# Tight Security of PMAC-type and CBC-type Message Authentication Codes

A thesis submitted to the *Indian Statistical Institute, Kolkata*
in partial fulfillment of the thesis requirements for the degree of
*Doctor of Philosophy* in *Computer Science*


Author:

Soumya Chattopadhyay

Supervisor:

Mridul Nandi


Applied Statistics Unit
Indian Statistical Institute, Kolkata
203 B. T. Road, Kolkata 700108
India


July, 2023

*Dedicated to Maa*

# Abstract

Message Authentication Codes (or MACs) are symmetric-key primitives that ensure the authenticity as well as the integrity of messages. The sender generates an authentication tag (based on a message and a secret key) which can be verified on the receiver's end. Two paradigms for building block cipher based MACs of the form Hash-then-PRP: 1) *Parallelizable* or PMAC-type, 2) *Sequential* or CBC-type. PMAC, sPMAC, PMAC1, LightMAC etc. are examples of PMAC-type MACs. Whereas OMAC, XCBC, TMAC, GCBC are examples of CBC-type MACs. Obtaining *length independent (tight) bounds* for these constructions has been a challenging problem. The goal of this thesis is to obtain length independent (tight) bounds for as many important constructions as possible and devise a novel technique that can be employed for various constructions and has a scope of generalization.

PMAC-TYPE MACS: Firstly, in chapter 3, we demonstrate why a claim about tight security of a PMAC variant proposed by Naito is wrong. Together with that, we state a necessary and sufficient condition to correctly establish that claim. Secondly, in the same chapter, we propose a variant of PMAC1 which has tight security for a reasonable range of message lengths. Then we prove the tight security of sPMAC for a weaker notion of independence (of hash). Next, in chapter 4, we analyze security bounds for LightMAC: We show tight security of 1k-LightMAC (single-key version of the original LightMAC) which holds for a range of lengths (both upper and lower bounded). Moreover, we show an attack on 1k-LightMAC for sufficiently small-length messages. Besides we propose two new variants of 1k-LightMAC, namely, LightMAC-swp and LightMAC-ds, both of which achieve length independent tight security for a fairly good range of lengths. Here we employ a novel sampling technique, dubbed "Reset-sampling", as a subroutine of H-coefficient setup. It helps get tight bounds. Then, in the last chapter (5) of this part, we try to get a generalized view of the PMAC family. We develop technical concepts necessary to cover a large class of parallelizable MACs of the form hash-then-PRP. As the main results of this chapter, we prove the security bound in terms of the collision probability of the underlying hash function, both for independent keys and single-keyed versions of a generic member of the PMAC family. As a corollary to this, we apply this result to get birthday-bound security for a simplified version of PMAC+, under some assumptions. Moreover, a similar bound for 1k-LightMAC as well follows directly from the main result.

CBC-TYPE MACS: In chapter 6, we obtain $O(\frac{q^2}{2^n} + \frac{q\ell^2}{2^n})$ bound for OMAC using reset-sampling . This is the best-known bound for it. Although it is not "length independent" in an exact sense, it *behaves* almost like a birthday bound with some consideration. We obtain similar bounds for XCBC and TMAC also. In this way, we become successful in establishing tight security for all CBC-MAC variants, except the original one.

# List of Publications

This thesis is based on the following research papers:

1. Bishwajit Chakraborty, Soumya Chattopadhyay, Ashwin Jha, and Mridul Nandi: *On Length Independent Security Bounds for the PMAC Family*, In IACR Transactions on Symmetric Cryptology, 2021(2): 423–445, 2021. https://doi.org/10.46586/tosc.v2021.i2.423-445

2. Soumya Chattopadhyay, Ashwin Jha, and Mridul Nandi: *Fine-tuning the ISO/IEC Standard LightMAC*, In Advances in Cryptology – ASIACRYPT, 2021, Proceedings: 490–519, 2021. https://doi.org/10.1007/978-3-030-92078-4_17

3. Soumya Chattopadhyay, Ashwin Jha, and Mridul Nandi: *Towards Tight Security Bounds for OMAC, XCBC and TMAC*, In Advances in Cryptology – ASIACRYPT 2022. https://doi.org/10.1007/978-3-031-22963-3_12

# Acknowledgements

There are many people I would like to thank for contributing to this thesis and to what I am today. In the following lines, I will not be able to mention the names of all of them. However, I express my gratitude to all the people who have helped me develop a taste in Mathematics and Music and my passion for solving real-life problems in this world.

I am grateful to my parents for supporting me throughout my career, making almost zero objection to the choices of paths I have taken so far.

I can not restrain myself from mentioning a few names. KC (Khitish Chattopadhyay) was the first person who invited me to the world of Mathematics. BKB (Binod Kumar Berry) was a teacher in my college whom I remember as a white-bearded man creating magic in front of the blackboard. I still remember and miss my teachers and classmates of MMath at ISI Kolkata.

Regarding my PhD, Mridul da (Prof. Mridul Nandi) is the first and foremost person to be named. Without him, this thesis would have never seen the light of day. All of his scoldings were a blessing and will remain as a cherishable memory of an extremely caring teacher and advisor. I would like to mention two more names in this regard: Mama and Ashwin. Mama aka Bishwajit was my colleague at ISI who motivated and actively stood by me during a crucial juncture of my PhD days. Ashwin was my senior at ISI. He is one of the most sincere and responsible persons in the field of Academics I have seen so far.

It has been a memorable journey from MMath to PhD at ISI Kolkata: Anindya, Anuj, Ritam da, Suprita, Abhishek, Samir, Anik, Arghya, Chandranan, Rahul, Abhinandan da, Dyuti, Soumen, Satyaki, Kushankur, Sourav, Sanjay da, Butu da, Srimanta da, Debrup da, Arijit (Bishnu) da... the list is quite long and far from being exhaustive. I thank you all.

I would like to express my gratitude to all the workers of ISI without whom no Institute can ever survive.

Lastly, the persons who have contributed to every aspect of my life, shaping the world outlook, and pouring a holistic attitude towards life: my most loving friends and comrades. Solidarity forever.

*Soumya Chattopadhyay*
*Kolkata*
*June, 2022*

# Contents

# List of Figures

# List of Tables

*"We both step and do not step in the same rivers. We are and are not."*

– Heraclitus (circa 535-475 BC)

# Part I

# Background

# Chapter 1

# Introduction

## 1.1 Cryptology: A Brief Note on its Past and Present

It is often said that human civilization has been concerned with "secure" communication for ages. However, it is not entirely true. Most of the time in history, it has been a *direct* concern for mainly those who held *power*, certainly not the majority of people. The insecurity of losing political power compelled the rulers of a society to take the question of "secure" exchange of information seriously. In ancient ages, Kings had promoted the knack of "code-making" (as well as "code-breaking") which was nothing but a nascent form of Cryptology.

However, it is only very recently that Cryptology has emerged as a scientific discipline for studying *secure communication through insecure channel*. The journey from Classical Cryptology to Modern Cryptology attained a qualitative leap just around the middle of the last century. A huge range of factors is behind this development. Not delving much into that, we can certainly say that the modern aspect of worldwide communication has made the question of "secure" communication a universal concern unlike before. Moreover, with the advent of the Internet and modern communicative capitalism, a complex relationship between the interests of the state and its citizens (and also among different groups of citizens) regarding the question of *security/ secrecy* has arisen. The question of security has manifested itself in many antagonistic forms like in the debate of *surveillance vs privacy* [27]. We can find a detailed history of the development of Cryptology in [56] – tracing its long journey from the ancient past to the present time of Internet.

Although sometimes, the two terms *Cryptography* and *Cryptology* are used interchangeably, Cryptology is actually a more general term [57]. It, as a discipline, contains two

complementary sub-disciplines: *Cryptography* and *Cryptanalysis*. Cryptography deals with designs of schemes for secure communication, whereas, Cryptanalysis deals with analyzing *attacks* on those schemes.

The modern journey of Cryptology started only after Claude Shannon laid the foundations of the area of information theory and cryptology in proper mathematical terms and rigor. Modern Cryptology is based on mathematical concepts like probability, complexity theory, number theory, etc. Shannon's two papers, named *A Mathematical Theory of Communication* [88] and *Communication Theory of Secrecy Systems* [89], in 1948–49, were pioneering in the area of *Modern Cryptology*. Together with these two works, another seminal paper by Diffie and Hellman needs to be mentioned. In their paper *New Directions in Cryptography* [31], published in 1976, they made some new identifications or security goals like *integrity, authenticity* etc. These new aspects add to the previously existing dimensions of *confidentiality.*

There are two paradigms for modern cryptographic schemes [92]: *Symmetric-key* (or secret key) and *Assymetric-key* (or public key). In symmetric-key setting, a common secret key is shared beforehand. The encryption and decryption of the messages (or plaintexts) use this shared key. Whereas in a public key setting, each communicating party holds different keys for the encryption and decryption processes. Except for the prerequisite of some *key-sharing protocol*, symmetric-key schemes are much faster than the public ones in a computational sense. However, these two paradigms are "complimentary" in a way since these initial key-sharing protocols required for any symmetric-key setting do not follow the symmetric-key paradigm itself. Instead, they are part of the public key cryptographic world. Thus most of the modern real-world communication protocols are a mixing of both symmetric and public key protocols.

In this thesis, we will be confined to the symmetric-key cryptographic setting, without being bothered about initial key-sharing protocols.

## 1.2 Provable Security in symmetric-key Cryptography

SECURITY OBJECTIVES: *Data confidentiality* (or privacy) and *data authenticity* (and a related idea of data integrity) are the basic objectives of any cryptographic scheme. Confidentiality ensures that a third party does not be able to eavesdrop when there is communication between two parties (through an insecure channel). Whereas, data authenticity (or integrity) ensures that one party after recieving a secret information from another party gets able to verify whether she has got the information from the authentic sender (or the secret information has not been tampered midway). *Encyption schemes*

like CTR [79], CBC encryption [79], OCB encryption [59, 85, 86], CMC [43], EME [44], HCTR [94], HEH [87] etc. are designed to fulfil the purpose of confidentiality. *Message Authentication Codes* or MACs like CBC family [5, 9, 10, 15, 37, 47, 60, 97], ECBC [84], FCBC [84], EMAC [84], PMAC variants [16, 39, 85, 98, 99], XMACR [6], PCS [12], LightMAC [62], LightMAC+[69] etc ensure data authenticity as well as integrity. *Authenticated encryption schemes* like OCB family [59, 85, 86], GCM [63], CCM [80], COLM [2] etc. ensure both confidentiality and authenticity.

PROVABLE SECURITY: Defining these security objectives in a precise mathematical way (with respect to a probabilistic game-playing model) and obtaining security bounds as mathematical proofs constitute the domain of operation of *provable security.*

In general, any cryptographic scheme consists of two things: a *primitive* and a *mode of operation*. Blockciphers such as AES [77] which operate on short (and fixed length) messages are used as primitives. Security results of the primitives are proven heuristically, mostly depending upon its analysis over a long period of time. Modes of operation like CBC or PMAC are used to accodomodate messages with variable length. Any mode of operation is based upon a primitive. Security results for modes of operation are *information-theoretic* results, i.e., an adversary is allowed to have unbounded amount of time.

## 1.3 Message Authentication Codes

Message Authentication Codes (or MACs) are symmetric-key scheme, which ensure message authenticity as well as integrity. A MAC scheme $\mathcal{M}$ is a pair of two algorithms:

- $\mathcal{M}^+$ – an algorithm which generates an authentication tag
- $\mathcal{M}^-$ – an algorithm which verifies the authenticity of a tag

The basic principle of a MAC scheme $\mathcal{M}$, instantiated with a secret key $K$, is as follows:

Whenever the sender wants to send a message $m$ to the receiver, she sends $(m, t)$, where the *authentication tag* $t$ is the output of the tag generating algorithm, i.e., $t = \mathcal{M}^+(K, m)$. When the receiver receives a pair $(m', t')$ of message and tag, she runs the verifying algorithm $\mathcal{M}^-(K, m', t')$, which checks the equality $t' =_? \mathcal{M}^+(K, m')$. See section 2.2.3 of chapter 2 for a formal definition of MAC.

There are several ways to construct MAC schemes. *Non-deterministic* MACs are constructed using *nonces* (an extra input other than the message and key). In deterministic MAC schemes, no nonce is used. A detailed discussion on this can be found in [51].

NONCE BASED MACS: Constructions like Wegman-Carter (WC) MACs [96], WMAC [14], XMACC [6], EWCDM [26], DWCDM [30], nEHtM [36] etc, are examples of *stateful* MACs. Nonce repetition is either strictly non-permissible or permissible in a restricted way in these cases. For examples, security of WC MAC is compromised when a nonce repeats, but in case of EWCDM or DWCDM, beyond birthday bound security is achived for unique nonce and birthday bound security for repeating nonce. In a different style, in some constructions such as XMACR [6], MACRX [7], RMAC [50], FRMAC [49], EHtM [66], RWMAC [66] etc, nonce values are sampled at random for each invocation of the algorithm. One thing is common in all these constructions: block cipher (or, keyed family of random permutations) is used as the underlying primitive. However, this is not the whole scenario. Some nonce based constructions use *public permutations* as their building blocks also.

MACS BASED ON PUBLIC PERMUTATIONS: In [34], Dutta and Nandi discussed why using block cipher as primitive is not the only option for building nonce based MACs. They replaced the block cipher in nEHtM with a public (random) permutation and dubbed the new constructions as nEHtM$_p$. They also showed that nEHtM$_p$ achieves beyond birthday bound security.

In this thesis, we will study block cipher based deterministic MAC schemes of the form *hash-then-PRP*. We will focus on two paradigms covering a large class of these kinds of MACs:

- *Parallelizable* or PMAC-type MACs
- *Sequential* or CBC-type MACs

Before entering into the discussion on these two types, we mention some other deterministic schemes of block cipher based MACs.

BLOCK CIPHER BASED MACS NOT OF THE FORM HASH-THEN-PRP: A good number of deterministic, block cipher based MAC schemes fall under the category of *Double Block Hash-then-Sum* or DbHtS [29]. Constructions like Sum-ECBC [97], PMAC+[98], 3kf9 [100], LightMAC+[69] had been proved to be beyond birthday bound secure DbHtS constructions by Datta et al. [29] and Kim et al. [58]. However, recently Shen et al. [90] has enhanced these results and made the beyond birthday bound securities valid also in case of multi-users.

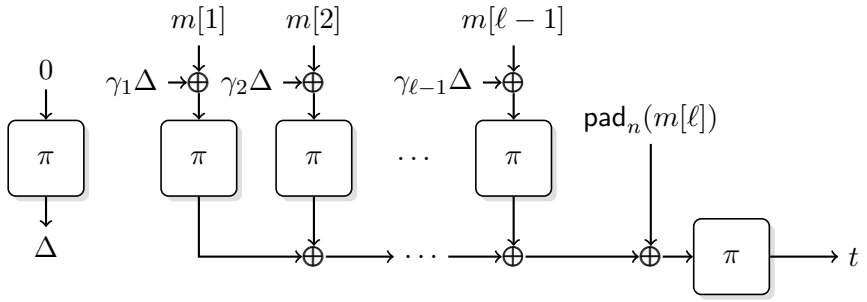


**Figure 1.3.1:** PMAC: Here, $\gamma_1, \gamma_2, \ldots$ is the sequence of Gray code and $\pi$ is a keyed block cipher (or a random permutation when we analyze the construction).

### 1.3.1 PMAC-type MACs

PMAC VARIANTS: The majority of block cipher based MACs iterate a block cipher in a sequential manner (e.g., Cipher Block Chaining or CBC)[9, 11, 15, 37]. On the other hand, PMAC, introduced by Black and Rogaway [16], is a parallelizable block cipher based MAC. A slightly simplified version of PMAC based on an $n$-bit block cipher $e_K$ can be described as follows (also illustrated in Fig. 1.3.1):

$$\mathsf{PMAC}_K(m_1, \ldots, m_\ell) := e_K\big(e_K(m_1 \oplus \gamma_1 \cdot \Delta) \oplus \cdots \oplus e_K(m_\ell \oplus \gamma_\ell \cdot \Delta)\big)$$

where $m_1, \ldots, m_\ell$ are $n$-bit elements, called blocks. The sequence of constants $\gamma_1, \gamma_2, \ldots$ is known as the Gamma code. Due to its full parallel nature, PMAC can possibly outperform sequential block cipher based MACs significantly under parallel implementation. We believe it is worth re-evaluating PMAC-type constructions under the current trend of "parallelizable" (pipeline, super-scalar, vector, and multi-core) CPUs. A forthcoming vectorized AES instruction [32] can make AES-based parallel design much faster.

The designers of PMAC proved an upper bound of $\sigma^2/2^n$ on the PRF-advantage for any adversary making a total of $q$ queries, each of length at most $\ell$ blocks (of $n$-bits), and a total of $\sigma \leq \ell q$ blocks. This was later improved to $\ell q^2/2^n$ by Minematsu and Matsushima [67], and then to $q\sigma/2^n$ by Nandi and Mandal [76]. Recently, Luykx et al. [61] showed that one can construct a pair of messages which will collide with probability roughly $\ell/2^n$, leading to a distinguishing attack with advantage $\ell/2^n$ for $q = 2$. Later Gaži et al. [40] constructed a $q$-query distinguishing attack for PMAC with advantage $\ell q^2/2^n$. However, this attack requires a large coset in the order of $\ell$ in $\ell$ consecutive masking elements. They have shown that the Gray code (used in the original PMAC) contains such a large coset. However, an existence of such large cosets for other types of masking (e.g., xtimes based masking in PMAC 1 [85]) is still an open problem.

LIGHTMAC: Lightweight cryptography endeavors to safeguard communications in resource-constrained environments. The advent of Internet of Things has given a great

impetus to this field of research in the last decade or so. As a result, several standardization efforts have tried to systematize the field, most notably the CAESAR competition [20], NIST lightweight cryptography standardization project [78], and the ISO/IEC standardization [1]. Specifically, the ISO/IEC-29192-6:2019 standard [1] specifies three message authentication code (or MAC) algorithms for lightweight applications. MACs are symmetric-key primitives that achieve data authenticity and integrity. The ISO/IEC standard recommends LightMAC [62], Tsudik's keymode [93] and Chaskey-12 [68] as the three MAC algorithms. In this paper, we focus on LightMAC.

LightMAC, by Luykx et al. [62], is a parallelizable block cipher-based MAC. For an $n$-bit block cipher $\mathsf{E}$ instantiated with keys $K_1$ and $K_2$, and a global parameter $s < n$, a simplified[1] version of LightMAC can be defined as:

$$\mathsf{LightMAC}_{K_1,K_2}(m) := \mathsf{E}_{K_2}(\mathsf{E}_{K_1}(x[1]) \oplus \cdots \oplus \mathsf{E}_{K_1}(x[\ell-1]) \oplus m[\ell] \| 10^{s-1}), \tag{1.1}$$

where $(m[1], \ldots, m[\ell])$ denotes the $(n-s)$-bit parsing of the input message $m$, and $x[i] = \langle i \rangle_s \| m[i]$ for $1 \leq i \leq \ell - 1$, where $\langle i \rangle_s$ denotes the $s$-bit binary representation of $x$. For obvious reasons $s$ is also called the counter size. The counter-based encoding in LightMAC is inherited from some earlier MAC designs such as the XOR MACs by Bellare et al. [6] and Bernstein's protected counter sums [12]. The use of counter-based encoding limits the *rate*—ratio of the number of $n$-bit blocks in the message $m$ to the number of block cipher calls required to process $m$. For example, LightMAC requires $4$ calls to process a message of length $3n$ bits when the counter size $s = n/4$, whence the rate is $3/4$. Ideally, the rate should be as high as possible, with rate $1$ or higher considered as holy grail. Dutta et al. [35] give optimal counter-based encoding strategies for some scenarios, resulting in significant speed-up. However, LightMAC still falls short on this account when compared to some other MAC schemes such as OMAC [47] and PMAC [16] etc.

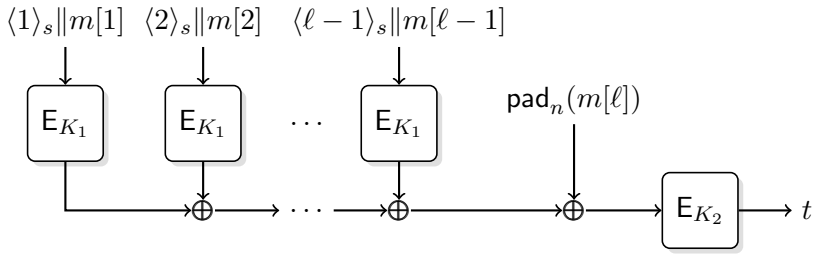


**Figure 1.3.2:** LightMAC evaluated over an $\ell$-block padded message $m$.

This construction and its security analysis are very similar to that of PCS [12]. These constructions can be viewed as the hash-then-PRP or hash-then-PRF composition [18] and hence the PRF-advantage is bounded by the collision probability of the underlying

[1] assuming all messages have length $(n-s)r$ for some $1 \leq r \leq 2^s$.

hash. The designers of LightMAC (similar to the analysis of PCS) have proved that underlying hash of the construction has about $1/2^n$ collision probability for any pair of messages. This would prove the PRF-advantage is about $q^2/2^n$ which is easily seen to be tight. However, the above composition cannot be applied when both underlying hash and the final block cipher call use same key. However, following the analysis of [75], the single key variant (i.e. $\mathsf{LightMAC}_{K,K}$) can be shown to have $\sigma q/2^n$ or $\ell \cdot q^2/2^n$ PRF-advantage.

However, LightMAC design is quite simple as it minimizes all auxiliary operations other than the block cipher call, which reduces the overhead to a minimum. For this reason, LightMAC is expected to have more compact implementations as compared to PMAC. Further, LightMAC is parallelizable like PMAC which enables it to exploit the parallel computing infrastructure, whenever available. As a result, LightMAC is a quite flexible algorithm, as it has qualities suitable for both memory-constrained environments as well as high performance computing.

### 1.3.2 CBC-type MACs

Given an $n$-bit block cipher $E$ instantiated with a key $K$, the CBC-MAC construction is defined recursively as follows: For any $x \in \{0,1\}^n$, $\mathsf{CBC}_{E_K}(x) := E_K(x)$. For all $m = (m[1], \ldots, m[\ell]) \in (\{0,1\}^n)^\ell$ where $\ell \geq 2$,

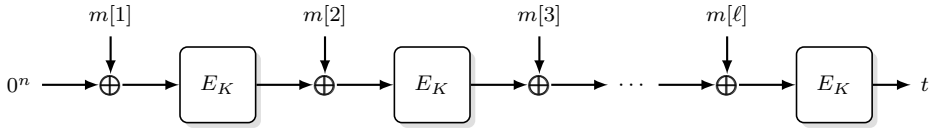


**Figure 1.3.3:** Evaluation of $\mathsf{CBC}_{E_K}$ function over an $\ell$-block message $m$..

we define

$$\mathsf{CBC}_{E_K}(m) := E_K(\mathsf{CBC}_{E_K}(m[1], \ldots, m[\ell-1]) \oplus m[\ell]) \tag{1.2}$$

For constructions like ECBC, FCBC and EMAC, Pietrzak [84] showed a bound of $O(q^2/2^n)$ for $\ell < 2^{n/8}$. Later, Jha and Nandi [52] discovered a flaw in the proof of the earlier bound and showed a bound of $O(q/2^{n/2})$ up to $\ell < 2^{n/4}$. However, in these constructions an extra (independent) block cipher is called at the end. Considering the number of block cipher calls, XCBC [15, 67], TMAC [60, 67] and OMAC [47, 74] are better choices (see chapter 6 for more details). XCBC uses two independent masking keys for the last block which are used depending on whether the last block is padded or not. In case of TMAC, these two keys are not independent but one is derived from the other. OMAC is much better in this respect. Here both keys are derived from the block cipher only. Classical bound for these constructions was $O(\sigma^2/2^n)$ [15, 60], $\sigma$ being

the total number blocks among all messages. Later in a series of work [48, 67, 74, 76] the improved bounds for XCBC, TMAC, PMAC and OMAC are shown in the form of $O(q^2\ell/2^n)$, $O(\sigma^2/2^n)$ and $O(\sigma q/2^n)$.

## 1.4 Outline of the Thesis: Motivation and Contributions

LENGTH INDEPENDENT SECURITY BOUNDS: It is well-known [8, 41, 75] that variable input length (VIL) pseudorandom functions (or PRFs) are good candidates for deterministic MACs. Indeed, almost all the security bounds on deterministic MAC schemes, in fact, quantify their PRF security. In the following discussion $q$ and $\ell$ denote the number of queries and an upper bound on query-length, respectively.

In applications where we process large messages or where most of the messages are of lengths much smaller than $\ell$, a bound of the form $O(q^2/2^n)$ (length independent) is much desired, as compared to say a bound of $O(\ell q^2/2^n)$. However, "length independence" in absolute sense is not achievable in these cases. We can at most obtain bounds where $\ell$ terms can be dropped for suitable ranges of $\ell$. *Key motivation* of this thesis is to have tight bounds (which are "length independent" in this sense) for as many important PMAC-type and CBC-type MAC constructions as possible. Moreover, another motivation is to develop a general strategy of getting these kinds of bounds. Chapterwise motivation and contributions are given below.

### 1.4.1 PMAC Variants

MOTIVATION: With respect to designs like PMAC, Gaži et al. [39] proved $O(q^2/2^n)$ bound for a simplified variant of PMAC, called sPMAC, albeit with comparatively expansive masking methods. For example, the masking function should be a 4-wise independent function. Most efficient algebraic instantiations of such a function require at least four keys and several field multiplications. Very recently, Naito [70] proposed a variant of PMAC1, which uses two powering-up maskings (instead of one used in PMAC1). He showed $O(q^2/2^n)$ advantage provided $\ell \leq 2^{n/2}$.

OUR CONTRIBUTIONS: Our contributions are threefold –

1. REVISITING NAITO'S VARIANT OF PMAC1: As of now, Naito's PMAC1 variant [70], sometimes also referred as NPMAC in this paper, is the only known rate-1 PMAC-like construction that achieves $\ell$-free security bound (for $\ell < 2^{n/2}$). We

show that *the security analysis of this construction is incorrect* (see Section 3.2). Further, we state an equivalent problem which must be solved to prove the $\ell$-free security of this construction. However, we are not able to solve that equivalent problem. So the exact security of Naito's variant is still an open problem. Naito subsequently updated the construction [72] in light of our observations. This updated variant achieves $\ell$-free security for $\ell < 2^{n/2}$ (see Section 3.3).

2. RELAXING THE SECURITY PRECONDITION FOR sPMAC: In [39], sPMAC is shown to have $\ell$-free security bound up to $\ell < 2^{n/2}$ when the underlying masking function is 4-wise independent hash. We *relax the* 4*-wise independence condition to* 2*-wise almost XOR universality* (see Section 3.3).

3. PMAC2 – A SIMPLE VARIANT OF PMAC1: As we still lack of an $\ell$-free secure PMAC variant with efficient masking function, our next part is aimed to solve this problem. We propose a simple variant of PMAC1, called PMAC2, and we show almost tight security $O(q/2^{n/2})$ (see Table 1.4.2). More precisely, we prove the following theorem (in Section 3.5).

   **Security Analysis of PMAC2**: Let $\ell$ denote the number of blocks present in the longest query and $\sigma$ denotes the total number of blocks present in $q$ queries altogether. Then,

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{PMAC2}}(q, \ell, \sigma) \leq \frac{2q^2 + \sigma}{2^n} + \mu$$

   where $\mu \leq \frac{q}{2^{n/2}}$ if $\ell \leq 2^{n/4}$ and $\mu \leq \frac{\sigma^{1.5}}{2^n}$ if $2^{n/4} < \ell \leq 2^{n-2}$.

**Table 1.4.1:** A comparative summary of several PMAC variants. Here $q$ denotes the number of queries, $\ell$ denotes the upper bound on query-length, and $\sigma$ denotes the upper bound on total number of blocks present in all queries.

| Mode | Security bound | Length restriction | Number of masking keys |
|---|---|---|---|
| PMAC [16] | $q^2\ell/2^n$ | - | 1 |
| PMAC1 [85] | $q^2\ell/2^n$ | - | 1 |
| NPMAC[1] [70] | $q^2/2^n$ | $\ell < 2^{n/2}$ | 2 |
| PMAC3 [72] | $q^2/2^n$ | $\ell < 2^{n/2}$ | 3 |
| PMAC2[Section 3.5] | $q/2^{n/2}$ | $\ell \leq 2^{n/4}$ | 1 |
| | $\sigma^{1.5}/2^n$ | $2^{n/4} < \ell \leq 2^{n-2}$ | 1 |

[1] The security analysis of this construction is shown to be incorrect in this paper.

**Publication status:** Chapter 3 is based on our paper [22], published in *IACR Transactions of Symmetric Cryptology 2021(2)*.

### 1.4.2 LightMAC and its single-key variants

MOTIVATION: ISO standards are widely used in communication protocols such as TLS, Bluetooth protocol, Zigbee etc. Being an ISO standard for lightweight cryptography, LightMAC is also widely recognized as a suitable MAC candidate for deployment in resource-constrained environments. Possibly, its simple and compact design and query-length independent security are the main reasons behind this perception. On a closer look, we see that the two independent keys greatly simplify the security argument of LightMAC. Due to the independence of keys, it can be viewed as an instance of the Hash-then-PRF paradigm [9, 95], and hence the PRF security bound follows directly from LightMAC output collision probability.

However, maintaining two block cipher keys could be a burden in memory-constrained environments. Currently LightMAC with 2 keys requires 256 bits for key (128-bit block cipher key). Instead, one-key variants of LightMAC use 128 bits, which is a significant optimization in memory footprint both in hardware and software. The problem is further aggravated when implementations store precomputed round keys to reduce latency. For example, in case of AES128 [77], this precomputation would require 176 bytes of memory per key. This motivates us to look into the problem of minimizing the number of keys in LightMAC, while maintaining the query-length independence. Specifically, we ask the following question:

† : *Is there a single-key LightMAC variant which achieves similar query-length independent bounds as two-key LightMAC?*

As it turns out, the answer to this question is not straightforward. Recall the description of LightMAC from Eq. (1.1). Let $y_i := \mathsf{E}_{K_1}(x_i)$ and $y^{\oplus} := y_1 \oplus \cdots \oplus y_{\ell-1} \oplus m_\ell \| 10^{s-1}$. We call $x_i$ and $y_i$ the $i$-th intermediate input and outputs, respectively and $y^{\oplus}$ and $t = E_{K_2}(y^{\oplus})$ the final input and output, respectively. There are two non-trivial bottlenecks (see section 4.1.2) in answering the above questions:

1. Collisions between intermediate input and final input, and
2. Collisions between intermediate output and final output.

The naive way to handle these two cases is to bound the probability of these events to $O(q^2\ell/2^n)$ as there are at most $q\ell$ intermediate inputs/outputs and $q$ final inputs/outputs. Clearly, this naive approach leads to a degradation in the security. So,

⋆ : *we need a more sophisticated strategy to prove the security of single-key LightMAC.*

Yet another approach is to explicitly separate the final inputs from intermediate inputs by fixing some input bit to 0 in intermediate inputs and 1 in final inputs. This will help in resolving the first bottleneck. However, the second bottleneck is still present. Hence, the resulting construction is not as straightforward as two-key LightMAC. Further, domain separation also introduces slight changes in the standardized design, which is not appreciated by end-users, in general. So,

$\star\star$ : *variants with very small modifications over the original LightMAC algorithm will be preferred.*

In this paper, we aim to answer † in affirmative using $\star$ and $\star\star$ as general guidelines.

OUR CONTRIBUTIONS: Our contributions are twofold:

First, in section 4.2, we study the single-key LightMAC, denoted as 1k-LightMAC, and give the following results:

(A) *1k-LightMAC is as secure as two-key LightMAC, while the query-lengths are lower bounded by* $(n-s)$ *bits and upper bounded by* $(n-s)\min\{2^{n/4}, 2^s\}$ *bits.* In other words, we show a security bound of $O(q^2/2^n)$ for 1k-LightMAC, while $(n-s) \leq \ell \leq (n-s)\min\{2^{n/4}, 2^s\}$.

(B) To justify the lower bound on the message length $\ell$ in the aforementioned result, we demonstrate *an* $O(n)$*-query forgery attack on* 1k-LightMAC *when the adversary is allowed to make short queries* of length less than $(n-s)$ bits.

Second, in section 4.4, we propose two close variants of 1k-LightMAC, dubbed as LightMAC-swp and LightMAC-ds and show the following results:

(C) LightMAC-swp — *a variant of* 1k-LightMAC*, obtained by swapping the position of message blocks and counter values — is as secure as two-key* LightMAC *while* $\ell \leq (n-s)\min\{2^{n/4}, 2^s\}$. Note that the security result for LightMAC-swp does not require the lower bound on $\ell$.

(D) LightMAC-ds — *a variant of* 1k-LightMAC*, obtained by introducing a* 1*-bit domain separation — is asymptotically as secure as two-key* LightMAC*, i.e., it achieves security bound of* $O(q^2/2^n)$ *while* $\ell \leq (n-s)2^{s-1}$. The restriction on length is due to the loss of 1-bit from counter for domain separation.

In order to circumvent the two bottlenecks discussed in section 4.1.2, we use a novel sampling approach, called the *reset-sampling* – a proof style much in the same vein as

**Table 1.4.2:** A comparative summary of several birthday-bound block cipher based MAC algorithms. Here $q$ denotes the number of queries, $\ell$ denotes the bound on query-length, and $s$ denotes the counter size.

| Mode | #BC Keys | Aux. memory[1] | PRF Bound | Restriction[2] |
|---|---|---|---|---|
| EMAC [5, 10] | 2 | 0 | $O\left(\frac{q}{2^{n/2}}\right)$ [52] | $\ell \leq n2^{n/4}$ |
| ECBC,FCBC [15] | 3 | 0 | $O\left(\frac{q}{2^{n/2}}\right)$ [53] | $\ell \leq n2^{n/4}$ |
| XCBC [15] | 1 | $2n$ | $O\left(\frac{q^2\ell}{2^n}\right)$ [67] | $\ell \leq n2^{n/3}$ |
| OMAC [47] | 1 | $n$ | $O\left(\frac{q^2\ell}{2^n}\right)$ [73] | $\ell \leq n2^{n/4}$ |
| PMAC [16] | 1 | $n$ | $\Theta(\frac{q^2\ell}{2^n})$ [39, 67, 76] | - |
| PMAC3 [70] | 2 | $3n$ | $O\left(\frac{q^2}{2^n}\right)$ [22, 70] | $\ell \leq n2^{n/2}$ |
| LightMAC [1, 62] | 2 | $s$ | $O\left(\frac{q^2}{2^n}\right)$ [62] | $\ell \leq (n-s)2^s$ |
| 1k-LightMAC | 1 | $s$ | $O\left(\frac{q^2}{2^n}\right)$ | $(n-s) \leq \ell \leq (n-s)\min\{2^{n/4}, 2^s\}$ |
| LightMAC-swp | 1 | $s$ | $O\left(\frac{q^2}{2^n}\right)$ | $\ell \leq (n-s)\min\{2^{n/4}, 2^s\}$ |
| LightMAC-ds | 1 | $s$ | $O\left(\frac{q^2}{2^n}\right)$ | $\ell \leq (n-s)2^{s-1}$ |

[1] The memory used to store masking keys or counter value.
[2] Upper bound on query-lengths for which the given security bound holds.

the reprogramming of random oracles [38]. At the highest level, reset-sampling can be viewed as a subroutine in H-coefficient [82, 83] or Expectation method [45] based proofs that can be employed in order to transform a possibly bad transcript into a good transcript given that certain conditions are fulfilled. In other words, it resets some bad transcript into a good transcript. For example, in our analysis of 1k-LightMAC and LightMAC-swp, we reset the intermediate outputs appropriately whenever the corresponding intermediate input collides with some final input.

Table 1.4.2 gives a comparison of LightMAC, 1k-LightMAC, LightMAC-swp, and LightMAC-ds with several popular birthday-bound block cipher based MAC mode of operation. We deliberately refrain from enumerating beyond-the-birthday bound modes for a fair comparison, as they require relatively more memory and/or key material (due to the BBB security requirement). From the table, it is clear that the four LightMAC candidates are overall better than other modes considering security vs block cipher key size and security vs auxiliary memory. Further, *1k-LightMAC is almost as good as LightMAC as long as* $(n-s) \leq \ell \leq (n-s)\min\{2^{n/4}, 2^s\}$.

PRACTICAL SIGNIFICANCE: Our results are restricted in terms of the length of messages, especially, 1k-LightMAC which effectively bounds the message length to roughly $2^{35.5}$ bytes for 128-bit block size. However, we believe that this is a minor issue. Indeed, many real life communication protocols limit the message lengths to much less than 1

Gigabyte. For example, SRTP [3] limits the payload length to at most 1 Megabyte. So, the impact of length restriction could, in fact, be minimal in most applications. Furthermore, we emphasize that 1k-LightMAC can be used as a drop-in replacement, since the required changes are minimal. This is particularly a compelling feature for the intended application area of the ISO/IEC-29192-6:2019 standard, i.e. resource constrained environments, where additional deployment or maintenance cost is highly undesirable. In summary, our results have significant practical importance due to the ISO/IEC standardization of LightMAC and the inherent advantages of 1k-LightMAC, LightMAC-swp, and LightMAC-ds over LightMAC.

**Publication status:** Chapter 4 is mostly based on our paper [23], published in *Advances in Cryptology – ASIACRYPT 2021*. The complete chapter is based on an extended version of this paper.

### 1.4.3 PMAC family: Towards a generalization

MOTIVATION: The main motivation of this chapter is to have tight analysis of a wide class of single keyed PMAC-type constructions. This might help us to find some parallel constructions which could have $\ell$-free PRF advantage. Let us first consider a variant of single keyed LightMAC in which we apply domain separation to the final block cipher call with the other intermediate calls. More precisely, let the final output be $e_K(1\|\mathsf{H}_K(m_1,\ldots,m_\ell))$ where $\mathsf{H}_K(m_1,\ldots,m_\ell) = \mathsf{chop}_1(e_{K_1}(0\|m_1\|\langle 1\rangle_s) \oplus \cdots \oplus e_{K_1}(0\|m_\ell\|\langle \ell\rangle_s))$ and $\mathsf{chop}_t$ denotes the function which chops $t$ bits. One may think that the analysis of this construction would be similar to the two-keyed LightMAC and hence we may get $\ell$-free bound for it. Unfortunately, the hash-then-PRP analysis is still not applicable to this single-key (or dependent system) setting. As the final inputs cannot collide with other intermediate inputs, the $q$ final outputs are stochastically dependent with the internal $\ell q$ outputs, since they cannot collide due to the domain separation. Hence, proving $q^2/2^n$ bound for this variant is still not obvious.

OUR CONTRIBUTIONS: We first define a family of parallelizable hash functions, denoted as xPHash, based on block cipher. The parallelizable MAC, denoted as xPMAC, is simply a hash-then-PRP where the hash function is xPHash. Many known parallel designs fall into this class. For example, PMAC, PMAC 1, LightMAC are instantiations of this general construction.

In this paper we analyze this class when it is instantiated by the single key, i.e., the key of the final call of the block cipher is same as that used in the hash computation. Let xPMAC be such a construction with an underlying hash function xPHash. In Theorem

5.2.1 of in Sect. 5.1, we show that PRF-advantage of xPMAC is at most

$$O(\frac{q^2}{2^n}) + \mathsf{coll}_{\mathsf{xPHash}}(q, \ell), \quad \forall \ell \leq 2^{n/4} \tag{1.3}$$

where $\mathsf{coll}_{\mathsf{xPHash}}(q, \ell)$ denotes the collision probability of the underlying hash function. A similar result also holds for a single-keyed construction (as shown in Theorem 5.2.2 in Sect. 5.1). In almost all constructions the collision probability is at least $q^2/2^n$ and hence the collision probability is the tight estimate of the PRF advantage. So, our result mainly reduce the PRF advantage computation to get an exact estimate of the maximum collision probability.

- The collision probability for the hash LightHash (the underlying hash for LightMAC) is about $1/2^n$. So we can conclude that the single keyed version of LightMAC has PRF advantage about $q^2/2^n$ which is clearly tight.
- Under some reasonable assumption, the same tight bound is true for PMAC where the masking of PMAC+ is used instead of the Gray coding.

This masking of PMAC+ is a 2-wise independent function. Gazi et al. proved the collision probability of the hash of PMAC based on 4-wise independent masking is about $1/2^n$. They have also showed a counter-example of 2-wise independent masking for which the collision probability is $\ell/2^n$. However, we have shown that the optimum collision probability can be achieved by a 2-wise independent masking.

**Publication status:** A research paper based on chapter 5 is *in preparation* for submission.

### 1.4.4 OMAC, XCBC and TMAC

MOTIVATION: Continuing the discussion in section 1.3.2, we can further state an interesting fact: in [55], Jha et al. showed that if we use PRF instead of a block cipher in these constructions there is an attack with roughly $\Omega(q^2\ell/2^n)$ advantage which is tight. No such attack is known in presence of block cipher. This gives an implicit motivation to study the exact security of these constructions in presence of block ciphers. In this chapter we aim to show birthday-bound security for these block cipher based MACs for a suitable range of query-length.

In a different paradigm but with similar motivations, recently Chattopadhyay et al. [23] showed birthday-bound security for another standardized MAC called LightMAC [62]. However, similar result for original PMAC [16] is still an open problem (although

a result is available for its variant in [22]). In addition to the improved bound for LightMAC, Chattopadhyay et al. proposed a new proof approach called the reset-sampling method. They also hinted (via a very brief discussion) that this method could be useful for proving better security for OMAC. However, the discussion in [23] is over-simplistic and contains no formal analysis of bad events. Indeed, the reset-sampling is more involved than anticipated in [23], giving rise to some crucial and tricky bad events. To their credit, they do say that

> A more formal and rigorous analysis of OMAC using reset-sampling will most probably require handling of several other bad events, and could be an interesting future research topic.

In this chapter of the thesis, we take up this research topic and give a complete and rigorous analysis.

OUR CONTRIBUTIONS: In section 6.1, we show that the PRF advantages for OMAC,

**Table 1.4.3:** Summary of security (PRF advantage) bounds for the CBC-MAC family. Here $n$, $q$, $\ell$, and $\sigma$ denote the block size, number of queries, maximum permissible message length, and sum of message lengths of all $q$ queries, respectively.

| Scheme | State-of-the-art | | This paper | |
|---|---|---|---|---|
| | Bound | Restriction | Bound | Restriction |
| CBC-MAC [37] | $O(\sigma q/2^n)$ [52, 53] | $\ell = o(2^{n/3})$ | - | - |
| EMAC [5, 10] | $O(q^2/2^n) + O(q\ell^2/2^n)$ [52, 53] | - | - | - |
| ECBC [15] | $O(q^2/2^n) + O(q\ell^2/2^n)$ [52, 53] | - | - | - |
| FCBC [15] | $O(q^2/2^n) + O(q\ell^2/2^n)$ [52, 53] | - | - | - |
| XCBC [15] | $O(q^2\ell/2^n)$ [67][1]<br>$O(\sigma^2/2^n)$ [48][1] | $\ell = o(2^{n/3})$<br>- | $O(q^2/2^n) + O(q\ell^2/2^n)$ | - |
| TMAC [60] | $O(q^2\ell/2^n)$ [67][1]<br>$O(\sigma^2/2^n)$ [48][1] | $\ell = o(2^{n/3})$<br>- | $O(q^2/2^n) + O(q\ell^2/2^n)$ | - |
| OMAC [47] | $O(\sigma q/2^n)$ [74] | $\ell = o(2^{n/3})$ | $O(q^2/2^n) + O(q\ell^2/2^n)$ | - |

[1] $\sigma^2$ and $q^2\ell$ are incomparable, as they depend on the query length distribution.

XCBC and TMAC are upper bounded by $O(q^2/2^n) + O(q\ell^2/2^n)$, which is almost tight in terms of the number of queries $q$ while $\ell \ll 2^{n/4}$. This bound is not exactly the birthday bound $O(q^2/2^n)$, but for any fixed target advantage, in terms of the limit on $q$ it behaves almost like the birthday bound for a fairly good range of $\ell$ (see the

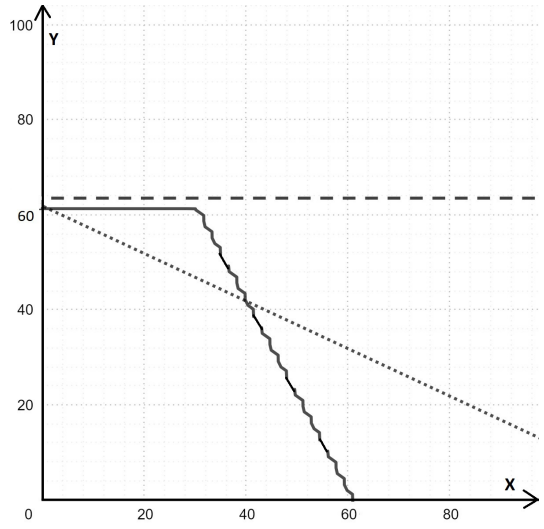


**Figure 1.4.1:** For $\epsilon = 2^{-1}$

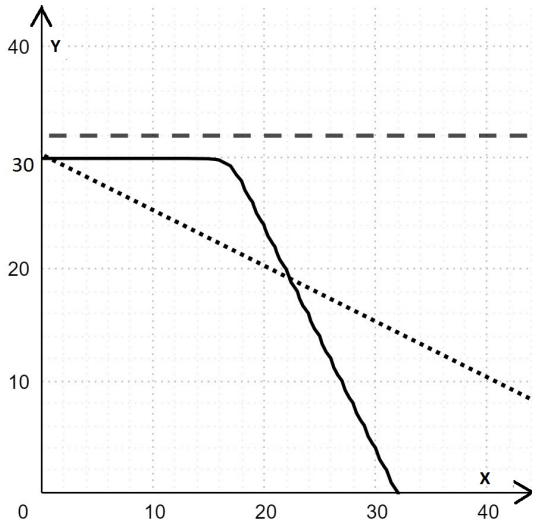


**Figure 1.4.2:** For $\epsilon = 2^{-64}$

**Figure 1.4.3: $(\log \ell, \log q)$-Trade-off Graph for the bounds of OMAC:** For two different choices of the target advantage, $\boldsymbol{\epsilon = 2^{-1}}$ (on the left), and $\boldsymbol{\epsilon = 2^{-64}}$ (on the right), the above graphs show the relation between $\boldsymbol{X = \log \ell}$ and $\boldsymbol{Y = \log q}$. Here $\boldsymbol{n = 128}$. The *dashed* curve, the *dotted* curve and the *continuous* curve represent the relations for ideal birthday bound, the bound in [74] and the *exact form* of the bound shown in this paper respectively.

following discussion). The proof of our security bound is given in section 6.2 and follows the recently introduced reset-sampling approach [23]. These improved bounds, in combination with previous results [52, 53] for EMAC, ECBC and FCBC, completely characterize (see Table 1.4.3) the security landscape of CBC-MAC variants for message lengths up to $2^{n/4}$ blocks.

A NOTE ON THE TIGHTNESS AND IMPROVEMENT IN BOUNDS: Comparison among the known best bound for OMAC [74] $B_1(\ell, q) = 10q^2\ell/2^n$, the ideal birthday bound $B_{\mathsf{id}} = q^2/2^n$ and the bound $B_2(\ell, q)$ proved in this paper (see Theorem 6.1.1) will be best understood if we look closely at those bounds in a slightly involved manner. We write "log" to mean log base 2. We show the trade-off curve of the parameters $X = \log \ell$ and $Y = \log q$ for a fixed choice of advantage value, say, $\epsilon = 2^{-a}$ for some $a \in \mathbb{N}$ in the graph[2] presented in Figure 1.4.3.

We can write our bound as $B_2(\ell, q) \approx \frac{16q^2}{2^n} + \frac{2q\ell^2}{2^n}$ as the remaining terms are dominated by these two terms. Now,

$$\begin{aligned} B_{\mathsf{id}}: &\quad Y = \frac{n-a}{2} \\ B_1: &\quad X + 2Y = n - a - \log 10 \\ B_2: &\quad \log(16 \cdot 2^{2Y} + 2 \cdot 2^{2X+Y}) = n - a. \end{aligned}$$

[2]Using GeoGebra Classic available at `https://www.geogebra.org/classic`

Looking at the equation related to the bound $B_2$ we can see that it is actually a combination of two linear equations: $2Y = n - a - 4$ and $2X + Y = n - a - 1$, the choice depending on whether $16q^2/2^n$ or $2q\ell^2/2^n$ dominates. Precisely, the curve expressing the relation between $\log \ell$ and $\log q$ in $B_2$ is $\{(X, Y) : X \leq n/4, Y = \min\{(n - a - 4)/2, n - a - 1 - 2X\}\}$. From the above linear equations two important facts about the curve related to $B_2$ can be noticed:

- It remains very close to the straight line corresponding to $B_{\mathsf{id}}$ from $(0, \frac{n-a-4}{2})$ to $(\frac{n-a+2}{4}, \frac{n-a-4}{2})$ and then moves downward.
- At around $(\frac{n-a+1}{3}, \frac{n-a-5}{3})$ it starts to degrade below the curve related to $B_1$ .

For example, if we take $(n, a) = (128, 32)$, the bound proved in this paper is very close to the birthday bound for $\ell \leq 2^{25}$ and even after degrading, it remains better than the bound in [74] till $\ell \leq 2^{32}$. Moreover, if we take $(n, a) = (128, 64)$, $q$ remains $2^{30}$ until $\ell \leq 2^{16}$ and degrades below the existing bound only after $\ell \geq 2^{22}$. Thus, if we consider the advantage in general terms, we can always take the minimum among the advantage proved in this paper and that proved in [74].

**Publication status:** Chapter 6 is based on our paper [24] which has been published in *Advances in Cryptology – ASIACRYPT 2022.*

# Chapter 2

# Preliminaries

## 2.1 Setup

BASIC NOTATIONS: For any positive integer $n$, we write $[n] := \{1, \ldots, n\}$. We write $x^q$ to denote a $q$-tuple $(x_1, \ldots, x_q)$. We write $X \leftarrow\!\$\ \mathcal{X}$ to represent that $X$ is a uniform random variable taking values from a finite nonempty set $\mathcal{X}$. For any non-empty set $A$, $A^*$ denotes the set $\cup_{i \geq 0} A^i$ consisting of all the finite strings of elements from $A$ (including the empty string also).

Throughout, $\boldsymbol{\rho}_{\mathcal{D}} \leftarrow\!\$\ \mathsf{Func}_{\mathcal{D}}$ denotes a random function, and $\boldsymbol{\pi} \leftarrow\!\$\ \mathsf{Perm}$ denotes a random permutation. We simply write the random function as $\boldsymbol{\rho}$, when $\mathcal{D}$ is understood from the context.

NOTATIONS ON BLOCKS: Throughout the paper $n$ denotes the security parameter as well as the bit size of the underlying permutation. We call the set $\mathfrak{B} := \{0,1\}^n$ block set and elements of the set *blocks*. We define $\mathfrak{B}^+ = \cup_{i \geq 1} \mathfrak{B}^i$. For any binary string $m \in \{0,1\}^*$, we denote the number of bits of $m$ as $|m|$ and we write $\|m\| := \lceil |m|/n \rceil$.[1] We use "$\|$" to denote concatenation operations on bit strings. For a message $m \in \{0,1\}^{nl}$, we write $m = m[1] \| \cdots \| m[l]$ with $m[i] \in \{0,1\}^n$ for all $i \in [l]$.

NOTATIONS ON BLOCK FUNCTIONS AND PERMUTATIONS: We call a function block function if the range of the function is the block set. The set of all block functions defined over a set $\mathcal{D}$ is denoted as $\mathsf{Func}_{\mathcal{D}}$. The set of all permutations over the block set (also called block permutation) is denoted as $\mathsf{Perm}$.

[1] When $m$ is a set we also write $|m|$ to denote the size of the set $m$. So the notation $|m|$ should be clear from the context.

A keyed block function $F$ with key space $\mathcal{K}$ and domain $\mathcal{D}$ is a block function over $\mathcal{K} \times \mathcal{D}$. We also view it as an indexed family of functions, where $\mathcal{K}$ is the index set, i.e., for each $K \in \mathcal{K}$, we associate a function $F_K(\cdot) := F(K, \cdot)$.

MULTISET: Informally, a multiset $\mathcal{X}$ is a variant of set in which we allow elements to repeat. One can equivalently define a multiset $\mathcal{X}$ by a set $\{(x, m) : x \in \mathcal{X}, x \text{ appears } m \text{ times in } \mathcal{X}\}$. We write $\mathcal{X}^o$ to denote the set of all elements $x$ which appears odd times in $\mathcal{X}$. Note that, $\mathcal{X}^o$ by definition is a set which can be empty. We say $\mathcal{X}$ is **evenly repeated** if $\mathcal{X}^o = \emptyset$.

**Example 2.1.** *Let $\mathcal{X} := \{a, b, a, b, b, c\}$ be a multiset. We represent it by the following set $\{(a, 2), (b, 3), (c, 1)\}$. Note that $\mathcal{X}^o = \{b, c\}$. Similarly, for a multiset $\mathcal{Y} := \{a, b, a, b, b, b, c, c\}$, $\mathcal{Y}^o = \emptyset$ and hence $\mathcal{Y}$ is evenly repeated.*

Given a block function $\pi$, we use shorthand notation $\pi^{\oplus}(\mathcal{X}) := \bigoplus_{x \in \mathcal{X}} \pi(x)$. With this notation, it is easy to see that (the empty sum represents $0^n$)

$$\pi^{\oplus}(\mathcal{X}) = \pi^{\oplus}(\mathcal{X}^o) \text{ for every multiset } \mathcal{X}, \tag{2.1}$$

and hence $\pi^{\oplus}(\mathcal{X}) = 0^n$ whenever $\mathcal{X}$ is evenly repeated multiset.

BINARY FIELD: In this paper, we view the block set $\mathfrak{B}$ as the Galois field $\mathrm{GF}(2^n)$. We fix a primitive polynomial $p(x) := p_0 \oplus p_1 x \oplus \cdots \oplus p_n x^n$ where $p_i \in \{0, 1\}$. Note that $p_0 = p_n = 1$ (as it is a primitive polynomial). The field multiplication "$\cdot$" between two field elements is defined through the primitive polynomial. We abuse the notation 2 to denote a primitive element of the underlying field $\mathrm{GF}(2^n)$.

## 2.2 Mathematical Notions

### 2.2.1 Hash Functions

In the following, let $H$ be a keyed block function with keyspace $\mathcal{K}$ and domain $\mathcal{D}$.

COLLISION PROBABILITY: For distinct $m, m' \in \mathcal{D}$, we define collision probability as

$$\mathsf{coll}_H(m, m') := \mathsf{Pr}[H(K, m) = H(K, m') : K \leftarrow\!\$\, \mathcal{K}].$$

When $\mathcal{D} \subseteq \{0, 1\}^*$, the collision probability can depend on the size of the inputs. We write

$$\mathsf{coll}_H(\ell) = \max_{\substack{m \neq m' \\ |m|, |m'| \leq \ell}} \mathsf{coll}_H(m, m').$$

We generalize the above definition for more than two inputs. For $q$ distinct inputs $m_1, \ldots, m_q \in \mathcal{D}$, we write

$$\mathsf{coll}_H(m^q) := \mathsf{Pr}(\exists i < j, H(K, m_i) = H(K, m_j) : K \leftarrow\!\$\, \mathcal{K}), \text{ and}$$
$$\mathsf{coll}_H(q, \ell, \sigma) := \max_{\substack{m^q : |m_i| \leq \ell \\ \sum_{i=1}^{q} |m_i| \leq \sigma}} \mathsf{coll}_H(m^q).$$

By using the union bound, $\mathsf{coll}_H(q, \ell, \sigma) \leq \binom{q}{2}\mathsf{coll}_H(\ell)$.

**Definition 2.2.1** (Universal hash function)**.** The keyed block function $H$ is called an $\epsilon$-universal hash if for all distinct $m, m' \in \mathcal{D}$, $\mathsf{coll}_H(m, m') \leq \epsilon$.

**Definition 2.2.2** (XOR universal hash function)**.** The keyed block function $H$ is called an $\epsilon$-almost XOR universal hash if for all distinct $m, m' \in \mathcal{D}$ and $\delta \in \mathfrak{B}$,

$$\mathsf{Pr}(H(K, m) \oplus H(K, m') = \delta : K \leftarrow\!\$\, \mathcal{K}) \leq \epsilon.$$

**Definition 2.2.3** ($k$-wise independent hash function)**.** The keyed block function $H$ is called a $k$-wise independent if for all distinct $m_1, \ldots, m_k \in \mathcal{D}$ and for all $y_1, \ldots, y_k \in \mathfrak{B}$,

$$\mathsf{Pr}(H(K, m_1) = y_1, \ldots, H(K, m_k) = y_k : K \leftarrow\!\$\, \mathcal{K}) = \frac{1}{2^{kn}}.$$

The following observations are easy to establish.

1. A random function is $k$-wise independent for any $k$.
2. A 2-wise independent hash function is $2^{-n}$-AXU.

For more details on hash functions, one can see the papers [21, 95, 96].

### 2.2.2 Pseudorandom Functions and the Hash-then-RP Paradigm

**Definition 2.2.4** (Pseudorandom function)**.** Let $F$ be a keyed block function over a finite set $\mathcal{D}$ with a finite key space $\mathcal{K}$. The *PRF-advantage* of any oracle adversary $\mathcal{A}$ against $F$ is defined as

$$\mathbf{Adv}^{\mathsf{prf}}_F(\mathcal{A}) := \left|\mathsf{Pr}(\mathcal{A}^{F_K} = 1 : K \leftarrow\!\$\, \mathcal{K}) - \mathsf{Pr}(\mathcal{A}^{\rho_{\mathcal{D}}} = 1)\right|.$$

The *maximum PRF-advantage of* $F$ is defined as

$$\mathbf{Adv}^{\mathsf{prf}}_F(q, \ell, \sigma) = \max_{\mathcal{A}} \mathbf{Adv}^{\mathsf{prf}}_F(\mathcal{A}),$$

where the maximum is taken over all adversaries $\mathcal{A}$ making at most $q$ queries, each of length at most $\ell$, and the total length of all queries at most $\sigma$, i.e., $\sigma \leq \ell q$.

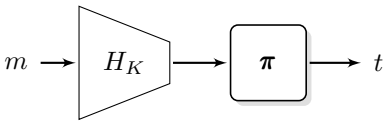


**Figure 2.2.1:** The Hash-then-RP paradigm.

HASH-THEN-RP CONSTRUCTION: Let $H : \mathcal{K} \times \mathcal{D} \to \mathfrak{B}$ be a keyed hash and $\boldsymbol{\pi}$ be an $n$-bit random permutation. The composition $\boldsymbol{\pi} \circ H_K$ is called the Hash-then-RP construction, where $K \leftarrow\!\$\, \mathcal{K}$. When $\boldsymbol{\pi}$ is replaced with $\boldsymbol{\rho}$, the resulting composition is called the Hash-then-RF. These constructions have been studied in [21, 91]. Many PRF constructions can be viewed as instances of Hash-then-RP/RF. For example, EMAC [5, 10], ECBC, FCBC [15], LightMAC [62] and protected counter sum [12]. Proposition 2.2.5 gives the PRF advantage for Hash-then-RP construction.

**Proposition 2.2.5.** *Let $H$ be a keyed block function with keyspace $\mathcal{K}$ and domain $\mathcal{D}$. Then, we have*

$$\mathbf{Adv}^{\mathsf{prf}}_{\boldsymbol{\pi} \circ H}(q, \ell, \sigma) \leq \mathsf{coll}_H(q, \ell, \sigma) + \frac{q(q-1)}{2^{n+1}}.$$

*So, if $H$ is an $\epsilon$-universal hash function, then*

$$\mathbf{Adv}^{\mathsf{prf}}_{\boldsymbol{\pi} \circ H}(q, \ell, \sigma) \leq \frac{q(q-1)}{2}\left(\epsilon + \frac{1}{2^n}\right).$$

We skip a formal proof here as Proposition 2.2.5 is a well-known result. The readers are referred to [39] for a formal proof.

### 2.2.3 Message Authentication Codes

Let $\mathcal{P}, \mathcal{K}, \mathcal{N}, \mathcal{T}$ denote the space of plaintexts (messages), keys, nonces and tags, respectively. A message authentication code (or, MAC) $\mathcal{M}$ is a pair $(\mathcal{M}^+, \mathcal{M}^-)$ where

$$\mathcal{M}^+ : \mathcal{K} \times \mathcal{N} \times \mathcal{P} \to \mathcal{T}$$
$$\mathcal{M}^- : \mathcal{K} \times \mathcal{N} \times \mathcal{P} \times \mathcal{T} \to \{\mathsf{T}, \mathsf{F}\}$$

We call $\mathcal{M}^+$ and $\mathcal{M}^-$ as tag generation and tag verification algorithms, respectively. For any $(K, N, m, t) \in \mathcal{K} \times \mathcal{N} \times \mathcal{P} \times \mathcal{T}$, $\mathcal{M}^-(K, N, m, t) := \mathsf{T}$ if $\mathcal{M}^+(K, N, m) = t$; otherwise $\mathcal{M}^-(K, N, m, t) := \mathsf{F}$. Note that, for any $(K, N, m) \in \mathcal{K} \times \mathcal{N} \times \mathcal{P}$, we always have $\mathcal{M}^-(K, N, m, \mathcal{M}^+(K, N, m)) = \mathsf{T}$. For brevity, we write $\mathcal{M}^+(K, \cdot)$ and $\mathcal{M}^-(K, \cdot)$ as $\mathcal{M}^+_K$ and $\mathcal{M}^-_K$, respectively, for any $K \leftarrow\!\$\, \mathcal{K}$. In this thesis, $\mathcal{N} = \emptyset$, $\mathcal{K}, \mathcal{T} \subset \mathfrak{B}^+$ and $\mathcal{P} \subset \{0,1\}^*$.

### 2.2.4 Security Definitions

DISTINGUISHERS: A $(q, T)$-distinguisher $\mathscr{A}$ is an interactive Turing machine, that makes at most $q$ oracle queries, each having at most $\ell$ length, sum of lengths of all queries not exceeding $\sigma$, runs in time at most $t$, and outputs a single bit. For any oracle $\mathcal{O}$, we write $\mathscr{A}^{\mathcal{O}}$ to denote the output of $\mathscr{A}$ after its interaction with $\mathcal{O}$. By convention, $t = \infty$ denotes computationally unbounded (information-theoretic) and deterministic distinguishers. In this paper, we assume that the distinguisher is non-trivial, i.e., it never makes a duplicate query. Let $\mathbb{A}(q, \ell, \sigma, t)$ (or, $\mathbb{A}(q, t)$) be the class of all non-trivial distinguishers limited to $q, \ell, \sigma$ parameters (or, $q$ queries) and $t$ computations. A distinguisher is also called an *adversary*.

PSEUDORANDOM FUNCTION: A $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$-*keyed function* $\mathsf{F}$ with key space $\mathcal{K}$, domain $\mathcal{X}$, and range $\mathcal{Y}$ is a function $\mathsf{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. We write $\mathsf{F}_K(X)$ for $\mathsf{F}(K, X)$.

The *pseudorandom function* or PRF advantage of any distinguisher $\mathscr{A}$ against a $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$-keyed function $\mathsf{F}$ is defined as

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{F}}(\mathscr{A}) = \mathbf{Adv}_{\mathsf{F};\boldsymbol{\rho}}(\mathscr{A}) := \left|\mathsf{Pr}[\mathscr{A}^{\mathsf{F}_\mathsf{K}} = 1 \ : \ \mathsf{K} \leftarrow\!\$\, \mathcal{K}] - \mathsf{Pr}[\mathscr{A}^{\boldsymbol{\rho}} = 1 \ : \ \boldsymbol{\rho} \leftarrow\!\$\, \mathsf{Func}(\mathcal{X}, \mathcal{Y})]\right|. \tag{2.2}$$

The *PRF security* of $\mathsf{F}$ against $\mathbb{A}(q, T)$ is defined as

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{F}}(q, T) := \max_{\mathscr{A} \in \mathbb{A}(q,T)} \mathbf{Adv}^{\mathsf{prf}}_{\mathsf{F}}(\mathscr{A}).$$

PSEUDORANDOM PERMUTATION: A $(\mathcal{K}, \mathbb{B})$-*block cipher* $\mathsf{E}$ with key space $\mathcal{K}$ and block space $\mathbb{B}$ is a $(\mathcal{K}, \mathbb{B}, \mathbb{B})$-keyed function, such that $E(K, \cdot)$ is a permutation over $\mathbb{B}$ for any key $K \in \mathcal{K}$. We write $\mathsf{E}_K(X)$ for $\mathsf{E}(K, X)$.

The *pseudorandom permutation* or PRP advantage of any distinguisher $\mathscr{A}$ against a $(\mathcal{K}, \mathbb{B})$-block cipher $\mathsf{E}$ is defined as

$$\mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(\mathscr{A}) = \mathbf{Adv}_{\mathsf{E};\boldsymbol{\pi}}(\mathscr{A}) := \left|\mathsf{Pr}[\mathscr{A}^{\mathsf{E}_\mathsf{K}} = 1 \ : \ \mathsf{K} \leftarrow\!\$\, \mathcal{K}] - \mathsf{Pr}[\mathscr{A}^{\boldsymbol{\pi}} = 1 \ : \ \boldsymbol{\pi} \leftarrow\!\$\, \mathsf{Perm}(n)]\right|. \tag{2.3}$$

The *PRP security* of $\mathsf{E}$ against $\mathbb{A}(q, T)$ is defined as

$$\mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(q, T) := \max_{\mathscr{A} \in \mathbb{A}(q,T)} \mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(\mathscr{A}).$$

SECURITY OF A MAC: An adversary $\mathscr{A}$ here is supposed to make queries to both $\mathcal{M}^+_K$ and $\mathcal{M}^-_K$ where $K \leftarrow\!\$\, \mathcal{K}$. It is said to *forge* the MAC $\mathcal{M}$ if it gets $\mathsf{T}$ as a response

to a query to the verification algorithm after playing the query-response game in any arbitrary order. The interactive algorithm $\mathscr{A}^{(\mathcal{M}_K^+, \mathcal{M}_K^-)}$ returns 1 if it can forge. We define *(MAC) Advantage of $\mathscr{A}$ against $\mathcal{M}$* as follows:

$$\mathbf{Adv}_{\mathcal{M}}^{\mathsf{mac}}(\mathscr{A}) = \mathsf{Pr}[\mathscr{A}^{(\mathcal{M}_K^+, \mathcal{M}_K^-)} = 1 \ : \ \mathsf{K} \leftarrow\!\$\, \mathcal{K}]$$

Security of a MAC is defined using this notion of MAC advantage. Suppose, an adversary is allowed to make at most $q_m$ (and $q_v$) queries to the tag generation (and tag verification algorithm), each of length at most $\ell_m$ (and $\ell_v$), sum of lengths of all queries not exceeding $\sigma_m$ (and $\sigma_v$) and $t$ is the maximum permissible time. Then,

$$\mathbf{Adv}_{\mathcal{M}}^{\mathsf{mac}}(q, \ell, \sigma, t) := \max\{\mathbf{Adv}_{\mathcal{M}}^{\mathsf{mac}}(\mathscr{A}) : \mathscr{A} \in \mathbb{A}(q, \ell, \sigma, t)\}$$

where $q := q_m + q_v$, $\ell := \ell_m + \ell_v$ and $\sigma := \sigma_m + \sigma_v$. Thus, security of the MAC $\mathcal{M}$ is parametrized by $q, \ell, \sigma$ and $t$. The less is the advantage, the more is the security of the MAC (with respect to the security parameters). The following proposition establishes a very interesting as well as useful relationship between the notions of PRF security and MAC security.

**Proposition 2.2.6** (PRF security implies MAC security)**.** *Suppose, $\mathcal{M}$ is a deterministic MAC. Then the following result holds:*

$$\mathbf{Adv}_{\mathcal{M}^+}^{\mathsf{prf}}(q_m, \ell_m, \sigma_m, t) \leq \epsilon \implies \mathbf{Adv}_{\mathcal{M}}^{\mathsf{mac}}(q, \ell, \sigma, t) \leq \epsilon + \frac{q_v}{|\mathcal{T}|}$$

Proof of this theorem can be found in [5]. More discussion on the topics covered in this section can be found in [19].

### 2.2.5 H-coefficient Technique

The H-coefficient technique by Patarin [82, 83] is a tool to upper bound the distinguishing advantage of any deterministic and computationally unbounded distinguisher $\mathscr{A}$ in distinguishing the real oracle $\mathcal{R}$ from the ideal oracle $\mathcal{I}$. The collection of all queries and responses that $\mathscr{A}$ made and received to and from the oracle, is called the transcript of $\mathscr{A}$, denoted as $\tau$.

Let $\Theta_1$ and $\Theta_0$ denote the transcript random variable induced by $\mathscr{A}$'s interaction with $\mathcal{R}$ and $\mathcal{I}$, respectively. Let $\mathcal{T}$ be the set of all transcripts. A transcript $\tau \in \mathcal{T}$ is said to be *attainable* if $\mathsf{Pr}[\Theta_0 = \tau] > 0$, i.e., it can be realized by $\mathscr{A}$'s interaction with $\mathcal{I}$. Following

these notations, we state the main result of H-coefficient technique in Theorem 2.2.7. A proof of this theorem is available in multiple papers, including [25, 45, 65, 83]. The key idea behind the proof is that the advantage of $\mathscr{A}$ is upper bounded by the statistical distance of the random variables $\Theta_1$ and $\Theta_0$.

**Theorem 2.2.7** (H-coefficient). *For $\epsilon_1, \epsilon_2 \geq 0$, suppose there is a set $\mathcal{T}_{\mathsf{bad}} \subseteq \mathcal{T}$, that we call the set of all bad transcripts, such that the following conditions hold:*

- $\mathsf{Pr}[\Theta_0 \in \mathcal{T}_{\mathsf{bad}}] \leq \epsilon_1$*; and*
- *For any $\tau \notin \mathcal{T}_{\mathsf{bad}}$, $\tau$ is attainable and* $\dfrac{\mathsf{Pr}[\Theta_1 = \tau]}{\mathsf{Pr}[\Theta_0 = \tau]} \geq 1 - \epsilon_2$.

*Then, for any computationally unbounded and deterministic distinguisher $\mathscr{A}$, we have*

$$\mathbf{Adv}_{\mathcal{R};\mathcal{I}}(\mathscr{A}) \leq \epsilon_1 + \epsilon_2.$$

## 2.3 Useful Lemmas

We write $x_1, \ldots, x_r \in_{\text{dist}} \mathcal{X}$ to mean that $x_i$'s are distinct elements of a set $X$ (of size $N$). The number of possible tuples $(x_1, \ldots, x_r)$ of distinct elements is denoted as $(N)_r$ which is same as $N(N-1)\cdots(N-r+1)$.

For a finite set $\mathcal{S}$, $\mathsf{Y}_1, \ldots, \mathsf{Y}_t \xleftarrow{\text{wor}} \mathcal{S}$ represents a without-replacement random sample. In other words, for all distinct $y_1, \ldots, y_q \in \mathcal{S}$, $\mathsf{Pr}[\mathsf{Y}_1 = y_1, \ldots, \mathsf{Y}_t = y_t] = 1/(|\mathcal{S}|)_t$. A without-replacement sample can be equivalently described through a random permutation $\boldsymbol{\pi}$ over a set $\mathcal{S}$ as follows: Let $x_1, \ldots, x_t$ be $t$ distinct elements from the set $\mathcal{S}$. Then $\mathsf{Y}_1, \ldots, \mathsf{Y}_t \xleftarrow{\text{wor}} \mathcal{S}$ where $\mathsf{Y}_i = \boldsymbol{\pi}(x_i)$ for all $i$. From this observation, we can conclude that any subset of a without-replacement random sample is also a without-replacement random sample. In particular, each element of the sample is uniformly distributed (but not independently).

We now state some useful results from linear algebra.

**Lemma 2.3.1.** *Let $\mathsf{Y}_1, \ldots, \mathsf{Y}_t \xleftarrow{\text{wor}} \mathcal{S} \subseteq F$ with $|\mathcal{S}| = N$ where $F$ denotes a finite field. Let $A$ be a $s \times t$ matrix over $F$ with rank $r$. We write the column vector $(\mathsf{Y}_1, \ldots, \mathsf{Y}_t)^{tr}$ as $\mathsf{Y}$. Then, for any $c \in F^s$, we have*

$$\mathsf{Pr}[A \cdot \mathsf{Y} = c] \leq \frac{1}{(N-t)^r}$$

**Proof**.Since the rank of the matrix $A$ is $r$, we can identify $1 \leq i_1 < \cdots < i_r \leq t$ such that $\mathsf{Y}_{i_1}, \ldots \mathsf{Y}_{i_r}$ will be uniquely determined by the other variables. After conditioning all other $\mathsf{Y}$ values, the probability that $A \cdot \mathsf{Y} = c$ would be at most $\frac{1}{(N-t+r-1)_r}$ which is less than $\frac{1}{(N-t)^r}$. □

**Lemma 2.3.2.** *Let $V_1, .., V_t \in F^s$ be nonzero $s$-dimensional vectors such that each entry of these vectors are either 0 or 1. If the rank of these $t$ vectors is $t-1$ then there is a binary nontrivial linear combination of $V_i$'s which gives zero vector. More formally, there exists $b_1, \ldots, b_t \in \{0,1\}$ (not all zeros) such that $\sum_i b_i \cdot V_i = 0^s$.*

Proof of the above lemma can be found in several books on Algebra like [13, 33, 46]. A simple corollary of the lemma says that if three nonzero distinct vectors are linearly dependent then the sum of these three vectors should be the zero vector.

# Part II

# PMAC-type MACs

# Chapter 3

# PMAC Variants

In this chapter, we analyze Naito's variant [70] of PMAC and discuss a flaw in its security proof. We further give a new PMAC-type construction, dubbed as PMAC2, and demonstrate its security proof. Moreover, we extend the result of length independent security bound for sPMAC, as proved in [39], to 2-wise almost xor universal hash encodings.

## 3.1 Revisiting Simplified PMAC

DESCRIPTION OF sPMAC: Gaži et al. [39] proposed a generalized version of PMAC, called sPMAC, to capture the underlying masking function for a wide class of PMAC variants. In what follows $\mathbb{N}$ denotes the set of all natural numbers.

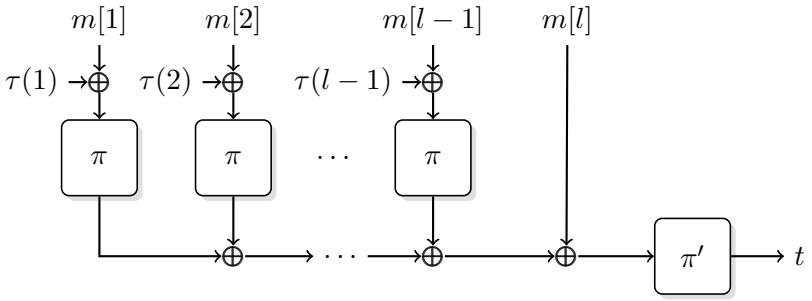


**Figure 3.1.1:** The simplified PMAC construction.

**Definition 3.1.1** (sPHash)**.** For any permutation $\pi \in \mathsf{Perm}$ and a block-valued function $\tau \in \mathsf{Func}_{\mathbb{N}}$ (referred as masking function), we define the *simplified PMAC hash* or sPHash over the message space $\mathfrak{B}^+$ as follows:

for all $m := (m[1], \ldots, m[l]) \in \mathfrak{B}^l$,

$$\mathsf{sPHash}_{\pi,\tau}(m) := m[l] \oplus \bigoplus_{i=1}^{l-1} \pi(x_\tau(m,i)), \text{ where } x_\tau(m,a) := m[a] \oplus \tau(a). \tag{3.1}$$

Clearly, sPHash is just an identity function for a single block message.

Now, given two permutations $\pi, \pi' \in \mathsf{Perm}$ and a masking function $\tau \in \mathsf{Func}_{\mathbb{N}}$, the simplified PMAC or sPMAC construction (illustrated in Figure 3.1.1) is defined as follows: for all $m \in \mathfrak{B}^+$,

$$\mathsf{sPMAC}_{\pi',\pi,\tau}(m) := \pi'\left(\mathsf{sPHash}_{\pi,\tau}(m)\right).$$

We call $K := (\pi', \pi, \tau)$ the key of sPMAC. A concrete variant of PMAC is determined whenever we fix a sampling mechanism of the key $K$.

sPMAC OVER ARBITRARY-LENGTH MESSAGES: For $m \in \{0,1\}^*$, we define

$$\overline{m} := m[1], \ldots, m[l] \xleftarrow{n} m$$

to be the function that partitions $m$ into $l = \frac{|m\|10^i|}{n}$ blocks of size $n$ bits, where $i$ is the smallest non-negative integer such that $|m\|10^i|$ divisible by $n$. Note that, we make the required concatenation even if $|m|$ is divisible by $n$. sPMAC can be easily extended for any arbitrary-length message $m \in \{0,1\}^*$, as $\mathsf{sPMAC}(m) := \mathsf{sPMAC}(\overline{m})$. As the padding rule is injective, there is no loss of generality in ignoring the padding and assuming all message sizes are multiple of $n$.

PMAC VARIANTS FROM sPMAC: Now, we describe some variants of PMAC as instantiations of sPMAC by defining the sampling mechanism of the key $K = (\boldsymbol{\pi}, \boldsymbol{\pi}', \boldsymbol{\tau})$.

1. PMAC: We get the original PMAC [16] construction by setting $\boldsymbol{\pi} \xleftarrow{\$} \mathsf{Perm}$, $\boldsymbol{\pi}' = \boldsymbol{\pi}$, and $\boldsymbol{\tau}(i) = \gamma_i \cdot \boldsymbol{\pi}(0)$, where $\gamma_i$ is the $i$th element of the Gray code sequence [42, 85].
2. PMAC1: We get PMAC1 [85] by setting $\boldsymbol{\pi} \xleftarrow{\$} \mathsf{Perm}$, $\boldsymbol{\pi}' = \boldsymbol{\pi}$, and $\boldsymbol{\tau}(i) = 2^i \cdot \boldsymbol{\pi}(0)$, where 2 is a fixed primitive element of the Galois field $\mathrm{GF}(2^n)$.
3. Gaži et al.'s variants: In [39], Gaži et al. discussed two variants of PMAC. In both of the cases, $\boldsymbol{\pi}, \boldsymbol{\pi}' \xleftarrow{\$} \mathsf{Perm}$ and $\boldsymbol{\tau}$ is sampled independent of $\boldsymbol{\pi}, \boldsymbol{\pi}'$. The two choices of $\boldsymbol{\tau}$ are the following:
   (a) $\boldsymbol{\tau}$ is a uniform random function.
   (b) $\boldsymbol{\tau}$ is a 4-wise independent hash function.

4. Naito's variant of PMAC1: Naito proposed another variant of PMAC by setting $\boldsymbol{\pi}, \boldsymbol{\pi}' \leftarrow\$ \mathsf{Perm}$, and $\boldsymbol{\tau}(i) = 2^i \cdot L_1 \oplus 2^{3i} \cdot L_2$ where $L_1, L_2 \leftarrow\$ \mathfrak{B}$. In rest of the thesis, we call this construction NPMAC.

**N.B.** In this thesis, for the sake of simplicity, we pad all the messages (including the one whose length is a multiple of $n$). In the original PMAC$(1)$, the last message block is padded only when it is incomplete (not a multiple of $n$). In case of a complete message $m$, final input in our simplified version is same as the final input of $m\|10^{n-1}$ in the original PMAC. Moreover, the message is processed in a slightly different manner in the original PMAC$(1)$: in case of a complete message there, a constant value is xored in the final input. Xoring a constant value does not affect the collision probability. These two considerations above imply that *our analyses are directly applicable to the actual PMAC$(1)$ constructions*.

UPPER BOUND ON THE PRF ADVANTAGE OF SPMAC: Any instance of sPMAC can be viewed as an instance of Hash-then-RP, as long as $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ are sampled independently. Thus, the result of Hash-then-RP is not applicable for PMAC and PMAC1 as $\boldsymbol{\pi}' = \boldsymbol{\pi}$.

In this thesis, we consider only those instances of sPMAC that follow the Hash-then-RP paradigm where $\boldsymbol{\pi}, \boldsymbol{\pi}', \boldsymbol{\tau}$ are all sampled independently. Moreover, $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ are random permutations and hence any PMAC variant (and its underlying hash) are completely determined once we fix a distribution for the masking function $\boldsymbol{\tau}$, say $\tau$. We write $\mathsf{sPHash}_\tau$ to represent $\mathsf{sPHash}_{\boldsymbol{\pi},\tau}$ and we write $\mathsf{sPMAC}_\tau(m) := \boldsymbol{\pi}'(\mathsf{sPHash}_\tau(m))$. We can restate Proposition 2.2.5 in context of PMAC variants as follows.

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{sPMAC}_{\boldsymbol{\tau}}}(q,\ell,\sigma) \leq \mathsf{coll}_{\mathsf{sPHash}_{\boldsymbol{\tau}}}(q,\ell,\sigma) + \frac{q(q-1)}{2^{n+1}} \tag{3.2}$$

$$\leq \frac{q(q-1)}{2} \cdot \mathsf{coll}_{\mathsf{sPHash}_{\boldsymbol{\tau}}}(\ell) + \frac{q(q-1)}{2^{n+1}} \tag{3.3}$$

LOWER BOUND ON THE PRF ADVANTAGE OF SPMAC: Fix $q$ distinct messages $m_1, \ldots, m_q$ such that

$$\mathsf{coll}_{\mathsf{sPHash}_{\boldsymbol{\tau}}}(q,\ell,\sigma) = \mathsf{coll}_{\mathsf{sPHash}_{\boldsymbol{\tau}}}(m_1,\ldots,m_q).$$

In other words, the message tuple maximizes the collision probability. Now, we define a (non-adaptive) PRF distinguisher $\mathcal{A}$ for sPMAC that exploits collisions in sPMAC outputs.

1. $\mathcal{A}$ makes $2q$ queries, namely $m_1, m_1\|0^n, \ldots, m_q, m_q\|0^n$ to its oracle $\mathcal{O}$ (which is either $\mathsf{sPMAC}_{\tau}$, i.e. the real oracle, or a random function, $\rho$, i.e. the ideal oracle).
2. $\mathcal{A}$ returns 1, if for some $i \neq j$, $\mathcal{O}(m_i) = \mathcal{O}(m_j)$ as well as $\mathcal{O}(m_i\|0^n) = \mathcal{O}(m_j\|0^n)$, and 0 otherwise.

Note that, in case of real oracle, collision for $m_i$ and $m_j$ implies collision for $m_i\|0^n$ and $m_j\|0^n$ too. So, $\mathsf{Pr}(\mathcal{A}^{\mathsf{sPMAC}_{\tau}} = 1) = \mathsf{coll}_{\mathsf{sPHash}_{\tau}}(m_1, \ldots, m_q)$, whereas, $\mathsf{Pr}(\mathcal{A}^{\rho} = 1) \leq \frac{q(q-1)}{2^{2n+1}}$. So,

$$\begin{aligned}\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{sPMAC}_{\tau}}(\mathcal{A}) &\geq \mathsf{coll}_{\mathsf{sPHash}_{\tau}}(m_1, \ldots, m_q) - \frac{q(q-1)}{2^{2n+1}}. \\ &\geq \mathsf{coll}_{\mathsf{sPHash}_{\tau}}(q, \ell, \sigma) - \frac{q(q-1)}{2^{2n+1}}. \end{aligned} \tag{3.4}$$

It is clear from Eq. (3.2) and (3.4) that $\mathsf{coll}_{\mathsf{sPHash}_{\tau}}(q, \ell, \sigma)$ is a very close estimate for $\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{sPMAC}_{\tau}}$, i.e., we have

$$\mathsf{coll}_{\mathsf{sPHash}_{\tau}}(q, \ell, \sigma) - \frac{q(q-1)}{2^{2n+1}} \leq \mathbf{Adv}^{\mathsf{prf}}_{\mathsf{sPMAC}_{\tau}}(\mathcal{A}) \leq \mathsf{coll}_{\mathsf{sPHash}_{\tau}}(q, \ell, \sigma) + \frac{q(q-1)}{2^{n+1}}. \tag{3.5}$$

In other words, $\left|\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{sPMAC}_{\tau}}(\mathcal{A}) - \mathsf{coll}_{\mathsf{sPHash}_{\tau}}(q, \ell, \sigma)\right| \leq \frac{q(q-1)}{2^{n+1}}$.

### 3.1.1 Collision Analysis of sPMAC [39]

We fix two distinct messages $m := (m[1], \ldots, m[l]), m' := (m'[1], \ldots, m'[l'])$ with number of blocks $l := l_m$ and $l' := l_{m'}$ respectively. We also assume $l \leq l'$. Let $m_{\text{chop}} := (m[1], \ldots, m[l-1])$ denote the message $m$ after removing the last block. Similarly, we write $m'_{\text{chop}}$ for the message $m'$. Let

$$\mathcal{V} := \{(M, a) \mid M \in \{m, m'\}; 1 \leq a \leq l_M - 1\}$$

be called index set. For any masking function $\tau$, recall the definition of $x_\tau$ (also referred as input function) from Eq. (3.1). $x_\tau$ can be viewed as a block function defined over $\mathcal{V}$. For a masking function $\tau$, we write the multiset corresponding to all inputs for the chopped message $m_{\text{chop}}$ as

$$\mathcal{X}_\tau(m_{\text{chop}}) := \{x_\tau(m, 1), x_\tau(m, 2), \ldots, x_\tau(m, l-1)\}.$$

We similarly define $\mathcal{X}_\tau(m'_{\text{chop}})$ for the message $m'$ and $\mathcal{X}_\tau(m_{\text{chop}}, m'_{\text{chop}}) := \mathcal{X}_\tau(m_{\text{chop}}) \cup \mathcal{X}_\tau(m'_{\text{chop}})$. Note that $\mathcal{X}_\tau(m_{\text{chop}}, m'_{\text{chop}})$ actually depends on $m_{\text{chop}}$ and $m'_{\text{chop}}$.

**Definition 3.1.2** (cross-canceling masking function)**.** A masking function $\boldsymbol{\tau}$ is called cross-canceling with respect to $m_{\text{chop}}$ and $m'_{\text{chop}}$ if $\mathcal{X}_{\boldsymbol{\tau}}(m_{\text{chop}}, m'_{\text{chop}})$ is evenly repeated. Let

$$\theta_{\boldsymbol{\tau}}(m_{\text{chop}}, m'_{\text{chop}}) := \mathsf{Pr}_{\boldsymbol{\tau}}(\boldsymbol{\tau} \text{ is cross-canceling with respect to } (m_{\text{chop}}, m'_{\text{chop}})),$$

and $\theta_{\boldsymbol{\tau}}(\ell) := \max \theta_{\boldsymbol{\tau}}(m_{\text{chop}}, m'_{\text{chop}})$, where the maximum is taken over all distinct $m_{\text{chop}}, m'_{\text{chop}}$ with $l, l' < \ell$. $\theta_{\boldsymbol{\tau}}(\ell)$ is referred as the cross-cancellation probability of $\boldsymbol{\tau}$.

A proof of the following lemma is available in [39, Lemma 2]. Similar result is also proved in [61, Proposition 1], albeit under a slightly different notational setup. We give another proof here for the sake of completeness.

**Lemma 3.1.3** ([39])**.** *For any random masking $\boldsymbol{\tau}$, we have*

$$\mathsf{coll}_{\mathsf{sPHash}_{\boldsymbol{\tau}}}(\ell) \leq \theta_{\boldsymbol{\tau}}(\ell) + \frac{1}{2^n - 2\ell}.$$

*Proof.* Let $m, m'$ be two distinct messages with $|m|, |m|' \leq \ell$. Now, the event $\mathsf{sPHash}_{\boldsymbol{\tau}}(m) = \mathsf{sPHash}_{\boldsymbol{\tau}}(m')$ can be divided in the following two disjoint events:

- $A$ : $\mathsf{sPHash}_{\boldsymbol{\tau}}(m) = \mathsf{sPHash}_{\boldsymbol{\tau}}(m') \wedge \boldsymbol{\tau}$ is cross-canceling with respect to $(m_{\mathsf{chop}}, m'_{\mathsf{chop}})$
- $B$ : $\mathsf{sPHash}_{\boldsymbol{\tau}}(m) = \mathsf{sPHash}_{\boldsymbol{\tau}}(m') \wedge \boldsymbol{\tau}$ is not cross-canceling with respect to $(m_{\mathsf{chop}}, m'_{\mathsf{chop}})$

The probability of event $A$ can be bounded by $\theta_{\boldsymbol{\tau}}(m_{\mathsf{chop}}, m'_{\mathsf{chop}})$. Let us look at the event $B$. For simplicity of notation let us denote the multiset $\mathcal{X}_{\tau}(m_{\text{chop}}, m'_{\text{chop}})$ by $\mathcal{X}$. Then from Eq. (2.1) we have $\bigoplus_{x \in \mathcal{X}^o} \boldsymbol{\pi}(x) = m[l] \oplus m[l']$. Since $\mathcal{X}^o \neq \emptyset$, we can choose some $x_1 \in \mathcal{X}^o$ and bound $\mathsf{Pr}[B]$ as follows:

$$\mathsf{Pr}[B] \leq \mathsf{Pr}_{\boldsymbol{\pi}}\left[\boldsymbol{\pi}(x_1) = \oplus_{x \neq x_1} \boldsymbol{\pi}(x) \oplus m[l] \oplus m'[l']\right] \leq \frac{1}{2^n - l - l'} \leq \frac{1}{2^n - 2\ell} \tag{3.6}$$

In the first inequality we are considering $x$-values only from $\mathcal{X}^o$. The second inequality follows from probability of $\boldsymbol{\pi}(x_1)$ after we sample all other $\boldsymbol{\pi}$-values in a without replacement manner. Since we are left with exactly one choice among at least $2^n - l - l'$ many values here, we get the bound. The third inequality is obvious.

Therefore,

$$\mathsf{coll}_{\mathsf{sPHash}_{\boldsymbol{\tau}}}(m, m') \leq \theta_{\boldsymbol{\tau}}(m, m') + \frac{1}{2^n - 2\ell}.$$

We get the required result by taking maximum over all $m, m'$ such that $m \neq m'$ and $|m|, |m'| \leq \ell$ in both sides of the above inequality. □

EXTENSION OF CROSS-CANCELLATION PROBABILITY OVER $q$ MESSAGES. In [39], the idea of cross-cancellation is defined for two messages. Here, we extend the idea to more than two messages. For the sake of simplicity of notation we will write $\theta_{\boldsymbol{\tau}}(m, m')$ (and $\tau$ is cross-canceling with respect to $m, m'$) instead of $\theta_{\boldsymbol{\tau}}(m_{\text{chop}}, m'_{\text{chop}})$ (and $\tau$ is cross-canceling with respect to $m_{\mathsf{chop}}, m'_{\mathsf{chop}}$). We say $\boldsymbol{\tau}$ to be cross-canceling with respect to $m^q$ if $\boldsymbol{\tau}$ is cross-canceling with respect to $m_i, m_j$ for some $1 \leq i < j \leq q$. Let

$$\theta_{\boldsymbol{\tau}}(m^q) := \mathsf{Pr}_{\boldsymbol{\tau}}(\boldsymbol{\tau} \text{ is cross-canceling with respect to } m^q),$$

and $\theta_{\boldsymbol{\tau}}(q, \ell, \sigma) := \max \theta_{\boldsymbol{\tau}}(m^q)$, where the maximum is taken over all $q$ distinct messages each with at most $\ell - 1$ blocks, having at most $\sigma - q$ blocks altogether.

**Lemma 3.1.4.** *For any random masking $\boldsymbol{\tau}$, we have*

$$\theta_{\boldsymbol{\tau}}(q, \ell, \sigma) \leq \mathsf{coll}_{\mathsf{sPHash}_{\boldsymbol{\tau}}}(q, \ell, \sigma) \leq \theta_{\boldsymbol{\tau}}(q, \ell, \sigma) + \frac{q(q-1)}{2(2^n - 2\ell)}.$$

*Proof.* Suppose, $m_1, \ldots, m_q$ are $q$ messages for which $\theta_{\boldsymbol{\tau}}(m^q) = \theta_{\boldsymbol{\tau}}(q, \ell, \sigma)$. Let $\mathcal{T}$ denote the set of all realizable masking functions. Let $\mathcal{T}_{i,j} \subseteq \mathcal{T}$ denote the set of all cross-canceling masking functions with respect to $(m_i, m_j)$. Then, $\theta_{\boldsymbol{\tau}}(m^q) := \mathsf{Pr}(\boldsymbol{\tau} \in \cup_{i<j} \mathcal{T}_{i,j})$. Let $m'_i = m_i \| 0^n$ for $1 \leq i \leq q$. Now, for any $\tau \in \mathcal{T}_{i,j}$, $\mathsf{sPHash}_\tau(m'_i) = \mathsf{sPHash}_\tau(m'_j)$ holds (also denoted as $\mathsf{coll}_{i,j}$). So,

$$\theta_{\boldsymbol{\tau}}(q, \ell, \sigma) = \mathsf{Pr}(\boldsymbol{\tau} \in \cup_{i<j} \mathcal{T}_{i,j}) \leq \mathsf{Pr}(\cup_{i<j} \mathsf{coll}_{i,j}) \leq \mathsf{coll}_{\mathsf{sPHash}_{\boldsymbol{\tau}}}(q, \ell, \sigma).$$

Now, we show the upper bound. We fix $q$ distinct messages $m_1, \ldots, m_q$ such that $\mathsf{coll}_{\mathsf{sPHash}_{\boldsymbol{\tau}}}(m^q) = \mathsf{coll}_{\mathsf{sPHash}_{\boldsymbol{\tau}}}(q, \ell, \sigma)$. Let $\mu := \mathsf{Pr}(\boldsymbol{\tau}$ is cross-canceling with respect to $m^q)$.

$$\begin{aligned}
\mathsf{coll}_{\mathsf{sPHash}_{\boldsymbol{\tau}}}(m^q) &\leq \mu + \sum_{\tau \in \mathcal{T} \setminus \cup_{i<j} \mathcal{T}_{i,j}} \mathsf{Pr}(\exists i < j, \boldsymbol{\pi}^{\oplus}(\mathcal{X}^o_\tau(m_i, m_j)) = m_i[l_i] \oplus m_j[l_j] \ \wedge \ \boldsymbol{\tau} = \tau) \\
&\leq \mu + \sum_{\tau \in \mathcal{T} \setminus \cup_{i<j} \mathcal{T}_{i,j}} \mathsf{Pr}(\exists i < j, \boldsymbol{\pi}^{\oplus}(\mathcal{X}^o_\tau(m_i, m_j)) = m_i[l_i] \oplus m_j[l_j]) \times \mathsf{Pr}(\boldsymbol{\tau} = \tau) \\
&\leq \mathsf{Pr}(\boldsymbol{\tau} \text{ is cross-canceling with respect to } m^q) + \frac{q(q-1)}{2(2^n - 2\ell)},
\end{aligned}$$

where the last inequality is obtained by conditioning on the output of $\boldsymbol{\pi}$ on all elements in $\mathcal{X}^o_\tau(m_i, m_j)$ except one. Note that this is possible only because $\mathcal{X}^o_\tau(m_i, m_j) \neq \emptyset$ since $\tau$ is not a cross-canceling function. □

**Corollary 3.1.5.** *For any random masking function $\boldsymbol{\tau}$, we have*

$$\theta_{\boldsymbol{\tau}}(q,\ell,\sigma) - \frac{q^2}{2^{2n+1}} \le \mathbf{Adv}^{\mathsf{prf}}_{\mathit{sPMAC}_{\boldsymbol{\tau}}}(q,\ell,\sigma) \le \theta_{\boldsymbol{\tau}}(q,\ell,\sigma) + \frac{q(q-1)}{2(2^n-2\ell)} + \frac{q(q-1)}{2^{n+1}}$$
$$\le \frac{q(q-1)}{2}\cdot\theta_{\boldsymbol{\tau}}(\ell) + \frac{q(q-1)}{2(2^n-2\ell)} + \frac{q(q-1)}{2^{n+1}}.$$

Corollary 3.1.5 follows from Eq. (3.2) and Lemma 3.1.4 in combination with the observation that $\theta_{\boldsymbol{\tau}}(q,\ell,\sigma) \le \binom{q}{2}\theta_{\boldsymbol{\tau}}(\ell)$.

To achieve $O(q^2/2^n)$ bound, it is sufficient to show $\theta_{\boldsymbol{\tau}}(\ell) \le c/2^n$ for some constant $c$ (should be independent of $\ell$). Sometimes, it is possible to show this for a range of values of $\ell$ instead of all values of $\ell$. Sometimes, it might be difficult to obtain $\ell$-free bound for $\theta_{\boldsymbol{\tau}}(\ell)$. However, it might be possible to show $\ell$-free bound for the $\theta_{\boldsymbol{\tau}}(q,\ell,\sigma)$ by considering all $q$ messages together. In this case, first part of the above corollary could be used to obtain an $\ell$-free security bound. When $\ell \le 2^{n-2}$, Corollary 3.1.5 is simplified to

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{sPMAC}_{\boldsymbol{\tau}}} \le \frac{q^2}{2}\cdot\left(\theta_{\boldsymbol{\tau}}(\ell) + \frac{3}{2^n}\right). \tag{3.7}$$

SOME EXAMPLES OF CROSS-CANCELLATION PROBABILITY: We list some known results on the cross-cancellation probability of some masking functions.

1. In [39], Gaži et al. show the following bounds on cross-cancellation probability:
   (a) If $\boldsymbol{\tau}$ is a uniform random function, then $\theta_{\boldsymbol{\tau}}(\ell) \le 2^{1-n}$.
   (b) If $\boldsymbol{\tau}$ is a 4-wise independent hash function, then $\theta_{\boldsymbol{\tau}}(\ell) \le 2^{2-n}$.
2. For the masking function $\boldsymbol{\tau}(i) = 2^i \cdot L_i \oplus 2^{3i}\cdot L_2$, Naito proved the following result [70, Section 4.2: Bounding $p^2_{\text{coll}}$] whenever $L_1, L_2 \xleftarrow{\$} \mathfrak{B}$:

$$\theta_{\boldsymbol{\tau}}(\ell) \le 2^{2-n}, \text{ while } \ell \le 2^{n/2}. \tag{3.8}$$

## 3.2 An Observation on Naito's PMAC Variant

In this section, we revisit a claim of [70] regarding the cross cancellation probability of two powering-up maskings.

### 3.2.1 A Flaw and Its Effect on the Proof of NPMAC [70]

As mentioned in section 3.1, Naito proved Eq. (3.8) with respect to the cross cancellation probability of two powering-up maskings. The proof relies on five cases [70, Section 4.2: Type-1 to Type-5]. The most crucial and general of these cases is Type-5. Naito made the following claim with respect to this case.

CLAIM IN [70, Type-5 case in Section 4.2]: *The following system of equations, denoted* $(\mathcal{E})$*, in* $L_1$ *and* $L_2$ *such that* $\{i_1, i_2\} \neq \{i_3, i_4\}$*,*

$$(2^{i_1} \oplus 2^{i_2})L_1 \oplus (2^{3i_1} \oplus 2^{3i_2})L_2 = c_1$$
$$(2^{i_3} \oplus 2^{i_4})L_1 \oplus (2^{3i_3} \oplus 2^{3i_4})L_2 = c_2$$

*has rank two (i.e. the equations are always linearly independent).*

The author argues as follows: If the equations are not linearly independent then $2^{i_1} \oplus 2^{i_2}\mathsf{eq}2^{i_3} \oplus 2^{i_4}$ and $2^{3i_1} \oplus 2^{3i_2}\mathsf{eq}2^{3i_3} \oplus 2^{3i_4}$. From this, by using simple calculation, one can obtain $i_1\mathsf{eq}i_2\mathsf{eq}i_3\mathsf{eq}i_4$. This leads to a contradiction of the assumption that $\{i_1, i_2\} \neq \{i_3, i_4\}$, and hence the linear dependence assumption is false. The author thus concludes that the system $(\mathcal{E})$ will always have rank 2. In other words, for fixed $i_1, i_2, i_3, i_4$, the system has a unique solution for the pair $(L_1, L_2)$.

FLAW IN THE ARGUMENT: Unfortunately, linear dependency and consistency of the two equations over $\mathrm{GF}(2^n)$ can be equivalently written as

$$2^{i_1} \oplus 2^{i_2} = c \cdot (2^{i_3} \oplus 2^{i_4}) \tag{3.9}$$
$$2^{3i_1} \oplus 2^{3i_2} = c \cdot (2^{3i_3} \oplus 2^{3i_4}) \tag{3.10}$$

where $c_2 = c \cdot c_1$. Clearly, whenever $c \neq 1$, the claim on $(\mathcal{E})$ is not correct. In [70], the author only considers the $c = 1$ case. Next, we show a concrete counterexample for this.

COUNTEREXAMPLE FOR THE RANK CLAIM: First, we can rewrite Eq. (3.9) and (3.10) as

$$(2^{i_1} \oplus 2^{i_2}) \cdot (2^{3i_3} \oplus 2^{3i_4}) = (2^{i_3} \oplus 2^{i_4}) \cdot (2^{3i_1} \oplus 2^{3i_2}) \tag{3.11}$$

We show a counterexample for $n = 16$. Similar examples can be constructed for other values of $n$ as well. Consider the field $\mathrm{GF}(2^{16})$ generated by $x = 2$ with multiplication defined by the minimal polynomial $x^{16} + x^5 + x^3 + x + 1$. Using simple algebra one can

show that $i_1 = 1$, $i_2 = 24$, $i_3 = 14$ and $i_4 = 18$ satisfies Eq. (3.11). Plugging in the same values in Eq. (3.10), one can get

$$c = 2^{12} \oplus 2^9 \oplus 2^8 \oplus 2^7 \oplus 2^6 \oplus 2^5 \oplus 2^2 \oplus 2 \oplus 1.$$

This proves that the system $(\mathcal{E})$ can be of rank 1 as well. And, the number of such $i_1, i_2, i_3, i_4$ is at least 1. Whereas, Naito incorrectly argues that the number of such quadruples is 0.

EFFECT ON THE CURRENT PROOF: The system $(\mathcal{E})$ is fixed once we fix the quadruple $(i_1, i_2, i_3, i_4)$. In [70], the number of $i_1, i_2, i_3, i_4$ indices corresponding to the system $(\mathcal{E})$ is bounded by $O(\ell^2)$ which can be further bounded by $O(2^n)$ (since $\ell \leq 2^{n/2}$). This bound is fine as long as the rank of system $(\mathcal{E})$ is 2, as this will mean that we get an overall cross-cancellation probability bound of $O(2^{-n})$. However, given the evidence that $(\mathcal{E})$ can have rank 1, a bound of $O(\ell^2)$ is not desirable, as it will result in an overall cross-cancellation probability bound of $O(\ell^2/2^n)$ which is worse than $O(\ell/2^n)$ bound for the existing PMAC.

### 3.2.2 Further Discussion on the Security of NPMAC

From previous discussions, it is clear that the question of $\ell$-free security for NPMAC is far from resolved. Going by the existing proof strategy [70], we get $\theta_{\boldsymbol{\tau}}(\ell) = O(\ell^2/2^n)$ bound. Looking ahead momentarily, Proposition 3.3.3 shows that we can achieve $O(\ell/2^n)$ for any $O(2^{-n})$-AXU masking function. This result also applies to NPMAC as the two powering-up maskings is obviously a $O(2^{-n})$-AXU. But, this is as far as we could reach. In what follows, we discuss some bottlenecks in resolving this question one way or another.

Let us denote the number of quadruples satisfying Eq. (3.11) by $N$. Our counterexample in the previous subsection shows that $N = \Omega(1)$ and due to Proposition 3.3.3 we can give a trivial upper bound of $N = O(\ell)$. Now, to prove or disprove the $\ell$-free security claim we need an exact estimate of $N$.

We could neither construct a counterexample where $N = \Omega(\ell)$, nor show that $N = O(1)$. This indeed looks like a hard problem requiring an involved analysis of the properties of $\mathrm{GF}(2^n)$. Interestingly, a similar hardness remains for PMAC1 as well [39, 61] that involves a study of the additive subgroups (and their cosets) of $\mathrm{GF}(2^n)$.

Note that, $(\mathcal{E})$ is a simplified version of the actual system of equation that we have to analyze. In the actual system, $c_1$ and $c_2$ are not arbitrary. In fact, for some $M_1, M_2, M_3, M_4 \in$

$\{m, m'\}$,

$$c_1 = M_1[i_1] \oplus M_2[i_2],\ c_2 = M_3[i_3] \oplus M_4[i_4],$$

$$\text{and thus, } c = (M_3[i_3] \oplus M_4[i_4]) \cdot (M_1[i_1] \oplus M_2[i_2])^{-1}.$$

Clearly the simplification, though sufficient to discuss the flaw, could possibly degrade the bound as we count some inconsistent systems of equations as well. We say that a quadruple $(i_1, i_2, i_3, i_4)$ is *valid* if the resulting system of equation $(\mathcal{E})$ is consistent. At the moment, we do not see any approach to exploit the exact nature of $c$ to get a better estimate for the number of valid quadruples satisfying Eq. (3.11).

In summary, to prove or disprove the $\ell$-free security of NPMAC, we have to bound:

*The number, $N$, of valid quadruples $(i_1, i_2, i_3, i_4)$ that satisfy*

$$(2^{i_1} \oplus 2^{i_2}) \cdot (2^{3i_3} \oplus 2^{3i_4}) = (2^{i_3} \oplus 2^{i_4}) \cdot (2^{3i_1} \oplus 2^{3i_2}).$$

We leave it as an open problem to find an exact estimate for $N$, which in turn gives tight security bound for NPMAC. In fact, even a sub-optimal bound better than $\Omega(1)$ (in case of lower bound) or $O(\ell)$ (in case of upper bound), say in the order of a slowly growing function of $\ell$, could be a great improvement.

## 3.3 Relaxing the Security Precondition for sPMAC

Gaži et al. [39] showed that a 4-wise independent masking function is sufficient to achieve $\ell$-free security bound up to $\ell \leq 2^{n/2}$. In this section, we relax the 4-wise independence condition to a weaker notion. Our relaxed notion of universality is inspired by the flaw discovered in section 3.2.

### 3.3.1 2-wise Almost XOR Universal Hash Function

We extend the definition of AXU hash functions to jointly consider two pairs of messages and their hash output differences.

**Definition 3.3.1** (2-wise AXU)**.** A hash function $H$ is called $\epsilon$ 2-wise AXU (or $\epsilon$-2AXU) if for any distinct $\{m_1, m_2\}, \{m_3, m_4\}$ and $\delta_1, \delta_2 \in \mathfrak{B}$, we have

$$\mathsf{Pr}(H(K, m_1) \oplus H(K, m_2) = \delta_1 : K \xleftarrow{\$} \mathcal{K}) \leq \epsilon,$$
$$\mathsf{Pr}(H(K, m_1) \oplus H(K, m_2) = \delta_1, H(K, m_3) \oplus H(K, m_4) = \delta_2 : K \xleftarrow{\$} \mathcal{K}) \leq \epsilon^2.$$

Clearly, any $\epsilon$-2AXU hash function is also an $\epsilon$-AXU hash function. We usually expect $\epsilon = O(1/2^n)$ and hence the joint probability for the two linear equations is $O(1/2^{2n})$.

Mennink defined a very close variant, called $\text{AXU}_3$, in [64]. In that definition $m_3 = m_1$ (and hence $m_2 \neq m_4$). He also gave an example of $\text{AXU}_3$ (and its higher order variants) using finite field arithmetic.

2AXU IS STRICTLY WEAKER THAN 4-WISE INDEPENDENCE: It is easy to see that a 4-wise independent hash function is $2^{-n}$-2AXU. However, every 2AXU hash function need not be 4-wise independent. Consider the following example due to Naito [71, 72]. Similar example can also be found in [64] as an example of $\text{AXU}_4$ (see [64] for definition) hash function.

**Example 3.1.** *Let $L_1, L_2, L_3 \xleftarrow{\$} \mathfrak{B}$. For a fixed primitive element 2 of $\text{GF}(2^n)$ and any $i$, let us define*

$$\boldsymbol{\tau}(i) := 2^i \cdot L_1 \oplus 2^{2i} \cdot L_2 \oplus 2^{3i} \cdot L_3.$$

*It can be easily shown that $\boldsymbol{\tau}$ is $O(2^{-n})$-2AXU. However, for any distinct $i_1, i_2, i_3, i_4$ and $y_1, y_2, y_3, y_4$ we cannot get probability $1/2^{4n}$ for the following event:*

$$2^{i_j} \cdot L_1 \oplus 2^{2i_j} \cdot L_2 \oplus 2^{3i_j} \cdot L_3 = y_j, \forall j \in \{1, 2, 3, 4\}.$$

*In other words, the above masking function is not 4-wise independent.*

*Remark* 3.3.2. The two powering-up maskings used in [70] is not $2^{-n}$-2AXU hash. However, Naito addressed this issue in [72] and proposed an alternate "three powering-up maskings" which is same as our example 3.1. He has given a dedicated proof for this construction whereas our proof for this one follows from our general treatment of 2AXU hash functions.

### 3.3.2 PRF Security of sPMAC

From Corollary 3.1.5, we know that the PRF advantage of sPMAC is bounded by the cross-cancellation probability of the underlying masking function. We have closely revisited all the existing proof strategies for upper bounding the cross-cancellation probability and have found a unified way to present all these proofs. This approach also helps in understanding the requirements from the masking function for achieving length independent PRF advantage. We state two results unifying the proofs of existing and some new constructions. The proofs of these results is postponed to section 3.4.

**Proposition 3.3.3.** *Suppose $\boldsymbol{\tau}$ is $\epsilon$-AXU. Then, $\theta_{\boldsymbol{\tau}}(\ell) \leq 2\ell\epsilon$. Hence, by using Corollary 3.1.5, we have*

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{sPMAC}_{\boldsymbol{\tau}}}(q,\ell,\sigma) \leq q^2\ell\epsilon + \frac{q^2}{2(2^n-2\ell)} + \frac{q^2}{2^{n+1}}.$$

Proposition 3.3.3 gives the security bound for PMAC and PMAC1 when the outer permutation is replaced by an independent random permutation and the masking key is sampled independently. A dedicated analysis is required when we consider outer permutation same as the inner one and the masking key is derived from the permutation, like the original PMAC and PMAC1.

The bound in Proposition 3.3.3 is not $\ell$-free as it has $q^2\ell\epsilon$ term (which came due to cross-cancellation probability). In the following result, we show how we can improve this term if we apply a stronger masking function. Gaži et al. [39] proved a similar result for 4-wise independent masking function. However, we can easily extend their result to the weaker notion of 2AXU masking function.

**Theorem 3.3.4.** *Suppose $\boldsymbol{\tau}$ is $\epsilon$-2AXU. Then, $\theta_{\boldsymbol{\tau}}(\ell) \leq \max\{2\epsilon, 4\ell^2\epsilon^2\}$. Hence, by using Corollary 3.1.5, we have*

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{sPMAC}_{\boldsymbol{\tau}}}(q,\ell,\sigma) \leq \max\{q^2\epsilon, 2q^2\ell^2\epsilon^2\} + \frac{q^2}{2(2^n-2\ell)} + \frac{q^2}{2^{n+1}}.$$

*So, when $\epsilon = 1/2^n$ and $\ell \leq 2^{\frac{n-1}{2}}$ then*

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{sPMAC}_{\boldsymbol{\tau}}}(q,\ell,\sigma) \leq \frac{5q^2}{2^{n+1}}.$$

Theorem 3.3.4 also works (up to $\ell \leq 2^{n/2}$) for a uniform random masking function and 4-wise independent masking function as these are also $1/2^n$-2AXU hash functions. However, in case of uniform random function, a more precise analysis (as shown in [39]) gives $\theta_{\boldsymbol{\rho}}(\ell) \leq 2/2^n$ for all values of $\ell$.

*Remark* 3.3.5. Our result is a bit stronger than the result proved in [39] as every 2AXU hash function need not be 4-wise independent hash function.

*Remark* 3.3.6. Theorem 3.3.4 gives an alternate proof of $\ell$-free security for Naito's updated variant [72] with three powering up masking (see example 3.1).

## 3.4 Proof of Theorem 3.3.4

Before we delve into the proofs of Proposition 3.3.3 and Theorem 3.3.4, we describe a graph-based description of input collisions that would help us to have some visual presentation of cross-canceling masking function.

### 3.4.1 Input Collision Graph and Covering Bound Lemma

GRAPH NOTATIONS: For a set $V$, let $[V]^2$ denote the set of all doubleton subsets of $V$. So, size of the set $[V]^2$ is $\binom{|V|}{2} := |V|(|V|-1)/2$. A graph $G$ is a pair $(V, E)$ where $E \subseteq [V]^2$. We call $V$ and $E$ the vertex and edge set of the graph, respectively. We also denote $V(G)$ and $E(G)$ to denote the vertex set and edge set of the graph $G$, respectively. An edge is an element $\{u, v\} \in E$ and we also say that $u$ is adjacent to $v$. Given a graph $G = (V, E)$ and a subset $V' \subseteq V$, the subgraph restricted at $V'$, denoted as $G(V')$, has vertex set $V'$ and the edge set $[V']^2 \cap E$. A path from $u$ to $v$ of length $t$ is a sequence of distinct elements $(w_0 := u, w_1, \ldots, w_t := v)$ such that $w_{i-1}$ is adjacent to $w_i$ for all $i \in [t]$. A component $C$ (or connected component) is a subset of $V$ such that for every $u, v \in C$ either $u = v$ or there is a path from $u$ to $v$. A component $C$ of a graph $G$ is called clique if all pairs of the components are adjacent. We call a graph $G$ *evenly partitioned* if all components of $G$ have even sizes.

INPUT COLLISION GRAPH: Recall the index set $\mathcal{V} := \{(M, a) \mid M \in \{m, m'\}; 1 \leq a \leq l_M - 1\}$ for two distinct messages $m$ and $m'$ of length $l = l_m$ and $l' = l_{m'}$, respectively, such that $l \leq l'$. To each masking function $\tau$, we associate a collision graph $G_\tau$ with the vertex set $\mathcal{V}$ such that any two vertices $(M_1, a_1)$ and $(M_2, a_2)$ are said to be *adjacent* if $x_\tau(M_1, a_1) = x_\tau(M_2, a_2)$. So an input collision graph is always disjoint union of cliques.

A graph $G'$ over $\mathcal{V}$ is called $\tau$-realizable if there is a realizable masking function $\tau$ such that $G_\tau = G'$. Let $\mathcal{G}$ be the set of all such realizable graphs. Among all realizable graphs, we are interested in some special graphs, namely evenly partitioned graph. Let $\mathcal{G}_{\text{even}}$ be the set of all realizable graphs which are evenly partitioned. The following observation is straightforward from the definition of cross-canceling masking function.

*Claim* 3.4.1. A masking function $\boldsymbol{\tau}$ is cross-canceling if and only if the induced input collision graph $G_{\boldsymbol{\tau}}$ is evenly partitioned.

Due to Corollary 3.1.5, it is now sufficient to bound the probability to realize any evenly partitioned graph (equivalent to realizing a cross-canceling masking function). Now, we identify a subset of vertices for which restricted subgraph over that subset is evenly partitioned whenever the graph is evenly partitioned. Let

$$\mathcal{V}^{=} := \{(M, a) : M \in \{m, m'\},\ a \leq l, l',\ m[a] = m'[a]\}.$$

So, $(m, a) \in \mathcal{V}^{=}$ if and only if $(m', a) \in \mathcal{V}^{=}$. For any such $(m, a)$, we obviously have $x_\tau(m, a) = x_\tau(m', a)$ for all masking functions $\tau$ (not necessarily cross-canceling masking function). Hence, for any realizable input collision graph $G_\tau$, $\{(m, a), (m', a)\}$ is an

edge of the graph and we call those edges *vertical* (all other edges will be non-vertical). On the other hand, if $(m, a) \notin \mathcal{V}^{=}$ then $(m, a)$ and $(m', a)$ are not adjacent whenever these are defined. Let $\mathcal{V}^{\neq} := \mathcal{V} \setminus \mathcal{V}^{=}$ and

$$I^{\neq} := \{a : \text{ either } (m, a) \in \mathcal{V}^{\neq} \text{ or } (m', a) \in \mathcal{V}^{\neq}\}.$$

We can rewrite the set $I^{\neq}$ as union of the interval $[l + 1, l']$ (can be the empty set) and $\{a : a \leq l, l' \text{ and } m[a] \neq m'[a]\}$. As $m \neq m'$, we have $\mathcal{V}^{\neq} \neq \emptyset$. Given any graph $G$ we denote $G^{\neq} := G(\mathcal{V}^{\neq})$, the subgraph restricted on the set of vertices $\mathcal{V}^{\neq}$.

Now any connected component of $G_\tau$ consists of a connected component of $G_\tau^{\neq}$ with some additional pairs of vertices from $\mathcal{V}^{=}$. Hence, we have the following result.

*Claim* 3.4.2. For all masking functions $\boldsymbol{\tau}$, $G_{\boldsymbol{\tau}}$ is evenly partitioned if and only if $G_{\boldsymbol{\tau}}^{\neq}$ is evenly partitioned.

Now, we explain a method by which we can obtain an upper bound on the cross-canceling probability $\theta_{\boldsymbol{\tau}}(\ell)$ or $\theta_{\boldsymbol{\tau}}(q, \ell, \sigma)$. Let $\mathcal{G}_{\text{even}}^{\neq}$ be the collection of all evenly partitioned realizable graphs over the vertex set $\mathcal{V}^{\neq}$. Due to above claim, this is same as the collection of all restricted subgraphs with vertex set $\mathcal{V}^{\neq}$ of all evenly partitioned realizable graphs.

**Definition 3.4.3** (covering collection of edges)**.** Let $\mathcal{I}$ be some index set such that for every $i \in \mathcal{I}$ we have an edge set $E_i \subseteq [\mathcal{V}^{\neq}]^2$. The collection $\mathcal{E} := \{E_i : i \in \mathcal{I}\}$ is said to cover evenly partitioned graphs if for all $G \in \mathcal{G}_{\text{even}}^{\neq}$, there exists $i := i_G \in \mathcal{I}$ such that $E_i \subseteq E(G)$.

For any edge $e := \{(M_1, a_1), (M_2, a_2)\} \in [\mathcal{V}]^2$, we say that event $e(\boldsymbol{\tau})$ holds if

$$\boldsymbol{\tau}(a_1) \oplus \boldsymbol{\tau}(a_2) = c_e := M_1[a_1] \oplus M_2[a_2].$$

We extend the above definition to an edge set $E$ as follows: An event $E(\boldsymbol{\tau})$ holds if for all edges $e \in E$, $e(\boldsymbol{\tau})$ holds. All these events are defined based on the randomness of $\boldsymbol{\tau}$ only and we simply write $\mathsf{Pr}(e)$ or $\mathsf{Pr}(E)$ to denote the probability that the corresponding event holds under the randomness of $\boldsymbol{\tau}$.

**Lemma 3.4.4** (Covering Bound Lemma)**.** *Suppose $\{E_i : i \in \mathcal{I}\}$ covers evenly partitioned graphs, then we have*

$$\mathsf{Pr}_{\boldsymbol{\tau}}(\boldsymbol{\tau} \textit{ is cross-canceling with respect to } (m, m')) \leq \sum_{i \in \mathcal{I}} \mathsf{Pr}(E_i)$$

*Proof.* Let $\mathcal{T}^*$ denote the set of all cross-canceling masking functions with respect to $(m, m')$. For every $E_i$, let $\mathcal{T}_i$ denote the set of all masking function $\tau$ such that $E_i \subseteq$

$E(G_\tau^{\neq})$. Now, we claim that $\mathcal{T}^* \subseteq \cup_i \mathcal{T}_i$. For any $\tau \in \mathcal{T}^*$, $G_\tau$ is an evenly partitioned graph and hence (using Claim 3.4.2) for some $i$, $E_i \subseteq E(G_\tau^{\neq}) \subseteq E(G_\tau)$. Thus, $\tau \in \mathcal{T}_i$. So the claim holds. The result follows from union bound. □

### 3.4.2 Proof of Proposition 3.3.3

Let $i$ be the smallest element in $I^{\neq}$. We use shorthand notation $e_i(v)$ and $e'_i(v)$ to denote edges $\{(m,i),v\}$ and $\{(m',i),v\}$, respectively, whenever these are defined. Let $\mathcal{V}_i^{\neq} := \mathcal{V}^{\neq} \setminus \{(m,i),(m',i)\}$.

As $(m',i)$ has an edge for any evenly partitioned graph $G \in \mathcal{G}^{\neq}_{\text{even}}$, there must exist $(M,j)$ with $j > i$ and $M \in \{m,m'\}$ such that $(m',i)$ is adjacent to $(M,j)$. So, we define the following collection of edge sets of size one.

$$\mathcal{E}_i := \{E_v := e'_i(v) : v \in \mathcal{V}_i^{\neq}\}.$$

From the above discussion, it is clear that this covers all evenly partitioned graphs. Now, using the fact that $\boldsymbol{\tau}$ is $\epsilon$-AXU, we have $\mathsf{Pr}(E_{(M,j)}) = \mathsf{Pr}(\boldsymbol{\tau}(i) \oplus \boldsymbol{\tau}(j) = m'[i] \oplus M[j]) \leq \epsilon$ (since $j \neq i$). So, using the covering bound lemma (Lemma 3.4.4) we have

$$\mathsf{Pr}_{\boldsymbol{\tau}}(\boldsymbol{\tau} \text{ is cross-canceling with respect to } (m,m')) \leq \sum_{v \in \mathcal{V}_i^{\neq}} \mathsf{Pr}(E_v) \leq (l+l')\epsilon.$$

As $l, l' \leq \ell$, we have $\theta(\ell) \leq 2\ell\epsilon$. This completes the proof. □

### 3.4.3 Resuming the Proof of Theorem 3.3.4

Here, we first assume that $|I^{\neq}| > 2$, and we denote the first, second and third smallest elements of $I^{\neq}$ as $i_1$, $i_2$ and $i_3$, respectively. For $1 \leq j \leq 3$, $\mathcal{V}_j^{\neq} := \mathcal{V}^{\neq} \setminus \{(m,i_j),(m',i_j)\}$, and we use shorthand notation $e_j(v)$ and $e'_j(v)$ to denote edges $\{(m,i_j),v\}$ and $\{(m',i_j),v\}$, respectively, whenever these are edges over $\mathcal{V}$ (they may not be edge as some of the vertices may not be present in $\mathcal{V}$).

In the previous proof for AXU masking function, edge sets are singleton and hence the probability for any such edge set can be at best $O(1/2^n)$ (as we deal with a single equation). Now, we are considering doubleton edge sets, hoping that probability to realize any edge set is about $O(1/2^{2n})$ (as we assume stronger masking function), to achieve better security. Consider the following collections of doubleton edge sets:

1. $\mathcal{E}_1 := \{\{e'_1(M,i_2), e'_3(v)\} : v \in \mathcal{V}_3^{\neq}, M \in \{m,m'\}\}$,

2. $\mathcal{E}_2 := \{\{e'_1(M_1, j_1), e'_2(M_2, j_2)\} : (M_1, j_1) \in \mathcal{V}_1^{\neq} \cap \mathcal{V}_2^{\neq}, (M_2, j_2) \in \mathcal{V}_2^{\neq}\}$.

We claim that the collection $\mathcal{E} := \mathcal{E}_1 \cup \mathcal{E}_2$ is a covering collection of edges. Fix any evenly partitioned graph $G$ over $\mathcal{V}^{\neq}$. The vertex $(m', i_1)$ should be adjacent to some other vertex.

CASE 1: Suppose, $(m', i_1)$ is adjacent to $(M, i_2)$ then the vertex $(m', i_3)$ should be adjacent to $(M, j)$ for some $j \neq i_3$. So, we can use an appropriate edge set from $\mathcal{E}_1$.

CASE 2: Suppose, $(m', i_1)$ is adjacent to $(M, j)$ for some $M \in \{m, m'\}$ and $j \geq i_3$. Then, $(m', i_2)$ should be adjacent to $(M, j)$ for some $j \neq i_2$. So, we can use an appropriate collection from $\mathcal{E}_2$.

Thus, $\mathcal{E}$ is indeed a covering collection of edges. Now, we fix any edge set $E := \{e'_1(M_1, i_2), e'_3(M_2, j)\} \in \mathcal{E}_1$ where $j \neq i_3$. Then, for $c_1 = m'[i_1] \oplus M_1[i_2]$ and $c_2 = m'[i_3] \oplus M_2[j]$, we have

$$\mathsf{Pr}(E) = \mathsf{Pr}(\boldsymbol{\tau}(i_1) \oplus \boldsymbol{\tau}(i_2) = c_1, \boldsymbol{\tau}(i_3) \oplus \boldsymbol{\tau}(j) = c_2) \leq \epsilon^2,$$

where the inequality follows from the definition of $\epsilon$-2AXU. Similarly, for any edge set $E \in \mathcal{E}_2$, one can show that $\mathsf{Pr}(E) \leq \epsilon^2$. Note that $|\mathcal{E}_1| \leq 2(l + l')$ and $|\mathcal{E}_2| \leq (l + l' - 2) \cdot (l + l' - 4)$. So, $|\mathcal{E}| \leq (l + l')^2 \leq 4\ell^2$. By using the covering bound Lemma (Lemma 3.4.4), we have

$$\mathsf{Pr}_{\boldsymbol{\tau}}(\boldsymbol{\tau} \text{ is cross-canceling with respect to } (m, m')) \leq \sum_{E \in \mathcal{E}} \mathsf{Pr}(E) \leq 4\ell^2\epsilon^2.$$

Now, the only remaining case is $|I^{\neq}| = 2$ ($|I^{\neq}|$ cannot be 1 as this would contradict the existence of evenly partitioned graph). In this case, we have only two possibilities of evenly partitioned graphs, each occurring with at most $\epsilon$ probability (using $\epsilon$-2AXU). So, we have

$$\mathsf{Pr}_{\boldsymbol{\tau}}(\boldsymbol{\tau} \text{ is cross-canceling with respect to } (m, m')) \leq 2\epsilon.$$

The result follows by combining the two cases for $|I^{\neq}|$. □

## 3.5 PMAC2: A Simple Variant of PMAC1

Now we propose a simple variant of PMAC1 which we call PMAC2 (see Fig. 3.5.1). Given any message $m' \in \{0,1\}^*$ we append a bit 1 followed by a smallest sequence of zeros so that the padded message has size multiple of $n$. Let $m := (m[1], \ldots, m[l]) \in$

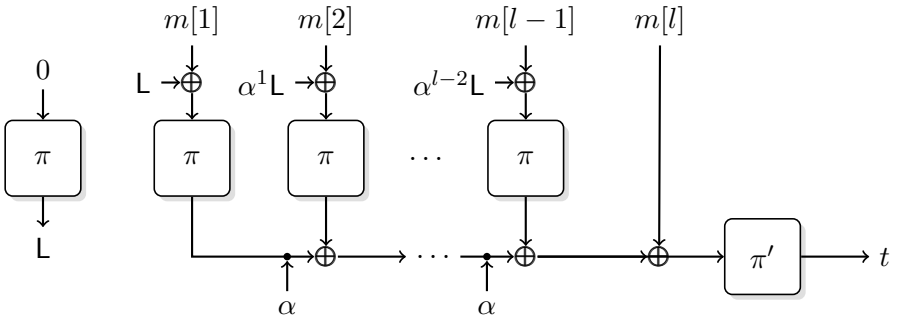


**Figure 3.5.1:** PMAC2: A message $m$ is padded with $10^*$ to get $m[1]||m[2]||\dots m[l]$ where each $m[i]$ is an $n$-bit string. L is obtained as $\boldsymbol{\pi}(0)$ where $\boldsymbol{\pi} \leftarrow\$$ Perm. Here $\alpha$ is a primitive element of the field $GF(2^n)$.

$\mathfrak{B}^l$ be a padded message. As it is an injective padding, we define the construction after the padding. Let $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ be two independent random permutations (for a real construction we use a block cipher instantiated by two independent keys). We compute the final output of PMAC2$(m)$ as follows:

PMAC2$_{\boldsymbol{\pi},\boldsymbol{\pi}'}(m)$

1 : **Input:** $m = m[1]\| \dots \|m[l]$

2 : $L \leftarrow \boldsymbol{\pi}(0)$

3 : **for** $i = 1$ *to* $l-1$, **do** $x[i] \leftarrow m[i] \oplus \alpha^{i-1} \cdot L$

4 : $H_{\boldsymbol{\pi}} \leftarrow m[l] \oplus \bigoplus_{i=1}^{l-1} \alpha^{l-i-1}\boldsymbol{\pi}(x[i])$

5 : **return** $\boldsymbol{\pi}'(H_{\boldsymbol{\pi}})$

**Theorem 3.5.1.** *(Main Theorem: Bound for Hash Collision Probability of PMAC2)*

$$\mathsf{coll}_H(q,\ell,\sigma) = \frac{q^2+\sigma}{2^n} + \mu$$

*where* $\mu \leq \begin{cases} \frac{q}{2^{n/2}} & \text{if } \ell \leq 2^{n/4} \\ \frac{\sigma^{1.5}}{2^n} & \text{if } 2^{n/4} < \ell \leq 2^{n-2}. \end{cases}$

We prove this theorem in the next subsection. The PRF-advantage of our construction will follow from hash-then-prp result:

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{PMAC2}}(q,\ell,\sigma) \leq \mathsf{coll}_H(q,\ell,\sigma) + \frac{q^2}{2^{n+1}}.$$

*Remark* 3.5.2. The original proof for PMAC works perfectly in the case of PMAC2 and hence the security of PMAC2 is also bounded by the security bound of PMAC. Our

result gives a different bound of PMAC2 which essentially gives tighter bounds in the cases of $\ell < 2^{n/4}$ and $\ell \geq 2^{n/4}$ such that $\ell < q$. In all other cases we can take the usual bound for PMAC. To be precise, we can always choose the minimum between the usual bound for PMAC and the bound obtained here.

### 3.5.1 Proof of Theorem 3.5.1

Let $m^q = (m_1, \ldots, m_q)$ be a $q$-tuple of distinct messages. Let $\ell_i = \|m_i\|$, $\ell := \max_i \ell_i$ and $\sigma := \sum_i \ell_i$. For simplicity we will write $H(m)$ instead of $H_{\boldsymbol{\pi}}(m)$ for any message $m$. We want to bound $\mathsf{coll}(m^q) := \mathsf{Pr}_{\boldsymbol{\pi} \leftarrow\$ \mathsf{Perm}}[\exists i \neq j,\ H(m_i) = H(m_j)]$. Note that we use the masking function $\tau_L(i) := \alpha^{i-1} \cdot L$ where $L = \boldsymbol{\pi}(0)$. For every $i \neq j$, we have already defined a graph $G_{\tau_L}(m_i, m_j)$ (defined previously as $G_{\tau_L}$ for any block function $\tau$ and a pair of distinct messages $m, m'$). Note that, we explicitly associated the graph $G_{\tau_L}$ with the corresponding message pair $(m_i, m_j)$ as we are dealing with multiple message pairs. We will drop this parametrization whenever the message pair is known from the context. Here, for simplicity, we will denote any vertex by $(k, a)$ instead of $(m_k, a)$. $G_{\tau_L}$ is essentially a disjoint union of cliques. For any clique $C$ in $G_{\tau_L}$ we define

$$\beta_C := \bigoplus_{(k,a) \in C} \alpha^{\ell_k - 1 - a}$$

**Definition 3.5.3.** A masking function $\tau_L$ (or simply $L$) is *cross linear canceling* for some $i \neq j$, if $\beta_C = 0$ for every clique $C$ in $G_L(m_i, m_j)$. We define

$$\theta'(m^q) := \mathsf{Pr}_L[\exists i \neq j,\ \tau_L \text{ is cross linear cancelling for } i, j].$$

AVOIDING ZERO INPUT. We first avoid zero block as an input of $\boldsymbol{\pi}$ since it already appears to define our masking key $L$. We define the following event:

$$\mathsf{bad}_0 : \exists i, a,\ x_i[a] = 0$$

Clearly, $\mathsf{Pr}[\mathsf{bad}_0] \leq \frac{\sigma}{2^n}$ as for every $(i, a)$, $\mathsf{Pr}(x_i[a] = 0) = 1/2^n$.

It is easy to see that if $L$ is cross linear canceling for $i, j$, then $H(m_i) = H(m_j)$. Therefore, a similar statement like Lemma 3.1.3 holds:

**Lemma 3.5.4.**

$$\mathsf{coll}(m^q) \leq \theta'(m^q) + \frac{q^2}{2(2^n - 2\ell)} + \frac{\sigma}{2^n}$$

*Proof.* Let $H(m_i) = H(m_j)$ for some $i < j \in [q]$. Then one of the following three events must happen:

- $\mathsf{bad}_0$
- $A(i,j) : \tau_L$ is cross linear canceling for $i, j \wedge H(m_i) = H(m_j)$
- $B(i,j) : \tau_L$ is not cross linear canceling for $i, j, \wedge H(m_i) = H(m_j) \ \wedge \ \neg\mathsf{bad}_0$

Therefore,

$$\begin{aligned}\mathsf{coll}(m^q) &\leq \mathsf{Pr}[\cup_{i<j} A(i,j)] + \mathsf{Pr}[\cup_{i<j} B(i,j)] + \mathsf{Pr}[\mathsf{bad}_0] \\ &\leq \theta'(m^q) + \mathsf{Pr}[\cup_{i<j} B(i,j)] + \frac{\sigma}{2^n}\end{aligned} \tag{3.12}$$

since $\mathsf{Pr}[\cup_{i<j} A(i,j)] \leq \theta'(m^q)$ and $\mathsf{Pr}[\mathsf{bad}_0] \leq \frac{\sigma}{2^n}$.

Let us now consider the event $B(i,j)$. That $\tau_L$ is not cross linear canceling for $i, j$ implies that there exists a component $C_1$ in the graph $G_{\tau_L}(m_i, m_j)$ such that $\beta_{C_1} \neq 0$. For any component $C$ of the graph, we get a unique value, say $x(C)$ such that $x_k(a) = x(C)$ for any $(k,a) \in C$. Note that for any two distinct components $C$ and $C'$, $x(C) \neq x(C')$. Thus

$$H(m_i) = H(m_j) \iff \beta_{C_1} \cdot \boldsymbol{\pi}(x(C_1)) = \bigoplus_{C \neq C_1} \beta_C \cdot \boldsymbol{\pi}(x(C)) \oplus m[\ell_i] \oplus m[\ell_j].$$

With the assumption $\neg\mathsf{bad}_0$, we can bound the probability of $B(i,j)$ using the randomness of $\boldsymbol{\pi}(x(C_1))$ (since $\beta_{C_1} \neq 0$) after we sample $\boldsymbol{\pi}$-values for all other components in a without replacement manner. Since the maximum number of components in $G_{\tau_L}(m_i, m_j)$ is $\ell_i + \ell_j$, we get

$$\mathsf{Pr}[B(i,j)] \leq \frac{1}{2^n - \ell_i - \ell_j} \leq \frac{1}{2^n - 2\ell} \tag{3.13}$$

Therefore, applying union bound on $\cup_{i<j} B(i,j)$ we get the required bound for $\mathsf{coll}(m^q)$ directly from Eq. (3.12). □

Now, it suffices to bound $\theta'(m^q)$. For the time being we assume that $\ell_i = \ell$ for all $i$. Later we will relax the assumption and complete our proof.

**Lemma 3.5.5.**

$$\theta'(m^q) \leq \frac{\min\{q\ell^2, 2q^2\ell\}}{2^{n+1}}$$

*Proof.* For any $i < j$, we let $a_{i,j} := \min I^{\neq}(m_i, m_j)$. Consider the following two events:

- $\mathsf{bad}_1 : \exists i \in [q], \exists b, c \in [\ell - 1], b < c,$ such that $x_i[b] = x_i[c]$
- $\mathsf{bad}_2 : \exists i < j \in [q], \exists b \in [\ell - 1], b \neq a_{i,j},$ such that $(x_i[a_{i,j}] = x_j[b]) \vee (x_i[a_{i,j}] = x_i[b])$

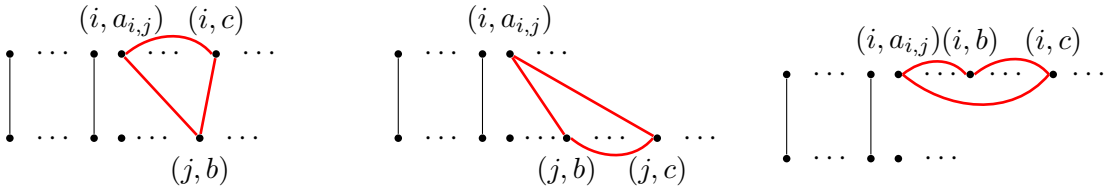


**Figure 3.5.2:** One of these is a necessary subgraph of a cross linear canceling graph for two messages with same block-lengths. A red or (solid) black line between two nodes signifies equality between them. Red is used when two blocks with different positions collide. Black is used when two blocks with same position collide.

Using randomness of $L$, we can easily bound the probability of the above two bad events.

$$\begin{aligned} \mathsf{Pr}[\mathsf{bad}_1] &\leq \frac{q\ell^2}{2^{n+1}} \\ \mathsf{Pr}[\mathsf{bad}_2] &\leq \frac{2q^2\ell}{2^{n+1}} \end{aligned} \tag{3.14}$$

We claim that if $L$ is cross linear canceling for some message pair $(m_i, m_j)$, then both bad events $\mathsf{bad}_1$ and $\mathsf{bad}_2$ hold. We first note that $2^n - 1 > \ell > 1$. Now consider the clique $C$ of $G_{\tau_L}(m_i, m_j)$ that contains $(i, a_{i,j})$. From the definition of $\beta_C$ and the assumption that $\ell_i = \ell_j$, we note that $\beta_C$ can be zero only if $C$ contains at least three vertices. Figure 3.5.2 illustrates all possible types of sub-clique of $C$, containing exactly three vertices, one of which is $(i, a_{i,j})$. It is obvious to see that at least two of the vertices must appear in the same query, whence we establish that $\mathsf{bad}_1$ holds. Further, Figure 3.5.2 shows all possible way in which $x_i[a_{i,j}]$ is connected to some vertex, which establishes that $\mathsf{bad}_2$ must hold. This validates our claim. The proof follows from Eq. (3.14). □

HANDLING DIFFERENT LENGTH QUERIES:

*Claim* 3.5.6. If two messages $m_i$ and $m_j$ are of different length then $\tau$ is not cross linear canceling for $i, j$.

To show this, suppose $\tau$ is cross linear canceling for $i, j$. Without any loss of generality assume $\|m_i\| > \|m_j\|$. Then each $\beta_C$ must be $0$. Thus the sum over all $\beta_C$ where $C$ is a clique in $G_\tau$ must also be $0$. Note that

$$\bigoplus_C \beta_C = \bigoplus_{i=\|m_j\|}^{\|m_i\|-1} \alpha^i$$

which can never be 0 since $\alpha$ is a primitive element of $GF(2^n)$.

Now, we group together all the messages with same block-lengths. Precisely, for any $l \in [\ell]$, we define the following notations:

$$S_l := \{i \in [q] : \|m_i\| = l\}; \quad s_l := |S_l|;$$

Note that $\sum_l s_l = q, \sum_l s_l l = \sigma$.

Moreover, for any $l \in [\ell]$, we define $m^{S_l} := (m_{i_1}, \ldots, m_{i_{s_l}})$ where $\{i_1, \ldots, i_{s_l}\}$ denotes the set $S_l$ in ascending order. Therefore,

$$\theta'(m^{S_l}) = \mathsf{Pr}[\exists i \neq j \in S_l \text{ s.t. } L \text{ is cross linear cancelling for } i, j].$$

Using Claim 3.5.6 and Lemma 3.5.5 we have

$$\theta'(m^q) \leq \sum_l \theta'(m^{S_l}) \leq \mu := \sum_l \mu_l \text{ where } \mu_l := \frac{\min\{s_l l^2, 2s_l^2 l\}}{2^{n+1}}. \tag{3.15}$$

In the remainder, we derive upper bounds on $\mu$ depending upon the range of $\ell$ values. First, consider $\ell \leq 2^{n/4}$. In this case, we have $\mu_l \leq \frac{s_l}{2^{n/2}}$ which implies

$$\mu \leq \frac{q}{2^{n/2}}. \tag{3.16}$$

Now, consider $\ell > 2^{n/4}$. Using the fact that for positive reals $a$ and $b$, $\sqrt{ab} \geq \min\{a, b\}$, we have

$$\begin{aligned} \mu = \sum_l \mu_l &\leq \sum_l \frac{\sqrt{2}(s_l l)^{1.5}}{2^{n+1}} \\ &\leq \frac{\sigma^{1.5}}{2^n}, \end{aligned} \tag{3.17}$$

where the second inequality follows from the fact that $\Sigma_i a_i^r \leq (\Sigma_i a_i)^r$ for positive $a_i$ and $r > 1$, and $\Sigma_l s_l l = \sigma$. Theorem 3.5.1 can be proved by plugging in the suitable values of $\mu$ from the above equations in Lemma 3.5.4, assuming $\ell \leq 2^{n-2}$.

## 3.6 Key Results At a Glance

- Security analysis of NPMAC as described in [70] is shown to be incorrect in section 3.2. Further we state an equivalent problem that must be solved in order to get

a length independent bound for this construction. This problem is still an open problem.

- Theorem 3.3.4 shows a length independent bound (upto $\ell < 2^{n/2}$) for sPMAC whenever the underlying masking function is $\epsilon$-2AXU.
- Theorem 3.5.1 shows that PMAC2 is a novel PMAC variant which attains a length independent security bound for $\ell \leq 2^{n/4}$.

# Chapter 4

# LightMAC and Its Single-key Variants

LightMAC is a suitable candidate for lightweight cryptographic implementation as a MAC. In this chapter, we briefly discuss the proof for a length independent bound for LightMAC, exploiting its hash-then-PRP nature. Then we prove results with similar bounds for single-key LightMAC and its two variants. Here, we employ a novel technique, called *reset-sampling*, under the general H-coefficient proof environment.

## 4.1 Revisiting LightMAC

LightMAC is a block cipher-based parallelizable PRF construction by Luykx et al. [62]. It uses a counter-based encoding of input message blocks, much in the same vein as some of the previously proposed constructions like XMACC and XMACR [6] and protected counter sums [12]. Algorithm 4.1.1 gives the algorithmic description of LightMAC and Figure 4.1.1 gives a pictorial illustration.

Throughout the rest of this paper, we refer to $x[i]$ and $y[i]$ as *intermediate input* and *output*, respectively, for all $i \in [\ell - 1]$ and $y^{\oplus}$ and $t$ are referred as the *final input* and *output*, respectively.

Note that, the block size $n$ and counter size $s$ are application specific parameters that are fixed before any invocation. In order to argue the security of LightMAC, we must have $\langle i \rangle_s \neq \langle j \rangle_s$. When $i = 2^s + j$ for some $j \in [2^s - 1]$, then $\langle i \rangle_s = \langle j \rangle_s$. So, the maximum number of blocks in the padded message, denoted $\ell_{\max}$, must be less than $2^s$. This will ensure that all the counters will be different.

**Algorithm 4.1.1** LightMAC based on an $n$-bit block cipher E instantiated with two keys $K_1, K_2$. Here $s$ denotes the counter size.

**function** $\mathsf{LightMAC}_{\mathsf{E}_{K_1},\mathsf{E}_{K_2}}(m)$
  $y^{\oplus} \leftarrow 0^n$
  $(m[1], \ldots, m[\ell]) \xleftarrow{n-s} m$
  **for** $i = 1$ **to** $\ell - 1$ **do**
    $x[i] \leftarrow \langle i \rangle_s \| m[i]$ ▷ encoding $\langle i \rangle_s$ and $m[i]$ into $x[i]$
    $y[i] \leftarrow E_{K_1}(x[i])$ ▷ encrypting the encoded input
    $y^{\oplus} \leftarrow y^{\oplus} \oplus y[i]$ ▷ accumulating the intermediate output
  **end for**
  $y^{\oplus} \leftarrow y^{\oplus} \oplus \mathsf{pad}_n(m[\ell])$ ▷ accumulating final block of message
  $t \leftarrow E_{K_2}(y^{\oplus})$ ▷ tag generation
  **return** $t$
**end function**

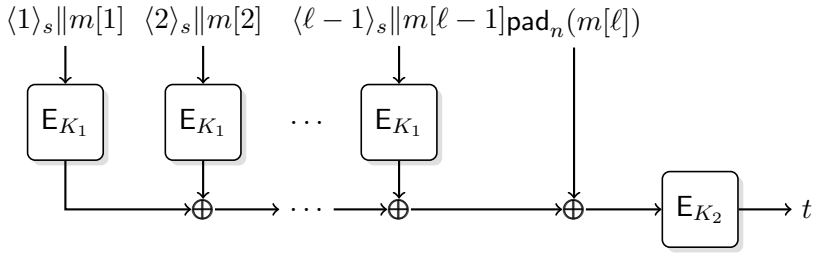


**Figure 4.1.1:** LightMAC evaluated over an $\ell$-block padded message $m$.

### 4.1.1 Hash-then-PRP and the Security of LightMAC

For some $\epsilon \geq 0$, a $(\mathcal{K}, \{0,1\}^{\leq (n-s)2^s}, \mathbb{B})$-keyed function $H$ is called an *$\epsilon$-universal hash function* if for all distinct $m, m' \in \{0,1\}^{\leq (n-s)2^s}$, we have

$$\Pr[\mathsf{K} \xleftarrow{\$} \mathcal{K}] H_{\mathsf{K}}(m) = H_{\mathsf{K}}(m') \leq \epsilon.$$

Universal hash functions are very useful in constructing PRFs via the Hash-then-PRP[1] paradigm [39, 95]. In this paradigm, given independently keyed $\epsilon$-universal hash function $H_{\mathsf{K}}$ and block cipher $\mathsf{E}_{\mathsf{K}'}$, we define the Hash-then-PRP composition as $\mathsf{E}_{K'} \circ H_K$. It is well-known that

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{E}_{\mathsf{K}'} \circ H_{\mathsf{K}}}(q, T) \leq \mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(q, T') + \binom{q}{2}\left(\frac{1}{2^n} + \epsilon\right), \tag{4.1}$$

where $T' = T + qO(T_{\mathsf{E}})$ and $T_{\mathsf{E}}$ denotes the runtime of $\mathsf{E}$.

We skip the proof of this result as it is available in multiple papers including [39, 54]. An informal justification for Eq. (4.1) is based on the observation that if the input to $\mathsf{E}_{\mathsf{K}'}$ is distinct for all $q$ queries then the outputs behave as "almost uniform at random". The probability that some inputs to $\mathsf{E}_{\mathsf{K}'}$ collide is bounded by $\binom{q}{2}\epsilon$.

[1] Here, we say PRP instead of PRF to highlight the use of block cipher based finalization.

PRF SECURITY OF LightMAC: Consider a $(\mathcal{K}, \{0,1\}^{\leq (n-s)2^s}, \mathbb{B})$-keyed function LightHash, defined by the following mapping:

$$\forall m \in \{0,1\}^{\leq (n-s)2^s}, \ \ \mathsf{LightHash}_{\mathsf{E}_{K_1}}(m) := y^{\oplus},$$

where $y^{\oplus}$ is the final input corresponding to $m$ in $\mathsf{LightMAC}_{\mathsf{E}_{K_1},\mathsf{E}_{K_2}}(m)$. Now, we can view LightMAC as an instantiation of Hash-then-PRP, by redefining LightMAC as

$$\mathsf{LightMAC}_{\mathsf{E}_{K_1},\mathsf{E}_{K_2}}(m) := \mathsf{E}_{K_2}(\mathsf{LightHash}_{\mathsf{E}_{K_1}}(m)).$$

Suppose, $\mathsf{LightHash}_{\boldsymbol{\pi}_1}$ is an $\epsilon_{\mathsf{LH}}$-universal hash for $\boldsymbol{\pi}_1 \xleftarrow{\$} \mathsf{Perm}(n)$. Then, using Eq. (4.1), we have

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{LightMAC}}(q,T) \leq 2\mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(\sigma,T') + \binom{q}{2}\left(\frac{1}{2^n} + \epsilon_{\mathsf{LH}}\right), \tag{4.2}$$

where $\sigma$ denotes the total number of blocks in all $q$ padded queries, and $T' = T + \sigma O(T_{\mathsf{E}})$ and $T_{\mathsf{E}}$ denotes the runtime of E.

In [35, 62], it has been shown that $\epsilon_{\mathsf{LH}} \leq 1/(2^n - 2\ell_{\max})$, where $\ell_{\max}$ is the upper bound on the query-length in blocks. It is simply because for any $m \neq m'$ with lengths $\ell, \ell'$ respectively, the event $\mathsf{LightHash}_{\boldsymbol{\pi}_1}(m) = \mathsf{LightHash}_{\boldsymbol{\pi}_1}(m')$ is identical with

$$\bigoplus_{i=1}^{\ell-1} \boldsymbol{\pi}_1(x[i]) \bigoplus_{j=1}^{\ell'-1} \boldsymbol{\pi}_1(x'[j]) = \mathsf{pad}_n(m[\ell]) \oplus \mathsf{pad}_n(m'[\ell']). \tag{4.3}$$

Now, since $m \neq m'$, either $(x[1], \ldots, x[\ell-1]) \neq (x'[1], \ldots, x'[\ell'-1])$, or

$$(x[1], \ldots, x[\ell-1]) = (x'[1], \ldots, x'[\ell'-1]) \wedge \mathsf{pad}_n(m[\ell]) \neq \mathsf{pad}_n(m'[\ell']).$$

The second case has zero probability. In the first case, assuming $\ell \geq \ell'$, we have at least one block say $x[i]$ which is distinct from all other blocks. Then, the probability of the event defined in Eq. (4.3) can be bounded above by probability that $\boldsymbol{\pi}_1(x[i])$ attains a certain value conditioned on the output of $\boldsymbol{\pi}_1$ on all other $x[j]$ and $x'[j']$ values for $j \in [\ell-1] \setminus \{i\}$ and $j' \in [\ell'-1]$. There are at most $2\ell_{\max}$ such values, i.e., $\boldsymbol{\pi}_1$ is already sampled on at most $2\ell_{\max}$ points. Therefore, the probability is bounded above by $1/(2^n - 2\ell_{\max})$.

By combining this bound with Eq. (4.2), we get the desired result for LightMAC in the following proposition.

**Proposition 4.1.1.** *For $\ell_{\max} < \min\{2^{n-2}, 2^s\}$, we have*

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{LightMAC}}(q,T) \leq 2\mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(\sigma,T') + \frac{1.5q^2}{2^n},$$

*where $\sigma$ denotes the total number of blocks in all $q$ padded queries, and $T' = T + \sigma O(T_{\mathsf{E}})$ and $T_{\mathsf{E}}$ denotes the runtime of* $\mathsf{E}$.

### 4.1.2 Bottlenecks for Single-key LightMAC

We have just seen that the query-length independent security argument for LightMAC comes quite easily from the Hash-then-PRP paradigm. This is possible because $\mathsf{K}_1$ and $\mathsf{K}_2$ are independent of each other. A natural direction to explore is the relaxation: $K_1 = K_2 = K$, i.e., LightMAC instantiated with a single key. Formally, we define the single-key LightMAC construction as follows:

$$\mathsf{1k\text{-}LightMAC}_{\mathsf{E}_K} := \mathsf{LightMAC}_{\mathsf{E}_K,\mathsf{E}_K}.$$

We remark that *the additional nomenclature* 1k-LightMAC *is just for the sake of brevity. Indeed,* 1k-LightMAC *and* LightMAC *are algorithmically equivalent*. We have just instantiated $K_1 = K_2 = K$.

First thing to note is that Hash-then-PRP is no longer applicable as the hash function $H_{\mathsf{K}}$ and block cipher $\mathsf{E}_{\mathsf{K}}$ are no longer independent. So, we have to look for a dedicated proof.

Suppose the adversary makes $q$ queries $m_1, \ldots, m_q$ and the corresponding tuple of intermediate inputs and outputs are denoted $x_i = (x_i[1], \ldots, x_i[\ell_i - 1])$ and $y_i = (y_i[1], \ldots, y_i[\ell_i - 1])$, respectively. Similarly, the final input and output for the $q$ queries is denoted $y_i^{\oplus}$ and $t_i$, respectively. Consider the events:

$$\texttt{Icoll}: \ \exists (i,a) \in [q] \times [\ell_i - 1], j \in [q], \text{ such that } x_i[a] = y_j^{\oplus};$$

$$\texttt{Ocoll}: \ \exists (i,a) \in [q] \times [\ell_i - 1], j \in [q], \text{ such that } y_i[a] = t_j;$$

`Icoll` denotes the event that a final input collides with some intermediate input and `Ocoll` denotes the analogous event for output collisions (see Figure 4.1.2).

In a dedicated proof we must take care of these cases as they may lead to inconsistent transcripts. For example, it is possible that $x_i[a] = y_j^{\oplus}$ (`Icoll` holds) but $y_i[a] \neq t_j$ or vice-versa. The probability of realizing such a transcript is zero in the real world. In fact, one can easily create such inconsistencies by first making a query $m_1 = \langle 1 \rangle_s$, and then making another query $m_2 = 10^{n-s-1} \| x$, where $x$ is any arbitrary bit string. Clearly, $x_2[1] = y_1^{\oplus}$, which implies that `Icoll` holds. In section 4.2.1, we show how this helps in mounting an efficient distinguishing attack on 1k-LightMAC using very short messages. Interestingly, if we swap the positions of counter and message block, then this trivial

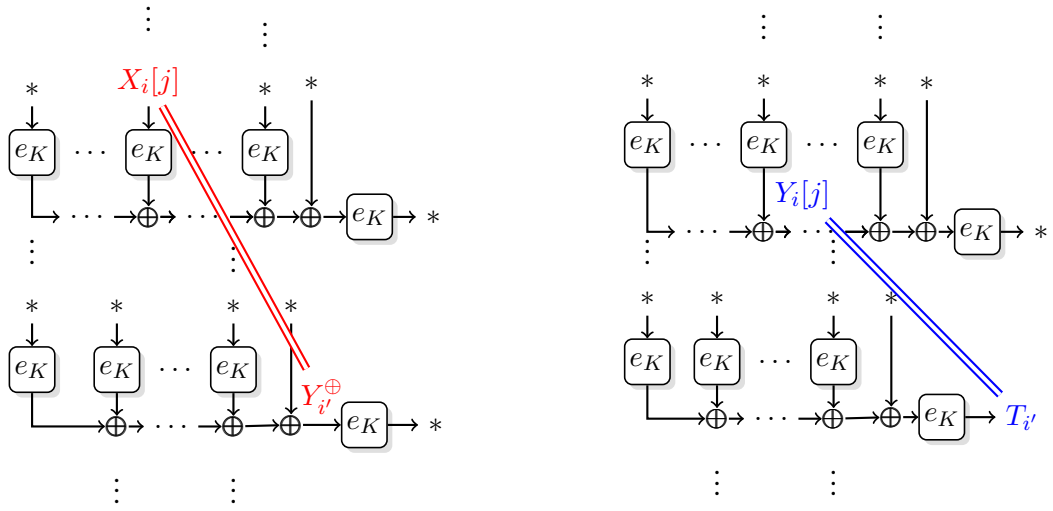


**Figure 4.1.2:** `Icoll` (left) and `Ocoll` (right) events. In each case, labels with same color are equal, and double lines between two labels signify equality between the corresponding variables.

collision is no longer possible. Indeed, in section 4.3 we show that the resulting variant is secure. Since our main goal is to study the standardized algorithm, we first simply assume that messages are at least $(n - s)$ bits long, thereby ensuring that at least one block cipher call is made in the hash layer. But, this only helps to avoid collisions in the corner case. We still have to consider the possibility of `Icoll` and `Ocoll` in the general case. We have to ensure that such inconsistencies do not occur with high probability. A straightforward bound on these events introduces a bound of the form $O(q^2\ell_{\max}/2^n)$ since there are at most $q\ell_{\max}$ many $(i, a)$ pairs and $q$ choices for $j$. However, we aim to do better than this. In the next two sections, we show how we can handle these events in better way.

## 4.2 Security of 1k-LightMAC

This section is devoted to the PRF security of 1k-LightMAC. First, we demonstrate a short messages attack on the construction that justifies the later imposition of a lower bound on the message lengths in order to prove security.

### 4.2.1 A Short Message Attack on 1k-LightMAC

Suppose the counter size $s \leq \frac{n-2}{2}$ and the adversary is allowed to make short length queries. Then, we construct a distinguisher $\mathscr{A}$ against 1k-LightMAC$_\pi$[2] in the following

[2]Note that, the attack is demonstrated for the best possible primitive, i.e., a random permutation. For actual instantiations, we may even get better attacks.

manner:

1. Initialize $\mathcal{L}_1 = \mathcal{L}_2 = \emptyset$.
2. For $(i, b) \in [q] \times \{0, 1\}$:
   (a) Make query $\mathsf{M}_i^{(b)} := \langle i \rangle_s \| b$. Let the corresponding oracle response be $\mathsf{T}_i^{(b)}$.
   (b) Insert the triple $(i, b, \mathsf{T}_i^{(b)})$ in the list $\mathcal{L}_1$ indexed by $(i, b)$.
3. For $j \in [2^q]$:
   (a) Consider $\langle j \rangle_q := j_1 \| \cdots \| j_q$.
   (b) Extract $(i, j_i, \mathsf{T}_i^{(j_i)})$ from $\mathcal{L}_1$ for all $i \in [q]$.
   (c) Compute $\mathsf{S}_j := \mathsf{T}_1^{(j_1)} \oplus \cdots \oplus \mathsf{T}_q^{(j_q)}$.
   (d) Insert the tuple $(j, \mathsf{S}_j)$ in the list $\mathcal{L}_2$ indexed by $j$.
4. In list $\mathcal{L}_2$, find $j$ such that $\lfloor \mathsf{S}_j \rfloor_{s+1} = 0^{s+1}$, where $\lfloor \mathsf{S}_j \rfloor_{s+1}$ denotes the least significant $(s+1)$ bits of $\mathsf{S}_j$.
5. Let

$$\mathsf{M}_j := j_1 \| 10^{n-s-2} \| \cdots \| j_q \| 10^{n-s-2} \| 0^{n-s-1},$$
$$\mathsf{M}_{j'} := \lceil \mathsf{S}_j \rceil_{n-s-1},$$

   where $\lceil \mathsf{S}_j \rceil_{n-s-1}$ denotes the most significant $(n-s-1)$ bits of $\mathsf{S}_j$.
6. Query $\mathsf{M}_j$ and $\mathsf{M}_{j'}$ and let the corresponding oracle responses be $\mathsf{T}_j$ and $\mathsf{T}_{j'}$.
7. If $\mathsf{T}_j = \mathsf{T}_{j'}$ then return 1, else return 0.

Observe that $\mathsf{Y}_j^{\oplus} = \mathsf{S}_j \oplus \left(0^{n-s-1} \| 10^s\right)$ and $\mathsf{Y}_{j'}^{\oplus} = \lceil \mathsf{S}_j \rceil_{n-s-1} \| 10^s$. Let GBA denote the event that $\mathscr{A}$ finds an index $j$ at line 4. Now, it is easy to see that $\mathsf{Y}_j^{\oplus} = \mathsf{Y}_{j'}^{\oplus}$ if and only if GBA occurs. Thus, $\mathscr{A}$ returns 1 with probability 1, when it is interacting with $\mathsf{1k\text{-}LightMAC}_{\boldsymbol{\pi}}$, provided it is able to find the index $j$ in list $\mathcal{L}_2$. On the other hand, $\mathscr{A}$ returns 1 with probability $2^{-n}$, when it is interacting with a uniform random function $\boldsymbol{\rho}$. Formally, we have

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{1k\text{-}LightMAC}_{\boldsymbol{\pi}}}(\mathscr{A}) &= \left| \mathsf{Pr}_{\boldsymbol{\pi}}[\mathscr{A}^{\mathsf{1k\text{-}LightMAC}_{\boldsymbol{\pi}}} = 1] - \mathsf{Pr}_{\boldsymbol{\rho}}[\mathscr{A}^{\boldsymbol{\rho}} = 1] \right| \\
&= \left| \mathsf{Pr}_{\boldsymbol{\pi}}[\mathscr{A}^{\mathsf{1k\text{-}LightMAC}_{\boldsymbol{\pi}}} = 1 \wedge \mathtt{GBA}] - \frac{1}{2^n} \right| \\
&= \left| \mathsf{Pr}_{\boldsymbol{\pi}}[\mathtt{GBA}] - \frac{1}{2^n} \right|
\end{aligned}$$

$$\geq \left| \mathsf{Pr}_{\boldsymbol{\rho}}[\mathtt{GBA}] - 1 + e^{-\frac{q(q-1)}{2^{n+1}}} - \frac{1}{2^n} \right|$$
$$\geq \left| e^{-\frac{2^q}{2^{s+1}}} - e^{-\frac{q(q-1)}{2^{n+1}}} + \frac{1}{2^n} \right|,$$

where the first inequality follows from the statistical distance between a without replacement sample of size $q$ and a with replacement sample of size $q$ (the so called birthday bound), and the second inequality follows from the generalized birthday attack and the fact that $|x| = |-x|$. Clearly, the advantage approaches $0.5$ for $q \approx s+1 = O(n)$. Thus, the single-keyed construction is insecure when the adversary is allowed to make short (less than $(n-s)$ bits) message queries.

Note that, the attack works by first finding out $2q$ intermediate input-output pairs by repeatedly creating `Icoll` event using Step 2(a) and 5. This is possible trivially because the adversary can fix final input via short $(s+1)$-bit queries. Throughout this section, we assume that messages are at least $(n-s)$-bit long. This assumption is used to avoid some trivial bad events, as discussed in section 4.1.2.

**Theorem 4.2.1.** *Let $q, \ell_{\min}, \ell_{\max}, \sigma, t > 0$. For $\ell_{\min} \geq 2$, $q + 4\ell_{\max} \leq 2^{n-1}$, the PRF security of 1k-LightMAC against $\mathbb{A}(q,T)$ is given by*

$$\mathbf{Adv}^{\mathsf{prf}}_{\text{1k-LightMAC}}(q,T) \leq \mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(\sigma+q,T') + \frac{1.5q^2}{2^n} + \frac{7.5q^3\ell^2_{\max}}{2^{2n}} + \frac{4q^4\ell^2_{\max}}{2^{3n}} + \frac{2\sigma}{2^n},$$

*where $q$ denotes the number of queries, $\ell_{\max}$ (res. $\ell_{\min}$) denotes an upper (res. lower) bound on the number of blocks in any padded query, $\sigma$ denotes the total number of blocks present in all $q$ queries, $T' = T + \sigma O(T_{\mathsf{E}})$ and $T_{\mathsf{E}}$ denotes the runtime of $\mathsf{E}$.*

*Further assuming $\ell_{\max} \leq \min\{2^{n/4}, 2^s\}$ and $q \leq \min\{2^{\frac{3n}{4}-2}, 2^{\frac{n}{2}-1.51}\}$, we have*

$$\mathbf{Adv}^{\mathsf{prf}}_{\text{1k-LightMAC}}(q,T) \leq \mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(\sigma+q,T') + \frac{4q^2}{2^n} + \frac{2\sigma}{2^n}.$$

The proof of this theorem is described in the rest of this section. First of all, we switch to the information-theoretic setting, i.e., $\mathsf{E}_{\mathsf{K}}$ is replaced with $\boldsymbol{\pi} \leftarrow\!\$\ \mathsf{Perm}(n)$ via a standard hybrid argument. Formally, we have

$$\mathbf{Adv}^{\mathsf{prf}}_{\text{1k-LightMAC}}(q,T) \leq \mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(\sigma+q,T') + \mathbf{Adv}^{\mathsf{prf}}_{\text{1k-LightMAC}_{\boldsymbol{\pi}}}(q,\infty). \tag{4.4}$$

So it is enough to bound the PRF security of 1k-LightMAC$_{\boldsymbol{\pi}}$, henceforth also referred as the real oracle. We apply the H-coefficient technique to bound this term. Fix any $\mathscr{A} \in \mathbb{A}(q,\infty)$ such that

$$\mathbf{Adv}^{\mathsf{prf}}_{\text{1k-LightMAC}_{\boldsymbol{\pi}}}(q,\infty) = \mathbf{Adv}^{\mathsf{prf}}_{\text{1k-LightMAC}_{\boldsymbol{\pi}}}(\mathscr{A}).$$

Going forward, we will bound the advantage of $\mathscr{A}$.

### 4.2.2 Description of Oracles and their Transcripts

#### 4.2.2.1 Real Oracle:

The real oracle corresponds to $\mathsf{1k\text{-}LightMAC}_{\boldsymbol{\pi}}$. It responds faithfully to all the queries made by $\mathscr{A}$. Once the query-response phase is over, it releases all the intermediate inputs and outputs to $\mathscr{A}$.

In addition, the real oracle releases three binary variables, namely, $\mathsf{FlagT}$, $\mathsf{FlagZ}$, and $\mathsf{FlagY}$, all of which are degenerately set to $0$. The utility of these flags will become apparent from the description of ideal oracle. For now, it is sufficient to note that these flags are degenerate in the real world.

Formally, we have $\Theta_1 := (\widetilde{\mathsf{M}}, \widetilde{\mathsf{T}}, \widetilde{\mathsf{X}}, \widetilde{\mathsf{Y}}, \mathsf{FlagT}, \mathsf{FlagZ}, \mathsf{FlagY})$, where

- $\widetilde{\mathsf{M}} = (\mathsf{M}_1, \ldots, \mathsf{M}_q)$ denotes the $q$-tuple of queries made by $\mathscr{A}$, where $\mathsf{M}_i \in \{0,1\}^{\leq (n-s)2^s}$ for all $i \in [q]$. In addition, for all $i \in [q]$, let $\ell_i := \left\lfloor \frac{|\mathsf{M}_i|}{n-s} \right\rfloor + 1$.
- $\widetilde{\mathsf{T}} = (\mathsf{T}_1, \ldots, \mathsf{T}_q)$ denotes the $q$-tuple of final outputs received by $\mathscr{A}$, where $\mathsf{T}_i \in \mathbb{B}$.
- $\widetilde{\mathsf{X}} = (\mathsf{X}_1, \ldots, \mathsf{X}_q)$, where $\mathsf{X}_i$ denotes the intermediate input tuple for the $i$-th query, i.e., for all $a \in [\ell_i - 1]$, $\mathsf{X}_i[a] = \langle a \rangle_s \| \mathsf{M}_i[a]$.
- $\widetilde{\mathsf{Y}} = (\mathsf{Y}_1, \ldots, \mathsf{Y}_q)$, where $\mathsf{Y}_i$ denotes the intermediate output tuple for the $i$-th query, i.e., for all $a \in [\ell_i - 1]$, $\mathsf{Y}_i[a] = \boldsymbol{\pi}(\mathsf{X}_i[a])$. In addition, let $\widetilde{\mathsf{Y}}^{\oplus} := (\mathsf{Y}_1^{\oplus}, \ldots, \mathsf{Y}_q^{\oplus})$, where $\mathsf{Y}_i^{\oplus} := \bigoplus_{a \in [q]} \mathsf{Y}_i[a] \oplus \mathsf{pad}_n(\mathsf{M}_i[\ell_i])$ for all $i \in [q]$.
- $\mathsf{Flag}\mathsf{I} = 0$ for all $\mathsf{I} \in \{\mathsf{T}, \mathsf{Z}, \mathsf{Y}\}$.

Note that, $\widetilde{\mathsf{X}}$ is completely determined from $\widetilde{\mathsf{M}}$. We have included it in the transcript just for the sake of simplicity. From the definition of $\mathsf{1k\text{-}LightMAC}$, we know that $\boldsymbol{\pi}(\mathsf{Y}_i^{\oplus}) = \mathsf{T}_i$ for all $i \in [q]$. So, *in the real world we always have* $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{Y}}^{\oplus}) \leftrightsquigarrow (\widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$, *i.e.,* $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{Y}}^{\oplus})$ *is permutation compatible with* $(\widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$. We keep this observation in our mind when we simulate the ideal oracle.

#### 4.2.2.2 Ideal oracle:

We reuse the variable notations from the real oracle description to represent the ideal oracle transcript $\Theta_0$, i.e., $\Theta_0 := (\widetilde{\mathsf{M}}, \widetilde{\mathsf{T}}, \widetilde{\mathsf{X}}, \widetilde{\mathsf{Y}}, \mathsf{FlagT}, \mathsf{FlagZ}, \mathsf{FlagY})$. This should not cause

any confusion, as we never consider the random variables $\Theta_1$ and $\Theta_0$ jointly, whence the probability distributions of the constituent variables will always be clear from the context. The ideal oracle transcript is described in three phases, each contingent on some predicates defined over the previous stages. Specifically, the ideal oracle first initializes $\mathsf{FlagT} = 0$, $\mathsf{FlagZ} = 0$, $\mathsf{FlagY} = 0$, and then follows the sampling mechanism given below:

PHASE I (QUERY-RESPONSE PHASE): In the query-response phase, the ideal oracle faithfully simulates $\boldsymbol{\rho} \leftarrow\$ \mathsf{Func}(\{0,1\}^{\leq(n-s)2^s}, \mathbb{B})$. Formally, for $i \in [q]$, at the $i$-th query $\mathsf{M}_i \in \{0,1\}^{\leq(n-s)2^s}$, the ideal oracle outputs $\mathsf{T}_i \leftarrow\$ \mathbb{B}$. The partial transcript generated at the end of the query-response phase is given by $(\widetilde{\mathsf{M}}, \widetilde{\mathsf{T}}, \widetilde{\mathsf{X}})$, where

- $\widetilde{\mathsf{M}} = (\mathsf{M}_1, \ldots, \mathsf{M}_q)$ and $\widetilde{\mathsf{T}} = (\mathsf{T}_1, \ldots, \mathsf{T}_q)$.
- $\widetilde{\mathsf{X}} = (\mathsf{X}_1, \ldots, \mathsf{X}_q)$, where $\mathsf{X}_i = (\mathsf{X}_i[1], \ldots, \mathsf{X}_i[\ell_i - 1])$ and $\mathsf{X}_i[a] := \langle a \rangle_s \| \mathsf{M}_i[a]$ for all $(i, a) \in [q] \times [\ell_i - 1]$.

Now, we define a predicate on $\widetilde{\mathsf{T}}$:

$$\mathtt{BadT}: \ \exists i \neq j \in [q], \text{ such that } \mathsf{T}_i = \mathsf{T}_j.$$

If BadT is true, then FlagT is set to 1, and $\widetilde{\mathsf{Y}} = (\mathsf{Y}_1, \ldots, \mathsf{Y}_q)$ is defined degenerately: $\mathsf{Y}_i[a] = 0^n$ for all $(i, a) \in [q] \times [\ell_i - 1]$. Otherwise, the ideal oracle proceeds to the next phase.

PHASE II (OFFLINE INITIAL SAMPLING PHASE): Onward, we must have $\mathsf{T}_i \neq \mathsf{T}_j$ whenever $i \neq j$, and $\mathsf{FlagT} = 0$, since this phase is only executed when BadT is false. In the offline phase, the ideal oracle initially makes the following sampling:

$$(\mathsf{R}_{x_1}, \ldots, \mathsf{R}_{x_{\sigma'}}) \xleftarrow{\text{wor}} \mathbb{B} \setminus \widetilde{\mathsf{T}},$$

where $(x_1, \ldots, x_{\sigma'})$ is an arbitrary ordering of the set

$$\mathbb{X}(\widetilde{\mathsf{X}}) := \{x : x = \mathsf{X}_i[a], (i, a) \in [q] \times [\ell_i - 1]\}.$$

Next, the ideal oracle sets

- $\mathsf{Z}_i[a] := \mathsf{R}_x$ if $x = \mathsf{X}_i[a]$, for all $(i, a) \in [q] \times [\ell_i - 1]$, and
- $\mathsf{Z}_i^{\oplus} := \bigoplus_{a=1}^{\ell_i - 1} \mathsf{Z}_i[a] \oplus \mathsf{pad}_n(\mathsf{M}_i[\ell_i])$.

At this stage we have $\mathsf{Z}_i[a] = \mathsf{Z}_j[b]$ if and only if $\mathsf{X}_i[a] = \mathsf{X}_j[b]$. In other words, $\widetilde{\mathsf{X}} \leftrightsquigarrow \widetilde{\mathsf{Z}}$. But *the same might not hold for* $\mathsf{Z}^{\oplus}$ *and* $\widetilde{\mathsf{T}}$. Now, we define four predicates on $(\widetilde{\mathsf{Z}}, \widetilde{\mathsf{X}})$:

$\texttt{BadZ1}:\ \exists i \neq j \in [q],$ such that $\mathsf{Z}_i^{\oplus} = \mathsf{Z}_j^{\oplus}$.

$\texttt{BadZ2}:\ \exists (i,a) \in [q] \times [\ell_i - 1],$ such that $\mathsf{X}_i[a] = \mathsf{Z}_i^{\oplus}$.

$\texttt{BadZ3}:\ \exists i \neq j \neq k \in [q], a \neq b \in [\ell_i - 1],$ such that

$$(\mathsf{X}_i[a] = \mathsf{Z}_j^{\oplus}) \wedge (\mathsf{X}_i[b] = \mathsf{Z}_k^{\oplus}).$$

$\texttt{BadZ4}:\ \exists i \neq j \neq k \in [q], a \in [\ell_i - 1], b \in [\ell_j - 1],$ such that

$$(\mathsf{X}_i[a] = \mathsf{Z}_j^{\oplus}) \wedge (\mathsf{X}_j[b] = \mathsf{Z}_k^{\oplus}).$$

We write $\texttt{BadZ} := \texttt{BadZ1} \vee \texttt{BadZ2} \vee \texttt{BadZ3} \vee \texttt{BadZ4}$. Looking ahead momentarily, BadZ will represent bad scenarios that are difficult to fix in the third stage. For example, BadZ1 leads to permutation incompatibility between $\mathsf{Z}^{\oplus}$ and $\widetilde{\mathsf{T}}$ which is not desirable. We will discuss utility of the other three predicates in the description of next phase.

If BadZ is true, then FlagZ is set to 1, and $\widetilde{\mathsf{Y}} = (\mathsf{Y}_1, \ldots, \mathsf{Y}_q)$ is again defined degenerately, as in the case of BadT. Otherwise, the ideal oracle proceeds to the next phase.

PHASE III (OFFLINE RESETTING PHASE): At this point, we know that BadZ is false. In this phase, we will define the complete transcript generated in the ideal world, i.e., $\Theta_0$, by appropriately defining $\widetilde{\mathsf{Y}}$. Remember, our goal is to maintain $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{Y}}^{\oplus}) \leftrightsquigarrow (\widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$.

**Definition 4.2.2** (full collision index)**.** Any query index $i \in [q]$ is called a full collision index if $\exists\ a \in [\ell_i - 1], j \in [q]$ such that $\mathsf{X}_i[a] = \mathsf{Z}_j^{\oplus}$. Additionally, let

- $\mathcal{I} := \{i \in [q] : \mathsf{Z}_j^{\oplus} = \mathsf{X}_i[a],$ for some $a \in [\ell_i - 1], j \in [q]\}$.
- $\mathcal{J} := \{j \in [q] : \mathsf{Z}_j^{\oplus} = \mathsf{X}_i[a]$ for some $(i,a) \in [q] \times [\ell_i - 1]\}$.
- $\mathsf{FCT} := \{(i,a,j) : i,j \in [q], a \in [\ell_i - 1]$ such that $\mathsf{Z}_j^{\oplus} = \mathsf{X}_i[a]\}$. Sometimes, we also use $\widetilde{\mathsf{FCT}} := \{(i,a) \in [q] \times [\ell_i - 1] : \exists j \in [q]$ such that $\mathsf{Z}_j^{\oplus} = \mathsf{X}_i[a]\}$.

We refer to $i \in \mathcal{I}$ and $j \in \mathcal{J}$ as full-collision and resetting index, respectively.

Observe that we can simply set $\widetilde{\mathsf{Y}} = \widetilde{\mathsf{Z}}$, whenever $\mathcal{I} = \emptyset$, since $\neg(\texttt{BadT} \vee \texttt{BadZ})$ holds. However, we need a more involved method when $\mathcal{I} \neq \emptyset$. Next, we use a novel sampling approach, called *reset-sampling*, in context of the sampling for $\widetilde{\mathsf{Y}}$.

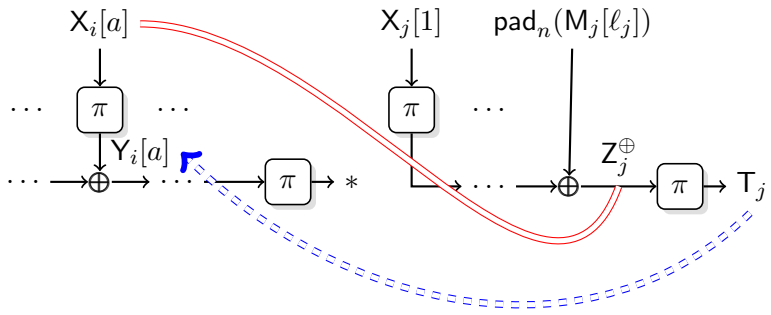


**Figure 4.2.1:** Resetting of $\mathsf{Y}_i[a]$ due to collision $\mathsf{X}_i[a] = \mathsf{Z}_j^{\oplus}$. The red double line represents a collision arising in phase II sampling. The blue dashed edge represents the corresponding resetting in phase III sampling.

*Reset-sampling:* The sampling of $\widetilde{\mathsf{Y}}$ is done in two stages:

STAGE 1: For all $(i, a) \in [q] \times [\ell_i - 1]$, set $\mathsf{Y}_i[a] = \mathsf{Z}_i[a]$.

STAGE 2: For all $(i, a, j) \in \mathsf{FCT}$, *reset* $\mathsf{Y}_i[a] = \mathsf{T}_j$.

Finally, define $\mathsf{Y}^{\oplus} := (\mathsf{Y}_1^{\oplus}, \ldots, \mathsf{Y}_q^{\oplus})$, where $\mathsf{Y}_i^{\oplus} = \bigoplus_{a \in [q]} \mathsf{Y}_i[q] \oplus \mathsf{pad}_n(\mathsf{M}_i[\ell_i])$.

In the second stage, we have reset $\mathsf{Y}_i[a]$ from $\mathsf{Z}_i[a]$ to $\mathsf{T}_j$ for all $(i, a, j) \in \mathsf{FCT}$. This fixes the previous inconsistency issue, i.e., $\mathsf{X}_i[a] = \mathsf{Z}_j^{\oplus}$ and $\mathsf{Y}_i[a] \neq \mathsf{T}_j$. Figure 4.2.1 gives a pictorial view of this step. The following must hold due to the condition $\neg$BadZ:

- For each $(i, a) \in \mathcal{I} \times [\ell_i - 1]$, there is a unique choice for $j$ (if exists) such that $\mathsf{Y}_i[a]$ is reset to $\mathsf{T}_j$. Otherwise, $\neg$BadZ1 is violated.
- Continuing the previous point, we must have $j \neq i$. Otherwise, $\neg$BadZ2 is violated. Indeed, $i = j$ incurs a trivial inconsistency: $(\mathsf{Y}_i[a] = \mathsf{T}_i) \wedge (\mathsf{X}_i[a] \neq \mathsf{Y}_i^{\oplus})$ due to the resetting mechanism.
- For each $i \in \mathcal{I}$, there exists at most one $a \in [\ell_i - 1]$, such that $\mathsf{Y}_i[a]$ is reset. Otherwise, $\neg$BadZ3 is violated.
- For all $j \in \mathcal{J}$, none of the intermediate outputs are reset. Otherwise, $\neg$BadZ4 is violated.

To summarize, the ideal oracle ensures that for each full collision index at most one intermediate output is reset, and the resetting index is uniquely determined. Further, a full collision index cannot be a resetting index. Thus, $\neg$BadZ helps in avoiding trivial inconsistencies as well as keeping the resetting to a minimum. Now, we define two predicates on $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{Z}}, \widetilde{\mathsf{Y}})$:

$\texttt{BadY1}: \exists i \neq j, k \in [q], \exists a \in [\ell_i - 1], b \in [\ell_k - 1],$ such that

$$(\mathsf{X}_i[a] = \mathsf{Z}_j^{\oplus}) \wedge (\mathsf{Y}_i^{\oplus} = \mathsf{X}_k[b]).$$

$\texttt{BadY2}: \exists i \neq j \neq k \in [q], \exists a \in [\ell_i - 1],$ such that $(\mathsf{X}_i[a] = \mathsf{Z}_j^{\oplus}) \wedge (\mathsf{Y}_i^{\oplus} = \mathsf{Y}_k^{\oplus})$.

We write $\texttt{BadY} := \texttt{BadY1} \vee \texttt{BadY2}$. It is easy to see that BadY simply handles the new inconsistencies that may arise due to the reset sampling. For example, BadY1 represents the scenario where resetting leads to collision between intermediate and final inputs. Similarly, BadY2 represents the scenario where resetting leads to collision between two final inputs.

If BadY is true, then FlagY is set to 1, and $\widetilde{\mathsf{Y}}$ is redefined degenerately, as in the case of BadT and BadZ. At this point, the ideal oracle transcript is completely defined.

Intuitively, if the ideal oracle is not sampling $\widetilde{\mathsf{Y}}$ degenerately at any stage, then we must have $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{Y}}^{\oplus}) \leftrightsquigarrow (\widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$. We justify this intuition in the following proposition.

**Proposition 4.2.3.** *For* $\neg(\texttt{BadT} \vee \texttt{BadZ} \vee \texttt{BadY})$, *we must have* $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{Y}}^{\oplus}) \leftrightsquigarrow (\widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$.

*Proof.* We have

- $\widetilde{\mathsf{X}} \leftrightsquigarrow \widetilde{\mathsf{Z}}$, by definition of $\widetilde{\mathsf{Z}}$. Moreover the resetting guarantees $\widetilde{\mathsf{Z}} \leftrightsquigarrow \widetilde{\mathsf{Y}}$. Thus, $\widetilde{\mathsf{X}} \leftrightsquigarrow \widetilde{\mathsf{Y}}$.
- We have $\mathsf{Y}_i[a] = \mathsf{T}_j$ if and only if $\mathsf{X}_i[a] = \mathsf{Z}_j^{\oplus}$. Now, $\neg\texttt{BadZ4}$ implies that $j \notin \mathcal{I}$ thus, $\mathsf{Y}_j^{\oplus} = \mathsf{Z}_j^{\oplus}$. Therefore, $\mathsf{Y}_i[a] = \mathsf{T}_j \Rightarrow \mathsf{X}_i[a] = \mathsf{Y}_j^{\oplus}$. Also, $\mathsf{X}_i[a] = \mathsf{Y}_j^{\oplus}$ implies $j \notin \mathcal{I}$ (due to $\neg\texttt{BadY1}$), thus, $\mathsf{Z}_j^{\oplus} = \mathsf{Y}_j^{\oplus}$. This gives us $\mathsf{X}_i[a] = \mathsf{Y}_j^{\oplus} \Rightarrow \mathsf{Y}_i[a] = \mathsf{T}_j$ from the second stage sampling of Y. Thus, $\mathsf{X}_i[a] = \mathsf{Y}_j^{\oplus} \Leftrightarrow \mathsf{Y}_i[a] = \mathsf{T}_j$.
- $\neg\texttt{BadZ} \wedge \neg\texttt{BadY}$ and definition of Y imply that $\mathsf{Y}_i^{\oplus}$'s are distinct. Also, $\neg\texttt{BadT}$ implies that $\mathsf{T}_i$'s are distinct. Thus $\widetilde{\mathsf{Y}}^{\oplus} \leftrightsquigarrow \widetilde{\mathsf{T}}$.

These observations suffice to conclude that $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{Y}}^{\oplus}) \leftrightsquigarrow (\widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$. □

### 4.2.3 Transcript Analysis

SET OF TRANSCRIPTS: Given the description of transcript random variable corresponding to the ideal oracle, we can define the set of transcripts $\mathcal{T}$ as the set of all tuples $\tau = (\widetilde{m}, \widetilde{t}, \widetilde{x}, \widetilde{y}, \mathrm{flagT}, \mathrm{flagZ}, \mathrm{flagY})$, where

- $\widetilde{m} = (m_1, \ldots, m_q)$, where $m_i \in \left(\{0,1\}^{\leq (n-s)2^s}\right)$ for $i \in [q]$. For $i \in [q]$, let $\ell_i = \left\lfloor \frac{|m_i|}{n-s} \right\rfloor + 1$.
- $\widetilde{t} = (t_1, \ldots, t_q)$, where $t_i \in \{0,1\}^n$ for $i \in [q]$;
- $\widetilde{x} = (x_1, \ldots, x_q)$, where $x_i = (x_i[1], \ldots, x_i[\ell_i - 1])$ for $i \in [q]$, and $x_i[a] = \langle a \rangle_s \| m_i[a]$ for all $a \in [\ell_i - 1]$;
- $\widetilde{y} = (y_1, \ldots, y_q)$, where $y_i = (y_i[1], \ldots, y_i[\ell_i - 1])$ for $i \in [q]$, and $y_i[a] \in \mathbb{B}$ for all $a \in [\ell_i - 1]$.
- $\mathrm{flagT}, \mathrm{flagZ}, \mathrm{flagY} \in \{0,1\}$.

Furthermore, the following must always hold:

1. if $\mathrm{flagI} = 1$ for some $\mathrm{I} \in \{\mathrm{T}, \mathrm{Z}, \mathrm{Y}\}$, then $y_i[a] = 0^n$ for all $(i,a) \in [q] \times [\ell_i - 1]$.
2. if $\mathrm{flagT} = 0$, then $t_i$'s are all distinct.
3. if $\mathrm{flagI} = 0$ for all $\mathrm{I} \in \{\mathrm{T}, \mathrm{Z}, \mathrm{Y}\}$, then $(\widetilde{x}, \widetilde{y}^{\oplus}) \leftrightsquigarrow (\widetilde{y}, \widetilde{t})$.

The first two conditions are obvious from the ideal oracle sampling mechanism. The last condition follows from Proposition 6.2.3 and the observation that in ideal oracle sampling for any $\mathsf{I} \in \{\mathsf{T}, \mathsf{Z}, \mathsf{Y}\}$, $\mathsf{FlagI} = 1$ if and only if $\mathtt{BadI}$ is true. Note that, condition 3 is vacuously true for real oracle transcripts.

BAD TRANSCRIPT: A transcript $\tau \in \mathcal{T}$ is called *bad* if and only if the following predicate is true:

$$(\mathsf{FlagT} = 1) \vee (\mathsf{FlagZ} = 1) \vee (\mathsf{FlagY} = 1).$$

In other words, we term a transcript bad if the ideal oracle sets $\widetilde{\mathsf{Y}}$ degenerately. Let

$$\mathcal{T}_{\mathsf{bad}} := \{\tau \in \mathcal{T} : \tau \text{ is bad.}\}.$$

All other transcript $\tau' = (\widetilde{m}, \widetilde{t}, \widetilde{x}, \widetilde{y}, \mathrm{flagT}, \mathrm{flagZ}, \mathrm{flagY}) \in \mathcal{T} \setminus \mathcal{T}_{\mathsf{bad}}$ are called *good*. From the preceding characterization of the set of transcripts, we conclude that for any good transcript $\tau'$, we must have $(\widetilde{x}, \widetilde{y}^{\oplus}) \leftrightsquigarrow (\widetilde{y}, \widetilde{t})$. Henceforth, we drop $\mathrm{flagT}$, $\mathrm{flagZ}$, $\mathrm{flagY}$ notations for any good transcript with an implicit understanding that $\mathrm{flagT} = \mathrm{flagZ} = \mathrm{flagY} = 0$.

To apply the H-coefficient theorem we have to upper bound the probability $\Pr[\Theta_0 \in \mathcal{T}_{\mathsf{bad}}]$ and lower bound the ratio $\Pr[\Theta_1 = \tau]/\Pr[\Theta_0 = \tau]$ for any $\tau \in \mathcal{T} \setminus \mathcal{T}_{\mathsf{bad}}$.

**Lemma 4.2.4** (bad transcript analysis). *For $4\ell_{\max} + q \leq 2^{n-1}$, we have*

$$\Pr\left[\Theta_0 \in \mathcal{T}_{\mathsf{bad}}\right] \leq \frac{3q^2}{2^{n+1}} + \frac{2.5q^3\ell_{\max}^2}{2^{2n}} + \frac{4q^3\ell_{\max}}{2^{2n}} + \frac{4q^4\ell_{\max}^2}{2^{3n}} + \frac{2\sigma}{2^n}.$$

The proof of this lemma is postponed to section 4.2.4.

GOOD TRANSCRIPT: Now, fix a good transcript $\tau = (\widetilde{m}, \widetilde{t}, \widetilde{x}, \widetilde{y})$. Let $\sigma' := |\widetilde{x}|$. Since, $\tau$ is good, we have $(\widetilde{x}, \widetilde{y}^{\oplus}) \leftrightsquigarrow (\widetilde{y}, \widetilde{t})$. Then, we must have $|\widetilde{y}^{\oplus}| = q$. Further, let $|\widetilde{x} \cap \widetilde{y}^{\oplus}| = r$. Thus, $|\widetilde{x} \cup \widetilde{y}^{\oplus}| = q + \sigma' - r$.

*Real world:* In the real world, the random permutation $\boldsymbol{\pi}$ is sampled on exactly $q+\sigma'-r$ distinct points. Thus, we have

$$\Pr\left[\Theta_1 = \tau\right] = \frac{1}{(2^n)_{q+\sigma'-r}}. \tag{4.5}$$

*Ideal world:* Here, the probability computation is slightly involved due to the two stage sampling employed in the ideal oracle. First of all, we have

$$\Pr\left[\widetilde{\mathsf{T}} = \widetilde{t}\right] = \frac{1}{2^{nq}}, \tag{4.6}$$

since each $\mathsf{T}_i$ is sampled uniformly from the set $\mathbb{B}$ independent of others. Now, observe that all the full collision and resetting indices are fully determined from the transcript $\tau$ itself. In other words, we can enumerate the set $\widetilde{\mathsf{FCT}}$. Now, since the transcript is good, we must have $|\widetilde{\mathsf{FCT}}| = |\widetilde{x} \cap \widetilde{y}^{\oplus}| = r$, and for all indices $(i, a) \notin \widetilde{\mathsf{FCT}}$, we have $\mathsf{Y}_i[a] = \mathsf{Z}_i[a]$. Thus, we have

$$\begin{aligned}\Pr\left[\mathsf{Y}_i[a] = y_a^i \wedge (i,a) \notin \widetilde{\mathsf{FCT}} \mid \widetilde{\mathsf{T}} = \widetilde{t}\right] &= \Pr\left[\mathsf{Z}_i[a] = y_a^i \wedge (i,a) \notin \widetilde{\mathsf{FCT}} \mid \widetilde{\mathsf{T}} = \widetilde{t}\right] \\ &= \frac{1}{(2^n - q)_{\sigma'-r}},\end{aligned} \tag{4.7}$$

where the second equality follows from the fact that truncation[3] of a without replacement sample from a set of size $(2^n - q)$ is still a without replacement sample from the same set. We have

$$\begin{aligned}\Pr\left[\Theta_0 = \omega\right] &= \Pr\left[\widetilde{\mathsf{T}} = \widetilde{t}\right] \times \Pr\left[\widetilde{\mathsf{Y}} = \widetilde{y} \mid \widetilde{\mathsf{T}} = \widetilde{t}\right] \\ &\leq \frac{1}{2^{nq}} \times \Pr\left[\mathsf{Y}_i[a] = y_i[a] \wedge (i,a) \notin \widetilde{\mathsf{FCT}} \mid \widetilde{\mathsf{T}} = \widetilde{t}\right] \\ &= \frac{1}{2^{nq}} \times \frac{1}{(2^n - q)_{\sigma'-r}}.\end{aligned} \tag{4.8}$$

[3] Removing some elements from the tuple.

The above discussion on good transcripts can be summarized in shape of the following lemma.

**Lemma 4.2.5.** *For any $\tau \in \mathcal{T} \setminus \mathcal{T}_{\mathsf{bad}}$, we have*

$$\frac{\Pr\left[\Theta_1 = \tau\right]}{\Pr\left[\Theta_0 = \tau\right]} \geq 1.$$

*Proof.* The proof follows from dividing Eq. (6.9) by Eq. (6.12). □ □

From H-coefficient Theorem 2.2.7 and Lemma 6.2.4 and 6.2.5, we get

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{1k\text{-}LightMAC}_{\boldsymbol{\pi}}}(\mathscr{A}) \leq \frac{3q^2}{2^{n+1}} + \frac{2.5q^3\ell^2_{\max}}{2^{2n}} + \frac{4q^3\ell_{\max}}{2^{2n}} + \frac{4q^4\ell^2_{\max}}{2^{3n}} + \frac{2\sigma}{2^n}. \tag{4.9}$$

Theorem 4.2.1 follows from Eq. (4.4) and (6.13).

### 4.2.4 Proof of Lemma 6.2.4

We have

$$\begin{aligned}
\Pr\left[\Theta_0 \in \mathcal{T}_{\mathsf{bad}}\right] &= \Pr\left[(\mathsf{FlagT} = 1) \vee (\mathsf{FlagZ} = 1) \vee (\mathsf{FlagY} = 1)\right] \\
&= \Pr\left[\mathtt{BadT} \vee \mathtt{BadZ} \vee \mathtt{BadY}\right] \\
&\leq \Pr\left[\mathtt{BadT}\right] \times \Pr\left[\mathtt{BadZ} | \neg \mathtt{BadT}\right] \times \Pr\left[\mathtt{BadY} | \neg(\mathtt{BadT} \vee \mathtt{BadZ})\right]
\end{aligned}$$

We will handle the three terms on the right hand side separately. Before delving further, we introduce few more notations.

FEW MORE NOTATIONS: For simplicity, we denote the last padded block of any message $m_i$ by $m_i[\ell_i]$ instead of $\mathsf{pad}_n(m_i[\ell_i])$. For any $(i, a)$ with $i \in [q], a \in [\ell_i]$, $\mathsf{Z}_i^{\oplus \setminus a}$ (res. $\mathsf{Y}_i^{\oplus \setminus a}$) denotes $\bigoplus_{b \neq a} \mathsf{Z}_i[b] \oplus m_i[\ell_i]$ (res. $\bigoplus_{b \neq a} \mathsf{Y}_i[b] \oplus m_i[\ell_i]$).

While we bound the probability of bad events, we need to deal with system of equations in Z variables. Note that Z can be viewed as $\boldsymbol{\pi}(\mathsf{X})$ for the corresponding X variable. We will often employ Lemma 2.3.1 implicitly (without referring at each application) to bound the probability that these system of equations hold.

1. Bounding $\Pr\left[\mathtt{BadT}\right]$: Since, we have at most $\binom{q}{2}$ choice for $i, j$, and for each such pair, $\mathsf{T}_i = \mathsf{T}_j$ holds with exactly $2^{-n}$ probability. Thus, we have

$$\Pr\left[\mathtt{BadT}\right] \leq \frac{q^2}{2^{n+1}}. \tag{4.10}$$

2. Bounding $\Pr[\texttt{BadZ}|\neg\texttt{BadT}]$: Here, we have four cases.

   (a) $\texttt{BadZ1} : \exists i \neq j \in [q]$, such that $\mathsf{Z}_i^{\oplus} = \mathsf{Z}_j^{\oplus}$. This is similar to BadT above. We have
   $$\Pr[\texttt{BadZ1}|\neg\texttt{BadT}] \leq \frac{q^2}{2 \cdot (2^n - q - 2\ell_{\max})}.$$

   (b) $\texttt{BadZ2} : \exists (i, a) \in [q] \times [\ell_i - 1]$, such that $\mathsf{X}_i[a] = \mathsf{Z}_i^{\oplus}$. It is easy to see that
   $$\Pr[\texttt{BadZ2}|\neg\texttt{BadT}] \leq \sum_{i=1}^{q} \frac{\ell_i - 1}{2^n - q - \ell_{\max}} \leq \frac{\sigma}{2^n - q - \ell_{\max}}.$$

   (c) $\texttt{BadZ3} : \exists i \neq j \neq k \in [q], a, b \in [\ell_i - 1]$, such that $(\mathsf{X}_i[a] = \mathsf{Z}_j^{\oplus}) \wedge (\mathsf{X}_i[b] = \mathsf{Z}_k^{\oplus})$. Here, $j \neq k$ implies that the system of equations has rank 2. Thus, using Lemma 2.3.1, we have
   $$\Pr[\texttt{BadZ3}|\neg\texttt{BadT}] \leq \frac{q^3\ell_{\max}^2}{12(2^n - q - 2\ell_{\max})^2}.$$

   (d) $\texttt{BadZ4} : \exists i \neq j \neq k \in [q], a \in [\ell_i - 1], b \in [\ell_j - 1]$, such that $(\mathsf{X}_i[a] = \mathsf{Z}_j^{\oplus}) \wedge (\mathsf{X}_j[b] = \mathsf{Z}_k^{\oplus})$. Using similar argumentation as above, we have,
   $$\Pr[\texttt{BadZ4}|\neg\texttt{BadT}] \leq \frac{q^3\ell_{\max}^2}{12(2^n - q - 2\ell_{\max})^2}.$$

   Combining all the four cases and assuming $q + 2\ell_{\max} \leq 2^{n-1}$, we have
   $$\Pr[\texttt{BadZ}|\neg\texttt{BadT}] \leq \frac{q^2}{2^n} + \frac{0.34q^3\ell_{\max}^2}{2^{2n}} + \frac{2\sigma}{2^n} \tag{4.11}$$

3. Bounding $\Pr[\texttt{BadY}|\neg(\texttt{BadT} \vee \texttt{BadZ})]$: Here, we have two cases:

   (a) $\texttt{BadY1} : \exists i, j, k \in [q], \exists a \in [\ell_i - 1], b \in [\ell_k - 1]$ such that $(\mathsf{X}_i[a] = \mathsf{Z}_j^{\oplus}) \wedge (\mathsf{Y}_i^{\oplus} = \mathsf{X}_k[b])$. By virtue of resetting mechanism and $\neg\texttt{BadZ}$, we arrive at an equivalent system of Z-equations
   $$\begin{aligned} \mathsf{Z}_j^{\oplus} &= \mathsf{X}_i[a] \\ \mathsf{Z}_i^{\oplus\setminus a} &= \mathsf{X}_k[b] \oplus \mathsf{T}_j \end{aligned}$$
   We claim that the system always has rank 2. This can be argued as follows: Suppose the system has rank less than 2. Then, we must have $\mathsf{Z}_j^{\oplus} \oplus \mathsf{X}_i[a] \oplus \mathsf{Z}_i^{\oplus\setminus a} \oplus \mathsf{X}_k[b] \oplus \mathsf{T}_j = 0^n$. However, $\widetilde{\mathsf{Z}}$ are sampled from $\mathbb{B} \setminus \widetilde{\mathsf{T}}$. Hence, $\mathsf{T}_j$ does not cancel out trivially. So, we must always have rank 2. Now if the rank is 2, then we can always rewrite the system of equations such that we have an

equation in $\mathsf{T}_j$ and another equation involving some $\mathsf{Z}$ variables. Then, the first equation holds with at most $1/2^n$ probability (using the randomness of $\mathsf{T}_j$) and conditioned on this the second equation holds with probability at most $1/(2^n - q - 2\ell_{\max})$. Thus, we have

$$\Pr\left[\mathtt{BadY1}|\neg(\mathtt{BadT} \vee \mathtt{BadZ})\right] \leq \frac{q^3\ell_{\max}^2}{2^n(2^n - q - 2\ell_{\max})}.$$

(b) $\mathtt{BadY2} : \exists i, j, k \in [q], \exists a \in [\ell_i - 1]$, such that $(\mathsf{X}_i[a] = \mathsf{Z}_j^{\oplus}) \wedge (\mathsf{Y}_i^{\oplus} = \mathsf{Y}_k^{\oplus})$. Here we get $\mathsf{X}_i[a] = \mathsf{Z}_j^{\oplus} \wedge \mathsf{Z}_i^{\oplus\setminus a} = \mathsf{Y}_k^{\oplus} \oplus \mathsf{T}_j$ which changes according to the following subcases:

*Case A: when $k \notin \mathcal{I}$:* Then the above system becomes

$$\begin{aligned}\mathsf{Z}_j^{\oplus} &= \mathsf{X}_i[a]\\ \mathsf{Z}_i^{\oplus\setminus a} &= \mathsf{Z}_k^{\oplus} \oplus \mathsf{T}_j\end{aligned}$$

Using similar argumentation as before we can conclude that the system has rank 2. Therefore, we have

$$\Pr\left[\mathtt{BadY2} \wedge \text{Case A}|\neg(\mathtt{BadZ} \vee \mathtt{BadT})\right] \leq \frac{q^3\ell_{\max}}{(2^n - q - 3\ell_{\max})^2}.$$

*Case B: when $k \in \mathcal{I}$:* In this case we have the following system of equations:

$$\begin{aligned}\mathsf{Z}_j^{\oplus} &= \mathsf{X}_i[a]\\ \mathsf{Z}_l^{\oplus} &= \mathsf{X}_k[b]\\ \mathsf{Z}_i^{\oplus\setminus a} \oplus \mathsf{Z}_k^{\oplus\setminus b} &= \mathsf{T}_j \oplus \mathsf{T}_l\end{aligned}$$

We must have $j \neq l$. Otherwise we will have $\mathsf{Z}_i^{\oplus} = \mathsf{Z}_k^{\oplus}$ which again violates $\neg\mathtt{BadZ}$. Thus, $j \neq l$. Now, $j \neq l$ and $\neg\mathtt{BadZ}$ implies that $\mathsf{Z}_j^{\oplus} \neq \mathsf{Z}_l^{\oplus}$. Then, following a similar line of argument as before, we conclude that the system has rank 3. Therefore, we have

$$\Pr\left[\mathtt{BadY2} \wedge \text{Case B}|\neg(\mathtt{BadZ} \vee \mathtt{BadT})\right] \leq \frac{q^4\ell_{\max}^2}{2^n(2^n - q - 4\ell_{\max})^2}.$$

Combining all the cases with the assumption that $q + 4\ell_{\max} \leq 2^{n-1}$, we have

$$\Pr\left[\mathtt{BadY}|\neg(\mathtt{BadT} \vee \mathtt{BadZ})\right] \leq \frac{2q^3\ell_{\max}^2}{2^{2n}} + \frac{4q^3\ell_{\max}}{2^{2n}} + \frac{4q^4\ell_{\max}^2}{2^{3n}}. \tag{4.12}$$

The result follows from summing up Eq. (4.10)-(4.12). □

## 4.3 LightMAC-swp

In section 4.1.2, we hinted that a simple change in message block pre-processing — how the counter and message blocks are concatenated — might help in avoiding the short message attack (see section 4.2.1) on 1k-LightMAC. Here we describe this variant and show that it achieves the same security bound as 1k-LightMAC, albeit without any restrictions on minimum length of the messages.

### 4.3.1 Description of LightMAC-swp

We obtain LightMAC-swp by just a small change in each intermediate input. It is described in Algorithm 4.3.1.

**Algorithm 4.3.1** LightMAC-swp based on an $n$-bit block cipher E instantiated with a key $K$. Here $s$ denotes the counter size.

**function** LightMAC $-$ swp$_{\mathsf{E}_K}(m)$
  $y^{\oplus} \leftarrow 0^n$
  $(m[1], \ldots, m[\ell]) \xleftarrow{n-s} m$
  **for** $i = 1$ **to** $\ell - 1$ **do**
    $x[i] \leftarrow m[i] \| \langle i \rangle_s$ ▷ encoding $\langle i \rangle_s$ and $m[i]$ into $x[i]$
    $y[i] \leftarrow E_K(x[i])$ ▷ encrypting the encoded input
    $y^{\oplus} \leftarrow y^{\oplus} \oplus y[i]$ ▷ accumulating the intermediate output
  **end for**
  $y^{\oplus} \leftarrow y^{\oplus} \oplus \mathsf{pad}_n(m[\ell])$ ▷ accumulating final block of message
  $t \leftarrow E_K(y^{\oplus})$ ▷ tag generation
  **return** $t$
**end function**

### 4.3.2 Security of LightMAC-swp

Note that for messages with more than one block, the same security proof for 1k-LightMAC works for LightMAC-swp too. For two messages one of which is single block, the inconsistency as discussed in section 4.1.2 can not happen here, since both counter and the padding rule apply to the "right" (i.e., least significant bits) of a message block and these two concatenated values are different. Therefore, in case of LightMAC-swp, the security result holds for single block messages also. This justification suffices to give the following result.

**Theorem 4.3.1.** *Let $q, \ell_{\min}, \ell_{\max}, \sigma, t > 0$. For $\ell_{\min} \geq 2$, $q + 4\ell_{\max} \leq 2^{n-1}$, the PRF security of 1k-LightMAC against $\mathbb{A}(q, T)$ is given by*

$$\mathbf{Adv}^{\mathsf{prf}}_{\text{1k-LightMAC}}(q, T) \leq \mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(\sigma + q, T') + \frac{1.5q^2}{2^n} + \frac{7.5q^3\ell^2_{\max}}{2^{2n}} + \frac{4q^4\ell^2_{\max}}{2^{3n}} + \frac{2\sigma}{2^n},$$

*where $q$ denotes the number of queries, $\ell_{\max}$ (res. $\ell_{\min}$) denotes an upper (res. lower) bound on the number of blocks in any padded query, $\sigma$ denotes the total number of blocks present in all $q$ queries, $T' = T + \sigma O(T_{\mathsf{E}})$ and $T_{\mathsf{E}}$ denotes the runtime of* E.

*Further assuming $\ell_{\max} \leq \min\{2^{n/4}, 2^s\}$ and $q \leq \min\{2^{\frac{3n}{4}-2}, 2^{\frac{n}{2}-1.51}\}$, we have*

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{1k\text{-}LightMAC}}(q,T) \leq \mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(\sigma+q,T') + \frac{4q^2}{2^n} + \frac{2\sigma}{2^n}.$$

## 4.4 LightMAC-ds: Another Variant of Single-key LightMAC

In the previous section we showed that single-key LightMAC achieves query-length independent security bounds while $\ell_{\min} \geq 2$ and $\ell_{\max} \leq 2^{n/4}$. Now, we propose a simple variant of LightMAC that achieves query-length independent security unconditionally.

### 4.4.1 Description of LightMAC-ds

For any $x \in \mathbb{B}$ and $k < n$, let $\mathsf{chop}_k(x)$ denote the most significant $n-k$ bits of $x$. The complete algorithmic description of LightMAC-ds is given in Algorithm 4.4.1. It

**Algorithm 4.4.1** LightMAC-ds based on an $n$-bit block cipher E instantiated with a single key $K$. Here the counter size is $s-1$. Highlighted lines point to the algorithmic differences with the LightMAC algorithm.

1: **function** LightMAC-ds$_{\mathsf{E}_K}(m)$
2: $y^{\oplus} \leftarrow 0^n$
3: $(m[1], \ldots, m[\ell]) \xleftarrow{n-s} m$
4: **for** $i = 1$ **to** $\ell - 1$ **do**
5: $x[i] \leftarrow 0 \| \langle i \rangle_{s-1} \| m[i]$ ▷ encoding $\langle i \rangle_{s-1}$ and $m[i]$ into $x[i]$
6: $y[i] \leftarrow E_K(x[i])$ ▷ encrypting the encoded input
7: $y^{\oplus} \leftarrow y^{\oplus} \oplus y[i]$ ▷ accumulating the intermediate output
8: **end for**
9: $y^{\oplus} \leftarrow y^{\oplus} \oplus \mathsf{pad}_n(m[\ell])$
10: $t \leftarrow E_K(1 \| \mathsf{chop}_1(y^{\oplus}))$
11: **return** $t$
12: **end function**

is clear from the description that LightMAC-ds uses the familiar technique of domain separation to generate two "almost independent" instances of E. Specifically, we fix the most significant 1-bit of the block cipher input to

- 0 in the processing of encoded message blocks (see line no. 5 in Algorithm 4.4.1).
- 1 in the tag generation call (see line no. 10 in Algorithm 4.4.1).

Since 1-bit is reserved for domain separation, the effective counter size is reduced to $s-1$ for some global parameter $s < n$. Thus, the maximum message length can be at most $(n-s)2^{s-1}$, which is a slight drop from $(n-s)2^s$ in case of LightMAC, for large value of $n$ and $s$.

### 4.4.2 Security of LightMAC-ds

Surprisingly (or not), the security argument for LightMAC-ds is quite similar to the one for single-key LightMAC. In fact, it is slightly easy to argue the security here, as we have already ensured $\neg$`Icoll` (see section 4.1.2) by the virtue of domain separation. However, we still have to handle `Ocoll` (see section 4.1.2) which would require a slight care while sampling the intermediate outputs in the ideal world. Note that, such complications do not arise in case of LightMAC for the obvious reason of independence between the primitives used to generate the intermediate and final outputs. The PRF security of LightMAC-ds is presented in Theorem 4.4.1.

**Theorem 4.4.1.** *Let $q, \ell_{\max}, T > 0$. For $q + 2\ell_{\max} \leq 2^{n-1}$, the PRF security of $\mathscr{A}$ against $\mathbb{A}(q,T)$ is given by*

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{LightMAC\text{-}ds}}(q,T) \leq \mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(\sigma+q,T') + \frac{2.5q^2}{2^n},$$

*where $\ell$ denotes an upper bound on the number of blocks in any padded query, $T' = T + O(T_{\mathsf{E}})$ and $T_{\mathsf{E}}$ denotes the runtime of* E.

As expected, the proof is quite similar and a bit easier than the proof of theorem 4.2.1. As the first step, we apply the hybrid argument to get

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{LightMAC\text{-}ds}}(q,T) \leq \mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(\sigma+q,T') + \mathbf{Adv}^{\mathsf{prf}}_{\mathsf{LightMAC\text{-}ds}_{\boldsymbol{\pi}}}(q,\infty). \tag{4.13}$$

We are interested in a bound on the PRF security of LightMAC-ds$_{\boldsymbol{\pi}}$, henceforth also referred as the real oracle. Fix any $\mathscr{A} \in \mathbb{A}(q,\infty)$ such that

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{LightMAC\text{-}ds}_{\boldsymbol{\pi}}}(q,\infty) = \mathbf{Adv}^{\mathsf{prf}}_{\mathsf{LightMAC\text{-}ds}_{\boldsymbol{\pi}}}(\mathscr{A}).$$

Going forward, we will bound the advantage of $\mathscr{A}$ using H-coefficient technique.

### 4.4.3 Description of Oracles and their Transcripts

#### 4.4.3.1 Real Oracle:

The real oracle is defined analogously as in the proof of Theorem 4.4.1. We describe it just for the sake of completeness. The real oracle faithfully responds to all the queries made by $\mathscr{A}$. Once the query-response phase is over, it releases all the intermediate inputs and outputs to $\mathscr{A}$. Additionally, the real oracle releases two binary flags, $\mathsf{FlagT}$ and $\mathsf{FlagZ}$, that are degenerately set to 0. Formally, we have

$$\Theta_1 := (\widetilde{\mathsf{M}}, \widetilde{\mathsf{T}}, \widetilde{\mathsf{X}}, \widetilde{\mathsf{Y}}, \mathsf{FlagT}, \mathsf{FlagZ}),$$

where

- $\widetilde{\mathsf{M}} = (\mathsf{M}_1, \ldots, \mathsf{M}_q)$ denotes the $q$-tuple of queries made by $\mathscr{A}$, where $\mathsf{M}_i \in \{0,1\}^{\leq (n-s)2^{s-1}}$ for all $i \in [q]$. In addition, for all $i \in [q]$, let $\ell_i := \left\lfloor \frac{|\mathsf{M}_i|}{n-s} \right\rfloor + 1$.
- $\widetilde{\mathsf{T}} = (\mathsf{T}_1, \ldots, \mathsf{T}_q)$ denotes the $q$-tuple of final outputs received by $\mathscr{A}$, where $\mathsf{T}_i \in \mathbb{B}$.
- $\widetilde{\mathsf{X}} = (\mathsf{X}_1, \ldots, \mathsf{X}_q)$, where $\mathsf{X}_i$ denotes the intermediate input tuple for the $i$-th query, i.e., for all $a \in [\ell_i - 1]$, $\mathsf{X}_i[a] = 0 \| \langle a \rangle_{s-1} \| \mathsf{M}_i[a]$.
- $\widetilde{\mathsf{Y}} = (\mathsf{Y}_1, \ldots, \mathsf{Y}_q)$, where $\mathsf{Y}_i$ denotes the intermediate output tuple for the $i$-th query, i.e., for all $a \in [\ell_i - 1]$, $\mathsf{Y}_i[a] = \boldsymbol{\pi}(\mathsf{X}_i[a])$. In addition, let $\widetilde{\mathsf{Y}}^{\oplus} := (\mathsf{Y}_1^{\oplus}, \ldots, \mathsf{Y}_q^{\oplus})$, where $\mathsf{Y}_i^{\oplus} := \bigoplus_{a \in [\ell_i - 1]} \mathsf{Y}_i[a] \oplus \mathsf{pad}_n(\mathsf{M}_i[\ell_i])$ for all $i \in [q]$.
- $\mathsf{FlagT} = \mathsf{FlagZ} = 0$.

Let $\mathsf{chop}_1(\widetilde{\mathsf{Y}}^{\oplus}) = (1 \| \mathsf{chop}_1(\mathsf{Y}_i[1]), \ldots, 1 \| \mathsf{chop}_1(\mathsf{Y}_i[\ell_i - 1]))$. It is straightforward to see that in the real world we always have $(\widetilde{\mathsf{X}}, \mathsf{chop}_1(\widetilde{\mathsf{Y}}^{\oplus})) \leftrightsquigarrow (\widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$, i.e., $(\widetilde{\mathsf{X}}, \mathsf{chop}_1(\widetilde{\mathsf{Y}}^{\oplus}))$ is permutation compatible with $(\widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$.

#### 4.4.3.2 Ideal oracle:

We reuse the notations from real oracle description to represent the variables in the ideal oracle transcript $\Theta_0$, i.e.

$$\Theta_0 := (\widetilde{\mathsf{M}}, \widetilde{\mathsf{T}}, \widetilde{\mathsf{X}}, \widetilde{\mathsf{Y}}, \mathsf{FlagT}, \mathsf{FlagZ}).$$

The ideal oracle transcript is described in two phases, with the second one contingent on some predicate defined over the first stage. Specifically, the ideal oracle initializes $\mathsf{FlagT} = \mathsf{FlagZ} = 0$, and then follows the sampling mechanism given below:

PHASE I (QUERY-RESPONSE PHASE): In the query-response phase, the ideal oracle faithfully simulates $\boldsymbol{\rho} \leftarrow\$ \mathsf{Func}(\{0,1\}^{\leq(n-s)2^{s-1}}, \mathbb{B})$. Formally, for $i \in [q]$, at the $i$-th query $\mathsf{M}_i \in \{0,1\}^{\leq(n-s)2^{s-1}}$, the ideal oracle outputs $\mathsf{T}_i \leftarrow\$ \mathbb{B}$. The partial transcript generated at the end of the query-response phase is given by $(\widetilde{\mathsf{M}}, \widetilde{\mathsf{T}}, \widetilde{\mathsf{X}})$, where

- $\widetilde{\mathsf{M}} = (\mathsf{M}_1, \ldots, \mathsf{M}_q)$ and $\widetilde{\mathsf{T}} = (\mathsf{T}_1, \ldots, \mathsf{T}_q)$.
- $\widetilde{\mathsf{X}} = (\mathsf{X}_1, \ldots, \mathsf{X}_q)$, where $\mathsf{X}_i = (\mathsf{X}_i[1], \ldots, \mathsf{X}_i[\ell_i - 1])$ and $\mathsf{X}_i[a] := 0 \| \langle a \rangle_{s-1} \| \mathsf{M}_i[a]$ for all $(i, a) \in [q] \times [\ell_i - 1]$.

Now, we define a predicate on $\widetilde{\mathsf{T}}$:

$$\mathtt{BadT}: \ \exists i \neq j \in [q], \text{ such that } \mathsf{T}_i = \mathsf{T}_j.$$

If BadT is true, then $\mathsf{FlagT} = 1$, and $\widetilde{\mathsf{Y}} = (\mathsf{Y}_1, \ldots, \mathsf{Y}_q)$ is defined degenerately: $\mathsf{Y}_i[a] = 0^n$ for all $(i, a) \in [q] \times [\ell_i - 1]$. Otherwise, the ideal oracle proceeds to the next phase.

PHASE II (OFFLINE SAMPLING PHASE): In the offline phase, the ideal oracle initially makes the following sampling:

$$(\mathsf{R}_{x_1}, \ldots, \mathsf{R}_{x_t}) \xleftarrow{\text{wor}} \mathbb{B} \setminus \widetilde{\mathsf{T}},$$

where $(x_1, \ldots, x_t)$ is an arbitrary ordering of the set

$$\mathbb{X}(\widetilde{\mathsf{X}}) := \{x : x = \mathsf{X}_i[a], (i, a) \in [q] \times [\ell_i - 1]\}.$$

Next, the ideal oracle sets

- $\mathsf{Z}_i[a] := \mathsf{R}_x$ if $x = \mathsf{X}_i[a]$, for all $(i, a) \in [q] \times [\ell_i - 1]$, and
- $\mathsf{Z}_i^{\oplus} := \bigoplus_{a=1}^{\ell_i - 1} \mathsf{Z}_i[a] \oplus \mathsf{pad}_n(\mathsf{M}_i[\ell_i])$.

At this stage we have $\mathsf{Z}_i[a] = \mathsf{Z}_j[b]$ if and only if $\mathsf{X}_i[a] = \mathsf{X}_j[b]$. In other words, $\widetilde{\mathsf{X}} \leftrightsquigarrow \widetilde{\mathsf{Z}}$. But *the same might not hold for* $\mathsf{chop}_1(\widetilde{\mathsf{Z}}^{\oplus})$ *and* $\widetilde{\mathsf{T}}$. Now, we define a predicate on $(\widetilde{\mathsf{Z}}, \widetilde{\mathsf{X}})$:

$$\mathtt{BadZ}: \ \exists i \neq j \in [q], \text{ such that } \mathsf{chop}_1(\mathsf{Z}_i^{\oplus}) = \mathsf{chop}_1(\mathsf{Z}_j^{\oplus}).$$

Note that, $\neg\texttt{BadZ}$ ensures $\mathsf{chop}_1(\widetilde{\mathsf{Z}}^{\oplus}) \leftrightsquigarrow \widetilde{\mathsf{T}}$, that when coupled with the $\widetilde{\mathsf{X}} \leftrightsquigarrow \widetilde{\mathsf{Z}}$ due to the sampling mechanism ensures $(\widetilde{\mathsf{X}}, \mathsf{chop}_1(\widetilde{\mathsf{Z}}^{\oplus})) \leftrightsquigarrow (\widetilde{\mathsf{Z}}, \widetilde{\mathsf{T}})$. Intuitively, this makes the ideal world almost similar to the real world.

If BadZ is true, then $\mathsf{FlagZ} = 1$, and $\widetilde{\mathsf{Y}} := (\mathsf{Y}_1, \ldots, \mathsf{Y}_q)$ is again defined degenerately, as in the case of BadT. Otherwise, $\widetilde{\mathsf{Y}} := \widetilde{\mathsf{Z}}$. At this point, the transcript random variable for the ideal world is completely determined.

### 4.4.4 Transcript Analysis

SET OF TRANSCRIPTS: Given the description of the transcript random variable corresponding to the ideal oracle, we can define the set of transcripts $\mathcal{T}$ as the set of all tuples $\tau = (\widetilde{m}, \widetilde{t}, \widetilde{x}, \widetilde{y}, \mathrm{flagT}, \mathrm{flagZ})$, where

- $\widetilde{m} = (m_1, \ldots, m_q)$, where $m_i \in \left(\{0,1\}^{\leq (n-s)2^{s-1}}\right)$ for $i \in [q]$. Let $\ell_i = \left\lfloor \frac{|m_i|}{n-s} \right\rfloor + 1$ for $i \in [q]$.
- $\widetilde{t} = (t_1, \ldots, t_q)$, where $t_i \in \{0,1\}^n$ for $i \in [q]$;
- $\widetilde{x} = (x_1, \ldots, x_q)$, where $x_i = (x_i[1], \ldots, x_i[\ell_i - 1])$ for $i \in [q]$, and $x_i[a] = 0 \| \langle a \rangle_{s-1} \| m_i[a]$ for all $a \in [\ell_i - 1]$;
- $\widetilde{y} = (y_1, \ldots, y_q)$, where $y_i = (y_i[1], \ldots, y_i[\ell_i - 1])$ for $i \in [q]$, and $y_i[a] \in \mathbb{B}$ for all $a \in [\ell_i - 1]$.
- $\mathrm{flagT}, \mathrm{flagZ} \in \{0,1\}$.

Furthermore, the following must always hold:

1. if $\mathrm{flagI} = 1$ for some $\mathrm{I} \in \{\mathrm{T}, \mathrm{Z}\}$, then $y_i[a] = 0^n$ for all $(i,a) \in [q] \times [\ell_i - 1]$.
2. if $\mathrm{flagT} = 0$, then $t_i$'s are all distinct.
3. if $\mathrm{flagI} = 0$ for all $\mathrm{I} \in \{\mathrm{T}, \mathrm{Z}\}$, then $(\widetilde{x}, \mathsf{chop}_1(\widetilde{\mathsf{Y}}^{\oplus})) \leftrightsquigarrow (\widetilde{y}, \widetilde{t})$.

BAD TRANSCRIPT: A transcript $\tau \in \mathcal{T}$ is called *bad* if and only if the following predicate is true:

$$(\mathsf{FlagT} = 1) \vee (\mathsf{FlagZ} = 1).$$

In other words, we term a transcript bad if the ideal oracle sets $\widetilde{\mathsf{Y}}$ degenerately. Let

$$\mathcal{T}_{\mathsf{bad}} := \{\tau \in \mathcal{T} : \tau \text{ is bad.}\}.$$

All other transcript $\tau' = (\widetilde{m}, \widetilde{t}, \widetilde{x}, \widetilde{y}, \mathrm{flagT}, \mathrm{flagZ}) \in \mathcal{T} \setminus \mathcal{T}_{\mathsf{bad}}$ are called *good*. It is pretty straightforward to deduce that for any good transcript we must have $(\widetilde{x}, \mathsf{chop}_1(\widetilde{y}^{\oplus})) \leftrightsquigarrow (\widetilde{y}, \widetilde{t})$.

**Lemma 4.4.2** (bad transcript analysis). *For $q + 2\ell_{\max} \leq 2^{n-1}$, we have*

$$\Pr\left[\Theta_0 \in \mathcal{T}_{\mathsf{bad}}\right] \leq \frac{2.5q^2}{2^n}.$$

*Proof.* We have

$$\begin{aligned}\Pr\left[\Theta_0 \in \mathcal{T}_{\mathsf{bad}}\right] &= \Pr\left[(\mathsf{FlagT} = 1) \vee (\mathsf{FlagZ} = 1)\right] \\ &= \Pr\left[\mathtt{BadT} \vee \mathtt{BadZ}\right] \\ &\leq \Pr\left[\mathtt{BadT}\right] \times \Pr\left[\mathtt{BadZ}|\mathtt{BadT}\right].\end{aligned}$$

We will handle the two terms on the right hand side separately:

1. Bounding $\Pr\left[\mathtt{BadT}\right]$: Since, we have at most $\binom{q}{2}$ choice for $i, j$, and for each such pair, $\mathsf{T}_i = \mathsf{T}_j$ holds with exactly $2^{-n}$ probability. Thus, we have

$$\Pr\left[\mathtt{BadT}\right] \leq \frac{q^2}{2^{n+1}}. \tag{4.14}$$

2. Bounding $\Pr\left[\mathtt{BadZ}|\neg\mathtt{BadT}\right]$: Fix two indices $i \neq j$. Now, we can have two cases:

   (a) $\ell_i = \ell_j$: Since $\mathsf{M}_i \neq \mathsf{M}_j$, we must have at least one index $a$, such that $\mathsf{M}_i[a] \neq \mathsf{M}_j[a]$, which implies that $\mathsf{X}_i[a] \neq \mathsf{X}_j[a]$. Further, note that $\mathsf{X}_i[a] \neq \mathsf{X}_k[b]$ for all $(k, b) \in \{i, j\} \times [\ell_k - 1]$. Then, by conditioning on the value of $\mathsf{Z}_k[b]$ for all $(k, b) \in \{i, j\} \times [\ell_k - 1] \setminus \{(i, a)\}$, we bound the probability that $\mathsf{chop}_1(\mathsf{Z}_i^{\oplus}) = \mathsf{chop}_1(\mathsf{Z}_i^{\oplus})$ to at most $2/(2^n - q - 2\ell_{\max})$, where the factor of 2 in the numerator is due to 1-bit chopping. There are at most $\binom{q}{2}$ choices for $i, j$, so in this case the probability is at most $q^2/(2^n - q - 2\ell_{\max})$.

   (b) $\ell_i \neq \ell_j$: Without loss of generality we assume that $\ell_i > \ell_j$. Then, applying exactly the same argumentation as used in the preceding case with $(i, a) = (i, \ell_i - 1)$, we can bound the probability in this case to at most $q^2/(2^n - q - 2\ell_{\max})$.

   Since the two cases are mutually exclusive, we have

$$\Pr\left[\mathtt{BadZ}|\neg\mathtt{BadT}\right] \leq \frac{q^2}{(2^n - q - 2\ell_{\max})}. \tag{4.15}$$

The result follows by summing up Eq. (4.14) and (4.15) and using $q+2\ell_{\max} \leq 2^{n-1}$. □

□

GOOD TRANSCRIPT: Fix a good transcript $\tau = (\widetilde{m}, \widetilde{t}, \widetilde{x}, \widetilde{y}, 0, 0)$. Let $\sigma' := |\widetilde{x}|$. Since, $\tau$ is good, we have $(\widetilde{x}, \mathsf{chop}_1(\widetilde{y}^{\oplus})) \rightsquigarrow (\widetilde{y}, \widetilde{t})$. Then, we must have $|\mathsf{chop}_1(\widetilde{y}^{\oplus})| = q$. Further, $\widetilde{x} \cap \mathsf{chop}_1(\widetilde{y}^{\oplus}) = \emptyset$ due to domain separation. Thus, $|\widetilde{x} \cup \mathsf{chop}_1(\widetilde{y}^{\oplus})| = q + \sigma'$.

*Real world:* In the real world, the random permutation $\boldsymbol{\pi}$ is sampled on exactly $q + \sigma'$ distinct points. Thus, we have

$$\Pr\left[\Theta_1 = \tau\right] = \frac{1}{(2^n)_{q+\sigma'}}. \tag{4.16}$$

*Ideal world:* In the ideal world, first $\widetilde{\mathsf{T}}$ is sampled in with replacement fashion from a set of size $2^n$. Then, exactly $\sigma'$ values are sampled corresponding to $\widetilde{\mathsf{Y}}$ in without replacement fashion from a set of size $2^n - q$. Thus, we have

$$\Pr\left[\Theta_0 = \tau\right] = \frac{1}{2^{nq}} \times \frac{1}{(2^n - q)_{\sigma'}}. \tag{4.17}$$

On dividing Eq. (4.16) by (4.17), we get

$$\frac{\Pr\left[\Theta_1 = \tau\right]}{\Pr\left[\Theta_0 = \tau\right]} \geq 1.$$

From H-coefficient Theorem 2.2.7 and Lemma 4.4.2, we get

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{LightMAC\text{-}ds}_{\boldsymbol{\pi}}}(\mathscr{A}) \leq \frac{2.5q^2}{2^n}. \tag{4.18}$$

Theorem 4.4.1 follows from Eq. (4.13) and (4.18).

## 4.5 Key Results At a Glance

- An $O(n)$-query forgery attack on 1k-LightMAC is shown in section 4.2.1 when the adversary is allowed to make short queries of length less than $n - s$.
- Theorem 4.2.1 shows that 1k-LightMAC is as secure as two-key LightMAC whenever the query-lengths are in the range $[n - s, (n - s)\min\{2^{n/4}, 2^s\}]$.

- Theorem 4.3.1 shows that LightMAC-swp – a novel single-key variant of LightMAC is as secure as two-key LightMAC whenever query-lengths are upper bounded by $(n-s)\min\{2^{n/4}, 2^s\}$.
- Theorem 4.4.1 shows that LightMAC-ds – another single-key variant of LightMAC achieves birthday bound security while the query-lengths are upper bounded by $(n-s)2^{s-1}$.

# Chapter 5

# PMAC Family: Towards a Generalization

Continuing the discussion in the previous chapters, we will try to have a generalized view of the PMAC family in this chapter. In this direction, we will give essential definitions required for this discussion. Moreover, some interesting general results will also be proven. Let us start with the general setup and will delve deeper step by step.

Here, for the sake of simplicity, we will use a slightly different version of the original PMAC-type constructions: instead of simply xoring the last block itself (with other intermediate outputs), we will xor its encrypted output.

## 5.1 Family of Parallelizable MACs

### 5.1.1 Hash Function Family

Suppose $\mathcal{H}$ is a family of hash functions mapping the set of all arbitrary binary strings $\{0,1\}^*$ to $\mathfrak{B}$. For any $m \neq m' \in \{0,1\}^*$, we define $\mathsf{coll}_{\mathcal{H}}(m, m')$ by the collision probability $\mathsf{Pr}[\mathsf{H}(m) = \mathsf{H}(m')]$ where $\mathsf{H} \leftarrow\!\$\ \mathcal{H}$. The maximum collision probability for a pair of distinct inputs having at most $\ell$ $n$-bit blocks is denoted as $\mathsf{coll}_{\mathcal{H}}(\ell)$ or $\mathsf{coll}_{\mathsf{H}}(\ell)$. This can be further generalized to $q$ distinct inputs $m_1, \ldots, m_q$. We write $\mathsf{coll}_{\mathcal{H}}(m_1, \ldots, m_q)$ to denote the probability $\mathsf{Pr}[\exists i \neq j, \mathsf{H}(m_i) = \mathsf{H}(m_j)]$ under the randomness of $\mathsf{H} \leftarrow\!\$\ \mathcal{H}$. The maximum collision probability for $q$ messages, denoted as $\mathsf{coll}_{\mathcal{H}}(q, \ell)$ or $\mathsf{coll}_{\mathsf{H}}(q, \ell)$ is then defined as $\max \mathsf{coll}_{\mathcal{H}}(m_1, \ldots, m_q)$ where the maximum is taken over $q$ distinct inputs $m_1, \ldots, m_q$ such that $\|m_i\|_n \leq \ell$. So, by definition $\mathsf{coll}_{\mathcal{H}}(2, \ell)$ is same as $\mathsf{coll}_{\mathcal{H}}(\ell)$.

**Definition 5.1.1** (universal hash function [21])**.** A hash function family $\mathcal{H}$ is called $\epsilon(\ell)$ *universal function* if $\mathsf{coll}_{\mathcal{H}}(\ell) \leq \epsilon(\ell)$, for all $\ell$.

For any hash function family $\mathcal{H}$, using the union bound we have

$$\mathsf{coll}_{\mathcal{H}}(q, \ell) \leq \binom{q}{2} \cdot \mathsf{coll}_{\mathcal{H}}(\ell).$$

For PHash (the underlying hash of PMAC), authors of [39] had shown that $\mathsf{coll}_{\mathsf{PHash}}(q, \ell)$ is about $\ell q^2/2^n$ and $\mathsf{coll}_{\mathsf{PHash}}(\ell)$ is about $\ell/2^n$. So the above inequality is tight.

*Remark* 5.1.2. It is worth to remark that $\mathsf{coll}_{\mathcal{H}}(q, \ell) = \Theta(\binom{q}{2} \cdot \mathsf{coll}_{\mathcal{H}}(\ell))$ is not true in general. In [17], authors had shown that the collision probability of the CBC-MAC is at least $\log \ell/2^n$. However, authors of [52, 84] had shown that the collision probability of CBC-MAC for $q$ messages is at most $q^2/2^n$ for all $\ell \leq 2^{n/4}$. This may happen due to a certain restriction imposed on the pair of messages for which the maximum collision probability is achieved. The same restriction may not be achieved for all pairs of $q$ messages. For example, in case of CBC-hash, the restriction is that one input should be a prefix of the other and the larger one contains $O(\ell)$ many zero blocks after the common prefix. This condition clearly cannot be achieved for all pairs of any $q$ messages for large $q$.

### 5.1.2 Hash-then-PRP

We first recall the hash-then-PRP paradigm and its known security analysis. Let $e$ be an $n$-bit blockcipher with a plaintext space $\mathfrak{B}$ and a key-space $\mathcal{K}$. Let $\mathcal{H}$ be a hash function family mapping arbitrary binary strings to $n$-bit block. On an input $m \in \{0,1\}^*$, Hash-then-PRP construction outputs $e_K(\mathsf{H}(m))$ where $K \xleftarrow{\$} \mathcal{K}$ and $\mathsf{H} \xleftarrow{\$} \mathcal{H}$ are sampled independently. The PRF advantage of this construction (based on random permutation $\boldsymbol{\pi}$ replacing the underlying block cipher $e_K$) can be shown to be at most

$$\mathsf{coll}_{\mathcal{H}}(q, \ell) \; + \; q^2/2^{n+1}.$$

By using PRP-PRF switching lemma [4], we can replace the random permutation by the random function $\boldsymbol{\rho}$ at the cost of $q^2/2^{n+1}$ advantage. The basic idea of the remaining part of the proof is that as long as there is no collision among the hash outputs, the $\boldsymbol{\rho}$ returns random values and hence the composition function behaves like a random function defined over the arbitrary input space. We now justify that this bound is tight for most of the cases. Note that $q^2/2^{n+1}$ is the probability for collision of a random function.

**(Case A) $\mathsf{coll}_{\mathcal{H}}(q,\ell) > cq^2/2^{n+1}$ for some constant $c > 1$:**
If an adversary queries those $q$ messages for which the collision probability of the underlying hash function is maximized then it gets collisions with probability $\mathsf{coll}_{\mathsf{H}}(q,\ell)$. Hence, the distinguishing advantage based on collision is at least $(c-1)q^2/2^{n+1}$. The maximum distinguishing advantage is at most $(c+1)q^2/2^{n+1}$.

**(Case B) $\mathsf{coll}_{\mathcal{H}}(q,\ell) \leq cq^2/2^{n+1}$ for some constant $c < 1$:**
Suppose there are $q$ inputs for which no collision on the hash functions occurs. This is true for LightMAC, PMAC and many other parallelizable MACs. For all these constructions, we can choose inputs which only differ in one block. Then, due to permutation property, no collision among final outputs is occurred. The distinguishing advantage for those adversaries based on non-collision is $q^2/2^{n+1}$. Hence, once again in those cases we obtain the tight bound.

#### 5.1.2.1 Hash-then-PRP with Single-Key:

The proof for independent hash-key based construction is not applicable for dependent key settings. We cannot bound the PRF advantage of the composition $\mathsf{H}^{e_K} \circ e_K$ where the hash function H is itself based on the block cipher $e_K$. A domain separation mechanism can avoid the collisions between a final input and intermediate inputs of the block cipher. But, this cannot avoid the non-collision among the $q$ final outputs and about $\sigma$ many intermediate outputs. If we carry similar analysis as hash-then-PRP (or $\sigma q/2^n$ analysis of PMAC as in [75]), we can only prove that the distinguishing advantage is at most $\sigma q/2^n$ where $\sigma$ is in the order of $\ell q$. However, the collision probability of the underlying hash function for $q$ messages may not be in that order. In fact, we show that some hash functions (namely the first chain hash of PMAC+, and the underlying hash of LightMAC) have collision probability in the order of $q^2/2^n$. Thus, the tight analysis of single-keyed constructions even with domain separation is still an open problem. In this paper we study the PRF advantage of the single-keyed composition for a special structure of hash function, namely a family of parallelizable hash. We define this family below.

### 5.1.3 A Family of Parallelizable Hash xPHash

We first define a family of parallelizable hash based on a masking key $\Delta$ and the block cipher $e_K$. However, the masking key can be derived by calling the block cipher for fixed inputs and hence it can be a purely single keyed construction. For the sake of simplicity we first begin with the case having an independent masking key, in case

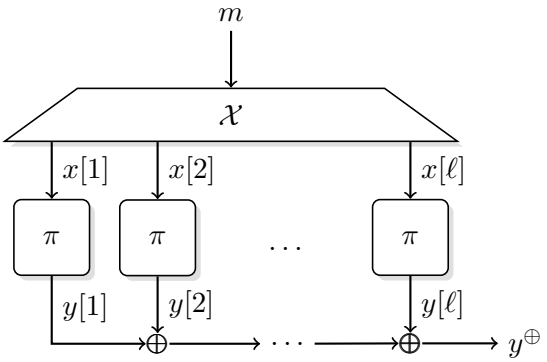


**Figure 5.1.1:** PHash for generic $\mathsf{PMAC}_{\mathcal{X},\pi}$

it is required (e.g., LightMAC does not require any masking key). A hash function is basically computed in three steps (see Fig. 5.1.1):

- **Encoding the Message**: We first encode the message $m$ into $(x[1], \ldots, x[\ell]) \in \mathfrak{B}^{\ell}$. This encoding can be deterministic (in this case no masking key is needed) or can be defined using a masking key $\Delta$. Let us denote this encoding as a function $\mathcal{X}(m)$. We also write $\ell$ as $\|m\|$. In general $\|m\|$ and $\|m\|_n$ are same or differ by at most one.
- **Parallel Block Cipher Calls**: Then, we apply block cipher in parallel to all these blocks obtained through encoding. More precisely, we compute $y[i] = e_K(x[i])$ for all $i \in [\ell]$.
- **Sum the Outputs of the Block Cipher**: Finally, the hash output is defined as the sum $y^{\oplus} = y[1] \oplus \cdots \oplus y[\ell]$.

We call the hash function xPHash and denote it as xPHash. We also write the final output as $\mathsf{xPHash}_{\mathcal{X},K}(m)$. When the encoding function is understood from the context we simply write $\mathsf{xPHash}_{\Delta,K}(m)$ or $\mathsf{xPHash}_K(m)$ (in case the masking key is not used). The hash for the simplified version of LightMAC, PMAC etc. can be instantiated by our generalized hash xPHash as demonstrated below. In all these cases we only have to describe the encoding function as the rest are same for all constructions.

**Hash for LightMAC**. Let $0 < s < n$ be some fixed parameter. Given a message $m$, we pad $10^d$ such that $|m\|10^d|$ is a multiple of $(n - s)$. We choose $d$ to be the smallest nonnegative integer which satisfies the above condition. Let the padded message be $(m[1], \ldots, m[\ell]) \in (\{0,1\}^{n-s})^{\ell}$. Now, the encoding of the message is defined as $\mathcal{X}(m) = (x[1], \ldots, x[\ell]) \in \mathfrak{B}^{\ell}$ where $x[i] = \langle i \rangle_s \| m[i]$ and $\langle i \rangle_s$ is the $s$-bit binary representation of $i$. So the maximum size of $\ell$ should be at most $2^s - 1$ and hence the maximum message size which can be processed in this encoding is at most $(n - s)(2^s - 1) - 1$.

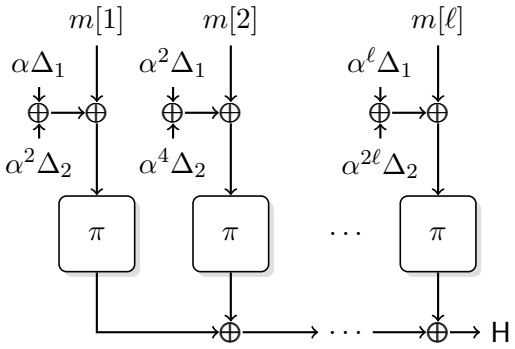


**Figure 5.1.2:** The underlying first chaining hash function used in PMAC+.

**Hash for PMAC**. Given a message $m$, we once again pad $10^d$ for a smallest non-negative $d$ such that $m\|10^d$ is a multiple of $n$. Let the padded message be $(m[1], \ldots, m[\ell]) \in \mathfrak{B}^\ell$. In this encoding a masking key $\Delta$ is used. Let $\gamma_1, \gamma_2, \ldots$ be the Gamma code. Now, the encoding of the message is defined as $\mathcal{X}(m) = (x[1], \ldots, x[\ell]) \in \mathfrak{B}^\ell$ where $x[i] = m[i] \oplus \gamma_i \cdot \Delta$.

**Hash for PMAC1**. The encoding function for PMAC1 is similar to PMAC. Let $\alpha$ be a primitive element in the underlying field $GF(2^n)$. We define $\mathcal{X}(m) = (x[1], \ldots, x[\ell]) \in \mathfrak{B}^\ell$ where $x[i] = m[i] \oplus \alpha^i \cdot \Delta$. In general for any sequence of distinct elements $\omega_1, \omega_2, \ldots, \omega_\ell$ we can define the $i$ block of the encoding of a PMAC-type construction as $x[i] = m[i] \oplus \omega_i \cdot \Delta$.

**Hash for PMAC+**. The encoding function for PMAC+ is little more stronger than PMAC in which two masking keys $\Delta_1, \Delta_2$ are used. In this encoding, the $i$ block of the encoding of a PMAC-type construction as $x[i] = m[i] \oplus \alpha^i \cdot \Delta_1 \oplus \alpha^{2i} \cdot \Delta_2$. This is illustrated in Fig 5.1.2.

N.B. In this chapter, by PMAC+ we mean a simplified version of the original PMAC+. This version is nothing but a variant of PMAC (except the direct xoring of the last block of message) where the endoding of PMAC+ is used instead of the Gamma code. The original PMAC+, as described by Kan Yasuda in [98], has two layers at the intermediate output level and falls in the cateory of DbHtS which has been proven to be beyond birthday bound secure by Datta et al. in [29] and Kim et al. in [58]. The version of PMAC+ we have used here has an advantage because it needs not go through an extra layer of computation (which includes $O(\ell)$ many multiplication by a primitive element of $\mathrm{GF}(2^n)$).

#### 5.1.3.1 Collision Probability for xPHash

We have already seen that the collision probability of the underlying hash function plays an important role in the PRF advantage. In this subsection, we show how to get an exact form of the collision probability. This would be useful in our tight analysis. We first begin with defining odd set for a message with respect to an encoding function.

**Definition 5.1.3** (odd set)**.** Let $\mathcal{X}$ be an encoding function. We define the odd-set corresponding to the message $m$ (with respect to the encoding function $\mathcal{X}$) as

$$\mathsf{odd}_{\mathcal{X}}(m) = \{x \in \mathfrak{B} \ : \ \#\{j \in [\ell] : x[j] = x\} \text{ is odd}\} \tag{5.1}$$

where $\mathcal{X}(m) = (x[1], \ldots, x[\ell])$.

So odd set basically collects all those blocks in $\mathcal{X}(m)$ which appears odd many times. The significance of the odd set definition is prominent when we compute $\mathsf{xPHash}_{\mathcal{X},K}(m)$. Note that $\mathsf{xPHash}_{\mathcal{X},K}(m)$ is same as $\bigoplus_{i=1}^{\ell} e_K(x[i])$. So for all those blocks $x$ which appear even times get canceled in the above sum. In other words,

$$\mathsf{xPHash}_{\mathcal{X},K}(m) = \bigoplus_{x \in \mathsf{odd}_{\mathcal{X}}(m)} e_K(x). \tag{5.2}$$

**Definition 5.1.4** (colliding hash)**.** We say that an encoding function $\mathcal{X}$ is colliding for a pair of messages $m_1, m_2$ if $\mathsf{odd}_{\mathcal{X}}(m_1) = \mathsf{odd}_{\mathcal{X}}(m_2)$. Otherwise, we call it non-colliding. We define colliding advantage as

$$\mathbf{Adv}^{\text{xcoll}}_{\mathcal{X}}(\ell) = \max_{\substack{m_1,m_2 \\ \|m_1\|,\|m_2\| \le \ell}} \mathsf{Pr}[\mathcal{X} \text{ is collding for } m_1, m_2] \tag{5.3}$$

Note that when $\mathcal{X}$ is deterministic, the above would be either zero or one. Whenever $\mathcal{X}$ is colliding for a pair of messages, the hash values also collide for sure. However, the converse need not be true. But, it can be shown that the collision probability of the hash for non-colliding $\mathcal{X}$ is very small. In particular, we have the following useful bound.

**Lemma 5.1.5.** *Let* $\mathsf{H} := \mathsf{xPHash}_{\mathcal{X},\boldsymbol{\pi}}$ *be a parallelizable hash function based on an encoding function* $\mathcal{X}$ *and the random permutation* $\boldsymbol{\pi}$ *(replacing the block cipher). Then,*

$$\mathsf{coll}_{\mathsf{H}}(q, \ell) \le \binom{q}{2} \times \left(\mathbf{Adv}^{\text{xcoll}}_{\mathcal{X}}(\ell) + \frac{1}{2^n - 2\ell}\right) \tag{5.4}$$

**Proof**. Suppose, $m_1, m_2$ are two messages with $\ell_1$, $\ell_2$ blocks respectively. Then, the collision event, denoted as coll, $\mathsf{xPHash}_{\mathcal{X},\boldsymbol{\pi}}(m_1) = \mathsf{xPHash}_{\mathcal{X},\boldsymbol{\pi}}(m_2)$ is equivalent to the

equation of random variables

$$\bigoplus_{x \in \mathsf{odd}(m_1)} \boldsymbol{\pi}(x) \;=\; \bigoplus_{x \in \mathsf{odd}(m_2)} \boldsymbol{\pi}(x) \tag{5.5}$$

The probability that $\mathsf{odd}(m_1) = \mathsf{odd}(m_2)$ holds, is at most $\mathbf{Adv}^{\mathrm{xcoll}}_{\mathcal{X}}(\ell)$.

For notational simplicity we denote $\boldsymbol{\pi}(x)$ as $\mathsf{Y}_x$. Whenever $\mathsf{odd}(m_1) \neq \mathsf{odd}(m_2)$ holds the collision event can be written as $\oplus_{x \in J} \mathsf{Y}_x = 0$ where $J = (\mathsf{odd}(m_1) \setminus \mathsf{odd}(m_2)) \cup (\mathsf{odd}(m_2) \setminus \mathsf{odd}(m_1)) \neq \emptyset$. So by applying Lemma 2.3.1, we have

$$\begin{aligned}\mathsf{Pr}[\mathsf{odd}(m_1) \neq \mathsf{odd}(m_2) \;\wedge\; \mathsf{coll}] &= \mathsf{Pr}[\mathsf{odd}(m_1) \neq \mathsf{odd}(m_2) \;\wedge\; \oplus_{x \in J} \mathsf{Y}_x = 0] \\ &\leq \frac{1}{2^n - 2\ell}.\end{aligned}$$

By combining both cases and applying union bound, the desired result follows. □

**Definition 5.1.6.** We say that $\mathcal{X}$ is $\epsilon$-blockwise universal if for all bit strings $m, m'$ (not necessarily distinct), and for all $i \neq j$ with $i \leq \|m\|$ and $j \leq \|m'\|$, the following holds:

$$\mathsf{Pr}[\mathcal{X}(m)[i] = \mathcal{X}(m')[j]] \leq \epsilon \tag{5.6}$$

Moreover, it is called $\epsilon$-blockwise regular if for all $c \in \mathfrak{B}$, $m \in \{0,1\}^*$ and $i \leq \|m\|$,

$$\mathsf{Pr}[\mathcal{X}(m)[i] = c] \leq \epsilon \tag{5.7}$$

### 5.1.4 A Parallelizable MAC Family

In this paper our main focus is to analyze single-keyed version of hash-then-PRP where the hash function is xPHash. More formally, we describe our MAC family, called xPMAC, as follows: Let $\mathcal{X}$ denote a collection of functions, also called encoding functions, from $\{0,1\}^*$ to $\mathfrak{B}^+$. We assume $\mathcal{X} \leftarrow\!\$\, \mathcal{X}$ and $K \leftarrow\!\$\, \mathcal{K}$ are independent.

$\mathsf{xPMAC}_{\mathcal{X},K}(m)$

1 : **Input** : $m \in \{0,1\}^*$
2 : $(x[1], ..., x[\ell]) \leftarrow \mathcal{X}(m)$
3 : $y[i] e_K[x[i]]$ **for** $i = 1, \ldots, \ell$
4 : $y^{\oplus} \leftarrow y[1] \oplus ... \oplus y[\ell]$
5 : $t \leftarrow e_K(y^{\oplus})$
6 : **return** $t$

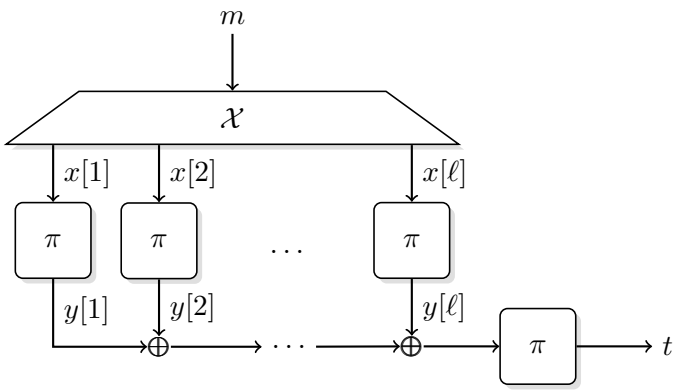


**Figure 5.1.3:** Generic $\mathsf{PMAC}_{\mathcal{X},\pi}$

If the key of $\mathcal{X}$ is $\Delta$ we sometimes write the above MAC as $\mathsf{xPMAC}_{\Delta,K}$ or simply $\mathsf{xPMAC}_K$ (whenever $\Delta$ is not present or when it is derived from the underlying block cipher itself). Let $\boldsymbol{\pi} \xleftarrow{\$} \mathsf{Perm}$ be the random permutation. When we replace the block cipher by the random permutation, we write the MAC as $\mathsf{xPMAC}_{\Delta,\boldsymbol{\pi}}$ or $\mathsf{xPMAC}_{\boldsymbol{\pi}}$ (see Fig.5.1.3).

WELL KNOWN CANDIDATES OF THE FAMILY: PMAC, PMAC1 and single-key version of LightMAC are some candidates of the family. We have already defined the underlying hash functions H for these constructions. The final output is simply defined as $e_K(\mathsf{H}(m))$ where the same $e_K$ is used to define the hash H. Note that the original LightMAC is defined for two independent keys and can be defined as $e_{K'}(\mathsf{H}(m))$.

## 5.2 Main Theorems

**Theorem 5.2.1** (main result)**.** *Suppose $\mathcal{X}$ is $\epsilon$-blockwise universal and independent of the block cipher key. Let xPMAC be the MAC based on $\mathcal{X}$ and $e_K$. Then,*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{xPMAC}}(q, \ell_{\max}, \sigma, t) \leq \mathbf{Adv}^{\mathrm{prp}}_{e}(\sigma, t') + \binom{q}{2} \times \mathbf{Adv}^{\mathrm{xcoll}}_{\mathcal{X}}(\ell_{\max}) + \frac{36q^2}{2^n} + 16q^2 \cdot \epsilon + \ell_{\max} \cdot \sigma \cdot \epsilon.$$

*provided $\ell_{\max} \leq 2^{n/4}$.*

**Theorem 5.2.2** (single-keyed version of the main result)**.** *Suppose $\mathcal{X}_\Delta$ is $\epsilon'$-block wise regular and universal where $\Delta = (\Delta_1, \ldots, \Delta_k)$ and $\Delta_i = e_K(i)$. Then the PRF advantage of*

*the single keyed construction, denoted as 1k-xPMAC is at most*

$$\mathbf{Adv}^{\mathrm{prf}}_{\textit{1k-xPMAC}}(q,\ell_{\max},\sigma,t) \le \mathbf{Adv}^{\mathrm{prp}}_{e}(\sigma,t') + \binom{q}{2} \times \mathbf{Adv}^{\mathrm{xcoll}}_{\mathcal{X}}(\ell_{\max}) + \frac{36q^2}{2^n} + 16q^2 \cdot \epsilon + \ell_{\max} \cdot \sigma \cdot \epsilon + \frac{k(k+2q)}{2^{n+1}} + \frac{4kq}{2^n} + \frac{2\sigma k}{2^n}.$$

*provided* $\ell_{\max} \le 2^{n/4}$.

We postpone the proof of the above two main theorems in Sect. 5.3. We note that for almost all encoding we can choose $\epsilon \le 2^{-n}$. Moreover, $k$ is usually very small constant. So, we can further simplify the bound for the main theorem as $q^2\mathbf{Adv}^{\mathrm{xcoll}}_{\mathcal{X}}(\ell_{\max}) + O(\frac{q^2}{2^n}) + O(\frac{q}{2^{\frac{n}{2}}})$.

### 5.2.1 Applications of Main Theorems

Now we give two applications of our main theorems.

**Proposition 5.2.3.** *The masking of underlying encoding* $\mathcal{X}_{\textit{PMAC+}}$ *of PMAC+ uses two masks* $\Delta_1, \Delta_2$ *and a primitive element* $\alpha$ *from* $\mathbb{F}_{2^n}$. *Suppose, the primitive element* $\alpha$ *is chosen such that* $1 + \alpha^i + \alpha^j + \alpha^k \neq 0$ *for all* $i, j, k \le 2^{n/4}$.*Then, for all* $\ell \le 2^{n/4}$,

$$\mathbf{Adv}^{\mathrm{xcoll}}_{\mathcal{X}_{\textit{PMAC+}}}(\ell) \le \frac{4}{2^n}$$

**Proof**. Let $m, m'$ be such that $\mathsf{odd}(m) = \mathsf{odd}(m')$. Note that for any $i, j$,

$$\begin{aligned} x[i] = x'[j] &\Rightarrow \Delta_1 + (\alpha^i + \alpha^j)\Delta_2 = (\alpha^i + \alpha^j)^{-1}(m[i] + m'[j]) \\ x[i] = x[j] &\Rightarrow \Delta_1 + (\alpha^i + \alpha^j)\Delta_2 = (\alpha^i + \alpha^j)^{-1}(m[i] + m[j]) \\ x'[i] = x'[j] &\Rightarrow \Delta_1 + (\alpha^i + \alpha^j)\Delta_2 = (\alpha^i + \alpha^j)^{-1}(m'[i] + m'[j]) \end{aligned} \tag{5.8}$$

Suppose, $p = \|m\|, p' = \|m'\|$.

**Case 1:** (One of $m$ and $m'$ is prefix of another.) Without any loss of generality, we can assume that $m'$ is a prefix of $m$. Then, $m = m'\|m[(p'+1)...p]$. Now, $\mathsf{odd}(m) = \mathsf{odd}(m') \Rightarrow \mathsf{odd}(m[(p'+1)...p]) = \varnothing \Rightarrow \ell - p$ is even.

*Case when* $p - p' = 2$ : In this case, $x[p'+1] = x[p'+2]$ which is actually $\Delta_1 + (\alpha^{p'+1} + \alpha^{p'+2})\Delta_2 = C$ for $C = (\alpha^{p'+1} + \alpha^{p'+2})^{-1}(m[p'+1] + m[p'+2])$ by 5.8. We are left with only one equation with random variables $\Delta_1, \Delta_2$ and fixed choice of coefficient and

constant. Therefore, Pr $\leq \frac{1}{N}$.

*Case when* $p - p' \geq 4$ : Let $j_1 \in [(p'+2)\dots p]$ such that $x[p'+1] = x[j_1]$. Since $p - p' \geq 4$ and $\mathsf{odd}(m[(p'+1)...p]) = \varnothing$, we can find distinct $i_2, j_2 \in [(p'+2)\dots p]$ which are different from $j_1$ and $x[i_2] = x[j_2]$. Now, from 5.8, we have a system of two equations with appropriate $C_1, C_2$:

$$\begin{aligned}\Delta_1 + (\alpha^{p'+1} + \alpha^{j_1})\Delta_2 &= C_1\\ \Delta_1 + (\alpha^{i_2} + \alpha^{j_2})\Delta_2 &= C_2\end{aligned}$$

The above system has rank=2 due to the assumption on $\alpha$. For each of $j_1, i_2, j_2$ we have at most $\ell$-many choices. Therefore, Pr $\leq \frac{\ell^3}{N^2}$.

**Case 2:** ($m, m'$ are not prefix of one another.) Let $m = (m[1], \dots, m[p])$ and $m' = (m'[1], \dots, m'[p'])$ with $p \geq p'$. We define

$$NEQ(m, m') = \{i \in [p'] : m[i] \neq m'[i]\}.$$

For simplicity it will be denoted by $NEQ$ henceforth. Note that $i \in NEQ \Leftrightarrow x[i] \neq x'[i]$.

*Case when* $|NEQ| = 1$ : Let $NEQ = \{i_1\}$. Then there exists $j \neq k$ such that $x[i] = x[j]$ and $x'[i] = x[k]$. Thus, from 5.8, we get the following system of equations with appropriate $C_1, C_2$:

$$\begin{aligned}\Delta_1 + (\alpha^{i_1} + \alpha^{j})\Delta_2 &= C_1\\ \Delta_1 + (\alpha^{i_1} + \alpha^{k})\Delta_2 &= C_2\end{aligned} \tag{5.9}$$

It has rank=2 since $j \neq k$. Therefore, in this case, Pr $\leq \frac{\ell^2}{N^2}$.

*Case when* $|NEQ| \geq 2$ : Suppose $i_1, i_2$ are the first two (increasing order wise) indices from $NEQ$. If $x[i_1]$(or $x[i_2]$) matches with $x[j]$ or $x'[j]$ and $x'[i_1]$(or $x'[i_2]$) matches with $x[k]$ or $x'[k]$ for some $j \neq k$, then we get a $\frac{\ell^2}{N^2}$ bound following the same line of argument as the previous case. Suppose it does not happen for both $x[i_1]$ and $x[i_2]$. Then, we have

$j_1, j_2$ (both different from $i_1, i_2$) such that $x[i_1] = x[j_1]$ and $x[i_2] = x[j_2]$. 5.8 gives us the following system of equations with appropriate $C_1, C_2$:

$$\begin{aligned} \Delta_1 + (\alpha^{i_1} + \alpha^{j_1})\Delta_2 &= C_1 \\ \Delta_1 + (\alpha^{i_2} + \alpha^{j_2})\Delta_2 &= C_2 \end{aligned} \tag{5.10}$$

By virtue of the assumption on $\alpha$, we have that the above system has rank=2 (otherwise $\alpha^{i_1} + \alpha^{j_1} = \alpha^{i_2} + \alpha^{j_2} \Rightarrow 1 + \alpha^{j_1 - i_1} + \alpha^{i_2 - i_1} + \alpha^{j_2 - i_1} = 0$). Therefore, $\Pr \leq \frac{\ell^2}{N^2}$. This completes the proof of our proposition. □

**Corollary 5.2.4.** *Let $\Delta_i = e_K(i)$, $i = 1, 2$ (masking keys are generated by the underlying block cipher). Let PMAC+′ denote the single-keyed hash-then-PRP based on the underlying single chain hash of PMAC+. Assume that the primitive element $\alpha$ satisfies the condition that for all $i, j, k \leq 2^{n/4}$, $\alpha^i + \alpha^j + \alpha^k \neq 1$ Then,*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{PMAC+}'}(q, \ell_{\max}, \sigma, t) \leq \mathbf{Adv}^{\mathrm{prp}}_{e}(\sigma, t') + \frac{54q^2}{2^n} + \left(\frac{q^2}{2^n}\right)^{1/2} + \frac{11q + 4\sigma}{2^n} \tag{5.11}$$

*provided $\ell_{\max} \leq 2^{n/4}$.*

This follows directly from our single-keyed version of the main theorem (with $k = 2$) and the above proposition. We note that the underlying encoding function is $2^{-n}$-blockwise universal and regular.

Now we state similar result for 1k-LightMAC also. We get this directly from the above single-keyed version of the main result, without any exclusive analysis for this construction. That is why this bound is slightly worse than what we get at chapter 4. Due to counter encoded with the message blocks, the colliding probability for the LightMAC encoding is zero. Moreover, it is 0-blockwise universal (we do not need regular property as there is no masking key). Hence, we have the following PRF advantage for 1k-LightMAC.

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{1k\text{-}LightMAC}}(q, \ell_{\max}, \sigma, t) \leq \mathbf{Adv}^{\mathrm{prp}}_{e}(\sigma, t') + \frac{36q^2}{2^n} \tag{5.12}$$

provided $\ell_{\max} \leq 2^{n/4}$.

## 5.3 Proof of the Main Theorem

By using the classical hybrid argument (moving from the block cipher $e_K$ to the random permutation $\boldsymbol{\pi}$), we have

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{xPMAC}}(q,\ell,\sigma,t) \leq \mathbf{Adv}^{\mathrm{prp}}_{e}(\sigma,t') + \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{xPMAC}_{\mathcal{X},\boldsymbol{\pi}}}(q,\ell,\sigma) \tag{5.13}$$

So it is enough to bound the PRF advantage of the hybrid construction $\mathsf{xPMAC}_{\mathcal{X},\boldsymbol{\pi}}$ (now onwards, we call it real world). We apply H-technique to bound the second term. We apply the extended version of H-technique in which after the query-response phase is over, the adversary gets some additional information. We first define how the real world works.

REAL ORACLE: The real world faithfully simulates the real construction and releases the internal variables, namely all random permutation outputs and the masking key. It first samples $\mathcal{X} \xleftarrow{\$} \mathcal{X}$ and $\boldsymbol{\pi} \xleftarrow{\$} \mathsf{Perm}$ independently. On $i$ query $\mathsf{M}_i$, it just returns $\mathsf{t}_i := \mathsf{xPMAC}_{\mathcal{X},\boldsymbol{\pi}}(\mathsf{M}_i)$. Once the query-response phase is over, it also returns the encoding function $\mathcal{X}$ (equivalently the masking key which uniquely determines the encoding function), and all outputs of $\boldsymbol{\pi}$ during the computation of the underlying hash $\mathsf{xPHash}(\mathsf{M}_i)$ for all $i$. More formally, let

$$\widetilde{\mathsf{M}} = (\mathsf{M}_1,\ldots,\mathsf{M}_q),\ \ \widetilde{\mathsf{T}} = (\mathsf{T}_1,\ldots,\mathsf{T}_q)$$

be the $q$-tuple of distinct queries and its corresponding responses. For all $i \in [q]$, we write $\mathsf{x}_i = \mathcal{X}(\mathsf{M}_i)$. Let $\ell_i = \|\mathsf{M}_i\|$ and so we can write $\mathsf{x}_i$ as $(\mathsf{x}_i[1],\ldots,\mathsf{x}_i[\ell_i])$ (tuple of intermediate inputs). For all $i, a$, we write $\mathsf{y}_i[a] := \boldsymbol{\pi}(\mathsf{x}_i[a])$. The tuple of intermediate outputs for $i$ query is $\mathsf{y}_i := (\mathsf{y}_i[1],\ldots,\mathsf{y}_i[\ell_i])$. We also write the tuples of all intermediate inputs and outputs for all queries as $\widetilde{\mathsf{X}} = (\mathsf{x}_1,\ldots,\mathsf{x}_q),\ \widetilde{\mathsf{Y}} = (\mathsf{y}_1,\ldots,\mathsf{y}_q)$ respectively. The transcript for the real oracle is defined as

$$\tau_{\mathrm{real}} = ((\tilde{\mathsf{M}},\tilde{\mathsf{t}}),\ \mathcal{X},\widetilde{\mathsf{Y}}) \in (\{0,1\}^*)^q \times \mathfrak{B}^q \times \mathcal{X} \times (\mathfrak{B}^+)^q$$

which uniquely determines $\tau'_{\mathrm{real}} := ((\tilde{\mathsf{M}},\tilde{\mathsf{t}}),\widetilde{\mathsf{X}},\widetilde{\mathsf{Y}}) \in (\{0,1\}^*)^q \times \mathfrak{B}^q \times (\mathfrak{B}^+)^q \times (\mathfrak{B}^+)^q$. Let $\mathsf{y}_i^{\oplus} = \mathsf{y}_i[1] \oplus \cdots \oplus \mathsf{y}_i[\ell_i]$.

**Definition 5.3.1** (permutation compatible)**.** Two tuples $\mathsf{a}, \mathsf{b}$ on same index set $J$ are said to be permutation compatible if $\mathsf{a}[j_1] = \mathsf{a}[j_2] \Leftrightarrow \mathsf{b}[j_1] = \mathsf{b}[j_2]$ for every $j_1, j_2 \in J$. It is denoted by $A \sim B$.

Note that two tuples $\mathsf{a}$ and $\mathsf{b}$ are permutation compatible if and only if there is a permutation $\pi$ such that $\pi(\mathsf{a}[j]) = \mathsf{b}[j]$ for all $j$. In other words, equality patterns among $\mathsf{a}$

values exactly matches with the equality patterns among $\mathtt{b}$ values. From the definition of xPMAC, we know that $\boldsymbol{\pi}(\mathsf{y}_i^{\oplus}) = \mathsf{t}_i$ for all $i$. So, $(\widetilde{\mathsf{X}}, \mathsf{y}^{\oplus})$ is permutation compatible with $(\widetilde{\mathsf{Y}}, \tilde{\mathsf{t}})$. We keep this observation in our mind when we simulate the ideal oracle.

**Definition 5.3.2** (real world transcript)**.** A transcript is a tuple $\tau = (\widetilde{m}, \widetilde{t}, \mathcal{X}, \widetilde{y}) \in (\{0,1\}^*)^q \times \mathfrak{B}^q \times \mathcal{X} \times \mathfrak{B}^\sigma$ where $\sigma = \sum_i \ell_i$ where $\ell_i = \|m_i\|$. As described above, a transcript $\tau$ determines uniquely a tuple $\tau' = (\widetilde{m}, \widetilde{t}, \widetilde{x}, \widetilde{y})$ (applying $\mathcal{X}$ function to $m_i$ for all $i$). We call it a *real world transcript* if $(\widetilde{x}, \widetilde{y}^{\oplus})$ is permutation compatible with $(\widetilde{y}, \widetilde{t})$ where $\widetilde{y}^{\oplus} = (y_1^{\oplus}, \ldots, y_q^{\oplus})$ and $y_i^{\oplus} = y_i[1] \oplus \cdots \oplus y_i[\ell_i]$.

From the definition of transcript for real oracle it is clear that real oracle always realizes a real world transcript. Suppose $\sigma'$ ($\leq \sigma$) is the number of distinct blocks present in $(\widetilde{y}, \widetilde{t})$. Then,

$$\mathsf{Pr}[\tau_{\text{real}} = \tau] = \frac{1}{|\mathcal{X}|} \times \frac{1}{(2^n)_{\sigma'}}. \tag{5.14}$$

IDEAL ORACLE: On $i$ query $\mathsf{M}_i$, the ideal oracle returns $\mathsf{T}_i$ values randomly from $\mathfrak{B}$. We define $\mathtt{BadT}$ holds if $\mathsf{T}$ values collide. Clearly,

$$\mathsf{Pr}(\mathtt{BadT}) \leq \frac{q^2}{2^{n+1}}. \tag{5.15}$$

Throughout the ideal oracle we define different bad events and we bound the probability of those events. When we proceed with the definition of the subsequent variables, we can assume that the all previous bad events do not hold (for the completeness of the ideal oracle, one can define the remaining transcript arbitrarily whenever a bad sets true). So, let us assume that $\mathsf{T}$ is a $q$-tuple of distinct blocks. We write query and response tuples as $\widetilde{\mathsf{M}}$ and $\widetilde{\mathsf{T}}$ respectively. Note that the query tuple is completely determined by the response tuple. We write $\ell_i = \|\mathsf{M}_i\|$ and $\ell_{\max} = \max_i \ell_i$. Once the query-response phase is over, it samples the encoding function $\mathcal{X}$ randomly and independently from the query and responses. We say that $\mathtt{BadEncode}$ holds if

1. the encoding function $\mathcal{X}$ is a colliding function for the message tuple or
2. For some $i$, there exists $a < b \leq \ell_i$ such that $\mathsf{x}_i[a] = \mathsf{x}_i[b]$.

As the hash function $\epsilon$-block-wise universal, the probability that $\mathsf{x}_i[a] = \mathsf{x}_i[b]$ holds for any fixed $i, a, b$, is at most $\epsilon$. So, by using the union bound,

$$\mathsf{Pr}(\mathtt{BadEncode}) \leq \binom{q}{2} \cdot \mathbf{Adv}_{\mathcal{X}}^{\text{xcoll}}(\ell) + \ell_{\max} \cdot \sigma \cdot \epsilon. \tag{5.16}$$

So let us assume that BadT and BadEncode do not hold. We can then sample the y values randomly depending on the equalities among x-values. We define the y values in two stages. At the first stage we sample z-values so that $\widetilde{\mathsf{Z}}$ is permutation compatible with $\widetilde{\mathsf{X}}$. Let $\mathcal{I}$ be the set of pairs $(i, a)$, for all $a \in [\ell_i]$ and $i \in [q]$. In the second stage, we reset some z-values chosen in the first stage. The revised tuple would be denoted as $\widetilde{\mathsf{Y}}$.

**First stage**: Let

$$\mathsf{z}_x \xleftarrow{\text{wor}} \mathfrak{B} \text{ for every } x \in \P := \{x : x = \mathsf{x}_i[a], (i, a) \in \mathcal{I}\}$$

and define $\mathsf{z}_i^{\oplus} := \oplus_a \mathsf{z}_i[a]$ where $\mathsf{z}_i[a] := \mathsf{z}_x$ for $x = \mathsf{x}_i[a]$. In other words, z-values sampled as if these are computed through a random permutation $\boldsymbol{\pi}$ on all x-values. More precisely, the distribution of $\widetilde{\mathsf{Z}} := (\mathsf{z}_i[a] : (i, a) \in \mathcal{I})$ is same as that of $(\boldsymbol{\pi}(\mathsf{x}_i[a]) : (i, a) \in \mathcal{I})$. So $\widetilde{\mathsf{Z}}$ is permutation compatible with $\widetilde{\mathsf{X}}$. As the final outputs are sampled independently, the permutation compatibility between $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{Z}}^{\oplus})$ and $(\widetilde{\mathsf{Z}}, \widetilde{\mathsf{T}})$ may not hold. Let us first define two types of collisions which would be required to define our second stage sampling to make it permutation compatible.

**Definition 5.3.3** (full collision and t-collision index)**.**

1. A query index $i \in [q]$ is called a *full collision index* if $\mathsf{x}_i[a] = \mathsf{z}_j^{\oplus}$ for some $(i, a) \in \mathcal{I}, j \in [q]$. The set of all full collision indices is denoted by $I_{\mathsf{FC}}$.

2. A query index $i \in [q] \setminus I_{\mathsf{FC}}$ is called a *t-collision index* if $\mathsf{z}_i[a] = \mathsf{t}_j$ for some $(i, a) \in \mathcal{I}, j \in [q]$. The set of all t-collision indices is denoted by $I_{\mathsf{TC}}$.

   By definition these two sets are disjoint.

We want $\mathsf{z}_i^{\oplus}$ to be fresh for any $i \in I^{\neq} := I_{\mathsf{FC}} \cup I_{\mathsf{TC}}$. This would actually help us to satisfy the permutation compatible property (and hence to be a real-world realizable transcript). Keeping this in mind, we define the bad event which occurs due to the first stage sampling (i.e. due to z-values)

**Definition 5.3.4** (BadZ)**.** for some $i, j, k \in [q], a \in [\ell_j], b \in [\ell_k]$,

BadZ1: $\mathsf{z}_i^{\oplus} = \mathsf{z}_j^{\oplus}$ for some $i \neq j$;

BadZ2: $\mathsf{z}_i^{\oplus} = \mathsf{x}_j[a] \wedge \mathsf{Z}_j^{\oplus} = \mathsf{x}_k[b]$;

BadZ3: $\mathsf{z}_j[a] = \mathsf{T}_i \wedge \mathsf{z}_j^{\oplus} = \mathsf{x}_k[b]$;

Let us assume that BadZ does not hold. So all the values of $\mathsf{z}^{\oplus} := (\mathsf{z}_i^{\oplus} : i \in [q])$ are distinct. The same is true for $\widetilde{\mathsf{T}}$. Hence,

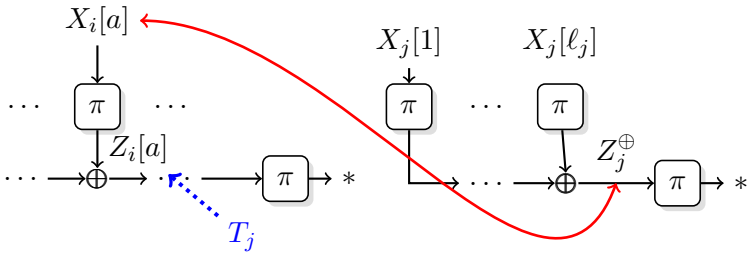


**Figure 5.3.1:** Resetting of z in case of full-collision. Here the red line represents a collision in the first stage sampling. The blue dotted edge represents the resetting in the second stage sampling.

$rvz^{\oplus} \sim \widetilde{\mathsf{T}}$. For all $i \in I_{\mathsf{FC}}$ there is $a$ for which $\mathsf{x}_i[a] = \mathsf{z}_j^{\oplus}$ for some $j \in [q]$. So we must redefine $\mathsf{z}_i[a]$ to $\mathsf{t}_j$. We should apply this revision to all those blocks for which full-collision happen. After doing this revision, for all $i \in I_{\mathsf{TC}}$, there is $a$ such that $\mathsf{z}_i[a] = \mathsf{t}_j$ for some $j$. We should not keep this as it is since $\mathsf{x}_i[a] \neq \mathsf{z}_j^{\oplus}$ (note, $i \notin I_{\mathsf{FC}}$). So we must revise them to some other values which do not appear anywhere in z values or t values.

**Second stage**: Sampling at the second stage is defined as follows: Given a good z we reset it to y now by keeping most of the variables unchanged.

- Let $J_{\mathsf{FC}}$ be the set of all query indices $j$ such that $\mathsf{z}_j^{\oplus} = \mathsf{x}_i[a]$ for some $(i, a) \in \mathcal{I}$. For all these $j$, we define $\mathsf{R}_j$ as $\mathsf{z}_i[a]$. For all other values of $j$, we sample $\mathsf{R}_j$ in a without replacement manner from the set $\mathfrak{B}^q \setminus (\{\mathsf{t}_1, ..., \mathsf{t}_q\} \cup \{\mathsf{z}_i[a] : (i, a) \in \mathcal{I}\})$.
- We initially set $\mathsf{y} \leftarrow \mathsf{z}$.
- For all those $(i, a)$, $\mathsf{x}_i[a] = \mathsf{z}_j^{\oplus}$ (i.e. $i \in I_{\mathsf{FC}}$), we define $\mathsf{y}_i[a] = \mathsf{t}_j$.
- For all those $(i, a)$ which does not satisfy the above condition and $\mathsf{x}_i[a] = \mathsf{t}_j$ (i.e. $i \in I_{\mathsf{TC}}$), we define $\mathsf{y}_i[a] \leftarrow \mathsf{R}_j$.

Note that for all $i \notin I^{\neq}$, $\mathsf{y}_i^{\oplus} = \mathsf{z}_i^{\oplus}$. In other words, due to the resetting of z values, only for $i \in I^{\neq}$, $\mathsf{z}_i^{\oplus}$ is changed to a new value $\mathsf{y}_i^{\oplus}$. For a good $\widetilde{\mathsf{Z}}$, $\mathsf{z}_i^{\oplus}$ is fresh for all these $i$. So we want that for all these $i$, $\mathsf{y}_i^{\oplus}$ is fresh. In other words, being not fresh for all those values would be defined to be bad.

**Definition 5.3.5** (BadY)**.** For some $i, j, k \in [q], a \in [\ell_i], b \in [\ell_k]$ such that $i < j$,

BadY1.1: $\mathsf{z}_i[a] = \mathsf{t}_j \wedge \mathsf{y}_i^{\oplus} = \mathsf{x}_k[b]$

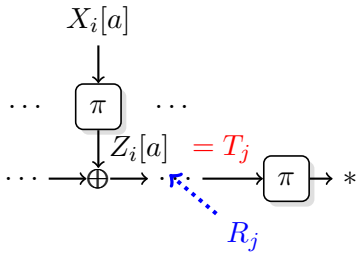


**Figure 5.3.2:** Resetting of z in case of t-collision. Here the red equality represents the collision in the first stage sampling. The blue dotted edge represents the resetting in the second stage sampling.

BadY1.2: $\mathsf{z}_i[a] = \mathsf{t}_j \wedge \mathsf{y}_i^{\oplus} = \mathsf{y}_k^{\oplus}$

BadY2.1: $\mathsf{x}_i[a] = \mathsf{z}_j^{\oplus} \wedge \mathsf{y}_i^{\oplus} = \mathsf{x}_k[b]$

BadY2.2: $\mathsf{x}_i[a] = \mathsf{z}_j^{\oplus} \wedge \mathsf{y}_i^{\oplus} = \mathsf{y}_k^{\oplus}$

**Lemma 5.3.6.**

$$\Pr[\mathtt{BadZ} \vee \mathtt{BadY}] \leq \frac{2q^2}{2^n} + \frac{33q^3\ell_{\max}^2}{2^{2n}} + \frac{\epsilon q^3 \ell_{\max}}{2^n - \sigma} + \frac{\epsilon q^2 \ell_{\max}^2}{2^n - \sigma} + \frac{3\epsilon q^4 \ell_{\max}^2}{(2^n - \sigma)^2}.$$

We give the proof of the lemma in Sect. 5.3.2. Finally, a transcript of the ideal world is the tuple,

$$\tau_{\text{ideal}} := ((\tilde{\mathsf{M}}, \tilde{\mathsf{t}}), \mathcal{X}, \widetilde{\mathsf{Y}}).$$

As before, this also uniquely determines the tuple $\tau'_{\text{ideal}} := ((\tilde{\mathsf{M}}, \tilde{\mathsf{t}}), \widetilde{\mathsf{X}}, \widetilde{\mathsf{Y}})$.

GOOD/BAD TRANSCRIPT: A transcript is called bad if one of the bad events holds. More precisely, bad is the event which is the union $\mathtt{BadT} \vee \mathtt{BadEncode} \vee \mathtt{BadZ} \vee \mathtt{BadY}$. The set of all bad transcripts is denoted as $\mathcal{V}_{\mathsf{bad}}$. From the Lemma 5.3.6, Eq. 5.15 and Eq. 5.16 the probability that an ideal transcript is bad is at most

$$\frac{3q^2}{2^n} + \frac{33q^3\ell_{\max}^2}{2^{2n}} + \frac{2\epsilon q^3 \ell_{\max}}{2^n} + \frac{2\epsilon q^2 \ell_{\max}^2}{2^n} + \frac{12\epsilon q^4 \ell_{\max}^2}{2^{2n}} + \ell_{\max} \cdot \sigma \cdot \epsilon + \binom{q}{2} \cdot \mathbf{Adv}_{\mathcal{X}}^{\text{xcoll}}(\ell). \quad (5.17)$$

Now we apply our assumption that $\ell_{\max} \leq 2^{n/2}$. This would simply the above bound for the bad event as

$$\frac{36q^2}{2^n} + 16q^2 \cdot \epsilon + \ell_{\max} \cdot \sigma \cdot \epsilon + \binom{q}{2} \cdot \mathbf{Adv}_{\mathcal{X}}^{\text{xcoll}}(\ell). \quad (5.18)$$

The set of all good transcripts is denoted by $\mathcal{V} \setminus \mathcal{V}_{\mathsf{bad}}$. It is easy to see from the above discussion that for a good transcript $(\tilde{\mathsf{y}}, \tilde{\mathsf{t}})$ is permutation compatible with $(\tilde{\mathsf{x}}, \tilde{\mathsf{y}^{\oplus}})$. So, a good transcript is always a real world transcript.

**Lemma 5.3.7.** *For all good transcript $\tau$,*

$$\mathsf{Pr}[\tau_{\text{real}} = \tau] \geq \mathsf{Pr}[\tau_{\text{ideal}} = \tau]$$

*Proof.* Let $\tau = (\widetilde{m}, \widetilde{t}, \mathcal{X}, \widetilde{y})$ be a good transcript which determines a tuple $\widetilde{x}$. Since $\tau$ is good $(\widetilde{x}, \widetilde{y}^{\oplus}) \sim (\widetilde{y}, \widetilde{t})$. Let $\sigma'$ be the number of distinct blocks present in $(x, y^{\oplus})$. Fix any $q$-tuple of distinct blocks $\widetilde{r} = (r_1, \ldots, r_q)$. To each $\widetilde{r}$, we define a function

$$f_{\widetilde{r}}(\widetilde{y}) = \widetilde{z} \text{ where } \begin{cases} z_i[a] = r_j \text{ if } y_i[a] = t_j \\ z_i[a] = t_j \text{ if } y_i[a] = r_j \\ z_i[a] = y_i[a] \text{ otherwise} \end{cases}$$

Note that for every good transcript there is a tuple $\widetilde{z}$ which is reset to $\widetilde{y}$ as defined in the second stage sampling. We claim that if $\widetilde{z}$ is any such tuple which reset to $\widetilde{y}$ in the second stage sampling, there exists $\widetilde{r}$ for which $f_{\widetilde{r}}(\widetilde{y}) = \widetilde{z}$. To see this we define $\widetilde{r}$ as follows. For all $i$ where $z_i^{\oplus} = x_j[a]$ for some $(j, a) \in \mathcal{I}$, we define $r_i$ as $z_j[a]$. For all $i$ such that there is some $(j, a) \in \mathcal{I}$ with $z_j[a] = t_i$, we define $r_i$ as $y_j[a]$. For all other $i$ values, we define $r_i$ arbitrarily. Now one can check that for this $\widetilde{r}$, $f_{\widetilde{r}}(\widetilde{y}) = \widetilde{z}$. So the number of $\widetilde{z}$ values which map to $\widetilde{y}$ is at most $2^{nq}$. In other words, the set $\mathcal{S}_{\widetilde{y}}$, of all pairs $(\widetilde{z}, \widetilde{r})$ which reset to $\widetilde{y}$ has size at most $2^{nq}$. Let us now calculate $\mathsf{Pr}[\tau_{\text{real}} = \tau]$ and $\mathsf{Pr}[\tau_{\text{ideal}} = \tau]$ for a good transcript $\tau$. We already know that

$$\mathsf{Pr}[\tau_{\text{real}} = \tau] = \frac{1}{|\mathcal{X}|} \times \frac{1}{(2^n)_{\sigma'}} \tag{5.19}$$

$$\begin{aligned} \mathsf{Pr}[\tau_{\text{ideal}} = \tau] &= \frac{1}{|\mathcal{X}|} \times \mathsf{Pr}[\widetilde{\mathsf{Y}} = y] \times \frac{1}{2^{nq}} \\ &\leq \frac{1}{|\mathcal{X}|} \times \frac{1}{2^{nq}} \times \sum_{(\widetilde{z},\widetilde{r}) \in \mathcal{S}_{\widetilde{y}}} \mathsf{Pr}[\widetilde{\mathsf{Z}} = \widetilde{z} \wedge \mathsf{R} = \widetilde{r}] \\ &\leq \frac{1}{|\mathcal{X}|} \times \frac{1}{2^{nq}} \times \sum_{(\widetilde{z},\widetilde{r}) \in \mathcal{S}_{\widetilde{y}}} \frac{1}{(2^n)_{\sigma'}} \\ &\leq \frac{1}{|\mathcal{X}|} \times \frac{1}{(2^n)_{\sigma'}} \end{aligned}$$

This proves our result. □ □

By using H-technique and the Eq. 5.18 (the bound for bad events) and Lemma 5.3.7 (the analysis for good transcripts) the proof of the theorem follows.

### 5.3.1 Proof of 5.2.2

Now we briefly describe how the similar proof for purely single-keyed construction works. The real world (actually the hybrid world where the block cipher is replaced by the random permutation as before). The ideal world should now check whether the inputs $1, \ldots, k$ appears anywhere to the intermediate computations (except to compute $\Delta$ values) and similarly, the $\Delta$ values appeared anywhere in the intermediate outputs. We need to add some additional bad events for this types of collision. More formally, the ideal world works as follows. It samples $\Delta_1, \ldots \Delta_k$ randomly (instead of $\mathcal{X}$) after the query-response is over. In addition to `BadEncode`, we also have to add a bad event for collision among $\Delta$ values and collision between $\Delta$ values and $rvt$ values. This costs extra $k(k+2q)/2^{n+1}$ probability. When we sample z values and y values we need to avoid $\Delta$ values. This would again cost $2\sigma k/2^n$ advantage. Finally, the collision between $\mathsf{z}^{\oplus}$ and $\mathsf{y}^{\oplus}$ values with $1, \ldots, k$ would cost $2kq/(2^n - \sigma)$.

So in addition to the previous bad event probability we need to add the following for bad events:

$$\frac{k(k+2q)}{2^{n+1}} + \frac{2kq}{2^n - \sigma} + \frac{2\sigma k}{2^n}.$$

By adding this additional bad probability to our previous theorem, we obtain our result for single-keyed construction.

### 5.3.2 Proof of Lemma 5.3.6

First we set some new notations and introduce some required definitions before entering the proof of our left over lemmas.

NOTATIONS: For any $(i, a) \in \mathcal{I}$, $x_i^{\backslash a}$ denotes the $(\ell_i - 1)$-tuple $(x[1], \ldots, x[a-1], x[a+1], \ldots, x[\ell_i])$. Similarly, $\mathsf{Z}_i^{\oplus \backslash a}$ (or $\mathsf{Y}_i^{\oplus \backslash a}$) denotes $\bigoplus_{b \neq a} \mathsf{Z}_i[b]$ (or $\bigoplus_{b \neq a} \mathsf{Y}_i[b]$). When an encoding $x_i = (x_i[1], \ldots, x_i[\ell_i])$ of $m_i$ is given, we will write $\mathsf{odd}(x_i)$ (or $\mathsf{odd}(x_i^{\backslash a})$) to mean $\mathsf{odd}(m_i)$ (or $\{x_i[b] : b \neq a, \#\{c \in [\ell_i] : c \neq a, x_i[c] = x_i[b]\}$ is odd $\}$).

GENERALIZED DEFINITION OF ODD-SET: $\mathsf{Z}_i[a]$, $\mathsf{Z}_i^{\oplus}$, $\mathsf{Z}_i^{\oplus \backslash a}$ are called z-terms. We define $\mathsf{odd}(\mathsf{Z}_i[a]) = \{x_i[a]\}$, $\mathsf{odd}(\mathsf{Z}_i^{\oplus}) = \mathsf{odd}(x_i)$, $\mathsf{odd}(\mathsf{Z}_i^{\oplus \backslash a}) = \mathsf{odd}(x_i^{\backslash a})$. We generalize the odd-set definition further for any linear combination of z-terms:

$$\mathsf{odd}(b_1 \cdot \mathsf{Z}_i[a] \oplus b_2 \cdot \mathsf{Z}_j^{\oplus} \oplus b_3 \cdot \mathsf{Z}_k^{\oplus \backslash b}) := b_1 \cdot \mathsf{odd}(\mathsf{Z}_i[a]) \triangle b_2 \cdot \mathsf{odd}(\mathsf{Z}_j^{\oplus}) \triangle b_3 \cdot \mathsf{odd}(\mathsf{Z}_k^{\oplus \backslash b})$$

for $b_1, b_2, b_3 \in \{0, 1\}$. Here, for sets $A, B$, $A \triangle B$ is the *symmetric difference* between $A, B$ defined as $(A \setminus B) \cup (B \setminus A)$. Also, $0 \cdot A$ means $\varnothing$ and $1 \cdot A$ means $A$. We say $b_1 \cdot \mathsf{Z}_i[a] \oplus$

$b_2 \cdot \mathsf{Z}_j^{\oplus} \oplus b_3 \cdot \mathsf{Z}_k^{\oplus\setminus b} = uivb'_1 \cdot \mathsf{Z}_{i'}[a'] \oplus b'_2 \cdot \mathsf{Z}_{j'}^{\oplus} \oplus b'_3 \cdot \mathsf{Z}_{k'}^{\oplus\setminus b'}$ if their corresponding odd sets are equal.

Any equation involving z-terms are called z-equations. All z-equations that we will consider for the rest of this section will be of the form

$$b_1 \cdot \mathsf{Z}_i[a] \oplus b_2 \cdot \mathsf{Z}_j^{\oplus} \oplus b_3 \cdot \mathsf{Z}_k^{\oplus\setminus b} = C$$

where $b_1, b_2, b_3 \in \{0,1\}$ and $C$ is a term without any z variable. For any such z-equation $\mathcal{E}$, the combination $b_1 \cdot \mathsf{Z}_i[a] \oplus b_2 \cdot \mathsf{Z}_j^{\oplus} \oplus b_3 \cdot \mathsf{Z}_k^{\oplus\setminus b}$ is denoted by $\mathsf{z}_{\mathcal{E}}$. Here we will deal only with system of equations having at most three z-equations. Rank of a system of z-equation can be computed as follows:

1. **One equation:** A z-equation $\mathcal{E}$ has rank=1 if $\mathsf{odd}(\mathsf{z}_{\mathcal{E}}) \neq \varnothing$ (we call it nonempty equation).
2. **Two equations:** A system of two nonempty z-equations $\mathcal{E}_1, \mathcal{E}_2$ has rank=2 if $\mathsf{odd}(\mathsf{z}_{\mathcal{E}_1}) \neq \mathsf{odd}(\mathsf{z}_{\mathcal{E}_2})$.
3. **Three equations:** A system of three nonempty distinct z-equations $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$ has rank=3 if $\mathsf{odd}(\mathsf{z}_{\mathcal{E}_1}) \triangle \mathsf{odd}(\mathsf{z}_{\mathcal{E}_2}) \triangle \mathsf{odd}(\mathsf{z}_{\mathcal{E}_3}) \neq \varnothing$.

While we bound the probability of bad events, we need to deal linear equations in z values. Note that z can be viewed as $\boldsymbol{\pi}(\mathsf{x})$ for the corresponding x value. To simplify some of the bad events, we need to consider some auxiliary bad events. Auxiliary bad event BadAux is defined as BadAux1 $\vee$ BadAux2.

**Definition 5.3.8** (BadAux1)**.** $\mathsf{Z}_i[1] = \mathsf{T}_j$ for some $i, j \in [q]$.

The following bad event says that more than one full collision or t-collision cannot occur for a same query index. This would help us to have simpler expression of $\mathsf{y}^{\oplus}$ values. In particular, for all those $i$, $\mathsf{z}_i^{\oplus}$ is reset to $\mathsf{y}_i^{\oplus}$, we only have to replace one block of z values by another block of either r value or t value depending on whether it corresponds to T-collision or full collision.

**Definition 5.3.9** (BadAux2)**.** For some $i, j, k \in [q], a, b \in [\ell_j]$,

$$\mathsf{Z}_j[a] \stackrel{*_1}{=} \mathsf{T}_i \text{ or } \mathsf{X}_j[a] \stackrel{*_2}{=} \mathsf{Z}_i^{\oplus}$$

and

$$\mathsf{Z}_j[b] \stackrel{*'_1}{=} \mathsf{T}_k \text{ or } \mathsf{X}_j[b] \stackrel{*'_2}{=} \mathsf{Z}_k^{\oplus}, a \neq b$$

**Bound for** `BadAux1`**:** Clearly, we have

$$\mathsf{Pr}[\texttt{BadAux1}] \leq \frac{q^2}{2^n} \tag{5.20}$$

`Remarks and Conventions.`

1. In what follows we will use rank of system as computed above and lemma 2.3.1 time and again to bound the probability of events. In the next calculation those references are mentioned everywhere we use it. After that the references will be omitted to avoid clumsiness.

2. While dealing with probability calculations of some bad event it will be assumed that all bad events whose probabilistic bound has been obtained before do not occur together with that event. For example bound for `BadZ` will actually mean bound for $\texttt{BadZ} \wedge \neg\texttt{BadT} \wedge \neg\texttt{BadEncode} \wedge \neg\texttt{BadAux}$.

**Bound for** `BadAux2`**:**

$$\mathsf{Pr}[\texttt{BadAux2}] \leq \frac{14q^3\ell_{\max}^2}{2^{2n}} \tag{5.21}$$

This is basically union of four events two of which are identical. We bound the probability for each case individually:

$*_1 \wedge *_1'$ : Corresponding system of equations to this case is $\mathsf{Z}_j[a] = \mathsf{T}_i \wedge \mathsf{Z}_j[b] = \mathsf{T}_k$. It has rank 2 since $\mathsf{Z}_j[a] =\equiv \mathsf{Z}_j[b]$ implies $\mathsf{X}_j[a] = \mathsf{X}_j[b]$ which is an `BadEncode` event due to $a \neq b$. Thus, applying lemma 2.3.1, we get $\mathsf{Pr} \leq \frac{q^3\ell_{\max}^2}{2^n(2^n-1)}$.

$*_1 \wedge *_2'$ : (having same probability with $*_2 \wedge *_1'$): Here, the system of equations under consideration is $\mathsf{Z}_j[a] = \mathsf{T}_i \wedge \mathsf{Z}_k^{\oplus} = \mathsf{X}_j[b]$. For this case, we have two possibilities as listed below.

> *Case 1:* $(\ell_k = 1, \mathsf{Z}_k[1] \not\equiv \mathsf{Z}_j[a])$
>
> This case implies rank of $\mathsf{Z}_j[a] = \mathsf{T}_i \wedge \mathsf{Z}_k^{\oplus} = \mathsf{X}_j[b]$ is 2. Thus, $\mathsf{Pr} \leq \frac{q^3\ell_{\max}^2}{2^n(2^n-\sigma)}$ by lemma 2.3.1. Note that $(\ell_k = 1, \mathsf{Z}_k[1] \equiv \mathsf{Z}_j[a])$ is not a valid possibility since it implies `BadAux1`.
>
> *Case 2:* $(\ell_k > 1, \mathsf{Z}_k[1] \not\equiv \mathsf{Z}_j[a])$
>
> This case implies that the system of equations of $*_1 \wedge *_2'$ has rank 2. Thus, $\mathsf{Pr} \leq \frac{q^3\ell_{\max}^2}{2^n(2^n-\sigma)}$ by lemma 2.3.1.

$*_2 \wedge *_2'$ : System of equations that we get in this case is $\mathsf{Z}_i^{\oplus} = \mathsf{X}_j[a] \wedge \mathsf{Z}_k^{\oplus} = \mathsf{X}_j[b]$. Now, $\mathsf{Z}_i^{\oplus} \equiv \mathsf{Z}_k^{\oplus}$ implies $\mathsf{X}_j[a] = \mathsf{X}_j[b]$ for $a \neq b$ which is a `BadEncode` event. So, $\mathsf{Z}_i^{\oplus} \not\equiv \mathsf{Z}_k^{\oplus}$ has rank 2. Thus, $\mathsf{Pr} \leq \frac{q^3\ell_{\max}^2}{(2^n-\sigma)^2}$ by lemma 2.3.1. After applying $\sigma < 2^{n-1}$ (since otherwise, the bound is vacuously true), we get the desired bound.

**Bound for** `BadZ` :

$$\mathsf{Pr}[\mathtt{BadZ}] \leq \frac{q^2}{2^n} + \frac{4q^3\ell_{\max}^2}{2^{2n}} \tag{5.22}$$

1. `BadZ1` : $\mathsf{Z}_i^{\oplus} = \mathsf{Z}_j^{\oplus}$ for some $i < j$. Therefore,

$$\mathsf{Pr}[\mathtt{BadZ1}] \leq \frac{q^2}{2 \cdot (2^n - \sigma)}$$

2. `BadZ2` : $\mathsf{z}_i^{\oplus} = \mathsf{X}_j[a] \wedge \mathsf{Z}_j^{\oplus} = \mathsf{X}_k[b]$ for some $i, j, k \in [q], a \in [\ell_j], b \in [\ell_k]$ such that $i < j$.

Now, $\mathsf{Z}_i^{\oplus} \equiv \mathsf{Z}_j^{\oplus}$ can not happen since it implies $\mathsf{odd}(\mathsf{x}_i) = \mathsf{odd}(\mathsf{x}_j)$ which is a `BadEncode` event. Thus, $\mathsf{Z}_i^{\oplus} \not\equiv \mathsf{Z}_j^{\oplus}$ and so the above system of equations has rank 2. Therefore,

$$\mathsf{Pr}[\mathtt{BadZ2}] \leq \frac{q^3\ell_{\max}^2}{2 \cdot (2^n - \sigma)^2}$$

as we can choose the indices in $q^3\ell_{\max}^2/2$ ways.

3. `BadZ3` : $\mathsf{Z}_j[a] = \mathsf{T}_i \wedge \mathsf{Z}_j^{\oplus} = \mathsf{X}_k[b]$ for some $i, j, k \in [q], a \in [\ell_j], b \in [\ell_k]$.

In this case, $\ell_j$ must be greater than 1. Otherwise, `BadAux1` holds. So, $\ell_j > 1$ implies that the system of z-equations defining `BadZ3` has rank 2. Therefore,

$$\mathsf{Pr}[\mathtt{BadZ3}] \leq \frac{q^3\ell_{\max}^2}{2^n(2^n - \sigma)}$$

Summing up bounds for all these cases and using the assumption that $\sigma \leq 2^{n-1}$, we get the desired bound for `BadZ`.

**Bound for** `BadY` :

1. `BadY1.1` : $\mathsf{Z}_i[a] = \mathsf{T}_j \wedge \mathsf{Y}_i^{\oplus} = \mathsf{X}_k[b]$ for some $i, j, k \in [q], a \in [\ell_i], b \in [\ell_k]$.

Here again note that $a > 1$ to avoid BadAux1 event. The defining system of equations is equivalent with $\mathsf{Z}_i[a] = \mathsf{T}_j \wedge \mathsf{Y}_i^{\oplus\setminus a} = \mathsf{X}_k[b] \oplus \mathsf{R}_j$. This system has only one z-equation. Randomness for the other equation will be calculated from $\mathsf{R}_j$ which can take a particular value with probability at most $1/(2^n - \sigma)$. So, in this case we have

$$\mathsf{Pr} \leq \frac{q^3 \ell_{\max}^2}{2^n(2^n - \sigma)}$$

2. BadY1.2 : $\mathsf{Z}_i[a] = \mathsf{T}_j \wedge \mathsf{Y}_i^{\oplus} = \mathsf{Y}_k^{\oplus}$ for some $i, j, k \in [q], a \in [\ell_i]$.

It has two subcases.

*Case 1:*($k \notin I_{\mathsf{TC}}$)

In this case the defining system of equations of BadY1.2 is equivalent with $\mathsf{Z}_i[a] = \mathsf{T}_j \wedge \mathsf{Y}_i^{\oplus\setminus a} = \mathsf{Y}_k^{\oplus} \oplus \mathsf{R}_j$ which is bounded by $\mathsf{Pr} \leq \frac{q^3 \ell_{\max}}{2^n(2^n - \sigma)}$.

*Case 2:*($k \in I_{\mathsf{TC}}$)

Here we get $\mathsf{Z}_i[a] = \mathsf{T}_j \wedge \mathsf{Y}_i^{\oplus\setminus a} = \mathsf{Y}_k^{\oplus\setminus c} \oplus \mathsf{R}_j \oplus \mathsf{R}_l \wedge \mathsf{Z}_k[c] = \mathsf{T}_l$ as an equivalent system to the defining system of equations. We claim that it has rank=3. Note that $\mathsf{Y}_i^{\oplus\setminus a} = \mathsf{Z}_i^{\oplus\setminus a}$ and $\mathsf{Y}_k^{\oplus\setminus c} = \mathsf{Z}_k^{\oplus\setminus c}$ because $\neg$BadAux2 prevents any tuple $\mathsf{R}_r$ to get reset at two different places. Therefore, $j \neq l$. Otherwise, it will imply $\mathsf{Z}_i^{\oplus} = \mathsf{Z}_k^{\oplus}$ which is an BadZ event. Rank 3 follows directly from $j \neq l$. Therefore,

$$\mathsf{Pr} \leq \frac{q^4 \ell_{\max}^2}{2^n(2^n - \sigma)^2}$$

3. BadY2.1 : $\mathsf{X}_i[a] = \mathsf{Z}_i^{\oplus} \wedge \mathsf{Y}_i^{\oplus} = \mathsf{X}_k[b]$ for some $i, j, k \in [q], a \in [\ell_i], b \in [\ell_k]$.

By simply following arguments like before we arrive at an equivalent system of z-equations $\mathsf{Z}_j^{\oplus} = \mathsf{X}_i[a] \wedge \mathsf{Z}_i^{\oplus\setminus a} = \mathsf{X}_k[b] \oplus \mathsf{T}_j$. When it has rank 2 (i.e., $\mathsf{Z}_j^{\oplus} \not\equiv \mathsf{Z}_i^{\oplus\setminus a}$) is easily taken care of getting

$$\mathsf{Pr} \leq \frac{q^3 \ell_{\max}^2}{2^n(2^n - \sigma)}$$

So, suppose $\mathsf{Z}_j^{\oplus} \equiv \mathsf{Z}_i^{\oplus\setminus a}$. Then we get a unique $p_{i,j}$ such that $p_{i,j}$ is the first index $b$ with $x_i[b] \neq x_j[b]$. That means $\mathsf{X}_j[p_{ij}] = \mathsf{X}_i[c]$ or $\mathsf{X}_i[p_{ij}] = \mathsf{X}_j[c]$ for some $c$. So, the system of z-equations $\mathsf{Z}_j^{\oplus} = \mathsf{X}_i[a] \wedge \mathsf{Z}_i^{\oplus\setminus a} = \mathsf{X}_k[b] \oplus \mathsf{T}_j$ reduces to a system of one z-equation and also gives a equation in terms of hash collision. It gives

$$\mathsf{Pr} \leq \frac{q^2 \ell_{\max}^2 \epsilon}{2^n - \sigma}$$

4. BadY2.2 : $\mathsf{x}_i[a] = \mathsf{z}_j^{\oplus} \wedge \mathsf{Y}_i^{\oplus} = \mathsf{Y}_k^{\oplus}$ for $i, j, k \in [q]$ and $a \in [\ell_i]$.

Here we get $\mathsf{x}_i[a] = \mathsf{z}_j^{\oplus} \wedge \mathsf{Z}_i^{\oplus\setminus a} = \mathsf{y}_k^{\oplus} \oplus \mathsf{t}_j$ which changes according to the following subcases:

*Case when $k \notin I_{\mathsf{FC}}$:* Then the above system becomes $\mathsf{Z}_j^{\oplus} = \mathsf{X}_i[a] \wedge \mathsf{Z}_i^{\oplus\setminus a} = \mathsf{Z}_k^{\oplus} \oplus \mathsf{T}_j$. If $\mathsf{Z}_j^{\oplus} \equiv \mathsf{Z}_i^{\oplus\setminus a} \oplus \mathsf{Z}_k^{\oplus}$, then $\mathsf{X}_i[a] = \mathsf{t}_j$ giving

$$\mathsf{Pr} \leq \frac{\epsilon \ell_{\max} q^3}{2^n - \sigma}$$

Otherwise, the system has rank=2. Therefore,

$$\mathsf{Pr} \leq \frac{q^3 \ell_{\max}}{(2^n - \sigma)^2}$$

*Case when $k \in I_{\mathsf{FC}}$:* In this case we have $\mathsf{Z}_j^{\oplus} = \mathsf{X}_i[a] \wedge \mathsf{Z}_l^{\oplus} = \mathsf{X}_k[b] \wedge \mathsf{Z}_i^{\oplus\setminus a} \oplus \mathsf{Z}_k^{\oplus\setminus b} = \mathsf{T}_j \oplus \mathsf{T}_l$ for $b \in [\ell_k]$, $l \in [q]$ and the others as before. If $\mathsf{Z}_j^{\oplus}, \mathsf{Z}_l^{\oplus}, \mathsf{Z}_i^{\oplus\setminus a} \oplus \mathsf{Z}_k^{\oplus\setminus b}$ are linearly independent, then

$$\mathsf{Pr} \leq \frac{q^4 \ell_{\max}^2}{(2^n - \sigma)^3}$$

Remaining case is when $\mathsf{Z}_j^{\oplus}, \mathsf{Z}_l^{\oplus}, \mathsf{Z}_i^{\oplus\setminus a} \oplus \mathsf{Z}_k^{\oplus\setminus b}$ are linearly dependent. Let us look at this case.

Rank=1 is discarded since it implies $\mathsf{Z}_j^{\oplus} = \mathsf{Z}_l^{\oplus}$ (by lemma 2.3.2) which is an BadZ event. From lemma 2.3.2 we have that there are four possibilities for rank=2:

- $\mathsf{Z}_j^{\oplus} \equiv \mathsf{Z}_l^{\oplus}$
- $\mathsf{Z}_j^{\oplus} \equiv \mathsf{Z}_l^{\oplus} \oplus \mathsf{Z}_i^{\oplus\setminus a} \oplus \mathsf{Z}_k^{\oplus\setminus b}$
- $\mathsf{Z}_j^{\oplus} \equiv \mathsf{Z}_i^{\oplus\setminus a} \oplus \mathsf{Z}_k^{\oplus\setminus b}$
- $\mathsf{Z}_l^{\oplus} \equiv \mathsf{Z}_i^{\oplus\setminus a} \oplus \mathsf{Z}_k^{\oplus\setminus b}$

First possibility is discarded again for being an BadZ event. Third and fourth ones are similar and have same probability. The second possibility is different-looking but for being a $\neg$BadT event, it has the same probability as the third/fourth one.

Third/fourth possibility: $\mathsf{Z}_j^{\oplus} \equiv \mathsf{Z}_i^{\oplus\setminus a} \oplus \mathsf{Z}_k^{\oplus\setminus b} \Rightarrow \mathsf{X}_i[a] = \mathsf{T}_j \oplus \mathsf{T}_l$ (similarly, $\mathsf{Z}_l^{\oplus} = \mathsf{Z}_i^{\oplus\setminus a} \oplus \mathsf{Z}_k^{\oplus\setminus b} \Rightarrow \mathsf{X}_k[b] = \mathsf{T}_j \oplus \mathsf{T}_l$).

Second possibility $\mathsf{Z}_j^{\oplus} \equiv \mathsf{Z}_l^{\oplus} \oplus \mathsf{Z}_i^{\oplus\setminus a} \oplus \mathsf{Z}_k^{\oplus\setminus b} \Rightarrow \mathsf{X}_i[a] \oplus \mathsf{X}_k[b] = \mathsf{T}_j \oplus \mathsf{T}_l$.

Therefore,

$$\mathsf{Pr} \leq \frac{3q^4\ell_{\max}^2\epsilon}{(2^n-\sigma)^2}$$

Now we sum all the above probabilities for BadY and apply the relation that $\sigma \leq 2^{n-1}$. We get

$$\mathsf{Pr}[\mathtt{BadY}] \leq \frac{15q^3\ell_{\max}^2}{2^{2n}} + \frac{\epsilon q^3\ell_{\max}}{2^n-\sigma} + \frac{\epsilon q^2\ell_{\max}^2}{2^n-\sigma} + \frac{3\epsilon q^4\ell_{\max}^2}{(2^n-\sigma)^2} \tag{5.23}$$

Summing up the bounds for BadAux, BadZ and BadY, we conclude the proof of the lemma.

## 5.4 Key Results At a Glance

- Theorem 5.2.1 shows a relationship between $\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{xPMAC}}$ and $\mathbf{Adv}^{\mathrm{xcoll}}_{\mathcal{X}}$, while the encoding $\mathcal{X}$ is blockwise universal and independent of the block cipher keys. Similar result is shown for the single-key version in theorem 5.2.2, while a blockwise regular and universal encoding is used. In both of these results, it is assumed that the maximum query-length is upper bounded by $2^{n/4}$.
- From these relationships we obtained security bounds of $O(q^2/2^n)$ for single-key versions of PMAC+ and LightMAC in section 5.2.1. For PMAC+, the result is subject to an additional assumption on the primitive element used for masking of the message blocks.

# Part III

# CBC-type MACs

# Chapter 6

# OMAC, XCBC and TMAC

In this chapter, we will prove security results for three important single keyed CBC-type constructions: OMAC[1] [47], XCBC [15] and TMAC [60]. We will follow the same two-stage sampling strategy, dubbed as the *reset-sampling* technique, as we did in chapter 4.

## 6.1 The CBC-MAC Family

Throughout, $n$ denotes the *block size*, $\mathbb{B} := \{0,1\}^n$, and any $x \in \mathbb{B}$ is referred as a *block*. For any non-empty $m \in \{0,1\}^*$, $(m[1], \ldots, m[\ell_m]) \xleftarrow{n} m$ denotes the *block parsing* of $m$, where $|m[i]| = n$ for all $1 \leq i \leq \ell_m - 1$ and $1 \leq |m[\ell_m]| \leq n$. In addition, we associate a boolean flag $\delta_m$ to each $m \in \{0,1\}^*$, which is defined as

$$\delta_m := \begin{cases} -1 & \text{if } |m| = n\ell_m, \\ 0 & \text{otherwise.} \end{cases}$$

For any $m \in \{0,1\}^{\leq n}$, we define

$$\overline{m} := \begin{cases} m \| 10^{n-|m|-1} & \text{if } |m| < n, \\ m & \text{otherwise.} \end{cases}$$

[1] This is same as CMAC [81]

CBC FUNCTION: The CBC function, based on a permutation $\pi \in \mathsf{Perm}(n)$,[2] takes as input a non-empty message $m \in \mathbb{B}^*$ and computes the output $\mathsf{CBC}_\pi(m) := y_m^\pi[\ell_m]$ inductively as described below:

$y_m^\pi[0] = 0^n$ and for $1 \le i \le \ell_m$, we have

$$\begin{aligned} x_m^\pi[i] &:= y_m^\pi[i-1] \oplus m[i], \\ y_m^\pi[i] &:= \pi(x_m^\pi[i]), \end{aligned} \tag{6.1}$$

where $(m[1], \ldots, m[\ell_m]) \xleftarrow{n} m$. For empty message, we define the CBC output as the constant $0^n$. Figure 6.1.1 illustrates the evaluation of CBC function over a 4-block message $m$.

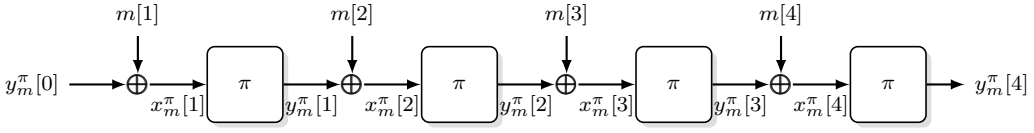


**Figure 6.1.1:** Evaluation of CBC function over a 4-block message $m$..

Given the definition of $\mathsf{CBC}_\pi$, one can easily define all the variants of CBC-MAC. Here we define XCBC, TMAC and OMAC, the three constructions that we study in this paper.

XCBC: The XCBC algorithm is a three-key construction, based on a permutation $\pi \in \mathsf{Perm}(n)$ and keys $(L_{-1}, L_0) \in \mathbb{B}^2$, that takes as input a non-empty message $m \in \{0,1\}^*$, and computes the output

$$\mathsf{XCBC}_{\pi, L_{-1}, L_0}(m) := t = \pi\left(\mathsf{CBC}_\pi\left(m^*\right) \oplus \overline{m[\ell_m]} \oplus L_{\delta_m}\right), \tag{6.2}$$

where $(m[1], \ldots, m[\ell_m]) \xleftarrow{n} m$, and $m^* := m[1] \| \cdots \| m[\ell_m - 1]$.

TMAC: The TMAC algorithm is a two-key construction, based on a permutation $\pi \in \mathsf{Perm}(n)$ and key $l \in \mathbb{B}$, that takes as input a non-empty message $m \in \{0,1\}^*$, and computes the output

$$\mathsf{TMAC}_{\pi, l}(m) := t = \pi\left(\mathsf{CBC}_\pi\left(m^*\right) \oplus \overline{m[\ell_m]} \oplus \mu_{\delta_m} \odot l\right), \tag{6.3}$$

where $(m[1], \ldots, m[\ell_m]) \xleftarrow{n} m$, $m^* := m[1] \| \cdots \| m[\ell_m - 1]$, $\mu_{-1}$ and $\mu_0$ are constants chosen from $\mathbb{F}_{2^n}$ (viewing $\mathbb{B}$ as $\mathbb{F}_{2^n}$), such that $\mu_{-1}, \mu_0, 1 \oplus \mu_{-1}, 1 \oplus \mu_0$ are all distinct and not equal to either 0 or 1, and $\odot$ denotes the field multiplication operation over

[2] Instantiated with a block cipher in practical applications.

$\mathbb{F}_{2^n}$ with respect to a fixed primitive polynomial. For the sake of uniformity, we define $L_{\delta_m} := \mu_{\delta_m} \odot l$ in context of TMAC.

OMAC: The OMAC algorithm is a single-keyed construction, based on a permutation $\pi \in \mathsf{Perm}(n)$, that takes as input a non-empty message $m \in \{0,1\}^*$, and computes the output

$$\mathsf{OMAC}_\pi(m) := t = \pi\left(\mathsf{CBC}_\pi\left(m^*\right) \oplus \overline{m[\ell_m]} \oplus \mu_{\delta_m} \odot \pi(0^n)\right), \tag{6.4}$$

where $(m[1], \ldots, m[\ell_m]) \xleftarrow{n} m$, $m^* := m[1] \| \cdots \| m[\ell_m - 1]$, $\mu_{-1}$ and $\mu_0$ are constants chosen analogously as in the case of TMAC. For the sake of uniformity, we define $L_{\delta_m} := \mu_{\delta_m} \odot \pi(0^n)$ in context of OMAC.

### 6.1.1 Tight Security Bounds for OMAC, XCBC and TMAC

The main technical result of this paper, given in Theorem 6.1.1, is a tight security bound for OMAC for a wide range of message lengths. The proof of this theorem is postponed to section 6.2. In addition, we also provide similar result for XCBC and TMAC in Theorem 6.1.2. We skip the proof since it is almost identical to the one for Theorem 6.1.1, and has slightly less relevance given that a more efficient and standardized algorithm OMAC already achieves similar security. In what follows we define

$$\epsilon'(q,\ell) := \frac{16q^2 + q\ell^2}{2^n} + \frac{8q^2\ell^4 + 32q^3\ell^2 + 2q^2\ell^3}{2^{2n}} + \frac{3q^3\ell^5 + 143q^3\ell^6 + 11q^4\ell^3}{2^{3n}} + \frac{17q^4\ell^6 + 5462q^4\ell^8}{2^{4n}}.$$

**Theorem 6.1.1** (OMAC bound). *Let $q, \ell, \sigma, T > 0$. For $q + \sigma \leq 2^{n-1}$, the PRF insecurity of OMAC, based on block cipher $\mathsf{E}_\mathsf{K}$, against $\mathbb{A}(q,T)$ is given by*

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathit{OMAC}_{\mathsf{E}_\mathsf{K}}}(q,\ell,\sigma,T) \leq \mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(q+\sigma, T') + \frac{4\sigma}{2^n} + \epsilon'(q,\ell), \tag{6.5}$$

*where $q$ denotes the number of queries, $\ell$ denotes an upper bound on the number of blocks per query, $\sigma$ denotes the total number of blocks present in all $q$ queries, $T' = T + \sigma O(T_\mathsf{E})$ and $T_\mathsf{E}$ denotes the runtime of* E.

**Theorem 6.1.2** (XCBC-TMAC bound). *Let $q, \ell, \sigma, T > 0$. For $q+\sigma \leq 2^{n-1}$, the PRF insecurity of XCBC and TMAC, based on block cipher $\mathsf{E}_\mathsf{K}$ and respective masking keys $(\mathsf{L}, L_{-1}, L_0)$, against $\mathbb{A}(q,T)$ is given by*

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathit{XCBC}_{\mathsf{E}_\mathsf{K},\mathsf{L}_{-1},\mathsf{L}_0}}(q,\ell,\sigma,T) \leq \mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(q+\sigma, T') + \epsilon'(q,\ell) \tag{6.6}$$

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathit{TMAC}_{\mathsf{E}_\mathsf{K},\mathsf{L}}}(q,\ell,\sigma,T) \leq \mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(q+\sigma, T') + \epsilon'(q,\ell) \tag{6.7}$$

*where $q$ denotes the number of queries, $\ell$ denotes an upper bound on the number of blocks per query, $\sigma$ denotes the total number of blocks present in all $q$ queries, $T' = T + \sigma O(T_{\mathsf{E}})$ and $T_{\mathsf{E}}$ denotes the runtime of* $\mathsf{E}$.

Proof of this theorem is almost same as that of Theorem 6.1.1. The bad event on a collision on zero block input is redundant and hence dropped here. Rest of the proof remains the same and so we skip the details.

*Remark* 6.1.3. Note that the actual advantage cannot exceed 1. Let us denote $\frac{q^2}{2^n} = \alpha$ and $\frac{q\ell^2}{2^n} = \beta$. Looking at $\epsilon(q,\ell)$ (where $\epsilon(q,\ell) = \epsilon'(q,\ell) + \frac{4\sigma}{2^n}$ in case of OMAC and $\epsilon(q,\ell) = \epsilon'(q,\ell)$ in case of XCBC,TMAC), we see that any term in the expression is upper bounded by $c \cdot \alpha^s \beta^t$ for some constant $c$ and $s, t \geq 0$ such that at least one of $s$ and $t$ is at least 1. As we can assume both $\alpha, \beta$ to be less than 1, each $\alpha^s\beta^t$ will be less than or equal to $\alpha$ or $\beta$. Thus, the above PRF-advantage expressions for $\mathsf{MAC} \in \{\mathsf{OMAC}, \mathsf{XCBC}, \mathsf{TMAC}\}$ can be written as

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{MAC}}(q,\ell,\sigma) = O\left(\frac{q^2}{2^n}\right) + O\left(\frac{q\ell^2}{2^n}\right).$$

A NOTE ON THE PROOF APPROACH: In the analysis of OMAC, XCBC and TMAC, we have to handle the case that the final input collides with some intermediate input, the so-called *full collision* event. In earlier works the probability of this event is shown to be $q^2\ell/2^n$ (as there are less than $q\ell$ many intermediate inputs and $q$ final inputs and any such collision happens with roughly $1/2^n$ probability). So, in a way they avoid handling this tricky event by disallowing it all together. In this work, we allow full collisions as long as the next intermediate input is not colliding with some other input (intermediate or final). Looking ahead momentarily, this is captured in `BadW3`. We can do this via the application of reset-sampling resulting in a more amenable $(q^2/2^n + q\ell^2/2^n)$ bound.

## 6.2 Proof of Theorem 6.1.1

*Input and Output Tuples:* In the context of CBC evaluation within OMAC, we refer to $x^\pi_m := (x^\pi_m[1], \ldots, x^\pi_m[\ell_m - 1])$ and $y^\pi_m := (y^\pi_m[0], \ldots, y^\pi_m[\ell_m - 1])$ as the *intermediate input* and *output* tuples, respectively, associated to $\pi$ and $m$. We define the final input variable as:

$$x^\pi_m[\ell_m] := y^\pi_m[\ell_m - 1] \oplus \overline{m[\ell_m]} \oplus \mu_{\delta_m} \odot \pi(0^n)$$

Clearly, the input and output tuples (including the final input) are well defined for OMAC. Analogous definitions are possible (and useful in proof) for XCBC and TMAC

as well. It is worth noting that the intermediate input tuple $x_m^\pi$ is uniquely determined by the intermediate output tuple $y_m^\pi$ and the message $m$, and it is independent of the permutation $\pi$. Going forward, we drop $\pi$ from the notations, whenever it is clear from the context.

As for the first step in the proof, we employ the standard hybrid argument to get

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{OMAC}_{\mathsf{E_K}}}(q,\ell,\sigma,T) \leq \mathbf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(q+\sigma,T') + \mathbf{Adv}^{\mathsf{prf}}_{\mathsf{OMAC}_{\boldsymbol{\pi}}}(q,\ell,\sigma,\infty). \tag{6.8}$$

Now, it is sufficient to bound $\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{OMAC}_{\boldsymbol{\pi}}}(q,\ell,\sigma,\infty)$, where the corresponding distinguisher $\mathscr{A}$ is computationally unbounded and deterministic. We employ the H-coefficient technique (see section 2.2.5) to bound this term. In addition, we also employ the recently introduced *reset-sampling* method [23] by Chattopadhyay et al. The remaining steps of the proof are given in the remainder of this section.

### 6.2.1 Oracle Description and Corresponding Transcripts

**Real Oracle:** The real oracle corresponds to $\mathsf{OMAC}_{\boldsymbol{\pi}}$. It responds faithfully to all the queries made by $\mathscr{A}$. Once the query-response phase is over, it releases all the intermediate inputs and outputs, as well as the masking keys $\mathsf{L}_{-1}$ and $\mathsf{L}_0$ to $\mathscr{A}$. Recall that $\mathsf{L} = \boldsymbol{\pi}(0^n)$.

In addition, the real oracle releases three binary variables, namely, $\mathsf{FlagT}$, $\mathsf{FlagW}$ and $\mathsf{FlagX}$, all of which are degenerately set to $0$. These flags are more of a technical requirement, and their utility will become apparent from the description of ideal oracle. For now, it is sufficient to note that these flags are degenerate in the real world.

Formally, we have

$$\Theta_1 := (\widetilde{\mathsf{M}}, \widetilde{\mathsf{T}}, \widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*, \widetilde{\mathsf{Y}}, \mathsf{L}_{-1}, \mathsf{L}_0, \mathsf{FlagT}, \mathsf{FlagW}, \mathsf{FlagX}),$$

where

- $\widetilde{\mathsf{M}} = (\mathsf{M}_1, \ldots, \mathsf{M}_q)$ denotes the $q$-tuple of queries made by $\mathscr{A}$, where $\mathsf{M}_i \in \{0,1\}^*$ for all $i \in [q]$. In addition, for all $i \in [q]$, let $\ell_i := \left\lceil \frac{|\mathsf{M}_i|}{n} \right\rceil$.
- $\widetilde{\mathsf{T}} = (\mathsf{T}_1, \ldots, \mathsf{T}_q)$ denotes the $q$-tuple of final outputs received by $\mathscr{A}$, where $\mathsf{T}_i \in \mathbb{B}$.
- $\widetilde{\mathsf{X}} = (\mathsf{X}_1, \ldots, \mathsf{X}_q)$, where $\mathsf{X}_i$ denotes the intermediate input tuple for the $i$-th query.
- $\widetilde{\mathsf{X}}^* = (\mathsf{X}_1[\ell_1], \ldots, \mathsf{X}_q[\ell_q])$, where $\mathsf{X}_i[\ell_i]$ denotes the final input for the $i$-th query.

- $\widetilde{\mathsf{Y}} = (\mathsf{Y}_1, \ldots, \mathsf{Y}_q)$, where $\mathsf{Y}_i$ denotes the intermediate output tuple for the $i$-th query.
- $\mathsf{L}_{-1}$ and $\mathsf{L}_0$ denote the two masking keys. Note that $\mathsf{L}_{-1}$ and $\mathsf{L}_0$ are easily derivable from $\mathsf{L}$. So we could have simply released $\mathsf{L}$. The added redundancy helps later to establish the analogous connection between this proof and the proof for XCBC.
- $\mathsf{FlagT} = \mathsf{FlagW} = \mathsf{FlagX} = 0$.

From the definition of OMAC, we know that $\boldsymbol{\pi}(\mathsf{X}_i[a]) = \mathsf{Y}_i[a]$ for all $(i, a) \in [q] \times [\ell_i]$. So, *in the real world we always have* $(0^n, \widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*) \leftrightsquigarrow (\mathsf{L}, \widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$*, i.e.,* $(0^n, \widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*)$ *is permutation compatible with* $(\mathsf{L}, \widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$. We keep this observation in our mind when we simulate the ideal oracle.

**Ideal Oracle:** We reuse the notations from real oracle description to represent the variables in the ideal oracle transcript $\Theta_0$, i.e.,

$$\Theta_0 := (\widetilde{\mathsf{M}}, \widetilde{\mathsf{T}}, \widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*, \widetilde{\mathsf{Y}}, \mathsf{L}_{-1}, \mathsf{L}_0, \mathsf{FlagT}, \mathsf{FlagW}, \mathsf{FlagX}).$$

This should not cause any confusion, as we never consider the random variables $\Theta_1$ and $\Theta_0$ jointly, whence the probability distributions of the constituent variables will always be clear from the context.

The ideal oracle transcript is described in three phases, each contingent on some predicates defined over the previous stages. Specifically, the ideal oracle first initializes $\mathsf{FlagT} = \mathsf{FlagW} = \mathsf{FlagX} = 0$, and then follows the sampling mechanism given below:

PHASE I (QUERY-RESPONSE PHASE): In the query-response phase, the ideal oracle faithfully simulates $\boldsymbol{\rho} \leftarrow\!\$\ \mathsf{Func}(\{0,1\}^*, \mathbb{B})$. Formally, for $i \in [q]$, at the $i$-th query $\mathsf{M}_i \in \{0,1\}^*$, the ideal oracle outputs $\mathsf{T}_i \leftarrow\!\$\ \mathbb{B}$. The partial transcript generated at the end of the query-response phase is given by $(\widetilde{\mathsf{M}}, \widetilde{\mathsf{T}})$, where

- $\widetilde{\mathsf{M}} = (\mathsf{M}_1, \ldots, \mathsf{M}_q)$ and $\widetilde{\mathsf{T}} = (\mathsf{T}_1, \ldots, \mathsf{T}_q)$.

Now, we define a predicate on $\widetilde{\mathsf{T}}$:

$$\texttt{BadT}: \ \exists i \neq j \in [q], \text{ such that } \mathsf{T}_i = \mathsf{T}_j.$$

If BadT is true, then $\mathsf{FlagT}$ is set to 1, and $\widetilde{\mathsf{X}}$, $\widetilde{\mathsf{X}}^*$, and $\widetilde{\mathsf{Y}}$ are defined degenerately: $\mathsf{X}_i[a] = \mathsf{Y}_i[b] = 0^n$ for all $i \in [q]$, $a \in [\ell_i]$, $b \in (\ell_i - 1]$. Otherwise, the ideal oracle proceeds to the next phase.

PHASE II (OFFLINE INITIAL SAMPLING PHASE): Onward, we must have $\mathsf{T}_i \neq \mathsf{T}_j$ whenever $i \neq j$, and $\mathsf{FlagT} = 0$, since this phase is only executed when BadT is false. In the offline phase, the ideal oracle's initial goal is to sample the input and output tuples in such a way that the intermediate input and output tuples are permutation compatible. For now we use notations W and Z, respectively, instead of X and Y, to denote the input and output tuples. This is done to avoid any confusions in the next step where we may have to reset some of these variables. To make it explicit, W and Z respectively denote the input and output tuples before resetting, and X and Y denote the input and output tuples after resetting.

Let P be a key-value table representing a partial permutation of $\mathbb{B}$, which is initialized to empty, i.e., the corresponding permutation is undefined on all points. We write P.domain and P.range to denote the set of all keys and values utilized till this point, respectively. The ideal oracle uses this partial permutation P to maintain permutation compatibility between intermediate input and output tuples, in the following manner:

Initial sampling

$\mathsf{L} \xleftarrow{\$} \mathbb{B} \setminus \widetilde{\mathsf{T}}$

$\mathsf{L}_{-1} \leftarrow \mu_{-1} \odot \mathsf{L}$

$\mathsf{L}_0 \leftarrow \mu_0 \odot \mathsf{L}$

$\mathsf{P}(0^n) \leftarrow \mathsf{L}$

**for** $i = 1$ **to** $q$ **do**

  $\mathsf{Z}_i[0] \leftarrow 0^n$

  **for** $a = 1$ **to** $\ell_i - 1$ **do**

    $\mathsf{W}_i[a] \leftarrow \mathsf{Z}_i[a-1] \oplus \mathsf{M}_i[a]$

    **if** $\mathsf{W}_i[a] \in \mathsf{P.domain}$

      $\mathsf{Z}_i[a] \leftarrow \mathsf{P}(\mathsf{W}_i[a])$

    **else**

      $\mathsf{Z}_i[a] \xleftarrow{\$} \mathbb{B} \setminus \left(\widetilde{\mathsf{T}} \cup \mathsf{P.range}\right)$

      $\mathsf{P}(\mathsf{W}_i[a]) \leftarrow \mathsf{Z}_i[a]$

  $\mathsf{W}_i[\ell_i] \leftarrow \mathsf{Z}_i[\ell_i - 1] \oplus \overline{\mathsf{M}}_i[\ell_i] \oplus \mathsf{L}_{\delta_{\mathsf{M}_i}}$

At this stage we have $\mathsf{Z}_i[a] = \mathsf{Z}_j[b]$ if and only if $\mathsf{W}_i[a] = \mathsf{W}_j[b]$ for all $(i,a) \in [q] \times [\ell_i - 1]$ and $(j,b) \in [q] \times [\ell_j - 1]$. In other words, $(0^n, \widetilde{\mathsf{W}}) \leftrightsquigarrow (\mathsf{L}, \widetilde{\mathsf{Z}})$. But it is obvious to see that the same might not hold between $(0^n, \widetilde{\mathsf{W}}, \widetilde{\mathsf{W}}^*)$ and $(\mathsf{L}, \widetilde{\mathsf{Z}}, \widetilde{\mathsf{T}})$. In the next stage our goal will be to reset some of the Z variables in such a way that the resulting input tuple is compatible with the resulting output tuple. However, in order to reset, we have to identify and avoid certain contentious input-output tuples.

IDENTIFYING CONTENTIOUS INPUT-OUTPTUT TUPLES: We define several predicates on $(\widetilde{\mathsf{W}}, \widetilde{\mathsf{W}}^*)$, each of which represents some undesirable property of the sampled input and output tuples.

First, observe that $\mathsf{L}$ is chosen outside the set $\widetilde{\mathsf{T}}$. This leads to the first predicate:

$$\texttt{BadW1}: \ \exists (i,a) \in [q] \times [\ell_i], \text{ such that } (\mathsf{W}_i[a] = 0^n) \text{ and } (\ell_i > 1 \implies a > 1).$$

since, if BadW1 is true, then $(0^n, \widetilde{\mathsf{W}}^*)$ is not compatible with $(\mathsf{L}, \widetilde{\mathsf{T}})$. In fact, $\neg$BadW1 implies that none of the inputs, except the first input which is fully in adversary's control, can possibly be $0^n$. This stronger condition will simplify the analysis greatly. The second predicate simply states that the final input tuple is not permutation compatible with the tag tuple, i.e., we have

$$\texttt{BadW2}: \ \exists i \neq j \in [q], \text{ such that } \mathsf{W}_i[\ell_i] = \mathsf{W}_j[\ell_j].$$

At this point, assuming $\neg(\texttt{BadW1} \vee \texttt{BadW2})$ holds true, the only way we can have permutation incompatibility is if $\mathsf{W}_i[a] = \mathsf{W}_j[\ell_j]$, for some $i, j \in [q]$ and $a \in [\ell_i - 1]$. A simple solution will be to reset $\mathsf{Z}_i[a]$ to $\mathsf{T}_j$, for all such $(i, a, j)$. In order to do this, we need that the following predicates must be false:

$$\texttt{BadW3}: \ \exists i,j,k \in [q], a \in [\ell_i - 1], b \in [\ell_k], \text{ such that}$$

$$(\mathsf{W}_i[a] = \mathsf{W}_j[\ell_j]) \wedge (\mathsf{W}_i[a+1] = \mathsf{W}_k[b]) \wedge \mathsf{Prefix}(\mathsf{M}_i, \mathsf{M}_j) < \max\{a+1, b\}.$$

$$\texttt{BadW4}: \ \exists i,j,k \in [q], a \neq b \in [\ell_i - 1], \text{ such that}$$

$$(\mathsf{W}_i[a] = \mathsf{W}_j[\ell_j]) \wedge (\mathsf{W}_i[b] = \mathsf{W}_k[\ell_k]).$$

$$\texttt{BadW5}: \ \exists i,j,k \in [q], a \in [\ell_i - 1], b \in [\ell_j - 1], \text{ such that}$$

$$(\mathsf{W}_i[a] = \mathsf{W}_j[\ell_j]) \wedge (\mathsf{W}_j[b] = \mathsf{W}_k[\ell_k]).$$

If BadW3 is true, then once $\mathsf{Z}_i[a]$ is reset, we lose the permutation compatibility since, the reset next input, i.e., $\mathsf{X}_i[a+1] = \mathsf{W}_i[a+1] \oplus \mathsf{Z}_i[a] \oplus \mathsf{T}_j = \mathsf{M}_i[a+1] \oplus \mathsf{T}_j \neq \mathsf{W}_k[b]$ with high probability, whereas $\mathsf{Z}_i[a+1] = \mathsf{Z}_k[b]$ with certainty. BadW4 simply represents the scenario where we may have to apply the initial resetting to two indices in a single message. Looking ahead momentarily, this may lead to contradictory *induced* resettings. Avoiding this predicate makes the resetting operation much more manageable. Similarly, avoiding BadW5, is just proactive prevention of contradictory resetting

at $\mathsf{Z}_i[a]$, since if BadW5 occurs, then we may have a case where $\mathsf{X}_j[\ell_j]$ is reset due to induced resetting, leading to the case, $\mathsf{X}_i[a] \neq \mathsf{X}_j[\ell_j]$ and $\mathsf{Y}_i[a] = \mathsf{T}_j$, where recall that $\mathsf{Y}_i[a]$ is the resetting value of $\mathsf{Z}_i[a]$. We write

$$\texttt{BadW} := \texttt{BadW1} \vee \texttt{BadW2} \vee \texttt{BadW3} \vee \texttt{BadW4} \vee \texttt{BadW5}.$$

If BadW is true, then FlagW is set to 1, and $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*, \widetilde{\mathsf{Y}})$ is again defined degenerately, as in the case of BadT. Otherwise, the ideal oracle proceeds to the next and the final phase, i.e., the resetting phase.

PHASE III.A INITIAL RESETTING PHASE: At this stage we must have $\neg(\texttt{BadT} \vee \texttt{BadW})$, i.e., $\mathsf{FlagW} = \mathsf{FlagT} = 0$. We describe the resetting phase in two sub-stages. First, we identify the indices affected by the initial resetting operation.

**Definition 6.2.1** (full collision index)**.** Any $(i, a, j) \in [q] \times [\ell_i - 1] \times [q]$ is called a full collision index (FCI) if $\mathsf{W}_i[a] = \mathsf{W}_j[\ell_j]$. Additionally, let

$$\mathsf{FCI} := \{(i, a, j) : i, j \in [q], a \in [\ell_i - 1], \text{ such that } (i, a, j) \text{ is an } FCI\}$$
$$\widetilde{\mathsf{FCI}} := \{(i, a) \in [q] \times [\ell_i - 1] : \exists j \in [q], \text{ such that } (i, a, j) \text{ is an } FCI\}$$

The first sub-stage, executes a resetting for full collision indices in the following manner:

1. For all $(i, a, j) \in \mathsf{FCI}$, define $\mathsf{Y}_i[a] := \mathsf{T}_j$;
2. For all $(i, a, j) \in \mathsf{FCI}$, define

$$\mathsf{X}_i[a+1] := \mathsf{W}_i[a+1] \oplus \mathsf{Z}_i[a] \oplus \mathsf{Y}_i[a] = \overline{\mathsf{M}}_i[a+1] \oplus \mathsf{T}_j \oplus 1_{a=\ell_i-1} \odot \mathsf{L}_{\delta_{\mathsf{M}_i}},$$

where $1_{a=\ell_i-1}$ is an indicator variable that evaluates to 1 when $a = \ell_i - 1$, and 0 otherwise.

Once the initial resetting is executed, it may result in new permutation incompatibilities. This necessitates further resettings, referred as *induced resettings*, which require that the following predicates are false:

$$\texttt{BadX1}: \ \exists (i, a, j) \in \mathsf{FCI}, k \in [q], b \in [\ell_k] \setminus \{1\}, \text{ such that}$$
$$(\mathsf{X}_i[a+1] = \mathsf{W}_k[b]) \vee (\mathsf{X}_i[a+1] = 0^n).$$

$\texttt{BadX2}$ : $\exists (i,a,j) \in \mathsf{FCI}, k \in [q],$ such that

$$(\mathsf{X}_i[a+1] = \mathsf{M}_k[1]) \wedge (\mathsf{M}_i[a+2,\ldots,\ell_i] = \mathsf{M}_k[2,\ldots,\ell_k]).$$

$\texttt{BadX3}$ : $\exists (i,a,j),(k,b,l) \in \mathsf{FCI},$ such that $(\mathsf{X}_i[a+1] = \mathsf{M}_k[1])$.

$\texttt{BadX4}$ : $\exists (i,a,k),(j,b,l) \in \mathsf{FCI},$ such that

$$(\mathsf{X}_i[a+1] = \mathsf{X}_j[b+1]) \wedge (\mathsf{Prefix}(\mathsf{M}_i,\mathsf{M}_j) < \max\{a+1,b+1\}).$$

Here, the variable highlighted in red denotes the update after initial resetting. Let's review these predicates in slightly more details. First, $\texttt{BadX1}$, represents the situation where after resetting the next input (highlighted text) collides with some intermediate input or $0^n$. This would necessitate induced resetting at $\mathsf{Z}_i[a+1]$. In other words, if $\texttt{BadX1}$ is false then no induced resettings occur, unless the next input collides with some first block input. This case is handled in the next two predicates. $\texttt{BadX2}$ represents the situation when the next input collides with a first block and the subsequent message blocks are all same. This would induce a chain of resetting going all the way to the final input. As $\texttt{BadT}$ is false, this would immediately result in a permutation incompatibility since tags are distinct. If $\texttt{BadX2}$ is false, then the chain of induced resetting must end at some point. $\texttt{BadX3}$ is used to avoid circular or contradictory resettings. It is analogous to $\texttt{BadW5}$ defined earlier. If it is false, then we know that the $k$-th message is free from resetting, so the induced resetting will be manageable. Finally, $\texttt{BadX4}$ represents the situation when two newly reset variables collide. We write

$$\texttt{BadX1234} := \texttt{BadX1} \vee \texttt{BadX2} \vee \texttt{BadX3} \vee \texttt{BadX4}$$

If $\texttt{BadX1234}$ is true, then $\mathsf{FlagX}$ is set to 1, and $(\widetilde{\mathsf{X}},\widetilde{\mathsf{X}}^*,\widetilde{\mathsf{Y}})$ is again defined degenerately, as in the cases of $\texttt{BadT}$ and $\texttt{BadW}$. Otherwise, the ideal oracle proceeds to the second and the final sub-stage of resetting.

PHASE III.B INDUCED RESETTING PHASE: Here, the goal is to execute the induced resettings necessitated by the initial resetting operation.

First, we define the *index of induced resetting* for each $(i,a) \in \widetilde{\mathsf{FCI}}$, as the smallest index $j$ such that $\mathsf{X}_i[a+1] = \mathsf{M}_j[1]$ and

$$\mathsf{Prefix}(\mathsf{M}_i[a{+}2,\ldots,\ell_i],\mathsf{M}_j[2,\ldots,\ell_j]) = \max\{\mathsf{Prefix}(\mathsf{M}_i[a{+}2,\ldots,\ell_i],\mathsf{M}_{j'}[2,\ldots,\ell_{j'}]) : j' \in [q]\},$$

i.e., $\mathsf{Prefix}(\mathsf{M}_i[a+2,\ldots,\ell_i],\mathsf{M}_j[2,\ldots,\ell_j])$ maximizes.

**Definition 6.2.2** (induced collision sequence)**.** A sequence of tuples $((i, a{+}1, j, 1), \ldots, (i, a{+}p+1, j, p+1))$ is called an induced collision sequence (ICS), if $(i, a) \in \widetilde{\mathsf{FCI}}$, and $j$ is the index of induced resetting for $(i, a)$, where $p := \mathsf{Prefix}(\mathsf{M}_i[a+2, \ldots, \ell_i], \mathsf{M}_j[2, \ldots, \ell_j])$. The individual elements of an ICS are referred as induced collision index (ICI). Additionally, we let

$$\mathsf{ICI} := \{(i, a, j, b) : i, j \in [q], a \in [\ell_i - 1], b \in [\ell_j - 1], \text{ and } (i, a, j, b) \text{ is an ICI.}\}$$
$$\widetilde{\mathsf{ICI}} := \{(i, a) \in [q] \times [\ell_i - 1] : \exists (j, b) \in [q] \times [\ell_j - 1], \text{ and } (i, a, j, b) \text{ is an ICI.}\}$$

Now, as anticipated, in the second sub-stage of resetting, we reset the induced collision indices in the following manner:

1. For all $(i, a, j, b) \in \mathsf{ICI}$, define $\mathsf{Y}_i[a] := \mathsf{Z}_j[b]$;
2. For all $(i, a, j, b) \in \mathsf{ICI}$, define

$$\mathsf{X}_i[a+1] := \mathsf{W}_i[a+1] \oplus \mathsf{Z}_i[a] \oplus \mathsf{Y}_i[a] = \overline{\mathsf{M}}_i[a+1] \oplus \mathsf{Z}_j[b] \oplus 1_{a=\ell_i-1} \odot \mathsf{L}_{\delta_{\mathsf{M}_i}},$$

   where $1_{a=\ell_i-1}$ is an indicator variable that evaluates to 1 when $a = \ell_i - 1$, and 0 otherwise.

Given $\neg$`BadX1234`, we know that the induced resetting must stop at some point before the final input. Now, it might happen that once the first chain of induced resetting stops, the next input again collides which may result in nested resetting or permutation incompatibility. The predicates `BadX5`, `BadX6`, and `BadX7` below represent these scenarios.

- `BadX5` : $\exists (i, a, k, b) \in \mathsf{ICI}, l \in [q], b \in [\ell_l - 1],$ such that

$$(\mathsf{X}_i[a+2+p] = \mathsf{W}_l[b]) \vee (\mathsf{X}_i[a+2+p] = 0^n),$$

  where $p := \mathsf{Prefix}(\mathsf{M}_i[a+2, \ldots, \ell_i], \mathsf{M}_k[2, \ldots, \ell_k])$.

- `BadX6` : $\exists (i, a) \in \widetilde{\mathsf{FCI}}, (j, b, k, c) \in \mathsf{ICI}$, such that $(\mathsf{X}_i[a+1] = \mathsf{X}_j[b+2+p])$, where $p := \mathsf{Prefix}(\mathsf{M}_j[b+2, \ldots, \ell_j], \mathsf{M}_k[2, \ldots, \ell_k])$.

- `BadX7` : $\exists (i, a, k, c), (j, b, l, d) \in \mathsf{ICI},$ such that

$$(\mathsf{X}_i[a+2+p] = \mathsf{X}_j[b+2+p']) \wedge (\mathsf{Prefix}(\mathsf{M}_i, \mathsf{M}_j) < \max\{a+2+p, b+2+p'\}),$$

where $p := \mathsf{Prefix}(\mathsf{M}_i[a{+}2, \ldots, \ell_i], \mathsf{M}_k[2, \ldots, \ell_k])$, and $p' := \mathsf{Prefix}(\mathsf{M}_j[b{+}2, \ldots, \ell_j], \mathsf{M}_l[2, \ldots, \ell_l])$.

Here, the variables highlighted in red and blue denote the update after initial resetting and induced resetting, respectively. These predicates are fairly self-explanatory. First BadX5 represents the situation that the immediate input after induced resetting collides with some intermediate input or $0^n$. This may cause permutation incompatibility and would lead to nested induced resetting at $\mathsf{Z}_i[a+2+p]$. BadX6 handles a similar collision with a full collision resetted variable, and BadX7 handles the only remaining case where the immediate inputs after two different induced resetting collides. Note that, $\neg(\texttt{BadX5} \vee \texttt{BadX6} \vee \texttt{BadX7})$ would imply that for each message resetting stops at some point before the final input, and the next input is fresh.[3] We write

$$\texttt{BadX} := \texttt{BadX1} \vee \texttt{BadX2} \vee \texttt{BadX3} \vee \texttt{BadX4} \vee \texttt{BadX5} \vee \texttt{BadX6} \vee \texttt{BadX7}.$$

If BadX is true, then FlagX is set to 1, and $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*, \widetilde{\mathsf{Y}})$ is again defined degenerately, as in the case of BadT and BadW. Otherwise, for any remaining index $(i, a) \in [q] \times (\ell_i - 1] \setminus (\widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}})$, the ideal oracle resets as follows:

1. define $\mathsf{Y}_i[a] := \mathsf{Z}_i[a]$;
2. define $\mathsf{X}_i[a+1] := \mathsf{W}_i[a+1]$.

At this point, the ideal oracle transcript is completely defined. Intuitively, if the ideal oracle is not sampling $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*, \widetilde{\mathsf{Y}})$ degenerately at any stage, then we must have $(0^n, \widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*) \leftrightsquigarrow (\mathsf{L}, \widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$. The following proposition justifies this intuition.

**Proposition 6.2.3.** *For* $\neg(\texttt{BadT} \vee \texttt{BadW} \vee \texttt{BadX})$, *we must have* $(0^n, \widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*) \leftrightsquigarrow (\mathsf{L}, \widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$.

*Proof.* Let $\neg(\texttt{BadT} \vee \texttt{BadW} \vee \texttt{BadX})$ hold. Recall that $(0^n, \widetilde{\mathsf{W}}, \widetilde{\mathsf{W}}^*)$ may not be permutation compatible with $(\mathsf{L}, \widetilde{\mathsf{Z}}, \widetilde{\mathsf{T}})$. For any $(i, a) \in \widetilde{\mathsf{FCI}}$, there exists $i' \in [\ell_{i'}]$ such that $\mathsf{W}_i[a] = \mathsf{W}_{i'}[\ell_{i'}]$ but $\mathsf{Z}_i[a] \neq \mathsf{T}_{i'}$. We apply the initial resetting to solve this issue. However, as a result of initial resetting, induced resetting takes place. Our goal is to show that the non-occurrence of the bad events assures that the compatibilty is attained in the final reset tuples $(0^n, \widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*)$ and $(\mathsf{L}, \widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$. We prove all possible cases as follows:

- $\mathsf{X}_i[a] = 0^n \iff \mathsf{Y}_i[a] = \mathsf{L}$: If $a = 1$ and $\mathsf{X}_i[a] = 0$, then $(i, a) \notin \widetilde{\mathsf{FCI}}$ due to $\neg$BadW1. Also, $(i, 1) \notin \widetilde{\mathsf{ICI}}$. Thus, $\mathsf{Y}_i[a] = \mathsf{Z}_i[a] = \mathsf{L}$ and the converse also holds. Otherwise, due to $\neg$BadX1, $\mathsf{X}_i[a]$ can not be equal to 0. Also, due to $\neg$BadW1, $\mathsf{Y}_i[a]$ can not be equal to $\mathsf{L}$.

[3] Does not collide with any other input.

- $\mathsf{X}_i[a] = \mathsf{X}_{i'}[\ell_{i'}] \iff \mathsf{Y}_i[a] = \mathsf{T}_{i'}$: For $(i,a) \in \widetilde{\mathsf{FCI}}$, this equivalence holds. Otherwise, $\mathsf{X}_i[a] = \mathsf{X}_{i'}[\ell_{i'}]$ can not hold due to $\neg(\texttt{BadX1} \vee \texttt{BadX5})$. Also $\mathsf{Y}_i[a] = \mathsf{T}_{i'}$ can not hold due to definition of $\widetilde{\mathsf{T}}$ and $\neg\texttt{BadX2}$.

- $\mathsf{X}_i[a] = \mathsf{X}_j[b] \iff \mathsf{Y}_i[a] = \mathsf{Y}_j[b]$: To prove this part we divide it in the following subcases:

  - $\boxed{(i,a),(i,b) \notin \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}}}$: Since in this case the variables are simply renamed due to definitions of resetting and $\neg\texttt{BadW3}$, the result follows from $\widetilde{\mathsf{W}} \leftrightsquigarrow \widetilde{\mathsf{Z}}$.

  - $\boxed{(i,a),(j,b) \in \widetilde{\mathsf{FCI}}}$: Since $(i,a),(j,b) \in \widetilde{\mathsf{FCI}}$, there exists unique $i', j' \in [q]$, such that $\mathsf{W}_i[a] = \mathsf{W}_{i'}[\ell_{i'}]$ and $\mathsf{W}_j[b] = \mathsf{W}_{j'}[\ell_{j'}]$. Now, note that $\mathsf{X}_i[a] = \mathsf{W}_i[a]$ and $\mathsf{X}_j[b] = \mathsf{W}_j[b]$ since $\widetilde{\mathsf{FCI}} \cap \widetilde{\mathsf{ICI}} = \emptyset$ due to $\neg\texttt{BadW4}$; $\mathsf{W}_{i'}[\ell_{i'}] = \mathsf{X}_{i'}[\ell_{i'}]$ and $\mathsf{W}_{j'}[\ell_{j'}] = \mathsf{X}_{j'}[\ell_{j'}]$ due to $\neg\texttt{BadW5}$. Therefore, we must have $\mathsf{X}_{j'}[\ell_{j'}] = \mathsf{W}_{j'}[\ell_{j'}] = \mathsf{W}_j[b] = \mathsf{X}_j[b] = \mathsf{X}_i[a] = \mathsf{W}_i[a] = \mathsf{W}_{i'}[\ell_{i'}] = \mathsf{X}_{i'}[\ell_{i'}]$, which is possible if and only if $i' = j'$ (since $\neg\texttt{BadW2}$ holds).

  - $\boxed{(i,a),(j,b) \in \widetilde{\mathsf{ICI}}}$: Since $(i,a),(j,b) \in \widetilde{\mathsf{ICI}}$, there exists $i', j' \in [q]$ and $a' \in [\ell_{i'} - 1], b' \in [\ell_{j'} - 1]$, such that $\mathsf{X}_i[a] = \mathsf{W}_{i'}[a']$ and $\mathsf{X}_j[b] = \mathsf{W}_{j'}[b']$. Further, $(i',a'),(j',b') \notin \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}}$ (due to $\neg\texttt{BadX3}$). If $\mathsf{X}_j[b] = \mathsf{X}_i[a]$, then we have $\mathsf{W}_{j'}[b'] = \mathsf{W}_{i'}[a']$. This gives us $\mathsf{Y}_j[b] = \mathsf{Z}_{j'}[b'] = \mathsf{Z}_{i'}[a'] = \mathsf{Y}_i[a]$ (due to $\widetilde{\mathsf{W}} \leftrightsquigarrow \widetilde{\mathsf{Z}}$). Similarly, $\mathsf{X}_i[a] \neq \mathsf{X}_j[b]$ implies $\mathsf{Y}_i[a] \neq \mathsf{Y}_j[b]$.

  - $\boxed{(i,a) \in \widetilde{\mathsf{FCI}} \text{ and } (j,b) \in \widetilde{\mathsf{ICI}}}$: Since $(i,a) \in \widetilde{\mathsf{FCI}}$, there exists a unique $i' \in [q]$, such that $\mathsf{X}_i[a] = \mathsf{W}_i[a] = \mathsf{W}_{i'}[\ell_{i'}] = \mathsf{X}_{i'}[\ell_{i'}]$ (the first equality is due to $\neg\texttt{BadW4}$, the second equality is due to the definition of full collision, the third equality is due to $\neg\texttt{BadW5}$). Since $(j,b) \in \widetilde{\mathsf{ICI}}$, we also have $\mathsf{X}_j[b] = \mathsf{W}_{j'}[b']$. If $\mathsf{X}_i[a] = \mathsf{X}_j[b]$, then $\mathsf{W}_{j'}[b'] = \mathsf{W}_{i'}[\ell_{i'}]$. Thus, $(j',b') = (i',\ell_{i'})$ due to $\neg\texttt{BadX3}$. Now, we have $\mathsf{Y}_i[a] = \mathsf{T}_{i'}$. Also, $\mathsf{Y}_j[b] = \mathsf{Y}_{j'}[b'] = \mathsf{Y}_{i'}[\ell_{i'}] = \mathsf{T}_{i'}$. Therefore, $\mathsf{Y}_i[a] = \mathsf{Y}_j[b]$. Moreover, $\mathsf{X}_i[a] \neq \mathsf{X}_j[b]$ implies that $\mathsf{Y}_i[a] \neq \mathsf{Y}_j[b]$ due to similar arguments as above and also $\neg\texttt{BadT}$.

  - $\boxed{(i,a) \in \widetilde{\mathsf{ICI}} \text{ and } (j,b) \in \widetilde{\mathsf{FCI}}}$: Similar as the above case.

  - $\boxed{(i,a) \in \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}} \text{ and } (j,b) \notin \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}}}$: Since $(j,b) \notin \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}}$, we have $\mathsf{X}_j[b] = \mathsf{W}_j[b]$ and $\mathsf{Y}_j[b] = \mathsf{Z}_j[b]$. Suppose, $(i,a) \in \widetilde{\mathsf{FCI}}$. Then $\mathsf{X}_i[a] = \mathsf{X}_j[b]$ is not possible since it would imply that $(j,b) \in \widetilde{\mathsf{FCI}}$. Also, $\mathsf{Y}_i[a] = \mathsf{Y}_j[b]$ is not possible since it would contradict the definition of $\widetilde{\mathsf{T}}$. Now, suppose, $(i,a) \in \widetilde{\mathsf{ICI}}$. Therefore, $\mathsf{X}_i[a] = \mathsf{W}_{i'}[a']$ for some $i' \in [q]$ and $a' \in [\ell_{i'} - 1]$. If $\mathsf{X}_i[a] = \mathsf{X}_j[b]$, then $\mathsf{W}_j[b] = \mathsf{X}_j[b] = \mathsf{X}_i[a] = \mathsf{W}_{i'}[a']$. So, $\mathsf{Y}_j[b] = \mathsf{Z}_j[b] = \mathsf{Z}_{i'}[a'] = \mathsf{Y}_i[a]$. Similarly, $\mathsf{X}_i[a] \neq \mathsf{X}_j[b]$ implies $\mathsf{Y}_i[a] \neq \mathsf{Y}_j[b]$.

– $\boxed{(i,a) \notin \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}} \text{ and } (j,b) \in \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}}}$: Similar as the above case.

□

### 6.2.2 Transcript Analysis

SET OF TRANSCRIPTS: Given the description of transcript random variable corresponding to the ideal oracle, we can now define the set of transcripts $\mathcal{V}$ as the set of all tuples $\nu = (\widetilde{m}, \widetilde{t}, \widetilde{x}, \widetilde{x}^*, \widetilde{y}, l_{-1}, l_0, \mathrm{flagT}, \mathrm{flagW}, \mathrm{flagX})$, where

- $\widetilde{m} = (m_1, \ldots, m_q)$, where $m_i \in \{0,1\}^*$ for $i \in [q]$. Let $\ell_i = \left\lceil \frac{|m_i|}{n} \right\rceil$ for $i \in [q]$.
- $\widetilde{t} = (t_1, \ldots, t_q)$, where $t_i \in \mathbb{B}$ for $i \in [q]$.
- $\widetilde{x} = (x_1, \ldots, x_q)$, where $x_i = (x_i[1], \ldots, x_i[\ell_i - 1])$ for $i \in [q]$.
- $\widetilde{x}^* = (x_1[\ell_1], \ldots, x_q[\ell_q])$.
- $\widetilde{y} = (y_1, \ldots, y_q)$, where $y_i = (y_i[0] = 0^n, y_i[1], \ldots, y_i[\ell_i - 1])$ for $i \in [q]$.
- $l_{-1} = \mu_{-1} \odot l, l_0 = \mu_0 \odot l$ where $l \in \mathbb{B}$ and $\mu_{-1}, \mu_0$ are constants chosen from $\mathbb{F}_{2^n}$ as defined before.
- $\mathrm{flagT}, \mathrm{flagW}, \mathrm{flagX} \in \{0,1\}$.

Furthermore, the following must always hold:

1. if $\mathrm{flagI} = 1$ for some $\mathrm{I} \in \{\mathrm{T}, \mathrm{W}\}$, then $x_i[a] = y_j[b] = 0^n$ for all $i, j \in [q]$, $a \in [\ell_i]$, and $b \in [\ell_j - 1]$.
2. if $\mathrm{flagT} = 0$, then $t_i$'s are all distinct.
3. if $\mathrm{flagI} = 0$ for all $\mathrm{I} \in \{\mathrm{T}, \mathrm{W}, \mathrm{X}\}$, then $x_i[a] = y_i[a-1] \oplus \overline{m}_i[a]$ and $(0^n, \widetilde{x}, \widetilde{y}^{\oplus}) \leftrightsquigarrow (l, \widetilde{y}, \widetilde{t})$.

The first two conditions are obvious from the ideal oracle sampling mechanism. The last condition follows from Proposition 6.2.3 and the observation that in ideal oracle sampling for any $\mathsf{I} \in \{\mathsf{T}, \mathsf{Z}, \mathsf{X}\}$, $\mathsf{FlagI} = 1$ if and only if $\mathsf{BadI}$ is true. Note that, condition 3 is vacuously true for real oracle transcripts.

BAD TRANSCRIPT: A transcript $\nu \in \mathcal{V}$ is called *bad* if and only if the following predicate is true:

$$(\mathsf{FlagT} = 1) \vee (\mathsf{FlagW} = 1) \vee (\mathsf{FlagX} = 1).$$

In other words, we term a transcript bad if the ideal oracle sets $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*, \widetilde{\mathsf{Y}})$ degenerately. Let

$$\mathcal{V}_{\mathsf{bad}} := \{\nu \in \mathcal{V} : \nu \text{ is bad.}\}.$$

All other transcript $\nu' = (\widetilde{m}, \widetilde{t}, \widetilde{x}, \widetilde{x}^*, \widetilde{y}, l_{-1}, l_0, \mathrm{flagT}, \mathrm{flagW}, \mathrm{flagX}) \in \mathcal{V} \setminus \mathcal{V}_{\mathsf{bad}}$ are called *good*. From the preceding characterization of the set of transcripts, we conclude that for any good transcript $\nu'$, we must have $(0^n, \widetilde{x}, \widetilde{x}^*) \rightsquigarrow (l, \widetilde{y}, \widetilde{t})$. Henceforth, we drop $\mathrm{flagT}$, $\mathrm{flagW}$, and $\mathrm{flagX}$ for any good transcript with an implicit understanding that $\mathrm{flagT} = \mathrm{flagW} = \mathrm{flagX} = 0$.

Following the H-coefficient mechanism, we have to upper bound the probability $\Pr[\Theta_0 \in \mathcal{V}_{\mathsf{bad}}]$ and lower bound the ratio $\Pr[\Theta_1 = \nu]/\Pr[\Theta_0 = \nu]$ for any $\nu \in \mathcal{V} \setminus \mathcal{V}_{\mathsf{bad}}$.

**Lemma 6.2.4** (bad transcript analysis)**.** *For $q + \sigma \leq 2^{n-1}$, we have*

$$\Pr[\Theta_0 \in \mathcal{V}_{\mathsf{bad}}] \leq \frac{4\sigma}{2^n} + \frac{16q^2 + q\ell^2}{2^n} + \frac{8q^2\ell^4 + 32q^3\ell^2 + 2q^2\ell^3}{2^{2n}} + \frac{3q^3\ell^5 + 143q^3\ell^6 + 11q^4\ell^3}{2^{3n}} + \frac{17q^4\ell^6 + 5462q^4\ell^8}{2^{4n}}.$$

The proof of this lemma is postponed to section 6.3.

GOOD TRANSCRIPT: Now, fix a good transcript $\nu = (\widetilde{m}, \widetilde{t}, \widetilde{x}, \widetilde{x}^*, \widetilde{y}, l_{-1}, l_0)$. Let $\sigma$ be the total number of blocks (and one additional for $0^n$) and $\sigma' := |\widetilde{x} \cup \{0^n\}|$. Since, $\nu$ is good, we have $(0^n, \widetilde{x}, \widetilde{x}^*) \rightsquigarrow (l, \widetilde{y}, \widetilde{t})$. Then, we must have $|\widetilde{x}^*| = q$. Further, let $|\widetilde{x} \cap \widetilde{x}^*| = r$. Thus, $|\{0^n\} \cup \widetilde{x} \cup \widetilde{x}^*| = q + \sigma' - r$.

*Real world:* In the real world, the random permutation $\boldsymbol{\pi}$ is sampled on exactly $q+\sigma'-r$ distinct points. Thus, we have

$$\Pr[\Theta_1 = \nu] = \frac{1}{(2^n)_{q+\sigma'-r}}. \tag{6.9}$$

*Ideal world:* In the ideal world, we employed a two stage sampling. First of all, we have

$$\Pr\left[\widetilde{\mathsf{T}} = \widetilde{t}, \mathsf{P}(0^n) = l\right] \leq \frac{1}{2^{nq}}, \tag{6.10}$$

since each $\mathsf{T}_i$ is sampled uniformly from the set $\mathbb{B}$ independent of others. Now, observe that all the full collision and induced collision indices are fully determined from the transcript $\nu$ itself. In other words, we can enumerate the set $\widetilde{\mathsf{CI}} := \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}}$. Now, since

the transcript is good, we must have $|\widetilde{\mathsf{CI}}| = \sigma - \sigma' + |\widetilde{x} \cap \widetilde{x}^*| = \sigma - \sigma' + r$, and for all indices $(i, a) \notin \widetilde{\mathsf{CI}}$, we have $\mathsf{Y}_i[a] = \mathsf{Z}_i[a]$. Thus, we have

$$\begin{aligned}\Pr\left[\mathsf{Y}_i[a] = y_a^i \wedge (i,a) \notin \widetilde{\mathsf{CI}} \mid \widetilde{\mathsf{T}} = \widetilde{t}\right] &= \Pr\left[\mathsf{Z}_i[a] = y_a^i \wedge (i,a) \notin \widetilde{\mathsf{CI}} \mid \widetilde{\mathsf{T}} = \widetilde{t}\right] \\ &= \frac{1}{(2^n - q)_{\sigma' - r}}, \end{aligned} \tag{6.11}$$

where the second equality follows from the fact that truncation[4] of a without replacement sample from a set of size $(2^n - q)$ is still a without replacement sample from the same set. We have

$$\begin{aligned}\Pr\left[\Theta_0 = \omega\right] &= \Pr\left[\widetilde{\mathsf{T}} = \widetilde{t}\right] \times \Pr\left[\widetilde{\mathsf{Y}} = \widetilde{y} \mid \widetilde{\mathsf{T}} = \widetilde{t}\right] \\ &\leq \frac{1}{2^{nq}} \times \Pr\left[\mathsf{Y}_i[a] = y_i[a] \wedge (i,a) \notin \widetilde{\mathsf{CI}} \mid \widetilde{\mathsf{T}} = \widetilde{t}\right] \quad = \frac{1}{2^{nq}(2^n - q)_{\sigma' - r}}. \end{aligned} \tag{6.12}$$

The above discussion on good transcripts can be summarized in shape of the following lemma.

**Lemma 6.2.5.** *For any $\nu \in \mathcal{V} \setminus \mathcal{V}_{\mathsf{bad}}$, we have* $\dfrac{\Pr\left[\Theta_1 = \nu\right]}{\Pr\left[\Theta_0 = \nu\right]} \geq 1.$

*Proof.* The proof follows from dividing (6.9) by (6.12). □

Using Theorem 2.2.7, and Lemma 6.2.4 and 6.2.5, we get

$$\begin{aligned}\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{OMAC}_\pi}(q, \ell, \sigma, \infty) \leq \frac{4\sigma}{2^n} + \frac{16q^2 + q\ell^2}{2^n} + \frac{8q^2\ell^4 + 32q^3\ell^2 + 2q^2\ell^3}{2^{2n}} \\ + \frac{3q^3\ell^5 + 143q^3\ell^6 + 11q^4\ell^3}{2^{3n}} + \frac{17q^4\ell^6 + 5462q^4\ell^8}{2^{4n}}. \end{aligned} \tag{6.13}$$

Theorem 6.1.1 follows from (6.8) and (6.13).

## 6.3 Proof of Lemma 6.2.4

In appendix A, we recall the definition and properties of a combinatorial tool, called structure graphs [9, 52], which will be highly useful in our proof. Our aim will be to bound the probability of bad events only when they occur in conjunction with some "manageable" structure graphs. In all other cases, we upper bound the probability by the probability of realizing an unmanageable structure graph. Formally, we say that the structure graph $\mathcal{G}_{\mathsf{P}}(\widetilde{\mathsf{M}})$ is manageable if and only if:

[4] Removing some elements from the tuple.

1. for all $i \in [q]$, we have $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i)) = 0$, i.e., each $\mathsf{M}_i$-walk is a path.
2. for all distinct $i, j \in [q]$, we have $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j)) \leq 1$.
3. for all distinct $i, j, k \in [q]$, we have $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) \leq 2$.
4. for all distinct $i, j, k, l \in [q]$, we have $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k, \mathsf{M}_l)) \leq 3$.

Let $\mathtt{unman}$ denote the event that $\mathcal{G}_{\mathsf{P}}(\widetilde{\mathsf{M}})$ is unmanageable. Then, using Corollary A.0.5, we have

$$\begin{aligned}
\Pr[\mathtt{unman}] &\leq \Pr[\exists i \in [q] : \mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i)) \geq 1] + \Pr[\exists i < j \in [q] : \mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j)) \geq 2] \\
&\quad + \Pr[\exists i < j < k \in [q] : \mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) \geq 3] \\
&\quad + \Pr[\exists i < j < k < l \in [q] : \mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k, \mathsf{M}_l)) \geq 4] \\
&\leq \sum_{i \in [q]} \frac{(\ell_i - 1)^2}{2^n} + \sum_{i<j \in [q]} \frac{(\ell_i + \ell_j - 2)^4}{2^{2n}} + \sum_{i<j<k \in [q]} \frac{(\ell_i + \ell_j + \ell_k - 3)^6}{2^{3n}} \\
&\quad + \sum_{i<j<k<l \in [q]} \frac{(\ell_i + \ell_j + \ell_k + \ell_l - 4)^8}{2^{4n}} \\
&\leq \frac{q\ell^2}{2^n} + \frac{8q^2\ell^4}{2^{2n}} + \frac{121.5q^3\ell^6}{2^{3n}} + \frac{5461.34q^4\ell^8}{2^{4n}}. \qquad (6.14)
\end{aligned}$$

From now on we only consider manageable graphs. Observe that apart from the fact that a manageable graph is just a union of $\mathsf{M}_i$-paths, there is an added benefit that it has no zero collision. Let $\mathtt{TU} := \neg(\mathtt{BadT} \vee \mathtt{unman})$ and $\mathtt{TUW} := \neg(\mathtt{BadT} \vee \mathtt{unman} \vee \mathtt{BadW})$. Now, we have

$$\begin{aligned}
\Pr[\Theta_0 \in \mathcal{V}_{\mathsf{bad}}] &= \Pr[(\mathsf{FlagT} = 1) \vee (\mathsf{FlagW} = 1) \vee (\mathsf{FlagX} = 1)] \\
&\overset{1}{\leq} \Pr[\mathtt{BadT} \vee \mathtt{BadW} \vee \mathtt{BadX}] \\
&\leq \Pr[\mathtt{BadT}] + \Pr[\mathtt{BadW} | \neg \mathtt{BadT}] + \Pr[\mathtt{BadX} | \neg(\mathtt{BadT} \vee \mathtt{BadW})] \\
&\overset{2}{\leq} \Pr[\exists i \neq j : \mathsf{T}_i = \mathsf{T}_j] + \Pr[\mathtt{BadW} | \neg \mathtt{BadT}] + \Pr[\mathtt{BadX} | \neg(\mathtt{BadT} \vee \mathtt{BadW})] \\
&\overset{3}{\leq} \frac{q^2}{2^{n+1}} + \Pr[\mathtt{unman}] + \Pr[\mathtt{BadW} | \mathtt{TU}] + \Pr[\mathtt{BadX} | \mathtt{TUW}] \\
&\overset{4}{\leq} \frac{0.5q^2 + q\ell^2}{2^n} + \frac{8q^2\ell^4}{2^{2n}} + \frac{122q^3\ell^6}{2^{3n}} + \frac{5462q^4\ell^8}{2^{4n}} \\
&\quad + \Pr[\mathtt{BadW} | \mathtt{TU}] + \Pr[\mathtt{BadX} | \mathtt{TUW}] \qquad (6.15)
\end{aligned}$$

Here, inequalities 1 and 2 follow by definition; 3 follows from the fact that $\mathsf{T}_i$ is chosen uniformly at random from $\mathbb{B}$ for each $i$; and 4 follows from (6.14).

BOUNDING $\Pr[\texttt{BadW}|\neg(\texttt{BadT} \vee \texttt{unman})]$: Let $\texttt{Ei} = \neg(\texttt{TU} \vee \texttt{BadW1} \vee \cdots \vee \texttt{BadWi})$. We have

$$\Pr[\texttt{BadW}|\texttt{TU}] \leq \Pr[\texttt{BadW1}|\texttt{TU}] + \Pr[\texttt{BadW2}|\texttt{E1}] + \Pr[\texttt{BadW3}|\texttt{E2}] + \Pr[\texttt{BadW4}|\texttt{E3}] + \Pr[\texttt{BadW5}|\texttt{E4}] \tag{6.16}$$

We bound the individual terms on the right hand side as follows:

*Bounding* $\Pr[\texttt{BadW1}|\texttt{TU}]$: Fix some $(i, a) \in [q] \times [\ell_i]$. The only way we can have $\mathsf{W}_i[a] = 0^n$, for $1 < a < \ell_i$, is if $\mathsf{Z}_i[a-1] = \mathsf{M}_i[a]$. This happens with probability at most $(2^n - q)^{-1}$. For $a = \ell_i$, the equation

$$\mu_{\delta_{\mathsf{M}_i}} \odot \mathsf{L} \oplus \mathsf{Z}_i[\ell_i - 1] \oplus \overline{\mathsf{M}}_i[\ell_i] = 0^n$$

must hold non-trivially. The probability that this equation holds is bounded by at most $(2^n - q - 1)^{-1}$. Assuming $q + 1 \leq 2^{n-1}$, and using the fact that there can be at most $\sigma$ choices for $(i, a)$, we have

$$\Pr[\texttt{BadW1}|\texttt{TU}] \leq \frac{2\sigma}{2^n}. \tag{6.17}$$

*Bounding* $\Pr[\texttt{BadW2}|\texttt{E1}]$: Fix some $i \neq j \in [q]$. Since $\neg\texttt{unman}$ holds, we know that $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j)) \leq 1$. We handle the two resulting cases separately:

(A) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j)) = 1$: Suppose the collision source of the only accident are $(i, a)$ and $(j, b)$. Then, we have the following system of two equations

$$\begin{aligned} \mathsf{Z}_i[a] \oplus \mathsf{Z}_j[b] &= \mathsf{M}_i[a+1] \oplus \mathsf{M}_j[b+1] \\ (\mu_{\delta_{\mathsf{M}_i}} \oplus \mu_{\delta_{\mathsf{M}_j}}) \odot \mathsf{L} \oplus \mathsf{Z}_i[\ell_i - 1] \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_i[\ell_i] \oplus \overline{\mathsf{M}}_j[\ell_j] \end{aligned}$$

Suppose $\delta_{\mathsf{M}_i} \neq \delta_{\mathsf{M}_j}$, i.e. $\mu_{\delta_{\mathsf{M}_i}} \oplus \mu_{\delta_{\mathsf{M}_j}} \neq 0^n$. Using the fact that $\neg\texttt{BadW1}$ holds, we infer that $\mathsf{L} \notin \{\mathsf{Z}_i[a], \mathsf{Z}_j[b], \mathsf{Z}_i[\ell_i - 1], \mathsf{Z}_j[\ell_j - 1]\}$. So, the two equations are linearly independent, whence the rank is 2 in this case. Again, using Lemma A.0.6, and the fact that there are at most $q^2/2$ choices for $i$ and $j$, and $\ell^2$ choices for $a$ and $b$, we get

$$\Pr\left[\texttt{BadW2} \wedge \text{Case A} \wedge \delta_{\mathsf{M}_i} \neq \delta_{\mathsf{M}_j} | \texttt{E1}\right] \leq \frac{q^2\ell^2}{2(2^n - q - \sigma + 2)^2}.$$

Now, suppose $\delta_{\mathsf{M}_i} = \delta_{\mathsf{M}_j}$, i.e. $\mu_{\delta_{\mathsf{M}_i}} \oplus \mu_{\delta_{\mathsf{M}_j}} = 0^n$. Then, we can rewrite the system as

$$\begin{aligned} \mathsf{Z}_i[a] \oplus \mathsf{Z}_j[b] &= \mathsf{M}_i[a+1] \oplus \mathsf{M}_j[b+1] \\ \mathsf{Z}_i[\ell_i - 1] \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_i[\ell_i] \oplus \overline{\mathsf{M}}_j[\ell_j] \end{aligned}$$

We can have two types of structure graphs relevant to this case, as illustrated in Figure 6.3.1. For type 1 all variables are distinct. So, the two equations are linearly

Type (1) Type (2)

**Figure 6.3.1:** Accident-1 manageable graphs for two messages. The solid and dashed lines correspond to edges in $\mathcal{W}_i$ and $\mathcal{W}_j$, respectively. $*$ denotes optional parts in the walk.

independent, whence the rank is 2 in this case. Again, using Lemma A.0.6, we get

$$\Pr\left[\texttt{BadW2} \wedge \text{Case A} \wedge \delta_{\mathsf{M}_i} = \delta_{\mathsf{M}_j} \wedge \text{Type 1} | \texttt{E1}\right] \leq \frac{q^2\ell^2}{2(2^n - q - \sigma + 2)^2}.$$

For type 2, it is clear that $\mathsf{Z}_j[\ell_j - 1] = \mathsf{Z}_i[\ell_i - 1]$. So, we can assume that the second equation holds trivially, thereby deriving a system in $\mathsf{Z}_i[a]$ and $\mathsf{Z}_j[b]$, with rank 1. Further, $a$ and $b$ are uniquely determined as $\ell_i - p$ and $\ell_j - p$, where $p$ is the longest common suffix of $\mathsf{M}_i$ and $\mathsf{M}_j$. So we have

$$\Pr\left[\texttt{BadW2} \wedge \text{Case A} \wedge \delta_{\mathsf{M}_i} = \delta_{\mathsf{M}_j} \wedge \text{Type 2} | \texttt{E1}\right] \leq \frac{q^2}{2(2^n - q - \sigma + 1)}.$$

(B) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j)) = 0$: In this case, we only have one equation of the form

$$(\mu_{\delta_{\mathsf{M}_i}} \oplus \mu_{\delta_{\mathsf{M}_j}}) \odot \mathsf{L} \oplus \mathsf{Z}_i[\ell_i - 1] \oplus \mathsf{Z}_j[\ell_j - 1] = \overline{\mathsf{M}}_i[\ell_i] \oplus \overline{\mathsf{M}}_j[\ell_j]$$

If $\delta_{\mathsf{M}_i} \neq \delta_{\mathsf{M}_j}$, we have an equation in three variables, namely $\mathsf{L}$, $\mathsf{Z}_i[\ell_i - 1]$, and $\mathsf{Z}_j[\ell_j - 1]$; and if $\delta_{\mathsf{M}_i} = \delta_{\mathsf{M}_j}$, we have an equation in two variables, namely $\mathsf{Z}_i[\ell_i - 1]$, and $\mathsf{Z}_j[\ell_j - 1]$. In both the cases, the equation can only hold non-trivially, i.e, rank is 1. Using Lemma A.0.6, we get

$$\Pr\left[\texttt{BadW2} \wedge \text{Case B} | \texttt{E1}\right] \leq \frac{q^2}{2(2^n - q - \sigma + 1)}.$$

On combining the three cases, we get

$$\Pr\left[\texttt{BadW2} | \texttt{E1}\right] \leq \frac{q^2}{2^n - q - \sigma + 1} + \frac{q^2\ell^2}{(2^n - q - \sigma + 2)^2}. \tag{6.18}$$

*Bounding* $\Pr\left[\texttt{BadW3} | \texttt{E2}\right]$: Fix some $i, j, k \in [q]$. Since $\neg\texttt{unman}$ holds, we must have $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) \leq 2$. Accordingly, we have the following three cases:

(A) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 2$: Suppose $(\alpha_1, \beta_1)$ and $(\alpha_2, \beta_2)$ are collision source leading to one of the accident, and $(\alpha_3, \beta_3)$ and $(\alpha_4, \beta_4)$ are collision source leading to the other accident. Then, considering $\mathsf{W}_i[a] = \mathsf{W}_j[\ell_j]$, we have the following system of equations

$$\begin{aligned}
\mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1] \\
\mathsf{Z}_{\alpha_3}[\beta_3] \oplus \mathsf{Z}_{\alpha_4}[\beta_4] &= \mathsf{M}_{\alpha_3}[\beta_3 + 1] \oplus \mathsf{M}_{\alpha_4}[\beta_4 + 1] \\
\mathsf{Z}_j[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a]
\end{aligned}$$

The first two equations are independent by definition. Further, using $\neg$BadW1, we can infer that the last equation is also independent of the first two equations. Thus the system has rank 3. There are at most $q^3/6$ choices for $(i, j, k)$, and for each such choice we have 3 choices for $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ and at most $\ell^5$ choices for $(\beta_1, \beta_2, \beta_3, \beta_4, a)$. Using Lemma A.0.6, we have

$$\Pr\left[\mathtt{BadW3} \wedge \text{Case A} | \mathtt{E2}\right] \leq \frac{q^3\ell^5}{2(2^n - q - \sigma + 3)^3}.$$

(B) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 1$: Suppose $(\alpha_1, \beta_1)$ and $(\alpha_2, \beta_2)$ are collision source leading to the accident. First consider the case $a < \ell_i - 1$ and $b < \ell_k$. In this case, we have the following system of equations

$$\begin{aligned}
\mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1] \\
\mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a] \\
\mathsf{Z}_i[a] \oplus \mathsf{Z}_k[b-1] &= \mathsf{M}_i[a+1] \oplus \mathsf{M}_k[b]
\end{aligned}$$

The first two equations are clearly independent. Further, since $\mathsf{M}_i \neq \mathsf{M}_k$, the last equation must correspond to a true collision as a consequence of the accident. So, the rank of the above system is 2. Once we fix $(i, j, k)$ and $(a, b)$, we have at most 3 choices for $(\alpha_1, \alpha_2)$, and $\beta_1$ and $\beta_2$ are uniquely determined as $a + 1 - p$ and $b - p$, where $p$ is the largest common suffix of $\mathsf{M}_i[1, \ldots, a+1]$ and $\mathsf{M}_k[1, \ldots, b]$. So, we have

$$\Pr\left[\mathtt{BadW3} \wedge \text{Case B} \wedge a < \ell_i - 1 \wedge b < \ell_k | \mathtt{E2}\right] \leq \frac{q^3\ell^2}{2(2^n - q - \sigma + 2)^2}.$$

Now, suppose $a = \ell_i - 1$. Then we can simply consider the first two equations

$$\begin{aligned}
\mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1] \\
\mathsf{Z}_j[\ell_i - 2] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[\ell_i - 1]
\end{aligned}$$

Clearly, the two equations are independent. We have at most $q^3$ choices for $(i, j, k)$, 3 choices for $(\alpha_1, \alpha_2)$, and $\ell^2$ choices for $(\beta_1, \beta_2)$. So we have

$$\Pr\left[\texttt{BadW3} \wedge \text{Case B} \wedge a = \ell_i - 1 | \texttt{E2}\right] \leq \frac{q^3\ell^2}{2(2^n - q - \sigma + 2)^2}.$$

The case where $a < \ell_i - 1$ and $b = \ell_k$ can be handled similarly by considering the first and the third equations.

(C) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 0$: In this case, we know that the three paths, $\mathcal{W}_i$, $\mathcal{W}_j$, and $\mathcal{W}_k$ do not collide. This implies that we must have $a = \ell_i - 1$, or $b = \ell_k$ or both, in order for $\mathsf{W}_i[a+1] = \mathsf{W}_k[b]$ to hold. First, suppose both $a = \ell_i - 1$ and $b = \ell_k$. Then, we have the following system of equations:

$$\begin{aligned} \mathsf{Z}_j[\ell_i - 2] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[\ell_i - 2] \\ (\mu_{\delta_{\mathsf{M}_i}} \oplus \mu_{\delta_{\mathsf{M}_k}}) \odot \mathsf{L} \oplus \mathsf{Z}_i[\ell_i - 1] \oplus \mathsf{Z}_k[\ell_k - 1] &= \overline{\mathsf{M}}_i[\ell_i] \oplus \overline{\mathsf{M}}_k[\ell_k] \end{aligned}$$

Using the properties of $\mu_{-1}$ and $\mu_0$, and $\neg\texttt{BadW1}$, we can conclude that the above system has rank 2. There are at most $q^3/6$ choices for $(i, j, k)$, and at most $\ell^2$ choices for $(a, b)$. So, we have

$$\Pr\left[\texttt{BadW3} \wedge \text{Case C} \wedge a = \ell_i - 1 \wedge b = \ell_k | \texttt{E2}\right] \leq \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2}.$$

The remaining two cases are similar. We handle the case $a = \ell_i - 1$ and $b < \ell_k$, and the other case can be handled similarly. We have the following system of equations

$$\begin{aligned} \mathsf{Z}_j[\ell_i - 2] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[\ell_i - 2] \\ \mu_{\delta_{\mathsf{M}_i}} \odot \mathsf{L} \oplus \mathsf{Z}_i[\ell_i - 1] \oplus \mathsf{Z}_k[b - 1] &= \overline{\mathsf{M}}_i[\ell_i] \oplus \mathsf{M}_k[b] \end{aligned}$$

If $\delta_{\mathsf{M}_i} \neq \delta_{\mathsf{M}_j}$, then using the same argument as above, we can conclude that the system has rank 2, and we get

$$\Pr\left[\texttt{BadW3} \wedge \text{Case C} \wedge a = \ell_i - 1 \wedge b < \ell_k \wedge \delta_{\mathsf{M}_i} \neq \delta_{\mathsf{M}_j} | \texttt{E2}\right] \leq \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2}.$$

So, suppose $\delta_{\mathsf{M}_i} = \delta_{\mathsf{M}_j}$. Now, in order for the second equation to be a consequence of the first equation, we must have $\mathsf{Z}_i[\ell_i - 2] = \mathsf{Z}_j[\ell_j - 1]$ and $\mathsf{Z}_i[\ell_i - 1] = \mathsf{Z}_k[b]$. The only we way this happens trivially is if $\mathsf{M}_i[1, \ldots, \ell_i - 1] = \mathsf{M}_j[1, \ldots, \ell_j - 1]$ and $\mathsf{M}_i[1, \ldots, \ell_i - 1] = \mathsf{M}_k[1, \ldots, b]$. But, then we have $b = \ell_i - 1$, and once we fix $(i, k)$ there's a unique choice for $j$, since $\mathsf{M}_j[1, \ldots, \ell_j - 1] = \mathsf{M}_i[1, \ldots, \ell_i - 1]$ and

$\overline{\mathsf{M}}_j[\ell_j] = \overline{\mathsf{M}}_i[\ell_i] \oplus \mathsf{M}_i[\ell_i - 2] \oplus \mathsf{M}_k[b]$. So, we get

$$\Pr\left[\texttt{BadW3} \wedge \text{Case C} \wedge a = \ell_i - 1 \wedge b < \ell_k \wedge \delta_{\mathsf{M}_i} = \delta_{\mathsf{M}_j} | \texttt{E2}\right] \leq \frac{q^2}{2(2^n - q - \sigma + 1)}.$$

By combining all three cases, we have

$$\Pr\left[\texttt{BadW3}|\texttt{E2}\right] \leq \frac{q^3\ell^5}{2(2^n - q - \sigma + 3)^3} + \frac{2q^3\ell^2}{(2^n - q - \sigma + 2)^2} + \frac{q^2}{2(2^n - q - \sigma + 1)}. \quad (6.19)$$

*Bounding* $\Pr\left[\texttt{BadW4}|\texttt{E3}\right]$*:* Fix some $i, j, k \in [q]$. The analysis in this case is very similar to the one in case of BadW3|E2. So we will skip detailed argumentation whenever possible. Since $\neg\texttt{unman}$ holds, we must have $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) \leq 2$. Accordingly, we have the following three cases:

(A) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 2$: This can be bounded by using exactly the same argument as used in Case A for BadW3|E2. So, we have

$$\Pr\left[\texttt{BadW4} \wedge \text{Case A}|\texttt{E3}\right] \leq \frac{q^3\ell^5}{2(2^n - q - \sigma + 3)^3}.$$

(B) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 1$: Suppose $(\alpha_1, \beta_1)$ and $(\alpha_2, \beta_2)$ are collision source leading to the accident. Without loss of generality we assume $a < b$. Specifically, $b \leq \ell_i - 1$ and $a \leq b - 2$ due to $\neg(\texttt{BadW2} \wedge \texttt{BadW3})$. First consider the case $b = \ell_i - 1$. In this case, considering $\mathsf{W}_i[b] = \mathsf{W}_k[\ell_k]$, we have the following system of equations

$$\begin{aligned} \mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1] \\ \mathsf{Z}_i[b-1] \oplus \mu_{\delta_{\mathsf{M}_k}} \odot \mathsf{L} \oplus \mathsf{Z}_k[\ell_k - 1] &= \overline{\mathsf{M}}_k[\ell_k] \oplus \mathsf{M}_i[b] \end{aligned}$$

Using a similar argument as used in previous such cases, we establish that the two equations are independent. Now, once we fix $(i, j, k)$, we have exactly one choice for $b$, at most 3 choices for $(\alpha_1, \alpha_2)$, and $\ell^2$ choices for $(\beta_1, \beta_2)$. So, we have

$$\Pr\left[\texttt{BadW4} \wedge \text{Case B} \wedge b = \ell_i - 1|\texttt{E3}\right] \leq \frac{q^3\ell^2}{2(2^n - q - \sigma + 2)^2}.$$

Now, suppose $b < \ell_i - 1$. Here we can have two cases:

(B.1) *$\mathcal{W}_i$ is involved in the accident*: Without loss of generality assume that $\alpha_1 = i$ and $\beta_1 \in [\ell_i - 1]$. Then, we have the following system of equations:

$$\mathsf{Z}_i[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] = \mathsf{M}_i[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1]$$

$$\mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] = \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a]$$
$$\mathsf{Z}_i[b-1] \oplus \mu_{\delta_{\mathsf{M}_k}} \odot \mathsf{L} \oplus \mathsf{Z}_k[\ell_k - 1] = \overline{\mathsf{M}}_k[\ell_k] \oplus \mathsf{M}_i[b]$$

Suppose $\mathsf{Z}_i[\beta_1] = \mathsf{Z}_i[a-1]$. Then, we must have $\beta_1 = a - 1$ as the graph is manageable.In this case, we consider the first two equations. It is easy to see that the two equations are independent, and once we fix $i, j, k$, there are at most 2 choices for $\alpha_2$ and $\ell^2$ choices for $(\beta_1, \beta_2)$, which gives a unique choice for $a$. So, we have

$$\Pr\left[\texttt{BadW4} \wedge \text{Case B.1} \wedge \beta_1 = a - 1 | \texttt{E3}\right] \leq \frac{q^3\ell^2}{2(2^n - q - \sigma + 2)^2}.$$

We get identical bound for the case when $\mathsf{Z}_i[\beta_1] = \mathsf{Z}_i[b-1]$. Suppose $\mathsf{Z}_i[\beta_1] \notin \{\mathsf{Z}_i[a-1], \mathsf{Z}_i[b-1]\}$. Then, using the fact that there is only one accident in the graph and that accident is due to $(i, \beta_1)$ and $(\alpha_2, \beta_2)$, we infer that $\mathsf{Z}_{\alpha_2}[\beta_2] \notin \{\mathsf{Z}_i[a-1], \mathsf{Z}_i[b-1]\}$. Now, the only way rank of the above system reduces to 2, is if $\mathsf{Z}_i[a-1] = \mathsf{Z}_k[\ell_k - 1]$ and $\mathsf{Z}_i[b-1] = \mathsf{Z}_j[\ell_j - 1]$ trivially. However, if this happens then $a$ and $b$ are uniquely determined by our choice of $(i, j, k, \beta_1, \alpha_2, \beta_2)$. See Figure 6.3.2 for the two possible structure graphs depending upon the value of $\alpha_2$. Basically, based on the choice of $\alpha_2$, $a \in \{\ell_k, \ell_k - \beta_2 + \beta_1\}$. Similarly, $b \in \{\ell_j, \ell_j - \beta_2 + \beta_1\}$. So, using Lemma A.0.6, we get

$$\Pr\left[\texttt{BadW4} \wedge \text{Case B.1} \wedge \beta_1 \notin \{a-1, b-1\} | \texttt{E3}\right] \leq \frac{2q^3\ell^2}{3(2^n - q - \sigma + 2)^2}.$$


**Figure 6.3.2:** Manageable graphs for case B.1. The solid, dashed and dotted lines correspond to edges in $\mathcal{W}_i$, $\mathcal{W}_j$, and $\mathcal{W}_k$, respectively.

(B.2) *$\mathcal{W}_i$ is not involved in the accident*: Without loss of generality assume $\alpha_1 = j$ and $\alpha_2 = k$. Then, we have the following system of equations:

$$\mathsf{Z}_j[\beta_1] \oplus \mathsf{Z}_k[\beta_2] = \mathsf{M}_j[\beta_1 + 1] \oplus \mathsf{M}_k[\beta_2 + 1]$$
$$\mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] = \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a]$$
$$\mathsf{Z}_i[b-1] \oplus \mu_{\delta_{\mathsf{M}_k}} \odot \mathsf{L} \oplus \mathsf{Z}_k[\ell_k - 1] = \overline{\mathsf{M}}_k[\ell_k] \oplus \mathsf{M}_i[b]$$

Since the graph is manageable, $\{\mathsf{Z}_i[a-1], \mathsf{Z}_i[b-1]\} \cap \{\mathsf{Z}_j[\ell_j - 1], \mathsf{Z}_k[\ell_k - 1]\} \neq \emptyset$.

Suppose $\{\mathsf{Z}_i[a-1], \mathsf{Z}_i[b-1]\} = \{\mathsf{Z}_j[\ell_j - 1], \mathsf{Z}_k[\ell_k - 1]\}$. Without loss of generality, assume $\mathsf{Z}_i[a-1] = \mathsf{Z}_k[\ell_k - 1]$ and $\mathsf{Z}_i[b-1] = \mathsf{Z}_j[\ell_j - 1]$. This can only happen if the resulting graph is of Type 2 form in Figure 6.3.2, which clearly shows that we have unique choices for $a$ and $b$ when we fix the other indices. Now, suppose $|\{\mathsf{Z}_i[a-1], \mathsf{Z}_i[b-1]\} \cap \{\mathsf{Z}_j[\ell_j - 1], \mathsf{Z}_k[\ell_k - 1]\}| = 1$. Then, we must have $\mathsf{Z}_i[a-1] \in \{\mathsf{Z}_j[\beta_1], \mathsf{Z}_k[\beta_2]\}$ since $a < b$. Without loss of generality we assume that $\mathsf{Z}_i[a-1] = \mathsf{Z}_k[\beta_2]$ and $\mathsf{Z}_i[b-1] = \mathsf{Z}_j[\ell_j - 1]$. Using similar argument as before, we conclude that $a$ and $b$ are fixed once we fix all other indices. So using Lemma A.0.6, we get

$$\Pr\left[\texttt{BadW4} \wedge \text{Case B.2} | \texttt{E3}\right] \leq \frac{2q^3\ell^2}{3(2^n - q - \sigma + 2)^2}.$$

(C) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 0$: In this case, we know that the three paths, $\mathcal{W}_i$, $\mathcal{W}_j$, and $\mathcal{W}_k$ do not collide. We have the following system of equations:

$$\mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] = \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a]$$
$$\mathsf{Z}_i[b-1] \oplus \mu_{\delta_{\mathsf{M}_k}}) \odot \mathsf{L} \oplus \mathsf{Z}_k[\ell_k - 1] = \overline{\mathsf{M}}_i[\ell_k] \oplus \mathsf{M}_i[b]$$

Using a similar analysis as in case C of BadW3|E2, we get

$$\Pr\left[\texttt{BadW4} \wedge \text{Case C} | \texttt{E3}\right] \leq \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2} + \frac{q^2}{2(2^n - q - \sigma + 1)}.$$

By combining all three cases, we have

$$\Pr\left[\texttt{BadW4} | \texttt{E3}\right] \leq \frac{q^3\ell^5}{2(2^n - q - \sigma + 3)^3} + \frac{3q^3\ell^2}{(2^n - q - \sigma + 2)^2} + \frac{q^2}{2(2^n - q - \sigma + 1)}. \quad (6.20)$$

*Bounding* $\Pr\left[\texttt{BadW5} | \texttt{E4}\right]$: Fix some $i, j, k \in [q]$. The analysis in this case is again similar to the analysis of BadW3|E2 and BadW4|E3. We have the following three cases:

(A) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 2$: This can be bounded by using exactly the same argument as used in Case A for BadW3|E2. So, we have

$$\Pr\left[\texttt{BadW5} \wedge \text{Case A} | \texttt{E4}\right] \leq \frac{q^3\ell^5}{2(2^n - q - \sigma + 3)^3}.$$

(B) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 1$: Suppose $(\alpha_1, \beta_1)$ and $(\alpha_2, \beta_2)$ are collision source leading to the accident. In this case, we have the following system of equations

$$\mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] = \mathsf{M}_{\alpha_1}[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1]$$

$$\mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] = \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a]$$
$$\mathsf{Z}_j[b-1] \oplus \mu_{\delta_{\mathsf{M}_k}} \odot \mathsf{L} \oplus \mathsf{Z}_k[\ell_k - 1] = \overline{\mathsf{M}}_k[\ell_k] \oplus \mathsf{M}_j[b]$$

We can have two sub-cases:

(B.1) Suppose the third equation is simply a consequence of the second equation. Then, we must have $\delta_{\mathsf{M}_i} = \delta_{\mathsf{M}_j}$ and $\mathsf{Z}_i[a-1] = \mathsf{Z}_j[b-1]$ and $\mathsf{Z}_j[\ell_j-1] = \mathsf{Z}_k[\ell_k-1]$ must hold trivially, since the graph is manageable. We claim that $a = b = \mathsf{Prefix}(\mathsf{M}_i[1], \mathsf{M}_j[1]) + 1$. If not, then $\mathsf{M}_i[\ell_i] = \mathsf{M}_j[\ell_j]$ which in conjunction with $\mathsf{Z}_j[\ell_j - 1] = \mathsf{Z}_k[\ell_k - 1]$ implies that $\mathsf{W}_i[\ell_i] = \mathsf{W}_j[\ell_j]$ which contradicts BadW2. So, using Lemma A.0.6, we get

$$\Pr\left[\mathtt{BadW5} \wedge \text{Case B.1} | \mathtt{E4}\right] \leq \frac{q^3\ell^2}{2(2^n - q - \sigma + 2)^2}.$$

(B.2) The second and third equation are independent. Considering the sub-system consisting of these two equations, and using Lemma A.0.6, we get

$$\Pr\left[\mathtt{BadW5} \wedge \text{Case B.2} | \mathtt{E4}\right] \leq \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2}.$$

(C) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 0$: We have the following system of equations:

$$\mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] = \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a]$$
$$\mathsf{Z}_i[b-1] \oplus \mu_{\delta_{\mathsf{M}_k}} \odot \mathsf{L} \oplus \mathsf{Z}_k[\ell_k - 1] = \overline{\mathsf{M}}_i[\ell_k] \oplus \mathsf{M}_i[b]$$

Let $r$ denote the rank of the above system. Using a similar analysis as in case B.1 above, we conclude that $a = b = \mathsf{Prefix}(\mathsf{M}_i[1], \mathsf{M}_j[1]) + 1$ if $r = 1$. Using Lemma A.0.6, we get

$$\Pr\left[\mathtt{BadW5} \wedge \text{Case C} \wedge r = 1 | \mathtt{E4}\right] \leq \frac{q^2}{2(2^n - q - \sigma + 1)}.$$
$$\Pr\left[\mathtt{BadW5} \wedge \text{Case C} \wedge r = 2 | \mathtt{E4}\right] \leq \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2}.$$

By combining all three cases, we have

$$\Pr\left[\mathtt{BadW5} | \mathtt{E4}\right] \leq \frac{q^3\ell^5}{2(2^n - q - \sigma + 3)^3} + \frac{5q^3\ell^2}{6(2^n - q - \sigma + 2)^2} + \frac{q^2}{2(2^n - q - \sigma + 1)}. \quad (6.21)$$

Further, from Eq. (6.16)-(6.21), we have

$$\Pr\left[\mathtt{BadW} | \mathtt{TU}\right] \leq \frac{2\sigma}{2^n} + \frac{5q^2}{2(2^n - q - \sigma + 1)} + \frac{7q^3\ell^2}{(2^n - q - \sigma + 2)^2} + \frac{3q^3\ell^5}{2(2^n - q - \sigma + 3)^3}. \quad (6.22)$$

BOUNDING $\Pr[\mathtt{BadX}|\mathtt{TUW}]$: For brevity of presentation, we skip the rest of the proof and keep it at Appendix A. In A.1, we show that

$$\Pr[\mathtt{BadX}|\mathtt{TUW}] \leq \frac{2\sigma}{2^n} + \frac{10q^2}{2^n - q - \sigma + 1} + \frac{15q^3\ell^2 + q^2\ell^3}{(2^n - q - \sigma + 2)^2} + \frac{12q^3\ell^6 + 6q^4\ell^3}{(2^n - q - \sigma + 3)^3} + \frac{8q^4\ell^6}{(2^n - q - \sigma + 4)^4} \tag{6.23}$$

Combining Eq. (6.15), (6.22), and (6.23), we have

$$\Pr[\Theta_0 \in \mathcal{V}_{\mathsf{bad}}] \leq \frac{4\sigma}{2^n} + \frac{16q^2 + q\ell^2}{2^n} + \frac{8q^2\ell^4 + 32q^3\ell^2 + 2q^2\ell^3}{2^{2n}} + \frac{3q^3\ell^5 + 143q^3\ell^6 + 11q^4\ell^3}{2^{3n}} + \frac{17q^4\ell^6 + 5462q^4\ell^8}{2^{4n}}. \tag{6.24}$$

## 6.4 Key Results At a Glance

Theorems 6.1.1 and 6.1.2 show that OMAC, XCBC and TMAC achieve $O(q^2/2^n)+O(q\ell^2/2^n)$ security. This bound is *almost* tight in term of $q$ if $\ell \ll 2^{n/4}$.

# Part IV

# Conclusion

# Chapter 7

# Summary and Future Works

In this chapter, we compile summary of the whole thesis in a chapterwise basis and give some direction for possible futute works.

## 7.1 Summary

### 7.1.1 Summary of chapter 3

In this chapter, we revisited some difficulties in designing a PMAC variant that has length independent security bound $O(q^2/2^n)$ up to $\ell < 2^{n/2}$. Particularly, we took a closer look at a recent PMAC variant by Naito [70] that claims to have length independent security bound. We showed that the security proof of this construction has a non-trivial gap which is not easy to fix. Indeed, we pose it as an open problem to prove or disprove the $\ell$-free security bound of $O(q^2/2^n)$ for Naito's construction. Apparently, this problem could be as hard as a similar problem posed in context of PMAC1 [85]. On a positive note, we show that 2AXU (see section 3.3) masking function is sufficient to achieve length independent security up to $\ell < 2^{n/2}$. This is a relaxation from the 4-wise independence condition used in [39]. Finally, we proposed a simple variant of PMAC1, called PMAC2, that achieves $\ell$-free security up to $\ell \leq 2^{n/4}$. For the range $2^{n/4} < \ell \leq 2^{n-2}$, PMAC2 still achieves $\ell$-free security while $\sigma < 2^{2n/3}$.

### 7.1.2 Summary of chapter 4

In this chapter, we studied the single-key instance of LightMAC, an ISO/IEC standard for lightweight message authentication codes. Our main contribution is a query-length independent security bound for 1k-LightMAC. Specifically, we showed that 1k-LightMAC achieves PRF security bound of $O(q^2/2^n)$ while $(n-s) \leq \ell \leq (n-s)\min\{2^{n/4}, 2^s\}$. Further, we proposed a slight variant of LightMAC, called LightMAC-ds that achieves security bound of $O(q^2/2^n)$ while $\ell \leq (n-s)2^{s-1}$.

### 7.1.3 Summary of chapter 5

In this chapter, we have reduced the problem of getting exact PRF advantage to find colliding probability for a pair of messages. So it would be an interesting open problem to study the collision probability for all masking of the form $\omega_i \cdot \Delta$ for some sequence of distinct nonzero elements $\omega_1, \omega_2, \ldots$. For example, analyzing colliding probability for PMAC1 is still an open problem. In fact, we do not know any deterministic sequence of constants $\omega_1, \omega_2, \ldots$ for which $\omega_i \cdot \Delta$ would give $q^2/2^n$ PRF security advantage.

### 7.1.4 Summary of chapter 6

In this chapter, we proved that OMAC, XCBC and TMAC are secure up to $q \leq 2^{n/2}$ queries, while the message length $\ell \leq 2^{n/4}$. As a consequence, we have proved that OMAC – a single-keyed CBC-MAC variant – achieves the same security level as some of the more elaborate CBC-MAC variants like EMAC and ECBC. This, in combination with the existing results [52, 53], shows that the security is tight up to $\ell \leq 2^{n/4}$ for all CBC-MAC variants except for the original CBC-MAC.

## 7.2 Possible Future Works

The reset sampling may, in future, find wide applications in the analysis of single-key variant of beyond birthday bound secure constructions, such as LightMAC+[69], PMAC+[98] etc.

As mentioned before, all the analysis in chapter 5 is made for simplified version of PMAC-type construction. In the original construction, the last message block is simply xor-ed instead of encrypting the block. We believe that the similar analysis will go through mostly. But this would involve much more notation and the proof will become

further complex. Moreover, some additional requirement for the encoding function would be required. Once this result is established, improving further for more general construction would be an interesting research problem.

It could be an interesting future problem to extend our analysis and derive similar bounds for CBC-MAC over prefix-free message space. In order to prove our claims, we employed reset-sampling method as mentioned in the chapter 4, which seems to be a promising tool in reducing the length-dependency in single-keyed iterated constructions. Indeed, we believe that this tool might even be useful in obtaining better security bounds for single-keyed variants of many beyond-the-birthday-bound constructions.

# Appendix A

# Structure Graphs

In this section, we recall the definition and properties of a combinatorial tool, called structure graphs [9, 52]. Fix a tuple of $q$ distinct messages $\widetilde{m} = (m_1, \ldots, m_q)$, where $m_i \in \mathbb{B}^{\ell_i}$. Let $\sigma_i = \sum_{j \in [i]} \ell_j$, and $\sigma_q = \sigma$. Note that, we have assumed that the last block of each message is full $n$-bit long (if not it can be suitably padded). Let $\mathcal{Q} := \{(i, a) \in [q] \times (\ell_i - 1]\}$, and $\leq$ be a natural ordering over $\mathcal{Q}$, defined as follows:

$$(i, a) \leq (i', a') \text{ if and only if } (i < i') \text{ or } (i = i' \text{ and } j \leq j').$$

In context of the poset $(\mathcal{Q}, \leq) = (\alpha_1 \leq \cdots \leq \alpha_\sigma)$, we can naturally define $\alpha_i + j$ as $\alpha_{i+j}$ for any $i \in [\sigma]$ and $j \in [\sigma - i]$. One can define subtraction analogously. Sometimes we also use the subset $\mathcal{Q}^+ := \mathcal{Q} \setminus \{(i, 0) : i \in [q]\}$. Going forward, we sometimes write $v_i[a]$ succinctly as $v_\alpha$ for any $\alpha = (i, a) \in \mathcal{Q}$ and any appropriately defined notation $v$.

For the message tuple $\widetilde{m}$ and a permutation $\pi \in \mathsf{Perm}(n)$, let $\widetilde{z}$ denote the intermediate output tuple generated in OMAC function evaluation over each of the $q$ messages in $\widetilde{m}$, i.e., $\mathsf{Z}_i[0] = 0^n$, $\mathsf{Z}_i[a] = \pi(\mathsf{Z}_i[a-1] \oplus m_i[a])$, for all $(i, a) \in \mathcal{Q}$. Let $\mathrm{in}(i, a) := \min\{(j, b) \leq (i, a) : \mathsf{Z}_i[a] = \mathsf{Z}_j[b]\}$.

STRUCTURE GRAPHS: Given the message tuple $\widetilde{m}$ and permutation $\pi$, the *structure graph* $\mathcal{G}_\pi(\widetilde{m}) := (\mathcal{V}, \mathcal{E})$, is an edge-labeled directed graph, where the set of vertices $\mathcal{V} = \{\mathrm{in}(\alpha) : \alpha \in \mathcal{Q}\}$, the set of edges $\mathcal{E} = \{e_\alpha := (\mathrm{in}(\alpha - 1), \mathrm{in}(\alpha)) : \alpha \in \mathcal{Q}^+\}$, and edge $e_\alpha$ is labeled $m_\alpha$ for all $\alpha \in \mathcal{Q}^+$. Note that, it is possible that $e_\alpha = e_\beta$ for some $\alpha, \beta \in \mathcal{Q}^+$, i.e., they represent the same edge with obviously the same label. When we consider a single message $m_r$, the resulting subgraph is simply a walk, that we call an $m_r$-walk and denote as $\mathcal{W}_r$, starting at node $(1, 0)$ and following the labels from $(m_r[1], \ldots, m_r[\ell_r - 1])$. So, a structure graph can also be viewed as an union of $m_i$-walks for all $i \in [q]$.

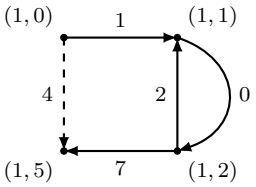


**Figure A.0.1:** Structure graph corresponding to the messages $m_1 = (1, 0, 2, 0, 7, 1)$ and $m_2 = (4, 1)$, and permutation $\pi$, with $\pi(1) = 2$, $\pi(2) = 3$ and $\pi(4) = 5$. The solid lines correspond to edges in $\mathcal{W}_1$, and dashed lines correspond to edges in $\mathcal{W}_2$.

**Example A.1.** *Let* $m_1 = (1, 0, 2, 0, 7, 1)$ *and* $m_2 = (4, 1)$ *be two messages and* $\pi(1) = 2$; $\pi(2) = 3$; $\pi(4) = 5$ *for some* $\pi \in \mathsf{Perm}$. *Then, we have* $z_1 = (0, 2, 3, 2, 3, 5)$ *and* $z_2 = (0, 5)$. *The corresponding structure graph* $\mathcal{G}_\pi(m_1, m_2)$, *illustrated in Figure A.0.1, has vertex set* $\mathcal{V} = \{(1, 0), (1, 1), (1, 2), (1, 5)\}$ *and edges set*

$$\mathcal{E} = \{((1, 0), (1, 1)), ((1, 1), (1, 2)), ((1, 2), (1, 1)), ((1, 2), (1, 5)), ((1, 0), (1, 5))\}.$$

COLLISIONS AND ACCIDENTS: Suppose that $\mathcal{G}_\pi(\widetilde{m})$ is revealed edge by edge in an orderly fashion following $(\mathcal{Q}^+, \leq)$. We say that an edge $e_\alpha$ leads to a *collision* if $\mathrm{in}(\alpha)$ is already present in the partially revealed graph. A collision formed by edges $e_\alpha$ and $e_\beta$ is generally denoted as $(\mathrm{in}(\alpha - 1), \mathrm{in}(\beta - 1); \gamma)$, where $\gamma = \mathrm{in}(\alpha) = \mathrm{in}(\beta)$. The only exception occurs when $\gamma = (1, 0)$ and there is no prior edge to $(1, 0)$, in which case the collision is denoted as $(\mathrm{in}(\alpha - 1); \gamma)$, since prior to $e_\alpha$ there's no edge pointing to $(1, 0)$. This exceptional case is referred as a *zero collision*, and all other collisions are referred as *true collisions*. We refer to $\mathrm{in}(\alpha - 1)$ (and $\mathrm{in}(\beta - 1)$, if applicable) as *collision source*.

Note that it is not possible to recover the intermediate output tuple, by just looking at a given structure graph. Indeed, multiple intermediate output tuples may give the same structure graphs. However, a structure graph does preserve the collision relation between intermediate outputs. More precisely, let $\mathsf{Z}_{\mathrm{in}(\alpha)}$ denote the variable for the intermediate output corresponding to the vertex $\mathrm{in}(\alpha)$, and $\widetilde{Z} = (\mathsf{Z}_{\mathrm{in}(\alpha)} : \alpha \in \mathcal{Q})$. Obviously, we must have $\mathsf{Z}_1[0] = 0^n$, otherwise the resulting intermediate output tuple is invalid. Now, any true collision $(\mathrm{in}(\alpha - 1), \mathrm{in}(\beta - 1); \gamma)$ implies a linear equation

$$\mathsf{Z}_{\mathrm{in}(\alpha-1)} \oplus \mathsf{Z}_{\mathrm{in}(\beta-1)} = m_\alpha \oplus m_\beta,$$

since both $\mathsf{Z}_{\mathrm{in}(\alpha-1)} \oplus m_\alpha$ and $\mathsf{Z}_{\mathrm{in}(\beta-1)} \oplus m_\beta$ must equal $\pi^{-1}(\mathsf{Z}_\gamma)$. Any new true collision can either give a linear equation that is linearly dependent on the linear equations due to previously discovered true collisions, or it may give an independent linear equation. True collisions of the latter type are referred as *accidents*. At a high level, accidents denote the "surprising" collisions in CBC function computation. Obviously, the number

of true collisions is at least the number of accidents. The following definition due to Jha and Nandi [52] gives a formula for the number of accidents.

**Definition A.0.1** ([52]). Consider the structure graph $\mathcal{G}_\pi(\widetilde{m})$ associated with the message tuple $\widetilde{m}$ and permutation $\pi$. Let $\mathcal{S}(\mathcal{G}_\pi(\widetilde{m}))$ be the system of linear equations formed by the true collisions of $\mathcal{G}_\pi(\widetilde{m})$, and let $r$ denote the rank of $\mathcal{S}(\mathcal{G}_\pi(\widetilde{m}))$. Let $\mathsf{Acc}(\mathcal{G}_\pi(\widetilde{m}))$ be the set of accidents of $\mathcal{G}_\pi(\widetilde{m})$. Then, the number of accidents is defined as

$$|\mathsf{Acc}(\mathcal{G}_\pi(\widetilde{m}))| = \begin{cases} r+1 & \text{if } \mathcal{G}_\pi(\widetilde{m}) \text{ has a zero collision,} \\ r & \text{otherwise.} \end{cases}$$

**Example A.2.** *Consider the structure graph from Figure A.0.1. Here, we have two true collisions, namely* $((1,0),(1,2);(1,1))$ *and* $((1,0),(1,2);(1,5))$*, and the associated system of equations is*

$$\mathsf{Z}_1[0] \oplus \mathsf{Z}_1[2] = m_1[1] \oplus m_1[3]$$
$$\mathsf{Z}_1[0] \oplus \mathsf{Z}_1[2] = m_1[5] \oplus m_2[1]$$

*Clearly, the two equations are dependent. So the graph has just one accident, and that accident is* $((1,0),(1,2);(1,1))$*, since it occurs before* $((1,0),(1,2);(1,5))$*. We encourage the readers to see [9, 52] for further exposition on true collisions and accidents.*

EXISTING RESULTS ON STRUCTURE GRAPHS: We now recall some known and useful combinatorial results on structure graphs. The proof of these results are already available in [9, 52].

**Lemma A.0.2** ([52]). *For any structure graph $G$, if there is a vertex $\alpha$ with in-degree $d$ then $\mathsf{Acc}(G) \geq d-1$. Moreover, if the graph has a zero collision then $\mathsf{Acc}(G) \geq d$.*

**Lemma A.0.3** ([9, 52]). *The number of structures graphs associated to $\widetilde{m}$ with $a$ accidents is at most $\binom{\sigma}{2}^a$. In particular, there exists exactly one structure graph with $0$ accidents.*

**Lemma A.0.4** ([9, 52]). *For any structure graph $G$ with $a$ accidents, we have*

$$\Pr_{\mathsf{P}}[\mathcal{G}_{\mathsf{P}}(\widetilde{m}) = G] \leq \frac{1}{(2^n - q - \sigma)^a},$$

*where $\mathsf{P}$ denotes the partial function, introduced in Phase II of the ideal world sampling (see section 6.2.1), that samples each new point in a without replacement manner from a set of size $(2^n - q)$.*

*Proof.* The proof of this lemma is identical to the proof of [52, Lemma 2], with a small change in the probability distribution. For all $\alpha \in \mathcal{Q}^+$, the values for $\mathsf{Z}_{\mathrm{in}(\alpha)}$ are now chosen in a without replacement manner from a set of size $(2^n - q)$ (instead of $2^n$ in case of [52, Lemma 2]). □

**Corollary A.0.5** ([9, 52]). *For $a \in \mathbb{N}$ and $q + \sigma < 2^{n-1}$, we have*

$$\Pr_{\mathsf{P}}[\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\widetilde{m})) \geq a] \leq \frac{\sigma^{2a}}{2^{an}},$$

AN EXTENSION OF LEMMA A.0.4: In addition to the system of linear equations $\mathcal{S}(\mathcal{G}_{\mathsf{P}}(\widetilde{m}))$, sometimes we also consider additional equations over the intermediate output variables tailor-made to our analysis. Suppose the system of these additional equations is denoted simply by $\mathcal{S}'(\widetilde{Z})$, and $\mathcal{S}'(\widetilde{Z}) \cup \mathcal{S}(\mathcal{G}_{\mathsf{P}}(\widetilde{m}))$ denotes the system consisting of equations from both $\mathcal{S}(\mathcal{G}_{\mathsf{P}}(\widetilde{m}))$ and $\mathcal{S}'(\widetilde{Z})$. Let $r$ denote the rank of the *combined system of equations*, $\mathcal{S}'(\widetilde{Z}) \cup \mathcal{S}(\mathcal{G}_{\mathsf{P}}(\widetilde{m}))$.

Suppose $|\widetilde{Z}| = t \leq \sigma$ and let $1_0$ be an indicator variable that results in 1 if $\mathcal{G}_{\mathsf{P}}(\widetilde{m})$ has a zero collision, and 0 otherwise. Then, we can have at most $(2^n - q)_{t+1_0}$ valid intermediate output tuples, since we must have $Z_{\mathrm{in}(\alpha)} \neq Z_{\mathrm{in}(\beta)}$ whenever $\mathrm{in}(\alpha) \neq \mathrm{in}(\beta)$, whence the probability to realize one such valid intermediate output tuple is $1/(2^n - q)_{t+1_0}$. Since the rank of $\mathcal{S}'(\widetilde{Z}) \cup \mathcal{S}(\mathcal{G}_{\pi}(\widetilde{m}))$ is $r$, by simple linear algebraic argument we know that by choosing the value of any $t - r$ variables, we may get a unique solution for the other $r$ variables, in such a way that it also realizes $\mathcal{G}_{\pi}(\widetilde{m})$ and satisfies $\mathcal{S}'(\widetilde{Z})$. Thus, the probability to realize an intermediate output tuple that results in $\mathcal{G}_{\mathsf{P}}(\widetilde{m})$ and also satisfies $\mathcal{S}'(\widetilde{Z})$ is bounded by at most $(2^n - q)_{t-r}/(2^n - q)_{t+1_0}$. We summarize this discussion in the following result which extends Lemma A.0.4. We remark that a similar result has already been proved in [28].

**Lemma A.0.6.** *For any structure graph $G$ and additional system of equations $\mathcal{S}'(\widetilde{Z})$, we have*

$$\Pr_{\mathsf{P}}\left[\mathcal{G}_{\mathsf{P}}(\widetilde{m}) = G \wedge \mathcal{S}'(\widetilde{Z}) \textit{ is satisfied}\right] \leq \frac{1}{(2^n - q - \sigma + r)^{r+1_0}},$$

*where $r$ denotes the rank of $\mathcal{S}'(\widetilde{Z}) \cup \mathcal{S}(\mathcal{G}_{\mathsf{P}}(\widetilde{m}))$ and $1_0$ is an indicator variable that results in 1 if $\mathcal{G}_{\mathsf{P}}(\widetilde{m})$ has a zero collision, and 0 otherwise.*

*Proof.* The result follows from the preceding discussion by using the fact that $t < \sigma$. □

## A.1 Bounding $\texttt{BadX}|\neg(\texttt{BadT} \vee \texttt{unman} \vee \texttt{BadW})$

Let $\texttt{TUW} := \neg(\texttt{BadT} \vee \texttt{unman} \vee \texttt{BadW})$, and $\texttt{Fi} = \neg(\texttt{TUW} \vee \texttt{BadX1} \vee \cdots \vee \texttt{BadXi})$. We have

$$\begin{aligned}\Pr[\texttt{BadX}|\texttt{TUW}] \leq \Pr[\texttt{BadX1}|\texttt{TUW}] + \Pr[\texttt{BadX2}|\texttt{F1}] + \Pr[\texttt{BadX3}|\texttt{F2}] + \Pr[\texttt{BadX4}|\texttt{F3}] \\ + \Pr[\texttt{BadX5}|\texttt{F4}] + \Pr[\texttt{BadX6}|\texttt{F5}] + \Pr[\texttt{BadX7}|\texttt{F6}] \qquad \text{(A.1)}\end{aligned}$$

We bound the individual terms on the right hand side as follows:

*Bounding* $\Pr[\texttt{BadX1}|\texttt{TUW}]$*:* Fix some $i, j, k \in [q]$. Since $\neg\texttt{unman}$ holds, we must have $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) \leq 2$. Accordingly, we have the following three cases:

(A) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 2$: This can be bounded by using exactly the same argument as used in Case A for $\texttt{BadW3}|\texttt{E2}$. So, we have

$$\Pr[\texttt{BadX1} \wedge \text{Case A}|\texttt{TUW}] \leq \frac{q^3\ell^5}{2(2^n - q - \sigma + 3)^3}.$$

(B) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 1$: Suppose $(\alpha_1, \beta_1)$ and $(\alpha_2, \beta_2)$ are collision source leading to the accident. Assuming $a < \ell_i - 1$ and $0 < b < \ell_k - 1$, we have the following system of equations

$$\begin{aligned}\mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1] \\ \mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a] \\ \mathsf{Z}_k[b-1] &= \mathsf{M}_i[a+1] \oplus \mathsf{M}_k[b] \oplus \mathsf{T}_j\end{aligned}$$

Since the three equations involve an odd number of Z variables and $\neg\texttt{BadW1}$ holds, we can straightaway conclude that the system has rank 3. Note that it is true irrespective of our choice of $a$, and $k$ as long as $b > 0$. So, we have

$$\Pr[\texttt{BadX1} \wedge \text{Case B} \wedge b > 0|\texttt{TUW}] \leq \frac{q^3\ell^4}{2(2^n - q - \sigma + 3)^3}.$$

If $b = 0$. The last equation simply boils down to the condition $\mathsf{T}_j = \mathsf{M}_i[a+1]$. This gives a unique choice of $j$, once we fix $i, a$. Further, there are $q$ choices for $k$, 3 choices for $(\alpha_1, \alpha_2)$, and $\ell^2$ choices for $(\beta_1, \beta_2)$. So, we get

$$\Pr[\texttt{BadX1} \wedge \text{Case B} \wedge b = 0|\texttt{TUW}] \leq \frac{q^2\ell^3}{2(2^n - q - \sigma + 2)^2}.$$

(C) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 0$: Assuming $a < \ell_i - 1$ and $0 < b < \ell_k - 1$, we have the following system of equations

$$\mathsf{Z}_j[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] = \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a]$$
$$\mathsf{Z}_k[b-1] = \mathsf{M}_i[a+1] \oplus \mathsf{M}_k[b] \oplus \mathsf{T}_j$$

Using the same argument as applied in case B above, we get

$$\Pr\left[\texttt{BadX1} \wedge \text{Case C} | \texttt{TUW}\right] \leq \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2}.$$

For $b = 0$, using the previous argument, we get

$$\Pr\left[\texttt{BadX1} \wedge \text{Case B} \wedge b = 0 | \texttt{TUW}\right] \leq \frac{\sigma}{2^n - q - \sigma + 1}.$$

By combining all three cases, we have

$$\Pr\left[\texttt{BadX1} | \texttt{TUW}\right] \leq \frac{2\sigma}{2^n} + \frac{q^3\ell^5}{(2^n - q - \sigma + 3)^3} + \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2} + \frac{q^2\ell^3}{2(2^n - q - \sigma + 2)^2}. \tag{A.2}$$

*Bounding* $\Pr\left[\texttt{BadX2} | \texttt{F1}\right]$: First note that once we fix $i$ and $k$, $a = \ell_i - \ell_k$, and $\mathsf{T}_j = \mathsf{M}_k[1] \oplus \mathsf{M}_i[a+1]$ which in combination with $\neg\texttt{BadW2}$ gives a unique choice for $j$. So we have at most $q^2/2$ choices for $(i, j, k, a)$. Now, considering $\mathsf{W}_i[a] = \mathsf{W}_j[\ell_j]$, we have the equation

$$\mathsf{Z}_j[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] = \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a]$$

Using Lemma A.0.6, we get

$$\Pr\left[\texttt{BadX2} | \texttt{F1}\right] \leq \frac{q^2}{2(2^n - q - \sigma + 1)}. \tag{A.3}$$

*Bounding* $\Pr\left[\texttt{BadX3} | \texttt{F2}\right]$: Recalling the counting argument of previous case, we know that $j$ is fixed once we choose $(i, k, a)$. Now, we can have three cases:

(A) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k, \mathsf{M}_l)) \geq 2$: We consider the system, consisting of any two accident equations and $\mathsf{W}_i[a] = \mathsf{W}_j[\ell_j]$. Using similar argument as used in Case A for

BadW3|E2, and the counting argument, we get

$$\Pr\left[\texttt{BadX3} \wedge \text{Case A}|\texttt{F2}\right] \leq \frac{3q^3\ell^5}{(2^n - q - \sigma + 3)^3}.$$

(B) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k, \mathsf{M}_l)) = 1$: Suppose $(\alpha_1, \beta_1)$ and $(\alpha_2, \beta_2)$ are collision source leading to the accident. We have the following system of equations

$$\begin{aligned}
\mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1] \\
\mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a] \\
\mathsf{Z}_k[b-1] \oplus \mu_{\delta_{\mathsf{M}_l}} \odot \mathsf{L} \oplus \mathsf{Z}_l[\ell_l - 1] &= \overline{\mathsf{M}}_l[\ell_l] \oplus \mathsf{M}_k[b]
\end{aligned}$$

We can have two sub-cases:

(B.1) Suppose the third equation is a consequence of the second equation. Using a similar line of arguments as used in case B.1 of BadW5|E4, we can conclude that $a$ and $b$ have some fixed choices. In particular, $a$ has at most $18$ choices. Then, using Lemma A.0.6, we have

$$\Pr\left[\texttt{BadX3} \wedge \text{Case B.1}|\texttt{F2}\right] \leq \frac{3q^3\ell^2}{(2^n - q - \sigma + 2)^2}.$$

(B.2) The last two equations are independent. Then, we can simply consider these two equations and ignore the accident equation. Using Lemma A.0.6, we have

$$\Pr\left[\texttt{BadX3} \wedge \text{Case B.2}|\texttt{F2}\right] \leq \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2}.$$

(C) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 0$: We have the following system of equations

$$\begin{aligned}
\mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a] \\
\mathsf{Z}_k[b-1] \oplus \mu_{\delta_{\mathsf{M}_l}} \odot \mathsf{L} \oplus \mathsf{Z}_l[\ell_l - 1] &= \overline{\mathsf{M}}_l[\ell_l] \oplus \mathsf{M}_k[b]
\end{aligned}$$

Let $r$ denote the rank of the above system. If $r = 1$, then we must have $\mu_{\delta_{\mathsf{M}_j}} = \mu_{\delta_{\mathsf{M}_l}}$ and $\overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a] \oplus \overline{\mathsf{M}}_l[\ell_l] \oplus \mathsf{M}_k[b] = 0^n$. Now, we can have two sub-cases:

(a) $\mathsf{M}_i[1, \ldots, a-1] = \mathsf{M}_k[1, \ldots, b-1]$ and $\mathsf{M}_j[1, \ldots, \ell_j - 1] = \mathsf{M}_l[1, \ldots, \ell_l - 1]$.

(b) $\mathsf{M}_i[1, \ldots, a-1] = \mathsf{M}_l[1, \ldots, \ell_l - 1]$ and $\mathsf{M}_j[1, \ldots, \ell_j - 1] = \mathsf{M}_k[1, \ldots, b-1]$.

In both cases, we consider the first equation for probability calculation. The two cases are similar. So we only handle the second case. The key idea here is to note that the $i$-th message shares $a-1$ block prefix with $l$-th message and similarly $k$-th message shares $b-1$ block prefix with $j$-th message.

Let $\mathcal{S} = \{\mathsf{M}_i[1,\ldots,j] : (i,j) \in [q] \times [\ell_i]\} \cup \{\perp\}$ be the set of all prefixes over all $q$ messages. For any $x \in \mathcal{S}$, we define $\mathsf{pred}(x) = x'$, where $x = x'\|y$ for some $y \in \mathbb{B}$. Consider a rooted tree $\mathcal{T}$ with vertices from $\mathcal{S}$ where $\perp$ acts as the root of the tree and there's a directed edge $(\mathsf{pred}(x), x)$ for each $x \in \mathcal{S} \setminus \{\perp\}$. Clearly, it is a directed tree rooted at $\perp$ and all leave nodes are the $1 \le i \le q$ messages $\mathsf{M}_i$. Except the leaves, all nodes have out-degree (denoted $d_x$ for the node $x$) at least one. A node is called *fork node* if it has out-degree at least two. Let $\mathsf{Fork}$ denote the set of all fork nodes. Now, $\sum_{x\in\mathcal{S}} d_x = \sigma - 1$. Then, we make the following claim.

*Claim* A.1.1. $\sum_{x\in\mathsf{Fork}}(d_x - 1) = q - 1$, and $|\mathsf{Fork}| \le q - 1$.

*Proof.* All edges at the node $x$ introduce exactly new $d_x - 1$ messages. So the first part is done. The second part also follows from the same argument. A new message is introduced only when we have a forking. So the number of messages should be at least the number of forking points. □

Now let us come back to our problem. We have to choose two prefixes $u$ and $v$ of length $a-1$ and $b-1$ blocks, respectively. Now, clearly $u$ and $v$ must be fork nodes in $\mathcal{T}$, as each of them have out-degree at least 2. We know that the number of fork nodes can be at most $q-1$ and for every such choices the probability that first equation holds is at most $1/(2^n - q - \sigma + 1)$. So, total probability for having this types of equation is at most

$$\sum_{u,v\in\mathsf{Fork}} \frac{d_u d_v}{2^n - q - \sigma + 1} \le \frac{1}{2^n - q - \sigma + 1}\left[\sum_{u\in\mathsf{Fork}}(d_u - 1)\right]^2 \le \frac{q^2}{2^n - q - \sigma + 1}.$$

So, combining the bound for the two sub-cases, we get

$$\Pr\left[\mathtt{BadX3} \wedge \text{Case C} \wedge r = 1 | \mathtt{F2}\right] \le \frac{2q^2}{2^n - q - \sigma + 1}.$$

If $r = 2$, we get

$$\Pr\left[\mathtt{BadX3} \wedge \text{Case C} \wedge r = 2 | \mathtt{F2}\right] \le \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2}.$$

Combining all three cases, we get

$$\Pr\left[\mathtt{BadX3}|\mathtt{F2}\right] \le \frac{3q^3\ell^5}{(2^n - q - \sigma + 3)^3} + \frac{20q^3\ell^2}{6(2^n - q - \sigma + 2)^2} + \frac{2q^2}{2^n - q - \sigma + 1}. \tag{A.4}$$

*Bounding* $\Pr\left[\mathtt{BadX4}|\mathtt{F3}\right]$*:* $(i,a),(j,b) \in \widetilde{\mathsf{FCI}}$ if and only if there exists $k,l \in [q]$, such that $\mathsf{W}_i[a] = \mathsf{W}_k[\ell_k]$ and $\mathsf{W}_j[b] = \mathsf{W}_l[\ell_l]$. We first note that fixing $(i,a)$, $(j,b)$, and anyone of $k$

and $l$ fixed the remaining index, since $\mathsf{T}_k \oplus \mathsf{T}_l \oplus \mathsf{M}_i[a] \oplus \mathsf{M}_j[b]$. So we can have at most $q^3\ell^2/6$ choices for $(i, j, k, l, a, b)$. As in the case of BadX3|¬F2, we can have three cases:

(A) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k, \mathsf{M}_l)) \geq 2$: Using similar argument as used in Case A for BadX3|F2, and the counting argument, we get

$$\Pr\left[\texttt{BadX4} \wedge \text{Case A} | \texttt{F3}\right] \leq \frac{5q^3\ell^6}{(2^n - q - \sigma + 3)^3}.$$

(B) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k, \mathsf{M}_l)) = 1$: Suppose $(\alpha_1, \beta_1)$ and $(\alpha_2, \beta_2)$ are collision source leading to the accident. We have the following system of equations

$$\begin{aligned}
\mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1] \\
\mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_k}} \odot \mathsf{L} \oplus \mathsf{Z}_k[\ell_k - 1] &= \overline{\mathsf{M}}_k[\ell_k] \oplus \mathsf{M}_i[a] \\
\mathsf{Z}_j[b-1] \oplus \mu_{\delta_{\mathsf{M}_l}} \odot \mathsf{L} \oplus \mathsf{Z}_l[\ell_l - 1] &= \overline{\mathsf{M}}_l[\ell_l] \oplus \mathsf{M}_j[b]
\end{aligned}$$

We can have two sub-cases:

(B.1) Suppose the third equation is a consequence of the second equation. Using prefix and suffix backtracing arguments, we can conclude that $a$, $b$, $\beta_1$ and $\beta_2$ can be chosen in at most $18\ell^2$ ways, once we fix $\alpha_1$ and $\alpha_2$. Then, using Lemma A.0.6, we have

$$\Pr\left[\texttt{BadX4} \wedge \text{Case B.1} | \texttt{F3}\right] \leq \frac{3q^3\ell^2}{(2^n - q - \sigma + 2)^2}.$$

(B.2) The last two equations are independent. Then, we can simply consider these two equations and ignore the accident equation. Using Lemma A.0.6, we have

$$\Pr\left[\texttt{BadX4} \wedge \text{Case B.2} | \texttt{F3}\right] \leq \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2}.$$

(C) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 0$: We have the following system of equations

$$\begin{aligned}
\mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a] \\
\mathsf{Z}_k[b-1] \oplus \mu_{\delta_{\mathsf{M}_l}} \odot \mathsf{L} \oplus \mathsf{Z}_l[\ell_l - 1] &= \overline{\mathsf{M}}_l[\ell_l] \oplus \mathsf{M}_k[b]
\end{aligned}$$

Using similar arguments as in case C of BadX3|F2, we get

$$\Pr\left[\texttt{BadX4} \wedge \text{Case C} \wedge r = 1 | \texttt{F3}\right] \leq \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2} + \frac{2q^2}{2^n - q - \sigma + 1}.$$

Combining all three cases, we get

$$\Pr\left[\texttt{BadX4}|\texttt{F3}\right] \leq \frac{5q^3\ell^6}{(2^n-q-\sigma+3)^3} + \frac{20q^3\ell^2}{6(2^n-q-\sigma+2)^2} + \frac{2q^2}{2^n-q-\sigma+1}. \tag{A.5}$$

*Bounding* $\Pr\left[\texttt{BadX5}|\texttt{F4}\right]$*:* Similar counting argument as used in previous cases apply here as well, i.e., index $j$ is fixed once we choose $(i,k,a)$. First, we handle the corner case, when $b=0$. In this case we get the system of equations

$$\begin{aligned}\mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j-1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a] \\ \mathsf{Z}_k[p+1] &= \mathsf{M}_i[a+2+p] \oplus \star\end{aligned}$$

where $\star = 0^n$ if $a+2+p \neq \ell_i$, and $\mu_{\delta_{\mathsf{M}_i}} \odot \mathsf{L}$ otherwise. In both the cases, the two equations are independent. So we get

$$\Pr\left[\texttt{BadX5} \wedge b=0|\texttt{F4}\right] \leq \frac{q^2\ell}{2(2^n-q-\sigma+2)^2}.$$

Assuming $b>0$, we can have three cases:

(A) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i,\mathsf{M}_j,\mathsf{M}_k,\mathsf{M}_l)) \geq 2$: Using similar argument as used in Case A for $\texttt{BadX3}|\texttt{F2}$, and the counting argument, we get

$$\Pr\left[\texttt{BadX5} \wedge \text{Case A}|\texttt{F4}\right] \leq \frac{3q^3\ell^5}{(2^n-q-\sigma+3)^3}.$$

(B) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i,\mathsf{M}_j,\mathsf{M}_k,\mathsf{M}_l)) = 1$: Suppose $(\alpha_1,\beta_1)$ and $(\alpha_2,\beta_2)$ are collision source leading to the accident. Assuming $a+2+p<\ell_i$ and $b<\ell_l$, we have the following system of equations

$$\begin{aligned}\mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1+1] \oplus \mathsf{M}_{\alpha_2}[\beta_2+1] \\ \mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j-1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a] \\ \mathsf{Z}_k[p+1] \oplus \mathsf{Z}_l[b-1] &= \mathsf{M}_l[b] \oplus \mathsf{M}_i[a+2+p]\end{aligned}$$

Now, we can have two cases:

B.1 Third equation is a consequence of the first equation. Then, using the previously used prefix and suffix backtracing arguments, we can conclude that $\beta_1$ and $\beta_2$ have fixed choices. In particular, we have at most 3 choices for

$(\beta_1, \beta_2)$. So, using Lemma A.0.6, we get

$$\Pr\left[\texttt{BadX5} \wedge \text{Case B.1} | \texttt{F4}\right] \leq \frac{3q^3\ell^2}{(2^n - q - \sigma + 2)^2}.$$

B.2 Third equation is independent of the first equation. Then, we simply consider the second and third equation, which are obviously independent. So, using Lemma A.0.6, we get

$$\Pr\left[\texttt{BadX5} \wedge \text{Case B.2} | \texttt{F4}\right] \leq \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2}.$$

Now, assume $a = \ell_i - p - 2$ and consider the two equations:

$$\begin{aligned} \mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1] \\ \mathsf{Z}_i[\ell_i - p - 3] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[\ell_i - p - 2] \end{aligned}$$

The two equations are obviously independent due to the presence of $\mathsf{L}$. So we get

$$\Pr\left[\texttt{BadX5} \wedge \text{Case B} \wedge a = \ell_i - p - 2 | \texttt{F4}\right] \leq \frac{q^3\ell^2}{(2^n - q - \sigma + 2)^2}.$$

The case where $a < \ell_i = p - 2$ and $b = \ell_l$ is similarly bounded.

(C) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 0$: Assuming $a + 2 + p < \ell_i$ and $b < \ell_l$, we have the following system of equations

$$\begin{aligned} \mathsf{Z}_i[a - 1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a] \\ \mathsf{Z}_k[p + 1] \oplus \mathsf{Z}_l[b - 1] &= \mathsf{M}_l[b] \oplus \mathsf{M}_i[a + 2 + p] \end{aligned}$$

The two equations are clearly independent due to the presence of $\mathsf{L}$, whence we have

$$\Pr\left[\texttt{BadX5} \wedge \text{Case C} \wedge a < \ell_i - p - 2 \wedge b < \ell_l | \texttt{F4}\right] \leq \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2}.$$

Suppose $a = \ell_i - p - 2$. Then we can simply consider the first equation, whence we get

$$\Pr\left[\texttt{BadX5} \wedge \text{Case C} \wedge a = \ell_i - p - 2 | \texttt{F4}\right] \leq \frac{q^2}{2(2^n - q - \sigma + 1)}.$$

Finally assume $a < \ell_i - p - 2$ and $b = \ell_l$. Then, we have the following system of equations

$$\begin{aligned}\mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a] \\ \mathsf{Z}_k[p+1] \oplus \mu_{\delta_{\mathsf{M}_l}} \odot \mathsf{L} \oplus \mathsf{Z}_l[\ell_l - 1] &= \overline{\mathsf{M}}_l[\ell_l] \oplus \mathsf{M}_i[a+2+p]\end{aligned}$$

Using similar argument as in case C of BadX3|F2, we get

$$\Pr\left[\texttt{BadX5} \wedge \text{Case C} \wedge a < \ell_i - p - 2 \wedge b = \ell_l | \texttt{F4}\right] \leq \frac{2q^2}{2^n - q - \sigma + 1} + \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2}.$$

Combining all three cases, we get

$$\Pr\left[\texttt{BadX5}|\texttt{F4}\right] \leq \frac{3q^3\ell^5}{(2^n - q - \sigma + 3)^3} + \frac{27q^3\ell^2}{6(2^n - q - \sigma + 2)^2} + \frac{3q^2}{(2^n - q - \sigma + 1)}. \tag{A.6}$$

*Bounding* $\Pr\left[\texttt{BadX6}|\texttt{F5}\right]$: $(i,a) \in \widetilde{\mathsf{FCI}}$ and $(j,b) \in \widetilde{\mathsf{ICI}}$ if and only if there exists $i', k, l \in [q]$ and $c \in [\ell_k - 1]$, such that $\mathsf{W}_i[a] = \mathsf{W}_{i'}[\ell_{i'}]$, $\mathsf{W}_j[b-c] = \mathsf{W}_l[\ell_l]$ and $\mathsf{X}_j[b-c+1] = \mathsf{M}_k[1]$. We first note that fixing $(i,a)$, $j$, $k$, $b-c$, and $i'$ fixes $b$, $c$, and $l$. So we can have at most $q^4\ell^2/12$ choices for $(i,j,k,i',l,a,b,c)$. As in the case of BadX3|¬F2, we can have three cases:

(A) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k, \mathsf{M}_l, \mathsf{M}_{i'})) \geq 2$: In this case we consider the two accident equations along with $\mathsf{W}_i[a] = \mathsf{W}_{i'}[\ell_{i'}]$ and $\mathsf{X}_i[a+1] = \mathsf{X}_j[b+1]$. We claim that all four equations are independent due to odd number of $\mathsf{Z}$ variables (last equation is univariate in $\mathsf{Z}_k[c]$). Then, we get

$$\Pr\left[\texttt{BadX6} \wedge \text{Case A}|\texttt{F5}\right] \leq \frac{4q^4\ell^6}{(2^n - q - \sigma + 4)^4}.$$

(B) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k, \mathsf{M}_l, \mathsf{M}_{i'})) = 1$: Suppose $(\alpha_1, \beta_1)$ and $(\alpha_2, \beta_2)$ are collision source leading to the accident. We have the following system of equations

$$\begin{aligned}\mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1] \\ \mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_{i'}}} \odot \mathsf{L} \oplus \mathsf{Z}_{i'}[\ell_{i'} - 1] &= \overline{\mathsf{M}}_{i'}[\ell_{i'}] \oplus \mathsf{M}_i[a] \\ \mathsf{Z}_j[b-c-1] \oplus \mu_{\delta_{\mathsf{M}_l}} \odot \mathsf{L} \oplus \mathsf{Z}_l[\ell_l - 1] &= \overline{\mathsf{M}}_l[\ell_l] \oplus \mathsf{M}_j[b-c] \\ \mathsf{Z}_k[c] &= \mathsf{M}_j[b+1] \oplus \mathsf{M}_i[a+1] \oplus \mathsf{T}_{i'}\end{aligned}$$

Again concentrating on whether the second and third equations are independent or not, and using similar argumentation as before, we get

$$\Pr\left[\texttt{BadX6} \wedge \text{Case B} | \texttt{F5}\right] \leq \frac{5q^4\ell^3}{(2^n - q - \sigma + 2)^3}.$$

(C) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 0$: We have the following system of equations

$$\begin{aligned}
\mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_{i'}}} \odot \mathsf{L} \oplus \mathsf{Z}_{i'}[\ell_{i'} - 1] &= \overline{\mathsf{M}}_{i'}[\ell_{i'}] \oplus \mathsf{M}_i[a] \\
\mathsf{Z}_j[b-c-1] \oplus \mu_{\delta_{\mathsf{M}_l}} \odot \mathsf{L} \oplus \mathsf{Z}_l[\ell_l - 1] &= \overline{\mathsf{M}}_l[\ell_l] \oplus \mathsf{M}_j[b-c] \\
\mathsf{Z}_k[c] &= \mathsf{M}_j[b+1] \oplus \mathsf{M}_i[a+1] \oplus \mathsf{T}_{i'}
\end{aligned}$$

Using similar arguments as in previous cases, we get

$$\Pr\left[\texttt{BadX6} \wedge \text{Case C} | \texttt{F5}\right] \leq \frac{q^4\ell^2}{4(2^n - q - \sigma + 3)^3} + \frac{2q^3\ell}{(2^n - q - \sigma + 2)^2}.$$

Combining all three cases, we get

$$\Pr\left[\texttt{BadX6} | \texttt{F5}\right] \leq \frac{4q^4\ell^6}{(2^n - q - \sigma + 4)^4} + \frac{6q^4\ell^3}{(2^n - q - \sigma + 3)^3} + \frac{2q^3\ell}{(2^n - q - \sigma + 2)^2}. \tag{A.7}$$

*Bounding* $\Pr\left[\texttt{BadX7} | \texttt{F6}\right]$: $(i, a), (j, b) \in \widetilde{\mathsf{ICI}}$ if and only if there exists $i', j', k, l \in [q]$, $c \in [\ell_k]$ and $d \in [\ell_l]$, such that $\mathsf{W}_i[a] = \mathsf{W}_{i'}[\ell_{i'}]$, $\mathsf{W}_j[b-c] = \mathsf{W}_{j'}[\ell'_j]$, $\mathsf{X}_i[a-c+1] = \mathsf{M}_k[1]$ and $\mathsf{X}_j[b-d+1] = \mathsf{M}_l[1]$. We first note that fixing $i$,, $j$, $k$, $l$, $a-c$, and $b-d$, fixes $a$, $b$, $c$, $d$, $i'$ and $j'$. So we can have at most $q^4\ell^2/12$ choices for $(i, j, k, l, i', j', a, b, c, d)$. As in the case of $\texttt{BadX3}|\neg\texttt{F2}$, we can have three cases:

(A) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k, \mathsf{M}_l, \mathsf{M}_{i'}, \mathsf{M}_{j'})) \geq 2$: In this case we consider the two accident equations along with $\mathsf{W}_i[a] = \mathsf{W}_{i'}[\ell_{i'}]$, $\mathsf{W}_j[b] = \mathsf{W}_{j'}[\ell_{j'}]$. As in the previous cases, we conclude that all four equations are independent. Then, we get

$$\Pr\left[\texttt{BadX7} \wedge \text{Case A} | \texttt{F6}\right] \leq \frac{4q^4\ell^6}{(2^n - q - \sigma + 4)^4}.$$

(B) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k, \mathsf{M}_l, \mathsf{M}_{i'})) = 1$: Suppose $(\alpha_1, \beta_1)$ and $(\alpha_2, \beta_2)$ are collision source leading to the accident. We have the following system of equations

$$\begin{aligned}
\mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1] \\
\mathsf{Z}_i[a-c-1] \oplus \mu_{\delta_{\mathsf{M}_{i'}}} \odot \mathsf{L} \oplus \mathsf{Z}_i[\ell_{i'} - 1] &= \overline{\mathsf{M}}_{i'}[\ell_{i'}] \oplus \mathsf{M}_i[a-c] \\
\mathsf{Z}_j[b-d-1] \oplus \mu_{\delta_{\mathsf{M}_{j'}}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_{j'} - 1] &= \overline{\mathsf{M}}_{j'}[\ell_{j'}] \oplus \mathsf{M}_j[b-d]
\end{aligned}$$

$$\mathsf{Z}_k[c] \oplus \mathsf{Z}_l[d] = \mathsf{M}_j[b+1] \oplus \mathsf{M}_i[a+1] \oplus \mathsf{T}_{i'} \oplus \mathsf{T}_{j'}$$

Again concentrating on whether the second and third equations are independent or not, and using similar argumentation as before, we get

$$\Pr\left[\texttt{BadX7} \wedge \text{Case B} | \texttt{F6}\right] \leq \frac{6q^4\ell^2}{(2^n - q - \sigma + 2)^3}.$$

(C) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 0$: We have the following system of equations

$$\begin{aligned}
\mathsf{Z}_i[a-c-1] \oplus \mu_{\delta_{\mathsf{M}_{i'}}} \odot \mathsf{L} \oplus \mathsf{Z}_{i'}[\ell_{i'} - 1] &= \overline{\mathsf{M}}_{i'}[\ell_{i'}] \oplus \mathsf{M}_i[a-c] \\
\mathsf{Z}_j[b-d-1] \oplus \mu_{\delta_{\mathsf{M}_{j'}}} \odot \mathsf{L} \oplus \mathsf{Z}_l[\ell_{j'} - 1] &= \overline{\mathsf{M}}_{j'}[\ell_{j'}] \oplus \mathsf{M}_j[b-d] \\
\mathsf{Z}_k[c] \oplus \mathsf{Z}_l[d] &= \mathsf{M}_j[b+1] \oplus \mathsf{M}_i[a+1] \oplus \mathsf{T}_{i'} \oplus \mathsf{T}_{j'}
\end{aligned}$$

Using similar arguments as in previous cases, we get

$$\Pr\left[\texttt{BadX7} \wedge \text{Case C} | \texttt{F6}\right] \leq \frac{q^4\ell^2}{4(2^n - q - \sigma + 3)^3} + \frac{2q^3}{(2^n - q - \sigma + 2)^2}.$$

Combining all three cases, we get

$$\Pr\left[\texttt{BadX7} | \texttt{F6}\right] \leq \frac{4q^4\ell^6}{(2^n - q - \sigma + 4)^4} + \frac{7q^4\ell^2}{(2^n - q - \sigma + 3)^3} + \frac{2q^3}{(2^n - q - \sigma + 2)^2}. \tag{A.8}$$

Finally, accumulating all the bounds from Eq. (A.2)-(A.8), and assuming $\ell < q$, we get Eq. (6.23), i.e.,

$$\begin{aligned}
\Pr\left[\texttt{BadX} | \texttt{TUW}\right] \leq \frac{2\sigma}{2^n} &+ \frac{10q^2}{2^n - q - \sigma + 1} + \frac{15q^3\ell^2 + q^2\ell^3}{(2^n - q - \sigma + 2)^2} \\
&+ \frac{12q^3\ell^6 + 6q^4\ell^3}{(2^n - q - \sigma + 3)^3} + \frac{8q^4\ell^6}{(2^n - q - \sigma + 4)^4}
\end{aligned}$$

# Bibliography


[1] ISO/IEC JTC 1/SC 27. Information technology — lightweight cryptography — part 6: Message authentication codes (MACs). ISO/IEC 29192-6, International Organization for Standardization, 2019.

[2] Elena Andreeva, Andrey Bogdanov, Nilanjan Datta, Atul Luykx, Bart Mennink, Mridul Nandi, Elmar Tischhauser, and Kan Yasuda. COLM. Specification document v1, CAESAR Submission, 2016. Online: `https://competitions.cr.yp.to/round3/colmv1.pdf` (Accessed: 02 September, 2019).

[3] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. The secure real-time transport protocol (SRTP). RFC 3711, IETF, 2004.

[4] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *Advances in Cryptology - EUROCRYPT 2006*, pages 409–426, 2006.

[5] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of cipher block chaining. In *Advances in Cryptology – CRYPTO '94, Proceedings*, pages 341–358, 1994.

[6] Mihir Bellare, Roch Guérin, and Phillip Rogaway. XOR macs: New methods for message authentication using finite pseudorandom functions. In *Advances in Cryptology – CRYPTO '95, Proceedings*, pages 15–28, 1995.

[7] Mihir Bellare, Oded Goldreich, and Hugo Krawczyk. Stateless evaluation of pseudorandom functions: Security beyond the birthday barrier. In *Advances in Cryptology – CRYPTO '99, Proceedings*, pages 270–287, 1999.

[8] Mihir Bellare, Oded Goldreich, and Anton Mityagin. The power of verification queries in message authentication and authenticated encryption. *IACR Cryptol. ePrint Arch.*, 2004:309, 2004.

[9] Mihir Bellare, Krzysztof Pietrzak, and Phillip Rogaway. Improved security analyses for CBC macs. In *Advances in Cryptology - CRYPTO 2005, Proceedings*, pages 527–545, 2005.

[10] A. Berendschot, B. den Boer, J. Boly, A. Bosselaers, J. Brandt, D. Chaum, I. Damgård, M. Dichtl, W. Fumy, M. van der Ham, C. Jansen, P. Landrock, B. Preneel, G. Roelofsen, P. de Rooij, and J Vandewalle. Final Report of RACE Integrity Primitives. *Lecture Notes in Computer Science, Springer-Verlag, 1995*, 1007, 1995.

[11] D. J. Bernstein. A Short Proof of the Unpredictability of Cipher Block Chaining, 2005. URL `http://cr.yp.to/antiforgery/easycbc-20050109.pdf`.

[12] Daniel J. Bernstein. How to stretch random functions: The security of protected counter sums. *J. Cryptology*, 12(3):185–192, 1999.

[13] Garrett Birkhoff and Saunders Mac Lane. *A Survey of Modern Algebra*. Universities Press, 1965.

[14] John Black and Martin Cochran. MAC reforgeability. In *Fast Software Encryption, FSE 2009*, pages 345–362, 2009.

[15] John Black and Phillip Rogaway. CBC macs for arbitrary-length messages: The three-key constructions. In *Advances in Cryptology - CRYPTO 2000, Proceedings*, pages 197–215, 2000.

[16] John Black and Phillip Rogaway. A block-cipher mode of operation for parallelizable message authentication. In *Advances in Cryptology - EUROCRYPT 2002, Proceedings*, pages 384–397, 2002.

[17] John Black and Phillip Rogaway. CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions. *J. Cryptology*, 18(2):111–131, 2005.

[18] John Black, Shai Halevi, Hugo Krawczyk, Ted Krovetz, and Phillip Rogaway. UMAC: Fast and Secure Message Authentication. In *Advances in Cryptology - CRYPTO '99*, pages 216–233, 1999.

[19] Dan Boneh and Victor Shoup. *A Graduate Course in Applied Cryptography*. online: `https://toc.cryptobook.us/`, v0.5 edition, 2020.

[20] CAESAR. Competition for authenticated encryption: Security, applicability and robustness. Online webpage, 2014. URL `http://competitions.cr.yp.to/caesar.html`.

[21] Larry Carter and Mark N. Wegman. Universal classes of hash functions. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.

[22] Bishwajit Chakraborty, Soumya Chattopadhyay, Ashwin Jha, and Mridul Nandi. On length independent security bounds for the PMAC family. *IACR Trans. Symmetric Cryptol.*, 2021(2):423–445, 2021.

[23] Soumya Chattopadhyay, Ashwin Jha, and Mridul Nandi. Fine-tuning the ISO/IEC Standard Lightmac. In *Advances in Cryptology - ASIACRYPT 2021, Proceedings*, pages 490–519, 2021.

[24] Soumya Chattopadhyay, Ashwin Jha, and Mridul Nandi. Towards tight security bounds for omac, xcbc and tmac. Springer-Verlag, 2022.

[25] Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In *Advances in Cryptology - EUROCRYPT 2014, Proceedings*, pages 327–350, 2014.

[26] Benoît Cogliati and Yannick Seurin. EWCDM: an efficient, beyond-birthday secure, nonce-misuse resistant MAC. In *Advances in Cryptology – CRYPTO '16, Proceedings*, pages 121–149, 2016.

[27] Julie E. Cohen. Surveillance vs. Privacy: Effects and Implications. *Cambridge Handbook of Surveillance Law*, pages 455–469, 2017.

[28] 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.

[29] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Goutam Paul. Double-block hash-then-sum: A paradigm for constructing bbb secure prf. *IACR Trans. Symmetric Cryptol.*, 2018(3):36–92, 2018.

[30] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Kan Yasuda. Encrypt or decrypt? to make a single-key beyond birthday secure nonce-based MAC. In *Advances in Cryptology – CRYPTO '18, Proceedings, Part I*, pages 631–661, 2018.

[31] Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[32] Nir Drucker, Shay Gueron, and Vlad Krasnov. Making AES great again: the forthcoming vectorized AES instruction. In *16th International Conference on Information Technology-New Generations – ITNG 2019, Proceedings*, volume 8544, pages 37–41, 2019.

[33] David S. Dummit and Richard S. Foote. *Abstract Algebra*. Wiley-India, 3rd edition, 2003.

[34] Avijit Dutta and Mridul Nandi. BBB secure nonce based mac using public permutations. In *Progress in Cryptology - AFRICACRYPT 2020, Proceedings*, pages 172–191, 2020.

[35] Avijit Dutta, Ashwin Jha, and Mridul Nandi. A new look at counters: Don't run like marathon in a hundred meter race. *IEEE Trans. Computers*, 66(11):1851–1864, 2017.

[36] Avijit Dutta, Mridul Nandi, and Suprita Talnikar. Beyond birthday bound secure MAC in faulty nonce model. In *Advances in Cryptology – EUROCRYPT '19, Proceedings, Part I*, pages 437–466, 2019.

[37] William F. Ehrsam, Carl H. W. Meyer, John L. Smith, and Walter L. Tuchman. Message Verification and Transmission Error Detection by Block Chaining. Patent 4074066, USPTO, 1976.

[38] Marc Fischlin, Anja Lehmann, Thomas Ristenpart, Thomas Shrimpton, Martijn Stam, and Stefano Tessaro. Random oracles with(out) programmability. In *Advances in Cryptology - ASIACRYPT 2010, Proceedings*, pages 303–320, 2010.

[39] Peter Gazi, Krzysztof Pietrzak, and Michal Rybár. The exact security of PMAC. *IACR Trans. Symmetric Cryptol.*, 2016(2):145–161, 2016.

[40] Peter Gazi, Krzysztof Pietrzak, and Michal Rybár. The exact security of PMAC. *IACR Cryptology ePrint Archive*, 2017:69, 2017.

[41] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th Annual Symposium on Foundations of Computer Science - FOCS '84, Proceedings*, pages 464–479, 1984.

[42] Frank Gray. Pulse code communication. Patent 2632058, USPTO, 1953.

[43] Shai Halevi and Phillip Rogaway. A tweakable enciphering mode. In *Advances in Cryptology – CRYPTO '03, Proceedings*, pages 482–499, 2003.

[44] Shai Halevi and Phillip Rogaway. A parallelizable enciphering mode. In *Topics in Cryptology – CT-RSA '04, Proceedings*, pages 292–304, 2004.

[45] Viet Tung Hoang and Stefano Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In *Advances in Cryptology – CRYPTO '16, Proceedings, Part I*, pages 3–32, 2016.

[46] Kenneth Hoffman and Ray Kunze. *Linear Algebra*. Prentice Hall, second edition, 2015.

[47] Tetsu Iwata and Kaoru Kurosawa. OMAC: one-key CBC MAC. In *Fast Software Encryption - FSE 2003, Revised Papers*, pages 129–153, 2003.

[48] Tetsu Iwata and Kaoru Kurosawa. Stronger Security Bounds for OMAC, TMAC, and XCBC. In *Progress in Cryptology - INDOCRYPT '03, Proceedings*, pages 402–415, 2003.

[49] Éliane Jaulmes and Reynald Lercier. FRMAC, a proc. fast randomized message authentication code. *IACR Cryptology ePrint Archive*, 2004(166), 2004. Online: `https://eprint.iacr.org/2004/166.pdf` (Accessed: 02 May, 2020).

[50] Éliane Jaulmes, Antoine Joux, and Frédéric Valette. On the Security of Randomized CBC-MAC Beyond the Birthday Paradox Limit: A New Construction. In *Fast Software Encryption – FSE '02, Proceedings*, pages 237–251, 2002.

[51] Ashwin Jha. *Provable Security of Symmetric-key Cryptographic Schemes*. PhD thesis, Indian Statistical Institute, 2019.

[52] Ashwin Jha and Mridul Nandi. Revisiting structure graphs: Applications to CBC-MAC and EMAC. *J. Math. Cryptol.*, 10(3-4):157–180, 2016.

[53] Ashwin Jha and Mridul Nandi. Revisiting structure graphs: Applications to CBC-MAC and EMAC. *IACR Cryptol. ePrint Arch.*, 2016:161, 2016.

[54] Ashwin Jha and Mridul Nandi. A survey on applications of H-Technique: Revisiting security analysis of prp and prf. *IACR Cryptol. ePrint Arch.*, 2018:1130, 2018.

[55] Ashwin Jha, Avradip Mandal, and Mridul Nandi. On the exact security of message authentication using pseudorandom functions. *IACR Trans. Symmetric Cryptol.*, 2017(1):427–448, 2017.

[56] David Kahn. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. Scribner, 1996.

[57] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography: Principles and Protocols*. Chapman and Hall/CRC, 2014.

[58] Seongkwang Kim, ByeongHak Lee, and Jooyoung Lee. Tight security bounds for double-block hash-then-sum macs. In *Advances in Cryptology - EUROCRYPT 2020, Proceedings, Part I*, pages 435–465, 2020.

[59] Ted Krovetz and Phillip Rogaway. The software performance of authenticated-encryption modes. In *Fast Software Encryption – FSE '11, Revised Selected Papers*, pages 306–327, 2011.

[60] Kaoru Kurosawa and Tetsu Iwata. TMAC: two-key CBC MAC. In *Topics in Cryptology - CT-RSA 2003, Proceedings*, pages 33–49, 2003.

[61] Atul Luykx, Bart Preneel, Alan Szepieniec, and Kan Yasuda. On the influence of message length in pmac's security bounds. In *Advances in Cryptology - EUROCRYPT 2016, Proceedings, Part I*, pages 596–621, 2016.

[62] 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, Revised Selected Papers*, volume 9783, pages 43–59, 2016.

[63] David A. McGrew and John Viega. The security and performance of the galois/-counter mode (GCM) of operation. In *Progress in Cryptology - INDOCRYPT 2004, Proceedings*, pages 343–355, 2004.

[64] Bart Mennink. Towards tight security of cascaded LRW2. In *Theory of Cryptography - TCC '18. Proceedings, Part II*, pages 192–222, 2018.

[65] Bart Mennink and Samuel Neves. Encrypted davies-meyer and its dual: Towards optimal security using mirror theory. In *Advances in Cryptology - CRYPTO 2017, Proceedings, Part III*, pages 556–583, 2017.

[66] Kazuhiko Minematsu. How to thwart birthday attacks against macs via small randomness. In *Fast Software Encryption – FSE '10, Proceedings*, pages 230–249, 2010.

[67] Kazuhiko Minematsu and Toshiyasu Matsushima. New Bounds for PMAC, TMAC, and XCBC. In *Fast Software Encryption – FSE '07, Revised Selected Papers*, pages 434–451, 2007.

[68] Nicky Mouha. Chaskey: a MAC algorithm for microcontrollers - status update and proposal of Chaskey-12. *IACR Cryptol. ePrint Arch.*, 2015:1182, 2015.

[69] Yusuke Naito. Blockcipher-based macs: Beyond the birthday bound without message length. In *Advances in Cryptology - ASIACRYPT 2017, Proceedings, Part III*, pages 446–470, 2017.

[70] Yusuke Naito. The exact security of PMAC with two powering-up masks. *IACR Trans. Symmetric Cryptol.*, 2019(2):125–145, 2019.

[71] Yusuke Naito. Personal communication, 2020.

[72] Yusuke Naito. The exact security of PMAC with three powering-up masks. *IACR Cryptol. ePrint Arch.*, 2020:731, 2020.

[73] Mridul Nandi. Fast and secure cbc-type MAC algorithms. In *Fast Software Encryption, Revised Selected Papers*, pages 375–393, 2009.

[74] Mridul Nandi. Improved Security Analysis for OMAC as a Pseudorandom Function. *J. Mathematical Cryptol.*, 3(2):133–148, 2009.

[75] Mridul Nandi. A Unified Method for Improving PRF Bounds for a Class of Blockcipher Based MACs. In *Fast Software Encryption – FSE '10, Revised Selected Papers*, pages 212–229, 2010.

[76] Mridul Nandi and Avradip Mandal. Improved security analysis of PMAC. *J. Mathematical Cryptol.*, 2(2):149–162, 2008.

[77] NIST. Announcing the ADVANCED ENCRYPTION STANDARD (AES). FIPS 197, National Institute of Standards and Technology, U. S. Department of Commerce, 2001. Online: `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf` (Accessed: 02 May, 2020).

[78] NIST. Lightweight cryptography standardization project. Online webpage, 2018. URL `https://csrc.nist.gov/projects/lightweight-cryptography`.

[79] Morris Dworkin (NIST). Recommendation for block cipher modes of operation – methods and techniques. NIST Special Publication 800-38A, National Institute of Standards and Technology, U. S. Department of Commerce, 2001. Online: `https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf` (Accessed: 02 May, 2020).

[80] Morris Dworkin (NIST). Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality. NIST Special Publication 800-38C, National Institute of Standards and Technology, U. S. Department of Commerce, 2001. Online: `https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf` (Accessed: 02 May, 2020).

[81] Morris Dworkin (NIST). Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. NIST Special Publication 800-38B, National Institute of Standards and Technology, U. S. Department of Commerce, 2005. Online: `https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38b.pdf` (Accessed: 02 May, 2020).

[82] Jacques Patarin. *Etude des Générateurs de Permutations Pseudo-aléatoires Basés sur le Schéma du DES*. PhD thesis, Université de Paris, 1991.

[83] Jacques Patarin. The "Coefficients H" Technique. In *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, pages 328–345, 2008.

[84] Krzysztof Pietrzak. A Tight Bound for EMAC. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, pages 168–179, 2006.

[85] Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In *Advances in Cryptology - ASIACRYPT 2004, Proceedings*, pages 16–31, 2004.

[86] Phillip Rogaway, Mihir Bellare, John Black, and Ted Krovetz. OCB: a block-cipher mode of operation for efficient authenticated encryption. In *ACM Conference on Computer and Communications Security – CCS '01, Proceedings*, pages 196–205, 2001.

[87] Palash Sarkar. Improving upon the TET mode of operation. In *Information Security and Cryptology – ICISC '07, Proceedings*, pages 180–192, 2007.

[88] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948.

[89] Claude Elwood Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28:656–715, 1949.

[90] Yaobin Shen, Lei Wang, Dawu Gu, and Jian Weng. Revisiting the Security of DbHtS MACs: Beyond-Birthday-Bound in the Multi-user Setting. In *Advances in Cryptology - CRYPTO 2021, Proceedings*, pages 309–336, 2021.

[91] Victor Shoup. On fast and provably secure message authentication based on universal hashing. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, Proceedings*, pages 313–328, 1996.

[92] Douglas Stinson. *Cryptography: Theory and Practice*. Chapman and Hall/CRC, 3rd edition edition, 2021.

[93] Gene Tsudik. Message authentication with one-way hash functions. In *IEEE - INFOCOM 1992, Proceedings*, pages 2055–2059, 1992.

[94] Peng Wang, Dengguo Feng, and Wenling Wu. HCTR: A variable-input-length enciphering mode. In *Information Security and Cryptology – CISC '05, Proceedings*, pages 175–188, 2005.

[95] Mark N. Wegman and Larry Carter. New classes and applications of hash functions. In *Symposium on Foundations of Computer Science - FOCS 1979, Proceedings*, pages 175–182, 1979.

[96] Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.

[97] Kan Yasuda. The sum of CBC MACs is a secure prf. In Josef Pieprzyk, editor, *Topics in Cryptology – CT-RSA '10, Proceedings*, pages 366–381, 2010.

[98] Kan Yasuda. A new variant of PMAC: beyond the birthday bound. In *Advances in Cryptology – CRYPTO '11, Proceedings*, pages 596–609, 2011.

[99] Kan Yasuda. PMAC with parity: Minimizing the query-length influence. In *Topics in Cryptology – CT-RSA '12, Proceedings*, pages 203–214, 2012.

[100] Liting Zhang, Wenling Wu, Han Sui, and Peng Wang. 3kf9: Enhancing 3gpp-mac beyond the birthday bound. In *Advances in Cryptology - ASIACRYPT 2012, Proceedings*, pages 296–312, 2012.