Design and Analysis of Some Symmetric Key Schemes for Encryption and Authentication

A thesis submitted to the Indian Statistical Institute
in partial fulfillment of the requirements for the Degree of
Doctor of Philosophy in Computer Science

Samir Kundu
Senior Research Fellow

Applied Statistics Unit
Indian Statistical Institute
Kolkata - 700108, India

Dedicated to

To My Family

LIST OF PUBLICATIONS

1. Debrup Chakraborty, Avijit Dutta and **Samir Kundu**, “Designing tweakable enciphering schemes using public permutations.", Advances in Mathematics of Communications, 2023, 17(4): 771-798.

   DOI: https://doi.org/10.3934/amc.2021021.

2. Debrup Chakraborty and **Samir Kundu**, "On the security of TrCBC." Information Processing Letters 179 (2023): 106320.

   DOI: https://doi.org/10.1016/j.ipl.2022.106320

3. Nilanjan Datta, Avijit Dutta and **Samir Kundu**, “Tight Security Bound of 2k-LightMAC_Plus.” Proceedings of 24th International Conference on Cryptology in India, INDOCRYPT 2023. (To Appear)

ACKNOWLEDGEMENT

I would like to take this opportunity to express my sincere gratitude to all those who have contributed to the completion of my PhD thesis.

First and foremost, I am deeply grateful to my supervisor, Dr. Debrup Chakraborty, for his unwavering support, guidance, and mentorship throughout my research journey. His valuable comments, motivating discussions and ideas help a lot in my research work. I am indebted to Dr. Avijit Dutta and Dr. Nilanjan Datta, who have helped me with their noble guidance, support, and encouragement being my senior and collaborator.

I am sincerely grateful to Prof. Bimal Kumar Roy, Prof. Palash Sarkar, Prof. Mridul Nandi and Prof. Subhamoy Maitra, for their invaluable help and guidance.

I thank Avishek, for his support and for motivating me in my low time of the PhD journey. I am also thankful to the former and current members of our research group, including Mostaf, Prabal, Nayana, Subhadip-da, Laltu-da, Amit-da, Diptendu-da, Butu-da, Soumya, Pritam, Biswajit, Suprita, Susanta, Sougata, Aniruddha, Jyotirmoy, Animesh, Chandranan, Gourab, Anup, Suman, Suvo with whom I have always had fruitful discussions and enjoyable times. In addition to the research group, I would like to express my thanks to Anurag, Chotu, Dipayan, Pitamber, Subhankar, Manab, Samir and Sumanta for their support and friendship throughout my journey.

I would like to express my heartfelt gratitude to my entire family for their unwavering support and guidance. My dream of pursuing research would be incomplete without the constant support and encouragement from my parents Mr. Dharani Kundu and Mrs. Kanchan Kundu, sister Laxmi Kundu and uncle Shantimoy Mondal. Last but not the least, I am grateful to Sumana, for her support, which has been a constant source of inspiration for me.

Samir Kundu

Samir Kundu

ABSTRACT

This thesis mainly focuses on the design and analysis of tweakable enciphering schemes (TESs) and message authentication codes (MACs).

Tweakable enciphering schemes are length preserving encryption schemes that provide security of a strong tweakable pseudorandom permutation. There are several constructions of TES using block ciphers as the main cryptographic primitive. Recently, public random permutations have been widely considered as a replacement for block ciphers in several cryptographic schemes, including Authenticated Encryption (AE) schemes, MACs, etc. However, to the best of our knowledge, a systematic study of constructing TESs using public random permutations is missing. We fill this gap by constructing TES using public permutations. We propose two main constructions with several variants. The basic construction, which we call ppTES is generically constructed using a public random permutation, a length expanding pseudorandom function (PRF) based on public random permutations and an almost xor-universal and almost-regular (AXUAR) hash function. We show a concrete instantiation of ppTES and prove its security using the H-Coefficient technique.

ppTES requires both forward and inverse calls to the public random permutation. Most public random permutations are designed with the goal of making the forward calls extremely fast. Thus, a TES construction that does not need computing the inverse of a permutation will have better efficiency. This fact leads us to design a TES that uses a public permutation but does not require the inverse calls to the permutation. We call this construction as IpTES. In addition to a public permutation, IpTES uses an AXUAR hash function. To ensure the inverse free property, we suitably use a two-round Feistel structure. We prove that IpTES is a birthday bound secure public permutation based TES.

The rest of the work is on MACs. TrCBC is a variant of the famous CBC MAC which was proposed by Zhang et al. in 2012. It was claimed that TrCBC is a secure MAC with significant efficiency advantages over other secure variants of CBC. The authors also mentioned the only disadvantage of TrCBC to be the fact that it produces

shorter tags; in particular, it was claimed that TrCBC can only produce secure tags of length less than $n/2$, where n is the block length of the underlying block cipher. We mount a concrete practical attack on TrCBC. We show that with high probability, an adversary can forge TrCBC with tag length $n/2 - 1$ with just three queries. We discuss some general scenarios of our concrete attack and also do a detailed analysis of the authors' security claims of TrCBC.

Next, we study variable output length pseudorandom functions and their use in constructing secure MACs, which can produce tags of varying lengths using the same key. In this regard, we propose a generic construction of converting a fixed output length PRF to a variable output length PRF and discuss its utility in constructing MACs. We also propose some modifications to the famous block cipher based MAC called PMAC to equip it to produce tags of varying lengths.

Finally, we do an extensive study of a newly proposed MAC, 2k-LightMAC_Plus. 2k-LightMAC_Plus was proposed by Datta et al. in FSE 2018, where the author proved that the scheme provides $2n/3$ bits of security. We improve this bound and show that 2k-LightMAC_Plus provably achieves $3n/4$ bit security. We also exhibit a matching attack on the construction and hence establish that our bound is tight. Our proof uses several components of Mirror Theory.

Contents

List of Figures

1 Introduction

This Chapter begins with a token introduction to Cryptography and a description of two broad cryptographic schemes, namely tweakable enciphering schemes and message authentication codes. Finally, we discuss in detail the scope of the thesis and provide a short description of the following Chapters.

1.1 Cryptography and The Objects of Our Interest

In our contemporary digital era, the concept of security primarily revolves around maintaining the confidentiality of information. This means ensuring that digital data transmitted via various channels remains concealed from unintended recipients in the network. It is essential to recognize that the need for confidentiality and security is a social construct. If no individuals were interested in the specific information being safeguarded, the concept of secrecy would lose its relevance. Consequently, we gauge security in qualitative terms, considering the determination and capability of potential adversaries who might be interested in the protected information. Cryptography, at its core, involves the art of concealing information. While a cryptosystem aims to maintain the secrecy of data, adversaries might attempt to break it, either by recovering the actual data or by extracting valuable insights from it. This practice of breaking cryptosystems is widely known as Cryptanalysis. The term Cryptology, derived from the combinations of cryptography and cryptanalysis, represents the scientific field dedicated to studying the art of secure communication.

Cryptography is broadly categorized into two parts: symmetric-key and public-key cryptography. Symmetric-key cryptography employs a single secret key for both encryption and decryption, which must be shared between both the communicating parties. Public-key cryptography uses a public key and a private key per entity. The public key of an entity allows to encrypt the messages for the corresponding entity and decrypt using the private key. Although public-key cryptography simplifies the key distribution problem, symmetric-key cryptography is generally computationally more efficient. As a result, a common approach in the real scenario is that for an initial key exchange, we use public-key cryptography, and for ongoing communication, we use symmetric-key cryptography. This thesis focuses on symmetric-key cryptography.

The two fundamental goals of symmetric-key cryptography are:

- **Confidentiality.** This ensures that any adversary accessing a communication channel cannot derive information about the content of messages exchanged between communicating parties. It maintains the privacy of the message contents.

- **Integrity.** It guarantees that the adversary has not made any unauthorized modifications to the exchanged messages. It prevents active adversaries from tampering with transmitted messages, ensuring the content's integrity.

In the symmetric key setting, confidentiality is achieved through encryption and integrity/authenticity is achieved through message authentication codes. Several kinds of encryption schemes are available in the literature which suit different application areas. In this thesis, we concentrate on a specific type of encryption scheme called a tweakable enciphering scheme. Additionally, this thesis studies message authentication codes. Next, we will provide a non-technical introduction to these cryptographic objects.

Tweakable Enciphering Schemes. A Tweakable Enciphering Scheme, or in short TES, is a deterministic length preserving encryption scheme,i.e., the encryption algorithm is not randomized and the length of the ciphertexts produced by such schemes is the same as the length of the plaintext. TES are secure against adaptive chosen plaintext and chosen ciphertext attacks, i.e., an efficient adversary should be unable to differentiate ciphertexts produced by a TES from random strings and should not be able to manipulate a ciphertext to decrypt into a meaningful message. An encryption scheme always takes as input a plaintext and a key and produces a ciphertext. A TES, in addition to a key and plaintext, takes as input a quantity called a *tweak*. A tweak is a public parameter that is meant to provide variability in the ciphertext, i.e., the same plaintext, when encrypted with the same key but with different tweaks, will produce different ciphertexts. The length preserving feature of TES, along with its usage of tweaks, makes it a suitable candidate for low-level disk encryption. This application for TES was first pointed out in [50]. Later, several constructions of TES have been proposed in the last two decades. An IEEE standard [53] specifies two TES, namely XCB [67, 20] and EME* [51, 48] as standards for disk encryption. A related object is a tweakable block-cipher [62], which is a TES with a fixed (and generally small) block length. TESs are designed to support plaintexts of arbitrary lengths and are sometimes called wide block modes. A more comprehensive survey of the tweakable enciphering schemes is presented in Section 2.6 of **Chapter 2**.

Message Authentication Codes. Message authentication codes, or in short MACs, are the algorithms that provide integrity of a message, i.e., MACs allow the sender to transmit a message in a way that if anyone modifies the messages in transit, then the receiver can detect such modification with high probability. Here, the adversary is an active adversary who can see the message, modify the message in transit and also create new messages. A MAC scheme is a pair of algorithms (*MAC generation algorithm, Verification algorithm*). On a given message and a key *MAC generation algorithm* outputs a *tag*. The sender sends this *tag* along with the message. Now, on a given message, the key and a *tag*, the *Verification algorithm* outputs either 0 or 1, where 1 indicates that the message is authentic and 0 indicates the message is not authentic. There are several paradigms for constructing message authentication codes. The most important properties sought for a MAC are its efficiency and the security that it provides. Until now, new constructions of MACs have been proposed, either more efficient than the previous ones, have a better security margin, or provide a functionality absent in the previous constructions.

1.2 Scope of the Thesis

As already stated, this thesis focuses on designing and analyzing Tweakable Enciphering Schemes(TES) and Message Authentication Codes (MAC)[1]. The thesis is divided into seven chapters. Chapters 3 and 4 deal with TESs, and Chapters 5, 6, and 7 deal with MACs. In this section, we provide a brief overview of the contents of the chapters that follow and also highlight our contributions.

Chapter 2 contains no new material. It introduces the general notation and defines some cryptographic objects which are used throughout the thesis. This Chapter also contains a brief survey of TESs and MACs.

In **Chapter 3**, we construct TESs using public random permutations. Public random permutations are cryptographic objects that, in recent times, have seen wide usage in the construction of different cryptographic schemes like hash functions, authenticated encryption, message authentication codes, etc. Public random permutations are seen as a more efficient alternative to block ciphers in certain scenarios. However, to our knowledge, a systematic study of constructing TES using public random permutations is missing. In this Chapter, we give a generic construction of a TES that uses a public random permutation as the main cryptographic object; we call our construction ppTES. In addition to public random permutations, ppTES uses a length expanding public permutation based pseudorandom function (PRF) and a hash function, which is both almost xor universal and almost regular. Further, we propose a concrete length expanding public permutation based PRF construction. We also propose a single keyed variant of ppTES. We prove the security of all our constructions and provide concrete security bounds. The material presented in this Chapter is based on the paper [18].

[1] All analysis and security claims in this thesis are in the classical setting, i.e., we do not consider quantum adversaries.

Most existing public permutations have the property that they are faster in the forward computation than in the inverse computation. Thus, a construction based on a random permutation will be more efficient if it does not contain calls to the inverse of the permutation. ppTES and its variant described in Chapter 3 requires both forward and inverse calls to the public random permutation. In **Chapter 4** we propose a new public permutation based TES called IpTES, which does not use any inverse call to the permutation. We thoroughly analyze its security and derive its concrete security bound.

In **Chapter 5**, we analyze a message authentication code called TrCBC [88], which was proposed by Zhang et al. in 2012. The authors claimed TrCBC to be a secure message authentication code (MAC) with some interesting properties. If TrCBC is instantiated with a block cipher with block length n, then it requires $\lceil \lambda/n \rceil$ block cipher calls for authenticating a λ-bit message and requires a single key, which is the block cipher key. This is quite interesting, as all known secure variants of CBC-MACs require more block-cipher calls than this. The authors state that TrCBC can have tag lengths of size less than $n/2$. We show a concrete attack on TrCBC. Particularly, we show that with high probability, an adversary can forge TrCBC with tag length $n/2-1$ with just three queries. The attack that we show can be applied to forge a large class of messages. The authors proved TrCBC to be a pseudorandom function (PRF). A scrutiny of the claimed PRF bound shows that for some recommended values of tag lengths, the bound turns out to be quite large. Thus, the security theorem does not imply security of TrCBC for all recommended tag lengths. The contents of this Chapter are based on the paper [21].

The heart of a message authentication code is the tag generation function, which is used to both generate tags and verify them; it is well known that when the tag generation function is a PRF, then the MAC can produce un-forgeable tags. A tag generation function for a deterministic MAC is generally a variable input length and fixed output length PRF. Most deterministic MACs designed with block ciphers, for example, variants of CBC-MAC [4, 54, 72], PMAC [14] etc., are of this type. These MACs are designed to generate tags of fixed length, and their security proofs also consider the tags to be of fixed length. It may be desirable in some scenarios, say for lightweight applications, that the MAC is equipped to produce tags of variable length. Until recently, this aspect of MACs has not been studied. Recently, Ghosh and Sarkar in [47] studied Wegman-Carter type MACs, which can produce tags of variable length MACs. In **Chapter 6** we study deterministic block cipher based MACs, which can produce variable length tags. Specifically, in this Chapter, we construct variable output length PRFs (vlPRF) and show how they can be used to construct variable tag length MACs. We also propose a modification of the PMAC scheme to enable it to securely generate variable length authentication tags.

In **Chapter 7**, we do an improved security analysis of an existing MAC called 2k-LightMAC_Plus. In ASIACRYPT'17, Naito [65] proposed a beyond-birthday-bound variant of the LightMAC construction, called LightMAC_Plus, which is built on three independently keyed n-bit block ciphers, and showed that the construction achieves $2n/3$-bits PRF security. In FSE'18, Datta et al. [37] have proposed a two-keyed variant of the LightMAC_Plus construction, called 2k-LightMAC_Plus, which is built on two independently keyed n-bit block ciphers, and showed that the construction achieves $2n/3$-bits PRF security. We show a tight security bound on the 2k-LightMAC_Plus construction. In particular, we show that it provably achieves security up to $2^{3n/4}$ queries. We also exhibit a matching attack on the construction with the same query complexity, hence establishing the tightness of our security bound. The contents of this Chapter are based on the paper [36].

In **Chapter 8**, we conclude the thesis with a summary of presentations and a discussion on future directions of research.

2 Preliminaries

2.1 Notations

Suppose $\mathcal{X}$ be a finite set then, $X \overset{\$}{\leftarrow} \mathcal{X}$ denotes that X is sampled uniformly at random from $\mathcal{X}$. If $(X_1, \ldots, X_r)$ is a sequence of r random variables than $X_1, \ldots, X_r \overset{\$}{\leftarrow} \mathcal{X}$ denotes that X_i's are independently and uniformly sampled from $\mathcal{X}$. Similarly, we write $X_1, \ldots, X_q \overset{\text{wor}}{\longleftarrow} \{0,1\}^n$ to denote that each X_i is sampled uniformly from $\{0,1\}^n \setminus \{X_1, \ldots, X_{i-1}\}$, i.e., $X_i \overset{\$}{\leftarrow} \{0,1\}^n \setminus \{X_1, \ldots, X_{i-1}\}$. For $q \in \mathbb{N}$, $[q]$ denotes the set $\{1, \ldots, q\}$. For a natural number n, the set of all binary strings of length n is denoted by $\{0,1\}^n$ and $\{0,1\}^{\geq n}$ denotes the set of all binary strings of length at least n. Therefore, $\{0,1\}^{\geq 0}$ is the set of all binary strings of arbitrary length (including the empty string ε) and denoted by $\{0,1\}^*$. For a natural number ℓ, $\{0,1\}^{\leq \ell}$ denotes the set of binary strings of length at most ℓ. An element of $\{0,1\}^n$ is called a *block*. For $x \in \{0,1\}^*$, $|x|$ denotes the length of x in bits. For $s \in \mathbb{N}$, $\mathsf{first}(s, x)$ denotes the first s bits of a binary string x whose length is at least s. For $x, y \in \{0,1\}^*$, $x \| y$ denotes the concatenation of x followed by y. For $x, y \in \{0,1\}^n$, we write $x \oplus y$ to denote their bitwise xor. For any $x \in \{0,1\}^*$, $\mathsf{parse}_n(x)$ parses x as $x_1 \| x_2 \| \ldots \| x_\ell$ where each x_i, for $i \in [\ell - 1]$, is a block and $0 \leq |x_\ell| \leq n$. For any $n \in \mathbb{N}$, we define an injective function pad_n that takes an arbitrary string $x \in \{0,1\}^*$ and returns $y \in (\{0,1\}^n)^*$, defined as follows:

$$\mathsf{pad}_n(x) \overset{\Delta}{=} x \| 10^d,$$

where d is the smallest integer such that $|\mathsf{pad}_n(x)|$ is a multiple of n. For two positive integers i, s such that $i < 2^s$, we write $\langle i \rangle_s$ to denote the s-bit representation of integer i. For $b \in \{0,1\}$, we consider the function fix_b that takes an n-bit binary string x and returns x except its most significant bit is changed to bit b. Similarly, for $b \in \{10, 11\}$, we consider the function fix_b that takes an n-bit binary string x and returns x except its two most significant bits are changed to b. For two pairs of positive integers $(i, j), (i', j') \in \mathbb{Z}^+ \times \mathbb{Z}^+$, we write $(i, j) \preceq (i', j')$ to denote that either $i < i'$ or $(i = i'$ and $j < j')$.

We write a q-tuple $\widetilde{x} = (x_1, \ldots, x_q)$ as $(x_i)_{i \in [q]}$. When all the elements of a tuple $\widetilde{x} = (x_1, \ldots, x_q)$ are distinct, then by abusing of notation, we often write $\widetilde{x}$ as the set $\widetilde{x} = \{x_i : i \in [q]\}$. We write $\mathcal{X}^{(q)}$ to denote the set of all q tuples whose all elements are distinct, i.e.,

$$\mathcal{X}^{(q)} = \{(x_1, \ldots, x_q) : x_i \neq x_j, \forall i \neq j\}.$$

For a sequence of elements $x^1, x^2, \ldots, x^s \in \{0,1\}^*$, we write x^i_a to denote the a-th block of the i-th element x^i. For integers $1 \leq b \leq a$, we write $\mathbf{P}(a, b)$ to denote $a(a-1)\ldots(a-b+1)$, where $\mathbf{P}(a, 0) = 1$ by convention.

The set of all functions from $\mathcal{X}$ to $\mathcal{Y}$ is denoted by $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$. When $\mathcal{Y} = \{0,1\}^n$, then we denote $\mathsf{Func}(\mathcal{X}, \{0,1\}^n)$ simply as $\mathsf{Func}_{\mathcal{X}}(n)$ and sometimes we write $\mathsf{Func}(n)$ by omitting $\mathcal{X}$ when the domain of the function is understood from the context. When $\mathcal{X} = \{0,1\}^n$ and $\mathcal{Y} = \{0,1\}^r$, then we denote $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$ as $\mathsf{Func}(n, r)$. We denote the set of all n bit permutations by $\mathsf{Perm}(n)$.

2.2 Adversary and Advantage

In this section, we discuss about a cryptographic adversary and its distinguishing advantages.

Adversaries and Oracles. A cryptographic adversary $\mathcal{A}$ is a randomized algorithm, who has access to an oracle $\mathcal{O}$. An oracle $\mathcal{O}$ is also an algorithm that provides cryptographic functionality or information within a cryptographic scheme for analysis and security evaluation. The interaction between the adversary $\mathcal{A}$ and the oracle $\mathcal{O}$ generates a set of pairs $\tau = \{(x_1, y_1), (x_2, y_2), \ldots, (x_q, y_q)\}$, where $x_1, x_2, \ldots, x_q$ are the q many queries to oracle $\mathcal{O}$ by the adversary $\mathcal{A}$ and $y_1, y_2, \ldots, y_q$ are the corresponding responses by the oracle $\mathcal{O}$. We consider the adversary as an adaptive adversary, which means the i-th query by the adversary depends on the previous $i-1$ responses.

Distinguishing Advantage. We consider two systems I and R and a distinguishing adversary $\mathcal{A}$. The adversary $\mathcal{A}$ is given access to either I or R. After completing the interaction with an oracle $\mathcal{O}$, $\mathcal{A}$ returns 1, denoted by $\mathcal{A}^{\mathcal{O}} \Rightarrow 1$. This kind of adversary is called a *distinguisher* and the game is called a *distinguishing game*. Now, the goal of the distinguishing adversary or distinguisher is to distinguish between the two systems I and R in a distinguishing game. The distinguishing advantage of the distinguisher is defined as

$$\mathbf{Adv}_{\mathsf{R}}^{\mathsf{I}}(\mathcal{A}) \overset{\Delta}{=} \Big| \Pr[\mathcal{A}^{\mathsf{I}} \Rightarrow 1] - \Pr[\mathcal{A}^{\mathsf{R}} \Rightarrow 1] \Big| ,$$

where the above probability refers to the probability computed over the probability spaces of the adversary $\mathcal{A}$ and the oracle $\mathcal{O}$. If we take the maximum over the advantages for all possible distinguishers $\mathcal{A}$, who make q queries, we get the maximum advantage and is defined as

$$\max_{\mathcal{A}} \mathbf{Adv}_{\mathsf{R}}^{\mathsf{I}}(\mathcal{A}).$$

Adversarial Resources. In the given definition of the distinguishing advantage of the adversary $\mathcal{A}$, the resources utilized by the distinguisher to distinguish algorithms I and R are not explicitly mentioned. However, two main resources commonly considered for adversaries are time complexity and query complexity. The time complexity (t) of an adversary $\mathcal{A}$ includes the time required for interacting with the oracle and the time required for local computations. Query complexity (q) refers to the number of queries made by $\mathcal{A}$ to the oracle. Additionally, data complexity (σ) is the total number of blocks queried by $\mathcal{A}$ to the oracle $\mathcal{O}$. The maximum advantage in distinguishing I and R, considering a class of adversaries with a maximum time complexity of t and a maximum query complexity of q and maximum data complexity σ, is defined as

$$\mathbf{Adv}^{\mathsf{I}}_{\mathsf{R}}(q,t,\sigma) \triangleq \max_{\mathcal{A}} \mathbf{Adv}^{\mathsf{I}}_{\mathsf{R}}(\mathcal{A}),$$

where the maximum is taken over all adversaries that make at most q queries with maximum running time t and maximum data complexity σ.

2.3 Basic Building Blocks

Block Cipher. A block cipher is a function $\mathsf{E} : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$, where $\mathcal{K} = \{0,1\}^k$. So a block cipher takes an n-bit input and produces an n-bit output under the action of a k-bit key; the values of n and k vary for different block ciphers and they are called the block length and key length, respectively. For any $K \in \mathcal{K}$ and $P \in \{0,1\}^n$, we will denote a block cipher by $\mathsf{E}_K(P)$ instead of $\mathsf{E}(K,P)$. It is a requirement that for any $K \in \mathcal{K}$, $\mathsf{E}_K(\cdot)$ must be a permutation, i.e., the function $\mathsf{E}_K : \{0,1\}^n \to \{0,1\}^n$ must be a bijection. If E_K is a bijection then for every $C \in \{0,1\}^n$, there exists only one $P \in \{0,1\}^n$ such that $C = \mathsf{E}_K(P)$. $\mathsf{E}_K(\cdot)$ has an inverse function denoted as $\mathsf{E}_K^{-1}(\cdot)$, such that $P = \mathsf{E}_K^{-1}(\mathsf{E}_K(P))$. A secure block cipher is assumed to be a strong pseudorandom permutation.

Pseudorandom Functions. Let $\mathsf{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a keyed function from $\mathcal{X}$ to $\mathcal{Y}$, which is denoted by $\mathsf{F}(k,x)$ or $\mathsf{F}_k(x)$, where $\mathcal{K}$ is called the key space, $\mathcal{X}$ is called the input space and $\mathcal{Y}$ is called the output space. We view $\mathsf{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ as a family of functions $\{\mathsf{F}_k\}_{k\in\mathcal{K}}$. Now consider a distinguisher $\mathcal{A}$, who has oracle access to either F_K, where $K \xleftarrow{\$} \mathcal{K}$, or a uniform random function from $\mathsf{Func}(\mathcal{X},\mathcal{Y})$. Suppose $\mathcal{A}$ makes at most q queries to its oracle and runs for time at most t. The task of the distinguisher $\mathcal{A}$ is to distinguish if its oracle is the function F_K or a uniform random function from $\mathsf{Func}(\mathcal{X},\mathcal{Y})$. We define the $\underline{\text{P}}$seudo $\underline{\text{R}}$andom $\underline{\text{F}}$unction (PRF) advantage of $\mathcal{A}$ as

$$\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{F}}(\mathcal{A}) \triangleq \mid \Pr[\mathcal{A}^{\mathsf{F}_K} \Rightarrow 1] - \Pr[\mathcal{A}^{\mathsf{RF}} \Rightarrow 1] \mid,$$

where $K \xleftarrow{\$} \mathcal{K}$ and $\mathsf{RF} \xleftarrow{\$} \mathrm{Func}(\mathcal{X}, \mathcal{Y})$. F is said to be a (q, t, ϵ) secure PRF, if

$$\mathbf{Adv}_{\mathsf{F}}^{\mathsf{PRF}}(q,t) \overset{\Delta}{=} \max_{\mathcal{A}} \mathbf{Adv}_{\mathsf{F}}^{\mathsf{PRF}}(\mathcal{A}) \leq \epsilon,$$

where the maximum is taken over all adversaries that make at most q queries and runs for the time at most t.

Pseudorandom Permutation and Strong Pseudorandom Permutation. Let $\mathsf{E} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$ be a keyed bijective function on $\mathcal{X}$, which is denoted by $\mathsf{E}(k, x)$ or $\mathsf{E}_k(x)$, where $\mathcal{K}$ is called the key space, $\mathcal{X}$ is called the domain space. For each key $k \in \mathcal{K}$, the map $\mathsf{E}_k()$ is a permutation over the domain space $\mathcal{X}$. Now consider a distinguisher $\mathcal{A}$, who has oracle access to either E_k where k is chosen uniformly from $\mathcal{K}$ or a permutation chosen uniformly from $\mathsf{Perm}(\mathcal{X})$. Suppose $\mathcal{A}$ makes at most q queries and runs for the time at most t. The task of $\mathcal{A}$ is to distinguish a permutation E from a random permutation. We consider the Pseudo Random Permutation (PRP) advantage of $\mathcal{A}$ as

$$\mathbf{Adv}_{\mathsf{E}}^{\mathrm{PRP}}(\mathcal{A}) \overset{\Delta}{=} |\ \Pr[\mathcal{A}^{\mathsf{E}_K} \Rightarrow 1] - \Pr[\mathcal{A}^{\Pi} \Rightarrow 1]\ |,$$

where $K \xleftarrow{\$} \mathcal{K}$ and $\Pi \xleftarrow{\$} \mathrm{Perm}(\mathcal{X})$. E is said to be a (q, t, ϵ) secure PRP, if

$$\mathbf{Adv}_{\mathsf{E}}^{\mathsf{PRP}}(q,t) \overset{\Delta}{=} \max_{\mathcal{A}} \mathbf{Adv}_{\mathsf{E}}^{\mathsf{PRP}}(\mathcal{A}) \leq \epsilon,$$

where the maximum is taken over all adversaries with maximum running time t that asks at most q queries.

Now, we define the security against those adversaries who have access to the keyed permutations as well as their inverse.

As previously, let $\mathsf{E} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$ be a keyed bijective function on $\mathcal{X}$. Now consider a distinguisher $\mathcal{A}$, who has oracle access to a permutation and its inverse over $\mathcal{X}$. Suppose $\mathcal{A}$ makes at most q queries with maximum running time t. The task of the distinguisher is to distinguish a permutation E from a random permutation. We consider the Strong Pseudo Random Permutation (SPRP) advantage of $\mathcal{A}$ as

$$\mathbf{Adv}_{\mathsf{E}}^{\mathrm{SPRP}}(\mathcal{A}) \overset{\Delta}{=} |\ \Pr[\mathcal{A}^{\mathsf{E}_K,\mathsf{E}_K^{-1}} \Rightarrow 1] - \Pr[\mathcal{A}^{\Pi,\Pi^{-1}} \Rightarrow 1]\ |,$$

where $K \xleftarrow{\$} \mathcal{K}$ and $\Pi \xleftarrow{\$} \mathrm{Perm}(\mathcal{X})$. E is said to be a (q, t, ϵ) secure SPRP, if

$$\mathbf{Adv}_{\mathsf{E}}^{\mathsf{SPRP}}(q, t) \overset{\Delta}{=} \max_{\mathcal{A}} \mathbf{Adv}_{\mathsf{E}}^{\mathsf{SPRP}}(\mathcal{A}) \leq \epsilon,$$

where the maximum is taken over all adversaries that make at most q queries with maximum running time t.

Theorem 2.3.1. (PRF-PRP Switching Lemma) *Let $\mathsf{F} : \{0,1\}^n \to \{0,1\}^n$ be a random function and $\mathsf{E} : \{0,1\}^n \to \{0,1\}^n$ be a random permutation, for a natural number $n \geq 1$. Suppose $\mathcal{A}$ be a distinguisher with oracle access, which asks at most q queries. Then*

$$\mathbf{Adv}_{\mathsf{E}}^{\mathsf{F}}(\mathcal{A}) \leq \frac{q(q-1)}{2^{n+1}}. \tag{2.1}$$

A short proof of this lemma can be found in [28].

Almost (XOR) Universal and Almost Regular Hash Function. Let $\mathcal{K}_h, \mathcal{X}$ be two non-empty finite sets and H be an n-bit keyed function $\mathsf{H} : \mathcal{K}_h \times \mathcal{X} \to \{0,1\}^n$. Then, H is said to be an ϵ-Almost Xor Universal (AXU) hash function if for any distinct $X, X' \in \mathcal{X}$ and for any $\delta \in \{0,1\}^n$,

$$\Pr[K_h \xleftarrow{\$} \mathcal{K}_h : \mathsf{H}_{K_h}(X) \oplus \mathsf{H}_{K_h}(X') = \delta] \leq \epsilon. \tag{2.2}$$

Moreover, H is said to be an ϵ-Almost Regular (AR) hash function if for any $X \in \mathcal{X}$ and for any $\delta \in \{0,1\}^n$,

$$\Pr[K_h \xleftarrow{\$} \mathcal{K}_h : \mathsf{H}_{K_h}(X) = \delta] \leq \epsilon. \tag{2.3}$$

A keyed hash function is said to be an $(\epsilon_{\mathrm{axu}}, \epsilon_{\mathrm{reg}})$-AXUAR hash function if it is ϵ_{axu}-AXU and ϵ_{reg}-AR hash function.

PolyHash Function. PolyHash [86] is one of the popular examples of an algebraic hash function, defined as follows: for a fixed key $k_h \in \{0,1\}^n$ and for a message $M \in \{0,1\}^*$, we first apply a padding rule 0^* i.e., pad the minimum number of zeros to the end of M, so that the total number of bits in the padded message becomes a multiple of n. Let the padded message be $M^* = M_1 \| M_2 \| \ldots \| M_l$ where $l = \lceil |M|/n \rceil$ and for each i, $|M_i| = n$. Then,

$$\mathsf{PolyHash}_{k_h}(M) = M_1 \cdot {k_h}^{l+1} \oplus M_2 \cdot {k_h}^{l} \oplus \ldots \oplus M_l \cdot k_h^2 \oplus \langle |M| \rangle_n \cdot k_h, \tag{2.4}$$

where l is the number of blocks of M^* and the multiplications in Equation (2.4) are in the field GF(2^n). If $M = \varepsilon$, the empty string, we define $\mathsf{Poly}_{k_h}(\varepsilon) = k_h^2 \oplus k_h$. Note that the use of the non-injective padding rule (i.e., appending 0^* at the end of the message) does not make the hash function insecure as the definition includes the message length information, which is the safeguard against the xor universal attack. The following well-known result says that the PolyHash defined in Equation (2.4) with an n-bit key is an $\left(\frac{\ell+1}{2^n}, \frac{\ell+1}{2^n}\right)$-AXUAR hash function, where ℓ is the maximum number of message blocks.

Lemma 2.3.2. $\mathsf{PolyHash}$ *as defined in Equation* (2.4) *is* $\left(\frac{\ell+1}{2^n}, \frac{\ell+1}{2^n}\right)$-AXUAR *hash function.*

Proof. Consider two distinct messages M^1 and M^2 with a maximum number of blocks less than ℓ. Thus according to Equation (2.4), for any $\delta \in \{0,1\}^n$,

$$\mathsf{PolyHash}_{k_h}(M^1) \oplus \mathsf{PolyHash}_{k_h}(M^2) \oplus \delta$$

is a non-zero polynomial on k_h of degree at most $\ell + 1$. Hence, it has at-most $\ell + 1$ roots in $\{0,1\}^n$. Thus for a uniform random choice of k_h from $\{0,1\}^n$, we have

$$\Pr[\mathsf{PolyHash}_{k_h}(M^1) \oplus \mathsf{PolyHash}_{k_h}(M^2) = \delta] \leq \frac{\ell+1}{2^n}. \tag{2.5}$$

Similarly, for any M^1 with ℓ many blocks, and any $\delta \in \{0,1\}^n$ the polynomial

$$\mathsf{PolyHash}_{k_h}(M^1) \oplus \delta,$$

is a non-zero polynomial on k_h of degree at most $\ell+1$ and thus for an uniform random k_h from $\{0,1\}^n$ we have

$$\Pr[\mathsf{PolyHash}_{k_h}(M^1) = \delta] \leq \frac{\ell+1}{2^n}. \tag{2.6}$$

Thus, fron Equations (2.5) and (2.6) it follows that PolyHash is a $\left(\frac{\ell+1}{2^n}, \frac{\ell+1}{2^n}\right)$-AXUAR hash function.

□

2.4 H-Coefficient Technique

The H-Coefficient technique is a powerful tool that is used to bound the distinguishing advantage between two random systems. Its formal introduction was made by Patarin in SAC'09 [77]. It regained attention since the work of Chen and Steinberger [30] to analyze the security of iterated Even-Mansour [43] cipher. Since then, it has been successfully used as a tool for upper bounding the statistical distance between the responses of two interactive systems. It is commonly used to prove the pseudo randomness of various cryptographic constructions against information theoretic distinguishers. The H-Coefficient technique is generally implemented as follows: Suppose we have an information theoretic deterministic distinguisher D with access to either the real oracle, i.e., the real construction, or the ideal oracle. Generally, the ideal oracle is considered as a uniform random function or permutation. The collection of all the queries made by D to the oracle and the responses received by D from the oracle, is called the *attack transcript* of D, denoted as $\tau \triangleq ((x_1, y_1), (x_2, y_2), \ldots, (x_q, y_q))$. Occasionally, we permit the oracle to disclose further internal information to D, but only after D completes all the queries and before it generates the final output. In such instances, the transcript of D carries the extra information about the oracle. So, the maximum distinguishing advantage of D in this scenario cannot be lower than that without the additional information. The transcript τ represents a random variable, and its randomness is solely derived from the randomness of the oracle with which D interacts.

Let us consider two random variables T_{re} and T_{id}, that takes the transcript τ resulting from the interaction between D and the real world or between D and the ideal world respectively. The probability of observing a transcript τ in the real world is referred to as the *real interpolation probability*, while the probability of observing a transcript τ in the ideal world is referred to as the *ideal interpolation probability*. A transcript τ is considered as an *attainable* transcript with respect to D if its ideal interpolation probability is non-zero (i.e., $\Pr[\mathsf{T}_{\text{id}} = \tau] > 0$). The collection of all attainable transcripts is denoted by $\mathcal{V}$. Now we state the theorem of H-Coefficient Technique [77, 30].

Theorem 2.4.1 (**H-Coefficient Technique**). *Suppose D be a fixed deterministic distinguisher with the oracle access to either the real oracle $\mathcal{O}_{\text{re}}$ or the ideal oracle $\mathcal{O}_{\text{id}}$ and $\mathcal{V} = \mathcal{V}_{\text{g}} \cup \mathcal{V}_{\text{b}}$, $\mathcal{V}_{\text{g}} \cap \mathcal{V}_{\text{b}} = \emptyset$, be some partition of the set of all attainable transcripts of D. Suppose there exists $\epsilon_{\text{ratio}} \geq 0$ such that for any $\tau \in \mathcal{V}_{\text{g}}$,*

$$\frac{\Pr[\mathsf{T}_{\text{re}} = \tau]}{\Pr[\mathsf{T}_{\text{id}} = \tau]} \geq 1 - \epsilon_{\text{ratio}},$$

and there exists $\epsilon_{\text{bad}} \geq 0$ such that $\Pr[\mathsf{T}_{\text{id}} \in \mathcal{V}_{\text{b}}] \leq \epsilon_{\text{bad}}$. Then,

$$\mathbf{Adv}_{\mathcal{O}_{\text{re}}}^{\mathcal{O}_{\text{id}}}(\mathsf{D}) \triangleq |\Pr[\mathsf{D}^{\mathcal{O}_{\text{re}}} \to 1] - \Pr[\mathsf{D}^{\mathcal{O}_{\text{id}}} \to 1]| \leq \epsilon_{\text{ratio}} + \epsilon_{\text{bad}}. \tag{2.7}$$

The proof of the theorem can be found in [29].

A tutorial introduction to the H-coefficient technique and detailed examples of applying this technique for proving security of some basic pseudo-random objects can be found in [55].

2.5 Permutation Based Cryptography

Cryptographic permutations are keyless public permutations that are designed to behave like random permutations. In recent years cryptographic permutations have started to evolve as a useful primitive in parallel to the block ciphers. The main characteristic of cryptographic permutations is that they are keyless and hence separate processing of the key and the data input is not required as in a block cipher. This makes cryptographic permutations a more efficient primitive compared to block ciphers in certain scenarios. Cryptographic permutations gained prominence during the SHA-3 competition, where many proposed schemes were built upon this primitive. The adoption of the permutation based Keccak sponge function as the SHA-3 standard further boosted the confidence in the community regarding the advantage of this approach [80]. In 2007, Bertoni et al. defined the cryptographic permutation based sponge function [8], which was initially aimed for hashing. Soon after, several efficient modes for encryption, authentication and authenticated encryption were developed [68, 6, 7]. In the recent day, permutation based constructions have emerged as a successful and fully established alternative to modes based on block ciphers. Notably, Ascon [40], the winner in NIST lightweight competition [76], is also based on permutation. Apart from the modes, several cryptographic permutations have also been designed which are claimed to be more efficient than standard block ciphers [9, 15, 5].

Besides the permutation based designs of encryption and authentication schemes, extensive research has been carried out in designing block ciphers and tweakable block ciphers using public random permutations. Even-Mansour (EM) [43] and Iterated Even-Mansour (IEM) [16, 32, 39, 34] ciphers are the main approaches for designing block ciphers and tweakable block ciphers from public random permutations. EM cipher is defined as $\mathrm{EM}(x) \triangleq \pi(x \oplus k_1) \oplus k_2$, where π is a public random permutation and k_1, k_2 are two independent keys. Iterating EM cipher for $r \geq 2$ times with r independent permutations and $r+1$ independent round keys defines the r-round IEM cipher, i.e. $\mathrm{EM}^r(x) \overset{\Delta}{=} k_{r+1} \oplus \pi_r(k_r \oplus \pi_{r-1}(\ldots(\pi_2(k_2 \oplus \pi_1(k_1 \oplus x))\ldots))$. A long line of research has studied the security of r-round IEM [16, 32, 39, 34]. Recently, Chen et al. have designed two public permutation based PRFs [31] which have been proven to be secure beyond the birthday bound.

2.6 Tweakable Enciphering Schemes

A *Tweakable Enciphering Scheme* (TES) is a tweak-based length preserving encryption scheme that encrypts variable length messages. A TES provides security against adaptive chosen plaintext and ciphertext attacks. In other words, an efficient adversary should be unable to distinguish ciphertexts produced from a TES from random strings and should not be able to manipulate a ciphertext to decrypt into meaningful information. A TES accepts an additional input called a "tweak" apart from the message and the key. The tweak is considered a public value that enhances the diversity of the resulting ciphertext. Due to the length preserving feature, TES is considered a suitable candidate for low-level disk encryption [50, 26, 19].

Formally, a TES is a function $\mathbf{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$, where $\mathcal{K} \neq \emptyset$ and $\mathcal{T} \neq \emptyset$ are the key space and the tweak space respectively. The message and the cipher spaces are $\mathcal{M}$. In general, we assume that $\mathcal{M} = \cup_{i>0}\{0,1\}^i$, but in certain scenarios $\mathcal{M}$ may be restricted to contain strings of some predefined lengths.

We shall sometimes write $\mathbf{E}_K^T(.)$ instead of $\mathbf{E}(K, T, .)$. The inverse of an enciphering scheme is $\mathbf{D} = \mathbf{E}^{-1}$ where $X = \mathbf{D}_K^T(Y)$ if and only if $\mathbf{E}_K^T(X) = Y$. An important property of a tweakable enciphering scheme is that it is length preserving, i.e., for every $x \in \mathcal{M}$ and every $T \in \mathcal{T}$, $|\mathbf{E}_K^T(x)| = |x|$.

Security of TES: Let $\mathrm{Perm}^{\mathcal{T}}(\mathcal{M})$ denote the set of all functions $\boldsymbol{\pi} : \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ where $\boldsymbol{\pi}(\mathcal{T}, .)$ is a length preserving permutation. Such a $\boldsymbol{\pi} \in \mathrm{Perm}^{\mathcal{T}}(\mathcal{M})$ is called a tweak indexed permutation. For a tweakable enciphering scheme $\mathbf{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$, we define the advantage of an adversary $\mathcal{A}$ has in distinguishing $\mathbf{E}$ and its inverse from a random tweak indexed permutation and its inverse in the following manner.

$$\begin{aligned}\mathbf{Adv}_{\mathbf{E}}^{\text{tSPRP}}(\mathcal{A}) = \Big| &\Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathbf{E}_K(.,.),\mathbf{E}_K^{-1}(.,.)} \Rightarrow 1\right] - \\ &\Pr\left[\boldsymbol{\pi} \xleftarrow{\$} \text{Perm}^{\mathcal{T}}(\mathcal{M}) : \mathcal{A}^{\boldsymbol{\pi}(.,.),\boldsymbol{\pi}^{-1}(.,.)} \Rightarrow 1\right]\Big|. \qquad (2.8)\end{aligned}$$

We assume that an adversary never repeats a query, i.e., it does not ask the encryption oracle with a particular value of (T, P) more than once and neither does it ask the decryption oracle with a particular value of (T, C) more than once. Furthermore, an adversary never queries its deciphering oracle with (T, C) if it got C in response to an encipher query (T, P) for some P. Similarly, the adversary never queries its enciphering oracle with (T, P) if it got P as a response to a decipher query of (T, C) for some C. These queries are called *pointless* as the adversary knows what it would get as responses for such queries.

In the last few years, there have been several proposals for TES constructions, with many of them using block ciphers as an underlying primitive. CMC [50], EME [51], EME* [48], FMix [11], AEZ [52] are constructed only using block ciphers whereas XCB [67, 20], HCTR [85], HCH [26], TET [49], HEH, HMCH [81] are constructed using block ciphers and universal hash functions. Also, there are a few TES constructions that use stream ciphers [23, 82, 33].

The security analysis of most block cipher based schemes is typically based on the assumption that the underlying block cipher is a strong pseudorandom permutation. This assumption holds as these schemes rely on the decryption functionality of the block cipher to decrypt the ciphertext. Notably, constructions like FMix [11], AEZ [52] and FAST [19] do not use the decryption property of the block cipher and prove their security bound using the pseudorandom function property of block cipher. Such schemes are called *inverse free* TESs. All the above constructions achieve birthday bound security. Dutta and Nandi [41] proposed a TES that relies on a tweakable block cipher and they provided beyond the birthday bound security.

2.6.1 Various Model of Designing TES

Traditionally TESs have been classified into three different groups based on their structure.

Hash-Encrypt-Hash. Hash-Encrypt-Hash approach was first introduced by Naor and Reingold [75]. The proposed construction utilizes an invertible ECB mode of encryption sandwiched between two invertible pairwise independent hash functions to create a wide block secure strong pseudorandom permutation. However, the description provided in their work was high-level, and subsequent work [74] did not fully specify a mode of operation. Also, note that the scheme is not a tweakable strong pseudorandom permutation, as the concept of tweak was proposed much later. In FSE'06, Chakraborty and Sarkar present PEP [25] using Hash-Encrypt-Hash approach. PEP incorporates a layer of ECB-type encryption between two layers of polynomial hashing. Halevi later proposed TET [49], a more efficient version of PEP. HEH proposed by Sarkar [81] is another construction in this category.

Encrypt-Mix-Encrypt. Encrypt-Mix-Encrypt is a type of construction that has a mixing layer between two encryption layers. CMC [50] is the first TES construction of this type, proposed by Halevi and Rogaway. They use a mixing layer between two CBC encryption layers. This is a sequential construction. Unlike CMC, EME [51] is a parallel construction. In EME, authors used two ECB type encryption layers with a mixing layer in between. EME* [48] is an extended version of EME, which can handle arbitrary message lengths. These constructions use both the forward and inverse direction of the underlying block cipher. FMix [11], proposed by Bhaumik et al., is a variant of CMC but does not require inverse calls of the underlying block cipher. AEZ [52] is also a recent addition to this category.

Hash-Counter-Hash. Hash-Counter-Hash is similar to Hash-Encrypt-Hash, but it uses counter-mode encryption instead of ECB in between two hash layers. Because of the use of counter-mode encryption, it easily handles variable length messages. The construction known as XCB [67] is the first of its kind in the Hash-Counter-Hash category, requiring five block cipher keys and two block cipher calls (in addition to those in counter mode encryption). Wang et al. introduced HCTR [85], which reduces the number of block cipher calls and utilizes a single block cipher key. A serious drawback of HCTR was that the security proof provided in [85] only guaranteed that the security degrades by a cubic bound on the data complexity of the adversary. As quadratic security bounds for TES were already known, so HCTR seemed to provide very weak security guarantees compared to the then known constructions. In an attempt to fix this situation, HCH [26] was proposed, which modified HCTR in various ways to produce a new mode that used one more block cipher call than HCTR but provided a quadratic security guarantee. HCH offered some more advantages over HCTR in terms of the number of keys used, etcetera. In [69], another variant of HCTR was proposed, which provides a quadratic security bound and later in [24] a quadratic security bound of the original HCTR construction (as proposed in [85]) was proved. Chakraborty et al. recently proposed FAST [19] a construction, based on the Hash-Counter-Hash paradigm, using only forward calls to the block cipher.

2.7 Message Authentication Codes

Message authentication codes (MAC) provide authentication in the symmetric key setting. It is assumed that the sender and the receiver share a common secret key K. Given a message x, the sender uses K to generate a footprint of the message. This footprint (commonly called a tag) is the message authentication code (MAC) for the message x. The sender transmits the pair $(x, \mathtt{tag})$ to the receiver. The receiver uses K to verify that $(x, \mathtt{tag})$ is a properly generated message-tag pair. Verification is generally performed by regenerating the tag on the message x and comparing the generated tag with the one received.

Formally, we see a MAC as a pair of algorithms: the tag generation algorithm and the verification algorithm. Both algorithms depend on a tag generation function $F : \mathcal{K} \times \mathcal{M} \to \{0,1\}^{\tau}$, where $\mathcal{K}$ is the key space, $\mathcal{M}$ is the message space and τ is the tag length. The tag generation algorithm receives as input a message $x \in \mathcal{M}$ and the key $K \in \mathcal{K}$, and generates $t = F_K(x)$ and finally outputs (x, t). The verification algorithm on receiving a message tag pair (x, t), computes $t' = F_K(x)$ and outputs true if $t' = t$ and false otherwise. We generally specify the MAC by the tag generation function $F_K(.)$ and sometimes $F_K(.)$ itself is called the message authentication code.

Security of MACs. The security of a MAC F is defined using an interaction of F with an adversary $\mathcal{A}$ [4]. It is assumed that $\mathcal{A}$ has an oracle access to $F_K()$, where $K \xleftarrow{\$} \mathcal{K}$. For a query $x \in \mathcal{M}$ of $\mathcal{A}$ the oracle responds by sending $y = F_K(x)$. Let, $\mathcal{A}$ query $x_1, x_2, \ldots, x_q$ and gets $y_1, y_2, \ldots, y_q$ as responses from the oracle. These queries are performed adaptively. Finally, $\mathcal{A}$ outputs a pair (x^*, y^*), where $x^* \notin \{x_1, x_2, \ldots, x_q\}$. This pair is called a forgery and it is said that $\mathcal{A}$ has successfully *forged* F if $F_K(x^*) = y^*$. The auth-advantage of $\mathcal{A}$ is defined as

$$\mathsf{Adv}_F^{\mathsf{auth}}(\mathcal{A}) = \Pr[K \xleftarrow{\$} \mathcal{K} : \mathcal{A} \text{ forges}].$$

We say that F is a (ϵ, t) secure MAC if for every adversary $\mathcal{A}$, which runs for time at most t, $\mathsf{Adv}_F^{\mathsf{auth}}(\mathcal{A}) \leq \epsilon$.

It is well known that if F is a secure PRF, then F is also a secure MAC. In particular, for any arbitrary adversary $\mathcal{A}$ for the MAC F there exists a PRF adversary $\mathcal{B}$ for F such that

$$\mathsf{Adv}_F^{\mathsf{auth}}(\mathcal{A}) \leq \mathsf{Adv}_F^{\mathsf{prf}}(\mathcal{B}) + \frac{1}{2^\tau}, \tag{2.9}$$

where $\mathcal{B}$ and $\mathcal{A}$ both run almost for the same time and ask almost the same number of queries.

MAC algorithms can be broadly categorized into four types: (i) MACs based on block ciphers; (ii) MACs based on tweakable block ciphers; (iii) MACs based on cryptographic hash functions; and finally, (iv) MACs based on universal hash functions. We discuss popular candidates in these categories in the following subsections.

2.7.1 MACs Based on Block Ciphers

Block cipher based MACs are most commonly used. In this approach, a block cipher is used as the main primitive. Processing of messages of arbitrary lengths is done using specific modes. Depending upon the modes applied, there are various MACs based on block cipher. These MACs can generally be categorized into two types: (a) Sequential MACs and (b) Parallel MACs.

Sequential MACs. Sequential MACs based on block cipher are built in sequential mode. CBC MAC [4] is most popular block cipher based sequential MAC. CBC MAC defined as

$$\mathsf{CBC}_K(M_1\| \ldots \|M_\ell) \triangleq \mathsf{E}_K(\ldots(\mathsf{E}_K(\mathsf{E}_K(M_1) \oplus M_2))\ldots M_\ell),$$

where E_K is a n-bit block cipher and $M = (M_1\| \ldots \|M_\ell)$ is the message. In [4], Bellare et al. proved that CBC MAC is secure for the fixed length messages where the message lengths are multiples of the block size. If the messages are prefixes of other messages, CBC MAC is not secure due to its vulnerability to length extension attacks. One potential solution involves incorporating the message length information as the initial block in CBC computation. However, this approach necessitates knowing the entire message before CBC computation, which might not always be feasible. To counter this vulnerability, Petrank et al. [78] proposed encrypting the CBC output with a separate block cipher and named it Encrypted CBC-MAC or EMAC. EMAC is defined as:

$$\mathsf{EMAC}_{K_1,K_2}(M) = \mathsf{E}_{K_2}(\mathsf{CBC}_{K_1}(M)).$$

In [78], Petrank et al. show that the EMAC is secure if the lengths of the messages are multiples of the block size. For the arbitrary length of messages, Black and Rogaway proposed a three-keyed variant of CBC MAC, called XCBC [13]. After that, Iwata et al. [58] proposed a two-keyed variant of CBC MAC, called TCBC and also a one-keyed version called OMAC [54].

Parallel MACs. The first parallel MAC was introduced by Bellare et al., called XOR MAC [3]. In this mode, the message M is parsed into ℓ many blocks of length b, where $b < n$. Then the i-th block M_i is prepended with $\langle i \rangle$, where $\langle i \rangle$ denotes the $n - b - 1$ bit encoding of integer i and then the block cipher E_K is applied over each $1 \| \langle i \rangle \| M_i$ and we take the xor of the block cipher outputs. Finally, we encrypt $0 \| IV$ with E_K, where IV is of size $n - 1$ bits and take the xor of all these block cipher outputs. When IV is random, then the scheme is called XMACR [3], an instance of a probabilistic MAC and when IV is a counter, then the scheme is called XMACC [3], an instance of a stateful MAC. Later, Bernstein proposed Protected Counter Sum in short PCS MAC, which is similar to XMACC, only the block cipher is replaced by a keyed function from $(b+c)$ bits to b bits. In 2002, Black and Roagway improved upon the XOR MAC and proposed a deterministic, parallelizable message authentication code, called PMAC [14]. In this mode, each message block M_i is masked with Δ_i, which is encrypted by E_K (except the final message block), where Δ_i is some function of the block cipher. The masked value of the final message block and all the block cipher outputs are xored together to produce an intermediate value. This intermediate value is again encrypted to produce the final tag. PMAC iterates a block cipher in a fully parallelizable way and it requires just one block-cipher invocation to process each message block. Mandal and Nandi [73] have shown the security bound of PMAC to be roughly $q\sigma/2^n$, where σ is the total number of message blocks processed, and q is the total number of queries made. Later, Gazi et al. [44] have demonstrated an attack with roughly $2^{n/2}$ data complexity and hence established the tightness of the bound. Later Yasuda [87] introduced PMAC with parity, which processes each sequence of r consecutive message blocks in PMAC like manner, but inserts the xor sum of those r blocks as an additional block. Zhang [89] introduced PMACX construction that generalizes PMAC with parity construction by multiplying the input with an MDS matrix before authentication. In 2016, Lyukx et al. proposed a lightweight variant of PMAC, called $\mathsf{LightMAC}$ [65]. In this mode, the message M is parsed into ℓ many blocks of length b, where $b < n$. Then the i-th block M_i (except the last message block) is prepended with $\langle i \rangle$, where $\langle i \rangle$ denotes the $n - b$ bit encoding of integer i

and then the block cipher E_K is applied over each $\langle i \rangle \| M_i$ and we take the xor of the block cipher outputs with the last message block with appropriate padding. Finally, we encrypt the xor with another independent block cipher $\mathsf{E}_{K'}$ to generate the tag. This is the first deterministic MAC that is proven to be secure independent of the length of the message.

2.7.2 MACs Based on Cryptographic Hash Functions

In 1996, Bellare et al. [2] proposed two cryptographic hash functions based MACs, called, NMAC and HMAC. NMAC uses a compression function $f : \{0,1\}^{b+k} \to \{0,1\}^k$. For a message $M = (M_1, \ldots, M_\ell)$, where each M_i is of size b-bit, NMAC is defined as follows:

$$\mathsf{NMAC}^f_{K_1,K_2}(M) \triangleq f(K_2, f(\ldots(f(f(K_1, M_1), M_2))\ldots), M_\ell),$$

where K_1 and K_2 are independent keys.

HMAC uses a IV-based hash function. Let f be compression function such that $f : \{0,1\}^{b+n} \to \{0,1\}^n$ and F be the Markel-Damagard hash function, defined as:

$$F^f(IV, M) \triangleq f(f(\ldots(f(f(K_1, M_1), M_2))\ldots), M_\ell).$$

Then HMAC defined as:

$$\mathsf{HMAC}(K, M) \triangleq F^f(K \oplus opad, F^f(K \oplus ipad, M)),$$

where M is the message, K is the key and $opad, ipad$ are two fixed constants.

2.7.3 MACs Based on Universal Hash Functions

In 1981, Wegman and Carter [17] proposed a completely different paradigm of constructing MACs from universal hash functions. In this paradigm, a universal hash function is applied on the message M which is masked with a random salt. The drawback of this approach is that one requires a fresh random string each time to authenticate a new message. To alleviate this problem, we are required to use a pseudorandom function and it should be applied over a nonce N each time one is required to authenticate a new message. This nonce-based MAC, known as the Wegman-Carter, in short, WC construction, generates tag as $\mathsf{WC}_{K_1,K_1}(M) \stackrel{\Delta}{=} H_{K_1}(M) \oplus F_{K_2}(N)$, where H is a universal hash function and F is a PRF. The WC construction gives optimal security when nonces are never reused. However, the inherent drawback of the scheme is that it loses security once a nonce is repeated. In fact, if the underlying hash function is a $\mathsf{PolyHash}$, then nonce repetition can reveal the hash key. To prevent this nonce-misuse problem, Black et al. [12] proposed a simple solution in which one applies a $2n$-bit PRF to nonce and the hash value, i.e., if the PRF takes $2n$-bit inputs, one can define the tag as $F_{K_2}(N \| H_{K_1}(M))$. However, designing $2n$-bit to n-bit PRF is non-trivial as one can apply 5-round feistel [64] or buttefly [1] construction. Due to the practical infeasibility of pseudorandom function, Shoup replaced the function with block cipher and renamed the construction as Wegman-Carter-Shoup, in short, WCS [125]. This construction offers the same level of security in nonce respecting and nonce misuse scenarios. However, to alleviate the nonce misuse problem of WCS, one can encrypt the output of WCS with an independent block cipher key, i.e.,$T = \mathsf{E}_{K_3}(H_{K_1}(M) \oplus \mathsf{E}_{K_2}(N))$. Although the construction gives security even when nonces repeat, but at the same time, the security of this construction becomes poorer than the original WCS construction in nonce respecting setting.

3 Designing Tweakable Enciphering Schemes Using Public Permutations

Although several modes for authentication, hash function, and authenticated encryption, have been developed using public permutations till date, to our knowledge, the only work which describes a TES built using a public random permutation is [6]. The construction in [6] uses four round Luby Rackoff construction using two pseudorandom functions and the pseudorandom functions are constructed using public permutations. Concrete security bounds and formal security proofs for the TES scheme are not provided in [6] and to the best of our knowledge, there is no provably secure public permutation based TES scheme. We initiate a study of such a construction in this chapter. Our concrete contributions are the following.

1. First, we propose a generic construction of a public permutation based TES, called ppTES. Our proposal closely resembles the HCTR construction. ppTES is designed using a public permutation π, a length expanding public permutation based pseudorandom function[1] $\mathsf{F}_k^{\pi'}$, where π and π' are two independent public random permutations over the same space. Additionally, ppTES uses a keyed hash function H_{k_h}, which is required to be both almost xor universal (AXU) and almost regular (AR) (we further call such functions as AXUAR functions). We prove that if $\mathsf{F}_k^{\pi'}$ is a secure length expanding public permutation based

[1] Informally, a length expanding PRF takes an input x and the number of blocks b and outputs b many blocks, where block refers to an element of $\{0,1\}^n$, for some fixed n.

PRF and the hash function is a secure AXUAR function, then ppTES is secure against adaptive chosen plaintext and ciphertext adversaries.

2. As our second contribution, we construct a length expanding public permutation based PRF, which we call ppCTR. ppCTR essentially is a counter mode of encryption where the block ciphers are replaced by the single round Even-Mansour [43] construction. We show that ppCTR offers a tight $n/2$ bit security. We use ppCTR and the PolyHash [86] function in ppTES construction to realize a concrete TES, which we call ppHCTR. ppHCTR requires two keys and two independent public permutations.

3. Finally, we propose ppHCTR+, a public permutation based TES that uses a single key and a single public permutation. Along with the permutation, ppHCTR+ also requires an AXUAR hash function and the only key required in ppHCTR+ is the hash key of the AXUAR hash function. We prove that ppHCTR+ is a birthday bound secure public permutation based TES.

3.1 TES Based on Public Random Permutation

A *tweakable enciphering scheme* (TES) $\mathfrak{T}$, on the public random permutation model, is a pair of algorithms $\mathfrak{T} = (\mathsf{Enc}^{\boldsymbol{\pi}}, \mathsf{Dec}^{\boldsymbol{\pi}})$, where $\mathsf{Enc}^{\boldsymbol{\pi}} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ and $\mathsf{Dec}^{\boldsymbol{\pi}} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ constructed by d many n-bit permutations $\boldsymbol{\pi} \stackrel{\Delta}{=} (\pi_1, \ldots, \pi_d)$ and $\mathcal{K}, \mathcal{T}, \mathcal{M}$ are three non-empty finite sets. As the TES is a length preserving permutation, $\mathsf{Enc}_k^{\boldsymbol{\pi}}(T, \cdot)$ for all $M \in \mathcal{M}$. A tweakable permutation is a mapping $\widetilde{\mathsf{\Pi}} : \mathcal{T} \times \mathcal{M} \to \mathcal{M}$, such that for all tweak $T \in \mathcal{T}$, $M \mapsto \widetilde{\mathsf{\Pi}}(T, M)$ is a permutation of $\mathcal{M}$. We will denote $\widetilde{\mathsf{\Pi}}(T, M)$ by $\widetilde{\mathsf{\Pi}}^T(M)$. The set of all tweakable permutations is denoted by $\mathsf{TP}(\mathcal{T}, \mathcal{M})$.

Now, we consider that $\pi_1, \ldots, \pi_d \xleftarrow{\$} \mathsf{Perm}(n)$ and the Chosen Ciphertext Attack (CCA) distinguisher D is given access to either the oracles $(\mathfrak{T}.\mathsf{Enc}_K^{\boldsymbol{\pi}}; \mathfrak{T}.\mathsf{Dec}_K^{\boldsymbol{\pi}}; \pi_1^{\pm}, \ldots, \pi_d^{\pm})$ for a random key $K \xleftarrow{\$} \mathcal{K}$ or the ideal oracles $(\widetilde{\mathsf{\Pi}}; \widetilde{\mathsf{\Pi}}^{-1}; \pi_1^{\pm}, \ldots, \pi_d^{\pm})$ for $\widetilde{\mathsf{\Pi}} \xleftarrow{\$} \mathsf{TP}(\mathcal{T}, \mathcal{M})$.

The superscript ± for the π_i's denotes that the distinguisher can query π_i in both the forward and reverse directions. The tweakable Strong Pseudo-Random Permutation (tSPRP) advantage of $\mathfrak{T}$ in public random permutation model with respect to the distinguisher D that makes q_e encryption queries, q_d decryption queries and altogether q_p primitive queries is

$$\mathbf{Adv}_{\mathfrak{T}}^{\mathrm{tSPRP}}(\mathsf{D}) \stackrel{\Delta}{=} |\ \Pr[\mathsf{D}^{\mathfrak{T}.\mathsf{Enc}_K^{\boldsymbol{\pi}};\mathfrak{T}.\mathsf{Dec}_K^{\boldsymbol{\pi}};\pi_1^{\pm},\ldots,\pi_d^{\pm}} \to 1] - \Pr[\mathsf{D}^{\widetilde{\Pi};\widetilde{\Pi}^{-1};\pi_1^{\pm},\ldots,\pi_d^{\pm}} \to 1]\ |,$$

where $K \xleftarrow{\$} \mathcal{K}, \pi_1, \ldots, \pi_d \xleftarrow{\$} \mathsf{Perm}(n)$ and $\widetilde{\Pi} \xleftarrow{\$} \mathsf{TP}(\mathcal{T}, \mathcal{M})$. We say that $\mathfrak{T}$ is a $(q_e, q_d, q_p, \ell, \sigma, t)$-secure tSPRP if

$$\mathbf{Adv}_{\mathfrak{T}}^{\mathrm{tSPRP}}(\mathsf{D}) \leq \epsilon,$$

for all CCA distinguishers D that make q_e encryption, q_d decryption, q_p primitive queries and run at most time t and the maximum number of blocks in an encryption or decryption query is ℓ length of and total query complexity is σ.

3.2 PRF Based on Public Random Permutation

Let $\mathsf{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a keyed function from $\mathcal{X}$ to $\mathcal{Y}$ constructed using d many n-bit permutations $\boldsymbol{\pi} \stackrel{\Delta}{=} (\pi_1, \ldots, \pi_d)$, where $\mathcal{K}$ is called the key space, $\mathcal{X}$ is called the input space and $\mathcal{Y}$ is called the output space. We consider the Pseudo Random Function (PRF) security of F under public permutation model where we assume that $\pi_1, \ldots, \pi_d \xleftarrow{\$} \mathsf{Perm}(n)$ and the distinguisher D is given access to either $(\mathsf{F}_K^{\boldsymbol{\pi}}; \pi_1^{\pm}, \ldots, \pi_d^{\pm})$ for a random key $K \xleftarrow{\$} \mathcal{K}$ or $(\mathsf{RF}; \pi_1^{\pm}, \ldots, \pi_d^{\pm})$ for $\mathsf{RF} \xleftarrow{\$} \mathsf{Func}(\mathcal{X}, \mathcal{Y})$. Query of the distinguisher to π_i is called the *primitive query* and query to $\mathsf{F}_K^{\boldsymbol{\pi}}$ or RF is called the *construction query*. We define the PRF advantage of F in public permutation model with respect to the distinguisher D that makes q construction queries and total q_p primitive queries as

$$\mathbf{Adv}_{\mathsf{F}}^{\mathrm{PRF}}(\mathsf{D}) \stackrel{\Delta}{=} |\ \Pr[\mathsf{D}^{\mathsf{F}_K^{\boldsymbol{\pi}};\pi_1^{\pm},\ldots,\pi_d^{\pm}} \to 1] - \Pr[\mathsf{D}^{\mathsf{RF};\pi_1^{\pm},\ldots,\pi_d^{\pm}} \to 1]\ |,$$

where $K \xleftarrow{\$} \mathcal{K}, \pi_1, \ldots, \pi_d \xleftarrow{\$} \mathsf{Perm}(n)$ and $\mathsf{RF} \xleftarrow{\$} \mathsf{Func}(\mathcal{X}, \mathcal{Y})$. F is said to be a (q, q_p, t, ϵ)-secure PRF if $\mathbf{Adv}_{\mathsf{F}}^{\mathrm{PRF}}(q, q_p, t) \triangleq \max_{\mathsf{D}} \mathbf{Adv}_{\mathsf{F}}^{\mathrm{PRF}}(\mathsf{D}) \leq \epsilon$, where the maximum is taken over all distinguishers D that makes q construction queries, total q_p primitive queries and runs for time at most t.

3.3 HCTR Construction

HCTR is a TES proposed by Wang et al. [85] and our main construction ppTES shares the basic structure of HCTR. Hence, in this section, we give a description of HCTR.

HCTR turns an n-bit strong pseudorandom permutation into a variable length tweakable strong pseudorandom permutation. The encryption and decryption algorithm of HCTR is shown in Fig. 3.3.1 and its pictorial representation is shown in Fig. 3.3.2.

We explain the encryption algorithm of HCTR using an example. The decryption algorithm can be understood in a similar way. Suppose the input message $M = (M_1 \| M_2)$ and for the sake of simplicity, we assume that $|M_1| = |M_2| = n$, i.e., M consists of two full blocks. Therefore, in step (2) of the algorithm, the variable $\mathbf{M_L}$ is assigned to M_1 and $\mathbf{M_R}$ is assigned to M_2. In step (3) of the algorithm, we evaluate the poly hash Poly_{k_h} on $(M_2 \| T)$ which results to $M_2 \cdot k_h^3 \oplus T \cdot k_h^2 \oplus \langle |M_2| + |T| \rangle \cdot k_h$ which is xored with the n-bit value M_1 to produce U. In step (4), we take the xor of U and its encryption $V = \mathsf{E}_k(U)$ to produce Z. In step (6), we compute the key stream $\mathbf{S} = S_1 \| S_2$ where each $|S_1| = |S_2| = n$. Since, $|\mathbf{M_R}| = n$, $\mathbf{C_R}$ will be $M_2 \oplus S_1$, which becomes the input along with tweak T to the poly hash function Poly_{k_h}. Evaluation of the poly hash on input $\mathbf{C_R} \| T$ results to $\mathbf{C_R} \cdot k_h^3 \oplus T \cdot k_h^2 \oplus \langle |\mathbf{C_R}| + |T| \rangle \cdot k_h$. Then the result is xored with V to produce $\mathbf{C_L}$, which is returned along with $\mathbf{C_R}$ as the encryption of $M = M_1 \| M_2$.

Wang et al. [85] have shown that HCTR is a secure TES against all adaptive chosen plaintext and chosen ciphertext adversaries that make roughly $2^{n/3}$ encryption and decryption queries. Later, Chakraborty and Nandi [24] improved its security bound to $O(\sigma^2/2^n)$, where σ is the total number of message blocks among all q queries.

```
$\mathsf{HCTR.Enc}_{k,k_h}(T, M)$
    $M_1\|\ldots\|M_l \leftarrow \mathsf{parse}_n(M)$;
    $\mathbf{M_L} \leftarrow M_1; \mathbf{M_R} \leftarrow (M_2\|\ldots\|M_l)$;
    $U \leftarrow \mathbf{M_L} \oplus \mathsf{Poly}_{k_h}(\mathbf{M_R}\|T)$;
    $V \leftarrow \mathsf{E}_k(U)$; $Z \leftarrow U \oplus V$;
    for $i = 1$ to $l$
        $S_i \leftarrow \mathsf{E}_k(Z \oplus i)$ ;
    $\mathbf{S} \triangleq S_1\|\ldots\|S_l$ ;
    $\mathbf{C_R} \leftarrow \mathsf{first}(|\mathbf{M_R}|, \mathbf{S}) \oplus \mathbf{M_R}$;
    $\mathbf{C_L} \leftarrow V \oplus \mathsf{Poly}_{k_h}(\mathbf{C_R}\|T)$;
    return $(\mathbf{C_L}\|\mathbf{C_R})$;
```

```
$\mathsf{HCTR.Dec}_{k,k_h}(T, C)$
    $C_1\|\ldots\|C_l \leftarrow \mathsf{parse}_n(C)$;
    $\mathbf{C_L} \leftarrow C_1; \mathbf{C_R} \leftarrow (C_2\|\ldots\|C_l)$;
    $V \leftarrow \mathbf{C_L} \oplus \mathsf{Poly}_{k_h}(\mathbf{C_R}\|T)$;
    $U \leftarrow \mathsf{E}_k^{-1}(V)$; $Z \leftarrow U \oplus V$;
    for $i = 1$ to $l$
        $S_i \leftarrow \mathsf{E}_k(Z \oplus i)$ ;
    $\mathbf{S} \triangleq S_1\|\ldots\|S_l$;
    $\mathbf{M_R} \leftarrow \mathsf{first}(|\mathbf{C_R}|, \mathbf{S}) \oplus \mathbf{C_R}$;
    $\mathbf{M_L} \leftarrow U \oplus \mathsf{Poly}_{k_h}(\mathbf{M_R}\|T)$;
    return $(\mathbf{M_L}\|\mathbf{M_R})$;
```

Figure 3.3.1: HCTR construction based on an n-bit block cipher E_k and an n-bit Polyhash function. The left part of the algorithm is the encryption function and the right part is the decryption function.

Recently, Dutta and Nandi [41] proposed a tweakable block cipher based HCTR, called *tweakable* HCTR, and showed its security beyond the birthday bound.

Remarks 3.3.1. *In [85], authors defined the output of thePolyHash to be the hash key k_h, when the input is an empty string ε. But that definition of the PolyHash function leads to an attack on the construction as reported in [57]. This attack does not work if the message space contains messages of length at least $n+1$. We redefine the output of the PolyHash for an empty input string to be $k_h^2 \oplus k_h$, which eliminates the message length restriction.*

Motivated by HCTR, we first replace the block cipher based counter mode part of HCTR with a public permutation based length expanding PRF, and the block cipher E_K (see Fig. 3.3.2) with a public permutation π. We show that such a combination yields a secure public permutation based TES, which we call ppTES as described in

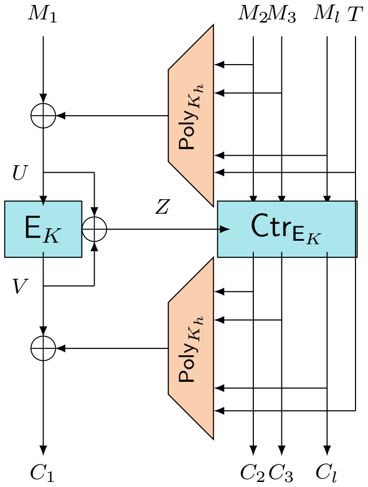

Figure 3.3.2: Pictorial description of HCTR, where E_K is the underlying block cipher, Poly_{K_h} is the poly-hash function and $\mathsf{Ctr}_{\mathsf{E}_K}$ is the counter mode encryption.

section 3.4. In section 3.6, we construct a public permutation based length expanding PRF, which we call ppCTR. Using ppCTR along with the the PolyHash function, we instantiate ppTES to realize a public permutation based TES, which we call ppHCTR. However, ppHCTR requires two independent public permutations, a key for the ppCTR and another independent hash key for the PolyHash function. Next, we go one step further to reduce the number of keys and permutations used in ppHCTR and come up with a single keyed (for the PolyHash function) and single permutation based TES construction, ppHCTR+. We describe ppHCTR+ in section 3.7.

3.4 ppTES : A Generic Public Permutation Based TES

ppTES is based on three cryptographic components: (i) an n-bit public random permutation π_1, (ii) an AXUAR hash function H_{k_h} which maps $\{0,1\}^*$ to $\{0,1\}^n$, and (iii) a public permutation based length expanding PRF $\mathsf{F}_k^{\pi_2}$, where π_2 is a n-bit independent public random permutation independent of π_1. The message space of ppTES is $\{0,1\}^{\geq n}$ and the tweak space is $\{0,1\}^{\mathtt{tw}}$. The working principle of ppTES is exactly the same as HCTR where the block cipher is replaced by a public permutation π_1

and the counter mode encryption is replaced by a public permutation based length expanding PRF $\mathsf{F}_k^{\pi_2}$.

The algorithmic description of encryption and decryption function of ppTES is shown in Fig. 3.4.1. The description in Fig. 3.4.1 mentions $\mathsf{F}_k^{\pi_2}$, which is a length expanding PRF. We describe this primitive next.

```
$\mathsf{ppTES.Enc}_{k,k_h}^{\pi_1,\pi_2}(T, M)$
    1. $M_1\| \ldots \|M_l \leftarrow \mathsf{parse}_n(M)$;
    2. $\mathbf{M_L} \leftarrow M_1; \mathbf{M_R} \leftarrow (M_2\| \ldots \|M_l)$;
    3. $U \leftarrow \mathbf{M_L} \oplus \mathsf{H}_{k_h}(\mathbf{M_R}\|T)$;
    4. $V \leftarrow \pi_1(U)$; $Z \leftarrow U \oplus V$;
    5. $\mathbf{S} \triangleq S_1\| \ldots \|S_{\ell-1} \leftarrow \mathsf{F}_k^{\pi_2}(Z, l)$;
    6. $\mathbf{C_R} \leftarrow \mathsf{first}(|\mathbf{M_R}|, \mathbf{S}) \oplus \mathbf{M_R}$;
    7. $\mathbf{C_L} \leftarrow V \oplus \mathsf{H}_{k_h}(\mathbf{C_R}\|T)$;
    8. **return** $(\mathbf{C_L}\|\mathbf{C_R})$;
```

```
$\mathsf{ppTES.Dec}_{k,k_h}^{\pi_1,\pi_2}(T, C)$
    1. $C_1\| \ldots \|C_l \leftarrow \mathsf{parse}_n(C)$;
    2. $\mathbf{C_L} \leftarrow C_1; \mathbf{C_R} \leftarrow (C_2\| \ldots \|C_l)$;
    3. $V \leftarrow \mathbf{C_L} \oplus \mathsf{H}_{k_h}(\mathbf{C_R}\|T)$;
    4. $U \leftarrow \pi_1^{-1}(V)$; $Z \leftarrow U \oplus V$;
    5. $\mathbf{S} \triangleq S_1\| \ldots \|S_{\ell-1} \leftarrow \mathsf{F}_k^{\pi_2}(Z, l)$;
    6. $\mathbf{M_R} \leftarrow \mathsf{first}(|\mathbf{C_R}|, \mathbf{S}) \oplus \mathbf{C_R}$;
    7. $\mathbf{M_L} \leftarrow U \oplus \mathsf{H}_{k_h}(\mathbf{M_R}\|T)$;
    8. **return** $(\mathbf{M_L}\|\mathbf{M_R})$;
```

Figure 3.4.1: ppTES based on an n-bit public random permutations π_1, an AXUAR hash function H_{k_h} and a public permutation based length expanding PRF $\mathsf{F}_k^{\pi_2}$. $M \in \{0,1\}^{\geq n}$ is the input message and $T \in \{0,1\}^{\mathtt{tw}}$ is the tweak. The left part of the algorithm is the encryption function and the right part is the decryption function.

As in the case of HCTR, to explain the encryption algorithm we use a two block message $M = (M_1\|M_2)$, where $|M_1| = |M_2| = n$. On input M, in step (2) of the algorithm, the variable $\mathbf{M_L}$ is assigned to M_1 and $\mathbf{M_R}$ is assigned to M_2. In step (3) of the algorithm, we evaluate the hash value H_{k_h} on $(M_2\|T)$, which is xored with the n-bit value M_1 to produce U. In step (4), we take the xor of U and its permuted value $V = \pi_1(U)$ to produce Z. In step (5), we compute the key stream $\mathbf{S} = S_1$ using length expanding PRF $\mathsf{F}_k^{\pi_2}$ where $|S_1| = n$. Since, $|\mathbf{M_R}| = n$, $\mathbf{C_R}$ will be $M_2 \oplus S_1$, which becomes the input along with tweak T to the hash function H_{k_h}. Then the resulting hash value is xored with V to produce $\mathbf{C_L}$, which is returned along with $\mathbf{C_R}$

as the encryption of $M = M_1 \| M_2$.

3.4.1 Length Expanding Pseudorandom Function

For an arbitrary large positive integer L, Let $\mathcal{F} \subseteq \mathsf{Func}(\{0,1\}^n \times \mathbb{N}, \cup_{0<i\leq L}\{0,1\}^{ni})$, such that $F \in \mathcal{F}$ if and only if the following two conditions are satisfied:

1. For every $x \in \{0,1\}^n$ and every $b \in [L]$, $|F(x,b)| = nb$.
2. For every $x \in \{0,1\}^n$ and every $b, b' \in [L]$, $b \geq b'$, $\mathsf{first}(nb', F(x,b)) = F(x,b')$.

We call a uniform random element of $\mathcal{F}$ a *length expanding random function.*

In Fig. 3.4.2, we give an algorithmic description of a length expanding random function ρ. The algorithm depicts ρ as a lazy sampler, which provides as output $\rho(x,b)$ upon receiving a query (x,b). For any input (x,b), it first checks whether x is a fresh element or not. If it is fresh, then it samples b many blocks uniformly at random from $\{0,1\}^{nb}$. If it is not fresh, then it first checks whether the number of requested blocks b' in the earlier query for input x is less than the number of requested blocks in the current query for the same input. In that case, it first fetches b' many blocks which are already stored at $\mathbb{T}[x]$, and then samples the remaining blocks, i.e., $b - b'$ blocks independently and uniformly at random from $\{0,1\}^{n(b-b')}$ which is appended with the first b' many fetched blocks and finally updates the entry $\mathbb{T}[x]$ with the output of the current query. The final case is if the number of requested blocks in the current query for input x is less than the number of requested blocks in the earlier query with the same input. Then it fetches the first b many blocks out of b' many blocks which are already stored at $\mathbb{T}[x]$ and returns it.

Informally, length expanding pseudorandom function is a function which is indistinguishable from a length expanding random function by any efficient distinguisher. For the sake of our construction, we require a public permutation based length expanding PRF which we formally define next.

Definition 3.4.1. Public Permutation Based Length Expanding PRF . *Let L be an arbitrary large positive integer and let $\mathsf{F} : \mathcal{K} \times \{0,1\}^n \times [L] \rightarrow \cup_{1\leq i\leq L}\{0,1\}^{ni}$ be* *a keyed function based on d many n-bit permutations $\boldsymbol{\pi} \overset{\Delta}{=} (\pi_1, \ldots, \pi_d)$ such that for any input $(x, b) \in \{0,1\}^n \times [L]$, $\mathsf{F}_k^{\boldsymbol{\pi}}(x, b)$ returns $(y_1, \ldots, y_b)$ where each $y_i \in \{0,1\}^n$. We consider the length expanding* PRF *security of* F *under public permutation model where we assume that $\pi_1, \ldots, \pi_d \xleftarrow{\$} \mathsf{Perm}(n)$ and the distinguisher* D *is given access to either of the world $(\mathsf{F}_K^{\boldsymbol{\pi}}, \pi_1^{\pm}, \ldots, \pi_d^{\pm})$ for a random key $K \xleftarrow{\$} \mathcal{K}$ or $(\rho, \pi_1^{\pm}, \ldots, \pi_d^{\pm})$, where ρ works as shown in Fig 3.4.2. We define the* LENPRF *advantage of* F *in public permutation model with respect to the distinguisher* D *that makes q construction queries and total q_p primitive queries as*

$$\mathbf{Adv}_{\mathsf{F}}^{\mathrm{LENPRF}}(\mathsf{D}) \overset{\Delta}{=} |\ \Pr[\mathsf{D}^{\mathsf{F}_K^{\boldsymbol{\pi}}, \pi_1^{\pm}, \ldots, \pi_d^{\pm}} \rightarrow 1] - \Pr[\mathsf{D}^{\rho, \pi_1^{\pm}, \ldots, \pi_d^{\pm}} \rightarrow 1]\ |,$$

where $K \xleftarrow{\$} \mathcal{K}, \pi_1, \ldots, \pi_d \xleftarrow{\$} \mathsf{Perm}(n)$. F is said to be a (q, q_p, σ, t)-secure LENPRF if $\mathbf{Adv}_{\mathsf{F}}^{\mathrm{LENPRF}}(q, q_p, \sigma, t) \overset{\Delta}{=} \max_{\mathsf{D}} \mathbf{Adv}_{\mathsf{F}}^{\mathrm{LENPRF}}(\mathsf{D}) \leq \epsilon$, where the maximum is taken over all distinguishers D that makes q construction queries with total $\sigma = (b_1 + \ldots + b_q)$ blocks, where b_i is the number of blocks requested at i-th construction query. It also makes total q_p primitive queries and runs for time at most t. As before, for information theoretic distinguisher, we omit the time parameter t and in the rest of this chapter, we assume the distinguisher is information theoretic.

Remarks 3.4.2. *The length expanding PRF is a weaker notion than the notion of variable output length PRF [10]. For a length expanding PRF, if two queries have the same input with different numbers of requesting blocks, then one output is a prefix of the other. In the case of variable output length PRF, outputs for two queries are completely random, even if they have the same input with a different number of requesting blocks.*

3.4.2 Security of ppTES

In this section, we show that if $\pi_1, \pi_2 \xleftarrow{\$} \mathsf{Perm}(n)$ are two independently sampled n-bit public random permutations, $K \xleftarrow{\$} \{0,1\}^n$ be a uniformly sampled n-bit key, H is an $(\epsilon_{\mathrm{axu}}, \epsilon_{\mathrm{reg}})$-AXUAR n-bit keyed hash function and $\mathsf{F}_K^{\pi_2}$ is a secure public permutation based length expanding PRF, then ppTES is a public permutation based secure TES

Algorithm for ρ

```
initialize:
  for all $x \in \{0,1\}^n$
    $\mathbb{T}[x] \leftarrow \bot; \mathbb{L}[x] \leftarrow \bot$;
  end for;
on input $(x, b) \neq (x', b')$;
  if $x = x'$
    if $b > b'$, then
      $Y \overset{\Delta}{=} (y_{b'+1}, y_{b'+2}, \ldots, y_b) \overset{\$}{\leftarrow} \{0,1\}^{n(b-b')}$;
      $\mathbb{T}[x] \leftarrow \mathbb{T}[x] \| Y$; $\mathbb{L}[x] \leftarrow b$; **return** $\mathbb{T}[x]$;
    else **return** $\mathbb{T}[x']_{1,\ldots,b}$;
    end if;
  else
    $Y \overset{\Delta}{=} (y_1, \ldots, y_b) \overset{\$}{\leftarrow} \{0,1\}^{nb}$;
    $\mathbb{T}[x] \leftarrow Y$; $\mathbb{L}[x] \leftarrow b$;
    **return** $\mathbb{T}[x]$;
  end if;
```

Figure 3.4.2: Algorithm corresponding to a length expanding random function. $\mathbb{T}[x]_{1,\ldots,b}$ denotes the first b many blocks stored at the x-th entry of table $\mathbb{T}$.

against all $(q_e, q_d, q_{p_1} + q_{p_2}, \ell, \sigma)$ information theoretic adaptive CCA distinguishers that make q_e many encryption, q_d many decryption queries with total σ many blocks queried among all $q \overset{\Delta}{=} q_e + q_d$ queries and ℓ is the maximum number of message blocks present in a single encryption or decryption query. Moreover, it also makes q_{p_1} primitive queries to π_1 and q_{p_2} primitive queries to π_2. Formally, the following result bounds the tSPRP advantage of ppTES in the public permutation model.

Theorem 3.4.3. *Let $\mathcal{K}_h$ be a finite and non-empty set, $\pi_1, \pi_2 \xleftarrow{\$} \mathsf{Perm}(n)$ be two independently sampled n-bit public random permutations and $K \xleftarrow{\$} \{0,1\}^n$ be an n-bit random key. Let $\mathsf{H} : \mathcal{K}_h \times \{0,1\}^* \rightarrow \{0,1\}^n$ be an $(\epsilon_{\text{axu}}, \epsilon_{\text{reg}})$-*AXUAR *$n$-bit keyed hash function. Let $\mathsf{F}_K^{\pi_2}$ be a secure *LENPRF*. Then, for any $(q_e, q_d, q_{p_1} + q_{p_2}, \ell, \sigma)$ information theoretic adaptive *CCA* distinguisher D against the *tSPRP* security of $\mathsf{ppTES}[\pi_1, \pi_2, K, \mathsf{H}]$ in the public permutation model, there exists a *LENPRF* adversary B against the length expanding PRF security of $\mathsf{F}_K^{\pi_2}$ in the public permutation model, where σ is the total number of message blocks queried, such that*

$$\mathbf{Adv}_{\mathsf{ppTES}}^{\text{tSPRP}}(\mathsf{D}) \quad \leq \quad \mathbf{Adv}_{\mathsf{F}}^{\text{LENPRF}}(\mathsf{B}) + q^2\epsilon_{\text{axu}} + 2qq_{p_1}\epsilon_{\text{reg}} + \frac{q^2}{2^{n+1}} + \frac{q(q-1)}{2^{n+1}}.$$

The proof of this result is given in section 3.5.

3.5 Proof of Theorem 3.4.3

As a matter of convenience, we refer to the construction $\mathsf{ppTES}[\pi_1, \pi_2, K, \mathsf{H}]$ as simply ppTES when the underlying primitives are assumed to be understood.

3.5.1 Initial Set Up

By Theorem 2.3.1, we have

$$\mathbf{Adv}_{\mathsf{ppTES}}^{\text{tSPRP}}(\mathsf{D}) \leq \mathbf{Adv}_{\mathsf{ppTES}}^{\pm\text{rnd}}(\mathsf{D}) + \frac{q(q-1)}{2^{n+1}}, \tag{3.1}$$

where n is the minimum message length allowed for ppTES. Therefore, we bound the ±rnd advantage of ppTES. Let D be any information theoretic non-trivial adaptive deterministic CCA distinguisher with access to the oracles in either of the following two worlds: in the real world, it interacts with $\mathcal{O}_{\text{re}} = (\mathsf{ppTES.Enc}_{K,K_h}^{\pi_1,\pi_2}, \mathsf{ppTES.Dec}_{K,K_h}^{\pi_1,\pi_2}, \pi_1^{\pm}, \pi_2^{\pm})$ for an n-bit random key K, a random hash key K_h and two independent n-bit random permutations π_1 and π_2 or in the ideal world it interacts with $\mathcal{O}_{\text{id}} = (\$_0, \$_1, \pi_1^{\pm}, \pi_2^{\pm})$, where $\$_0$ and $\$_1$ are two independent random functions that output uniform random strings for every distinct input. Now, our goal is to upper bound the maximum advantage in distinguishing the real world from the ideal one.

For doing this, as the first step of the proof, we replace $\mathsf{F}_K^{\pi_1,\pi_2}$ with the function ρ as described in Fig. 3.4.2. We call the resulting construction as ppTES^*.

This replacement comes at the cost of the length expanding PRF security of $\mathsf{F}_K^{\pi'}$ in the random permutation model, where the PRF adversary B simulates D as follows: it first samples a hash key $K_h \xleftarrow{\$} \mathcal{K}_h$ and an n-bit random permutation $\pi \xleftarrow{\$} \mathsf{Perm}(n)$. Then, for any input (M, T), it computes

$$Z \leftarrow \pi_1(\mathsf{H}_{K_h}(\mathbf{M_R}\|T) \oplus \mathbf{M_L}) \oplus \mathsf{H}_{K_h}(\mathbf{M_R}\|T) \oplus \mathbf{M_L}.$$

Then it calls its own oracle with $(Z, \lceil\frac{|M|}{n}\rceil)$ as input and receives the $n\lceil\frac{|M|}{n}\rceil$ bit output $\mathbf{S}$. Then it masks the first $|\mathbf{M_R}|$ bits of $\mathbf{S}$ with $\mathbf{M_R}$ and produces the ciphertext blocks $\mathbf{C_R}$ which is hashed along with T and the hash output is masked with $\pi_1(\mathsf{H}_{K_h}((\mathbf{M_R}\|T) \oplus \mathbf{M_L}))$ to generate the first ciphertext block $\mathbf{C_L}$. For any primitive query x made by D to π_1, B accordingly returns the value $\pi_1(x)$. Similarly, it returns the response for backward query to π_1. For any primitive query x made by D to π_2, B forwards the query to its own oracle and returns the received response. Similarly, it returns the response for backward query to π_2. Finally, B outputs the same bit as returned by D. Therefore, we have

$$\mathbf{Adv}^{\pm\mathrm{rnd}}_{\mathsf{ppTES}}(\mathsf{D}) \leq \mathbf{Adv}^{\mathrm{LENPRF}}_{\mathsf{F}}(\mathsf{B}) + \underbrace{\mathbf{Adv}^{\pm\mathrm{rnd}}_{\mathsf{ppTES}^*}(\mathsf{D})}_{\delta^*}. \tag{3.2}$$

3.5.2 Attack Transcript

Our main goal is to bound δ^*, i.e., we need to distinguish the two worlds: the real world $\mathcal{O}_{\mathrm{re}} = (\mathsf{ppTES}^*.\mathsf{Enc}^{\pi_1,\pi_2}_{K,K_h}, \mathsf{ppTES}^*.\mathsf{Dec}^{\pi_1,\pi_2}_{K,K_h}, \pi_1^{\pm}, \pi_2^{\pm})$ from the ideal world $\mathcal{O}_{\mathrm{id}} = (\$_0, \$_1, \pi_1^{\pm}, \pi_2^{\pm})$, where K is an n-bit random key, K_h is a random hash key and π_1, π_2 are two independent n-bit random permutations. Since we consider the maximum distinguishing advantage, let us assume that D is the information theoretic non-trivial adaptive CCA distinguisher for which the distinguishing advantage is maximum. Let D makes q_e (resp. q_d) encryption (resp. decryption) queries and q_{p_1} primitive queries to π_1 and q_{p_2} primitive queries to π_2. Since our proof is in the random permutation model, D can query the primitive in forward and reverse directions. After the interaction is over, the real world returns the hash key K_h and the ideal world samples a dummy hash key $K_h \xleftarrow{\$} \mathcal{K}_h$ and returns it to D. Finally, D outputs a single bit. Let $\tau \triangleq \{(T^1, M^1, C^1), (T^2, M^2, C^2), \ldots, (T^q, M^q, C^q)\}$ be the list of construction queries and responses (i.e., including encryption and decryption queries), $\tau_{p_1} \triangleq \{(x_1, y_1), (x_2, y_2), \ldots, (x_{q_{p_1}}, y_{q_{p_1}})\}$ and $\tau_{p_2} \triangleq \{(u_1, v_1), (u_2, v_2), \ldots, (u_{q_{p_2}}, v_{q_{p_2}})\}$ be the two list of primitive queries and responses to π_1 and π_2 respectively made by D. The triplet $\tau' = (\tau, \tau_{p_1}, \tau_{p_2}, K_h)$ constitutes the query transcript of the attack.

3.5.3 Definition and Probability of Bad Transcripts

In this section, we define bad transcripts and bound their probability in the ideal world. From transcript τ', we derive the following notation: for $i \in q$, $U_i = M_1^i \oplus \mathsf{H}_{k_h}(M_2^i \| \ldots \| M_{l_i}^i \| T^i)$, $V_i = C_1^i \oplus \mathsf{H}_{k_h}(C_2^i \| \ldots \| C_{l_i}^i \| T^i)$ and $Z_i = U_i \oplus V_i$. Having set up the notation, we identify an event to be bad if for any two construction queries there is a collision in the Z_i values or there is a non-trivial input or output collision of the permutation π_1.

Definition 3.5.1 (Bad Transcript for ppTES^*). *An attainable transcript $\tau' = (\tau, \tau_p, \tau'_p, K_h)$ is called* **bad** *for ppTES^* if any of the following conditions hold:*

- $\mathsf{B.1}$ *: $\exists\ i \neq j \in [q]$ such that, $U^i = U^j$.*

- $\mathsf{B.2}$ *: $\exists\ i \neq j \in [q]$ such that $V^i = V^j$.*

- $\mathsf{B.3}$ *: $\exists\ i \in [q]$ and $j \in [q_p]$ such that $U^i = x_j$.*

- $\mathsf{B.4}$ *: $\exists\ i \in [q]$ and $j \in [q_p]$ such that $V^i = y_j$.*

- $\mathsf{B.5}$ *: $\exists\ i, j \in [q]$ such that $Z^i = Z^j$.*

Lemma 3.5.2. *Let T_{id} be the random variable that takes the transcript resulting from the interaction between the distinguisher and the ideal world and $\mathcal{V}_{\mathrm{b}}$ be the set of all* *attainable* **bad** *transcripts for* ppTES^*. *Then we have,*

$$\Pr[\mathsf{T}_{\rm id} \in \mathcal{V}_{\rm b}] \le \epsilon_{\rm bad} = q^2\epsilon_{\rm axu} + 2qq_p\epsilon_{\rm reg} + \frac{q^2}{2^{n+1}}.$$

Proof. By the union bound,

$$\Pr[\mathsf{T}_{\rm id} \in \mathcal{V}_{\rm b}] \le \sum_{i=1}^{4} \Pr[\mathsf{B.i}] + \Pr[\mathsf{B.5} \mid \overline{\mathsf{B.1}} \wedge \overline{\mathsf{B.2}} \wedge \overline{\mathsf{B.3}} \wedge \overline{\mathsf{B.4}}]. \tag{3.3}$$

In the following, we bound the probability of all the bad events individually. The lemma will follow by adding the individual bounds.

Bounding $\mathsf{B.1}$. For two fixed values of i and j, we compute the probability of the event $U^i = U^j$. Note that $U^i = U^j$ implies the hash equation: $\mathsf{H}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{H}_{K_h}(\mathbf{M}^j_{\mathbf{R}} \| T^j) = M^i_1 \oplus M^j_1$. By fixing the value of all other random variables in the hash equation, the probability of this event is bounded by the AXU advantage of the hash function. Therefore, by summing over all possible choices of i and j, we have

$$\Pr[\mathsf{B.1}] \le \binom{q}{2}\epsilon_{\rm axu}. \tag{3.4}$$

Bounding $\mathsf{B.2}$. This event is similar to that of $\mathsf{B.1}$ where we consider the output collision of π. Note that, $V^i = V^j$ implies the hash equation: $\mathsf{H}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{H}_{K_h}(\mathbf{C}^j_{\mathbf{R}} \| T^j) = C^i_1 \oplus C^j_1$. Similar to $\mathsf{B.1}$, we bound the event using the AXU advantage of the hash function and thus we have

$$\Pr[\mathsf{B.2}] \le \binom{q}{2}\epsilon_{\rm axu}. \tag{3.5}$$

Bounding $\mathsf{B.3}$. For two fixed values of i and j, we compute the probability of the event $U^i = x_j$. Note that $U^i = x_j$ implies the hash equation: $\mathsf{H}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) = M^i_1 \oplus x_j$. By fixing the value of all other random variables in the hash equation, the probability of this event is bounded by the AR advantage of the hash function. Therefore, by summing over all possible choices of i and j, we have

$$\Pr[\mathsf{B.3}] \leq q q_{p_1} \epsilon_{\text{reg}}. \tag{3.6}$$

Bounding $\mathsf{B.4}$. For two fixed values of i and j, we compute the probability of the event $V^i = y_j$. Note that $V^i = y_j$ implies the hash equation: $\mathsf{H}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) = C^i_1 \oplus y_j$. Similar to $\mathsf{B.3}$, we bound the event using the AR advantage of the hash function and thus we have

$$\Pr[\mathsf{B.4}] \leq q q_{p_1} \epsilon_{\text{reg}}. \tag{3.7}$$

Bounding $\mathsf{B.5} \mid \overline{\mathsf{B.1}} \wedge \overline{\mathsf{B.2}} \wedge \overline{\mathsf{B.3}} \wedge \overline{\mathsf{B.4}}$. To bound this event, we first fix the values of i and j. Note that $Z^i = Z^j$ implies $U^i \oplus V^i = U^j \oplus V^j$. Now, due to the condition, we have $U^i \neq U^j$ and $V^i \neq V^j$. Therefore, we obtain the following hash equation:

$$\mathsf{H}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{H}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{H}_{K_h}(\mathbf{M}^j_{\mathbf{R}} \| T^j) \oplus \mathsf{H}_{K_h}(\mathbf{C}^j_{\mathbf{R}} \| T^j) = W, \tag{3.8}$$

where $W = M^i_1 \oplus M^j_1 \oplus C^i_1 \oplus C^j_1$. W.l.o.g, we assume that $i < j$. If the j-th query is an encryption query, then C^j_1 is uniformly distributed in the ideal world and if the j-th query is a decryption query, then M^j_1 is uniformly distributed in the ideal world. Combining the above two arguments and by varying over all possible choices of indices, we have

$$\Pr[\mathsf{B.5}] \leq \frac{\binom{q}{2}}{2^n}. \tag{3.9}$$

The proof follows from Equation (3.3)-Equation (3.7) and Equation (3.9). □

3.5.4 Analysis of Good Transcript

In this section, we show that for a good transcript $\tau' = (\tau, \tau_{p_1}, \tau_{p_2}, k_h)$, realizing τ' is almost as likely in the real world as in the ideal world.

Lemma 3.5.3. *Let $\tau' = (\tau, \tau_{p_1}, \tau_{p_2}, k_h)$ be a good transcript. Then*

$$\frac{\Pr[\mathsf{T}_{\text{re}} = \tau']}{\Pr[\mathsf{T}_{\text{id}} = \tau']} \geq 1.$$

Proof. Since, in the ideal world, the encryption and the decryption oracle behave perfectly random, we have

$$\Pr[\mathsf{T}_{\mathrm{id}} = \tau'] = \frac{1}{|\mathcal{K}_h|} \frac{1}{\mathbf{P}(2^n, q_{p_1})} \cdot \frac{1}{\mathbf{P}(2^n, q_{p_2})} \cdot \frac{1}{2^{n\sigma}}, \tag{3.10}$$

where σ is the total number of blocks queried among all q construction queries that include encryption and decryption queries.

Real Interpolation Probability. Since τ' is a good transcript, all the inputs and outputs of π_1 are fresh. Moreover, all Z_i values are distinct. Therefore, the outputs of ρ are all uniformly random. Since, there are total $q_{p_1} + q$ many invocations of π_1, we have

$$\Pr[\mathsf{T}_{\mathrm{re}} = \tau'] = \frac{1}{|\mathcal{K}_h|} \frac{1}{\mathbf{P}(2^n, q_{p_1} + q)} \cdot \frac{1}{\mathbf{P}(2^n, q_{p_2})} \cdot \frac{1}{(2^n)^{\sigma - q}}. \tag{3.11}$$

By doing a simple algebraic calculation, it is easy to see that the ratio of Equation (3.11) to Equation (3.10) is at least 1 and hence proves the result. □

By combining Lemma 3.5.2, Lemma 3.5.3, Theorem 2.4.1, Equation (3.1) and Equation (3.2), the result follows. □

3.6 ppCTR: Public Permutation Based Length Expanding PRF

In this section, we propose ppCTR, a public permutation based length expanding PRF. Our proposed construction is a public permutation variant of the block cipher based standard counter mode encryption where the block cipher is replaced by a single round EM [43] cipher. The working principle of ppCTR is as follows: it takes an n-bit public random permutation π and an n-bit random key k from $\mathrm{GF}(2^n)$. Then for any n-bit input value z and an integer b, it outputs b many blocks where the j-th block

S_j is defined as follows:

$$S_j \triangleq \pi(z \oplus \gamma^j k) \oplus \gamma^j \cdot k,\ \ j \in [b],$$

where γ is the root of any fixed primitive polynomial of degree n of GF(2^n). In the following section, we state and prove that ppCTR is a public permutation based secure LENPRF against all adversaries that make roughly $2^{n/2}$ construction and primitive queries. It is needless to say that the above bound is tight as EM cipher is known to have a tight birthday bound security [43].

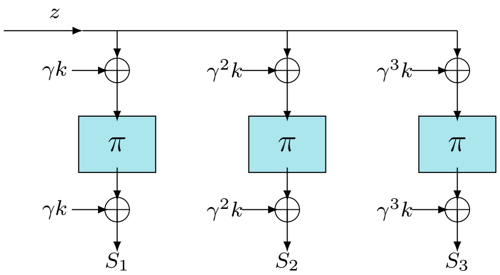

Figure 3.6.1: ppCTR construction with an n-bit input z and an integer $b = 3$ and corresponding output $S_1 \| S_2 \| S_3$. π is the public random permutation, k is the key and γ is the root of a primitive polynomial of GF(2^n).

3.6.1 Security Analysis of ppCTR

In this section, we show that ppCTR is a public permutation based length expanding PRF.

Theorem 3.6.1. *Let $\pi \xleftarrow{\$} \mathsf{Perm}(n)$ be an n-bit public random permutation and let $K \xleftarrow{\$} \{0,1\}^n$ be an n-bit random key. Then, for any (q, q_p, σ) adversary D against the LENPRF security of $\mathsf{ppCTR}[\pi, K]$, we have*

$$\mathbf{Adv}^{\text{LENPRF}}_{\mathsf{ppCTR}}(\mathsf{D}) \ \leq\ \frac{\sigma^2}{2^n} + \frac{2\sigma q_p}{2^n},$$

where σ is the total number of blocks queried across all q queries.

Proof. Let $\mathsf{D}_{\max}$ be the distinguisher with maximum distinguishing advantage in distinguishing the following two worlds: (a) in the real world it interacts with $\mathcal{O}_{\mathrm{re}} = (\mathsf{ppCTR}[\pi, K], \pi^{\pm})$ for a random n-bit key K and a random n-bit permutation π and (b) in the ideal world it has access to $\mathcal{O}_{\mathrm{id}} = (\rho, \pi^{\pm})$, where ρ works in the similar way as shown in Fig. 3.4.2. It makes q construction queries and q_p primitive queries. After the interaction is over, the real world returns K to $\mathsf{D}_{\max}$ and the ideal world randomly samples a dummy key $K \xleftarrow{\$} \{0,1\}^n$ and returns to $\mathsf{D}_{\max}$. Finally, $\mathsf{D}_{\max}$ outputs a bit. Let $\tau \overset{\Delta}{=} \{(z_1, b_1, \mathbf{S}^1), (z_2, b_2, \mathbf{S}^2), \ldots, (z_q, b_q, \mathbf{S}^q)\}$ be the list of construction queries and responses, where $\mathbf{S}^i = (S^i_1, \ldots, S^i_{b_i})$ and $\tau_p \overset{\Delta}{=} \{(x_1, y_1), (x_2, y_2), \ldots, (x_{q_p}, y_{q_p})\}$ be the list of primitive queries and responses to π made by $\mathsf{D}_{\max}$. Let $\sigma = (b_1 + \ldots + b_q)$ denote the total number of blocks queried across all q queries. The triplet $\tau' = (\tau, \tau_p, K)$ constitutes the query transcript of the attack. We define a relation $\sim$ over τ such that $(z_i, b_i, \mathbf{S}_i) \sim (z_j, b_j, \mathbf{S}_j)$ if and only if $z_i = z_j$. Thus, $\sim$ induces a partition on τ and let us assume we have r many such partitions. Each partition contains c_i many elements and therefore, $c_1 + \ldots + c_r = q$. Note that, there exists a total ordering among b_i values in each component. This allows us to sort the elements of each component in the ascending order of their b values. After rearrangement, we have the following:

$$\begin{cases} \{(z_1, b^1_1, \mathbf{S}^1_1), \ldots, (z_1, b^1_{c_1}, \mathbf{S}^1_{c_1})\} \\ \{(z_2, b^2_1, \mathbf{S}^2_1), \ldots, (z_2, b^2_{c_1}, \mathbf{S}^2_{c_2})\} \\ \quad \vdots \qquad\quad \vdots \qquad\quad \vdots \qquad\quad \vdots \\ \{(z_r, b^r_1, \mathbf{S}^r_1), \ldots, (z_r, b^r_{c_1}, \mathbf{S}^r_{c_1})\} \end{cases}$$

Note that, for each $i \in [r]$, $b^i_{c_i} \geq b^i_{c_i - 1} \geq \ldots \geq b^i_1$ and $\mathbf{S}^i_j$ is a prefix of $\mathbf{S}^i_{j+1}$ for all $j \in [c_i]$.

3.6.2 Definition and Probability of Bad Transcripts

In this section, we define bad transcripts and bound their probability in the ideal world. Informally, we define an event to be bad if it introduces any non-trivial input or output collision of the permutation π.

Definition 3.6.2. (**Bad Transcript for ppCTR**) : *An attainable transcript* $\tau' = (\tau, \tau_p, K)$ *is called a* **bad** *transcript for* ppCTR *if any of the following conditions hold:*

- B.1 : $\exists\, i \neq j \in [r]$, $\alpha \in [\ell_{c_i}]$ *and* $\beta \in [\ell_{c_j}]$ *such that* $z_i \oplus \gamma^\alpha K = z_j \oplus \gamma^\beta K$.

- B.2 : $\exists\, i \in [r]$, $j \in [q_p]$ *and* $\alpha \in [\ell_{c_i}]$ *such that* $z_i \oplus \gamma^\alpha K = x_j$.

- B.3 : $\exists\, i \neq j \in [r]$, $\alpha \in [\ell_{c_i}]$ *and* $\beta \in [\ell_{c_j}]$ *such that* $S^i_\alpha \oplus \gamma^\alpha K = S^j_\beta \oplus \gamma^\beta K$.

- B.4 : $\exists\, i \in [r]$, $j \in [q_p]$ *and* $\alpha \in [\ell_{c_i}]$ *such that* $S^i_\alpha \oplus \gamma^\alpha K = y_j$.

Lemma 3.6.3. *Let* T_{id} *be the random variable that takes the transcript resulting from the interaction between the distinguisher and the ideal world and* $\mathcal{V}_{\text{b}}$ *be the set of all attainable* **bad** *transcripts for* ppCTR*. Then we have,*

$$\Pr[\mathsf{T}_{\text{id}} \in \mathcal{V}_{\text{b}}] \leq \epsilon_{\text{bad}} = \frac{\sigma^2}{2^n} + \frac{2\sigma q_p}{2^n}.$$

Proof. By the union bound,

$$\Pr[\mathsf{T}_{\text{id}} \in \mathcal{V}_{\text{b}}] \leq \sum_{i=1}^{4} \Pr[\mathsf{B.i}]. \tag{3.12}$$

In the following, we bound the probability of all the bad events individually. The lemma will follow by adding the individual bounds.

Bounding B.1. To bound this event, we first fix a value of the indices $i \neq j \in [r]$ and $\alpha \in [\ell_{c_i}], \beta \in [\ell_{c_j}]$. For such a fixed choice of indices, we bound the probability of the event $z_i \oplus \gamma^\alpha K = z_j \oplus \gamma^\beta K$. Now, if $\alpha = \beta$, then the probability of the event is zero as $z_i \neq z_j$. Therefore, we assume that $\alpha \neq \beta$. For this choice of indices, we write the event as

$$K = (\gamma^\alpha \oplus \gamma^\beta)^{-1}(z^i \oplus z^j). \tag{3.13}$$

The probability of Equation (3.13) is 2^{-n}, due to the randomness of the key K. Therefore, by varying over all possible choices of i, j, α and β, we have

$$\Pr[\mathsf{B.1}] \leq \frac{\sigma^2}{2^{n+1}}. \tag{3.14}$$

Bounding $\mathsf{B.2}$. For a fixed choice of $i \in [r], j \in [q_p]$ and $\alpha \in [\ell_{c_i}]$, the probability of the event $K = \gamma^{-\alpha}(z^i \oplus x_j)$ is bounded by 2^{-n} due to the randomness of K. Therefore, by varying over all possible choices of i, j and α, we have

$$\Pr[\mathsf{B.2}] \leq \frac{q_p}{2^n}(b_{c_1} + \cdots + b_{c_r}) \leq \frac{\sigma q_p}{2^n}. \tag{3.15}$$

Bounding $\mathsf{B.3}$. Bounding this event is similar to that of $\mathsf{B.1}$. To bound this event, we first fix the value of the indices $i \neq j \in [r]$ and $\alpha \in [\ell_{c_i}], \beta \in [\ell_{c_j}]$. For such a fixed choice of indices, we bound the probability of the event $S^i_\alpha \oplus \gamma^\alpha K = S^j_\beta \oplus \gamma^\beta K$. Now we have the following two cases:

- **Case A.** Let us consider that $\alpha = \beta$. As $i \neq j$, without loss of generality, we assume that $i < j$. Therefore, the event boils down to $S^i_\alpha = S^j_\alpha$, which is bounded by 2^{-n} due to the randomness of S^j_α. Therefore, by varying over all possible choices of i, j and α, we have

$$\Pr[\mathsf{B.3}] \leq \frac{\sigma^2}{2^{n+1}}$$

- **Case B.** if $\alpha \neq \beta$, then the event can be equivalently written as

$$K = (\gamma^\alpha \oplus \gamma^\beta)^{-1}(S^i_\alpha \oplus S^i_\beta). \tag{3.16}$$

  Since, $\alpha \neq \beta$, we have $\gamma^\alpha \oplus \gamma^\beta \neq 0$ and therefore, the probability of Equation (3.16) is 2^{-n} due to the randomness of the key K. Therefore, by varying over all possible choices of i, j, α and β, we have

$$\Pr[\mathsf{B.3}] \leq \frac{\sigma^2}{2^{n+1}}.$$

By taking the maximum of the above two, we have

$$\Pr[\mathsf{B.3}] \leq \frac{\sigma^2}{2^{n+1}}. \tag{3.17}$$

Bounding $\mathsf{B.4}$. Bounding this event is exactly identical to that of $\mathsf{B.2}$, where we use the randomness of K to bound the event. Therefore, we have

$$\Pr[\mathsf{B.4}] \leq \frac{q_p}{2^n}(b_{c_1} + \cdots + b_{c_r}) \leq \frac{\sigma q_p}{2^n}. \tag{3.18}$$

The proof follows from Equation (3.12) and Equation (3.14)-Equation (3.18). □

3.6.3 Analysis of Good Transcript

In this section, we show that for a good transcript $\tau' = (\tau, \tau_p, k)$, realizing τ' is almost as likely in the real world as in the ideal world.

Lemma 3.6.4. *Let $\tau' = (\tau, \tau_p, k)$ be a good transcript. Then*

$$\frac{\Pr[\mathsf{T}_{\text{re}} = \tau']}{\Pr[\mathsf{T}_{\text{id}} = \tau']} \geq 1.$$

Proof. Consider a good transcript $\tau' = (\tau, \tau_p, k)$. In the ideal world, ρ randomly samples nb_{c_i} bit output for i-th class and the key k is sampled uniformly from $\{0,1\}^n$ and independent of all other sampled random variables. Thus, we have

$$\Pr[\mathsf{T}_{\text{id}} = \tau'] = \frac{1}{2^n} \cdot \frac{1}{\mathbf{P}(2^n, q_p)} \cdot \prod_{i=1}^{r} \frac{1}{2^{nqb_{c_i}}}. \tag{3.19}$$

For computing the real interpolation probability, as τ' is good, all the inputs and outputs of π are distinct. The total number of π invocations including the primitive queries is $(b_{c_1} + \ldots + b_{c_r} + q_p)$. Therefore,

$$\Pr[\mathsf{T}_{\text{re}} = \tau'] = \frac{1}{2^n} \cdot \frac{1}{\mathbf{P}(2^n, b_{c_1} + \ldots + b_{c_r} + q_p)}. \tag{3.20}$$

It is trivial to see that the ratio of Equation (3.20) to Equation (3.19) is at least 1. Hence, the result of Lemma 3.6.4 follows. Finally, by combining Lemma 3.6.3, Lemma 3.6.4 and Theorem 2.4.1, the result of Theorem 3.6.1 follows. □

3.6.4 ppHCTR : An Instantiation of ppTES with ppCTR and PolyHash

We instantiate the public permutation based length expanding PRF $\mathsf{F}_k^{\pi_2}$ of $\mathsf{ppTES}[\pi_1, \pi_2, k, \mathsf{H}]$ with $\mathsf{ppCTR}[\pi_2, k]$ and its underlying AXUAR hash function H_{k_h} with the PolyHash function Poly_{k_h}, as described in Equation (2.4), to realize a practical candidate of a public permutation based TES, referred to as $\mathsf{ppHCTR}[\pi_1, \pi_2, k, \mathsf{Poly}_{k_h}]$. We assume that the tweak is μ blocks long, i.e., $\mathtt{tw} = n\mu$ and thus, for any $i \in [q]$, the maximum degree of $\mathsf{Poly}_{k_h}(M_2^i \| \ldots \| M_{l_i}^i \| T^i)$ is $\hat{l}_i + \mu$, where $\hat{l}_i = \lceil \frac{|\mathbf{M}_\mathbf{R}^i|}{n} \rceil$. Since $\hat{l}_i \leq \ell$ for all $i \in [q]$, where ℓ denotes the maximum number of message blocks among all q queries, therefore the AXU and the AR advantage of the PolyHash function is $(\ell + \mu)/2^n$. Note that ppHCTR requires two independent n-bit random permutations π_1 and π_2, an n-bit random key K and an independent n-bit random hash key K_h for the PolyHash function. Security result of ppHCTR follows trivially from Theorem 3.4.3 and Theorem 3.6.1 which can be summarized as follows:

Theorem 3.6.5. *Let $\pi_1, \pi_2 \xleftarrow{\$} \mathsf{Perm}(n)$ be two independent n-bit public random permutations and let $K \xleftarrow{\$} \{0,1\}^n$ be an n-bit random key. Let $K_h \xleftarrow{\$} \{0,1\}^n$ be an n-bit random hash key of* PolyHash *function as described in Equation* (2.4)*. Then, for any $(q_e, q_d, q_{p_1} + q_{p_2}, \ell, \sigma)$ information theoretic non-trivial adaptive* CCA *distinguisher D against the* tSPRP *security of $\mathsf{ppHCTR}[\pi_1, \pi_2, K, \mathsf{Poly}_{K_h}]$, we have*

$$\mathbf{Adv}_{\mathsf{ppHCTR}}^{\mathrm{tSPRP}}(\mathsf{D}) \;\leq\; \frac{\sigma^2}{2^n} + \frac{2\sigma q_{p_2}}{2^n} + \frac{q^2\ell}{2^n} + \frac{2qq_{p_1}\ell}{2^n} + \frac{\mu q^2}{2^n} + \frac{2\mu q q_p}{2^n} + \frac{q^2}{2^{n+1}} + \frac{q(q-1)}{2^{n+1}},$$

where $q = q_e + q_d$, ℓ is the maximum number of message blocks and μ is the number of tweak blocks.

3.7 ppHCTR+ : A Single-Keyed Variant of ppHCTR

In the last section, we have seen that ppHCTR, a public permutation based TES, requires two independent n-bit public random permutations and two independent n-bit keys. In this section, we propose a single permutation and single keyed variant of ppHCTR, referred to as ppHCTR+. The construction is based on an n-bit public random permutation π and an n-bit random hash key of the PolyHash function as described in Equation (2.4). We consider that the tweak size is μ blocks long. The encryption and decryption algorithm of ppHCTR+ is shown in Fig. 3.7.1.

```
$\mathsf{ppHCTR+.Enc}^{\pi}_{k_h}(T, M)$
  1. $(M_1 \| \dots \| M_l) \leftarrow \mathsf{parse}_n(M)$;
  2. $\mathbf{M_L} \leftarrow M_1; \mathbf{M_R} \leftarrow (M_2 \| \dots \| M_l)$;
  3. $U \leftarrow M_L \oplus \mathsf{Poly}_{k_h}(\mathbf{M_R} \| T)$;
  4. $V \leftarrow \pi(U)$; $Z \leftarrow U \oplus V$;
  5. for $j = 1$ to $l - 1$
  6.     $Z_j \leftarrow Z \oplus j$;
  7.     $S_j \leftarrow \pi(Z_j) \oplus Z_j$;
  8. $\mathbf{S} \triangleq (S_1 \| \dots \| S_{l-1})$;
  9. $\mathbf{C_R} \leftarrow \mathbf{M_R} \oplus \mathsf{first}(|\mathbf{M_R}|, \mathbf{S})$;
 10. $C_L \leftarrow V \oplus \mathsf{Poly}_{k_h}(\mathbf{C_R} \| T)$;
 11. return $(\mathbf{C_L} \| \mathbf{C_R})$;
```

```
$\mathsf{ppHCTR+.Dec}^{\pi}_{k_h}(T, C)$
  1. $(C_1 \| \dots \| C_l) \leftarrow \mathsf{parse}_n(C)$;
  2. $\mathbf{C_L} \leftarrow C_1; \mathbf{C_R} \leftarrow (C_2 \| \dots \| C_l)$;
  3. $V \leftarrow C_1 \oplus \mathsf{Poly}_{k_h}(\mathbf{C_R} \| T)$;
  4. $U \leftarrow \pi^{-1}(V)$; $Z \leftarrow U \oplus V$;
  5. for $j = 1$ to $l - 1$
  6.     $Z_j \leftarrow Z \oplus j$;
  7.     $S_j \leftarrow \pi(Z_j) \oplus Z_j$;
  8. $\mathbf{S} \triangleq (S_1 \| \dots \| S_{l-1})$;
  9. $\mathbf{M_R} \leftarrow \mathbf{C_R} \oplus \mathsf{first}(|\mathbf{C_R}|, \mathbf{S})$;
 10. $M_L \leftarrow V \oplus \mathsf{Poly}_{k_h}(\mathbf{M_R} \| T)$;
 11. return $(\mathbf{M_L} \| \mathbf{M_R})$;
```

Figure 3.7.1: ppHCTR+ based on an n-bit public random permutation π and an n-bit random hash key k_h. The left part is the encryption algorithm and the right part is its decryption algorithm.

To see the dataflow of the encryption algorithm, we consider an input message $M = (M_1 \| M_2)$, where $|M_1| = |M_2| = n$, i.e., M consists of two full blocks. Therefore, in step (2) of the algorithm, the variable $\mathbf{M_L}$ is assigned to M_1 and $\mathbf{M_R}$ is assigned to M_2. In step (3) of the algorithm, we evaluate the poly hash Poly_{k_h} on $(M_2 \| T)$ which results to $M_2 \cdot k_h^3 \oplus T \cdot k_h^2 \oplus \langle |M_2| + |T| \rangle \cdot k_h$ which is xored with the n-bit value M_1 to produce U. In step (4), we take the xor of U and $V = \pi(U)$ to produce Z. In steps (6) and (7), we compute the key stream $\mathbf{S} = S_1$ where each $|S_1| = n$ by

$S_1 = \pi(Z \oplus 1) \oplus (Z \oplus 1)$. Since, $|\mathbf{M_R}| = n$, $\mathbf{C_R}$ will be $M_2 \oplus S_1$, which becomes the input along with tweak T to the poly hash function Poly_{k_h}. Evaluation of the poly hash on input $\mathbf{C_R} \| T$ results to $\mathbf{C_R} \cdot k_h^3 \oplus T \cdot k_h^2 \oplus \langle |\mathbf{C_R}| + |T| \rangle \cdot k_h$. Then the result is xored with V to produce $\mathbf{C_L}$, which is returned along with $\mathbf{C_R}$ as the encryption of $M = M_1 \| M_2$. The decryption works in a similar way.

3.7.1 Security Result of ppHCTR+

The security result of ppHCTR+ is as follows:

Theorem 3.7.1. *Let $\pi \xleftarrow{\$} \mathsf{Perm}(n)$ be an n-bit public random permutation and let $K_h \xleftarrow{\$} \{0,1\}^n$ be an n-bit random hash key of* PolyHash *function as described in Equation (2.4). Then, for any $(q_e, q_d, q_p, \ell, \sigma)$ information theoretic non-trivial adaptive* CCA *distinguisher* D *against the* tSPRP *security of* ppHCTR+$[\pi, \mathsf{Poly}_{K_h}]$*, we have*

$$\mathbf{Adv}^{\mathrm{tSPRP}}_{\mathsf{ppHCTR+}}(\mathsf{D}) \;\leq\; \frac{9\sigma^2}{2^n} + \frac{6\mu\sigma^2}{2^n} + \frac{4q_p\sigma(\mu+1)}{2^n} + \frac{q(q-1)}{2^{n+1}},$$

where σ is the total number of message blocks for all $q \stackrel{\Delta}{=} q_e + q_d$ queries and μ is the number of tweak blocks.

3.8 Proof of Theorem 3.7.1

In section 3.6.4, we propose ppHCTR, which uses two independent random permutations and two independent random keys, which allows us to use the generic security result of ppTES in order to derive the security result of ppHCTR. However, for the single keyed variant of it, we cannot use the generic result of ppTES due to the input/output dependency and that demands an independent security proof for ppHCTR+.

For the sake of simplicity, we refer ppHCTR+$[\pi, \mathsf{Poly}_{K_h}]$ as ppHCTR+ when the underlying primitives are assumed to be understood. By Theorem 2.3.1, we have

$$\mathbf{Adv}^{\mathrm{tSPRP}}_{\mathsf{ppHCTR+}}(\mathsf{D}) \leq \mathbf{Adv}^{\pm\mathrm{rnd}}_{\mathsf{ppHCTR+}}(\mathsf{D}) + \frac{q(q-1)}{2^{n+1}}, \tag{3.21}$$

where recall that n is the minimum message length allowed for $\mathsf{ppHCTR+}$. Therefore, we bound the ±rnd advantage of $\mathsf{ppHCTR+}$. Let D be any information theoretic non-trivial adaptive deterministic CCA distinguisher with access to the oracles in either of the following two worlds: in the real world it interacts with $\mathcal{O}_{\mathrm{re}} = (\mathsf{ppHCTR+.Enc}^{\pi}_{K_h}, \mathsf{ppHCTR+.Dec}^{\pi}_{K_h}, \pi^{\pm})$ for an n-bit random hash key K_h and a random n-bit permutation π or in the ideal world it interacts with $\mathcal{O}_{\mathrm{id}} = (\$_0, \$_1, \pi^{\pm})$, where $\$_0$ and $\$_1$ are two independent random functions such that for any input, it responds with uniform values. Now, our goal is to upper bound the maximum advantage in distinguishing the real world from the ideal one.

Let D be the maximum distinguishing advantage achieving distinguisher that makes q_e (resp. q_d) encryption (resp. decryption) queries and q_p primitive queries. After the interaction is over, the underlying hash key is revealed to D and finally, D outputs a bit. Let $\tau \stackrel{\Delta}{=} \{(T^1, M^1, C^1), (T^2, M^2, C^2), \ldots, (T^q, M^q, C^q)\}$ be the list of construction queries and responses and $\tau_p \stackrel{\Delta}{=} \{(x_1, y_1), (x_2, y_2), \ldots, (x_{q_p}, y_{q_p})\}$ be the list of primitive queries and responses where each T^i is exactly μ blocks long. The triplet $\tau' = (\tau, \tau_p, K_h)$ constitutes the query transcript of the attack. Now, we characterize the set of bad transcripts and good transcripts.

3.8.1 Definition and Probability of Bad Transcripts

In this section, we define bad transcripts and bound their probabilities in the ideal world. The defining criterion of the bad event is any non-trivial collision in the input or output of the permutation. As defined in Fig. 3.7.1, $\mathbf{M^i_R}$ denotes $M^i_2 \| \ldots \| M^i_{l_i}$ and $\mathbf{C^i_R}$ denotes $C^i_2 \| \ldots \| C^i_{l_i}$. Moreover, for a transcript τ', we denote $U^i = \mathsf{Poly}_{K_h}(\mathbf{M^i_R} \| T^i) \oplus M^i_1, V^i = \mathsf{Poly}_{K_h}(\mathbf{C^i_R} \| T^i) \oplus C^i_1$ and $Z^i_\alpha = U^i \oplus V^i \oplus \langle \alpha \rangle$.

Definition 3.8.1. **(Bad Transcript for $\mathsf{ppHCTR+}$)** : *An attainable transcript*

$\tau' = (\tau, \tau_p, K_h)$ is called a **bad** *transcript for* ppHCTR+ *if any of the following conditions hold:*

- B.1 *: $\exists\ i \neq j \in [q]$ such that, $U^i = U^j$.*
- B.2 *: $\exists\ i, j \in [q]$ and $\alpha \in [l_j - 1]$ such that, $U^i = Z^j_\alpha$.*
- B.3 *: $\exists\ i, j \in [q]$, $\alpha \in [l_i - 1]$ and $\beta \in [l_j - 1]$ with $(i, \alpha) \neq (j, \beta)$ such that $Z^i_\alpha = Z^j_\beta$, where $(i, \alpha) \neq (j, \beta)$.*
- B.4 *: $\exists\ i \neq j \in [q]$ such that $V^i = V^j$.*
- B.5 *: $\exists\ i, j \in [q]$ and $\alpha \in [l_j - 1]$ such that $V^i = Z^j_\alpha \oplus M^j_{\alpha+1} \oplus C^j_{\alpha+1}$.*
- B.6 *: $\exists\ i, j \in [q]$, $\alpha \in [l_i - 1]$ and $\beta \in [l_j - 1]$ with $(i, \alpha) \neq (j, \beta)$ such that $Z^i_\alpha \oplus M^i_{\alpha+1} \oplus C^i_{\alpha+1} = Z^j_\beta \oplus M^j_{\beta+1} \oplus C^j_{\beta+1}$.*
- B.7 *: $\exists\ i \in [q]$ and $j \in [q_p]$ such that $U^i = x_j$.*
- B.8 *: $\exists\ i \in [q]$, $j \in [q_p]$ and $\alpha \in [l_i - 1]$ such that $Z^i_\alpha = x_j$.*
- B.9 *: $\exists\ i \in [q]$ and $j \in [q_p]$ such that $V^i = y_j$.*
- B.10 *: $\exists\ i \in [q]$, $j \in [q_p]$ and $\alpha \in [l_i - 1]$ such that $Z^i_\alpha \oplus M^i_{\alpha+1} \oplus C^i_{\alpha+1} = y_j$.*

Lemma 3.8.2. *Let T_{id} be the random variable that takes the transcript resulting from the interaction between the distinguisher and the ideal world and $\mathcal{V}_{\mathrm{b}}$ be the set of all attainable bad transcripts for* ppHCTR+*. Then, by assuming $q \leq \sigma$, we have*

$$\Pr[\mathsf{T}_{\mathrm{id}} \in \mathcal{V}_{\mathrm{b}}] \leq \epsilon_{\mathrm{bad}} = \frac{9\sigma^2}{2^n} + \frac{6\mu\sigma^2}{2^n} + \frac{4q_p\sigma(\mu+1)}{2^n}.$$

Proof. By the union bound,

$$\Pr[\mathsf{T}_{\mathrm{id}} \in \mathcal{V}_{\mathrm{b}}] \leq \sum_{i=1}^{10} \Pr[\mathsf{B.i}]. \tag{3.22}$$

In the following, we bound the probability of all the bad events individually. The lemma will follow by adding the individual bounds.

NOTATION. We consider that the tweak is μ blocks long, i.e., $\mathtt{tw} = n\mu$. Therefore, for any $i \in [q]$, the maximum degree of $\mathsf{Poly}_{k_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i)$ is $\hat{l}_i + \mu$, where $\hat{l}_i \triangleq \lceil \frac{|\mathbf{M}^i_{\mathbf{R}}|}{n} \rceil$. Let $\hat{\ell}_{i,j}$ denotes $\max\{\hat{l}_i, \hat{l}_j\} + \mu$ and $\hat{\sigma} = q\mu + (\hat{l}_1 + \ldots + \hat{l}_q)$ denotes the total number of message blocks of $\mathbf{M}^i_{\mathbf{R}}$ (including the tweak blocks) across all q queries. Therefore, $\sigma = (\hat{\sigma} - q\mu + q)$ which implies that $\sigma - q = \hat{l}_1 + \ldots + \hat{l}_q$. Since, $\hat{\ell}_{i,j} \leq \hat{l}_i + \hat{l}_j + \mu$, we have

$$\sum_{1 \leq i < j \leq q} \hat{\ell}_{i,j} \leq \binom{q}{2}\mu + \sum_{1 \leq i < j \leq q} (\hat{l}_i + \hat{l}_j) \leq (q-1)\hat{\sigma} \leq q\sigma + \mu q^2. \tag{3.23}$$

Bounding $\mathsf{B.1}$. Bounding this event is equivalent to bounding

$$\mathsf{Poly}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{M}^j_{\mathbf{R}} \| T^j) = M^i_1 \oplus M^j_1.$$

If $\mathbf{M}^i_{\mathbf{R}} \| T^i = \mathbf{M}^j_{\mathbf{R}} \| T^j$ then the probability of this event is zero, otherwise it is bounded by the AXU advantage of the $\mathsf{PolyHash}$ and hence from Equation (3.23) and by assuming $q \leq \sigma$, we have

$$\Pr[\mathsf{B.1}] \leq \sum_{1 \leq i < j \leq q} \frac{\hat{\ell}_{i,j}}{2^n} \leq \frac{q\sigma + \mu q^2}{2^n} \leq \frac{\sigma^2(\mu+1)}{2^n}. \tag{3.24}$$

Bounding $\mathsf{B.2}$. To bound the probability of $\mathsf{B.2}$, we first fix the value of i, j and α. Note that $Z^j_\alpha = Z^j \oplus \langle \alpha \rangle$. Therefore, $U^i = Z^j_\alpha$ implies $U^i \oplus U^j \oplus V^j = \langle \alpha \rangle$. Now, this essentially implies the following hash equation:

$$\mathsf{Poly}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{M}^j_{\mathbf{R}} \| T^j) \oplus \mathsf{Poly}_{K_h}(\mathbf{C}^j_{\mathbf{R}} \| T^j) = M^i_1 \oplus M^j_1 \oplus C^j_1 \oplus \langle \alpha \rangle. \tag{3.25}$$

Based on the values of i and j, we have the following two subcases:

- **Case A:** If $i \neq j$, then we first assume that $i < j$. Then, if the j-th query is an encryption query, then C^j_1 is random and therefore, by conditioning on the hash key and using the randomness of C^j_1, probability of Equation (3.25) can be bounded by 2^{-n} as C^j_1 is uniformly distributed in the ideal world. Similarly, if the j-th query is a decryption query, then M^j_1 is random and therefore by conditioning on the hash key and using the randomness of M^j_1, the probability of Equation (3.25) can be bounded by 2^{-n} as M_1^j is uniformly distributed in the ideal world. Therefore, by varying over possible choices of i and (j, α), we have

$$\Pr[\mathsf{B.2}] \leq \frac{q\sigma}{2^n}.$$

On the other hand, if $i > j$, then by conditioning all other random variables, we bound the probability of the event using the AXU advantage of the $\mathsf{PolyHash}$ function. Therefore, we have

$$\Pr[\mathsf{B.2}] \leq \sum_{1 \leq i < j \leq q} \frac{\hat{\ell}_{i,j}}{2^n} \leq \frac{q\sigma + \mu q^2}{2^n}.$$

By considering the maximum of the above two, we have

$$\Pr[\mathsf{B.2}] \leq \frac{q\sigma + \mu q^2}{2^n}. \tag{3.26}$$

- **Case B:** If $i = j$, then, Equation (3.25) boils down to the following hash equation:

$$\mathsf{Poly}_{K_h}(\mathbf{C}_{\mathbf{R}}^i \| T^i) = C_1^i \oplus \langle \alpha \rangle. \tag{3.27}$$

Note that for a fixed choice of i and α, Equation (3.27) can be bounded by the AR advantage of the $\mathsf{PolyHash}$ function. Therefore,

$$\Pr[\mathsf{B.2}] = \sum_{i=1}^{q} \sum_{\alpha=1}^{\hat{l}_i} \frac{\hat{l}_i + \mu}{2^n} = \frac{1}{2^n} \sum_{i=1}^{q} \hat{l}_i^2 + \frac{1}{2^n} \sum_{i=1}^{q} \hat{l}_i \mu \leq \frac{\sigma^2 + q^2}{2^n} + \frac{\mu\sigma}{2^n}. \tag{3.28}$$

By considering both the cases and by assuming $q \leq \sigma$, we have

$$\Pr[\mathsf{B.2}] \leq \frac{\sigma^2 + q^2 + \mu\sigma}{2^n} + \frac{q\sigma + \mu q^2}{2^n} \leq \frac{3\sigma^2(\mu + 1)}{2^n}. \tag{3.29}$$

Bounding $\mathsf{B.3}$. To bound the probability of $\mathsf{B.3}$, we first fix the value of i, j, α and

β such that $(i, \alpha) \neq (j, \beta)$. Note that $Z^i_\alpha = Z^j_\beta$ implies the following hash equation:

$$\mathsf{Poly}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{M}^j_{\mathbf{R}} \| T^j) \oplus \mathsf{Poly}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{C}^j_{\mathbf{R}} \| T^j) = W,$$

where $W = M^i_1 \oplus M^j_1 \oplus C^i_1 \oplus C^j_1 \oplus \langle\alpha\rangle \oplus \langle\beta\rangle$. Note that for $i = j$, the probability of this event is zero. For $i \neq j$, without loss of generality, we assume that $i < j$, if the j-th query is an encryption query, then C^j_1 is uniformly distributed in the ideal world which is used to bound the probability of the event by conditioning the hash key and all other random variables. Similarly, if the j-th query is a decryption query, then M^j_1 is uniformly distributed in the ideal world, which is used to bound the probability of the event by conditioning the hash key and all other random variables. Combining the above two arguments with the assumption $q \leq \sigma$ and by varying over all possible choices of indices, we have

$$\Pr[\mathsf{B.3}] = \frac{\binom{\sigma - q}{2}}{2^n} \leq \frac{\sigma^2 + q^2}{2^{n+1}} \leq \frac{\sigma^2}{2^n}. \tag{3.30}$$

Bounding B.4. Bounding this event is equivalent to bounding

$$\mathsf{Poly}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{C}^j_{\mathbf{R}} \| T^j) = C^i_1 \oplus C^j_1.$$

If $\mathbf{C}^i_{\mathbf{R}} \| T^i = \mathbf{C}^j_{\mathbf{R}} \| T^j$ then the probability of this event is zero, otherwise it is bounded by the AXU advantage of the $\mathsf{PolyHash}$ and hence from Equation (3.23) and by the assumption $q \leq \sigma$, we have

$$\Pr[\mathsf{B.4}] \leq \sum_{1 \leq i < j \leq q} \frac{\hat{\ell}_{i,j}}{2^n} \leq \frac{q\sigma + \mu q^2}{2^n} \leq \frac{\sigma^2(\mu + 1)}{2^n}. \tag{3.31}$$

Bounding B.5. We first fix the values of i, j and α and compute the probability of $V^i = M^j_{\alpha+1} \oplus C^j_{\alpha+1} \oplus Z^j_\alpha$. This event boils down to computing the probability of the following event:

$$\mathsf{Poly}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{M}^j_{\mathbf{R}} \| T^j) \oplus \mathsf{Poly}_{K_h}(\mathbf{C}^j_{\mathbf{R}} \| T^j) = W,$$

where $W = C^i_1 \oplus M^j_{\alpha+1} \oplus C^j_{\alpha+1} \oplus M^j_1 \oplus C^j_1 \oplus \langle\alpha\rangle$. Now, we have two subcases as follows:

- **Case A:** if $i = j$, then we have $\mathsf{Poly}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) = C^i_1 \oplus M^i_{\alpha+1} \oplus C^i_{\alpha+1} \oplus M^i_1 \oplus C^i_1 \oplus \langle \alpha \rangle$, which can be bounded using the AR advantage of the $\mathsf{PolyHash}$ function after conditioning all other random variables. Therefore, by assuming $q \leq \sigma$, we have

$$\Pr[\mathsf{B.5}] = \sum_{i=1}^{q} \sum_{\alpha=1}^{\hat{l}_i} \frac{\hat{l}_i + \mu}{2^n} = \frac{1}{2^n} \sum_{i=1}^{q} \hat{l}_i^2 + \frac{1}{2^n} \sum_{i=1}^{q} \hat{l}_i \mu \leq \frac{2\sigma^2}{2^n} + \frac{\mu\sigma}{2^n}. \tag{3.32}$$

- **Case B:** Now we consider the case when $i \neq j$ and without loss of generality we assume that $i < j$. Then, by fixing the hash key K_h, the probability of the above event is the probability over the random draw of C^j_1 (if j-th query is an encryption query) or M^j_1 (if j-th query is a decryption query), which is at most 2^{-n}. Therefore, varying over all the possible choice of i, j and α and $q \leq \sigma$, we have

$$\Pr[\mathsf{B.5}] \leq \frac{q\sigma}{2^n} \leq \frac{\sigma^2}{2^n}. \tag{3.33}$$

Taking the maximum of Equation (3.32) and (3.33), we have

$$\Pr[\mathsf{B.5}] \leq \frac{2\sigma^2}{2^n} + \frac{\mu\sigma}{2^n}. \tag{3.34}$$

Bounding $\mathsf{B.6}$. To bound this event we first fix i, j and α, β and then we compute the probability of $M^i_{\alpha+1} \oplus C^i_{\alpha+1} \oplus Z^i_\alpha = M^j_{\beta+1} \oplus C^j_{\beta+1} \oplus Z^j_\beta$. Now, we have the following subcases based on the values of i and j.

- **Case A:** If $i = j$, then the above event boils down to the following event $M^i_{\alpha+1} \oplus C^i_{\alpha+1} \oplus M^i_{\beta+1} \oplus C^i_{\beta+1} = \langle \alpha \rangle \oplus \langle \beta \rangle$. Since $\alpha \neq \beta$, without loss of generality, we assume that $\alpha < \beta$. Therefore, using the randomness of C^i_β (if i-th query is encryption) or using the randomness of M^i_β (if i-th query is decryption), the probability of the event is bounded by 2^{-n}. By summing over all possible values of i, α and β, we have

$$\Pr[\mathsf{B.6}] \leq \sum_{i=1}^{q} \frac{\binom{\hat{l}_i}{2}}{2^n} \leq \frac{1}{2^{n+1}} (\sum_{i=1}^{q} \hat{l}_i)^2 = \frac{(\sigma - q)^2}{2^{n+1}} \leq \frac{\sigma^2 + q^2}{2^{n+1}}. \tag{3.35}$$

- **Case B:** If $i \neq j$, then we bound the probability of the event similar to that of B.3, that is $1/2^n$ and therefore, by summing over all possible values of i, j, α and β, we have

$$\Pr[\mathsf{B.6}] \leq \frac{\sigma^2 + q^2}{2^{n+1}}. \tag{3.36}$$

By taking the maximum of Equation (7.1) and (3.36) and by assuming $q \leq \sigma$, we have

$$\Pr[\mathsf{B.6}] \leq \frac{\sigma^2 + q^2}{2^{n+1}} \leq \frac{\sigma^2}{2^n}. \tag{3.37}$$

Bounding B.7. Bounding this event is equivalent to bounding $\mathsf{Poly}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) = M^i_1 \oplus x_j$. This event is bounded by the AR advantage of the PolyHash and hence from Equation (3.23) and by assuming $q \leq \sigma$, we have

$$\Pr[\mathsf{B.7}] \leq \sum_{i=1}^{q} \sum_{j=1}^{q_p} \frac{\hat{l}_i + \mu}{2^n} \leq \frac{(\sigma - q) q_p}{2^n} + \frac{\mu q q_p}{2^n} \leq \frac{q_p \sigma (\mu + 1)}{2^n}. \tag{3.38}$$

Bounding B.8. To bound the probability of B.8, we first fix the value of i, j and α. Note that $Z^i_\alpha = x_j$ implies the following hash equation: $\mathsf{Poly}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) = M^i_1 \oplus C^i_1 \oplus \langle \alpha \rangle \oplus x_j$. If the construction query comes after the primitive query, then we can bound the probability of the event using the randomness of C^i_1 (if the construction query is an encryption query) or using the randomness of M^i_1 (if the construction query is a decryption query). Therefore, by conditioning the hash key and all other random variables, the bound will be 2^{-n}. Therefore, we have

$$\Pr[\mathsf{B.8}] = \frac{(\sigma - q) q_p}{2^n} \leq \frac{\sigma q_p}{2^n}.$$

On the other hand, if the primitive query comes after the construction query, then we condition every other random variable and bound the probability of this event by using the AR advantage of the $\mathsf{PolyHash}$ function. Therefore, we have

$$\Pr[\mathsf{B.8}] \leq \sum_{i=1}^{q} \sum_{j=1}^{q_p} \frac{\hat{l}_i + \mu}{2^n} \leq \frac{(\sigma - q)q_p}{2^n} + \frac{\mu q q_p}{2^n} \leq \frac{q_p(\sigma + q\mu)}{2^n}.$$

Therefore, by taking the maximum of the above two and by assuming $q \leq \sigma$, we have

$$\Pr[\mathsf{B.8}] \leq \frac{q_p\sigma(\mu+1)}{2^n}. \tag{3.39}$$

Bounding $\mathsf{B.9}$. Bounding this event is equivalent to bounding $\mathsf{Poly}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) = C^i_1 \oplus y_j$. This event is bounded by the AR advantage of the $\mathsf{PolyHash}$ and hence from Equation (3.23) and by assuming $q \leq \sigma$, we have

$$\Pr[\mathsf{B.9}] \leq \sum_{i=1}^{q} \sum_{j=1}^{q_p} \frac{\hat{l}_i + \mu}{2^n} \leq \frac{(\sigma - q)q_p}{2^n} + \frac{\mu q q_p}{2^n} \leq \frac{q_p\sigma(\mu+1)}{2^n}. \tag{3.40}$$

Bounding $\mathsf{B.10}$. To bound the probability of $\mathsf{B.10}$, we first fix the value of i, j and α. Note that $M^i_{\alpha+1} \oplus C^i_{\alpha+1} \oplus Z^i_\alpha = y_j$ implies the hash equation: $\mathsf{Poly}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) = W$, where $W = M^i_{\alpha+1} \oplus C^i_{\alpha+1} \oplus M^i_1 \oplus C^i_1 \oplus \langle \alpha \rangle \oplus y_j$. Similar to $\mathsf{B.8}$, we bound the event as

$$\Pr[\mathsf{B.10}] \leq \frac{q_p\sigma(\mu+1)}{2^n}. \tag{3.41}$$

The proof follows from Equation (3.22), Equation (7.5)-Equation (4.18) and $q \leq \sigma$. □

3.8.2 Analysis of Good Transcript

In this section, we show that for a good transcript $\tau' = (\tau, \tau_p, k_h)$, realizing τ' is almost as likely in the real world as in the ideal world.

Lemma 3.8.3. *Let $\tau' = (\tau, \tau_p, k_h)$ be a good transcript. Then*

$$\frac{\Pr[\mathsf{T}_{\mathrm{re}} = \tau']}{\Pr[\mathsf{T}_{\mathrm{id}} = \tau']} \geq 1.$$

Proof. Since, in the ideal world, the encryption and the decryption oracle behave perfectly random, we have

$$\Pr[\mathsf{T}_{\mathrm{id}} = \tau'] = \frac{1}{|\mathcal{K}_h|} \frac{1}{\mathbf{P}(2^n, q_p)} \frac{1}{2^{n\sigma}}, \tag{3.42}$$

where σ is the total number of message blocks queried among all q queries.

Real Interpolation Probability. Since τ' is a good transcript, all the inputs and outputs of π are fresh as we have eliminated all the internal input and output collisions of π, including the primitive queries, while defining the bad events. Since there are total $\sigma + q_p$ invocation of π, including the primitive queries, therefore, the required probability is,

$$\Pr[\mathsf{T}_{\mathrm{re}} = \tau'] = \frac{1}{|\mathcal{K}_h|} \frac{1}{\mathbf{P}(2^n, q_p)} \frac{1}{\mathbf{P}(2^n - q_p, \sigma)}. \tag{3.43}$$

By doing a simple algebraic calculation, it is easy to show that the ratio of Equation (3.43) to Equation (3.42) is at least 1. This proves Lemma 3.8.3. □

By combining Lemma 3.8.2, Lemma 3.8.3, Theorem 2.4.1 and Equation (3.21), the result of Theorem 3.7.1 follows. □

Discussion. We would like to note here that a simple birthday bound attack reveals the hash key of the Polyhash function for ppHCTR and ppHCTR+. This would allow an adversary to generate the ciphertext for any plaintext. The same attack also works for HCTR construction. A simple remedy for this problem is to introduce additional permutation calls after the hash evaluation in the upper and bottom layers. This would resolve the problem of revealing the hash difference to any adversary, which in turn makes the recovery of the hash key difficult. A formal security analysis of this modified construction is beyond the scope of this chapter.

4

IpTES: An Inverse-free Tweakable Enciphering Schemes Using Public Permutations

In the previous Chapter, we have proposed tweakable enciphering schemes using public permutations, which use both the forward and inverse directions of the permutations. Most existing public random permutations are more efficient in the forward direction than in the inverse direction. Thus, avoiding the inverse call of the permutations in a construction will make the scheme faster. Therefore, constructing inverse-free TES using public permutations is an interesting and important problem. In this Chapter, we design an inverse-free TES using public permutations called IpTES and also provide adequate arguments in favor of its security.

There are several inverse-free TESs in literature, called FMix [11], AEZ [52] and FAST [19], which are constructed using block-ciphers. IpTES bears structural similarities with block cipher based TESs, most notably with the construction FAST[19], whose main building blocks are a block cipher and an AXU hash function.

4.1 IpTES : A Inverse-Free Single-Keyed TES

We construct IpTES using an n-bit public random permutation π and an AXUAR hash function $H : \mathcal{K}_h \times \{0,1\}^* \to \{0,1\}^n$. IpTES takes as input a arbitrary long message M, a n-bit key K_h and a μ blocks long tweak T. The encryption and decryption algorithm of IpTES is shown in Fig. 4.1.1. To achieve the inverse-free property, we use a two round $\mathsf{Feistel}$ type construction based on π. The $\mathsf{Feistel}$ structure is shown in Fig. 4.1.2.

4.2 Security Proof

Theorem 4.2.1. *Let $\pi \xleftarrow{\$} \mathsf{Perm}(n)$ be an n-bit public random permutation and let $K_h \xleftarrow{\$} \{0,1\}^n$ be n-bit random hash key of the (ϵ, δ)-AXUAR hash function H. Then, for any $(q_e, q_d, q_p, \ell, \sigma)$ information theoretic non-trivial adaptive* CCA *distinguisher* D *against the* tSPRP *security of* $\mathsf{IpTES}[\pi, K_h]$, *we have*

$$\mathbf{Adv}^{\mathrm{tSPRP}}_{\mathsf{IpTES}}(\mathsf{D}) \leq \frac{3q^2\epsilon}{2} + \frac{q}{2^n} + \frac{3q^2}{2^{n+1}} + 2qq_p\delta + \frac{4qq_p}{2^n} + 4qq_p\epsilon + 2\sigma\delta + \frac{4q\sigma}{2^n} + \frac{2\sigma^2}{2^n},$$

where σ is the total number of message blocks for all $q \overset{\Delta}{=} q_e + q_d$ queries.

Proof. When the underlying primitives are understood, for the sake of simplicity, we denote $\mathsf{IpTES}[\pi, H_{K_h}]$ as IpTES. From the Theorem 2.3.1, we have

$$\mathbf{Adv}^{\mathrm{tSPRP}}_{\mathsf{IpTES}}(\mathsf{D}) \leq \mathbf{Adv}^{\pm\mathrm{rnd}}_{\mathsf{IpTES}}(\mathsf{D}) + \frac{q(q-1)}{2^{n+1}}. \tag{4.1}$$

Now, we deal with the ±rnd advantage of IpTES instead of the tSPRP advantage. Consider any information theoretic CCA adversary D, who has oracle access of either $\mathcal{O}_{\mathsf{re}} = (\mathsf{IpTES.Enc}^{\pi}_{K_h}, \mathsf{IpTES.Dec}^{\pi}_{K_h}, \pi^{\pm})$ or $\mathcal{O}_{\mathsf{id}} = (\$_0, \$_1, \pi^{\pm})$, where K_h is a random n-bit hash key, π is a random n-bit permutation and $\$_0, \$_1$ are two independent random function. Also, note that the minimum message length for IpTES is $2n$-bit.

Let, the adversary D make q_e many encryption queries, q_d many decryption queries and q_p many primitive queries. Note that, $q = q_e + q_d$. Suppose that $\tau \overset{\Delta}{=} \{(T^1, M^1, C^1), (T^2, M^2, C^2), \ldots, (T^q, M^q, C^q)\}$ be the queries-response set of the construction and $\tau_p \overset{\Delta}{=} \{(x_1, y_1), (x_2, y_2), \ldots, (x_{q_p}, y_{q_p})\}$ be the queries-response set of

```
$\mathsf{lpTES.Enc}^{\pi}_{K_h}(T, M)$
$(M_1\|\ldots\|M_l) \leftarrow \mathsf{parse}_n(M)$;
$\mathbf{M_R} \leftarrow (M_3\|\ldots\|M_l)$;
$U \leftarrow M_1 \oplus H_{K_h}(T\|\mathbf{M_R})$;
$V \leftarrow M_2 \oplus H_{K_h}(T\|\mathbf{M_R}\|M_1)$;
$(X, Y) \leftarrow \mathsf{Feistel}_{\pi}(U, V)$;
$Z \leftarrow V \oplus X$;
for $j = 1$ to $l - 2$
    $Z_j \leftarrow Z \oplus \langle j \rangle$;
    $S_j \leftarrow \pi(Z_j) \oplus Z_j$;
$\mathbf{S} \overset{\Delta}{=} (S_1\|\ldots\|S_{l-2})$;
$\mathbf{C_R} \leftarrow \mathbf{M_R} \oplus \mathsf{first}(|\mathbf{M_R}|, \mathbf{S})$;
$C_1 \leftarrow Y \oplus H_{K_h}(T\|\mathbf{C_R})$;
$C_2 \leftarrow X \oplus H_{K_h}(T\|\mathbf{C_R}\|C_1)$;
return $(C_1\|C_2\|\mathbf{C_R})$;
```

```
$\mathsf{lpTES.Dec}^{\pi}_{K_h}(T, C)$
$(C_1\|\ldots\|C_l) \leftarrow \mathsf{parse}_n(C)$;
$\mathbf{C_R} \leftarrow (C_3\|\ldots\|C_l)$;
$X \leftarrow C_2 \oplus H_{K_h}(T\|\mathbf{C_R}\|C_1)$;
$Y \leftarrow C_1 \oplus H_{K_h}(T\|\mathbf{C_R})$;
$(U, V) \leftarrow \mathsf{Feistel}^{-1}_{\pi}(X, Y)$;
$Z \leftarrow V \oplus X$;
for $j = 1$ to $l - 2$
    $Z_j \leftarrow Z \oplus \langle j \rangle$;
    $S_j \leftarrow \pi(Z_j) \oplus Z_j$;
$\mathbf{S} \overset{\Delta}{=} (S_1\|\ldots\|S_{l-2})$;
$\mathbf{M_R} \leftarrow \mathbf{C_R} \oplus \mathsf{first}(|\mathbf{C_R}|, \mathbf{S})$;
$M_1 \leftarrow U \oplus H_{K_h}(T\|\mathbf{M_R})$;
$M_2 \leftarrow V \oplus H_{K_h}(T\|\mathbf{M_R}\|M_1)$;
return $(M_1\|M_2\|\mathbf{M_R})$;
```

Figure 4.1.1: lpTES based on an n-bit public random permutation π and an n-bit random hash key k_h. The left part is the encryption algorithm and the right part is its decryption algorithm.

```
$\mathsf{Feistel}_{\pi}(U, V)$
$X \leftarrow U \oplus \pi(V)$;
$Y \leftarrow V \oplus \pi(X)$;
return $(X, Y)$;
```

```
$\mathsf{Feistel}^{-1}_{\pi}(X, Y)$
$V \leftarrow Y \oplus \pi(X)$;
$U \leftarrow X \oplus \pi(V)$;
return $(U, V)$;
```

Figure 4.1.2: Two round Feistel based on an n-bit public random permutation π.

the primitive π. After all the interactions are over, we reveal the hash key is to D. Thus, $\tau' = (\tau, \tau_p, K_h)$ is the query transcript of the attack. Our objective is to find the upper bound on the maximum advantage in distinguishing the real world from the ideal. Now, we define the bad transcripts and upper bound their probabilities in the ideal world.

4.2.1 Definition and Probability of Bad Transcripts

In this section, we define the bad transcripts and bound the probabilities of occurrence of these transcripts in the ideal world. $\mathbf{M^i_R}$ denotes $(M^i_3\|\ldots\|M^i_{l_i})$, $\mathbf{C^i_R}$ denotes $(C^i_3\|\ldots\|C^i_{l_i})$, $U^i = M^i_1 \oplus H_{K_h}(T^i\|\mathbf{M^i_R})$, $V^i = M^i_2 \oplus H_{K_h}(T^i\|\mathbf{M^i_R}\|M^i_1)$, $X^i = C^i_1 \oplus H_{K_h}(T^i\|\mathbf{C^i_R}\|C^i_2)$, $Y^i = C^i_2 \oplus H_{K_h}(T^i\|\mathbf{C^i_R})$ and $Z^i_\alpha = V^i \oplus X^i \oplus \langle\alpha\rangle$.

Definition 4.2.2. (**Bad Transcript for lpTES**) : *If any of the following conditions holds for an attainable transcript $\tau' = (\tau, \tau_p, K_h)$, is called as a **bad** transcript:*

– $\mathsf{B.1}$: $\exists\ i \neq j \in [q]$ *such that,* $V^i = V^j$.

– $\mathsf{B.2}$: $\exists\ i \neq j \in [q]$ *such that,* $X^i = X^j$.

– $\mathsf{B.3}$: $\exists\ i, j \in [q]$ *such that,* $X^i = V^j$.

– $\mathsf{B.4}$: $\exists\ i \neq j \in [q]$ *such that,* $U^i \oplus X^i = U^j \oplus X^j$.

– $\mathsf{B.5}$: $\exists\ i \neq j \in [q]$ *such that,* $V^i \oplus Y^i = V^j \oplus Y^j$.

– $\mathsf{B.6}$: $\exists\ i \neq j \in [q]$ *such that,* $U^i \oplus X^i = V^j \oplus Y^j$.

– $\mathsf{B.7}$: $\exists\ i \in [q]$, $j \in [q_p]$ *such that,* $X^i = x_j$.

– $\mathsf{B.8}$: $\exists\ i \in [q]$, $j \in [q_p]$ *such that,* $V^i = x_j$.

– $\mathsf{B.9}$: $\exists\ i \in [q]$, $j \in [q_p]$ *such that,* $U^i \oplus X^i = y_j$.

– $\mathsf{B.10}$: $\exists\ i \in [q]$, $j \in [q_p]$ *such that,* $V^i \oplus Y^i = y_j$.

– $\mathsf{B.11}$: $\exists\ i, j \in [q]$, $\alpha \in [l_i - 2]$ *such that,* $V^i = Z^j_\alpha$.

– $\mathsf{B.12}$: $\exists\ i, j \in [q]$, $\alpha \in [l_i - 2]$ *such that,* $X^i = Z^j_\alpha$.

– $\mathsf{B.13}$: $\exists\ i, j \in [q]$, $\alpha \in [l_i - 2]$ *such that,* $U^i \oplus X^i = Z^j_\alpha \oplus M^j_{\alpha+2} \oplus C^j_{\alpha+2}$.

– $\mathsf{B.14}$: $\exists\ i, j \in [q]$, $\alpha \in [l_i - 2]$ *such that,* $V^i \oplus Y^i = Z^j_\alpha \oplus M^j_{\alpha+2} \oplus C^j_{\alpha+2}$.

– $\mathsf{B.15}$: $\exists\ i, j \in [q]$, $\alpha \in [l_i - 2]$ *and* $\alpha' \in [l_j - 2]$ *with* $(i, \alpha) \neq (j, \alpha')$ *such that,* $Z^i_\alpha = Z^j_{\alpha'}$.

– B.16 : $\exists\ i, j \in [q]$, $\alpha \in [l_i - 2]$ *and* $\alpha' \in [l_j - 2]$ *with* $(i, \alpha) \neq (j, \alpha')$ *such that,* $Z^i_\alpha \oplus M^i_{\alpha+2} \oplus C^i_{\alpha+2} = Z^j_{\alpha'} \oplus M^j_{\alpha'+2} \oplus C^j_{\alpha'+2}$.

– B.17 : $\exists\ i \in [q]$, $j \in [q_p]$ *and* $\alpha \in [l_i - 2]$ *such that,* $Z^i_\alpha = x_j$.

– B.18 : $\exists\ i \in [q]$, $j \in [q_p]$ *and* $\alpha \in [l_i - 2]$ *such that,* $Z^i_\alpha \oplus M^i_{\alpha+2} \oplus C^i_{\alpha+2} = y_j$.

4.2.2 Analysis of Bad Transcripts:

Bounding B.1. Bounding this event is equivalent to bounding

$$H_{K_h}(T^i \| \mathbf{M_R}^i \| M^i_1) \oplus H_{K_h}(T^j \| \mathbf{M_R}^j \| M^j_1) = M^i_2 \oplus M^j_2.$$

If $T^i \| \mathbf{M_R}^i \| M^i_1 = T^j \| \mathbf{M_R}^j \| M^j_1$ then the probability of this event is zero, otherwise it is bounded by the AXU advantage of the H_{K_h} and hence, we have

$$\Pr[\mathsf{B.1}] \leq \sum_{1 \leq i < j \leq q} \epsilon \leq \frac{q^2 \epsilon}{2}. \tag{4.2}$$

Bounding B.2. Bounding this event is equivalent to bounding

$$H_{K_h}(T^i \| \mathbf{C_R}^i \| C^i_1) \oplus H_{K_h}(T^j \| \mathbf{C_R}^j \| C^j_1) = C^i_2 \oplus C^j_2.$$

If $T^i \| \mathbf{C_R}^i \| C^i_1 = T^j \| \mathbf{C_R}^j \| C^j_1$ then the probability of this event is zero, otherwise it is bounded by the AXU advantage of the H_{K_h} and hence, we have

$$\Pr[\mathsf{B.2}] \leq \sum_{1 \leq i < j \leq q} \epsilon \leq \frac{q^2 \epsilon}{2}. \tag{4.3}$$

Bounding B.3. To bound the probability of B.3, first we fix the values of i and j. Now, this essentially implies the following hash equation:

$$H_{K_h}(T^i \| \mathbf{C_R}^i \| C^i_i) \oplus H_{K_h}(T^j \| \mathbf{M_R}^j \| M^j_1) = C^i_2 \oplus M^j_2. \tag{4.4}$$

–**Case A:** Consider $i = j$. If it is an encryption query, then C^i_2 is uniformly distributed in the ideal world and if it is a decryption query, then M_2^j is uniformly distributed in the ideal world. Therefore, by varying all possible choices of i, we have

$$\Pr[\mathsf{B.3}] \leq q/2^n. \tag{4.5}$$

–**Case B:** Consider the case $i \neq j$. This event is the same as B.2, so,

$$\Pr[\mathsf{B.3}] \leq \frac{q^2\epsilon}{2}. \tag{4.6}$$

Hence, considering both cases, we have

$$\Pr[\mathsf{B.3}] \leq \frac{q^2\epsilon}{2} + \frac{q}{2^n}. \tag{4.7}$$

Bounding B.4. To bound the probability of B.4, first we fix the values of i and j. Now, this essentially implies the following equation:

$$\begin{aligned} H_{K_h}(T^i \| \mathbf{M_R}^i) \oplus H_{K_h}(T^i \| \mathbf{C_R}^i \| C_1^i) \oplus H_{K_h}(T^j \| \mathbf{M_R}^j) \oplus H_{K_h}(T^j \| \mathbf{C_R}^j \| C_1^j) \\ = M_1^i \oplus C_2^i \oplus M_1^j \oplus C_2^j. \end{aligned} \tag{4.8}$$

Without loss of generality, suppose $i < j$. Then, if j-th query is an encryption query, then C_2^j is random. Similarly, if j-th query is a decryption query, then M_1^j is random. So, conditioning on the hash keys and the randomness of M_1^j or C_2^j, the probability of the Equ. (4.8) can be bounded by 2^{-n}. Therefore, by varying over possible choices of i and j, we have

$$\Pr[\mathsf{B.4}] \leq \sum_{1 \leq i < j \leq q} \frac{1}{2^n} \leq \frac{q^2}{2^{n+1}}. \tag{4.9}$$

Bounding B.5. To bound the probability of B.4, first we fix the values of i and j. Now, this essentially implies the following equation:

$$\begin{aligned} H_{K_h}(T^i \| \mathbf{M_R}^i \| M_1^i) \oplus H_{K_h}(T^i \| \mathbf{C_R}^i) \oplus H_{K_h}(T^j \| \mathbf{M_R}^j \| M_1^j) \oplus H_{K_h}(T^j \| \mathbf{C_R}^j) \\ = M_2^i \oplus C_1^i \oplus M_2^j \oplus C_1^j. \end{aligned} \tag{4.10}$$

This event is the same as B.4. Therefore,

$$\Pr[\mathsf{B.5}] \leq \frac{q^2}{2^{n+1}}. \tag{4.11}$$

Bounding B.6. Bounding this event is equivalent to bounding

$$H_{K_h}(T^i \| \mathbf{M_R}^i) \oplus H_{K_h}(T^i \| \mathbf{C_R}^i \| C_1^i) \oplus H_{K_h}(T^j \| \mathbf{M_R}^j \| M_1^j) \oplus H_{K_h}(T^j \| \mathbf{C_R}^j)$$
$$= M_1^i \oplus C_2^i \oplus M_2^j \oplus C_1^j.$$

With the similar argument as B.4, we can say that,

$$\Pr[\mathsf{B.6}] \leq \sum_{1 \leq i < j \leq q} \frac{1}{2^n} \leq \frac{q^2}{2^{n+1}}. \tag{4.12}$$

Bounding B.7. Bounding this events is equivalent to bounding $H_{K_h}(T^i \| \mathbf{C_R}^i \| C_1^i) = C_2^i \oplus x_j$ and it is bounded by the AR advantage of H_{K_h}. Hence,

$$\Pr[\mathsf{B.7}] \leq \sum_{i=1}^{q} \sum_{j=1}^{q_p} \delta \leq q.q_p.\delta. \tag{4.13}$$

Bounding B.8. Bounding this events is equivalent to bounding $H_{K_h}(T^i \| \mathbf{M_R}^i \| M_1^i) = M_2^i \oplus x_j$ and it is bounded by the AR advantage of H_{K_h}. Hence,

$$\Pr[\mathsf{B.8}] \leq \sum_{i=1}^{q} \sum_{j=1}^{q_p} \delta \leq q.q_p.\delta. \tag{4.14}$$

Bounding B.9. To bound the probability of B.9, first we fix the values of i and j. Now, this essentially implies the following event:

$$H_{K_h}(T^i \| \mathbf{M_R}^i) \oplus H_{K_h}(T^i \| \mathbf{C_R}^i \| C_1^i) = M_1^i \oplus C_2^i \ \oplus y_j.$$

If the primitive query comes before the construction query, then the probability of the event is bound by the randomness of C_2^i or M_1^i according to the encryption query or description query, respectively. Therefore, conditioning on all the random variables, the bound of this event will be $1/2^n$. Hence, we have

$$\Pr[\mathsf{B.9}] \leq \frac{qq_p}{2^n}. \tag{4.15}$$

Also, if the primitive query comes after the construction query, then the probability of the event is bound by the AXU advantage of the hash function. Therefore, we have

$$\Pr[\mathsf{B.9}] \leq qq_p\epsilon. \tag{4.16}$$

Hence,

$$\Pr[\mathsf{B.9}] \leq \frac{qq_p}{2^n} + qq_p\epsilon. \tag{4.17}$$

Bounding B.10. To bound the probability of B.10, first we fix the values of i and j. Now, this essentially implies the following event:

$$H_{K_h}(T^i \| \mathbf{M_R}^i \| M_1^i) \oplus H_{K_h}(T^i \| \mathbf{C_R}^i) = M_2^i \oplus C_1^i \ \oplus y_j.$$

Bounding this event is similar to that of B.9. Therefore, we have

$$\Pr[\mathsf{B.10}] \leq \frac{qq_p}{2^n} + qq_p\epsilon. \tag{4.18}$$

Bounding B.11. Bounding this event is equivalent to bounding

$$H_{K_h}(T^i \| \mathbf{M_R}^i \| M_1^i) \oplus H_{K_h}(T^j \| \mathbf{M_R}^j \| M_1^j) \oplus H_{K_h}(T^j \| \mathbf{C_R}^j \| C_1^j) = A,$$

where $A = M_2^i \oplus M_2^j \oplus C_2^j \oplus \langle \alpha \rangle$.

Now we have two subcases as follows:

- **Case A:** If $i = j$, then we have $H_{K_{h_2}}(T^i \| \mathbf{C_R}^i \| C_1^i) = C_2^i \oplus \langle \alpha \rangle$. This can be bounded using the AR advantage of the H_{K_h} after conditioning all other random variables. Therefore,

$$\Pr[\mathsf{B.11}] \leq \sum_{i=1}^{q} \sum_{\alpha=1}^{l_i - 2} \delta \leq \sigma\delta. \tag{4.19}$$

- **Case B:** Now consider that $i \neq j$. We first assume that $i < j$, then the probability of the event is bound by the randomness of C_2^j or M_2^j according to the encryption query or description query respectively. Therefore, conditioning on all the random variables, the bound of this event will be $1/2^n$. Therefore by varying over all possible i and (j, α), we have

  $$\Pr[\mathsf{B.11}] \leq \frac{q\sigma}{2^n}. \tag{4.20}$$

  Now, if $i > j$, then by conditioning all the random variables the probability of the event is bounded by the AXU advantage of the hash function. Therefore, we have

  $$\Pr[\mathsf{B.11}] \leq q\sigma\epsilon. \tag{4.21}$$

  taking the maximum of the above two, we have

  $$\Pr[\mathsf{B.11}] \leq \frac{q\sigma}{2^n}. \tag{4.22}$$

Therefore, from both the cases we have

$$\Pr[\mathsf{B.11}] \leq \sigma\delta + \frac{q\sigma}{2^n}. \tag{4.23}$$

Bounding B.12. To bound the probability of $\mathsf{B.12}$, first we fix the values of i and (j, α). Now, this essentially implies the following event:

$$H_{K_h}(T^i \| \mathbf{C_R}^i \| C_1^i) \oplus H_{K_h}(T^j \| \mathbf{C_R}^j \| C_1^j) \oplus H_{K_h}(T^j \| \mathbf{M_R}^j \| M_1^j) = M_2^j \oplus C_2^i \oplus C_2^j \oplus \langle\alpha\rangle.$$

This is similar as $\mathsf{B.11}$. Therefore,

$$\Pr[\mathsf{B.12}] \leq \sigma\delta + \frac{q\sigma}{2^n}. \tag{4.24}$$

Bounding $\mathsf{B.13}$. Bounding this event is equivalent to bounding

$$H_{K_h}(T^i \| \mathbf{M_R}^i) \oplus H_{K_h}(T^i \| \mathbf{C_R}^i \| C_1^i) \oplus H_{K_h}(T^j \| \mathbf{M_R}^j \| M_1^j) \oplus H_{K_h}(T^j \| \mathbf{C_R}^j \| C_1^j) = W,$$

where $W = M_1^i \oplus M_2^j \oplus C_2^i \oplus C_2^j \oplus M_{\alpha+2}^j \oplus C_{\alpha+2}^j \oplus \langle \alpha \rangle$.

If $i = j$ and it is an encryption query, then in the ideal world, $C_{\alpha+2}^j$ is random. Also, if $i = j$ and it is a decryption query, then in the ideal world $M_{\alpha+2}^j$ is random. So, fixing the hash keys and the randomness of $C_{\alpha+2}^j$ or $M_{\alpha+2}^j$, the probability of this event is $1/2^n$. Therefore in this case,

$$\Pr[\mathsf{B.13}] \leq q\sigma/2^n. \tag{4.25}$$

If $i \neq j$, then without loss of generality, we suppose that $i < j$. Then, by fixing the hash keys, the probability of the above event is the probability over the random draw of C_2^j (if j-th query is an encryption query) or M_2^j (if j-th query is a decryption query), which is at most 2^{-n}. Therefore, in this case,

$$\Pr[\mathsf{B.13}] \leq q\sigma/2^n. \tag{4.26}$$

Therefore, from both the cases, we have

$$\Pr[\mathsf{B.13}] \leq q\sigma/2^n. \tag{4.27}$$

Bounding $\mathsf{B.14}$. Bounding this event is equivalent to bounding

$$H_{K_h}(T^i \| \mathbf{M_R}^i \| M_1^i) \oplus H_{K_h}(T^i \| \mathbf{C_R}^i) \oplus H_{K_h}(T^j \| \mathbf{M_R}^j \| M_1^j) \oplus H_{K_h}(T^j \| \mathbf{C_R}^j \| C_1^j) = W,$$

where $W = M_2^i \oplus C_1^i \oplus M_2^j \oplus C_2^j \oplus M_{\alpha+2}^j \oplus C_{\alpha+2}^j \oplus \langle \alpha \rangle$.

The probability analysis is same as $\mathsf{B.13}$, so

$$\Pr[\mathsf{B.14}] \leq q\sigma/2^n. \tag{4.28}$$

Bounding $\mathsf{B.15}$. To bound the probability of $\mathsf{B.15}$, first we fix the values of i, j, α and α' such that $(i,\alpha) \neq (j,\alpha')$. Now, this essentially implies the following event:

$$H_{K_h}(T^i \| \mathbf{M_R}^i \| M_1^i) \oplus H_{K_h}(T^i \| \mathbf{C_R}^i \| C_1^i) \oplus H_{K_h}(T^j \| \mathbf{M_R}^j \| M_1^j) \oplus H_{K_h}(T^j \| \mathbf{C_R}^j \| C_1^j) = A,$$

where $A = M_2^i \oplus M_2^j \oplus C_2^i \oplus C_2^j \oplus \langle \alpha \rangle \oplus \langle \alpha' \rangle$. For $i = j$ the probability of this event is zero. For $i \neq j$, without loss of generality, let $i < j$, if the j-th query is an encryption query, then C_2^j is uniformly distributed in the ideal world which is used to bound the probability of the event by conditioning the hash key and all other random variables. Similarly, if the j-th query is a decryption query, then M_2^j is uniformly distributed in the ideal world, which is used to bound the probability of the event by conditioning the hash key and all other random variables.
Therefore in this case,

$$\Pr[\mathsf{B.15}] \leq \frac{\sigma^2}{2^n}. \tag{4.29}$$

Bounding $\mathsf{B.16}$. Bounding this event is equivalent to bounding

$$H_{K_h}(T^i \| \mathbf{M_R}^i \| M_1^i) \oplus H_{K_h}(T^i \| \mathbf{C_R}^i \| C_1^i) \oplus H_{K_h}(T^j \| \mathbf{M_R}^j \| M_1^j) \oplus H_{K_h}(T^j \| \mathbf{C_R}^j \| C_1^j) = A,$$

where $A = M_2^i \oplus M_2^j \oplus C_2^i \oplus C_2^j \oplus M_{\alpha+2}^i \oplus C_{\alpha+2}^i \oplus M_{\alpha'+2}^j \oplus C_{\alpha'+2}^j \oplus \langle \alpha \rangle \oplus \langle \alpha' \rangle$.

The probability analysis is same as $\mathsf{B.15}$, so

$$\Pr[\mathsf{B.16}] \leq \frac{\sigma^2}{2^n}. \tag{4.30}$$

Bounding $\mathsf{B.17}$. To bound the probability of $\mathsf{B.17}$, first we fix the values of i and j. Now, this becomes the event

$$H_{K_h}(T^i \| \mathbf{M_R}^i \| M_1^i) \oplus H_{K_h}(T^i \| \mathbf{C_R}^i \| C_1^j) = M_2^i \oplus C_2^i \oplus x_j \oplus \langle \alpha \rangle.$$

If the primitive query comes before the construction query then the probability of the event is bound by the randomness of C_2^i or M_2^i according to the encryption query or description query, respectively. Therefore, conditioning on all the random variables, the bound of this event will be $1/2^n$. Hence, we have

$$\Pr[\mathsf{B.17}] \leq \frac{qq_p}{2^n}. \tag{4.31}$$

Also, if the primitive query comes after the construction query then the probability of the event is bound by the AXU advantage of the hash function. Therefore, we have

$$\Pr[\mathsf{B.17}] \leq qq_p\epsilon. \tag{4.32}$$

Hence,

$$\Pr[\mathsf{B.17}] \leq \frac{qq_p}{2^n} + qq_p\epsilon. \tag{4.33}$$

Bounding $\mathsf{B.18}$. To bound the probability of $\mathsf{B.18}$, first we fix the values of i and j. Now, this becomes the event

$$H_{K_h}(T^i \| \mathbf{M_R}^i \| M_1^i) \oplus H_{K_h}(T^i \| \mathbf{C_R}^i \| C_1^i) = M_2^i \oplus C_2^i \oplus M_{\alpha+2}^i \oplus C_{\alpha+2}^i \oplus y_j \oplus \langle \alpha \rangle.$$

The probability analysis is same as $\mathsf{B.17}$, so

$$\Pr[\mathsf{B.18}] \leq \frac{qq_p}{2^n} + qq_p\epsilon. \tag{4.34}$$

4.2.3 Analysis of Good Transcript

In this section, we show that for a good transcript $\tau' = (\tau, \tau_p, K_h)$, realizing τ' is almost as likely in the real world as in the ideal world.
Let $\tau' = (\tau, \tau_p, K_h)$ be a good transcript. Then

$$\frac{\Pr[\mathsf{T}_{\text{re}} = \tau']}{\Pr[\mathsf{T}_{\text{id}} = \tau']} \geq 1. \tag{4.35}$$

Proof. Since, in the ideal world, the encryption and the decryption oracle behave perfectly random, we have

$$\Pr[\mathsf{T}_{\mathrm{id}} = \tau'] = \frac{1}{|\mathcal{K}_h|}\frac{1}{\mathbf{P}(2^n, q_p)}\frac{1}{2^{n\sigma}}, \tag{4.36}$$

where σ is the total number of message blocks queried among all q queries.

Real Interpolation Probability. Since τ' is a good transcript, all the inputs and outputs of π are fresh. Therefore, the required probability is,

$$\Pr[\mathsf{T}_{\mathrm{re}} = \tau'] = \frac{1}{|\mathcal{K}_h|}\frac{1}{\mathbf{P}(2^n, q_p)}\frac{1}{\mathbf{P}(2^n - q_p, \sigma)}. \tag{4.37}$$

By doing a simple algebraic calculation, it is easy to show that the ratio of Equation (3.11) to Equation (3.10) is at least 1. □

Now, by combining probabilities of the bad transcripts and Equation (4.35), the result of Theorem 4.2.1 follows. □

5 On the Security of TrCBC

CBC-MAC and its variants are widely used and are parts of different standards. It is known that the basic CBC-MAC is not secure as a variable input length MAC; more precisely, CBC-MAC is only secure if the message space is prefix-free [78], i.e., the message space does not contain any two messages where one is a prefix of the other. In case the message space is not prefix-free, a simple length extension attack [4] can be performed to obtain a forgery with probability 1. However, CBC-MAC is optimal in terms of the number of block-cipher calls. If CBC-MAC is instantiated with a block-cipher of block length n, then to authenticate a message of λ bits, it requires a single key (which is the key of the underlying block cipher) and $\lceil \lambda/n \rceil$ block-cipher calls. This makes CBC-MAC a good choice for applications where only fixed-length messages are required to be authenticated.

Over the years, several modifications over the basic CBC-MAC have been proposed to accommodate general message spaces. Some notable constructions in this direction are EMAC [78], XCBC[13], CMAC [42], TMAC [58], OMAC [54], GCBC1 [72], GCBC2 [72]. All these variants require some extra overhead compared to the basic CBC-MAC either in terms of the number of keys used and/or in the number of block-cipher calls required.

TrCBC is a variant of CBC-MAC proposed in [88]. The motivation of TrCBC construction was to provide a CBC-MAC like message authentication code which works for general message spaces but whose overhead in terms of the number of keys and block cipher calls is exactly the same as the CBC-MAC. In [88], it is claimed that TrCBC achieves this with the limitation that it can produce only short tags whose lengths are less than $n/2$-bits, where n is the block length of the underlying block cipher. The main idea of the construction is to truncate the output of CBC-MAC. Truncated CBC-MACs have been analyzed in [45] through a construction called TCBC which is quite different from TrCBC. We discuss more about TCBC later in the chapter.

In this Chapter, we show that TrCBC is insecure. A variant of the length extension attack can be mounted on TrCBC which produces $(n/2-1)$-bit tags with a success probability of $1/4$. The basic attack can be extended to make it work for a large class of messages.

The authors of [88] also claimed TrCBC to be a psudorandom function (PRF) and proved an upper bound for the PRF advantage of an adversary for TrCBC. The bound on the PRF advantage that the authors proved does not suggest TrCBC to be a PRF where the tag length is $(n/2-1)$-bits. We analyze the PRF bound and conclude that the security theorem for TrCBC does not really imply that TrCBC is a secure MAC for all suggested tag lengths.

5.1 CBC-MAC

Consider the map $\mathsf{CBC} : \mathcal{K} \times \mathcal{M} \rightarrow \{0,1\}^n$, where $\mathcal{M} \subseteq \cup_{i>0}\{0,1\}^{ni}$. For $M \in \mathcal{M}$, let $\mathsf{parse}(M) = M_1||M_2||\cdots||M_\ell$ and $C_0 = 0^n$, $C_i = E_K(M_i \oplus C_{i-1})$ for $i \in [\ell]$, where $E_K : \{0,1\}^n \rightarrow \{0,1\}^n$ is a block cipher. We define $\mathsf{CBC}(K,M) = C_\ell$. We often denote $\mathsf{CBC}(K,M)$ by $\mathsf{CBC}_K(M)$. A schematic view of the function $\mathsf{CBC}_K(M)$ is shown in Figure 5.1.1. The function CBC is called the CBC-MAC and it is a secure MAC if the underlying block cipher E is a pseudorandom function and the message space $\mathcal{M}$ is prefix-free, i.e., for any two distinct $x, y \in \mathcal{M}$, x is not a prefix of y. For practical purposes, CBC is used in scenarios where the message space contains strings of fixed length; such message spaces are prefix-free.

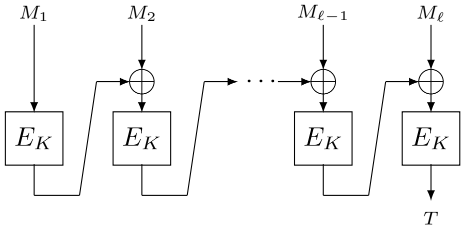

Figure 5.1.1: The function $\mathsf{CBC}_K(M)$. E_K is a block cipher of block size n and $M = M_1||\dots||M_\ell$, where $|M_i| = n$, for $i \in [\ell]$.

Let $X_1, X_2, \dots, X_\ell \in \{0,1\}^n$, then it is easy to see that for any $1 < k < \ell$,

$$\mathsf{CBC}_K\left(X_1\|\cdots\|X_k\|\cdots\|X_\ell\right) = \mathsf{CBC}_K\left(\mathsf{CBC}_K(X_1\|\cdots\|X_k) \oplus X_{k+1}\|X_{k+2}\|\cdots\|X_\ell\right). \tag{5.1}$$

Thus, if $\mathsf{CBC}_K(X_1\|\cdots\|X_k) = T$, then

$$\mathsf{CBC}_K\left(X_1\|\cdots\|X_k\|\cdots\|X_\ell\right) = \mathsf{CBC}_K\left(T \oplus X_{k+1}\|X_{k+2}\|\cdots\|X_\ell\right).$$

This property can be easily translated into a forgery attack: an adversary queries $X_1\|\cdots\|X_k$ and gets response as $T = \mathsf{CBC}_K\left(X_1\|\cdots\|X_k\right)$; further, it queries $T \oplus X_{k+1}\|X_{k+2}\|\cdots\|X_\ell$ and gets the response T_1; and finally it produces $(X_1\|\cdots\|X_k\|\cdots\|X_\ell, T_1)$ as a forgery. From Equation (5.1), it is easy to verify that the forgery will be successful with probability 1. This specific attack is called the *length extension attack* and this cannot be mounted if the message space is prefix-free.

5.2 The Scheme TrCBC

TrCBC instantiated with a block cipher $E : \mathcal{K} \times \{0,1\}^n \rightarrow \{0,1\}^n$ is described in details in Figure 5.2.1. A schematic diagram of the same is shown in Figure 5.2. TrCBC takes a random key $K \xleftarrow{\$} \mathcal{K}$ and a message $M \in \{0,1\}^*$ as input and returns a tag $T \in \{0,1\}^\tau$ of length $\tau < n/2$.

A simplified view of TrCBC in terms of the function CBC would be useful. Let $M \in \{0,1\}^*$, where $|M| = \lambda > 0$. Let $x_1||x_2||\dots||x_\ell = \mathsf{parse}_n(M)$ and let $r = |x_\ell|$.

```
MAC Algorithm: $\mathsf{TrCBC}_K(M)$
Input: $K \xleftarrow{\$} \mathcal{K},\ M \in \{0,1\}^*$.
Output: $T \in \{0,1\}^\tau$, where $\tau < n/2$.

    $M_1\|\cdots\|M_\ell \leftarrow \mathsf{parse_n}(M)$;
    $Y \leftarrow 0^n$;
    for $i \leftarrow 1$ to $\ell - 1$ do
        $X \leftarrow Y \oplus M_i$;
        $Y \leftarrow E_K(X)$;
    end for
    if $|M_\ell| = n$ then
        $X \leftarrow Y \oplus M_\ell$;
        $Y \leftarrow E_K(X)$;
        $T \leftarrow \mathsf{MSB}_\tau(Y)$;
    else
        $X \leftarrow Y \oplus \mathsf{Pad}_n(M_\ell)$;
        $Y \leftarrow E_K(X)$;
        $T \leftarrow \mathsf{LSB}_\tau(Y)$;
    end if
    return $T$.
```

Figure 5.2.1: Specification of TrCBC instantiated with an n-bit block cipher E_K.

Notice that $r = \lambda - (\ell - 1)n = \lambda - (\lceil \lambda/n \rceil - 1)n$. We define TrCBC as

$$\mathsf{TrCBC}_K(M) \quad = \quad \begin{cases} \mathsf{MSB}_\tau\left(\mathsf{CBC}_K(M)\right) & \text{if} \quad r = n, \\ \mathsf{LSB}_\tau\left(\mathsf{CBC}_K(M||10^{n-r-1})\right) & \text{if} \quad r < n, \end{cases} \tag{5.2}$$

where $\mathsf{MSB}_\tau(x)$ and $\mathsf{LSB}_\tau(x)$ return τ many most significant bits and τ many least significant bits of x respectively.

5.3 An Attack on TrCBC

It was claimed in [88] that TrCBC is a secure MAC if the underlying block cipher is a pseudorandom permutation. We show that an adversary making just three queries to the MAC oracle can successfully forge TrCBC with probability 1/4.

We consider TrCBC instantiated with a block cipher of block length n (which is even). We fix the tag length $\tau = n/2 - 1$. Let x_1, x_2, x_3 be fixed but arbitrary strings such that $|x_1| = |x_3| = n$ and $|x_2| = n - 2$. We set $M_1 = x_1$, $M_2 = x_2||10$ and $M_3 = x_3$. The three queries which the adversary asks, along with the responses, are as follows.

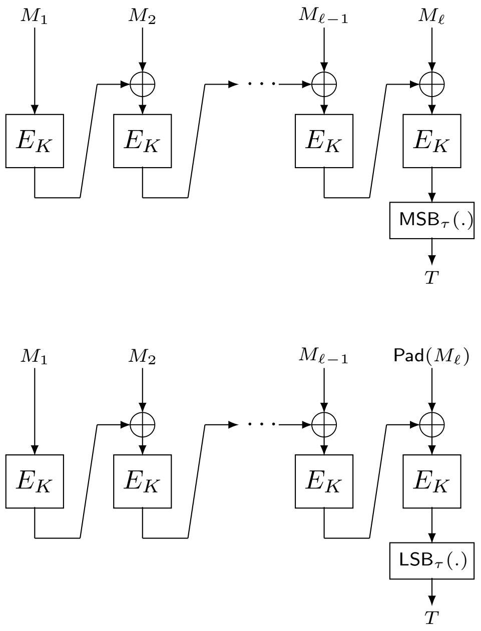

Figure 5.2.2: The TrCBC construction. The first figure is for the full block messages, i.e., the message length is a multiple of the block size n, and the second figure is for messages whose length is not a multiple of n. $\mathsf{Pad}_n(M_\ell) = M_\ell \| 10^{n-|M_\ell|-1}$ and $\tau < n/2$.

1. Query $X^{(1)} = M_1||M_2$, and get T_1 as response.

2. Query $X^{(2)} = M_1||x_2$, and get T_2 as response.

3. Query $X^{(3)} = M_1||M_2||M_3$, and get T_3 as response.

Finally, the adversary submits (M^*, T^*) as the forgery, where

$$M^* = (T_1 \| b_1^* b_2^* \| T_2) \oplus M_3, \quad T^* = T_3 \quad \text{where} \quad b_1^*, b_2^* \xleftarrow{\$} \{0,1\}.$$

Note that (M^*, T^*) is a valid forgery, as M^* which is a single block message has never been queried to the oracle. We are left to show that this forgery is successful with high probability. We claim that for any choice of $K \in \mathcal{K}$,

$$\Pr[\mathsf{TrCBC}_K(M^*) = T^*] = 1/4,$$

where the probability is over the choice of b_1^*, b_2^*. We substantiate our claim below.

From the TrCBC construction and our simplified description of TrCBC in Equation (5.2) we get,

$$\begin{aligned}
T_1 &= \mathsf{TrCBC}_K(M_1||M_2) = \mathsf{MSB}_\tau\left(\mathsf{CBC}_K\left(M_1\|M_2\right)\right)\\
T_2 &= \mathsf{TrCBC}_K(M_1||x_2) = \mathsf{LSB}_\tau\left(\mathsf{CBC}_K\left(M_1\|x_2\|10\right)\right) = \mathsf{LSB}_\tau\left(\mathsf{CBC}_K\left(M_1\|M_2\right)\right)\\
T_3 &= \mathsf{TrCBC}_K(M_1||M_2||M_3) = \mathsf{MSB}_\tau\left(\mathsf{CBC}_K\left(M_1||M_2||M_3\right)\right).
\end{aligned}$$

As $|T_1| = |T_2| = \tau = n/2 - 1$, we have for some $b_1, b_2 \in \{0, 1\}$,

$$T_1\|b_1b_2\|T_2 = \mathsf{CBC}_K(M_1||M_2). \tag{5.3}$$

Hence, using Equation (5.3) and Equation (5.1)

$$\begin{aligned}
\mathsf{MSB}_\tau\left(\mathsf{CBC}_K(T_1\|b_1b_2\|T_2 \oplus M_3)\right) &= \mathsf{MSB}_\tau\left(\mathsf{CBC}_K\left(\mathsf{CBC}_K(M_1||M_2) \oplus M_3\right)\right)\\
&= \mathsf{MSB}_\tau\left(\mathsf{CBC}_K\left(M_1||M_2||M_3\right)\right)\\
&= T_3,
\end{aligned}$$

and

$$\mathsf{TrCBC}_K(M^*) = \mathsf{MSB}_\tau\left(\mathsf{CBC}_K((T_1\|b_1^*b_2^*\|T_2) \oplus M_3)\right).$$

As b_1^*, b_2^* are chosen uniformly at random from $\{0, 1\}$, so with probability $1/4$, we have $(b_1^*, b_2^*) = (b_1, b_2)$, and thus

$$\Pr[\mathsf{TrCBC}_K(M^*) = T_3] = \frac{1}{4},$$

as claimed.

5.4 Discussions

The source of insecurity. The following are the main characteristics of TrCBC:

1. For full block messages, TrCBC is exactly the CBC-MAC scheme, except that

instead of the full output only a part of the output is produced as the tag, in particular $\tau < n/2$ most significant bits only forms the tag.

2. For messages which are not full block, a deterministic padding is applied and the CBC-MAC of the padded message is computed and the least significant τ bits are output as a tag.

The idea behind such a design seems to be separating the outputs for full block and incomplete block messages and the authors thought that a small tag length would prevent a length extension type of attack. But, as only a deterministic padding scheme is applied, hence for any message $M \in \{0,1\}^{mn}$ where the last block is not 0^n almost all bits of $\mathsf{CBC}_K(M)$ can be recovered with just two queries to TrCBC. To see this, let $M = M'||x$, where $|x| = n$ and $x \neq 0^n$. Let $x = a_n \cdots a_2 a_1$, where $a_i \in \{0,1\}$, and j be the smallest element in $[n]$ such that $a_j = 1$. Let $M_1 = M'||a_n a_{n-1} \cdots a_{j-1}$. Then, following the padding scheme in TrCBC we have,

$$\begin{aligned} \mathsf{TrCBC}_K(M) &= \mathsf{MSB}_\tau(\mathsf{CBC}_K(M)), \\ \mathsf{TrCBC}_K(M_1) &= \mathsf{LSB}_\tau(\mathsf{CBC}_K(M)). \end{aligned}$$

Our attack essentially uses the above property of TrCBC to recover 2τ many bits of $\mathsf{CBC}_K(M)$. This property can be further used to forge a large class of messages, which we will describe next.

A generic attack. Consider a message $X = X_1||X_2||\ldots||X_\ell$, for $\ell \geq 2$ and suppose there exists $k \in [\ell-1]$ such that $X_k \neq 0^n$, i.e, $X_1||X_2||\ldots||X_{\ell-1}$ is not the all zero string. Let, $X_k = x\|10^m$, where the first 1 (from the right) in X_k followed by $m \geq 0$ zeros. As before, we fix the tag length $\tau = n/2 - 1$. Let an adversary query with the three queries specified below:

1. $X^{(1)} = X_1||X_2||\ldots||X_k$.

2. $X^{(2)} = X_1||X_2||\ldots\|X_{k-1}||x$.

3. $X^{(3)} = X_1||X_2||\ldots||X_\ell$.

Let the responses to the above three queries be T_1, T_2, T_2 respectively, and let

$$M^* = ((T_1 \| b_1^* b_2^* \| T_2) \oplus X_{k+1}) \| X_{k+2} \| \ldots \| X_\ell,$$

where $b_1^*, b_2^* \xleftarrow{\$} \{0,1\}$. Then following the same arguments as in Section 5.3 it is easy to verify that (M^*, T_3) will be a forgery with a success probability $1/4$.

Provable security of TrCBC. In [88] the authors claim TrCBC to be a PRF. We restate the theorem in [88] next.

Theorem 5.4.1. *Let $R \xleftarrow{\$} \mathsf{Func}(*, \tau)$ and $P \xleftarrow{\$} \mathsf{Perm}(n)$. Let TrCBC_P be the construction where the block cipher in TrCBC is replaced by P. $\mathcal{A}$ is an adversary who asks at most q queries, having an aggregate length of at most σ blocks, then*

$$\left|\Pr\left[\mathcal{A}^{\mathsf{TrCBC}_P(.)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{R(.)} \Rightarrow 1\right]\right| \leq \frac{\sigma(\sigma-1)}{2^{n+1}} + \frac{\sigma(\sigma-1)}{2^{n-2\tau+1}}.$$

Theorem 5.4.1 claims a bound on the PRF advantage of an adversary attacking TrCBC. If we investigate the bound a bit closely, it is clear that the bound on the advantage can be really large for some suggested parameter values of TrCBC. For example, for $\tau = n/2 - 1$, the dominant term in the bound is $\sigma(\sigma-1)/8$, thus for any $\sigma > 3$ the bound becomes meaningless. As a PRF advantage (which is a difference of two probabilities), being less than 1 is a trivial information. Thus, though the theorem is correct, the bound does not guarantee that TrCBC is a PRF for all suggested parameter values and hence, the provable security theorem, though correct, does not imply security of TrCBC for all suggested tag lengths.

The inadequacy of the theorem gets more clear when we see it in light of our attack. For our basic version of the attack, the adversary uses only three queries with query lengths 2 blocks, 2 blocks and 3 blocks respectively. Thus the query complexity (the total aggregate query length) of our adversary is $\sigma = 7$ and it has a large forgery advantage of $1/4$. Based on Theorem 5.4.1 and Equation (2.9), the forgery advantage of an adversary with query complexity 7 would be at most

$$\frac{7(7-1)}{2^{n+1}} + \frac{7(7-1)}{2^3} + \frac{1}{2^{n/2-1}},$$

which is greater than five irrespective of the value of n. Thus, technically our attack does not refute the provable security claim.

It is worth investigating for which tag length(s) the bound implies security of TrCBC. According to Equation (2.9), the forgery advantage of any adversary $\mathcal{A}$ with query complexity σ attacking TrCBC will be upper bounded by

$$\frac{\sigma(\sigma-1)}{2^{n+1}} + \frac{\sigma(\sigma-1)}{2^{n-2\tau+1}} + \frac{1}{2^\tau}.$$

Taking $\tau = n/2 - \alpha$, where $1 \leq \alpha < n/2$ we have the bound as

$$\frac{\sigma(\sigma-1)}{2^{n+1}} + \frac{\sigma(\sigma-1)}{2^{2\alpha+1}} + \frac{1}{2^{n/2-\alpha}} > \frac{1}{2^{2\alpha+1}} + \frac{1}{2^{n/2-\alpha}}. \tag{5.4}$$

A simple computation shows that the expression on the right-hand side of Equation (5.4) attains the minima at $\alpha = n/6$, which suggests that the best-suited value of τ will be $n/2 - n/6 = n/3$.

As suggested by the authors, the allowed value of τ is less than $n/2$. It is common knowledge that shorter tags give lesser security; hence from a user's perspective, the maximum length of a tag, which is supported by the MAC and the application at hand, is chosen. Thus, given the specification of TrCBC it would be alluring for a user to use the largest possible tag length, which is $n/2 - 1$, and as we show, this choice can be disastrous. Thus, the provable security guarantee of TrCBC which the authors provide through Theorem 5.4.1 is very confusing without a proper interpretation of the bound.

Our analysis shows that the PRF bound suggests maximum security when the tag length is $n/3$. If we consider a block cipher with a block length of 128 bits, this translates to tags of length around 42 bits. For most applications, such short tag lengths would not be tolerated. But TrCBC can provide adequate security when instantiated with block ciphers with large block lengths (say 256 bits) and when the tag length is appropriately selected.

The case of TCBC . Security properties of truncated $\mathsf{CBC\text{-}MAC}$ has been studied in details in [45]. In [45] a scheme called TCBC is described as

$$\mathsf{TCBC}_K(X) = \mathsf{MSB}_\tau\left(\mathsf{CBC}_K\left(\mathsf{pad1}(X)\right)\right).$$

Where $\mathsf{pad1}(x)$ appends a 1 followed by sufficiently many zeros to X to make the length of the resulting string a multiple of n. In particular if $x_1||x_2||\ldots||x_\ell = \mathsf{parse}_n(X)$ then

$$\mathsf{pad1}(X) \quad = \quad \begin{cases} X||10^{n-|x_\ell|-1} & \text{if} \quad |x_\ell| < n. \\ X||10^{n-1} & \text{if} \quad |x_\ell| = n. \end{cases}$$

It has been proved in [45] that TCBC is a secure pseudorandom function. In particular, if TCBC is instantiated with a random permutation, then any adversary making q queries with length at most $\lambda < 2^{n/4}$ cannot distinguish TCBC from a random function with probability more than $\epsilon(\lambda, q) = O(\frac{q(q+\lambda)}{2^{n-\tau}} + \frac{\lambda q^2}{2^n})$.

It is important to note the differences between TrCBC and TCBC. The padding scheme of TCBC injectively maps any string in $\{0,1\}^*$ to the set of strings $\cup_{i\geq 1}\{0,1\}^{ni}$, whereas the padding scheme for TrCBC is not injective. Also, for any message it is not possible for an adversary to know more than τ bits of the final output of TCBC, but as we already showed for TrCBC it is possible to know 2τ many bits of the output for a large class of messages and this helps in the forgery attack. Finally, TCBC requires one more block cipher call than TrCBC for full block messages.

6 Variable Output Length Message Authentication Codes

A message authentication code (MAC) generally produces fixed length authentication tags. An important question addressed in [47] is the following. Is it possible to construct MACs which can produce tags of variable lengths and the users have the liberty to choose the tag lengths? The authors of [47] points out a discussion in the Crypto Forum Research Group which voices concerns regarding the suggested use of tags of different lengths in UMAC [12]. As a part of this discussion Wagner [84] warns that such uses may lead to unexpected vulnerabilities and it is better to use just a single tag length with a specific key. As claimed in [47] we also could not find any work which addresses this issue of variable tag length MACs.

In the context of authenticated encryption (AE) there have been some discussions on the variable tag length issue. In [79], a formal treatment of variable tag length AEs was done and there it was pointed out that it is desirable that an AE scheme securely generates variable length authentication tags as this protects it against abuse and makes them more usable in lightweight scenarios. These reasons are valid for MACs also and it is important to have MACs which can securely generate tags of different tag lengths.

A formal study of message authentication codes that produce variable length tags was first presented by Sarkar and Ghosh [47]. The study in [47] encompasses nonce based Wegman-Carter(WC) type MACs. They show concrete attacks on WC type

MACs when used to generate variable length tags and also construct secure WC type nonce based MACs and prove their security.

We consider deterministic MACs (MACs which does not use nonces or states) based on PRFs and thus extend the work in [47]. We show that variable output length pseudorandom functions (vlPRF) can be used as MACs with variable tag lengths. In a variable output length PRF, the output length can be specified by the user. We specify the syntax and security of vlPRF and propose two generic constructions. The first construction uses a fixed input and fixed output length PRF and an AXU hash function. The other one converts a variable input length but fixed output length PRF into a vlPRF. The later construction can be generically used to convert a secure deterministic MAC into a variable output length MAC with very little overhead. In particular, for widely used block cipher based MACs like (secure variants of) CBC-MAC, PMAC, etc., the extra overhead would be just one block-cipher call. Finally, we explore if a given deterministic MAC can be converted into a variable output length MAC with no extra overhead. In this regard, we propose vlPMAC which is a modification of PMAC [14] to enable it to produce tags of variable lengths.

6.1 Variable Output-Length PRF (vlPRF)

Let, $\mathcal{K} \subset \{0,1\}^*$ and $L \subset \mathbb{N}$ be finite sets, and $F : \mathcal{K} \times \{0,1\}^{\leq \ell} \times L \to \cup_{i \in L}\{0,1\}^i$ be a function family where for every $k \in \mathcal{K}$, $x \in \{0,1\}^{\leq \ell}$ and $\tau \in L$, $|F(k,x,\tau)| = \tau$. As is customary, we will further denote $F(k,x,\tau)$ by $F_k(x,\tau)$, and consider F to be a family of functions $F = \{F_k\}_{k \in \mathcal{K}}$.

We define the vlPRF-advantage of an adversary $\mathcal{A}$ in distinguishing the family F as

$$\mathsf{Adv}_F^{\mathsf{vlPRF}}(\mathcal{A}) = \left| \Pr[k \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{F_k(\cdot,\cdot)} \Rightarrow 1] - \Pr[\mathcal{A}^{\$(\cdot,\cdot)} \Rightarrow 1] \right|,$$

where $\$(x,i)$ outputs a uniform random element in $\{0,1\}^i$ for each distinct input (x,i). We call the function family $F = \{F_k\}_{k \in \mathcal{K}}$ a vlPRF if $\mathsf{Adv}_F^{\mathsf{vlPRF}}(\mathcal{A})$ is small for all the efficient adversaries $\mathcal{A}$.

The above definition is a slight variant of the definition in [63]. Unlike a (fixed output length) PRF, a vlPRF in addition to its normal input, takes in an extra positive integer $\tau \in L$, which specifies the output length of the function F. For distinct $(x, \tau) \in \{0,1\}^{\leq \ell} \times L$, the output of $F(x, \tau)$ is indistinguishable from an uniform random element in $\{0,1\}^{\tau}$.

6.2 Variable Output-Length MAC (vlMAC)

A variable output length MAC (vlMAC) is a MAC that can produce variable output length tags. As a traditional MAC, a vlMAC is a pair of algorithms: the tag generation algorithm and the verification algorithm. Both algorithms depend on a tag generation function $F : \mathcal{K} \times \mathcal{M} \times L \to \mathcal{T}$, where $\mathcal{K}$ is the key space, $\mathcal{M}$ is the message space, $L \subset \mathbb{N}$ is the set of allowed tag lengths and $\mathcal{T}$ the tag space. For any $x \in \mathcal{M}$, and $\tau \in L$, $|F_k(x, \tau)| = \tau$. The tag generation algorithm receives as input a message $x \in \mathcal{M}$ and a desired tag length $\tau \in L$ and computes $t = F_k(x, \tau)$ and finally outputs (x, t). The verification algorithm on receiving a message tag pair (x, t), computes $t' = F_k(x, |t|)$ and outputs true if $t' = t$ and false otherwise. We generally specify the MAC by the tag generation function $F_k(.,.)$.

The security of a variable length MAC F_k against an adversary $\mathcal{A}$ is described as follows. $\mathcal{A}$ has oracle access to the tag generation algorithm. $\mathcal{A}$ makes q many tag generation queries $(x^1, \tau^1), (x^2, \tau^2), \ldots (x^q, \tau^q)$ and gets the corresponding responses $t^1, t^2, \ldots, t^q$. These queries are made adaptively, and for each i, $|t^i| = \tau^i$. Finally $\mathcal{A}$ outputs $(\tilde{x}, \tilde{t}) \in \mathcal{M} \times \mathcal{T}$, such that $(\tilde{x}, |\tilde{t}|) \neq (x^i, \tau^i)$, for any $i \in [q]$. We say that successfully forges F for tag length τ if $F_k(\tilde{x}, |\tilde{t}|) = \tilde{t}$. Let $\mathsf{Succ}_{\mathcal{A}}[\tau]$ be the event that $\mathcal{A}$ successfully forges for tag length τ, then

$$\mathsf{Adv}_F^{\mathrm{auth}}[\tau](\mathcal{A}) = \Pr[\mathsf{Succ}_{\mathcal{A}}[\tau]]. \tag{6.1}$$

Tag Truncation is Not Secure: Consider a secure MAC, with tag generation function $F_K : \mathcal{K} \times \mathcal{M} \to \{0,1\}^n$. Suppose that this MAC is used to generate variable length tags of length at most n, i.e. when a tag of length τ is required for a message

$x \in \mathcal{M}$, the tag generation algorithm outputs $\mathsf{MSB}_\tau(F_K(x))$.

This is not secure according to the above definition. Consider an adversary who queries a message x and seeks a $n-1$ bit tag and gets t as a response. So, $|t| = n-1$. After this single tag generation query $\mathcal{A}$ presents its forgery attempt as $(x, t\|0)$. Note, this is a valid forgery attempt as it has never asked a tag generation query for a n bit tag. It is easy to see that $\mathcal{A}$ is successful in its forgery with probability $1/2$.

The next Theorem says that if F is a vlPRF, then F is also a secure tag generating function for a variable output length MAC.

Theorem 6.2.1. *Let $F : \mathcal{K} \times L \to \cup_{i\in L}\{0,1\}^i$ be such that $|F_k(x,\tau)|$ for all $k \in \mathcal{K}$, $x \in \mathcal{M}$ and $\tau \in L$. If $\mathcal{A}$ be a vlMAC adversary for F, then there exists a vlPRF adversary $\mathcal{B}$ of F.*

In particular, for every vlMAC adversary $\mathcal{A}$ that attacks F, who makes at most q_g many tag generation queries, there exists a vlPRF adversary $\mathcal{B}$ that attacks F, who makes at most $q = q_g + 1$ many queries, such that

$$\mathsf{Adv}_F^{\mathsf{auth}}[\tau](\mathcal{A}) \leq \mathsf{Adv}_F^{\mathsf{vlPRF}}(\mathcal{B}) + \frac{1}{2^\tau},$$

where τ is the tag length for the forgery attempt of $\mathcal{A}$.

Proof. Let $\mathcal{A}$ be an arbitrary vlMAC adversary for F. We construct a vlPRF adversary $\mathcal{B}$ with oracle $\mathcal{O}$ that runs adversary $\mathcal{A}$ as follows:

On receiving a query (x,τ) from $\mathcal{A}$, $\mathcal{B}$ returns $t = \mathcal{O}(x,\tau)$ to $\mathcal{A}$ and continues until $\mathcal{A}$ stops querying. Finally, when $\mathcal{A}$ outputs a forgery $(\tilde{x}, \tilde{t})$, if $\mathcal{O}(\tilde{x}, |\tilde{t}|) = \tilde{t}$ the $\mathcal{B}$ outputs a 1; otherwise, it outputs a zero.

In the real world, the oracle $\mathcal{O}$ is $F(\cdot,\cdot)$ and $\mathcal{B}$ outputs a 1 if $\mathcal{A}$ successfully forges. Thus, we have

$$\Pr[\mathcal{B}^{F(\cdot,\cdot)} \Rightarrow 1] = \mathsf{Adv}_F^{\mathsf{auth}}[\tau](\mathcal{A}). \tag{6.2}$$

In the ideal world $\mathcal{B}$'s oracle is $\$(\cdot,\cdot)$ and thus in the ideal world $\mathcal{O}(\tilde{x}, |\tilde{t}|)$ will be a uniform random string in $\{0,1\}^{|\tilde{t}|}$. The the probability that $\mathcal{O}(\tilde{x}, |\tilde{t}|)$ would be equal to $\mathcal{A}$'s forged tag $\tilde{t}$ would be at most $1/2^{|\tilde{t}|}$. If the tag length of $\mathcal{A}$'s forgery is τ, i.e if $|\tilde{t}| = \tau$, we have

$$\Pr[\mathcal{B}^{\$(\cdot,\cdot)} \Rightarrow 1] \leq \frac{1}{2^{\tau}}. \tag{6.3}$$

Thus using Equations. (6.2),(6.3) and the vlPRF advantage of $\mathcal{B}$, we have

$$\mathsf{Adv}_F^{\mathsf{auth}}[\tau](\mathcal{A}) \leq \mathsf{Adv}_F^{\mathsf{vlPRF}}(\mathcal{B}) + \frac{1}{2^{\tau}}.$$

□

6.3 Constructing vlPRF from Fixed Input Length and Fixed Output Length PRF

Let $F : \mathcal{K} \times \{0,1\}^r \rightarrow \{0,1\}^n$ be a PRF family with fixed input length r, and fixed output length n. Let $H : \mathcal{K}' \times \{0,1\}^{\leq \ell} \rightarrow \{0,1\}^n$ be a universal hash family. Let $L \subset [2^n - 1]$. We construct a function family $F' : (\mathcal{K} \times \mathcal{K}') \times \{0,1\}^{\leq \ell - n} \times L \rightarrow \cup_{i \in L}\{0,1\}^i$ as shown in the algorithm in Figure 6.3.1.

```
$F'_{K,h}(x, \tau)$
    $z \leftarrow H_h(x \| \langle \tau \rangle_n)$;
    $m \leftarrow \lceil \frac{\tau}{n} \rceil$;
    for $i \leftarrow 1$ to $m$,
        $C_i \leftarrow F_K(z \oplus \langle i \rangle_n)$;
    end for
    $C \leftarrow C_1 || C_2 || \cdots || C_m$;
    return $\mathsf{MSB}_\tau(C)$
```

Figure 6.3.1: Constructing a vlPRF F' from a PRF F. And, $(h, K) \in \mathcal{K}' \times \mathcal{K}$, $(x, \tau) \in \mathcal{M} \times L$

Theorem 6.3.1. *Let $F : \mathcal{K} \times \{0,1\}^r \to \{0,1\}^n$ be a* PRF *and $H : \mathcal{K}' \times \{0,1\}^{\leq \ell} \to \{0,1\}^n$ be ϵ-AXU and F' be as defined in Figure 6.3.1. Let $\mathcal{A}$ be an arbitrary* vlPRF *adversary attacking F', and $\mathcal{A}$ asks q queries $(x^1, \tau^1), (x^2, \tau^2), \ldots .(x^q, \tau^q)$ to its oracle. Let $\tau_{\max} = \max\{\tau^1, \tau^2, \ldots, \tau^q\}$. Then there exists a* PRF *adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}_{F'}^{\mathsf{vlPRF}}(\mathcal{A}) \leq \mathsf{Adv}_{F}^{\mathsf{PRF}}(\mathcal{B}) + \epsilon q^2 \left\lceil \frac{\tau_{max}}{n} \right\rceil .$$

Moreover, if $\mathcal{A}$ runs for time t then $\mathcal{B}$ runs for time $O(t)$ and asks at most $q\lceil \tau_{max}/n \rceil$ queries to its oracle.

Proof. First notice that for any $(x, \tau) \in \mathcal{M} \times L$ and any $(K, h) \in \mathcal{K} \times \mathcal{K}'$, $|F'_{K,h}(x, \tau)| = \tau$, a required property of vlPRF.

If we replace F_K used in the construction of F' with a random function $\rho()$, we call the resulting algorithm as $F'[\rho]$.

Let $\mathcal{A}$ be an arbitrary vlPRF adversary for F', who makes q distinct queries to its oracle. We construct a PRF adversary $\mathcal{B}$ attacking F. Consider $\mathcal{B}$'s oracle to be $\mathcal{O}$, which is either the function $F_K(.)$ for an uniform random K, or a random function ρ from $\mathsf{Func}(r, n)$. $\mathcal{B}$ selects a h uniformly at random from $\mathcal{K}'$ and runs the adversary $\mathcal{A}$ as follows. On receiving a query (x^i, τ^i) from $\mathcal{A}$, it computes $m^i = \lceil \tau^i / n \rceil$ and $z^i = H_h(x^i \| \langle \tau^i \rangle)$ and sends

$$\mathsf{MSB}_{\tau^i}\left(\mathcal{O}\left(z^i \oplus \langle 1 \rangle_n\right) \| \ldots \| \mathcal{O}\left(z^i \oplus \langle m^i \rangle_n\right)\right)$$

to $\mathcal{A}$. $\mathcal{B}$ continues the above procedure as long as $\mathcal{A}$ queries, and finally $\mathcal{A}$ outputs $b \in \{0,1\}$ and $\mathcal{B}$ also outputs b. Thus, if $\mathcal{B}$'s oracle is F_K then for a query (x^i, τ^i), $\mathcal{A}$ receives as response $F'_{K,h}(x^i, \tau^i)$, and if $\mathcal{B}$'s oracle is a random function ρ, then $\mathcal{A}$ receives as response $F'[\rho]$. Thus, we have

$$\Pr[K \xleftarrow{\$} \mathcal{K} : \mathcal{B}^{F_K()} \Rightarrow 1] = \Pr[K \xleftarrow{\$} \mathcal{K}, h \xleftarrow{\$} \mathcal{K}' : \mathcal{A}^{F'_{K,h}(.,.)} \Rightarrow 1] \qquad (6.4)$$

$$\Pr[\rho \xleftarrow{\$} \mathsf{Func}(r, n) : \mathcal{B}^{F_K()} \Rightarrow 1] = \Pr[\mathcal{A}^{F'[\rho](.,.) \Rightarrow 1}]. \qquad (6.5)$$

Consider a procedure $\$(.,.)$ which when queried with (x^i, τ^i) returns a uniform random element from $\{0,1\}^{\tau^i}$. The procedures $\$(.,.)$ and $F'[\rho]$ are indistinguishable to $\mathcal{A}$ unless there is a collision in the set Dom defined as

$$\mathsf{Dom} = \bigcup_{i \in [q]} S_i,$$

where

$$S_i = \left\{ z^i \oplus \langle 1 \rangle_n, z^i \oplus \langle 2 \rangle_n, \ldots, z^i \oplus \langle m^i \rangle_n \right\}.$$

Let COLL be the event that there is a collision in the set Dom, then we have

$$\Pr[\mathcal{A}^{F'[\rho](.,.)}] - \Pr[\mathcal{A}^{\$(.,.)}] = \Pr[\mathsf{COLL}]. \tag{6.6}$$

Using Equations (6.4),(6.5),(6.6) and the definitions of PRF advantage of $\mathcal{B}$ and vlPRF advantage of $\mathcal{A}$, we have

$$\mathsf{Adv}_{F'}^{\mathsf{vlprf}}(\mathcal{A}) \leq \mathsf{Adv}_{F}^{\mathrm{prf}}(\mathcal{B}) + \Pr[\mathsf{COLL}]. \tag{6.7}$$

Finally, we are left with bounding $\Pr[\mathsf{COLL}]$. We consider the event $z^i \oplus \langle j \rangle_n = z^{i'} \oplus \langle j' \rangle_n$, where $(i,j) \neq (i',j')$. We have several cases to consider:

Case 1: $i = i'$. In this case $j \neq j'$ and thus

$$\Pr[z^i \oplus \langle j \rangle_n = z^{i'} \oplus \langle j' \rangle_n] = 0$$

Case 2: $i \neq i'$. Here we have $z^i = H_h(x^i \| \langle \tau^i \rangle_n)$ and $z^{i'} = H_h(x^{i'} || \langle \tau^{i'} \rangle_n)$. As $i \neq i'$ and all queries of $\mathcal{A}$ are distinct, hence we have $(x^i, \langle \tau^i \rangle_n) \neq (x^{i'}, \langle \tau^{i'} \rangle_n)$ and thus $x^i \| \langle \tau^i \rangle_n \neq x^{i'} \| \langle \tau^{i'} \rangle_n$. Hence,

$$\begin{aligned} & \Pr\left[z^i \oplus \langle j \rangle_n = z^{i'} \oplus \langle j' \rangle_n\right] \\ = \; & \Pr\left[H_h\left(x^i \| \langle \tau^i \rangle_n\right) \oplus H_h\left(x^{i'} \| \langle \tau^{i'} \rangle_n\right) = \langle j \rangle_n \oplus \langle j' \rangle_n\right] \\ \leq \; & \epsilon \end{aligned} \tag{6.8}$$

The last equation is true as $x^i \| \langle \tau^i \rangle_n \neq x^{i'} \| \langle \tau^{i'} \rangle_n$ and H is a ϵ-AXU.

To complete our calculations, we need to find how many distinct equations of the type

$$z^i \oplus z^{i'} = \langle j \rangle_n \oplus \langle j' \rangle_n \tag{6.9}$$

are possible where $(i, j) \neq (i', j')$ and $i \neq i'$. When $i \neq i'$, there are $\binom{q}{2}$ distinct pairs of $(z^i, z^{i'})$.

Let $m_{\max} = \max\{m^1, m^2, \ldots, m^q\}$ and $S = \{\langle 1 \rangle_n, \langle 2 \rangle_n, \ldots \langle m_{\max} \rangle_n\}$. If $S \oplus S = \{p \oplus q : p, q \in S\}$, then $|S \oplus S| < m_{\max}$. Thus, the number of distinct values of $\langle j \rangle_n \oplus \langle j' \rangle_n$, $j, j' \in [m_{\max}]$ is at most $m_{\max}$. Thus, the total number of distinct equations of the form in Equation (6.9) is at most $q^2 m_{\max}$. As $m_{\max} = \lceil \frac{\tau_{max}}{n} \rceil$, hence by using Equation (6.6) and the union bound, we have

$$\Pr[\mathsf{COLLD}] \leq \epsilon q^2 \left\lceil \frac{\tau_{max}}{n} \right\rceil,$$

as desired. □

We observe some properties of the construction of F' below:

1. Construction of F' is generic; any ϵ-AXU hash function and PRF can be plugged in place of H and F, respectively. In particular, a polynomial hash can be used in place of H and a block cipher can be used in place of F.

2. The number of calls to the PFR F depends on the value of τ. Thus, for bigger output lengths, more calls to F are required. The calls to F can be parallelized, and thus F' can be efficiently implemented in both software and hardware.

3. The construction bears similarity with the Wegman-Carter paradigm of constructing PRFs. A very similar construction as F' has been reported in [47] in the context of deterministic Wegman-Carter type variable tag length MACs. But the construction in [47] does not consider output lengths greater than n.

6.4 Constructing vlPRF from Variable Input Length and Fixed Output Length PRF

In this section, we construct a vlPRF G' from a variable input length and fixed output length PRF G where the output length of G is fixed to n. We consider the case where the output length of G' is at most n. Such a construction would have applications in converting an ordinary deterministic MAC to a variable output length MAC. This application is discussed in detail in Section 6.5.

Consider a variable input length and fixed output length PRF family $G : \mathcal{K} \times \{0,1\}^{\leq \ell} \rightarrow \{0,1\}^n$. We construct a vlPRF $G' : \mathcal{K} \times \mathcal{M} \times L \rightarrow \cup_{i \in L}\{0,1\}^i$, where $L \subseteq [n]$ and $\mathcal{M} = \{0,1\}^{\leq \ell - n}$. The construction is:

$$G'_K(x, \tau) = \mathsf{MSB}_\tau(G_K(x \| \langle \tau \rangle_n)). \tag{6.10}$$

Note, here $\tau \leq n$.

Theorem 6.4.1. *If $G : \mathcal{K} \times \{0,1\}^{\leq \ell} \rightarrow 0, 1^n$ is a PRF, then G' as described in Equation (6.10) is a PRF.*

Proof. The only thing to note here is that if $(x, \tau) \neq (x', \tau')$, then $x \| \langle \tau \rangle_n \neq x' \| \langle \tau' \rangle_n$. Thus, for distinct inputs to G', G is also called on distinct inputs, which ensures that G' is also a PRF. □

6.5 Variable Length MACs Using vlPRF

As is evident from Theorem 6.2.1 a vlPRF can be used as a tag generation function for a vlMAC. With the constructions discussed in Sections 6.3 and 6.4, we have two different possibilities which we discuss next.

1. The construction in Section 6.3 allows us to construct a vlPRF of arbitrary output lengths using a fixed input-output length PRF. Thus, this construction can be instantiated with a block cipher and any AXU hash function to be an

efficient MAC. This MAC resembles a deterministic Wegman-Carter type MAC with the additional functionality of variable tag lengths. It is to be observed that irrespective of the tag length, the security of the MAC would be $O(q^2\ell)/2^n$ where ℓ is the maximum length of a message. Thus, the security of the MAC would depend on the block length of the block cipher, and larger tags will not provide more security if the tag lengths exceed n. Thus, using this construction for generating large tags (larger than n) will not have any security advantage. Though there may be applications where tag lengths greater than n may be required for functionality, not for enhanced security and this construction may be useful in such scenarios.

2. The construction in Section 6.4 converts a variable input length and fixed output length (say n) PRF into a vlPRF whose output length can be at most n. This construction can be useful in converting an existing fixed output length MAC to a variable output length MAC. This can only produce tags of length up to n, and this functionality would be of practical interest to accommodate the MAC in applications where only short tags can be tolerated, say in case of a lightweight scenario. Most block cipher based MACs, say variants of CBC MAC, PMAC etc are variable input length but fixed input length PRFs. These MACs can be easily converted into variable tag length MACs using the construction in Equation (6.10). This would incur a minimal extra overhead, like in the case of PMAC, OMAC etc., this would lead to only one extra block cipher call.

6.6 vlPMAC: Variable Output-Length Variant of PMAC

As discussed in Section 6.5, PMAC can be easily converted into a variable output length MAC just by using the construction in Equation (6.10) with an extra overhead of one block-cipher call. Here, we propose a variable output length variant of PMAC [14] called vlPMAC where the number of block-cipher calls is the same as PMAC.

```
**MAC Algorithm:** $\mathsf{vlPMAC}_K(M, \tau)$
**Input:** $K \xleftarrow{\$} \mathcal{K},\ M \in \{0,1\}^*, \tau \in \mathcal{T}$.
**Output:** $T \in \{0,1\}^\tau$, where $\tau < n$.

01. $M_1 \| \cdots \| M_\ell \leftarrow \mathsf{parse_n}(M)$;
02. $L \leftarrow E_K(\tau)$;
03. if $\ell = 1$ then
04.     if $|M_\ell| = n$ then
05.         $X_\ell \leftarrow M_\ell \oplus \gamma^{-1} \cdot L$;
06.     else
07.         $X_\ell \leftarrow M_\ell \oplus \gamma^{-2} \cdot L$;
08. else
09.     for $i \leftarrow 1$ to $\ell - 1$ do
10.         $X_i \leftarrow M_i \oplus \gamma^i \cdot L$;
11.         $Y_i \leftarrow E_K(X_i)$;
12.     end for
13.     $\Sigma \leftarrow Y_1 \oplus Y_2 \oplus \cdots \oplus Y_{\ell-1} \oplus \mathsf{Pad}_n(M_\ell)$;
14.     if $|M_\ell| = n$ then
15.         $X_\ell = \Sigma \oplus L \cdot \gamma^{-1}$;
16.     else
17.         $X_\ell = \Sigma$;
18. $T \leftarrow \mathsf{MSB}_\tau(E_K(X_\ell))$;
19. return $T$.
```

Figure 6.5.1: Specification of vlPMAC construction based on an n-bit block cipher E_K, where γ is the root of a primitive polynomial in $GF(2^n)$. $\mathsf{Pad_n(M_\ell) = M_\ell \| 10^{n-|M_\ell|-1}}$.

The underlying primitive is a block cipher $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$, considered secure as a PRP. We denote $E(K, \cdot)$ as $E_K(\cdot)$ and each $E_K(\cdot)$ is a permutation on $\{0,1\}^n$. vlPMAC takes a randomly sampled key K, a message M of arbitrary length and the desired length of the tag $\tau \le n$ as input and returns a τ-bit tag. Let us denote $\mathsf{vlPMAC}(K, M, \tau)$ as $\mathsf{vPlMAC}_K(M, \tau)$. The algorithm of vlPMAC is described in Figure 6.5.1. In the algorithm γ is a root of a fixed primitive polynomial of degree n over $GF(2^n)$.

vlPMAC is a small modification of the original PMAC algorithm [14]. The differences are described below:

1. In line 02 of Figure 6.5.1, L is computed by encrypting the desired tag length τ

using the block cipher. whereas in the original PMAC construction L is set to $E_K(0)$. Setting $L = E_K(\tau)$ makes the output tag dependent on the tag length τ. Hence, tags of two different lengths of the same message generated by vlPMAC will be different.

2. Messages of length less or equal to n are treated differently than in the original construction. This handling of messages of single blocks have similarity with the construction PAuth reported in [27].

The next theorem asserts that vlPMAC is a vlPRF and thus a secure variable output length MAC.

Theorem 6.6.1. *Let* $\mathsf{vlPMAC}[\mathsf{Perm}(n)]$ *be the* vlPMAC *construction in Figure 6.5.1 where the block-cipher* $E_K(\cdot)$ *is replaced by a permutation* π *drawn uniformly from* $\mathsf{Perm}(n)$*. Let* $\mathcal{A}$ *be any adversary which attacks* $\mathsf{vlPMAC}[\mathsf{Perm}(n)]$ *and asks at most* q *queries which consists of a total of* σ *many* n*-bit blocks. Then,*

$$\mathsf{Adv}^{\mathsf{vlPRF}}_{\mathsf{vlPMAC}[\mathsf{Perm}(n)]}(\mathcal{A}) \leq \frac{7\sigma^2}{2^n} \tag{6.11}$$

The next section is devoted to the proof of Theorem 6.6.1.

6.7 Proof of Theorem 6.6.1

The proof bears similarity with the original security proof of PMAC in [14].

As a first step, we replace the uniform random permutation in $\mathsf{vlPRF}_{\mathsf{vlPMAC}[\mathsf{Perm}(n)]}$ with a function chosen uniformly at random from $\mathsf{Func}(n, n)$, and we call the resulting scheme as $\mathsf{vlPRF}_{\mathsf{vlPMAC}[\mathsf{Func}(n,n)]}$. We suppose that $\mathcal{A}$ makes a total of q queries, where the i^{th} query is (M^i, τ^i), also each M^i contains ℓ^i blocks, and $\sigma = \ell^1 + \ell^2 + \cdots + \ell^q$. Notice that the i^{th} query of $\mathcal{A}$ results in $\ell^i + 1$ many calls to the block cipher. Thus, all the q queries of $\mathcal{A}$ the block cipher is called with at most $\sigma + q \leq 2\sigma$ distinct inputs, thus by the PRP-PRF switching lemma (see Theorem 2.3.1), we have

$$\begin{aligned}\mathsf{Adv}^{\mathsf{vlPRF}}_{\mathsf{vlPMAC}[\mathsf{Perm}(n)]}(\mathcal{A}) &\leq \mathsf{Adv}^{\mathsf{vlPRF}}_{\mathsf{vlPMAC}[\mathsf{Func}(n,n)]}(\mathcal{A}) + \frac{1}{2^n}\binom{2\sigma}{2} \\ &\leq \mathsf{Adv}^{\mathsf{vlPRF}}_{\mathsf{vlPMAC}[\mathsf{Func}(n,n)]}(\mathcal{A}) + \frac{4\sigma^2}{2^n} \qquad (6.12)\end{aligned}$$

We are left with bounding the quantity $\mathsf{Adv}^{\mathsf{vlPRF}}_{\mathsf{vlPMAC}[\mathsf{Func}(n,n)]}(\mathcal{A})$ in Equation 6.12. For that purpose, we need to define a few events. Consider the random experiments $\mathsf{XColl1}(M,\tau)$, $\mathsf{XColl2}((M,\tau),(M',\tau'))$ and $\mathsf{XColl3}((M,\tau),(M',\tau'))$ defined in Figure 6.7.1.

We define $\mathsf{Coll}_1(M,\tau)$ to be the event that $\mathsf{XColl1}(M,\tau)$ returns true. Similarly, $\mathsf{Coll}_2((M,\tau),(M',\tau'))$ is the event that $\mathsf{XColl2}((M,\tau),(M',\tau'))$ returns true and $\mathsf{Coll}_3((M,\tau),(M',\tau'))$ denote the event $\mathsf{XColl3}((M,\tau),(M',\tau'))$ returns true.

The events $\mathsf{Coll}_1, \mathsf{Coll}_2, \mathsf{Coll}_3$ denotes non-trivial collisions in the domain of the random function which replaced the block cipher in the construction of vlMAC. We claim that the events $\mathsf{Coll}_1(M,\tau)$, $\mathsf{Coll}_2((M,\tau),(M',\tau'))$, $\mathsf{Coll}_3((M,\tau),(M',\tau'))$ are the failure events and if they do not occur then the output of $\mathsf{vlPRF}_{\mathsf{vlPMAC}[\mathsf{Func}(n,n)]}$ is indistinguishable from a random string.

Let $\mathsf{P}_1(\ell) = \max \mathsf{Coll}_1(M,\tau)$ where the maximum is taken over all (M,τ) where M contains ℓ blocks. Similarly, $\mathsf{P}_2(\ell,\ell') = \max \mathsf{Coll}_2\left((M,\tau),(M',\tau')\right)$, where the maximum is taken over all $(M,\tau),(M',\tau')$ where M contains ℓ blocks and M' contains ℓ' blocks. Finally, $\mathsf{P}_3(\ell,\ell') = \max \mathsf{Coll}_3\left((M,\tau),(M',\tau')\right)$, where the maximum is taken over all $(M,\tau),(M',\tau')$ where M contains ℓ blocks and M' contains ℓ' blocks. Then we have,

$$\mathsf{Adv}^{\mathsf{vlPRF}}_{\mathsf{vlPMAC}[\mathsf{Func}(n,n)]}(\mathcal{A}) \leq \max_{\substack{\ell^1,\ldots,\ell^q:\\ \sum \ell^i = \sigma}} \left\{ \sum_{1\leq r\leq q} \mathsf{P}_1(\ell^r) + \sum_{1\leq r<s\leq q} \mathsf{P}_2\left(\ell,\ell'\right) + \sum_{1\leq r<s\leq q} \mathsf{P}_3\left(\ell,\ell'\right) \right\} \tag{6.13}$$

Now bound the quantities $\mathsf{P}_1(\ell), \mathsf{P}_2\left(\ell,\ell'\right)$ and $\mathsf{P}_3\left(\ell,\ell'\right)$.

We refer to the variables in Figure 6.7.1. First, suppose $\overline{\mathsf{MColl}}' = \overline{\mathsf{MColl}} \setminus \{X'_{\ell'}\}$

```
XColl1$(M, \tau)$
$L \xleftarrow{\$} \{0,1\}^n$;
if $\ell = 1$ then
    if $|M_\ell| = n$ then $X_1 \leftarrow M_1 \oplus \gamma^{-1} \cdot L$;
    else $X_1 \leftarrow M_1 \oplus \gamma^{-2} \cdot L$;
else
    for $i \leftarrow 1$ to $\ell - 1$ do
        $X_i \leftarrow M_i \oplus \gamma^i \cdot L$;
        $Y_i \xleftarrow{\$} \{0,1\}^n$;
    end for
    $\Sigma \leftarrow Y_1 \oplus Y_2 \oplus \cdots \oplus Y_{\ell-1} \oplus \mathsf{Pad}(M_\ell)$;
    if $|M_\ell| = n$ then
        $X_\ell = \Sigma \oplus L \cdot \gamma^{-1}$;
    else
        $X_\ell = \Sigma$;
$\mathcal{X} \leftarrow \{X_1, \ldots, X_\ell\}$;
if collision occurs in $\{\tau\} \cup \mathcal{X}$ then
    return true;
else return false;
```

```
XColl2$((M, \tau), (M', \tau'))$
if $\tau \neq \tau'$ then
    return false
end if
$\mathsf{MColl} \leftarrow \{i \in [\min\{\ell, \ell' - 1\}] : M_i = M'_i\}$;
$\overline{\mathsf{MColl}} \leftarrow [\ell'] \setminus \mathsf{MColl}$;
for $i \leftarrow 1$ to $\ell' - 1$ do
    if $i \in \mathsf{MColl}$ then
        $X'_i \leftarrow X_i,\ Y'_i \leftarrow Y_i$;
    if $i \in \overline{\mathsf{MColl}}$ then
        $X'_i \leftarrow M'_i \oplus \gamma^i \cdot L,\ Y'_i \xleftarrow{\$} \{0,1\}^n$;
end for
$\Sigma' \leftarrow \sum_{i=1}^{\ell'-1} Y'_i \oplus \mathsf{Pad}(M'_{\ell'})$;
if $|M'_{\ell'}| = n$ then
    $X'_{\ell'} \leftarrow \Sigma' \oplus \gamma^{-1} \cdot L$;
else
    $X'_{\ell'} \leftarrow \Sigma'$;
$\mathcal{X}'_1 \leftarrow \{X'_i : i \in \overline{\mathsf{MColl}}\}$;
if $\mathcal{X} \cap \mathcal{X}' \neq \phi$ then
    return true
else return false
```

```
XColl3$((M, \tau), (M', \tau'))$
if $\tau = \tau'$ then
    return false;
$L' \xleftarrow{\$} \{0,1\}^n$;
for $i \leftarrow 1$ to $\ell' - 1$ do
    $X'_i \leftarrow M'_i \oplus \gamma^i \cdot L',\ Y'_i \leftarrow \{0,1\}^n$;
$\Sigma' \leftarrow \sum_{i=1}^{\ell'-1} Y'_i \oplus \mathsf{Pad}(M'_{\ell'})$;
if $|M'_{\ell'}| = n$ then
    $X'_{\ell'} \leftarrow \Sigma' \oplus \gamma^{-1} \cdot L'$;
else
    $X'_{\ell'} \leftarrow \Sigma'$;
$\mathcal{X}'_2 \leftarrow \{X'_1, X'_2, \ldots, X'_{\ell'}\}$;
if $\mathcal{X} \cap \mathcal{X}'_2 \neq \phi$ then
    return true;
else return false;
```

Figure 6.7.1: Description of the collision events of vIPMAC.

and define

$$E_1 = \{\tau\} \quad E_2 = \{X_1, \ldots, X_{\ell-1}\} \quad E_3 = \{X_\ell\}$$

$$E_4 = \{X'_j : j \in \overline{\mathsf{MColl}}'\} \quad E_5 = \{X'_{\ell'}\} \quad E_6 = \{\tau'\} \quad E_7 = \{X'_1, \ldots, X'_{\ell'-1}\}.$$

Let, $\mathsf{B}(E_i, E_j)$ be the event that represents the collision between E_i and E_j.

Bounding $\mathsf{B}(E_1, E_2)$: $\Pr[X_i = \tau] = \Pr[M_i \oplus \gamma^i \cdot L = \tau] = 1/2^n$, since γ^i is nonzero and L is chosen uniformly random.

Bounding $\mathsf{B}(E_1, E_3)$: If $|M_\ell| < n$ and $\ell \geq 2$ then Σ is random. Therefore,

$$\Pr[X_\ell = \tau] = \Pr[\Sigma = \tau] = 1/2^n.$$

If $|M_\ell| < n$ and $\ell = 1$ then

$$\Pr[X_\ell = \tau] = \Pr[M_1 \oplus \gamma^{-2} \cdot L = \tau] = 1/2^n,$$

as L is chosen uniformly random.

If $|M_\ell| = n$ and $\ell \geq 2$ then Σ is random and independent from L, so

$$\Pr[X_\ell = \tau] = \Pr[\Sigma = \tau \oplus x^{-1} \cdot L] = 1/2^n.$$

If $|M_\ell| = n$ and $\ell = 1$ then

$$\Pr[X_\ell = \tau] = \Pr[M_1 \oplus \gamma^{-1} \cdot L = \tau] = 1/2^n,$$

as L is chosen uniformly random.

Bounding $\mathsf{B}(E_2, E_2)$: Suppose $i, j \in [\ell - 1]$ and $i < j$. So, $\Pr[X_i = X_j] = \Pr[M_i \oplus M_j = (\gamma^i \oplus \gamma^j) \cdot L] = 1/2^n$ because $\gamma^i \neq \gamma^j$ and L is random.

Bounding $\mathsf{B}(E_2, E_3)$: If $|M_\ell| < n$, then $\Pr[X_i = X_\ell] = \Pr[M_i \oplus \gamma^i \cdot L = \Sigma] = 1/2^n$, as Σ is uniformly random and independent to L.

If $|M_\ell| = n$, then $\Pr[X_i = X_\ell] = \Pr[M_i \oplus \gamma^i \cdot L = \Sigma \oplus \gamma^{-1} \cdot L] = 1/2^n$, as $\gamma^i \neq \gamma^{-1}$

(assume that, $i < 2^{n-2}$) and Σ is uniformly random and independent to L.

Considering the probabilities of those events we get,

$$\Pr[\mathsf{P}_1(\ell)] \leq \binom{\ell+1}{2}\frac{1}{2^n}. \tag{6.14}$$

Now, we have bound the events for $\mathsf{P}_2\,(\ell,\ell')$.

Bounding $\mathsf{B}(E_2, E_4)$: In this event, $i \in [\ell-1]$ and $j \in \overline{\mathsf{MColl}}'$. The event can be written as

$$\begin{aligned}\Pr[X_i = X'_j] &= \Pr[M_i \oplus \gamma^i \cdot L = M'_j \oplus \gamma^j \cdot L] \\ &= \Pr[M_i \oplus M'_j = (\gamma^i \oplus \gamma^j) \cdot L].\end{aligned}$$

If $i \neq j$ then $\gamma^i \neq \gamma^j$, thus the probability is $1/2^n$ (as L is random). If $i = j$ then the probability is zero since $M_i \neq M'_j$.

Bounding $\mathsf{B}(E_2, E_5)$: First consider the case $|M'_{\ell'}| < n$. Here, $\Pr[X_i = X'_{\ell'}] = \Pr[M_i \oplus \gamma^i \cdot L = \Sigma'] = 1/2^n$, as Σ' is uniformly random and independent to L.
If $|M'_{\ell'}| = n$, then $\Pr[X_i = X'_{\ell'}] = \Pr[M_i \oplus \gamma^i \cdot L = \Sigma' \oplus \gamma^{-1} \cdot L] = 1/2^n$ because $\gamma^i \neq \gamma^{-1}$ and Σ' is uniformly random and independent to L.

Bounding $\mathsf{B}(E_3, E_4)$: This event is similar to $\mathsf{B}(E_2, E_5)$, so the probability of this event is also $1/2^n$.

Bounding $\mathsf{B}(E_3, E_5)$:

Case 1. Consider the case $|M_\ell| < n$ and $|M'_{\ell'}| < n$. If $\ell \neq \ell'$, w.l.g. take $\ell > \ell'$. Now, $\Pr[X_\ell = X'_{\ell'}] = \Pr[\Sigma = \Sigma'] = 1/2^n$, as the contribution of $Y_{\ell-1}$ in Σ is not used in the definition of Σ'. If $\ell = \ell'$ and for any $i < \ell$ such that $M_i \neq M'_i$, then $\Pr[X_\ell = X'_{\ell'}] = 1/2^n$, and if $\ell = \ell'$ and for all $i < \ell$ such that $M_i = M'_i$, then $M_\ell = M'_{\ell'}$, so $\Pr[X_\ell = X'_{\ell'}] = 0$.

Case 2. Consider the case $|M_\ell| = n$ and $|M'_{\ell'}| = n$. In this case $\Pr[X_\ell = X'_{\ell'}] = \Pr[\Sigma \oplus \gamma^{-1} \cdot L = \Sigma' \oplus \gamma^{-1} \cdot L] = \Pr[\Sigma = \Sigma'] = 1/2^n$, using the same argument described in **Case 1**.

Case 3. Consider the case $|M_\ell| < n$ and $|M'_{\ell'}| = n$. Then $\Pr[X_\ell = X'_{\ell'}] = \Pr[\Sigma =$

$\Sigma' \oplus \gamma^{-1} \cdot L] = 1/2^n$, since Σ and Σ' are random and independent to L.

Case 4. Consider the case $|M_\ell| = n$ and $|M'_{\ell'}| < n$. This event is similar to **Case 3**, so the probability is $1/2^n$.

Considering the probabilities of those events, we get,

$$\Pr[\mathsf{P}_2\left(\ell, \ell'\right)] \leq \frac{\ell\ell'}{2^n}, \tag{6.15}$$

as $|E_2 \cup E_3| \cdot |E_4 \cup E_5| < \ell\ell'$.

Now, we compute a bound for $\mathsf{P}_3\left(\ell, \ell'\right)$.

<u>**Bounding** $\mathsf{B}(E_2, E_7)$:</u> In this event $\Pr[X_i = X'_j] = \Pr[M_i \oplus \gamma^i \cdot L = M'_j \oplus \gamma^j \cdot L'] = 1/2^n$, as L and L' are uniformly random and independent.

<u>**Bounding** $\mathsf{B}(E_2, E_5)$:</u> First consider the case $|M'_{\ell'}| < n$. Here, $\Pr[X_i = X'_{\ell'}] = \Pr[M_i \oplus \gamma^i \cdot L = \Sigma'] = 1/2^n$, as Σ' is uniformly random and independent to L.
If $|M'_{\ell'}| = n$, then $\Pr[X_i = X'_{\ell'}] = \Pr[M_i \oplus \gamma^i \cdot L = \Sigma' \oplus \gamma^{-1} \cdot L'] = 1/2^n$ because $\gamma^i \neq \gamma^{-1}$ and Σ' is uniformly random and independent to L and L'.

<u>**Bounding** $\mathsf{B}(E_3, E_7)$:</u> This event is similar to $\mathsf{B}(E_2, E_5)$, so the probability of this event is also $1/2^n$.

<u>**Bounding** $\mathsf{B}(E_3, E_5)$:</u> **Case 1.** Consider the case $|M_\ell| < n$ and $|M'_{\ell'}| < n$. Now, $\Pr[X_\ell = X'_{\ell'}] = \Pr[\Sigma = \Sigma'] = 1/2^n$, as Σ and Σ' are uniformly random and independent.

Case 2. Consider the case $|M_\ell| = n$ and $|M'_{\ell'}| = n$. In this case $\Pr[X_\ell = X'_{\ell'}] = \Pr[\Sigma \oplus \gamma^{-1} \cdot L = \Sigma' \oplus \gamma^{-1} \cdot L'] = 1/2^n$, as Σ and Σ' are uniformly random and independent to L and L'.

Case 3. Consider the case $|M_\ell| < n$ and $|M'_{\ell'}| = n$. Then $\Pr[X_\ell = X'_{\ell'}] = \Pr[\Sigma = \Sigma' \oplus \gamma^{-1} \cdot L'] = 1/2^n$, since Σ and Σ' are random and independent to L'.

Case 4. Consider the case $|M_\ell| = n$ and $|M'_{\ell'}| < n$. This event is similar to **Case 3**, so the probability is $1/2^n$.

Considering the probabilities of all the above events, we get,

$$\Pr[\mathsf{P}_3\left(\ell,\ell'\right)] \leq \frac{\ell\ell'}{2^n}, \tag{6.16}$$

as $|E_2 \cup E_3| \cdot |E_5 \cup E_7| < \ell\ell'$.

Therefore, from Equations (6.14),(6.15) and (6.16) we get,

$$\begin{aligned}
& \max_{\substack{\ell^1,\ldots,\ell^q:\\ \sum \ell^i=\sigma}} \left\{ \sum_{1\leq r\leq q} \mathsf{P}_1(\ell^r) + \sum_{1\leq r<s\leq q} \mathsf{P}_2\left((\ell^r,\ell^s)\right) + \sum_{1\leq r<s\leq q} \mathsf{P}_3\left((\ell^r,\ell^s)\right) \right\} \\
\leq \quad & \max_{\substack{\ell^1,\ldots,\ell^q:\\ \sum \ell^i=\sigma}} \left\{ \binom{\ell+1}{2}\frac{1}{2^n} + 2\sum_{1\leq r<s\leq q} \frac{\ell^r\ell^s}{2^n} \right\} \\
\leq \quad & \frac{(\sigma+1)^2}{2^{n+1}} + \frac{\sigma^2}{2^n} \\
\leq \quad & \frac{3\sigma^2}{2^n}. \qquad (6.17)
\end{aligned}$$

Finally, using the Equations (6.12), (6.13) and (6.17), we get,

$$\mathsf{Adv}^{\mathsf{vlPRF}}_{\mathsf{vlPMAC[Perm}(n)]}(\mathcal{A}) \leq \frac{7\sigma^2}{2^n}. \tag{6.18}$$

7 Tight Security Bound of 2k-LightMAC_Plus

In FSE'16 [65], Luykx et al. have proposed LightMAC, which has been standardized by ISO/IEC standardization process. LightMAC is a block cipher based PRF that operates in parallel mode, i.e., for an n-bit block cipher E instantiated with two independently sampled keys K_1, K_2, and with a global counter size s, the LightMAC function is defined as follows:

$$\mathsf{LightMAC}_{\mathsf{E}_{K_1,K_2}}(M) = \mathsf{E}_{K_2}\left(\sum_{i=1}^{\ell-1} \mathsf{E}_{K_1}(\langle i \rangle_s \| M[i]) \oplus \mathsf{pad}_n(M[\ell])\right),$$

where $\langle i \rangle_s$ denotes the s bit encoding of the integer i and $(M[1], \ldots, M[\ell])$ denotes the $n-s$ bit parsing of message M, where each $M[i]$ is an $n-s$ bit string, and pad_n is an injective function that takes a message and appends to it a suitable number of 10^* to make the length of the padded string to be exactly n. However, this design comes at the cost of a reduced rate of construction, where the rate of a construction is determined by the ratio of the total number of n-bit message blocks in a message M to the total number of primitive calls with block size n required to process the message M. Despite having a reduced rate, the design of LightMAC is simple in the sense that it minimizes all auxiliary operations other than having the block cipher calls, which allows to have a low overhead cost, and hence obtains a more compact implementation than PMAC [14]. Moreover, due to the inherent parallelism in the design of the scheme, LightMAC outperforms all the other popular sequential MAC constructions in terms of throughput in the parallel computing infrastructure.

7.1 Beyond Birthday Bound Secure Variants of LightMAC

Over the years, there have been many proposals of variants of LightMAC construction achieving beyond the birthday bound security. In 2017, Naito [70] proposed LightMAC_Plus construction based on three block cipher keys and showed that it gives $2n/3$-bit security. In fact, LightMAC_Plus is the first beyond the birthday bound-secure PRF whose proven security bound does not depend on the message length. In the same paper, the author has also proposed LightMAC_Plus2 [70] that provides a higher security bound than LightMAC_Plus or LightMAC, but it comes at the increased number of block cipher calls. In CT-RSA'18 [71], Naito has improved the bound of the LightMAC_Plus construction from $q^3/2^{2n}$ to $q_t^2 q_v/2^{2n}$, where q_t is the number of tagging queries and q_v is the number of verification queries. This security bound implies that LightMAC_Plus is secure up to 2^n tagging queries if the number of verification queries is 1. Later, in [61], Leurent et al. have shown a forging attack on the construction that achieves a constant success probability when the number of tagging queries is $2^{3n/4}$ and the number of verification queries is 1, which in turn invalidates the security claim of Naito [71] on LightMAC_Plus. In EUROCRYPT'20, Kim et al. [56] have claimed an improved security bound (but did not supply any formal proof to back up the claim) of LightMAC_Plus construction from $2n/3$-bits to $3n/4$-bits, and due to the result of [61], the improved bound of LightMAC_Plus turns out to be the tight one.

In FSE'18, Datta et al. [37] proposed a two-keyed variant of LightMAC_Plus, called 2K-LightMAC_Plus, where the sum function used in the finalization phase uses the same block cipher key that is independent to the block cipher key used in the internal hash computation of 2K-LightMAC_Plus. Authors have shown that 2K-LightMAC_Plus achieves $2n/3$-bits security bound. In [71], Naito has proposed a single-keyed variant of LightMAC_Plus, dubbed as LightMAC_Plus-1k, in which a single block cipher key is used in the entire construction. However, the $2n$-bits output (Σ, Θ) of the internal hash computation is domain separated by setting their two most significant bits to it 10 and 11, respectively. Moreover, the checksum of the message blocks after padded with the string 0^{n-s} is masked with the Σ value. Author has shown that LightMAC_Plus-1k achieves $2n/3$-bits security. Recently, Song [83] proposed another variant of the single-keyed LightMAC_Plus construction dubbed as 1k-LightMAC_Plus, in which a single block cipher is used throughout the construction and the $2n$-bit hash value is domain separated by setting their most significant bit to 0 and 1 respectively. It has been shown in [83] that 1k-LightMAC_Plus also achieves $2n/3$-bits security bound.

Therefore, to summarize, only the LightMAC_Plus construction has been claimed to achieve a tight $3n/4$-bit security bound [56], and all its existing reduced-keyed variants achieve only $2n/3$-bits security. Therefore, the motivation for this chapter stems from asking the question

> *Can we prove a tight $3n/4$-bit security bound on any reduced-keyed variants of the* LightMAC_Plus *construction?*

7.2 Our Contribution

In this chapter, we answer the above question affirmatively and show that the construction achieves a tight security bound up to $2^{3n/4}$ queries (ignoring the maximum message length). In particular, we have shown an upper bound on the PRF advantage of 2k-LightMAC_Plus in roughly of the order of $2^{3n/4}$ queries, provided the maximum number of message blocks in a query is at most $\min\{2^{n-2}-1, 2^s\}$, and the total number of distinct message blocks across all queries is at most 2^n, where n denotes the block size of the block cipher and s denotes the size of the block counter. Moreover, we have also shown a matching PRF attack on the construction with query complexity in roughly of the order of $2^{3n/4}$ queries. The schematic diagram of 2k-LightMAC_Plus is shown in Fig. 7.4.2 and its algorithmic description is shown in Fig. 7.4.1 respectively. However, to prove the security bound of the construction, we deeply rely on the result of the mirror theory, where we lower the bound on the number of solutions of a given system of equations.

The following result from linear algebra will be very useful in establishing the security bound of our construction. Proof of this result can be found in Proposition 1 of [38].

Lemma 7.2.1. *Let* $(Z_1, \ldots, Z_q) \xleftarrow{\text{wor}} \mathcal{X} \subseteq \{0,1\}^n$ *with* $|\mathcal{X}| = N > q$. *Let* A *be a* $k \times q$ *binary matrix with rank* r. *We denote the column vector* $(Z_1, \ldots, Z_q)^{\mathtt{tr}}$ *as* $\widetilde{\mathsf{Z}}$. *Then, for any* $\widetilde{\mathsf{c}} \in (\{0,1\}^n)^k$, *we have*

$$\Pr[A \cdot \widetilde{\mathsf{Z}} = \widetilde{\mathsf{c}}] \leq \frac{1}{(N-q+r)_r}.$$

7.3 Mirror Theory

Suppose $\mathsf{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ be an an undirected edge-labelled acylic graph, where $\mathcal{V} = \{P_1, \ldots, P_\alpha\}$ and $\mathcal{E}$ be the vertex and edge set of G respectively and $\mathcal{L} : \mathcal{E} \to \{0,1\}^n$ be the edge labelling function. For an edge $\{P_i, P_j\} \in \mathcal{E}$, we write $\mathcal{L}(\{P_i, P_j\}) = \lambda_{ij}$.

Consider a path $\mathcal{P}$ and a cycle $\mathcal{C}$ in the graph G. Now, we define the label of the path as $\mathcal{L}(\mathcal{P}) \triangleq \sum_{e \in \mathcal{P}} \mathcal{L}(e)$ and the label of the cycle as $\mathcal{L}(\mathcal{C}) \triangleq \sum_{e \in \mathcal{C}} \mathcal{L}(e)$. We say the graph G is **good** if the graph is acyclic and $\mathcal{L}(\mathcal{P}) \neq \mathbf{0}$ for any path $\mathcal{P}$ in the graph G . For such a good graph G, we associate a system of bivariate affine equations as follows:

$$\mathcal{E}_{\mathsf{G}} = Y_i \oplus Z_j = \lambda_{ij} \ \forall \ \{Y_i, Z_j\} \in \mathcal{E}.$$

In the mentioned set of bivariate affine equations, the variables correspond to the graph's vertices. Two variables are considered involved in an equation if their corresponding vertices are connected by an edge in the graph. The constants of the equations are the labels of the corresponding edges of the graph. So, for the system of affine equations $\mathcal{E}_{\mathsf{G}}$, the variables are Y_i's and Z_i's. Now, we define an equivalence relation $\sim$ over $\mathcal{V}$ such that $u \sim v$ if and only if $(u, v) \in \mathcal{E}$. This equivalence relation $\sim$ makes a partition on $\mathcal{V}$ and each partition is called a component. The size of a component is the number of elements (i.e., the number of vertices) present in the partition. The set of components in G is denoted by $\mathsf{comp}(\mathsf{G}) = (\mathsf{C}_1 \sqcup \ldots \sqcup \mathsf{C}_\alpha \sqcup \mathsf{D}_1 \sqcup \ldots \sqcup \mathsf{D}_\beta)$

where we assume that there are α many components of G (i.e., $\mathsf{C}_1, \ldots, \mathsf{C}_\alpha$) whose size greater than 2 and β many components of G (i.e., $\mathsf{D}_1, \ldots, \mathsf{D}_\beta$) having size exactly 2. Suppose, $\mathsf{C} = \mathsf{C}_1 \sqcup \ldots \sqcup \mathsf{C}_\alpha$ and $\mathsf{D} = \mathsf{D}_1 \sqcup \ldots \sqcup \mathsf{D}_\beta$. Let the total number of edges in C be denoted by q_c and the total number of edges in the graph G is denoted by q. Then, it is easy to see that $q = q_c + \beta$.

Notations: For the i-th component of C, i.e., C_i, which is acyclic and edge-labelled graph, let $\mathcal{V}_{\mathsf{C}_i}$ be the set of vertices of the component C_i and w_i denotes the cardinality of the set $\mathcal{V}_{\mathsf{C}_i}$. Let $\mathcal{V}_{\mathsf{C}}$ denotes the set of vertices of C. For $1 \leq i \leq \alpha$, we write $\sigma_i = w_1 + w_2 + \ldots + w_i$, with the convention that $\sigma_0 = 0$. Note that $q_c = \sigma_\alpha - \alpha$ as each component C_i is a tree. Let $h(\mathsf{G})$ denote the number of solutions to the graph G. Let $h_c(i)$ denote the number of solutions for the subgraph $\mathsf{C}_1 \sqcup \mathsf{C}_2 \sqcup \ldots \sqcup \mathsf{C}_i$ and $h_d(i)$ denote the number of solutions for the subgraph $\mathsf{C} \sqcup \mathsf{D}^i$ where $\mathsf{D}^i \triangleq \mathsf{D}_1 \sqcup \mathsf{D}_2 \sqcup \ldots \sqcup \mathsf{D}_i$. Therefore, $h_d(0) = h_c(\alpha)$ and $h_d(\beta) = h(\mathsf{G})$.

Definition 7.3.1. *Let $\mathcal{E}_{\mathsf{G}}$ be a system of equations corresponding to a good acyclic edge-labeled graph G (as defined above). An injective function $\Phi : \mathcal{V} \to \{0,1\}^n$, is said to be an injective solution to $\mathcal{E}_{\mathsf{G}}$ if $\Phi(P_i) \oplus \Phi(P_j) = \lambda_{ij}$ for all $\{P_i, P_j\} \in \mathcal{E}$ such that $\mathcal{L}(\{P_i, P_j\}) = \lambda_{ij}$.*

In [35], authors have proved that if G is a good acyclic edge-labeled graph such that it is decomposed into finitely many components of size greater than 2 and exactly 2, then the number of injective solutions chosen from $\{0,1\}^n$, to $\mathcal{E}_{\mathsf{G}}$, is very close to the average number of solutions until the number of edges in $\mathcal{E}$ is roughly $2^{3n/4}$. Formally, the result is as follows:

Theorem 7.3.2. *Let $\mathsf{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ be a good acylic edge-labelled graph with $|\mathcal{E}| = q$ edges and s vertices such that G is decomposed into α many components $\mathsf{C}_1 \sqcup \ldots \sqcup \mathsf{C}_\alpha$ of size at least 3 and β many components $\mathsf{D}_1 \sqcup \ldots \sqcup \mathsf{D}_\beta$ of size exactly 2. For $1 \leq i \leq \alpha$, let w_i be the total number of vertices of $\mathsf{C}_1 \sqcup \ldots \sqcup \mathsf{C}_i$ and q_c be the total number of edges in $\mathsf{C}_1 \sqcup \ldots \sqcup \mathsf{C}_\alpha$. Let $\sigma_\alpha = w_1 + w_2 + \ldots + w_\alpha$ be the total number of vertices of $\mathsf{C}_1 \sqcup \mathsf{C}_2 \sqcup \ldots \sqcup \mathsf{C}_\alpha$. Then the total number of injective solutions to $\mathcal{E}_{\mathsf{G}}$ which are* *chosen from* $\{0,1\}^n$ *is at least:*

$$\frac{(2^n)_s}{2^{nq}}\left(1-\frac{9q_c^2}{4\cdot 2^n}-\frac{9q_c^2q}{2^{2n}}-\frac{24q^2q_c}{2^{2n}}-\frac{6qq_c}{2^{2n}}-\frac{40q^2}{2^{2n}}-\frac{16q^4}{2^{3n}}\right).$$

We refer the interested reader to [35] for proof of the result.

7.4 2k-LightMAC_Plus

In this section, we revisit the 2k-LightMAC_Plus construction proposed by Datta et al. [37]. The algorithmic specification and the pictorial description of the construction are depicted in Fig. 7.4.1 and Fig. 7.4.2 respectively. We would like to point out 2k-LightMAC_Plus is structurally similar to 1k-LightMAC_Plus [83], except that the block cipher key used in the finalization phase is independent of the block cipher key used in the hash function.

```
Algorithm 2k-LightMAC_Plus[E]
$(M[1],\ldots,M[\ell]) \xleftarrow{n-s} M$;
for $i = 1$ to $\ell$ do
    $X[i] \leftarrow \langle i\rangle_s \| M[i]$;
    $Y[i] \leftarrow \mathsf{E}_{K_1}(X[i])$;
end for;
$\Sigma' \leftarrow Y[1] \oplus Y[2] \oplus \ldots \oplus Y[\ell]$;
$\Theta' \leftarrow 2^{\ell}Y[1] \oplus 2^{\ell-1}Y[2] \oplus \ldots \oplus 2Y[\ell]$;
$\Sigma \leftarrow \mathsf{fix}_0(\Sigma'),\quad \Theta \leftarrow \mathsf{fix}_1(\Theta')$;
$T \leftarrow \mathsf{E}_{K_2}(\Sigma) \oplus \mathsf{E}_{K_2}(\Theta)$;
return $T$;
```

Figure 7.4.1: Algorithmic Specification of the 2k-LightMAC_Plus construction proposed by Datta et al. [37]. fix_0 and fix_1 are two functions that take an n-bit input and return an n-bit output string such that its most significant bit is set to 0 and 1 respectively. s denotes the size of the block counter. $\langle i\rangle_s$ denotes the s bit binary representation of integer i.

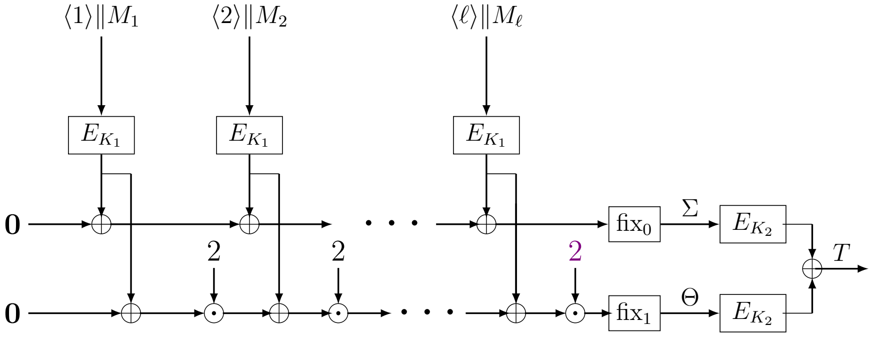

Figure 7.4.2: Pictorial description of the 2k-LightMAC_Plus [37].

7.4.1 Security Result of 2k-LightMAC_Plus

The existing security result of 2k-LightMAC_Plus by Datta et al. [37] shows that the construction is secured against all information-theoretic distinguishers under the pseudorandom permutation assumption of the underlying block cipher of 2k-LightMAC_Plus that makes roughly up to $2^{2n/3}$ queries such that the maximum number of message blocks in a query is at most $\min\{2^{n-2}-1, 2^s\}$, where n being the block size of the underlying block cipher and s denotes the size of the block counter. Now, we state and prove that 2k-LightMAC_Plus is secured against all information-theoretic distinguishers under the pseudorandom permutation assumption of the underlying block cipher of 2k-LightMAC_Plus that makes roughly up to $2^{3n/4}$ queries such that the maximum number of message blocks in a query is at most $\min\{2^{n-2}-1, 2^s\}$, and the total number of message blocks $\sigma \leq 2^n$. Formally, we state the following security result:

Theorem 7.4.1. *Let $\mathcal{K}$ be a finite and non-empty set. Let $\mathsf{E} : \mathcal{K} \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a block cipher. Then, the PRF advantage for any (q, ℓ, σ, t) adversary against* 2k-LightMAC_Plus[E] *is given by,*

$$\begin{aligned}\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{2k\text{-}LightMAC\_Plus[E]}}(q,\ell,\sigma,t) \;\leq\;& 2\mathbf{Adv}^{\mathrm{PRP}}_{\mathsf{E}}(\sigma+2q,t') + \frac{96q^4}{2^{3n}} + \frac{8\sqrt{2}q^2}{2^{3n/2}} + \frac{7q^{4/3}}{2^n}\\ &+\frac{39q^{8/3}}{2^{2n}} + \frac{244q^2}{2^{2n}} + \frac{32q^3}{2^{3n}} + \frac{6\sigma}{2^n} + \frac{q}{2^n} + \frac{8}{2^n},\end{aligned}$$

where $\ell \leq \min\{2^{n-2} - 1, 2^s\}$*, is the maximum number of message blocks in a query,* $\sigma \leq 2^n$*, is the total number of distinct message blocks queried, and* $t' = O((\sigma + 2q)t)$*.*

7.5 Proof of Theorem 7.4.1

As the first step of the proof, we replace the underlying block ciphers E_{K_1} and E_{K_2} of the construction with a pair of uniformly sampled n-bit random permutations P_1 and P_2 at the cost of the prp advantage of E and denote the resulting construction as $\mathsf{2k\text{-}LightMAC\_Plus}^*[\mathsf{P}_1, \mathsf{P}_2]$, i.e.,

$$\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{2k\text{-}LightMAC\_Plus[E]}}(q, \sigma, t) \leq 2\mathbf{Adv}^{\mathrm{PRP}}_{\mathsf{E}}(\sigma, t') + \mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{2k\text{-}LightMAC\_Plus}^*[\mathsf{P}_1,\mathsf{P}_2]}(q, \sigma).$$

We write $\mathsf{2k\text{-}LightMAC\_Plus}$ or $\mathsf{2k\text{-}LightMAC\_Plus}^*$ instead of $\mathsf{2k\text{-}LightMAC\_Plus[E]}$ or $\mathsf{2k\text{-}LightMAC\_Plus}^*[\mathsf{P}_1, \mathsf{P}_2]$ whenever the primitives are understood from the context. Now, our goal is to upper bound the information-theoretic PRF security of $\mathsf{2k\text{-}LightMAC\_Plus}^*$. For doing this, we bound the PRF security of $\mathsf{2k\text{-}LightMAC\_Plus}^*$ in terms of the distinguishing advantage of an information-theoretic distinguisher D in distinguishing the output of $\mathsf{2k\text{-}LightMAC\_Plus}^*$ from the output of an ideal world that consists of a random function RF which outputs a random n-bit tag T on every input $M \in \mathcal{M}$. We assume that the distinguisher D makes q queries to the oracle in either of the two worlds and at the end of the interaction, the oracle releases some additional information to D. If D interacts with the oracle in the real world, then it releases $\widetilde{\Sigma} = (\Sigma_1, \Sigma_2, \ldots, \Sigma_q)$ and $\widetilde{\Theta} = (\Theta_1, \Theta_2, \ldots, \Theta_q)$. However, if D interacts with the oracle in the ideal world, then the oracle also releases $\widetilde{\Sigma}$, $\widetilde{\Theta}$ tuple, where the tuple $\widetilde{\Sigma}$, and $\widetilde{\Theta}$ are computed in the ideal world as described in the following section.

7.5.1 Description of The Ideal World

The ideal oracle consists of two phases: (i) online phase in which for each queried message M^i, the oracle samples the response T_i uniformly at random from $\{0, 1\}^n$ and returns it to the distinguisher D. If it happens that any of the sampled responses are all zero strings, then we set the bad flag $\mathsf{Bad\text{-}Tag}$ to 1 and abort the game, i.e.,

$$\mathsf{Bad\text{-}Tag} \leftarrow 1 : \exists i \in [q] : T_i = 0^n.$$

When all the queries and responses are over, the offline phase of the ideal world begins. In this phase, we consider a function $\mathcal{L}_1$, which is initially undefined at every point of its domain. The oracle of the ideal world computes $X^i_j = \langle j \rangle_s \| M^i_j$ values for all $i \in [q], j \in [\ell_i]$ and samples Y^i_j as follows: (a) if X^i_j is fresh in $\widetilde{X}$, then Y^i_j is uniformly sampled from outside of the set $\mathsf{Ran}(\mathcal{L}_1)$ followed by including it to the set $\mathsf{Ran}(\mathcal{L}_1)$; (ii) on the other hand, if X^i_j collides with some previous $X^{i'}_{j'}$ value, where $(i', j') \preceq (i, j)$, then Y^i_j is set to the value $Y^{i'}_{j'}$. When all the Y^i_j, for $i \in [q], j \in [\ell_i]$ are determined, the oracle computes the tuple (Σ_i, Θ_i) for all $i \in [q]$ as

$$\Sigma_i = \mathsf{fix}_0(Y^i_1 \oplus Y^i_2 \oplus \ldots \oplus Y^i_{\ell_i}), \Theta_i = \mathsf{fix}_1(2^{\ell_i} Y^i_1 \oplus 2^{\ell_i - 1} Y^i_2 \oplus \ldots \oplus 2Y^i_{\ell_i}).$$

After the computation of the tuple $(\widetilde{\Sigma}, \widetilde{\Theta})$ is over, we set the bad flag $\mathsf{Bad1}$ to 1, if there exists two pairs (Σ_i, Θ_i) and (Σ_j, Θ_j) such that $(\Sigma_i, \Theta_i) = (\Sigma_j, \Theta_j)$ holds, i.e.,

$$\mathsf{Bad1} \leftarrow 1 : \exists i \neq j \in [q] : (\Sigma_i, \Theta_i) = (\Sigma_j, \Theta_j).$$

Moreover, we set the bad flag $\mathsf{Bad2}$ to 1, if there exists two pairs (Σ_i, T_i) and (Σ_j, T_j) such that $(\Sigma_i, T_i) = (\Sigma_j, T_j)$ holds, i.e.,

$$\mathsf{Bad2} \leftarrow 1 : \exists i \neq j \in [q] : (\Sigma_i, T_i) = (\Sigma_j, T_j).$$

Similarly, we set the bad flag $\mathsf{Bad3}$ to 1, if there exists two pairs (Θ_i, T_i) and (Θ_j, T_j) such that $(\Theta_i, T_i) = (\Theta_j, T_j)$ holds, i.e.,

$$\mathsf{Bad3} \leftarrow 1 : \exists i \neq j \in [q] : (\Theta_i, T_i) = (\Theta_j, T_j).$$

We set the bad flag $\mathsf{Bad4}$ to 1 if there exists three distinct indices $i_1, i_2, i_3 \in [q]$ such that $\Sigma_{i_1} = \Sigma_{i_2}, \Theta_{i_2} = \Theta_{i_3}, T_{i_1} \oplus T_{i_2} \oplus T_{i_3} = 0^n$ holds, i.e.,

$$\mathsf{Bad4} \leftarrow 1 : \exists i_1, i_2, i_3 \in [q] : \Sigma_{i_1} = \Sigma_{i_2}, \Theta_{i_2} = \Theta_{i_3}, T_{i_1} \oplus T_{i_2} \oplus T_{i_3} = 0^n.$$

We set the bad flag Bad5 to 1 if there exists four distinct indices $i_1, i_2, i_3, i_4 \in [q]$ such that $\Sigma_{i_1} = \Sigma_{i_2}, \Theta_{i_2} = \Theta_{i_3}, \Sigma_{i_3} = \Sigma_{i_4}$ holds, i.e.,

$$\mathsf{Bad5} \leftarrow 1 : \exists i_1, i_2, i_3, i_4 \in [q] : \Sigma_{i_1} = \Sigma_{i_2}, \Theta_{i_2} = \Theta_{i_3}, \Sigma_{i_3} = \Sigma_{i_4}.$$

We set the bad flag Bad6 to 1 if there exists four distinct indices $i_1, i_2, i_3, i_4 \in [q]$ such that $\Theta_{i_1} = \Theta_{i_2}, \Sigma_{i_2} = \Sigma_{i_3}, \Theta_{i_3} = \Theta_{i_4}, T_{i_1} \oplus T_{i_2} \oplus T_{i_3} \oplus T_{i_4} = 0^n$ holds, i.e.,

$$\mathsf{Bad6} \leftarrow 1 : \exists i_1, i_2, i_3, i_4 \in [q] : \Theta_{i_1} = \Theta_{i_2}, \Sigma_{i_2} = \Sigma_{i_3}, \Theta_{i_3} = \Theta_{i_4}, T_{i_1} \oplus T_{i_2} \oplus T_{i_3} \oplus T_{i_4} = 0^n.$$

Finally, we set the bad flag Bad7 to 1 if the number of colliding pairs for Σ or Θ values is at least $q^{2/3}$, i.e.,

$$\mathsf{Bad7} \leftarrow 1 : \begin{cases} |\{(i,j) : i \neq j \in [q], \Sigma_i = \Sigma_j\}| \geq q^{2/3} \text{ or} \\ |\{(i,j) : i \neq j \in [q], \Theta_i = \Theta_j\}| \geq q^{2/3}. \end{cases}$$

The offline phase of the ideal world is depicted in Fig. 7.5.1.

Therefore, we summarize the interaction of D with the oracle in the following attack transcript

$$\tau = \{(M_1, T_1, \Sigma_1, \Theta_1), (M_2, T_2, \Sigma_2, \Theta_2), \ldots, (M_q, T_q, \Sigma_q, \Theta_q)\}.$$

Let T_{re} denote the random variable that takes a transcript τ realized in the real world. Similarly, T_{id} denotes the random variable that takes a transcript τ realized in the ideal world. The probability of realizing a transcript τ in the ideal (resp. real) world is called the *ideal* (*resp. real*) *interpolation probability.* A transcript τ is said to be attainable with respect to D if its ideal interpolation probability is non-zero, and $\mathsf{\Theta}$ denotes the set of all such attainable transcripts.

Offline Phase of $\mathcal{O}_{\text{ideal}}$, Initialize $\mathcal{L}_1 = \emptyset$

```
$\forall i \in [q]$ : compute $(\Sigma_i, \Theta_i) \leftarrow$ $\mathsf{Internal}^{\mathcal{L}_1}(M^i)$
    $\forall j \in [\ell_i]$ :   $X^i_j \leftarrow \langle j \rangle_s \| M^i_j$;
    if $\mathcal{L}_1(X^i_j) = \top$, then
      $\mathcal{L}_1(X^i_j) \leftarrow Y^i_j \xleftarrow{\$} \overline{\mathsf{Ran}(\mathcal{L}_1)}$;
    else $Y^i_j \leftarrow \mathcal{L}_1(X^i_j)$;
    $\Sigma_i := \mathsf{fix}_0(Y^i_1 \oplus \cdots \oplus Y^i_{\ell_i})$;
    $\Theta_i := \mathsf{fix}_1(2^{\ell_i} Y^i_1 \oplus \cdots \oplus 2^2 Y^i_{\ell_i - 1} \oplus 2Y^i_{\ell_i})$;
    return $(\Sigma_i,\ \Theta_i)$;
Let $\widetilde{\Sigma} = (\Sigma_1, \ldots, \Sigma_q), \widetilde{\Theta} = (\Theta_1, \ldots, \Theta_q)$;
if $\exists i \neq j \in [q] : (\Sigma_i, \Theta_i) = (\Sigma_j, \Theta_j)$, then $\boxed{\mathsf{Bad1} \leftarrow 1}$ , $\perp$;
if $\exists i \neq j \in [q] : (\Sigma_i, T_i) = (\Sigma_j, T_j)$, then $\boxed{\mathsf{Bad2} \leftarrow 1}$ , $\perp$;
if $\exists i \neq j \in [q] : (\Theta_i, T_i) = (\Theta_j, T_j)$, then $\boxed{\mathsf{Bad3} \leftarrow 1}$ , $\perp$;
if $\exists i_1, i_2, i_3 \in [q] : \Sigma_{i_1} = \Sigma_{i_2}, \Theta_{i_2} = \Theta_{i_3}, T_{i_1} \oplus T_{i_2} \oplus T_{i_3} = 0^n$, then $\boxed{\mathsf{Bad4} \leftarrow 1}$ , $\perp$;
if $\exists i_1, i_2, i_3, i_4 \in [q] : \Sigma_{i_1} = \Sigma_{i_2}, \Theta_{i_2} = \Theta_{i_3}, \Sigma_{i_3} = \Sigma_{i_4}$, then $\boxed{\mathsf{Bad5} \leftarrow 1}$ , $\perp$;
if $\exists i_1, i_2, i_3, i_4 \in [q] : \Theta_{i_1} = \Theta_{i_2}, \Sigma_{i_2} = \Sigma_{i_3}, \Theta_{i_3} = \Theta_{i_4}, T_{i_1} \oplus T_{i_2} \oplus T_{i_3} \oplus T_{i_4} = 0^n$,
  then $\boxed{\mathsf{Bad6} \leftarrow 1}$ , $\perp$;
$\mathcal{F}_\Sigma \leftarrow \{(i,j) \in [q]^2 : \exists i \neq j,\ \Sigma^i = \Sigma^j\}$; $\mathcal{F}_\Theta \leftarrow \{(i,j) \in [q]^2 : \exists i \neq j,\ \Theta^i = \Theta^j\}$;
if $|\mathcal{F}_\Sigma| \geq q^{2/3} \vee |\mathcal{F}_\Theta| \geq q^{2/3}$, then $\boxed{\mathsf{Bad7} \leftarrow 1}$ , $\perp$;
return $\left( (\widetilde{X}_i, \widetilde{Y}_i)_{i \in [q]}, (\widetilde{\Sigma}, \widetilde{\Theta}) \right)$;
```

Figure 7.5.1: Offline phase of the Ideal oracle $\mathcal{O}_{\text{ideal}}$: Boxed statements denote bad events. Whenever a bad event is set to 1, the oracle immediately aborts (denoted as $\perp$) and returns the remaining values of the transcript in any arbitrary manner. So, if we proceed further we can surely assume that the event $\perp$ (and so any bad event so far) does not hold. We write $\top$ when the value of a variable is not defined.

Now, we prove the security of the construction using the H-Coefficient technique 2.4.1. We need to identify the set of bad transcripts and compute an upper bound for their probability in the ideal world. Then, we need to lower bound the ratio of the real to ideal interpolation probability for a good transcript.

Remarks 7.5.1. *Note that $\overline{\mathsf{Bad}_4}$ allows the good graph to have a path of length three (an N-type graph) such that the sum of the labels of the edges of the path is non-zero. On the other hand, Bad_5 is stronger than Bad_6. Note that $\overline{\mathsf{Bad}_5}$ allows the graph to have path length at most three (an N-type graph), whereas $\overline{\mathsf{Bad}_6}$ allows the graph to have path length at most four (an W-type graph) such that the sum of the labels of the edges of the path is non-zero. The asymmetry between Bad_5 and Bad_6 arises because it is easy to bound Bad_6 with the condition $T_{i_1} \oplus T_{i_2} \oplus T_{i_3} \oplus T_{i_4} = 0^n$*

7.5.2 Definition and Probability of Bad Transcripts

In this section, we define and bound the probability of bad transcripts in the ideal world. We say that an attainable transcript τ is a **bad** transcript if any bad flags, defined in the offline phase of the ideal world as shown in Fig. 7.5.1, is set to 1. Recall that $\mathsf{BadT} \subseteq \Theta$ be the set of all attainable bad transcripts and $\mathsf{GoodT} = \Theta \setminus \mathsf{BadT}$ be the set of all attainable good transcripts. We bound the probability of bad transcripts in the ideal world as follows. Before we proceed to bound the above events in the ideal world, we state the following two lemmas that upper bounds the collision probability between two Σ (or Θ) values for two distinct queries. We emphasize that the following result will be frequently used in upper bounding the probability of the above bad events.

Lemma 7.5.2. *For distinct two messages M_α and M_β, we have*

$$(i)\ \Pr[\Sigma_\alpha = \Sigma_\beta] \leq \frac{4}{2^n}, \quad (ii)\ \Pr[\Theta_\alpha = \Theta_\beta] \leq \frac{4}{2^n}.$$

Proof. We prove only (i) as the proof of (ii) is exactly similar to (i). Suppose the number of blocks of M_α and M_β be ℓ_α and ℓ_β respectively. Without loss of generality, we assume that $\ell_\alpha \leq \ell_\beta$. Now,

$$\Sigma_\alpha = \Sigma_\beta \Rightarrow \mathsf{msb}_{n-1}\Bigg(\underbrace{\bigoplus_{i=1}^{\ell_\alpha} Y_\alpha[i] \oplus \bigoplus_{i=1}^{\ell_\beta} Y_\beta[i]}_{\mathfrak{F}}\Bigg) = 0^{n-1}. \tag{7.1}$$

For computing the probability of the above event, we consider the following three cases.

1. $(\ell_\alpha = \ell_\beta) \wedge (\exists a \in [\ell_\alpha] : X_\alpha[a] \neq X_\beta[a]) \wedge (\forall i \in [\ell_\alpha] \setminus \{a\} : X_\alpha[i] = X_\beta[i])$
2. $(\ell_\alpha = \ell_\beta) \wedge (\exists a, b \in [\ell_\alpha] : X_\alpha[a] \neq X_\beta[a] \wedge X_\alpha[b] \neq X_\beta[b])$
3. $(\ell_\alpha \neq \ell_\beta)$.

Case 1: Since $X_\alpha[a] \neq X_\beta[a] \Rightarrow Y_\alpha[a] \neq Y_\beta[a]$ and $X_\alpha[i] = X_\beta[i] \Rightarrow Y_\alpha[i] = Y_\beta[i]$, for $i \in [\ell_\alpha] \setminus \{a\}$, $\mathfrak{F} \neq 0^n$. So, the probability of $\Sigma_\alpha = \Sigma_\beta$ is $1/2^{n-1}$.

Case 2: Suppose $\exists a_1, a_2, \ldots, a_j \in [\ell_\alpha]$, $j \geq 2$ such that, for all $i \in [j]$, $X_\alpha[a_i] \neq X_\beta[a_i]$. After eliminating all the same outputs between $\{Y_\alpha[i] : 1 \leq i \leq \ell_\alpha\}$ and $\{Y_\beta[i] : 1 \leq i \leq \ell_\beta\}$, we have

$$\mathfrak{F} = \bigoplus_{i=1}^{j} \left(Y_\alpha[a_i] \oplus Y_\beta[a_i]\right).$$

Since $\mathfrak{F}$ has at most $\ell_\alpha + \ell_\beta$ outputs, the probability of $\mathfrak{F} = 0^n$ is $1/(2^n - \ell_\alpha - \ell_\beta - 1)$.

Case 3: Without loss of generality, we assume that $\ell_\alpha < \ell_\beta$. Similarly from the previous case, after eliminating the same outputs between $\{Y_\alpha[i] : 1 \leq i \leq \ell_\alpha\}$ and $\{Y_\beta[i] : 1 \leq i \leq \ell_\beta\}$, we have

$$\mathfrak{F} = \bigoplus_{i=1}^{j} Y_\alpha[a_i] \oplus \bigoplus_{i=1}^{k} Y_\beta[a_i],$$

where $a_1, \ldots, a_j \in [\ell_\alpha]$ and $b_1, \ldots, b_k \in [\ell_\beta]$. Also, by the similar argument of case 2, we have the probability of $\mathfrak{F} = 0^n$ is at most $1/(2^n - \ell_\alpha - \ell_\beta - 1)$. Hence,

$$\begin{aligned} \Pr[\Sigma_\alpha = \Sigma_\beta] &\leq \frac{2}{(2^n - \ell_\alpha - \ell_\beta - 1)} \\ &\leq \frac{4}{2^n}, \text{ assuming } \ell_\alpha + \ell_\beta \leq 2^{n-1}. \qquad \square \end{aligned}$$

Now, we are ready to bound the probability of the above bad events and hence, we bound the probability of realizing a bad transcript in the ideal world as follows:

Lemma 7.5.3 (Bad Lemma). *Let us define the event* $\mathsf{BadT} := \mathsf{Bad\text{-}Tag} \vee \mathsf{Bad1} \vee \mathsf{Bad2} \vee \mathsf{Bad3} \vee \mathsf{Bad4} \vee \mathsf{Bad5} \vee \mathsf{Bad6} \vee \mathsf{Bad7}_a \vee \mathsf{Bad7}_b$. *Let* τ' *be any attainable transcript and* X_{id} *be defined as above. Then*

$$\Pr[\mathsf{X}_{\text{id}} \in \mathsf{BadT}] \;\leq\; \frac{204q^2}{2^{2n}} + \frac{80q^4}{2^{3n}} + \frac{8\sqrt{2}q^2}{2^{3n/2}} + \frac{8}{2^n} + \frac{q}{2^n} + \frac{6\sigma}{2^n} + \frac{4q^{4/3}}{2^n} + \frac{32q^3}{2^{3n}}.$$

Proof. We upper bound the probability of individual bad events in the ideal world and then by the virtue of the union bound, we sum up the bounds to obtain the overall bound on the probability of bad transcripts in the ideal world.

1. Bound for Bad-Tag : For a fixed $i \in [q]$, the probability that $T_i = 0^n$ is exactly 2^{-n}, which follows from the uniform sampling of the output for the i-th query in the ideal world. Therefore, by varying over all possible choices for i, we have

$$\Pr[\mathsf{Bad\text{-}Tag}] = \Pr[\exists i \in [q] : T_i = 0^n] \leq \frac{q}{2^n}. \tag{7.2}$$

2. Bound for Bad1 : For a fixed $i \neq j \in [q]$, $(\Sigma_i, \Theta_i) = (\Sigma_j, \Theta_j)$ implies the following two equations:

$$\mathcal{E} = \begin{cases} \underbrace{\mathsf{msb}_{n-1}\Big((Y_i[1] \oplus \ldots \oplus Y_i[\ell_i]) \oplus (Y_j[1] \oplus \ldots \oplus Y_j[\ell_j]) \Big)}_{S_1} = 0^{n-1} \\ \underbrace{\mathsf{msb}_{n-1}\Big((2^{\ell_i} Y_i[1] \oplus \ldots \oplus 2Y_i[\ell_i]) \oplus (2^{\ell_j} Y_j[1] \oplus \ldots \oplus 2Y_j[\ell_j]) \Big)}_{S_2} = 0^{n-1}, \end{cases}$$

where ℓ_i and ℓ_j denotes the number of blocks of message M_i and M_j. We bound the probability of the above equation holds in the three disjoint cases as follows:

1. $(\ell_i = \ell_j) \wedge (\exists a \in [\ell_i] : X_i[a] \neq X_j[a]) \wedge (\forall \alpha \in [\ell_i] \setminus \{a\} : X_i[\alpha] = X_j[\alpha])$
2. $(\ell_i = \ell_j) \wedge (\exists a, b \in [\ell_i] : X_i[a] \neq X_j[a] \wedge X_i[b] \neq X_j[b])$
3. $(\ell_i \neq \ell_j)$.

Case 1: Since $X_i[a] \neq X_j[a] \Rightarrow Y_i[a] \neq Y_j[a]$ and $X_i[\alpha] = X_j[\alpha] \Rightarrow Y_i[\alpha] = Y_j[\alpha]$, for $\alpha \in [\ell_i] \setminus \{a\}$, $\bigoplus_{t=1}^{\ell_i} Y_i[t] \oplus \bigoplus_{t=1}^{\ell_j} Y_j[t] \neq 0^{n-1}$. So, the probability of $S_1 = 0^{n-1}$ is $1/2^n$ and also the probability of $S_2 = 0^{n-1}$ is $1/2^{n-1}$. Thus, the probability that satisfies equation $\mathcal{E}$ is $1/2^{2n-2}$.

Case 2: Suppose $\exists a_1, a_2, \ldots, a_p \in [\ell_i]$, $p \geq 2$ such that, for all $t \in [p]$, $X_i[a_t] \neq X_j[a_t]$. After eliminating all the same outputs between $\{Y_i[\alpha] : 1 \leq \alpha \leq \ell_i\}$ and $\{Y_j[\alpha] : 1 \leq \alpha \leq \ell_j\}$, we have

$$S_1 = \mathsf{msb}_{n-1}\left(\bigoplus_{t=1}^{p} (Y_i[a_t] \oplus Y_j[a_t])\right), \quad S_2 = \mathsf{msb}_{n-1}\left(\bigoplus_{t=1}^{p} 2^{\ell_i - a_t + 1} (Y_i[a_t] \oplus Y_j[a_t])\right). \tag{7.3}$$

Note that, there are at most $\ell_i + \ell_j$ outputs in S_1 and S_2. Therefore, the numbers of possibilities for $Y_i[a_1]$ and $Y_i[a_2]$ are at least $2^n - (\ell_i + \ell_j - 2)$ and $2^n - (\ell_i + \ell_j - 1)$ respectively. Therefore, by fixing the values to the other output variables of equations in $\mathcal{E}$, the equations in $\mathcal{E}$ provide a unique solution for $Y_i[a_1]$ and $Y_i[a_2]$. As a result, the probability that equation $\mathcal{E}$ is satisfied is at most $4/(2^n - (\ell_i + \ell_j - 2))(2^n - (\ell_i + \ell_j - 1))$.

Case 3: Without loss of generality, we assume that $\ell_i < \ell_j$. Similar to the previous case, after eliminating the same outputs between $\{Y_i[\alpha] : 1 \leq \alpha \leq \ell_i\}$ and $\{Y_j[\alpha] : 1 \leq \alpha \leq \ell_j\}$, we have

$$\begin{aligned} S_1 &= \mathsf{msb}_{n-1}\left(\bigoplus_{t=1}^{p_1} Y_i[a_t] \oplus \bigoplus_{t=1}^{p_2} Y_j[a_t]\right), \\ S_2 &= \mathsf{msb}_{n-1}\left(\bigoplus_{t=1}^{p_1} 2^{\ell_i - a_t + 1} Y_i[a_t] \oplus \bigoplus_{t=1}^{p_2} 2^{\ell_j - a_t + 1} Y_j[a_t]\right), \end{aligned} \tag{7.4}$$

where $a_1, \ldots, a_{p_1} \in [\ell_i]$ and $b_1, \ldots, b_{p_2} \in [\ell_j]$. By $\ell_i < \ell_j$, we have $\ell_j \in \{b_1, \ldots, b_{p_2}\}$ and $\ell_j \neq 1$. Since, there are at most $\ell_i + \ell_j$ outputs in S_1 and in S_2, the number of possibilities for $Y_j[b_1]$ and $Y_j[\ell_j]$ is at least $(2^n - (\ell_i + \ell_j - 2))(2^n - (\ell_i + \ell_j - 1))$. By fixing the values to the other output variables of equations in $\mathcal{E}$, the equations in $\mathcal{E}$ provide a unique solution for $Y_j[b_1]$ and $Y_j[\ell_j]$. As a result, the probability that equation $\mathcal{E}$ is satisfied is at most $4/(2^n - (\ell_i + \ell_j - 2))(2^n - (\ell_i + \ell_j - 1))$.

Therefore, we see that for each of the above case, equations in $\mathcal{E}$ holds with probability at most $4/(2^n - (\ell_i + \ell_j - 2))(2^n - (\ell_i + \ell_j - 1))$. Therefore, we have

$$\Pr[\mathsf{Bad1}] \leq \frac{4\binom{q}{2}}{(2^n - (\ell_i + \ell_j - 2))(2^n - (\ell_i + \ell_j - 1))} \leq \frac{8q^2}{2^{2n}}, \tag{7.5}$$

where the second last inequality follows due to the fact that $\ell_i + \ell_j - 1 \leq 2^{n-1}$.

3. <u>Bound for Bad2:</u> To bound the probability of the event $\mathsf{Bad2}$, for a fixed choice of indices $i \neq j \in [q]$,

$$\Pr[\Sigma_i = \Sigma_j, T_i = T_j] \overset{(1)}{=} \Pr[\Sigma_i = \Sigma_j] \cdot \Pr[T_i = T_j] \overset{(2)}{=} \frac{4}{2^n} \times \frac{1}{2^n} = \frac{4}{2^{2n}},$$

where (1) follows due to the fact that the distribution of T_i is independent over the distribution of Σ_i in the ideal world and (2) follows from Lemma 7.5.2 and from the event that $T_i = T_j$ holds with probability 2^{-n}. Therefore, by varying over all possible choices of indices, we have

$$\Pr[\mathsf{Bad2}] = \Pr[\exists i \neq j \in [q] : (\Sigma_i, T_i) = (\Sigma_j, T_j)] \leq \frac{2q^2}{2^{2n}} \tag{7.6}$$

4. <u>Bound for Bad3:</u> We bound the probability of the event $\mathsf{Bad3}$ in a similar way as we have bounded the probability of the event $\mathsf{Bad2}$. Using the exact argument as used in bounding the probability of the event $\mathsf{Bad2}$, we similarly bound the probability of the event $\mathsf{Bad3}$ and hence, we have

$$\Pr[\mathsf{Bad3}] \leq \frac{2q^2}{2^{2n}}. \tag{7.7}$$

5.<u>Bound for $\mathsf{Bad4}$:</u> To obtain the bound for $\mathsf{Bad4}$, we first define an auxiliary bad event

$$\mathsf{Aux\text{-}Bad} := Y_i[j] \in \{0^n, 0^{n-1}1\}.$$

It is easy to see that $\mathsf{Pr}[\mathsf{Aux\text{-}Bad}] \leq \frac{2\sigma}{2^n}$, if σ is the total number of blocks over all the q queries. Now, we will obtain the bound for $\mathsf{Bad4}$, assuming that the auxiliary bad doesn't occur. Suppose ℓ is the maximum number of message blocks among all the q queries. After fixing a triplet (i_1, i_2, i_3), $\Sigma_{i_1} = \Sigma_{i_2}, \Theta_{i_2} = \Theta_{i_3}$ can be represented by a system of three linear equations as follows:

$$\begin{aligned}
\Sigma'_{i_1} &= \Sigma'_{i_2} \oplus 0^{n-1}b_1 \Leftrightarrow \bigoplus_{j=1}^{t} A_{1,j} \cdot Y[j] = 0^{n-1}b_1, \\
\Theta'_{i_2} &= \Theta'_{i_3} \oplus 0^{n-1}b_2 \Leftrightarrow \bigoplus_{j=1}^{t} A_{2,j} \cdot Y[j] = 0^{n-1}b_2,
\end{aligned} \tag{7.8}$$

for some $A_{\alpha,\beta}$, b_{ij}, where $i \in [2], j \in [2]$ and $t \leq 3\ell$. The i-th row of the augmented matrix $(A|B)$ is denoted as $(A|B)_i$ and we denote the i-th row of the coefficient matrix A as A_i for $i = 1, 2$. Now, we assume that $b_1 = b_2 = 0$. If $\mathsf{Aux\text{-}Bad}$ doesn't occur then (i) A_1 contains at least three 1's, and (ii) A_2 contains at least two distinct entries and at most two 2^α for each α. Thus, A_2 is not a multiple of A_1, and hence rank of A is at least 2. For other choices of b_1, b_2 also we can also show that the rank of A is at least 2. Thus, for a fixed choice of indices $i_1, i_2, i_3 \in [q]$ as follows:

$$\begin{aligned}
&\Pr[\Sigma_{i_1} = \Sigma_{i_2}, \Theta_{i_2} = \Theta_{i_3}, T_{i_1} \oplus T_{i_2} \oplus T_{i_3} = 0^n \wedge \overline{\mathsf{Aux\text{-}Bad}}] \\
&= \Pr[\Sigma_{i_1} = \Sigma_{i_2}, \Theta_{i_2} = \Theta_{i_3} \wedge \overline{\mathsf{Aux\text{-}Bad}}] \cdot \Pr[T_{i_1} \oplus T_{i_2} \oplus T_{i_3} = 0^n] \\
&= \frac{4}{(2^n - 3\ell)(2^n - 3\ell - 1)} \times \frac{1}{2^n - 2} \\
&\leq \frac{32}{2^{3n}},
\end{aligned}$$

assuming $\ell \leq 2^{n-2} - 1$. Here we have used the facts that the distribution of $T_{i_1}, T_{i_2}, T_{i_3}$ are chosen uniformly at random and they are independent over the distribution of Σ_i

in the ideal world. Therefore, by varying over all possible choices of indices, we have

$$\Pr[\mathsf{Bad4}] \leq \Pr[\mathsf{Aux\text{-}Bad}] + \Pr[\mathsf{Bad4} \wedge \overline{\mathsf{Aux\text{-}Bad}}] \leq \frac{2\sigma}{2^n} + \frac{32q^3}{2^{3n}}. \tag{7.9}$$

6.__**Bound for** $\mathsf{Bad5}$**:**__ To obtain the bound for $\mathsf{Bad5}$, we first define an auxiliary bad event

$$\mathsf{Aux\text{-}Bad} := Y_i[j] \in \{0^n, 0^{n-1}1\}.$$

It is easy to see that $\mathsf{Pr}[\mathsf{Aux\text{-}Bad}] \leq \frac{2\sigma}{2^n}$, if σ is the total number of blocks over all the q queries. Now, we will obtain the bound for $\mathsf{Bad5}$ conditioned on the auxiliary bad doesn't happen. Suppose ℓ is the maximum number of message blocks among all the q queries. For the Σ and Θ collision, we can simply eliminate all the same input blocks. Let us denote

$$\mathsf{Bad5}_{i_1,i_2,i_3,i_4} \Leftrightarrow \Sigma_{i_1} = \Sigma_{i_2} \wedge \Theta_{i_2} = \Theta_{i_3} \wedge \Sigma_{i_3} = \Sigma_{i_4},$$

for $(i_1, i_2, i_3, i_4) \in [q]^4$. Therefore,

$$\mathsf{Bad5} \Leftrightarrow \bigvee_{(i_1,i_2,i_3,i_4)\in[q]^4} \mathsf{Bad5}_{i_1,i_2,i_3,i_4}.$$

After fixing a quadruple (i_1, i_2, i_3, i_4), $\mathsf{Bad5}_{i_1,i_2,i_3,i_4}$ can be represented by a system of three linear equations as follows;

$$\begin{aligned}
\Sigma'_{i_1} = \Sigma'_{i_2} \oplus 0^{n-1}b_1 &\Leftrightarrow \bigoplus_{j=1}^{t} A_{1,j} \cdot Y[j] = 0^{n-1}b_1, \\
\Theta'_{i_2} = \Theta'_{i_3} \oplus 0^{n-1}b_2 &\Leftrightarrow \bigoplus_{j=1}^{t} A_{2,j} \cdot Y[j] = 0^{n-1}b_2, \\
\Sigma'_{i_3} = \Sigma'_{i_4} \oplus 0^{n-1}b_3 &\Leftrightarrow \bigoplus_{j=1}^{t} A_{3,j} \cdot Y[j] = 0^{n-1}b_3,
\end{aligned} \tag{7.10}$$

for some $A_{\alpha,\beta}$, b_i, where $i \in [3]$. Suppose

$$B = \begin{bmatrix} 0^{n-1}b_1 \\ 0^{n-1}b_2 \\ 0^{n-1}b_3 \end{bmatrix}$$

Therefore, $(A|B)$ be the augmented matrix and A be the coefficient matrix of the system of equations. The i-th row of the augmented matrix is denoted by $(A|B)_i$ and the i-th row of the coefficient matrix is denoted by A_i for $i = 1, 2, 3$. We analyse the following cases depending on the B matrix as follows.

Case 1. $\underline{B \textbf{ is all zero matrix}}$. We fix (i_1, i_2, i_3, i_4) and consider the matrix A. First let us consider the case $\ell_{i_2} = \ell_{i_3}$. Now, assuming that Aux-Bad doesn't occur, we have the following four properties:

(P1) Both A_1 and A_3 contains at least three 1's. This is due to the fact that there are Σ' collisions in A_1 and A_3,

(P2) All the entries of A_2 should look like 2^β for some β,

(P3) A_2 contains at most two 2^α for each α, and

(P4) Since there is Θ collision for A_2, it contains at least two distinct elements.

It is easy to see that the above properties ensure that A_2 is not a multiple of A_1, and hence, the rank of the coefficient matrix A is at least 2. This implies that, either rank of A is 3, or $A_1 = A_3$, or $A_2 = xA_1 + yA_3$, for some nonzero values x, y. We define three cases as follows:

(a) $\mathcal{T}_1 \overset{\Delta}{=} \{(i_1, i_2, i_3, i_4) \in [q]^4 : A \text{ has rank } 3\}$,

(b) $\mathcal{T}_2 \overset{\Delta}{=} \{(i_1, i_2, i_3, i_4) \in [q]^4 : A_1 = A_3\}$,

(c) $\mathcal{T}_3 \overset{\Delta}{=} \{(i_1, i_2, i_3, i_4) \in [q]^4 : A_2 = xA_1 \oplus yA_3 \text{ for some non-zero x,y}\}$.

Case (1a): Since the matrix is full ranked, the probability of Y-variables which satisfies system of equation is bounded by $1/(2^n - t)(2^n - t - 1)(2^n - t - 2)$. So we have

$$\Pr\left[\bigvee_{(i_1,i_2,i_3,i_4)\in\mathcal{T}_1}\mathsf{Bad5}_{i_1,i_2,i_3,i_4}\right]\leq\frac{q^4}{(2^n-4\ell)(2^n-4\ell-1)(2^n-4\ell-2)}\leq\frac{8q^4}{2^{3n}},\quad(7.11)$$

as $t\leq 4\ell$.

Case (1b): To bound the probability of $\mathsf{Bad5}_{i_1,i_2,i_3,i_4}$ for $(i_1,i_2,i_3,i_4)\in\mathcal{T}_2$, we define an equivalence relation $\sim$ on $[q]^2$, where $(i_1,i_2)\sim(i_3,i_4)$ implies $A_1=A_3$ for A, which means that $\Sigma'_{i_1}=\Sigma'_{i_2}\Leftrightarrow\Sigma'_{i_3}=\Sigma'_{i_4}$. Assume that the relation $\sim$ partitions $[q]^2$ into r many subsets, namely $\mathcal{I}_1,\ldots,\mathcal{I}_r$, i.e., $[q]^2=\mathcal{I}_1\sqcup\cdots\sqcup\mathcal{I}_r$. Now, we consider the event $\Sigma'_{i_1}=\Sigma'_{i_2}$ for all $(i_1,i_2)\in\mathcal{I}_j$, $j=1,\ldots,r$, denoted by $\mathcal{F}_j$. Then, we have

$$\Pr[\mathcal{F}_j]\leq 2/2^n.$$

Therefore, we have

$$\begin{aligned}\Pr\left[\bigvee_{(i_1,i_2,i_3,i_4)\in\mathcal{T}_2}\mathsf{Bad5}_{i_1,i_2,i_3,i_4}\right] &\leq \Pr\left[\bigvee_{j\in[r]}\bigvee_{(i_1,i_2),(i_3,i_4)\in\mathcal{I}_j}\mathsf{Bad5}_{i_1,i_2,i_3,i_4}\right]\\ &\leq \sum_{j=1}^{r}\Pr[\mathcal{F}_j]\dot{\mathrm{P}}\mathrm{r}\left[\bigvee_{(i_1,i_2),(i_3,i_4)\in\mathcal{I}_j}\left(\Theta'_{i_2}=\Theta'_{i_3}\right)\middle|\mathcal{F}_j\right]\\ &\leq \sum_{j=1}^{r}\frac{2}{2^n}.\min\left\{\frac{2|\mathcal{I}_j|^2}{2^n},1\right\},\end{aligned}\quad(7.12)$$

where $\ell\leq 2^n/16$. Using the given condition $\sum_{j=1}^{r}|\mathcal{I}_j|=q^2$, $\min\left\{\frac{2|\mathcal{I}_j|^2}{2^n},1\right\}$ have maximum value when $r=\lfloor q^2/2^{\frac{n-1}{2}}\rfloor+1$ and $|\mathcal{I}_j|=2^{\frac{n-1}{2}}$, for $j=1,\ldots,r-1$ and $|\mathcal{I}_r|=q^2-(r-1)2^{\frac{n-1}{2}}$. Hence,

$$\Pr\left[\bigvee_{(i_1,i_2,i_3,i_4)\in\mathcal{T}_2}\mathsf{Bad5}_{i_1,i_2,i_3,i_4}\right]\leq\frac{2\sqrt{2}q^2}{2^{3n/2}}+\frac{2}{2^n}.\quad(7.13)$$

Case (1c): Now we consider the case $(i_1,i_2,i_3,i_4)\in\mathcal{T}_3$. Properties (P1) - (P4) ensure that (i) A_1 and A_3 intersect at most two positions and can not be disjoint, and (ii) A_2 can have at most three different elements. So, we can find a submatrix of order 3×3

$$\begin{bmatrix} 1 & 1 & 0 \\ 2^\alpha & 2^\alpha \oplus 2^\beta & 2^\beta \\ 0 & 1 & 1 \end{bmatrix},$$

where $\alpha \neq \beta$. Since all the elements of A_2 is a power of 2, there must exist some γ such that $2^\alpha \oplus 2^\beta = 2^\gamma$. We define

$$\mathsf{NEQ}_{i,j} \triangleq \{\mu \in [\min\{\ell_i, \ell_j\}] : M_i[\mu] \neq M_j[\mu]\} \sqcup \{\mu : \min\{\ell_i, \ell_j\} < \mu \leq \max\{\ell_i, \ell_j\}\}.$$

Since $xA_1 \oplus yA_3$ gives at most three nonzero elements in A_2, $\mathsf{NEQ}_{i_2,i_3} = \{\alpha, \beta, \gamma\}$. Now consider that M_{i_2} and M_{i_3} are given with $\mathsf{NEQ}_{i_2,i_3} = \{\alpha, \beta, \gamma\}$, where $2^\alpha \oplus 2^\beta \oplus 2^\gamma = 0$ and $\alpha < \beta < \gamma$. We have to find M_{i_1} and M_{i_4} such that $(i_1, i_2, i_3, i_4) \in \mathcal{T}_3$. In this scenario, A_2 is determined uniquely. After choosing distinct $x, y \in \{2^\alpha, 2^\beta, 2^\gamma\}$, A_1 and A_3 are fixed, such that $xA_1 \oplus yA_3 = A_2$. If A_2 contains every nonzero element exactly twice and if $x = 2^\alpha$ and $y = 2^\beta$, then we can find a submatrix of order 3×6

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 \\ 2^\alpha & 2^\beta & 2^\gamma & 2^\alpha & 2^\beta & 2^\gamma \\ 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

with other elements are 0's. As, there are at most two possibilities that M_{i_1} yielding A_1 and M_{i_4} yielding A_3 each, M_{i_1} and M_{i_4} can be chosen at most 24 possible ways. Therefore, we have,

$$\Pr\left[\bigvee_{(i_1,i_2,i_3,i_4) \in \mathcal{T}_3} \mathsf{Bad5}_{i_1,i_2,i_3,i_4}\right] \leq \frac{24\binom{q}{2}}{(2^n - 4\ell - 1)(2^n - 4\ell - 2)} \leq \frac{96q^2}{2^{2n}}. \qquad (7.14)$$

By considering all three sub-cases we have

$$\Pr\left[\bigvee_{(i_1,i_2,i_3,i_4)\in\mathcal{T}_1\bigsqcup\mathcal{T}_2\bigsqcup\mathcal{T}_3}\mathsf{Bad5}_{i_1,i_2,i_3,i_4}\right]\leq\frac{8q^4}{2^{3n}}+\frac{2\sqrt{2}q^2}{2^{3n/2}}+\frac{2}{2^n}+\frac{96q^2}{2^{2n}}. \tag{7.15}$$

Now we consider the case where $\ell_{i_2}\neq\ell_{i_3}$. W.l.o.g. assume that $\ell_{i_2}>\ell_{i_3}$. We observe that property (P1), (P4) remain as it is, and property (P2) gets modified to the fact that all the entries of A_2 should now look like 2^β or $2^{\ell_{i_2}-\ell_{i_3}+\beta}$ for some β. Similar to the previous analysis, this may result in three sub-cases 1a, 1b, and 1c. We can easily bound 1a and 1b identically. Now, we claim that 1c can not happen in this case. This is due to the fact that (i) The length difference between the two messages ensures that the contribution of Y-variables can not be canceled out (as the coefficients are different depending on the length of the message), (ii) one can have at least 2 different and at most 3 different entries in A_2, (iii) Both A_1, A_3, and $A_1\oplus A_3$ must contain at least 3 1's. Combining the cases, we have

$$\Pr[\mathsf{Bad5\text{-}1}\mid\overline{\mathsf{Aux\text{-}Bad}}]\leq\frac{8q^4}{2^{3n}}+\frac{2\sqrt{2}q^2}{2^{3n/2}}+\frac{2}{2^n}+\frac{96q^2}{2^{2n}}. \tag{7.16}$$

Case 2: **B is a non-zero matrix.** Let us fix (i_1,i_2,i_3,i_4). Now depending on the values of b_1,b_2,b_3 we have the cases as follows:

Case (2a): This case corresponds to $b_1=b_3=0$, and $b_2=1$. In this event, it is clear that $(A|B)_2$ can not be written as a linear combination of $(A|B)_1$ and $(A|B)_3$. So, the rank of $(A|B)$ is either 2 or 3. Thus, we have

$$\Pr[\mathsf{Bad5\text{-}2a}\mid\overline{\mathsf{Aux\text{-}Bad}}]\leq\frac{8q^4}{2^{3n}}+\frac{2\sqrt{2}q^2}{2^{3n/2}}+\frac{2}{2^n}.$$

Case (2b): This case corresponds to $b_1=b_3=1$, and $b_2=0$. In this event A_2 follows the conditions (P2)-(P4). Since A_2 contains at least 2 distinct elements and $b_1=b_3=1,b_2=0$, $(A|B)_2$ can not written as a linear combination of $(A|B)_1$ and

$(A|B)_3$. So, the rank of $(A|B)$ is either 2 or 3. Thus, we have

$$\Pr[\mathsf{Bad5\text{-}2b} \mid \overline{\mathsf{Aux\text{-}Bad}}] \leq \frac{8q^4}{2^{3n}} + \frac{2\sqrt{2}q^2}{2^{3n/2}} + \frac{2}{2^n}.$$

Case (2c): This case corresponds to $b_1 \neq b_3$, and $b_2 = 0$. In this event $(A|B)_1 \neq (A|B)_3$. Also, there exists at least one column in $(A|B)$ where the corresponding elements of A_1 and A_3 are distinct. Due to this reason $(A|B)_2$ can not written as a linear combination of $(A|B)_1$ and $(A|B)_3$. So, the rank of $(A|B)$ is 3. Thus, we have

$$\Pr[\mathsf{Bad5\text{-}2c} \mid \overline{\mathsf{Aux\text{-}Bad}}] \leq \frac{16q^4}{2^{3n}}.$$

Case (2d): This case corresponds to $b_1 \neq b_3$, and $b_2 = 1$. This is the same as **Case 2c**. Thus the probability of this event is bounded by

$$\Pr[\mathsf{Bad5\text{-}2d} \mid \overline{\mathsf{Aux\text{-}Bad}}] \leq \frac{16q^4}{2^{3n}}.$$

Case (2e): This case corresponds to $b_1 = b_2 = b_3 = 1$. In this event, any of the cases may happen among Case 1a, Case 1b and Case 1c. Thus the probability of this event is bounded by

$$\Pr[\mathsf{Bad5\text{-}2e} \mid \overline{\mathsf{Aux\text{-}Bad}}] \leq \frac{8q^4}{2^{3n}} + \frac{2\sqrt{2}q^2}{2^{3n/2}} + \frac{2}{2^n} + \frac{96q^2}{2^{2n}}.$$

Thus, summing all the above five cases, we have

$$\Pr[\mathsf{Bad5\text{-}2} \mid \overline{\mathsf{Aux\text{-}Bad}}] \leq \frac{56q^4}{2^{3n}} + \frac{6\sqrt{2}q^2}{2^{3n/2}} + \frac{6}{2^n} + \frac{96q^2}{2^{2n}}. \tag{7.17}$$

Finally, by combining all the cases, we obtain:

$$\begin{aligned}\Pr[\mathsf{Bad5}] &\leq \Pr[\mathsf{Bad5} \mid \overline{\mathsf{Aux\text{-}Bad}}] + \Pr[\mathsf{Aux\text{-}Bad}] \\ &\leq \Pr[\mathsf{Bad5\text{-}1} \mid \overline{\mathsf{Aux\text{-}Bad}}] + \Pr[\mathsf{Bad5\text{-}2} \mid \overline{\mathsf{Aux\text{-}Bad}}] + \Pr[\mathsf{Aux\text{-}Bad}] \\ &\leq \frac{64q^4}{2^{3n}} + \frac{8\sqrt{2}q^2}{2^{3n/2}} + \frac{2\sigma+8}{2^n} + \frac{192q^2}{2^{2n}}. \end{aligned} \tag{7.18}$$

7. <u>Bound for $\mathsf{Bad6}$</u>: To obtain the bound for $\mathsf{Bad6}$, we first define an auxiliary bad event

$$\mathsf{Aux\text{-}Bad} := Y_i[j] \in \{0^n, 0^{n-1}1\}.$$

It is easy to see that $\mathsf{Pr}[\mathsf{Aux\text{-}Bad}] \leq \frac{2\sigma}{2^n}$, if σ is the total number of blocks over all the q queries. Now, we will obtain the bound for $\mathsf{Bad6}$, assuming that the auxiliary bad doesn't occur. Suppose ℓ is the maximum number of message blocks among all the q queries. After fixing a quadruple (i_1, i_2, i_3, i_4), $\Theta_{i_1} = \Theta_{i_2}, \Sigma_{i_2} = \Sigma_{i_3}, \Theta_{i_3} = \Theta_{i_4}$ can be represented by a system of three linear equations as follows:

$$\begin{aligned}
\Theta'_{i_1} &= \Theta'_{i_2} \oplus 0^{n-1}b_1 \Leftrightarrow \bigoplus_{j=1}^{t} A_{1,j} \cdot Y[j] = 0^{n-1}b_1,\\
\Sigma'_{i_2} &= \Sigma'_{i_3} \oplus 0^{n-1}b_2 \Leftrightarrow \bigoplus_{j=1}^{t} A_{2,j} \cdot Y[j] = 0^{n-1}b_2,\\
\Theta'_{i_3} &= \Theta'_{i_4} \oplus 0^{n-1}b_3 \Leftrightarrow \bigoplus_{j=1}^{t} A_{3,j} \cdot Y[j] = 0^{n-1}b_3,
\end{aligned} \tag{7.19}$$

for some $A_{\alpha,\beta}$, b_α, where $\alpha \in [3]$ and $t \leq 4\ell$. The i-th row of the augmented matrix $(A|B)$ is denoted as $(A|B)_i$ and we denote the i-th row of the coefficient matrix A as A_i for $i = 1, 2$. Now we claim that if $\mathsf{Bad\text{-}Aux}$ doesn't occur, then the rank of A is at least 2, for any choice of (b_1, b_2). Thus, for a fixed choice of indices $i_1, i_2, i_3, i_4 \in [q]$ as follows:

$$\begin{aligned}
&\Pr[\Theta_{i_1} = \Theta_{i_2}, \Sigma_{i_2} = \Sigma_{i_3}, \Theta_{i_3} = \Theta_{i_4}, T_{i_1} \oplus T_{i_2} \oplus T_{i_3} \oplus T_{i_4} = 0^n \wedge \overline{\mathsf{Aux\text{-}Bad}}]\\
&= \Pr[\Theta_{i_1} = \Theta_{i_2}, \Sigma_{i_2} = \Sigma_{i_3}, \Theta_{i_3} = \Theta_{i_4} \wedge \overline{\mathsf{Aux\text{-}Bad}}] \cdot \Pr[T_{i_1} \oplus T_{i_2} \oplus T_{i_3} \oplus T_{i_4} = 0^n]\\
&= \frac{4}{(2^n - 4\ell)(2^n - 4\ell - 1)} \times \frac{1}{2^n - 3} = \frac{16}{2^{3n}},
\end{aligned}$$

assuming $\ell \leq 2^{n-2} - 1$. Note that we have used the facts that the distribution of $T_{i_1}, T_{i_2}, T_{i_3}, T_{i_4}$ are chosen uniformly at random and they are independent over the distribution of Y_i values in the ideal world. Therefore, by varying over all possible choices of indices, we have

$$\Pr[\mathsf{Bad6}] \leq \Pr[\mathsf{Aux\text{-}Bad}] + \Pr[\mathsf{Bad6} \wedge \overline{\mathsf{Aux\text{-}Bad}}] \leq \frac{2\sigma}{2^n} + \frac{16q^4}{2^{3n}}. \tag{7.20}$$

8. **<u>Bound for $\mathsf{Bad7}_a$ and $\mathsf{Bad7}_b$:</u>** We bound only the probability of the event $\mathsf{Bad7}_a$ as the analysis of bounding the probability of the event $\mathsf{Bad7}_b$ is exactly similar to that of bounding the probability of the event $\mathsf{Bad7}_a$. To bound the probability of the event $\mathsf{Bad7}_a$, we define an indicator random variable. For each $i \neq j \in [q]$, we define $\mathbb{X}_{i,j}$ which is defined as follows:

$$\mathbb{X}_{i,j} = \begin{cases} 1, & \text{if } \Sigma_i = \Sigma_j \\ 0, & \text{otherwise} \end{cases}$$

Note that, $\Pr[\mathbb{X}_{i,j} = 1] = \Pr[\Sigma_i = \Sigma_j]$ and therefore, from Lemma 7.5.2, we have

$$\Pr[\mathbb{X}_{i,j} = 1] = \frac{4}{2^n}.$$

We define another random variable $\mathbb{X} := \sum_{i,j} \mathbb{X}_{i,j}$. Therefore, we have

$$\begin{aligned} \Pr[\mathsf{Bad7}_a] &= \Pr[|\{(i,j) \in [q] \times [q] : i \neq j, \Sigma_i = \Sigma_j\}| > q^{2/3}] \\ &= \Pr[\mathbb{X} > q^{2/3}] \leq \frac{\mathbf{E}[\mathbb{X}]}{q^{2/3}} \leq \frac{4\binom{q}{2}}{2^n \cdot q^{2/3}} \leq \frac{2q^{4/3}}{2^n}. \end{aligned} \tag{7.21}$$

Using the exact argument as used in bounding the probability of the event $\mathsf{Bad7}_a$, we similarly bound the probability of the event $\mathsf{Bad7}_b$ and hence, we have

$$\Pr[\mathsf{Bad7}_b] \leq \frac{2q^{4/3}}{2^n} \tag{7.22}$$

Finally, the result follows as a sum of the probabilities of all these bad events. □

7.5.3 Analysis of Good Transcript

In this section, we lower bound the ratio of the probability of realizing a good transcript τ in the real and the ideal world. Let τ be a good transcript, where

$$\tau = \{(M_1, T_1, \widetilde{X}_1, \widetilde{Y}_1, \Sigma_1, \Theta_1), (M_2, T_2, \widetilde{X}_2, \widetilde{Y}_2, \Sigma_2, \Theta_2), \ldots, (M_q, T_q, \widetilde{X}_q, \widetilde{Y}_q, \Sigma_q, \Theta_q)\}.$$

In order to compute the real or ideal interpolation probability, let σ denote the distinct number of message blocks among all q queries. As a result of that, the ideal interpolation probability becomes $2^{-nq}/(2^n)_\sigma$.

Now, to compute the real interpolation probability, we first note that the permutation P_1 is invoked on a total of σ distinct input-output pairs and P_2 is invoked on at most $2q$ input-output pairs. Therefore, we have

$$\begin{aligned}
\Pr[\mathsf{T}_{\text{re}} = \tau] &= \Pr[\mathsf{P}_1(X^i_j) = Y^i_j, \forall i \in [q], j \in [\ell_i], \mathsf{P}_2(\Sigma_i) \oplus \mathsf{P}_2(\Theta_i) = T_i, \forall i \in [q]] \\
&= \Pr[\mathsf{P}_1(X^i_j) = Y^i_j, \forall i \in [q], j \in [\ell_i]] \cdot \Pr[\underbrace{\mathsf{P}_2(\Sigma_i) \oplus \mathsf{P}_2(\Theta_i) = T_i, \forall i \in [q]}_{\mathsf{E}}] \\
&= \frac{1}{(2^n)_\sigma} \cdot \Pr[\mathsf{E}] \qquad (7.23)
\end{aligned}$$

Therefore, it now boils down to compute a lower bound on the probability of the event E. To do this, we first consider that τ is a good transcript. As a result of it, none of the bad flags defined in the offline phase of the ideal world have been set to 1. Now, we consider the tuple $\widetilde{\Sigma} = (\Sigma_1, \Sigma_2, \ldots, \Sigma_q), \widetilde{\Theta} = (\Theta_1, \Theta_2, \ldots, \Theta_q)$ corresponding to the good transcript τ. From the two tuples $\widetilde{\Sigma}$ and $\widetilde{\Theta}$, we construct an edge labeled graph G as follows: for each $i \in [q], \Sigma_i$ and Θ_i represents the vertices of the graph and for each $i \in [q]$, we put an edge between the vertices Σ_i and Θ_i with the label of the edge being T_i. Moreover, for any $i \neq j$, if $\Sigma_i = \Sigma_j$, then we merge the corresponding two vertices into one. Similarly, for any $i \neq j$, if $\Theta_i = \Theta_j$, then we merge the corresponding two vertices into one. This will end up with an edge-labeled graph having the following properties:

1. The graph does not have any cycle of length 2, otherwise the bad event $\mathsf{Bad1}$

would have been hold true.

2. The label of an edge of any path is non-zero; otherwise bad event $\mathsf{Bad\text{-}Tag}$ would have been held true.

3. For a path of length two in the graph, the xor of the label of the edges of the path is non-zero; otherwise, the bad event $\mathsf{Bad2}$ or the bad event $\mathsf{Bad3}$ would have been held true.

4. The graph does not have any odd length cycle.

5. The graph contains a path of length three, which we call N path, such that the xor of the label of the edges of the path is non-zero; otherwise, bad event $\mathsf{Bad4}$ would have been held true.

6. The graph does not have any M-path, otherwise bad event $\mathsf{Bad5}$ would have been hold true. A pictorial description of the M path is shown in (b) of Fig. 3

7. The graph contains a W path such that the xor of the label of the edges of the path is non-zero; otherwise, bad event $\mathsf{Bad6}$ would have been hold true. A pictorial description of the W path is shown in (a) of Fig. 3

8. The last three properties ensure that the graph does not have any cycle of length 4 or above and it does not have any path of length more than 4. Hence, the graph G becomes acyclic. Therefore, G is a collection of some disjoint components.

9. Finally, due to $\overline{\mathsf{Bad7}_a}$ and $\overline{\mathsf{Bad7}_b}$, each component is of size at most $q^{2/3}$.

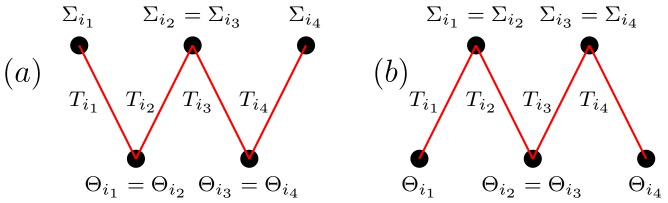
Σ_{i_1}
$\Sigma_{i_2} = \Sigma_{i_3}$
Σ_{i_4}
(a)
T_{i_1}
T_{i_2}
T_{i_3}
T_{i_4}
$\Theta_{i_1} = \Theta_{i_2}$
$\Theta_{i_3} = \Theta_{i_4}$
$\Sigma_{i_1} = \Sigma_{i_2}$
$\Sigma_{i_3} = \Sigma_{i_4}$
(b)
T_{i_1}
T_{i_2}
T_{i_3}
T_{i_4}
Θ_{i_1}
$\Theta_{i_2} = \Theta_{i_3}$
Θ_{i_4}

Figure 7.5.2: (a) represents a W-path and (b) represents a M-path.

Therefore, computing a lower bound on the probability of the event E is equivalent to computing a lower bound on the number of injective solutions which are chosen from $\{0,1\}^n$ to $\mathcal{E}_{\mathsf{G}}$. Therefore, by applying Theorem 7.3.2, we have

$$\Pr[\mathsf{E}] \geq \frac{1}{2^{nq}}\Big(1 - \epsilon_{\text{ratio}}\Big). \tag{7.24}$$

Therefore, from Equation ((7.23)) and Equation ((7.24)), we have

$$\Pr[\mathsf{T}_{\text{re}} = \tau] \quad \geq \quad \frac{1}{(2^n)_\sigma} \cdot \frac{1}{2^{nq}} \cdot \Big(1 - \epsilon_{\text{ratio}}\Big) \tag{7.25}$$

where ϵ_{ratio} is defined as follows:

$$\epsilon_{\text{ratio}} \overset{\Delta}{=} \frac{9q_c^2}{4 \cdot 2^n} + \frac{9q_c^2 q}{2^{2n}} + \frac{24q^2 q_c}{2^{2n}} + \frac{6qq_c}{2^{2n}} + \frac{40q^2}{2^{2n}} + \frac{16q^4}{2^{3n}}. \tag{7.26}$$

where q_c denotes the total number of edges in the components having a size greater than two. Since $q_c \leq q^{2/3} \leq q$, we have

$$\epsilon_{\text{ratio}} \leq \frac{9q^{4/3}}{4 \cdot 2^n} + \frac{9q^{7/3}}{2^{2n}} + \frac{24q^{8/3}}{2^{2n}} + \frac{6q^{5/3}}{2^{2n}} + \frac{40q^2}{2^{2n}} + \frac{16q^4}{2^{3n}} \tag{7.27}$$

Finally, the result follows by taking the ratio of real to ideal interpolation probability, and by combining Lemma 7.5.3 and Equation ((7.27)). □

7.6 Matching Attack on 2k-LightMAC_Plus

In this section, we show an information-theoretic distinguishing attack on the construction 2k-LightMAC_Plus based on random permutations $\mathsf{P}_1, \mathsf{P}_2$ with $2^{3n/4}$ query complexity which establishes the proven information-theoretic security bound of the construction 2k-LightMAC_Plus is tight. The distinguishing attack essentially follows a similar technique as described in [61]. Broadly speaking, we consider a computationally unbounded adversary $\mathcal{A}$ that makes a sufficient number of queries to the construction so that it satisfies a given relation $\mathcal{R}$. Once $\mathcal{A}$ gets a quadruple that satisfies the relation $\mathcal{R}$; it tries to distinguish. Note that it has been assumed that $s \leq n/4$ for the attack. Details of the attack are given as follows:

1. Perform the following for different choices of $x \leq 2^{3n/4}$:
   - (a) Make queries to the construction 2k-LightMAC_Plus on the following three inputs: (i) $0\|x$, (ii) $1\|x$, (iii) $2\|x$.
   - (b) $L[x] \triangleq \|_{i=0}^{2}\Big(\mathsf{2k\text{-}LightMAC\_Plus}(i\|x)\Big)$.
2. For each (x_1, x_2, x_3, x_4) such that $L[x_1] \oplus L[x_2] \oplus L[x_3] \oplus L[x_4] = 0^{3n}$, do the following:
   - (a) Make four additional queries to the construction 2k-LightMAC_Plus with the following inputs: (i) $3\|x_1$, (ii) $3\|x_2$, (iii) $3\|x_3$, (iv) $3\|x_4$.
   - (b) If $\bigoplus_{i=1}^{4}\mathsf{2k\text{-}LightMAC\_Plus}(3\|x_i) = 0^n$ output 1.
3. Output 0.

7.6.1 Attack Idea

Due to the presence of collisions in the fix functions in the finalization process, we can construct a matching attack by utilizing differences in Σ' and/or Θ' that are absorbed by the fix functions. Our approach involves finding a quadruple of messages $(M_1 := u\|x_1, M_2 := u\|x_2, M_3 := u\|x_4, M_4 := u\|x_4)$ such that two values collide within half of the state. Specifically, we search for quadruples that satisfy a relation

$\mathcal{R}(M_1, M_2, M_3, M_4)$ defined as:

$$\mathcal{R}(M_1, M_2, M_3, M_4) \triangleq \begin{cases} \Sigma'(M_1) = \Sigma'(M_2) \oplus 0^{n-1}1 \\ \Theta'(M_2) = \Theta'(M_3) \oplus 0^{n-1}1 \\ \Sigma'(M_3) = \Sigma'(M_4) \oplus 0^{n-1}1 \\ \Theta'(M_4) = \Theta'(M_2) \oplus 0^{n-1}1 \end{cases}$$

Note that, a quadruple (M_1, M_2, M_3, M_4) satisfies the relation $\mathcal{R}$, we must have

$$\bigoplus_{i=1}^{4} \textsf{2k-LightMAC\_Plus}(M_i) = 0^n.$$

Now, it is easy to see that our choice of messages, as shown in the attack algorithm, ensures the following:

$$\mathcal{R}(M_1, M_2, M_3, M_4) \Leftrightarrow \begin{cases} \mathsf{E}_{K_1}(\langle 2\rangle \| x_1) = \mathsf{E}_{K_1}(\langle 2\rangle \| x_2) \oplus 0^{n-1}1 \\ 2\mathsf{E}_{K_1}(\langle 2\rangle \| x_2) = 2\mathsf{E}_{K_1}(\langle 2\rangle \| x_3) \oplus 0^{n-1}1 \\ \mathsf{E}_{K_1}(\langle 2\rangle \| x_3) = \mathsf{E}_{K_1}(\langle 2\rangle \| x_4) \oplus 0^{n-1}1 \\ 2\mathsf{E}_{K_1}(\langle 2\rangle \| x_4) = 2\mathsf{E}_{K_1}(\langle 2\rangle \| x_1) \oplus 0^{n-1}1 \end{cases}$$

$$\Leftrightarrow \begin{cases} \bigoplus_{i=1}^{4} \mathsf{E}_{K_1}(\langle 2\rangle \| x_i) = 0^n \\ \mathsf{E}_{K_1}(\langle 2\rangle \| x_1) = \mathsf{E}_{K_1}(\langle 2\rangle \| x_2) \oplus 0^{n-1}1 \\ \mathsf{E}_{K_1}(\langle 2\rangle \| x_1) = \mathsf{E}_{K_1}(\langle 2\rangle \| x_4) \oplus 0^{n-1}1 \end{cases}$$

Therefore, $\mathcal{R}$ defines a $3n$-bit relation which is independent of u, so that several quadruples can be made easily that satisfy $\mathcal{R}$. Now we consider a list:

$$L = \{\textsf{2k-LightMAC\_Plus}(0\|x)\|\textsf{2k-LightMAC\_Plus}(1\|x)\|\textsf{2k-LightMAC\_Plus}(2\|x)\},$$

where $x \in [2^{3n/4}]$ and looking for a quadruples (x_1, x_2, x_3, x_4) such that $L(x_1) \oplus L(x_2) \oplus L(x_3) \oplus L(x_4) = 0^{3n}$. This leads to an attack: we look for a quadruple (x_1, x_2, x_3, x_4)

such that

$$\forall u \in \{0,1,2\},\ \bigoplus_{i=1}^{4} \mathsf{2k\text{-}LightMAC\_Plus}(u \| x_i) = 0^n.$$

We expect, on average, one random quadruple (with 2^{3n} potential quadruples and a $3n$-bit filtering), and one quadruple satisfying $\mathcal{R}$ (also a $3n$-bit condition). The correct quadruple is checked with 4 extra queries (as given in line 2(a) of the algorithm). It is easy to see that the distinguisher succeeds with probability $(1-\frac{1}{2^n})$. This is due to the fact that the probability that line 2(b) gets executed for (i) the real construction is 1, and for (ii) a random function is $\frac{1}{2^n}$.

7.6.2 Attack Complexity

It is easy to see that the number of queries made by the adversary is $\tilde{\mathcal{O}}(2^{3n/4})$. The searching required for step (iii) is done with at most $\tilde{\mathcal{O}}(2^{3n})$ operations, and using $\mathcal{O}(2^{3n/4})$ memory size (to store all the lists). We would like to point out that one can improve on the time complexity of the attack following the technique used in [61], which can report a quadruple used in line 2(a) in $\tilde{\mathcal{O}}(2^{3n/2})$ operations.

8 Conclusion

Here we provide a summary of our contributions and point out a few directions of future work.

8.1 Summary of Contributions

This thesis deals with designing of tweakable enciphering schemes based on public random permutations and also design and analysis of message authentication codes. In Chapters 3 to 7 of this thesis the main technical contributions are reported.

Though there are several tweakable enciphering schemes reported in literature there was no concrete proposal of a TES designed with public random permutations. Our work closes this gap. In Chapter 3, we design tweakable enciphering schemes built on a random permutation. We initiate the study with a generic construction of a public permutation based TES, called ppTES. Then we construct ppCTR, a public permutation based length expanding PRF and finally, we propose a single keyed and single permutation based TES which we call ppHCTR+. To the best of our knowledge, this is the first provably secure public permutation based TES. Our constructions, both ppTES and ppHCTR+ require both the forward and inverse calls of the permutation. Most existing public random permutations are more efficient in their forward calls compared to the inverse calls, thus an inverse free construction like [19, 11] is worth studying. In Chapter 4, we proposed IpTES, an inverse free tweakable enciphering scheme based on public random permutation. We proved concrete security bounds for all the constructions.

In Chapter 5, we revisited the security of TrCBC. Our study shows that TrCBC is not secure for all suggested tag lengths. In particular, for a tag length of $n/2-1$, we showed a concrete and practical attack with high success probability, which uses only three queries to the MAC. The security theorem for TrCBC, though correct, does not imply security of TrCBC for all suggested parameters. Our study re-confirms the need to study claimed security bounds in security theorems for cryptographic constructions before choosing safe parameter values for the system. We do not see any easy way to fix TrCBC such that it retains the interesting requirement of a single key and $\lceil \lambda/n \rceil$ many block cipher calls for authenticating a λ-bit message with a good security margin. It is worth mentioning here that GCBC1/GCBC2 [72] achieves this to a large extent, i.e., when $\lambda > n$ GCBC1/GCBC2 indeed produces a secure MAC with a single key and $\lceil \lambda/n \rceil$ block cipher calls.

In Chapter 6, we study block-cipher based deterministic MACs which can produce variable length tags. Specifically, in this chapter, we construct variable output length PRFs (vlPRF), and show how they can be used to construct variable tag length MACs. We also propose a modification of the PMAC scheme, called vlPMAC, to enable it to securely generate variable length authentication tags. Variable tag length block cipher based MACs are a new addition to the literature.

In Chapter 7, we have shown that the upper bound on the PRF advantage of the construction 2k-LightMAC_Plus is roughly of the order of $q^4/2^{3n}$. The bound holds when the maximum number of message blocks in a query is at most $\min\{2^{n-2}-1, 2^s\}$, and the total number of distinct message blocks across all q queries is at most 2^n. Where n denotes the block size of the block cipher and s denotes the size of the block counter. Moreover, we have also shown a matching PRF attack on the construction with query complexity roughly of the order of $2^{3n/4}$ queries. Thus, we get a tight security bound of order $3n/4$-bits for 2k-LightMAC_Plus.

8.2 Future Work

We plan to address the following problems in the near future:

1. There are extensive experimental performance data of most TESs in several computing platforms [66, 22, 19]. It would be nice to know the exact efficiency characteristic of the TES constructions reported in this thesis in both software and hardware. Particularly, schemes built on public random permutations are well-suited for lightweight applications. Thus it would be important to measure the efficiency of our schemes when implemented with constrained processors and microcontrollers. We wish to do such implementations in the future and further optimize the schemes for specific platforms based on the performance data.

2. A systematic study of implementation related vulnerabilities of existing TES including the ones reported in this thesis can be an interesting and important direction of future work.

3. Almost all TES provide birthday bound security. An important question to ask is how we can design TES to achieve security beyond birthday bound. The only work in this direction so far is [41] which achieves a beyond birthday bound security using tweakable block ciphers. Designing TES which achieves beyond birthday bound security using block ciphers or public permutations is still open and we wish to explore this direction.

4. Recently some TESs have been analyzed in the quantum setting [46]. Analyzing security of the proposed schemes against quantum adversaries would be an interesting future line of work. The structure of the Even-Mansour cipher has been used in our TES designs. It has been shown that the Even-Mansour scheme is insecure against quantum adversaries [59, 60]. It is important to analyze our TESs in light of these quantum attacks.

5. We have provide the $3n/4$-bit tight security bound of 2k-LightMAC_Plus. Proving $3n/4$-bit security of single-keyed variant of LightMAC_Plus construction

would be a direction of future work.

Bibliography

[1] William Aiello and Ramarathnam Venkatesan. Foiling birthday attacks in length-doubling transformations: Benes: a non-reversible alternative to feistel. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 307–320. Springer, 1996.

[2] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, pages 1–15, 1996.

[3] Mihir Bellare, Roch Guérin, and Phillip Rogaway. XOR MACs: New methods for message authentication using finite pseudorandom functions. In *Annual International Cryptology Conference*, pages 15–28. Springer, 1995.

[4] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences*, 61(3):362–399, 2000.

[5] Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. Gimli : A cross-platform permutation. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 299–320. Springer, 2017.

[6] Guido Bertoni, Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Farfalle: parallel permutation-based cryptography. *IACR Trans. Symmetric Cryptol.*, 2017(4):1–38, 2017.

[7] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge-based pseudo-random number generators. In Stefan Mangard and François-

Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2010.

[8] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions. In *ECRYPT hash workshop*, volume 2007. Citeseer, 2007.

[9] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 313–314. Springer, 2013.

[10] Srimanta Bhattacharya and Mridul Nandi. Revisiting variable output length XOR pseudorandom function. *IACR Cryptol. ePrint Arch.*, 2019:249, 2019.

[11] Ritam Bhaumik and Mridul Nandi. An inverse-free single-keyed tweakable enciphering scheme. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 159–180. Springer, 2015.

[12] John Black, Shai Halevi, Hugo Krawczyk, Ted Krovetz, and Phillip Rogaway. UMAC: Fast and secure message authentication. In *Advances in Cryptology—CRYPTO'99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15–19, 1999 Proceedings 19*, pages 216–233. Springer, 1999.

[13] John Black and Phillip Rogaway. CBC MACs for arbitrary-length messages: The three-key constructions. In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, volume 1880 of *Lecture Notes in Computer Science*, pages 197–215. Springer, 2000.

[14] John Black and Phillip Rogaway. A block-cipher mode of operation for parallelizable message authentication. In *EUROCRYPT 2002*, pages 384–397, 2002.

[15] Andrey Bogdanov, Miroslav Knezevic, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede. Spongent: A lightweight hash function. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*, pages 312–325. Springer, 2011.

[16] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Francois-Xavier Standaert, John Steinberger, and Elmar Tischhauser. Key-alternating ciphers in a provable setting: Encryption using a small number of public permutations. In *Advances in Cryptology – EUROCRYPT 2012*, pages 45–62. Springer, 2012.

[17] J Lawrence Carter and Mark N Wegman. Universal classes of hash functions. In *Proceedings of the ninth annual ACM symposium on Theory of computing*, pages 106–112, 1977.

[18] Debrup Chakraborty, Avijit Dutta, and Samir Kundu. Designing tweakable enciphering schemes using public permutations. *Advances in Mathematics of Communications*, 17(4):771–798, 2023.

[19] Debrup Chakraborty, Sebati Ghosh, Cuauhtemoc Mancillas López, and Palash Sarkar. FAST: Disk encryption and beyond. *Adv. Math. Commun.*, 16(1):185–230, 2022.

[20] Debrup Chakraborty, Vicente Hernandez-Jimenez, and Palash Sarkar. Another look at XCB. *Cryptography and Communications*, 7(4):439–468, 2015.

[21] Debrup Chakraborty and Samir Kundu. On the security of TrCBC. *Information Processing Letters*, 179:106320, 2023.

[22] Debrup Chakraborty, Cuauhtemoc Mancillas-López, Francisco Rodríguez-Henríquez, and Palash Sarkar. Efficient hardware implementations of BRW polynomials and tweakable enciphering schemes. *IEEE Trans. Computers*, 62(2):279–294, 2013.

[23] Debrup Chakraborty, Cuauhtemoc Mancillas-López, and Palash Sarkar. STES: A stream cipher based low cost scheme for securing stored data. *IACR Cryptology ePrint Archive*, 2013:347, 2013.

[24] Debrup Chakraborty and Mridul Nandi. An improved security bound for HCTR. In *Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers*, pages 289–302, 2008.

[25] Debrup Chakraborty and Palash Sarkar. A new mode of encryption providing a tweakable strong pseudo-random permutation. In *Fast Software Encryption: 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers 13*, pages 293–309. Springer, 2006.

[26] Debrup Chakraborty and Palash Sarkar. HCH: A New Tweakable Enciphering Scheme Using the Hash-Counter-Hash Approach. *IEEE Transactions on Information Theory*, 54(4):1683–1699, 2008.

[27] Debrup Chakraborty and Palash Sarkar. On modes of operations of a block cipher for authentication and authenticated encryption. *Cryptogr. Commun.*, 8(4):455–511, 2016.

[28] Donghoon Chang and Mridul Nandi. A short proof of the PRP/PRF switching lemma. *Cryptology ePrint Archive*, 2008.

[29] Shan Chen, Rodolphe Lampe, Jooyoung Lee, Yannick Seurin, and John Steinberger. Minimizing the two-round Even-Mansour cipher. In *Annual Cryptology Conference*, pages 39–56. Springer, 2014.

[30] Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In *EUROCRYPT 2014. Proceedings*, pages 327–350, 2014.

[31] Yu Long Chen, Eran Lambooij, and Bart Mennink. How to build pseudorandom functions from public random permutations. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, pages 266–293, 2019.

[32] Benoit Cogliati and Yannick Seurin. On the provable security of the iterated Even-Mansour cipher against related-key and chosen-key attacks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 584–613. Springer, 2015.

[33] Paul Crowley and Eric Biggers. Adiantum: length-preserving encryption for entry-level processors. *IACR Trans. Symmetric Cryptol.*, 2018(4):39–61, 2018.

[34] Yuanxi Dai, Yannick Seurin, John Steinberger, and Aishwarya Thiruvengadam. Indifferentiability of iterated Even-Mansour ciphers with non-idealized key-schedules: Five rounds are necessary and sufficient. In *Annual International Cryptology Conference*, pages 524–555. Springer, 2017.

[35] Nilanjan Datta, Avijit Dutta, and Kushankur Dutta. Improved security bound of (E/D)WCDM. *IACR Trans. Symmetric Cryptol.*, 2021(4):138–176, 2021.

[36] Nilanjan Datta, Avijit Dutta, and Samir Kundu. Tight security bound of 2k-LightMAC_Plus. *Cryptology ePrint Archive*, 2023.

[37] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Goutam Paul. Double-block hash-then-sum: a paradigm for constructing BBB secure PRF. *IACR Transactions on Symmetric Cryptology*, pages 36–92, 2018.

[38] Nilanjan Datta, Avijit Dutta, Mridul Nandi, Goutam Paul, and Liting Zhang. Single key variant of PMAC_Plus. *IACR Trans. Symmetric Cryptol.*, 2017(4):268–305, 2017.

[39] Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. Key recovery attacks on iterated Even-Mansour encryption schemes. *Journal of Cryptology*, 29(4):697–728, 2016.

[40] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2. *NIST LWC*, 2019.

[41] Avijit Dutta and Mridul Nandi. Tweakable HCTR: A BBB secure tweakable enciphering scheme. In *Progress in Cryptology - INDOCRYPT 2018 - 19th International Conference on Cryptology in India, New Delhi, India, December 9-12, 2018, Proceedings*, pages 47–69, 2018.

[42] Morris Dworkin. The CMAC mode for authentication. *Recommendation for Block Cipher Modes of Operation*, 2005.

[43] Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *J. Cryptology*, 10(3):151–162, 1997.

[44] Peter Gaži, Krzysztof Pietrzak, and Michal Rybár. The exact security of PMAC. *IACR Transactions on Symmetric Cryptology*, pages 145–161, 2016.

[45] Peter Gazi, Krzysztof Pietrzak, and Stefano Tessaro. The exact PRF security of truncation: Tight bounds for keyed sponges and truncated CBC. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 368–387. Springer, 2015.

[46] Sebati Ghosh and Palash Sarkar. Breaking tweakable enciphering schemes using Simon's algorithm. *Des. Codes Cryptogr.*, 89(8):1907–1926, 2021.

[47] Sebati Ghosh and Palash Sarkar. Variants of Wegman-Carter message authentication code supporting variable tag lengths. *Des. Codes Cryptogr.*, 89(4):709–736, 2021.

[48] Shai Halevi. EME*: Extending EME to handle arbitrary-length messages with associated data.

[49] Shai Halevi. Invertible universal hashing and the TET encryption mode. In *Advances in Cryptology-CRYPTO 2007: 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007. Proceedings 27*, pages 412–429. Springer, 2007.

[50] Shai Halevi and Phillip Rogaway. A Tweakable Enciphering Mode. In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 482–499. Springer, 2003.

[51] Shai Halevi and Phillip Rogaway. A Parallelizable Enciphering Mode. In Tatsuaki Okamoto, editor, *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 292–304. Springer, 2004.

[52] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 15–44. Springer, 2015.

[53] IEEE Security in Storage Working Group (SISWG). PRP Modes Comparison, November 2007. http://siswg.org/. IEEE p1619.2.

[54] Tetsu Iwata and Kaoru Kurosawa. OMAC: one-key CBC MAC. In Thomas Johansson, editor, *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers*, volume 2887 of *Lecture Notes in Computer Science*, pages 129–153. Springer, 2003.

[55] Ashwin Jha and Mridul Nandi. A survey on applications of H-technique: Revisiting security analysis of PRP and PRF. *Entropy*, 24(4):462, 2022.

[56] Seongkwang Kim, ByeongHak Lee, and Jooyoung Lee. Tight security bounds for Double-Block Hash-then-Sum MACs. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 435–465. Springer, 2020.

[57] Manish Kumar. Security of XCB and HCTR. In *M.Tech.(Computer Science) Thesis.* Indian Statistical Institute, Kolkata, 2018.

[58] Kaoru Kurosawa and Tetsu Iwata. TMAC: two-key CBC MAC. In Marc Joye, editor, *Topics in Cryptology - CT-RSA 2003, The Cryptographers' Track at the RSA Conference 2003, San Francisco, CA, USA, April 13-17, 2003, Proceedings*, volume 2612 of *Lecture Notes in Computer Science*, pages 33–49. Springer, 2003.

[59] Hidenori Kuwakado and Masakatu Morii. Security on the quantum-type Even-Mansour cipher. In *2012 International Symposium on Information Theory and its Applications*, pages 312–316, 2012.

[60] Gregor Leander and Alexander May. Grover meets Simon - quantumly attacking the FX-construction. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 161–178. Springer, 2017.

[61] Gaëtan Leurent, Mridul Nandi, and Ferdinand Sibleyras. Generic attacks against beyond-birthday-bound MACs. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 306–336. Springer, 2018.

[62] Moses Liskov, Ronald L. Rivest, and David A. Wagner. Tweakable Block Ciphers. *J. Cryptology*, 24(3):588–613, 2011.

[63] Eik List and Mridul Nandi. ZMAC+–an efficient variable-output-length variant of ZMAC. *IACR Transactions on Symmetric Cryptology*, pages 306–325, 2017.

[64] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2):373–386, 1988.

[65] Atul Luykx, Bart Preneel, Elmar Tischhauser, and Kan Yasuda. A MAC mode for lightweight block ciphers. In *Fast Software Encryption: 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers 23*, pages 43–59. Springer, 2016.

[66] Cuauhtemoc Mancillas-López, Debrup Chakraborty, and Francisco Rodríguez-Henríquez. Reconfigurable hardware implementations of tweakable enciphering schemes. *IEEE Trans. Computers*, 59(11):1547–1561, 2010.

[67] David A. McGrew and Scott R. Fluhrer. The Security of the Extended Codebook (XCB) Mode of Operation. In Carlisle M. Adams, Ali Miri, and Michael J. Wiener, editors, *Selected Areas in Cryptography*, volume 4876 of *Lecture Notes in Computer Science*, pages 311–327. Springer, 2007.

[68] Bart Mennink, Reza Reyhanitabar, and Damian Vizár. Security of full-state keyed sponge and duplex: Applications to authenticated encryption. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 465–489. Springer, 2015.

[69] Kazuhiko Minematsu and Toshiyasu Matsushima. Tweakable enciphering schemes from Hash-Sum-Expansion. In *Progress in Cryptology–INDOCRYPT 2007: 8th International Conference on Cryptology in India, Chennai, India, December 9-13, 2007. Proceedings 8*, pages 252–267. Springer, 2007.

[70] Yusuke Naito. Blockcipher-based MACs: Beyond the birthday bound without message length. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part III*, volume 10626 of *Lecture Notes in Computer Science*, pages 446–470. Springer, 2017.

[71] Yusuke Naito. Improved security bound of LightMAC_Plus and its single-key variant. In Nigel P. Smart, editor, *Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings*, volume 10808 of *Lecture Notes in Computer Science*, pages 300–318. Springer, 2018.

[72] Mridul Nandi. Fast and secure CBC-Type MAC algorithms. In Orr Dunkelman, editor, *Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 22-25, 2009, Revised Selected Papers*, volume 5665 of *Lecture Notes in Computer Science*, pages 375–393. Springer, 2009.

[73] Mridul Nandi and Avradip Mandal. Improved security analysis of PMAC. *Journal of Mathematical Cryptology*, 2(2):149–162, 2008.

[74] Moni Naor. A pseudo-random encryption mode. *http://siswg. org/*, 2002.

[75] Moni Naor and Omer Reingold. On the construction of pseudo-random permutations: Luby-Rackoff revisited. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 189–199, 1997.

[76] NIST. Online: https://csrc.nist.gov/projects/lightweight-cryptography.

[77] Jacques Patarin. The "Coefficients H" Technique. In *Selected Areas in Cryptography, SAC*, pages 328–345, 2008.

[78] Erez Petrank and Charles Rackoff. CBC MAC for real-time data sources. *J. Cryptol.*, 13(3):315–338, 2000.

[79] Reza Reyhanitabar, Serge Vaudenay, and Damian Vizár. Authenticated encryption with variable stretch. In *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22*, pages 396–425. Springer, 2016.

[80] Phillip Rogaway, Mihir Bellare, and John Black. SHA-3 standard. *ACM Transactions on Information and System Security (TISSEC)*, 6(3):365–403, 2003.

[81] Palash Sarkar. Efficient Tweakable Enciphering Schemes from (Block-Wise) Universal Hash Functions. *IEEE Transactions on Information Theory.*, 55(10):4749–4760, 2009.

[82] Palash Sarkar. Tweakable enciphering schemes from stream ciphers with IV. *IACR Cryptol. ePrint Arch.*, 2009:321, 2009.

[83] Haitao Song. A single-key variant of LightMAC_Plus. *Symmetry*, 13(10):1818, 2021.

[84] Wagner D. CFRG discussion on UMAC., September 2005. https://marc.info/?l=cfrg&m=143336318527073&w=2.

[85] Peng Wang, Dengguo Feng, and Wenling Wu. HCTR: A variable-input-length enciphering mode. In *Information Security and Cryptology, First SKLOIS Conference, CISC 2005, Beijing, China, December 15-17, 2005,Proceedings*, pages 175–188, 2005.

[86] Mark N Wegman and J Lawrence Carter. New classes and applications of hash functions. In *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*, pages 175–182. IEEE, 1979.

[87] Kan Yasuda. PMAC with parity: Minimizing the query-length influence. In Orr Dunkelman, editor, *Topics in Cryptology - CT-RSA 2012 - The Cryptographers' Track at the RSA Conference 2012, San Francisco, CA, USA, February 27 - March 2, 2012.*, volume 7178 of *Lecture Notes in Computer Science*, pages 203–214. Springer, 2012.

[88] Liting Zhang, Wenling Wu, Peng Wang, and Bo Liang. TrCBC: Another look at CBC-MAC. *Inf. Process. Lett.*, 112(7):302–307, 2012.

[89] Yusi Zhang. Using an error-correction code for fast, beyond-birthday-bound authentication. In *Cryptographers' Track at the RSA Conference*, pages 291–307. Springer, 2015.