions. As a result, the previous combinatorial result is not applicable directly, and as of now, it seems hard to extend when there are multiple tweaks in play, even for a very small tweak space. Indeed, coming up with a similar result for even AXU hash-based tweakey schedule, let alone the linear tweakey schedule, seems technically challenging.

On the other hand, there is no existing analysis for TEML with arbitrary number of rounds and arbitrary tweak sizes.

It is our firm belief that such analyses (even with a loose security bound) could shed some light on the provable security of the high level design strategy of the popular TWEAKEY framework. Given the apparent need for such an excursion, the technical challenges in reusing/extending the existing results on key alternating ciphers, and inspired by the pragmatic approach from [212, 89], we employ the coupling technique to derive probably a non-tight yet meaningful security bound for arbitrary number of rounds and arbitrary tweak size. We also note that, apart from giving some security guarantee for arbitrary number of rounds, the coupling-based analysis is also useful in getting a good indication on what could be the tight security bound. This could serve as a motivation and a plausible target bound for future endeavors in this

Comparison of Sequential Indifferentiability Results on TEML. Note that, both [316] and this work build over [91]. In fact, our result can be seen as a generalization of [316], for larger key (key size $\geq rn$-bit for $r \geq 1$) and typical linear TWEAKEY schedules. Table 9.1 gives a comparison of the three results.

Table 9.1: *Comparison of sequential indifferentiablity results on TEML. The column* ***Complex.*** *indicates the simulator query/time complexity.*

Rounds	Primitives	Key Size	Complex.	Bounds	Ref.
4	4	n	$O(q^2)$	$O\left(\frac{q^4}{N}\right)$	[91]
4	2	n	$O(q^2)$	$O\left(\frac{q^4}{N}\right)$	[316]
$r+3$	$r+3$	rn	$O\left(q^{r+1}\right)$	$O\left(\frac{q^{2r+2}}{N}\right)$	Section 4.3

Conclusions and Future Work. To conclude, we first showed that the $2r$-round TEML with a specific class of linear tweak-key mixing, and αn-bit tweaks, is IND-CCA secure up to $2^{\frac{r-\alpha}{r}n}$ queries. The main ingredient of our proof is the well-known coupling technique. Our main technical contribution was a refreshed approach to get an upper bound on the probability of failure in coupling, which could be of independent interest.

In particular, we think that this approach might also be useful in the analysis of the Feistel network with linear tweak and key absorption. As with several other coupling-based security bounds [209, 211, 89], we believe that our IND-CCA bound is also not tight. Indeed, we conjecture that beyond a constant $c \geq 4$, the number of rounds can be effectively reduced by half, i.e., to r whenever $r \geq c$, while maintaining the same security level, i.e., up to $2^{\frac{r-\alpha}{r}n}$ queries.

Second, diverting our focus to the sequential indifferentiability setting, we showed that $(r+3)$ rounds are both necessary and sufficient for security of TEML with rn-bit (twea)key and a special class of linear (twea)key mixing function. As a direct consequence of

our results, we gave a sound provable security footing for iterated round-based TBCs, notably following the design paradigm TWEAKEY, that employ a linear tweak-key mixing.

Full Indifferentiability of IEM. The challenge of proving full indifferentiability for the iterated Even-Mansour (IEM) construction with a non-trivial key schedule remains an unresolved issue in cryptographic research. One potential avenue for progress could involve adapting the proof strategy used by Dai et al. [105], for IEM under a weak bijective key schedule (refer to Section 4.1 for a formal definition). However, this adaptation appears to be quite difficult and may not yield a solution. Given these obstacles, a completely new approach may be necessary to tackle this problem effectively.

9.2 Analysis of Authenticated Encryption Schemes

9.2.1 Notes on The Security of MTProto

Since Section 5.5 extensively covers the implications of our subversion attack and provides detailed strategies for mitigating this attack, we add a note on implicit assumptions regarding the security proof of MTProto. Additionally, we reveal our discussion with the Telegram team concerning our findings.

A Note on the Independence of Message Distribution. In the proof of Theorem 5.5.1, we assume that the messages are independent of the keys. This assumption is necessary to avoid input-triggered subversion [111], where the adversary could manipulate messages by introducing spurious space characters, rearranging certain letters, or reordering multiple lines in a message.[1]

However, we note that such attacks, while of theoretical interest, are not particularly practical. First, these attacks typically incur either significant latency or result in a very low number of extracted key bits. Second, and more importantly, on the decryption end, one can use language-specific techniques to detect any anomalies in the decrypted message. For example, one can restrict the set of allowed characters and check for unnecessary spaces before encryption. Additionally, based on English language usage statistics, thresholds can be established to determine if messages have been tampered with. Therefore, in highly cautious applications, which should be the standard for any secure messaging app, it is reasonable to assume that the messages are indeed independent of the keys.

Responsible Disclosure to Telegram. We followed the standard responsible disclosure policy and reported our findings to the Telegram security team in August 2021, along with our suggestion to drop the randomness from the padding algorithm. As noted above, they countered our findings by noting that the official Telegram apps are open source and support reproducible builds that can be verified by independent researchers

[1] In the context of Telegram, this would correspond to reordering chat messages.

who regularly audit the security of Telegram apps. In addition, they also asserted that cryptographic keys can be leaked through various side channels as well. We pointed out that targeted attacks at individuals are still a concerning possibility, and that the presence of side channels only accentuates the impact of our attack, but the Telegram security team did not believe it was a meaningful threat. We concluded our exchange with the Telegram team by mentioning the issue of closed-source (or open source without reproducible builds) third-party Telegram-compatible clients which will still be vulnerable to such mass-surveillance, unless the algorithm is updated. Subsequently, in early December 2023, we also informed several popular third-party Telegram-compatible clients about our findings.

9.2.2 Context-Committing AEADs

In Chapter 6, we prove the CMT-4 security of single-pass schemes, presenting various variants under different assumptions about their components. Notably, we demonstrated that the recently proposed scheme, `Triplex`, achieves CMT-4 security up to half the tag size.

Our findings reveal an intriguing connection between context-committing and leakage-resilient schemes. Although these two security notions are distinct and do not inherently imply each other, the shared design principles enable the creation of efficient schemes that fulfill both goals. An interesting direction for future research is the design of leakage-resilient schemes that maintain commitment beyond half the tag size. Another promising direction is to investigate whether a formal relationship can be established between CIML2 security and CMT-4 security.

9.3 Provable Security in The Quantum Setting

This part is divided to two chapters. Chapter 7 starts with the analysis of all $2n$-bit to n-bit compression functions. We start by showing that these compressing PRFs that are built using two n-bit to n-bit PRF calls are susceptible to either classical or quantum attacks. Furthermore, we show classical or quantum attacks for classes of constructions using three PRF calls. Among the constructions that may be secure, we select `TNT` , `LRQ` , and `LRWQ` , as they are the most efficient invertible ones, which allows them to also be used as tweakable block ciphers. We then prove their PRF security against quantum distinguishers that use less than $2^{n/5}$ queries. Our results, also imply that these constructions are quantum secure TBCs up to $2^{n/6}$ CPA quantum queries. We conjecture that these constructions are secure up to $2^{n/3}$ adversarial queries, and leave the issue of improving the security bound as an interesting open problem.

In Chapter 8, we uncover a flaw in the proof of quantum security for the Luby-Rackoff. In particular, for the technique to work, it is critical that bad databases are only described with information that is actually present in the database. For some constructions, notably the Luby-Rackoff construction, this means that part of the input to the construction will

never appear in the database, and cannot be used to characterize bad databases. On a positive note, we restore the security of the 4-round Luby-Rackoff construction in the *non-adaptive* setting, and prove the security of the 4-round `MistyR` and 5-round `MistyL` constructions.

Towards a Quantum Indistinguishability for qPRP. One of the significant challenges in the realm of post-quantum provable security is developing oracle techniques capable of handling random permutations. Recently, Majenz et al. [223] introduced a generalization of Zhandry's oracle tailored for random permutations. A promising direction for future research would be to integrate this permutation oracle into our existing framework, thereby expanding its capabilities to provide quantum PRP security bounds for a broader range of cryptographic schemes.

Quantum Attacks. In the realm of provable security, quantum distinguishers frequently leverage Simon's algorithm, which is designed for period finding in quantum computations. An interesting direction for future research would be to identify invariants within symmetric schemes that, while secure in the classical setting, are significantly weakened or even broken under quantum attacks. Additionally, exploring new quantum algorithms that could further challenge the security of symmetric cryptographic schemes in the quantum setting would be invaluable. This could lead to a deeper understanding of potential vulnerabilities and the development of more robust quantum-resistant schemes.

Appendices

A

Linear Algebra Results

Here, we present fundamental results in Linear Algebra that will be useful in the development of Section 7.2. For a more comprehensive overview of Linear Algebra, we refer the reader to [138].

A.1 Operator Norm

A useful property of the operator norm is that the operator norm of $A \otimes A'$ is the product of the operator norms of A and A'. We prove this property below.

Lemma A.1.1. $\|A \otimes A'\| = \|A\| \cdot \|A'\|$.

Proof. Note that by definition one has

$$\begin{aligned} A \otimes A' &= \left(\sum_{i=1}^{r} \sigma_i |x_i\rangle\langle y_i| \right) \otimes \left(\sum_{i'=1}^{r'} \sigma'_{i'} |x'_{i'}\rangle\langle y'_{i'}| \right) \\ &= \sum_{i,i'} \sigma_i \sigma'_{i'} \left(|x_i\rangle\langle y_i| \otimes |x'_{i'}\rangle\langle y'_{i'}| \right) \\ &= \sum_{i,i'} \sigma_i \sigma'_{i'} \left(|x_i\rangle \otimes |x'_{i'}\rangle \right) \left(\langle y_i| \otimes \langle y'_{i'}| \right). \end{aligned}$$

Since $|x_1\rangle, \ldots, |x_r\rangle$ are independent and orthonormal and $|x'_1\rangle, \ldots, |x'_{r'}\rangle$ are independent and orthonormal, $\{|x_i\rangle \otimes |x'_{i'}\rangle \mid 1 \leq i \leq r, 1 \leq i' \leq r'\}$ also forms a set of independent and orthonormal vectors in the tensor product space $\mathbb{C}[\mathcal{X}_1] \otimes \mathbb{C}[\mathcal{X}'_1]$, and similarly, $\{|y_i\rangle \otimes |y'_{i'}\rangle \mid 1 \leq i \leq r, 1 \leq i' \leq r'\}$ also forms a set of independent and orthonormal vectors in the tensor product space $\mathbb{C}[\mathcal{X}_0] \otimes \mathbb{C}[\mathcal{X}'_0]$. Thus,

$$A \otimes A' = \sum_{i,i'} \sigma_i \sigma'_{i'} \left(|x_i\rangle \otimes |x'_{i'}\rangle \right) \left(\langle y_i| \otimes \langle y'_{i'}| \right)$$

is a singular value decomposition of $A \otimes A'$, and consequently

$$\|A \otimes A'\| = \max_{i,i'} \sigma_i \sigma'_{i'} = \left(\max_i \sigma_i \right) \cdot \left(\max_{i'} \sigma'_{i'} \right) = \|A\| \cdot \|A'\|.$$

□

A.2 Frobenius Norm

The *Frobenius Norm* of the linear operator A is defined as

$$\|A\|_F := \sqrt{\sum_{x\in\mathcal{X}_0} \|A|x\rangle\|^2} = \sqrt{\sum_{x\in\mathcal{X}_0, y\in\mathcal{X}_1} \left|\langle y \mid A \mid x\rangle\right|^2}.$$

The following property establishes that the Frobenius norm acts as an upper bound on the operator norm.

Lemma A.2.1. *For any $|\psi\rangle \in \mathbb{C}[\mathcal{X}_0]$, we have*

$$\|A\| \leq \|A\|_F$$

Proof. Note that by definition for any $|\psi\rangle \in \mathbb{C}[\mathcal{X}_0]$

$$\begin{aligned}
\|A|\psi\rangle\| &= \|A \sum_{x\in\mathcal{X}_0} |x\rangle\langle x||\psi\rangle\| \\
&\leq \sum_{x\in\mathcal{X}_0} \|\langle x|\psi\rangle A|x\rangle\| && \text{(Triangle Inequality)} \\
&= \sum_{x\in\mathcal{X}_0} |\langle x|\psi\rangle| \cdot \|A|x\rangle\| \\
&\leq \sqrt{\sum_{x\in\mathcal{X}_0} |\langle x|\psi\rangle|^2} \cdot \sqrt{\sum_{x\in\mathcal{X}_0} \|A|x\rangle\|^2} && \text{(Cauchy-Schwarz)} \\
&= \||\psi\rangle\| \cdot \|A\|_F.
\end{aligned}$$

This gives the inequality

$$\|A\| = \sup_{\||\psi\rangle\|=1} \|A|\psi\rangle\| \leq \|A\|_F.$$

□

A.3 Control Registers and Controlled Operators

Consider a linear operator $A : \mathbb{C}[\mathcal{X}] \otimes \mathbb{C}[\mathcal{X}_0'] \longrightarrow \mathbb{C}[\mathcal{X}] \otimes \mathbb{C}[\mathcal{X}_1']$, and a set of linear operators $\{A_x : \mathbb{C}[\mathcal{X}_0'] \longrightarrow \mathbb{C}[\mathcal{X}_1'] \mid x \in \mathcal{X}\}$, such that for every $x \in \mathcal{X}$ and every $|\psi\rangle \in \mathbb{C}[\mathcal{X}_0']$, we have

$$A(|x\rangle \otimes |\psi\rangle) = |x\rangle \otimes A_x|\psi\rangle.$$

Then, A is called a *controlled operator* and the register containing the part of the input corresponding to $\mathbb{C}[\mathcal{X}]$ is called the *control register* of A. The following property gives an upper bound on the norm of A by maximizing the norms of the specified family of linear operators.

Lemma A.3.1.

$$\|A\| \leq \max_{x\in\mathcal{X}} \|A_x\|$$

Proof. For any $|\phi\rangle \in \mathbb{C}[\mathcal{X}]$ and any $|\psi\rangle \in \mathbb{C}[\mathcal{X}_0']$, we have

$$\begin{aligned}
\|A(|\phi\rangle \otimes |\psi\rangle)\| &= \left\| \sum_{x\in\mathcal{X}} \langle x|\phi\rangle A(|x\rangle \otimes |\psi\rangle) \right\| \\
&= \left\| \sum_{x\in\mathcal{X}} \langle x|\phi\rangle |x\rangle \otimes A_x|\psi\rangle \right\| \\
&= \left\| \sum_{x\in\mathcal{X}, y\in\mathcal{X}_0'} \langle x|\phi\rangle\langle y|\psi\rangle |x\rangle \otimes A_x|y\rangle \right\| \\
&= \left\| \sum_{x\in\mathcal{X}, y\in\mathcal{X}_0', z\in\mathcal{X}_1'} \langle x|\phi\rangle\langle y|\psi\rangle\langle z \mid A_x \mid y\rangle |x\rangle \otimes |z\rangle \right\| \\
&= \left\| \sum_{x\in\mathcal{X}, z\in\mathcal{X}_1'} \langle x|\phi\rangle \left(\sum_{y\in\mathcal{X}_0'} \langle y|\psi\rangle\langle z \mid A_x \mid y\rangle \right) |x\rangle \otimes |z\rangle \right\| \\
&= \sqrt{\sum_{x\in\mathcal{X}, z\in\mathcal{X}_1'} |\langle x|\phi\rangle|^2 \cdot \left| \sum_{y\in\mathcal{X}_0'} \langle y|\psi\rangle\langle z \mid A_x \mid y\rangle \right|^2} \\
&= \sqrt{\sum_{x\in\mathcal{X}} |\langle x|\phi\rangle|^2 \cdot \sum_{z\in\mathcal{X}_1'} \left| \sum_{y\in\mathcal{X}_0'} \langle y|\psi\rangle\langle z \mid A_x \mid y\rangle \right|^2} \\
&= \sqrt{\sum_{x\in\mathcal{X}} |\langle x|\phi\rangle|^2 \cdot \left\| \sum_{z\in\mathcal{X}_1'} \left(\sum_{y\in\mathcal{X}_0'} \langle y|\psi\rangle\langle z \mid A_x \mid y\rangle \right) |z\rangle \right\|^2} \\
&= \sqrt{\sum_{x\in\mathcal{X}} |\langle x|\phi\rangle|^2 \cdot \left\| \sum_{y\in\mathcal{X}_0'} \langle y|\psi\rangle \left(\sum_{z\in\mathcal{X}_1'} \langle z \mid A_x \mid y\rangle |z\rangle \right) \right\|^2} \\
&= \sqrt{\sum_{x\in\mathcal{X}} ||\langle x|\phi\rangle|^2 \cdot \left\| \sum_{y\in\mathcal{X}_0'} \langle y|\psi\rangle A_x|y\rangle \right\|^2} \\
&= \sqrt{\sum_{x\in\mathcal{X}} |\langle x|\phi\rangle|^2 \cdot \|A_x|\psi\rangle\|^2} \\
&\leq \sqrt{\sum_{x\in\mathcal{X}} |\langle x|\phi\rangle|^2} \cdot \max_{x\in\mathcal{X}} \|A_x|\psi\rangle\| = \max_{x\in\mathcal{X}} \|A_x|\psi\rangle\|.
\end{aligned}$$

By the definition of operator norms this concludes our proof. □

A.4 Trace Norm

The following Lemma establishes a relation between the trace norm of the outer product of two vectors and their respective norms.

Lemma A.4.1. *Let $\mathcal{H}$ be a finite dimensional complex Hilbert space. Let $|\phi\rangle, |\psi\rangle \in \mathcal{H}$ be two (not necessarily distinct) vectors, such that $\||\phi\rangle\|, \||\psi\rangle\| \leq 1$. Then, we have*

$$\||\psi\rangle\langle\phi|\|_1 = \||\phi\rangle\langle\psi|\|_1 = \||\phi\rangle\| \cdot \||\psi\rangle\| \leq \min\{\||\phi\rangle\|, \||\psi\rangle\|\}.$$

Proof. The inequality is obvious. Without loss of generality, we assume that $\||\phi\rangle\|, \||\psi\rangle\| > 0$, otherwise the statement is vacuously true. Next, as a proof of this Lemma is elementary, we provide two proofs of slightly different flavors: one that is based purely on definitions, and a slightly more derivative in nature.

1. The matrix $|\phi\rangle\langle\psi|$ has rank 1, whence $\||\phi\rangle\langle\psi|\|_1 = \||\phi\rangle\langle\psi|\| = \||\phi\rangle\| \cdot \||\psi\rangle\|$.
2. We have

$$\begin{aligned}
\||\phi\rangle\langle\psi|\|_1 &= \mathsf{Tr}(\sqrt{|\psi\rangle\langle\phi||\phi\rangle\langle\psi|}) \\
&= \||\phi\rangle\|\,\mathsf{Tr}(\sqrt{|\psi\rangle\langle\psi|}) \\
&= \||\phi\rangle\| \cdot \||\psi\rangle\| \cdot \mathsf{Tr}\left(\sqrt{\left(\frac{|\psi\rangle}{\|\psi\|}\right)\left(\frac{\langle\psi|}{\|\psi\|}\right)}\right) \\
&= \||\phi\rangle\| \cdot \||\psi\rangle\| \cdot \mathsf{Tr}\left(\left(\frac{|\psi\rangle}{\|\psi\|}\right)\left(\frac{\langle\psi|}{\|\psi\|}\right)\right),
\end{aligned}$$

   where the last equality follows from the fact that trace of a rank-1 projection matrix[1] is 1. Finally, $\||\psi\rangle\langle\phi|\|_1 = \||\phi\rangle\langle\psi|\|_1$ follows from the same argumentation as applied to $\||\psi\rangle\langle\phi|\|_1$.

□

[1] In the orthonormal basis containing $|\psi\rangle/\||\psi\rangle\|$.

This page intentionally left blank.

Bibliography

[1] Abdalla, M., Belaïd, S., Fouque, P.A.: Leakage-resilient symmetric encryption via re-keying. In: Bertoni, G., Coron, J.S. (eds.) CHES 2013. LNCS, vol. 8086, pp. 471–488. Springer, Berlin, Heidelberg (Aug 2013)

[2] Adams, D.C., Gilchrist, J.: The CAST-256 Encryption Algorithm. RFC 2612 (Jun 1999), https://www.rfc-editor.org/info/rfc2612

[3] Advanced Encryption Standard (AES). National Institute of Standards and Technology, NIST FIPS PUB 197, U.S. Department of Commerce (Nov 2001)

[4] Agrawal, D., Archambeault, B., Rao, J.R., Rohatgi, P.: The EM side-channel(s). In: Kaliski, Jr., B.S., Koç, Çetin Kaya., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 29–45. Springer, Berlin, Heidelberg (Aug 2003)

[5] Aiello, W., Rajagopalan, S., Venkatesan, R.: High-speed pseudorandom number generation with small memory. In: Knudsen, L.R. (ed.) FSE'99. LNCS, vol. 1636, pp. 290–304. Springer, Berlin, Heidelberg (Mar 1999)

[6] Albrecht, M.R., Mareková, L., Paterson, K.G., Stepanovs, I.: Four Attacks and a Proof for Telegram. In: Security and Privacy – IEEE-S&P 2022, Proceedings. pp. 87–106 (2022)

[7] Aldous, D.J.: Random walks on finite groups and rapidly mixing Markov chains. In: Séminaire de Probabilités XVII. Lecture Notes in Mathematics, vol. 986, pp. 243–297. Springer (1983)

[8] Andreeva, E., Bogdanov, A., Dodis, Y., Mennink, B., Steinberger, J.P.: On the Indifferentiability of Key-Alternating Ciphers. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology - CRYPTO 2013 (Proceedings, Part I). Lecture Notes in Computer Science, vol. 8042, pp. 531–550. Springer (2013), full version available at http://eprint.iacr.org/2013/061

[9] Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., Tokita, T.: Camellia: A 128-bit block cipher suitable for multiple platforms - Design and analysis. In: Stinson, D.R., Tavares, S.E. (eds.) SAC 2000. LNCS, vol. 2012, pp. 39–56. Springer, Berlin, Heidelberg (Aug 2001)

[10] Armour, M., Poettering, B.: Substitution Attacks against Message Authentication. IACR Trans. Symmetric Cryptol. 2019(3), 152–168 (2019)

[11] Armour, M., Poettering, B.: Subverting Decryption in AEAD. In: Cryptography and Coding – IMACC 2019, Proceedings. pp. 22–41 (2019)

[12] von Arx, T., Paterson, K.G.: On the cryptographic fragility of the Telegram ecosystem. Cryptology ePrint Archive, Report 2022/595 (2022), `https://eprint.iacr.org/2022/595`

[13] Ateniese, G., Magri, B., Venturi, D.: Subversion-Resilient Signature Schemes. In: Computer and Communications Security – ACM-CCS 2015, Proceedings. pp. 364–375 (2015)

[14] Aumasson, J.P., Bernstein, D.J.: SipHash: A fast short-input PRF. In: Galbraith, S.D., Nandi, M. (eds.) INDOCRYPT 2012. LNCS, vol. 7668, pp. 489–508. Springer, Berlin, Heidelberg (Dec 2012)

[15] Baek, J., Susilo, W., Kim, J., Chow, Y.: Subversion in Practice: How to Efficiently Undermine Signatures. IEEE Access 7, 68799–68811 (2019)

[16] Banik, S., Bogdanov, A., Luykx, A., Tischhauser, E.: SUNDAE: Small universal deterministic authenticated encryption for the internet of things. IACR Trans. Symm. Cryptol. 2018(3), 1–35 (2018)

[17] Bao, Z., Guo, C., Guo, J., Song, L.: TNT: How to tweak a block cipher. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part II. LNCS, vol. 12106, pp. 641–673. Springer, Cham (May 2020)

[18] Bao, Z., Guo, J., Iwata, T., Minematsu, K.: Zocb and zotr: Tweakable blockcipher modes for authenticated encryption with full absorption. IACR Transactions on Symmetric Cryptology pp. 1–54 (Jun 2019)

[19] Barthe, G., Belaïd, S., Dupressoir, F., Fouque, P.A., Grégoire, B., Strub, P.Y., Zucchini, R.: Strong non-interference and type-directed higher-order masking. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) ACM CCS 2016. pp. 116–129. ACM Press (Oct 2016)

[20] Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The skinny family of block ciphers and its low-latency variant mantis. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology – CRYPTO 2016. pp. 123–153. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)

[21] Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS, pp. 123–153. Springer Berlin Heidelberg (2016)

[22] Bellare, M.: Practice-Oriented Provable-Security, pp. 1–15. Springer Berlin Heidelberg (1999)

[23] Bellare, M., Bernstein, D.J., Tessaro, S.: Hash-function based PRFs: AMAC and its multi-user security. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 566–595. Springer, Berlin, Heidelberg (May 2016)

[24] Bellare, M., Boldyreva, A., Knudsen, L.R., Namprempre, C.: Online ciphers and the hash-CBC construction. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 292–309. Springer, Berlin, Heidelberg (Aug 2001)

[25] Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer, Berlin, Heidelberg (May 2000)

[26] Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Koblitz, N. (ed.) CRYPTO'96. LNCS, vol. 1109, pp. 1–15. Springer, Berlin, Heidelberg (Aug 1996)

[27] Bellare, M., Canetti, R., Krawczyk, H.: Pseudorandom functions revisited: The cascade construction and its concrete security. In: 37th FOCS. pp. 514–523. IEEE Computer Society Press (Oct 1996)

[28] Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A Concrete Security Treatment of Symmetric Encryption. In: Symposium on Foundations of Computer Science - FOCS '97. pp. 394–403. IEEE Computer Society (1997)

[29] Bellare, M., Guérin, R., Rogaway, P.: XOR MACs: New Methods for Message Authentication Using Finite Pseudorandom Functions, pp. 15–28. Springer Berlin Heidelberg (1995)

[30] Bellare, M., Hoang, V.T.: Efficient schemes for committing authenticated encryption. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part II. LNCS, vol. 13276, pp. 845–875. Springer, Cham (May / Jun 2022)

[31] Bellare, M., Jaeger, J., Kane, D.: Mass-Surveillance without the State: Strongly Undetectable Algorithm-Substitution Attacks. In: Computer and Communications Security – ACM-CCS 2015, Proceedings. p. 1431–1440 (2015)

[32] Bellare, M., Kilian, J., Rogaway, P.: The security of the cipher block chaining message authentication code. Journal of Computer and System Sciences 61(3), 362–399 (2000)

[33] Bellare, M., Kohno, T.: A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 491–506. Springer, Berlin, Heidelberg (May 2003)

[34] Bellare, M., Krovetz, T., Rogaway, P.: Luby-Rackoff Backwards: Increasing Security by Making Block Ciphers Non-invertible. In: Nyberg, K. (ed.) Advances in Cryptology - EUROCRYPT '98. Lecture Notes in Computer Science, vol. 1403, pp. 266–280. Springer (1998)

[35] Bellare, M., Namprempre, C.: Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In: Okamoto, T. (ed.) Advances in Cryptology - ASIACRYPT 2000. Lecture Notes in Computer Science, vol. 1976, pp. 531–545. Springer (2000)

[36] Bellare, M., Paterson, K.G., Rogaway, P.: Security of Symmetric Encryption against Mass Surveillance. In: Advances in Cryptology – CRYPTO 2014, Proceedings. pp. 1–19 (2014)

[37] Bellare, M., Ristenpart, T.: Multi-Property-Preserving Hash Domain Extension and the EMD Transform. In: Lai, X., Chen, K. (eds.) Advances in Cryptology - ASIACRYPT 2006. Lecture Notes in Computer Science, vol. 4284, pp. 299–314. Springer (2006)

[38] Bellare, M., Rogaway, P.: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In: ACM Conference on Computer and Communications Security. pp. 62–73 (1993)

[39] Bellare, M., Rogaway, P.: Collision-resistant hashing: Towards making UOWHFs practical. In: Kaliski, Jr., B.S. (ed.) CRYPTO'97. LNCS, vol. 1294, pp. 470–484. Springer, Berlin, Heidelberg (Aug 1997)

[40] Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Berlin, Heidelberg (May / Jun 2006)

[41] Bellizia, D., Bronchain, O., Cassiers, G., Grosso, V., Guo, C., Momin, C., Pereira, O., Peters, T., Standaert, F.X.: Mode-level vs. implementation-level physical security in symmetric cryptography - A practical guide through the leakage-resistance jungle. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part I. LNCS, vol. 12170, pp. 369–400. Springer, Cham (Aug 2020)

[42] Berndt, S., Wichelmann, J., Pott, C., Traving, T., Eisenbarth, T.: ASAP: Algorithm Substitution Attacks on Cryptographic Protocols. IACR Cryptol. ePrint Arch. 2020, 1452 (2020)

[43] Bernstein, D.J.: The Poly1305-AES Message-Authentication Code. In: Gilbert, H., Handschuh, H. (eds.) Fast Software Encryption - FSE 2005. Lecture Notes in Computer Science, vol. 3557, pp. 32–49. Springer (2005)

[44] Bernstein, D.J., Duif, N., Lange, T., Schwabe, P., Yang, B.Y.: High-speed high-security signatures. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 124–142. Springer, Berlin, Heidelberg (Sep / Oct 2011)

[45] Berti, F., Guo, C., Peters, T., Standaert, F.X.: Efficient leakage-resilient MACs without idealized assumptions. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part II. LNCS, vol. 13091, pp. 95–123. Springer, Cham (Dec 2021)

[46] Berti, F., Koeune, F., Pereira, O., Peters, T., Standaert, F.X.: Ciphertext integrity with misuse and leakage: Definition and efficient constructions with symmetric primitives. In: Kim, J., Ahn, G.J., Kim, S., Kim, Y., López, J., Kim, T. (eds.) ASIACCS 18. pp. 37–50. ACM Press (Apr 2018)

[47] Berti, F., Pereira, O., Peters, T., Standaert, F.X.: On leakage-resilient authenticated encryption with decryption leakages. IACR Trans. Symm. Cryptol. 2017(3), 271–293 (2017)

[48] Bhargavan, K., Leurent, G.: On the Practical (In-)Security of 64-bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN. In: Computer and Communications Security – ACM-CCS 2016, Proceedings. pp. 456–467 (2016)

[49] Bhaumik, R., Bonnetain, X., Chailloux, A., Leurent, G., Naya-Plasencia, M., Schrottenloher, A., Seurin, Y.: QCB: Efficient quantum-secure authenticated encryption. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part I. LNCS, vol. 13090, pp. 668–698. Springer, Cham (Dec 2021)

[50] Bhaumik, R., Cogliati, B., Ethan, J., Jha, A.: On quantum secure compressing pseudorandom functions. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part III. LNCS, vol. 14440, pp. 34–66. Springer, Singapore (Dec 2023)

[51] Bhaumik, R., Cogliati, B., Ethan, J., Jha, A.: Mind the bad norms: Revisiting compressed oracle-based quantum indistinguishability proofs. Cryptology ePrint Archive, Paper 2024/1478 (2024), `https://eprint.iacr.org/2024/1478`

[52] Bhaumik, R., List, E., Nandi, M.: ZCZ - achieving n-bit SPRP security with a minimal number of tweakable-block-cipher calls. In: Advances in Cryptology - ASIACRYPT 2018, Proceedings, Part I. pp. 336–366 (2018)

[53] Biham, E., Anderson, R., Knudsen, L.: Serpent: A new block cipher proposal. In: International workshop on fast software encryption. pp. 222–238. Springer (1998)

[54] Biryukov, A.: Skipjack, pp. 586–587. Springer US (1998)

[55] Biryukov, A., Khovratovich, D., Nikolić, I.: Distinguisher and Related-Key Attack on the Full AES-256. In: Halevi, S. (ed.) Advances in Cryptology - CRYPTO 2009. Lecture Notes in Computer Science, vol. 5677, pp. 231–249. Springer (2009)

[56] Black, J.: The ideal-cipher model, revisited: An uninstantiable blockcipher-based hash function. In: Robshaw, M.J.B. (ed.) FSE 2006. LNCS, vol. 4047, pp. 328–340. Springer, Berlin, Heidelberg (Mar 2006)

[57] Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: UMAC: Fast and Secure Message Authentication. In: Wiener, M.J. (ed.) Advances in Cryptology - CRYPTO '99. Lecture Notes in Computer Science, vol. 1666, pp. 216–233. Springer (1999)

[58] Black, J., Rogaway, P.: A block-cipher mode of operation for parallelizable message authentication. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 384–397. Springer, Berlin, Heidelberg (Apr / May 2002)

[59] Blum, M., Micali, S.: How to generate cryptographically strong sequences of pseudorandom bits. SIAM Journal on Computing 13(4), 850–864 (1984)

[60] Bogdanov, A., Knudsen, L.R., Leander, G., Standaert, F.X., Steinberger, J.P., Tischhauser, E.: Key-Alternating Ciphers in a Provable Setting: Encryption Using a Small Number of Public Permutations - (Extended Abstract). In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology - EUROCRYPT 2012. Lecture Notes in Computer Science, vol. 7237, pp. 45–62. Springer (2012)

[61] Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 41–69. Springer, Berlin, Heidelberg (Dec 2011)

[62] Boneh, D., Zhandry, M.: Quantum-secure message authentication codes. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 592–608. Springer, Berlin, Heidelberg (May 2013)

[63] Bonnetain, X., Naya-Plasencia, M.: Hidden shift quantum cryptanalysis and implications. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part I. LNCS, vol. 11272, pp. 560–592. Springer, Cham (Dec 2018)

[64] Bonnetain, X., Naya-Plasencia, M., Schrottenloher, A.: On quantum slide attacks. In: Paterson, K.G., Stebila, D. (eds.) SAC 2019. LNCS, vol. 11959, pp. 492–519. Springer, Cham (Aug 2019)

[65] Bonnetain, X., Naya-Plasencia, M., Schrottenloher, A.: Quantum security analysis of AES. IACR Trans. Symm. Cryptol. 2019(2), 55–93 (2019)

[66] Bonnetain, X., Schrottenloher, A., Sibleyras, F.: Beyond quadratic speedups in quantum attacks on symmetric schemes. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part III. LNCS, vol. 13277, pp. 315–344. Springer, Cham (May / Jun 2022)

[67] Brassard, G., Høyer, P., Tapp, A.: Quantum cryptanalysis of hash and claw-free functions. In: Lucchesi, C.L., Moura, A.V. (eds.) LATIN 1998. LNCS, vol. 1380, pp. 163–169. Springer, Berlin, Heidelberg (Apr 1998)

[68] Campbell, C.: Design and specification of cryptographic capabilities. IEEE Communications Society Magazine 16(6), 15–19 (Nov 1978)

[69] Canetti, R., Goldreich, O., Halevi, S.: The Random Oracle Methodology, Revisited (Preliminary Version). In: Vitter, J.S. (ed.) Symposium on Theory of Computing - STOC '98. pp. 209–218. ACM (1998), https://doi.org/10.1145/276698.276741, full version available at http://arxiv.org/abs/cs.CR/0010019

[70] Carroll, M.: Telegram will 'probably' hit one billion users within the year - what is it and why is it so popular? Sky News (2024), https://news.sky.com/story/telegram-will-probably-hit-one-billion-users-within-the-year-what-is-it-and-why-is-it-so-popular-13117068

[71] Carter, J.L., Wegman, M.N.: Universal classes of hash functions (extended abstract). In: Proceedings of the ninth annual ACM symposium on Theory of computing - STOC '77. STOC '77, ACM Press (1977)

[72] Chailloux, A., Naya-Plasencia, M., Schrottenloher, A.: An efficient quantum collision search algorithm and implications on symmetric cryptography. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 211–240. Springer, Cham (Dec 2017)

[73] Chakraborti, A., Datta, N., Jha, A., Mancillas-López, C., Nandi, M., Sasaki, Y.: Elastic-tweak: A framework for short tweak tweakable block cipher. In: Progress in Cryptology - INDOCRYPT 2021, Proceedings. pp. 114–137 (2021)

[74] Chakraborty, D., Sarka, P.: HCH: A New Tweakable Enciphering Scheme Using the Hash-Encrypt-Hash Approach. In: Barua, R., Lange, T. (eds.) Progress in Cryptology - INDOCRYPT 2006. Lecture Notes in Computer Science, vol. 4329, pp. 287–302. Springer (2006)

[75] Chakraborty, D., Sarkar, P.: A New Mode of Encryption Providing a Tweakable Strong Pseudo-random Permutation. In: Robshaw, M. (ed.) Fast Software Encryption - FSE 2006. Lecture Notes in Computer Science, vol. 4047, pp. 293–309. Springer (2006)

[76] Chakraborty, D., Sarkar, P.: A general construction of tweakable block ciphers and different modes of operations. IEEE Trans. Information Theory 54(5), 1991–2006 (2008)

[77] Chari, S., Jutla, C.S., Rao, J.R., Rohatgi, P.: Towards sound approaches to counteract power-analysis attacks. In: Wiener, M.J. (ed.) CRYPTO'99. LNCS, vol. 1666, pp. 398–412. Springer, Berlin, Heidelberg (Aug 1999)

[78] Chen, R., Huang, X., Yung, M.: Subvert KEM to Break DEM: Practical Algorithm-Substitution Attacks on Public-Key Encryption. In: Advances in Cryptology – ASIACRYPT 2020, Proceedings, Part II. pp. 98–128 (2020)

[79] Chen, S., Lampe, R., Lee, J., Seurin, Y., Steinberger, J.: Minimizing the Two-Round Even-Mansour Cipher, pp. 39–56. Springer Berlin Heidelberg (2014)

[80] Chen, S., Lampe, R., Lee, J., Seurin, Y., Steinberger, J.P.: Minimizing the Two-Round Even-Mansour Cipher. In: Garay, J.A., Gennaro, R. (eds.) Advances in Cryptology - CRYPTO 2014 (Proceedings, Part I). Lecture Notes in Computer Science, vol. 8616, pp. 39–56. Springer (2014), full version available at `http://eprint.iacr.org/2014/443`

[81] Chen, S., Steinberger, J.: Tight Security Bounds for Key-Alternating Ciphers. In: Nguyen, P.Q., Oswald, E. (eds.) Advances in Cryptology - EUROCRYPT 2014. Lecture Notes in Computer Science, vol. 8441, pp. 327–350. Springer (2014), full version available at `http://eprint.iacr.org/2013/222`

[82] Choi, W., Lee, J., Lee, Y.: Building prfs from tprps: Beyond the block and the tweak length bounds. IACR Cryptol. ePrint Arch. p. 918 (2022)

[83] Chung, K.M., Fehr, S., Huang, Y.H., Liao, T.N.: On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In: Canteaut, A., Standaert, F.X. (eds.) EUROCRYPT 2021, Part II. LNCS, vol. 12697, pp. 598–629. Springer, Cham (Oct 2021)

[84] Clavier, C., Coron, J.S., Dabbous, N.: Differential power analysis in the presence of hardware countermeasures. In: Koç, Çetin Kaya., Paar, C. (eds.) CHES 2000. LNCS, vol. 1965, pp. 252–263. Springer, Berlin, Heidelberg (Aug 2000)

[85] Cogliati, B., Dodis, Y., Katz, J., Lee, J., Steinberger, J.P., Thiruvengadam, A., Zhang, Z.: Provable Security of (Tweakable) Block Ciphers Based on Substitution-Permutation Networks. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology - CRYPTO 2018 - Proceedings, Part 1. Lecture Notes in Computer Science, vol. 10991, pp. 722–753. Springer (2018)

[86] Cogliati, B., Ethan, J., Jha, A.: Subverting telegram's end-to-end encryption. IACR Trans. Symm. Cryptol. 2023(1), 5–40 (2023)

[87] Cogliati, B., Ethan, J., Jha, A., Saha, S.K.: On large tweaks in tweakable Even-Mansour with linear tweak and key mixing. IACR Trans. Symm. Cryptol. 2023(4), 330–364 (2023)

[88] Cogliati, B., Ethan, J., Lallemand, V., Lee, B., Lee, J., Minier, M.: CTET+: A beyond-birthday-bound secure tweakable enciphering scheme using a single pseudorandom permutation. IACR Trans. Symm. Cryptol. 2021(4), 1–35 (2021)

[89] Cogliati, B., Lampe, R., Seurin, Y.: Tweaking Even-Mansour Ciphers. In: Gennaro, R., Robshaw, M. (eds.) Advances in Cryptology - CRYPTO 2015 (Proceedings, Part I). Lecture Notes in Computer Science, vol. 9215, pp. 189–208. Springer (2015), full version available at http://eprint.iacr.org/2015/539

[90] Cogliati, B., Lee, J., Seurin, Y.: New constructions of macs from (tweakable) block ciphers. IACR Trans. Symmetric Cryptol. 2017(2), 27–58 (2017)

[91] Cogliati, B., Seurin, Y.: On the provable security of the iterated even-mansour cipher against related-key and chosen-key attacks. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9056, pp. 584–613. Springer (2015), full version available at http://eprint.iacr.org/2015/069

[92] Cogliati, B., Seurin, Y.: On the provable security of the iterated Even-Mansour cipher against related-key and chosen-key attacks. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part I. LNCS, vol. 9056, pp. 584–613. Springer, Berlin, Heidelberg (Apr 2015)

[93] Cogliati, B., Seurin, Y.: EWCDM: An efficient, beyond-birthday secure, nonce-misuse resistant MAC. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 121–149. Springer, Berlin, Heidelberg (Aug 2016)

[94] Cogliati, B., Dodis, Y., Katz, J., Lee, J., Steinberger, J., Thiruvengadam, A., Zhang, Z.: Provable Security of (Tweakable) Block Ciphers Based on Substitution-Permutation Networks, pp. 722–753. Springer International Publishing (2018)

[95] Cogliati, B., Ethan, J., Jha, A., Nandi, M., Saha, A.: On the number of restricted solutions to constrained systems and their applications. Cryptology ePrint Archive, Paper 2024/1163 (2024), https://eprint.iacr.org/2024/1163

[96] Cogliati, B., Jean, J., Peyrin, T., Seurin, Y.: A long tweak goes a long way: High multi-user security authenticated encryption from tweakable block ciphers. Cryptology ePrint Archive, Paper 2022/846 (2022), https://eprint.iacr.org/2022/846, https://eprint.iacr.org/2022/846

[97] Coron, J.S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård Revisited: How to Construct a Hash Function. In: Shoup, V. (ed.) Advances in Cryptology - CRYPTO 2005. Lecture Notes in Computer Science, vol. 3621, pp. 430–448. Springer (2005)

[98] Coron, J.S., Giraud, C., Prouff, E., Renner, S., Rivain, M., Vadnala, P.K.: Conversion of security proofs from one leakage model to another: A new issue. In: Schindler, W., Huss, S.A. (eds.) COSADE 2012. LNCS, vol. 7275, pp. 69–81. Springer, Berlin, Heidelberg (May 2012)

[99] Coron, J.S., Patarin, J., Seurin, Y.: The random oracle model and the ideal cipher model are equivalent. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 1–20. Springer, Berlin, Heidelberg (Aug 2008)

[100] Crow, B., Widjaja, I., Kim, J., Sakai, P.: Ieee 802.11 wireless local area networks. IEEE Communications Magazine 35(9), 116–126 (Sep 1997)

[101] Crowley, P.: Mercy: A Fast Large Block Cipher for Disk Sector Encryption. In: Schneier, B. (ed.) Fast Software Encryption - FSE 2000. Lecture Notes in Computer Science, vol. 1978, pp. 49–63. Springer (2000)

[102] Czajkowski, J., Hülsing, A., Schaffner, C.: Quantum indistinguishability of random sponges. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 296–325. Springer, Cham (Aug 2019)

[103] Daemen, J., Rijmen, V.: Probability distributions of correlation and differentials in block ciphers. Cryptology ePrint Archive, Report 2005/212 (2005), https://eprint.iacr.org/2005/212

[104] Dai, W., Hoang, V.T., Tessaro, S.: Information-theoretic indistinguishability via the chi-squared method. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part III. LNCS, vol. 10403, pp. 497–523. Springer, Cham (Aug 2017)

[105] Dai, Y., Seurin, Y., Steinberger, J., Thiruvengadam, A.: Indifferentiability of iterated Even-Mansour ciphers with non-idealized key-schedules: Five rounds are necessary and sufficient. Cryptology ePrint Archive, Report 2017/042 (2017), https://eprint.iacr.org/2017/042

[106] Dai, Y., Seurin, Y., Steinberger, J.P., Thiruvengadam, A.: Indifferentiability of iterated even-mansour ciphers with non-idealized key-schedules: Five rounds are necessary and sufficient. In: Advances in Cryptology - CRYPTO 2017, Proceedings, Part III. pp. 524–555 (2017)

[107] Damgård, I.: A design principle for hash functions. In: Brassard, G. (ed.) CRYPTO'89. LNCS, vol. 435, pp. 416–427. Springer, New York (Aug 1990)

[108] Damgård, I., Nielsen, J.B.: Expanding pseudorandom functions; or: From known-plaintext security to chosen-plaintext security. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 449–464. Springer, Berlin, Heidelberg (Aug 2002)

[109] Datta, N., Dutta, A., Nandi, M., Paul, G.: Double-block hash-then-sum: A paradigm for constructing BBB secure PRF. IACR Trans. Symm. Cryptol. 2018(3), 36–92 (2018)

[110] Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. Commun. ACM 51(1), 107–113 (Jan 2008), https://doi.org/10.1145/1327452.1327492

[111] Degabriele, J.P., Farshim, P., Poettering, B.: A More Cautious Approach to Security Against Mass Surveillance. In: Fast Software Encryption – FSE 2015, Revised Selected Papers. pp. 579–598 (2015)

[112] Data encryption standard. National Bureau of Standards, NBS FIPS PUB 46, U.S. Department of Commerce (Jan 1977)

[113] Dhar, C., Ethan, J., Jejurikar, R., Khairallah, M., List, E., Mandal, S.: Context-committing security of leveled leakage-resilient aead. IACR Transactions on Symmetric Cryptology 2024(2), 348–370 (Jun 2024)

[114] Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory 22(6), 644–654 (1976)

[115] Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory 22(6), 644–654 (1976)

[116] Dinur, I.: Tight indistinguishability bounds for the XOR of independent random permutations by Fourier analysis. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part I. LNCS, vol. 14651, pp. 33–62. Springer, Cham (May 2024)

[117] Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1.2: Lightweight authenticated encryption and hashing. J. Cryptol. 34(3), 33 (2021), https://doi.org/10.1007/s00145-021-09398-9

[118] Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1.2: Lightweight authenticated encryption and hashing. Journal of Cryptology 34(3), 33 (Jul 2021)

[119] Dodis, Y., Gennaro, R., Håstad, J., Krawczyk, H., Rabin, T.: Randomness extraction and key derivation using the CBC, cascade and HMAC modes. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 494–510. Springer, Berlin, Heidelberg (Aug 2004)

[120] Dodis, Y., Grubbs, P., Ristenpart, T., Woodage, J.: Fast message franking: From invisible salamanders to encryptment. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part I. LNCS, vol. 10991, pp. 155–186. Springer, Cham (Aug 2018)

[121] Dodis, Y., Katz, J., Steinberger, J.P., Thiruvengadam, A., Zhang, Z.: Provable security of substitution-permutation networks. IACR Cryptology ePrint Archive 2017, 16 (2017), http://eprint.iacr.org/2017/016

[122] Dodis, Y., Pietrzak, K.: Leakage-resilient pseudorandom functions and side-channel attacks on Feistel networks. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 21–40. Springer, Berlin, Heidelberg (Aug 2010)

[123] Dodis, Y., Puniya, P.: On the relation between the ideal cipher and the random oracle models. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 184–206. Springer, Berlin, Heidelberg (Mar 2006)

[124] Dodis, Y., Ristenpart, T., Shrimpton, T.: Salvaging Merkle-Damgård for Practical Applications. In: Joux, A. (ed.) Advances in Cryptology - EUROCRYPT 2009. Lecture Notes in Computer Science, vol. 5479, pp. 371–388. Springer (2009)

[125] Dolmatov, V.: GOST R 34.12-2015: Block Cipher "Kuznyechik". RFC 7801 (Mar 2016), https://www.rfc-editor.org/info/rfc7801

[126] Duc, A., Dziembowski, S., Faust, S.: Unifying leakage models: From probing attacks to noisy leakage. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 423–440. Springer, Berlin, Heidelberg (May 2014)

[127] Duc, A., Faust, S., Standaert, F.X.: Making masking security proofs concrete - or how to evaluate the security of any leaking device. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part I. LNCS, vol. 9056, pp. 401–429. Springer, Berlin, Heidelberg (Apr 2015)

[128] Dutta, A.: Minimizing the two-round tweakable even-mansour cipher. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020. pp. 601–629. Springer International Publishing, Cham (2020)

[129] Dworkin, M.J.: Sp 800-38c. recommendation for block cipher modes of operation: the ccm mode for authentication and confidentiality. Tech. rep., Gaithersburg, MD, USA (2004)

[130] Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: 49th FOCS. pp. 293–302. IEEE Computer Society Press (Oct 2008)

[131] Ehrsam, W.F., Meyer, C.H.W., Smith, J.L., Tuchman, W.L.: Message Verification and Transmission Error Detection by Block Chaining. Patent 4074066, USPTO (1976)

[132] Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. In: Imai, H., Rivest, R.L., Matsumoto, T. (eds.) ASIACRYPT'91. LNCS, vol. 739, pp. 210–224. Springer, Berlin, Heidelberg (Nov 1993)

[133] Even, S., Mansour, Y.: A Construction of a Cipher from a Single Pseudorandom Permutation. Journal of Cryptology 10(3), 151–162 (1997)

[134] Farshim, P., Procter, G.: The Related-Key Security of Iterated Even-Mansour Ciphers. In: Leander, G. (ed.) Fast Software Encryption - FSE 2015. Lecture Notes in Computer Science, vol. 9054, pp. 342–363. Springer (2015), full version available at http://eprint.iacr.org/2014/953

[135] Faust, S., Pietrzak, K., Schipper, J.: Practical leakage-resilient symmetric cryptography. In: Prouff, E., Schaumont, P. (eds.) CHES 2012. LNCS, vol. 7428, pp. 213–232. Springer, Berlin, Heidelberg (Sep 2012)

[136] Ferguson, N., Lucks, S., Schneier, B., Whiting, D., Bellare, M., Kohno, T., Callas, J., Walker, J.: The Skein Hash Function Family. SHA3 Submission to NIST (Round 3) (2010), http://www.skein-hash.info/sites/default/files/skein1.3.pdf

[137] Fluhrer, S.R., McGrew, D.A.: The Security of the Extended Codebook (XCB) Mode of Operation. In: Adams, C., Miri, A., Wiener, M. (eds.) SAC 2007: Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 4876, pp. 311–327. Springer (2007)

[138] Friedberg, S., Insel, A., Spence, L.: Linear Algebra Pearson New International Edition. Pearson Education, Limited (2013)

[139] Goldenberg, D., Hohenberger, S., Liskov, M., Schwartz, E.C., Seyalioglu, H.: On Tweaking Luby-Rackoff Blockciphers. In: Kurosawa, K. (ed.) Advances in Cryptology - ASIACRYPT 2007. Lecture Notes in Computer Science, vol. 4833, pp. 342–356. Springer (2007)

[140] Goldreich, O., Goldwasser, S., Micali, S.: On the cryptographic applications of random functions. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO'84. LNCS, vol. 196, pp. 276–288. Springer, Berlin, Heidelberg (Aug 1984)

[141] Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. Journal of the ACM 33(4), 792–807 (Oct 1986)

[142] Goldwasser, S., Micali, S.: Probabilistic Encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)

[143] Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing 17(2), 281–308 (Apr 1988)

[144] Goubin, L., Patarin, J.: DES and differential power analysis (the "duplication" method). In: Koç, Çetin Kaya., Paar, C. (eds.) CHES'99. LNCS, vol. 1717, pp. 158–172. Springer, Berlin, Heidelberg (Aug 1999)

[145] Goudarzi, D., Rivain, M.: How fast can higher-order masking be in software? In: Coron, J.S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 567–597. Springer, Cham (Apr / May 2017)

[146] Gouget, A., Patarin, J., Toulemonde, A.: (Quantum) cryptanalysis of misty schemes. In: Hong, D. (ed.) ICISC 20. LNCS, vol. 12593, pp. 43–57. Springer, Cham (Dec 2020)

[147] Granger, R., Jovanovic, P., Mennink, B., Neves, S.: Improved masking for tweakable blockciphers with applications to authenticated encryption. In: Fischlin, M., Coron, J.S. (eds.) Advances in Cryptology - EUROCRYPT 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9665, pp. 263–293. Springer (2016), http://dx.doi.org/10.1007/978-3-662-49890-3_11

[148] Grassi, L., Naya-Plasencia, M., Schrottenloher, A.: Quantum algorithms for the k-xor problem. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part I. LNCS, vol. 11272, pp. 527–559. Springer, Cham (Dec 2018)

[149] Grochow, T., List, E., Nandi, M.: Dovemac: A tbc-based PRF with smaller state, full security, and high rate. IACR Trans. Symmetric Cryptol. 2019(3), 43–80 (2019)

[150] Groß, H., Mangard, S., Korak, T.: An efficient side-channel protected AES implementation with arbitrary protection order. In: Handschuh, H. (ed.) CT-RSA 2017. LNCS, vol. 10159, pp. 95–112. Springer, Cham (Feb 2017)

[151] Grover, L.K.: A fast quantum mechanical algorithm for database search. In: 28th ACM STOC. pp. 212–219. ACM Press (May 1996)

[152] Gueron, S., Langley, A., Lindell, Y.: AES-GCM-SIV: Nonce Misuse-Resistant Authenticated Encryption. RFC 8452 (Apr 2019), https://www.rfc-editor.org/info/rfc8452

[153] Gueron, S., Lindell, Y.: GCM-SIV: Full nonce misuse-resistant authenticated encryption at under one cycle per byte. In: Ray, I., Li, N., Kruegel, C. (eds.) ACM CCS 2015. pp. 109–119. ACM Press (Oct 2015)

[154] Gunsing, A., Bhaumik, R., Jha, A., Mennink, B., Shen, Y.: Revisiting the indifferentiability of the sum of permutations. Cryptology ePrint Archive, Paper 2023/840 (2023), https://eprint.iacr.org/2023/840, https://eprint.iacr.org/2023/840

[155] Guo, C., Lin, D.: Separating invertible key derivations from non-invertible ones: sequential indifferentiability of 3-round Even-Mansour. Designs, Codes and Cryptography pp. 1–21 (2015), available at http://dx.doi.org/10.1007/s10623-015-0132-0

[156] Guo, C., Lin, D.: Indifferentiability of 3-round even-mansour with random oracle key derivation. IACR Cryptol. ePrint Arch. p. 894 (2016)

[157] Guo, C., Pereira, O., Peters, T., Standaert, F.X.: Authenticated encryption with nonce misuse and physical leakage: Definitions, separation results and first construction - (extended abstract). In: Schwabe, P., Thériault, N. (eds.) LATINCRYPT 2019. LNCS, vol. 11774, pp. 150–172. Springer, Cham (Oct 2019)

[158] Guo, C., Shen, Y., Wang, L., Gu, D.: Beyond-birthday secure domain-preserving PRFs from a single permutation. Designs, Codes and Cryptography (Aug 2018), https://doi.org/10.1007/s10623-018-0528-8

[159] Halderman, J.A., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., Felten, E.W.: Lest we remember: Cold boot attacks on encryption keys. In: van Oorschot, P.C. (ed.) USENIX Security 2008. pp. 45–60. USENIX Association (Jul / Aug 2008)

[160] Halevi, S.: EME*: Extending EME to Handle Arbitrary-Length Messages with Associated Data. In: Canteaut, A., Viswanathan, K. (eds.) Progress in Cryptology - INDOCRYPT 2004. Lecture Notes in Computer Science, vol. 3348, pp. 315–327. Springer (2004)

[161] Halevi, S.: Invertible Universal Hashing and the TET Encryption Mode. In: Menezes, A. (ed.) Advances in Cryptology - Crypto 2007. Lecture Notes in Computer Science, vol. 4622, pp. 412–429. Springer (2007)

[162] Halevi, S.: Invertible universal hashing and the TET encryption mode. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 412–429. Springer, Berlin, Heidelberg (Aug 2007)

[163] Halevi, S., Rogaway, P.: A Tweakable Enciphering Mode. In: Boneh, D. (ed.) Advances in Cryptology - Crypto 2003. Lecture Notes in Computer Science, vol. 2729, pp. 482–499. Springer (2003)

[164] Halevi, S., Rogaway, P.: A Parallelizable Enciphering Mode. In: Okamoto, T. (ed.) Topics in Cryptology - CT-RSA 2004. Lecture Notes in Computer Science, vol. 2964, pp. 292–304. Springer (2004)

[165] Hall, C., Wagner, D., Kelsey, J., Schneier, B.: Building PRFs from PRPs. In: Krawczyk, H. (ed.) Advances in Cryptology - CRYPTO '98. Lecture Notes in Computer Science, vol. 1462, pp. 370–389. Springer (1998)

[166] Herbst, C., Oswald, E., Mangard, S.: An AES smart card implementation resistant to power analysis attacks. In: Zhou, J., Yung, M., Bao, F. (eds.) ACNS 06International Conference on Applied Cryptography and Network Security. LNCS, vol. 3989, pp. 239–252. Springer, Berlin, Heidelberg (Jun 2006)

[167] Hirose, S.: Some plausible constructions of double-block-length hash functions. In: Robshaw, M.J.B. (ed.) FSE 2006. LNCS, vol. 4047, pp. 210–225. Springer, Berlin, Heidelberg (Mar 2006)

[168] Hirose, S., Park, J.H., Yun, A.: A simple variant of the Merkle-Damgård scheme with a permutation. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 113–129. Springer, Berlin, Heidelberg (Dec 2007)

[169] Hoang, V.T., Rogaway, P.: On generalized Feistel networks. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 613–630. Springer, Berlin, Heidelberg (Aug 2010)

[170] Hoang, V.T., Rogaway, P.: On Generalized Feistel Networks. In: Rabin, T. (ed.) Advances in Cryptology - CRYPTO 2010. Lecture Notes in Computer Science, vol. 6223, pp. 613–630. Springer (2010)

[171] Hoang, V.T., Tessaro, S.: Key-Alternating Ciphers and Key-Length Extension: Exact Bounds and Multi-user Security. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 (Proceedings, Part I). Lecture Notes in Computer Science, vol. 9814, pp. 3–32. Springer (2016)

[172] Hoang, V.T., Tessaro, S.: Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 3–32. Springer, Berlin, Heidelberg (Aug 2016)

[173] Hodges, P., Stebila, D.: Algorithm Substitution Attacks: State Reset Detection and Asymmetric Modifications. IACR Trans. Symmetric Cryptol. 2021(2), 389–422 (2021)

[174] Hosoyamada, A., Iwata, T.: 4-round Luby-Rackoff construction is a qPRP. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part I. LNCS, vol. 11921, pp. 145–174. Springer, Cham (Dec 2019)

[175] Hosoyamada, A., Iwata, T.: 4-round Luby-Rackoff construction is a qPRP. Cryptology ePrint Archive, Report 2019/243 (2019), https://eprint.iacr.org/2019/243

[176] Hosoyamada, A., Iwata, T.: On tight quantum security of HMAC and NMAC in the quantum random oracle model. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part I. LNCS, vol. 12825, pp. 585–615. Springer, Cham, Virtual Event (Aug 2021)

[177] Hosoyamada, A., Iwata, T.: Provably quantum-secure tweakable block ciphers. IACR Trans. Symm. Cryptol. 2021(1), 337–377 (2021)

[178] Hosoyamada, A., Sasaki, Y., Xagawa, K.: Quantum multicollision-finding algorithm. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 179–210. Springer, Cham (Dec 2017)

[179] Hosoyamada, A., Yasuda, K.: Building quantum-one-way functions from block ciphers: Davies-Meyer and Merkle-Damgård constructions. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part I. LNCS, vol. 11272, pp. 275–304. Springer, Cham (Dec 2018)

[180] Ishai, Y., Sahai, A., Wagner, D.: Private circuits: Securing hardware against probing attacks. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 463–481. Springer, Berlin, Heidelberg (Aug 2003)

[181] Iwata, T., Kurosawa, K.: On the Pseudorandomness of the AES Finalists - RC6 and Serpent. In: Schneier, B. (ed.) Fast Software Encryption - FSE 2000. Lecture Notes in Computer Science, vol. 1978, pp. 231–243. Springer (2000)

[182] Iwata, T., Minematsu, K.: Stronger security variants of GCM-SIV. IACR Trans. Symm. Cryptol. 2016(1), 134–157 (2016), https://tosc.iacr.org/index.php/ToSC/article/view/539

[183] Iwata, T., Minematsu, K., Peyrin, T., Seurin, Y.: ZMAC: A fast tweakable block cipher mode for highly secure message authentication. In: Advances in Cryptology - CRYPTO 2017, Proceedings, Part III. pp. 34–65 (2017)

[184] Jakobsen, J., Orlandi, C.: On the CCA (in)Security of MTProto. In: Security and Privacy in Smartphones and Mobile Devices – SPSM@CCS 2016, Proceedings. pp. 113–116 (2016)

[185] Jean, J.: TikZ for Cryptographers. https://www.iacr.org/authors/tikz/ (2016)

[186] Jean, J., Nikolic, I., Peyrin, T.: Tweaks and Keys for Block Ciphers: The TWEAKEY Framework. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology - ASIACRYPT 2014 (Proceedings, Part II). Lecture Notes in Computer Science, vol. 8874, pp. 274–288. Springer (2014)

[187] Jean, J., Nikolic, I., Peyrin, T., Seurin, Y.: The deoxys AEAD family. Journal of Cryptology 34(3), 31 (Jul 2021)

[188] Jean, J., Nikolić, I., Peyrin, T.: Tweaks and Keys for Block Ciphers: The TWEAKEY Framework, pp. 274–288. Springer Berlin Heidelberg (2014)

[189] Jha, A., List, E., Minematsu, K., Mishra, S., Nandi, M.: XHX - A framework for optimally secure tweakable block ciphers from classical block ciphers and universal hashing. In: Progress in Cryptology - LATINCRYPT 2017, Revised Selected Papers. pp. 207–227 (2017)

[190] Jha, A., Nandi, M.: On rate-1 and beyond-the-birthday bound secure online ciphers using tweakable block ciphers. Cryptography and Communications 10(5), 731–753 (2018)

[191] Jha, A., Nandi, M.: Tight security of cascaded LRW2. J. Cryptol. 33(3), 1272–1317 (2020)

[192] Kahn, D.: The Codebreakers. Scribner, New York, 2nd ed. edn. (1996), description based on publisher supplied metadata and other sources.

[193] Kaliski, B.: PKCS #5: Password-Based Cryptography Specification Version 2.0. RFC 2898 (Sep 2000), https://www.rfc-editor.org/info/rfc2898

[194] Kang, J.S., Yi, O., Hong, D., Cho, H.S.: Pseudorandomness of MISTY-type transformations and the block cipher KASUMI. In: Varadharajan, V., Mu, Y. (eds.) ACISP 01. LNCS, vol. 2119, pp. 60–73. Springer, Berlin, Heidelberg (Jul 2001)

[195] Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking symmetric cryptosystems using quantum period finding. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 207–237. Springer, Berlin, Heidelberg (Aug 2016)

[196] Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Quantum differential and linear cryptanalysis. IACR Trans. Symm. Cryptol. 2016(1), 71–94 (2016), https://tosc.iacr.org/index.php/ToSC/article/view/536

[197] Katagi, M.: The 128-Bit Blockcipher CLEFIA. RFC 6114 (Mar 2011), https://www.rfc-editor.org/info/rfc6114

[198] Katz, J., Lindell, Y.: Introduction to Modern Cryptography. Chapman and Hall/CRC Press, first edn. (2007)

[199] Knudsen, L.R., Rijmen, V.: Known-Key Distinguishers for Some Block Ciphers. In: Kurosawa, K. (ed.) Advances in Cryptology - ASIACRYPT 2007. Lecture Notes in Computer Science, vol. 4833, pp. 315–324. Springer (2007)

[200] Knudsen, L.R., Robshaw, M.J.: The Block Cipher Companion. Springer Berlin Heidelberg (2011)

[201] Koblitz, N.: Elliptic curve cryptosystems. Mathematics of Computation 48(177), 203–209 (1987)

[202] Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) CRYPTO'96. LNCS, vol. 1109, pp. 104–113. Springer, Berlin, Heidelberg (Aug 1996)

[203] Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M.J. (ed.) CRYPTO'99. LNCS, vol. 1666, pp. 388–397. Springer, Berlin, Heidelberg (Aug 1999)

[204] Krawczyk, H.: LFSR-based Hashing and Authentication. In: Desmedt, Y. (ed.) Advances in Cryptology - CRYPTO '94. Lecture Notes in Computer Science, vol. 839, pp. 129–139. Springer (1994)

[205] Krovetz, T., Rogaway, P.: The software performance of authenticated-encryption modes. In: Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 306–327. Springer, Berlin, Heidelberg (Feb 2011)

[206] Kuwakado, H., Morii, M.: Quantum distinguisher between the 3-round feistel cipher and the random permutation. In: IEEE International Symposium on Information Theory, ISIT 2010, Proceedings. pp. 2682–2685. IEEE (2010)

[207] Lai, X.: On the Design and Security of Block Ciphers. Ph.D. thesis, ETH Zürich (1992)

[208] Lai, X., Massey, J.L.: A proposal for a new block encryption standard. In: Damgård, I. (ed.) EUROCRYPT'90. LNCS, vol. 473, pp. 389–404. Springer, Berlin, Heidelberg (May 1991)

[209] Lampe, R., Patarin, J., Seurin, Y.: An Asymptotically Tight Security Analysis of the Iterated Even-Mansour Cipher. In: Wang, X., Sako, K. (eds.) Advances in

Cryptology - ASIACRYPT 2012. Lecture Notes in Computer Science, vol. 7658, pp. 278–295. Springer (2012)

[210] Lampe, R., Seurin, Y.: How to Construct an Ideal Cipher from a Small Set of Public Permutations. In: Sako, K., Sarkar, P. (eds.) Advances in Cryptology - ASIACRYPT 2013 (Proceedings, Part I). Lecture Notes in Computer Science, vol. 8269, pp. 444–463. Springer (2013), full version available at `http://eprint.iacr.org/2013/255`

[211] Lampe, R., Seurin, Y.: Tweakable Blockciphers with Asymptotically Optimal Security. In: Moriai, S. (ed.) Fast Software Encryption - FSE 2013. Lecture Notes in Computer Science, vol. 8424, pp. 133–151. Springer (2013)

[212] Lampe, R., Seurin, Y.: Security Analysis of Key-Alternating Feistel Ciphers. In: Cid, C., Rechberger, C. (eds.) Fast Software Encryption - FSE 2014. Lecture Notes in Computer Science, vol. 8540, pp. 243–264. Springer (2014)

[213] Lampe, R., Seurin, Y.: Tweakable blockciphers with asymptotically optimal security. In: Moriai, S. (ed.) FSE 2013. LNCS, vol. 8424, pp. 133–151. Springer, Berlin, Heidelberg (Mar 2014)

[214] Landecker, W., Shrimpton, T., Terashima, R.S.: Tweakable Blockciphers with Beyond Birthday-Bound Security. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology - CRYPTO 2012. Lecture Notes in Computer Science, vol. 7417, pp. 14–30. Springer (2012), full version available at `http://eprint.iacr.org/2012/450`

[215] Len, J., Grubbs, P., Ristenpart, T.: Partitioning oracle attacks. In: Bailey, M., Greenstadt, R. (eds.) USENIX Security 2021. pp. 195–212. USENIX Association (Aug 2021)

[216] Liskov, M., Rivest, R.L., Wagner, D.: Tweakable block ciphers. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 31–46. Springer, Berlin, Heidelberg (Aug 2002)

[217] Liskov, M., Rivest, R.L., Wagner, D.A.: Tweakable block ciphers. J. Crypto. 24(3), 588–613 (2011)

[218] Liu, C., Chen, R., Wang, Y., Wang, Y.: Asymmetric Subversion Attacks on Signature Schemes. In: Information Security and Privacy – ACISP 2018, Proceedings. pp. 376–395 (2018)

[219] Luby, M., Rackoff, C.: How to construct pseudo-random permutations from pseudo-random functions (abstract). In: Williams, H.C. (ed.) CRYPTO'85. LNCS, vol. 218, p. 447. Springer, Berlin, Heidelberg (Aug 1986)

[220] Luby, M., Rackoff, C.: How to construct pseudorandom permutations from pseudorandom functions. SIAM Journal on Computing 17(2) (1988)

[221] Luby, M., Rackoff, C.: How to Construct Pseudorandom Permutations from Pseudorandom Functions. SIAM Journal on Computing 17(2), 373–386 (1988)

[222] Lynch, B.: Why journalists and dissidents turn to telegram. Index On Censorship (2021), https://www.indexoncensorship.org/2021/06/telegram/

[223] Majenz, C., Malavolta, G., Walter, M.: Permutation superposition oracles for quantum query lower bounds (2024)

[224] Mandal, A., Patarin, J., Seurin, Y.: On the Public Indifferentiability and Correlation Intractability of the 6-Round Feistel Construction. In: Cramer, R. (ed.) Theory of Cryptography Conference - TCC 2012. Lecture Notes in Computer Science, vol. 7194, pp. 285–302. Springer (2012), full version available at http://eprint.iacr.org/2011/496

[225] Mangard, S.: Hardware countermeasures against DPA – A statistical analysis of their effectiveness. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 222–235. Springer, Berlin, Heidelberg (Feb 2004)

[226] Mangard, S., Popp, T., Gammel, B.M.: Side-channel leakage of masked CMOS gates. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 351–365. Springer, Berlin, Heidelberg (Feb 2005)

[227] Mao, S., Zhang, Z., Hu, L., Li, L., Wang, P.: Quantum security of TNT. Cryptology ePrint Archive, Paper 2023/1280 (2023), https://eprint.iacr.org/2023/1280, https://eprint.iacr.org/2023/1280

[228] Matsui, M.: The first experimental cryptanalysis of the data encryption standard. In: Desmedt, Y. (ed.) CRYPTO'94. LNCS, vol. 839, pp. 1–11. Springer, Berlin, Heidelberg (Aug 1994)

[229] Matsui, M.: New block encryption algorithm MISTY. In: Biham, E. (ed.) FSE'97. LNCS, vol. 1267, pp. 54–68. Springer, Berlin, Heidelberg (Jan 1997)

[230] Maurer, U.M.: A Simplified and Generalized Treatment of Luby-Rackoff Pseudo-random Permutation Generator. In: Rueppel, R.A. (ed.) Advances in Cryptology - EUROCRYPT '92. Lecture Notes in Computer Science, vol. 658, pp. 239–255. Springer (1992)

[231] Maurer, U.M., Pietrzak, K.: The Security of Many-Round Luby-Rackoff Pseudo-Random Permutations. In: Biham, E. (ed.) Advances in Cryptology - EUROCRYPT 2003. Lecture Notes in Computer Science, vol. 2656, pp. 544–561. Springer (2003)

[232] Maurer, U.M., Renner, R., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer, Berlin, Heidelberg (Feb 2004)

[233] McGrew, D., Bailey, D.: AES-CCM Cipher Suites for Transport Layer Security (TLS). RFC 6655 (Jul 2012), https://www.rfc-editor.org/info/rfc6655

[234] Medwed, M., Standaert, F.X., Großschädl, J., Regazzoni, F.: Fresh re-keying: Security against side-channel and fault attacks for low-cost devices. In: Bernstein, D.J., Lange, T. (eds.) AFRICACRYPT 10. LNCS, vol. 6055, pp. 279–296. Springer, Berlin, Heidelberg (May 2010)

[235] Menezes, A., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press (1996)

[236] Mennink, B.: Optimally Secure Tweakable Blockciphers. In: Leander, G. (ed.) Fast Software Encryption - FSE 2015. Lecture Notes in Computer Science, vol. 9054, pp. 428–448. Springer (2015), full version available at `http://eprint.iacr.org/2015/363`

[237] Mennink, B.: Optimally secure tweakable blockciphers. IACR Cryptology ePrint Archive 2015, 363 (2015)

[238] Mennink, B.: Towards tight security of cascaded LRW2. In: Theory of Cryptography - TCC 2018, Proceedings, Part II. pp. 192–222 (2018)

[239] Mennink, B., Neves, S.: Encrypted Davies-Meyer and its dual: Towards optimal security using mirror theory. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part III. LNCS, vol. 10403, pp. 556–583. Springer, Cham (Aug 2017)

[240] Merkle, R.C.: A certified digital signature. In: Brassard, G. (ed.) CRYPTO'89. LNCS, vol. 435, pp. 218–238. Springer, New York (Aug 1990)

[241] Micali, S., Reyzin, L.: Physically observable cryptography (extended abstract). In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Berlin, Heidelberg (Feb 2004)

[242] Miles, E., Viola, E.: Substitution-permutation networks, pseudorandom functions, and natural proofs. J. ACM 62(6), 46:1–46:29 (Dec 2015)

[243] Minematsu, K.: Improved Security Analysis of XEX and LRW Modes. In: Biham, E., Youssef, A.M. (eds.) Selected Areas in Cryptography - SAC 2006. Lecture Notes in Computer Science, vol. 4356, pp. 96–113. Springer (2006)

[244] Minematsu, K.: Beyond-Birthday-Bound Security Based on Tweakable Block Cipher. In: Dunkelman, O. (ed.) Fast Software Encryption - FSE 2009. Lecture Notes in Computer Science, vol. 5665, pp. 308–326. Springer (2009)

[245] Minematsu, K., Iwata, T.: Tweak-Length Extension for Tweakable Blockciphers. In: Groth, J. (ed.) Cryptography and Coding - IMACC 2015. Lecture Notes in Computer Science, vol. 9496, pp. 77–93. Springer (2015)

[246] Mitsuda, A., Iwata, T.: Tweakable Pseudorandom Permutation from Generalized Feistel Structure. In: Baek, J., Bao, F., Chen, K., Lai, X. (eds.) ProvSec 2008. Lecture Notes in Computer Science, vol. 5324, pp. 22–37. Springer (2008)

[247] Morris, B., Rogaway, P., Stegers, T.: How to encipher messages on a small domain. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 286–302. Springer, Berlin, Heidelberg (Aug 2009)

[248] Nachef, V., Patarin, J., Treger, J.: Generic attacks on misty schemes. In: Abdalla, M., Barreto, P.S.L.M. (eds.) Progress in Cryptology - LATINCRYPT 2010, Proceedings. Lecture Notes in Computer Science, vol. 6212, pp. 222–240. Springer (2010)

[249] Naito, Y.: Full prf-secure message authentication code based on tweakable block cipher. In: Provable Security - ProvSec 2015, Proceedings. pp. 167–182 (2015)

[250] Naito, Y.: Optimally indifferentiable double-block-length hashing without post-processing and with support for longer key than single block. In: Schwabe, P., Thériault, N. (eds.) LATINCRYPT 2019. LNCS, vol. 11774, pp. 65–85. Springer, Cham (Oct 2019)

[251] Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. In: 38th FOCS. pp. 458–467. IEEE Computer Society Press (Oct 1997)

[252] Naor, M., Reingold, O.: From unpredictability to indistinguishability: A simple construction of pseudo-random functions from MACs (extended abstract). In: Krawczyk, H. (ed.) CRYPTO'98. LNCS, vol. 1462, pp. 267–282. Springer, Berlin, Heidelberg (Aug 1998)

[253] Naor, M., Reingold, O.: Synthesizers and their application to the parallel construction of pseudo-random functions. Journal of Computer and System Sciences 58(2), 336–375 (1999)

[254] National Institute of Standards and Technology: FIPS 113: Computer Data Authentication (May 1985), withdrawn on September 01, 2008.

[255] National Institute of Standards and Technology: FIPS 180-1: Secure Hash Standard (April 1995)

[256] Nielsen, J.B.: Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 111–126. Springer, Berlin, Heidelberg (Aug 2002)

[257] Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press (Jun 2012)

[258] Nikova, S., Rijmen, V., Schläffer, M.: Secure hardware implementation of nonlinear functions in the presence of glitches. Journal of Cryptology 24(2), 292–321 (Apr 2011)

[259] Patarin, J.: Pseudorandom Permutations Based on the DES Scheme. In: Cohen, G.D., Charpin, P. (eds.) EUROCODE '90. Lecture Notes in Computer Science, vol. 514, pp. 193–204. Springer (1990)

[260] Patarin, J.: Security of Random Feistel Schemes with 5 or More Rounds. In: Franklin, M.K. (ed.) Advances in Cryptology - CRYPTO 2004. Lecture Notes in Computer Science, vol. 3152, pp. 106–122. Springer (2004)

[261] Patarin, J.: A proof of security in $o(2^n)$ for the benes scheme. In: Progress in Cryptology - AFRICACRYPT 2008, Proceedings. pp. 209–220 (2008)

[262] Patarin, J.: The "Coefficients H" Technique. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) Selected Areas in Cryptography - SAC 2008. Lecture Notes in Computer Science, vol. 5381, pp. 328–345. Springer (2008)

[263] Patarin, J.: The "Coefficients H" Technique, pp. 328–345. Springer Berlin Heidelberg (2009)

[264] Patarin, J.: Security of balanced and unbalanced Feistel schemes with linear non equalities. Cryptology ePrint Archive, Report 2010/293 (2010), https://eprint.iacr.org/2010/293

[265] Peters, T., Shen, Y., Standaert, F.X.: Multiplex: TBC-based authenticated encryption with sponge-like rate. Cryptology ePrint Archive, Paper 2024/294 (2024), https://eprint.iacr.org/2024/294, https://eprint.iacr.org/2024/294

[266] Petit, C., Standaert, F.X., Pereira, O., Malkin, T., Yung, M.: A block cipher based pseudo random number generator secure against side-channel key recovery. In: Abe, M., Gligor, V. (eds.) ASIACCS 08. pp. 56–65. ACM Press (Mar 2008)

[267] Peyrin, T., Seurin, Y.: Counter-in-tweak: Authenticated encryption modes for tweakable block ciphers. In: Advances in Cryptology - CRYPTO 2016, Proceedings, Part I. pp. 33–63 (2016)

[268] Pietrzak, K.: A leakage-resilient mode of operation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 462–482. Springer, Berlin, Heidelberg (Apr 2009)

[269] Pietrzak, K., Sjödin, J.: Range extension for weak PRFs; the good, the bad, and the ugly. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 517–533. Springer, Berlin, Heidelberg (May 2007)

[270] Preneel, B., Govaerts, R., Vandewalle, J.: Hash Functions Based on Block Ciphers: A Synthetic Approach. In: Stinson, D.R. (ed.) Advances in Cryptology - CRYPTO '93. Lecture Notes in Computer Science, vol. 773, pp. 368–378. Springer (1993)

[271] Procter, G.: A Note on the CLRW2 Tweakable Block Cipher Construction. IACR Cryptology ePrint Archive, Report 2014/111 (2014), available at http://eprint.iacr.org/2014/111

[272] Reinsel, D., Gantz, J., Rydning, J.: The Digitization of the World: From Edge to Core. IDC White Paper (2018)

[273] Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446 (Aug 2018), https://www.rfc-editor.org/info/rfc8446

[274] Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the Association for Computing Machinery 21(2), 120–126 (Feb 1978)

[275] Rogaway, P.: Authenticated-encryption with associated-data. In: Atluri, V. (ed.) ACM Conference on Computer and Communications Security - CCS 2002. pp. 98–107. ACM (2002)

[276] Rogaway, P.: Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In: Advances in Cryptology - ASIACRYPT 2004, Proceedings. pp. 16–31 (2004)

[277] Rogaway, P.: Nonce-based symmetric encryption. In: Roy, B.K., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 348–359. Springer, Berlin, Heidelberg (Feb 2004)

[278] Rogaway, P.: Nonce-Based Symmetric Encryption. In: Roy, B.K., Meier, W. (eds.) Fast Software Encryption - FSE 2004. Lecture Notes in Computer Science, vol. 3017, pp. 348–359. Springer (2004)

[279] Rogaway, P., Shrimpton, T.: Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In: Roy, B.K., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 371–388. Springer, Berlin, Heidelberg (Feb 2004)

[280] Rogaway, P., Shrimpton, T.: A Provable-Security Treatment of the Key-Wrap Problem. In: Vaudenay, S. (ed.) Advances in Cryptology - EUROCRYPT 2006. Lecture Notes in Computer Science, vol. 4004, pp. 373–390. Springer (2006)

[281] Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 373–390. Springer, Berlin, Heidelberg (May / Jun 2006)

[282] Rogaway, P., Zhang, H.: Online Ciphers from Tweakable Blockciphers. In: Kiayias, A. (ed.) Topics in Cryptology - CT-RSA 2011. Lecture Notes in Computer Science, vol. 6558, pp. 237–249. Springer (2011)

[283] Russell, A., Tang, Q., Yung, M., Zhou, H.: Cliptography: Clipping the Power of Kleptographic Attacks. In: Advances in Cryptology – ASIACRYPT 2016, Proceedings, Part II. pp. 34–64 (2016)

[284] Russell, A., Tang, Q., Yung, M., Zhou, H.: Generic Semantic Security against a Kleptographic Adversary. In: Computer and Communications Security – ACM-CCS 2017, Proceedings. pp. 907–922 (2017)

[285] Russell, A., Tang, Q., Yung, M., Zhou, H.: Correcting Subverted Random Oracles. In: Advances in Cryptology – CRYPTO 2018, Proceedings, Part II. pp. 241–271 (2018)

[286] Salowey, J.A., McGrew, D., Choudhury, A.: AES Galois Counter Mode (GCM) Cipher Suites for TLS. RFC 5288 (Aug 2008), https://www.rfc-editor.org/info/rfc5288

[287] Sarkar, P.: Improving upon the TET mode of operation. In: Nam, K.H., Rhee, G. (eds.) ICISC 07. LNCS, vol. 4817, pp. 180–192. Springer, Berlin, Heidelberg (Nov 2007)

[288] Schneier, B.: Description of a new variable-length key, 64-bit block cipher (Blowfish). In: Anderson, R.J. (ed.) FSE'93. LNCS, vol. 809, pp. 191–204. Springer, Berlin, Heidelberg (Dec 1994)

[289] Schroeppel, R.: The Hasty Pudding Cipher. AES submission to NIST (1998), https://www.princeton.edu/~rblee/HPC/index.htm

[290] Shannon, C., Weaver, W.: The Mathematical Theory of Communication. University of Illinois Press (1998), https://books.google.de/books?id=IZ77BwAAQBAJ

[291] Shannon, C.E.: Communication theory of secrecy systems. Bell Systems Technical Journal 28(4), 656–715 (1949)

[292] Shen, Y., Peters, T., Standaert, F.X., Cassiers, G., Verhamme, C.: Triplex: an efficient and one-pass leakage-resistant mode of operation. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 135–162 (Aug 2022)

[293] Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing 26(5), 1484–1509 (Oct 1997)

[294] Shoup, V.: OAEP reconsidered. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 239–259. Springer, Berlin, Heidelberg (Aug 2001)

[295] Simmons, G.J.: The Prisoners' Problem and the Subliminal Channel. In: Advances in Cryptology – CRYPTO 1983, Proceedings. pp. 51–67 (1983)

[296] Simon, D.R.: On the power of quantum computation. In: 35th FOCS. pp. 116–123. IEEE Computer Society Press (Nov 1994)

[297] Simon, D.R.: Finding Collisions on a One-Way Street: Can Secure Hash Functions Be Based on General Assumptions? In: Nyberg, K. (ed.) Advances in Cryptology - EUROCRYPT '98. Lecture Notes in Computer Science, vol. 1403, pp. 334–345. Springer (1998)

[298] Song, F., Yun, A.: Quantum security of NMAC and related constructions - PRF domain extension against quantum attacks. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 283–309. Springer, Cham (Aug 2017)

[299] Standaert, F.X., Pereira, O., Yu, Y.: Leakage-resilient symmetric cryptography under empirically verifiable assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 335–352. Springer, Berlin, Heidelberg (Aug 2013)

[300] Steinberger, J.: Improved Security Bounds for Key-Alternating Ciphers via Hellinger Distance. IACR Cryptology ePrint Archive, Report 2012/481 (2012), available at http://eprint.iacr.org/2012/481

[301] Stevens, M., Bursztein, E., Karpman, P., Albertini, A., Markov, Y.: The First Collision for Full SHA-1. In: Advances in Cryptology – CRYPTO 2017, Proceedings, Part I. pp. 570–596 (2017)

[302] Struck, P., Weishäupl, M.: Constructing committing and leakage-resilient authenticated encryption. IACR Trans. Symm. Cryptol. 2024(1), 497–528 (2024)

[303] Struck, P., Weishäupl, M.: Constructing committing and leakage-resilient authenticated encryption. Cryptology ePrint Archive, Paper 2024/190 (2024), https://eprint.iacr.org/2024/190, https://eprint.iacr.org/2024/190

[304] The Telegram Team: FAQ for the Technically Inclined – What about IND-CCA? Telegram website (2024), online: https://core.telegram.org/techfaq/mtproto_v1#what-about-ind-cca. Accessed: June 18, 2024

[305] The Telegram Team: FAQ for the Technically Inclined – Why did you use a custom protocol? Telegram website (2024), online: https://core.telegram.org/techfaq#q-why-did-you-go-for-a-custom-protocol. Accessed: June 18, 2024

[306] The Telegram Team: Mobile Protocol. Telegram website (2024), online: https://core.telegram.org/mtproto. Accessed: June 18, 2024

[307] The Telegram Team: Mobile Protocol: Detailed Description (v.1.0, DEPRECATED). Telegram website (2024), online: https://core.telegram.org/mtproto/description_v1. Accessed: June 18, 2024

[308] Tiri, K., Verbauwhede, I.: Securing encryption algorithms against DPA at the logic level: Next generation smart card technology. In: Walter, C.D., Koç, Çetin Kaya., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 125–136. Springer, Berlin, Heidelberg (Sep 2003)

[309] Vaudenay, S.: Decorrelation: A Theory for Block Cipher Security. Journal of Cryptology 16(4), 249–286 (2003)

[310] Veyrat-Charvillon, N., Medwed, M., Kerckhof, S., Standaert, F.X.: Shuffling against side-channel attacks: A comprehensive study with cautionary note. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 740–757. Springer, Berlin, Heidelberg (Dec 2012)

[311] Wang, P., Feng, D., Wu, W.: The Security of the Extended Codebook (XCB) Mode of Operation. In: Feng, D., Lin, D., Yung, M. (eds.) CISC 2005: Information Security and Cryptology. Lecture Notes in Computer Science, vol. 3822, pp. 175–188. Springer (2005)

[312] Wegman, M.N., Carter, J.L.: New classes and applications of hash functions. In: 20th Annual Symposium on Foundations of Computer Science (sfcs 1979). IEEE (Oct 1979)

[313] Wegman, M.N., Carter, L.: New hash functions and their use in authentication and set equality. Journal of Computer and System Sciences 22, 265–279 (1981)

[314] Whiting, D., Housley, R., Ferguson, N.: Counter with CBC-MAC (CCM). No. 3610 in Request for Comments, RFC Editor (Sep 2003), https://www.rfc-editor.org/info/rfc3610, available at http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/ccm/ccm.pdf

[315] Wootters, W.K., Zurek, W.H.: A single quantum cannot be cloned. Nature 299(5886), 802–803 (Oct 1982)

[316] Xu, S., Da, Q., Guo, C.: Minimizing even-mansour ciphers for sequential indifferentiability (without key schedules). In: Progress in Cryptology - INDOCRYPT 2022, Proceedings. pp. 125–145 (2022)

[317] Yao, A.C.: Theory and Applications of Trapdoor Functions (Extended Abstract). In: Symposium on Foundations of Computer Science - FOCS '82. pp. 80–91. IEEE Computer Society (1982)

[318] Yoneyama, K., Miyagawa, S., Ohta, K.: Leaky Random Oracle. IEICE Transactions 92-A(8), 1795–1807 (2009)

[319] Young, A.L., Yung, M.: The Dark Side of "Black-Box" Cryptography, or: Should We Trust Capstone? In: Advances in Cryptology – CRYPTO 1996, Proceedings. pp. 89–103 (1996)

[320] Young, A.L., Yung, M.: Kleptography: Using Cryptography Against Cryptography. In: Advances in Cryptology – EUROCRYPT 1997, Proceeding. pp. 62–74 (1997)

[321] Yu, Y., Standaert, F.X.: Practical leakage-resilient pseudorandom objects with minimum public randomness. In: Dawson, E. (ed.) CT-RSA 2013. LNCS, vol. 7779, pp. 223–238. Springer, Berlin, Heidelberg (Feb / Mar 2013)

[322] Yu, Y., Standaert, F.X., Pereira, O., Yung, M.: Practical leakage-resilient pseudorandom generators. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM CCS 2010. pp. 141–151. ACM Press (Oct 2010)

[323] Zhandry, M.: How to construct quantum random functions. In: 53rd FOCS. pp. 679–687. IEEE Computer Society Press (Oct 2012)

[324] Zhandry, M.: A note on the quantum collision and set equality problems. Quantum Information and Computation 15(7 & 8), 557–567 (may 2015)

[325] Zhandry, M.: How to record quantum queries, and applications to quantum indifferentiability. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 239–268. Springer, Cham (Aug 2019)

[326] Zheng, Y., Matsumoto, T., Imai, H.: On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypotheses. In: Brassard, G. (ed.) Advances in Cryptology - CRYPTO '89. Lecture Notes in Computer Science, vol. 435, pp. 461–480. Springer (1989)

This page intentionally left blank.

Saarland University
Department of Computer Science

Provable Security of Symmetric-Key Cryptographic Schemes in Classical and Quantum Frameworks

Jordan Ethan

Saarbrücken, 2024