Tight Security Analysis of the Public Permutation-Based PMAC_Plus

Avijit Dutta[1] and Mridul Nandi[2] and Suprita Talnikar[2]

TCG-CREST, India
Indian Statistical Institute, Kolkata.
avirocks.dutta13@gmail.com,mridul.nandi@gmail.com,suprita45@gmail.com

Abstract. Yasuda proposed a variable input-length PRF in CRYPTO 2011, called PMAC_Plus, based on an n-bit block cipher. PMAC_Plus is a rate-1 construction and inherits the well-known PMAC parallel network with a low additional cost. However, unlike PMAC, PMAC_Plus is secure roughly up to $2^{2n/3}$ queries. Zhang et al. proposed 3kf9 in ASIACRYPT 2012, Naito proposed LightMAC_Plus in ASIACRYPT 2017, and Iwata et al. proposed GCM-SIV2 in FSE 2017 – all of them secure up to around $2^{2n/3}$ queries. Their structural designs and corresponding security proofs were unified by Datta et al. in their framework *Double-block Hash-then-Sum* (DbHtS). Leurent et al. in CRYPTO 2018 and then Lee et al. in EUROCRYPT 2020 established a tight security bound of $2^{3n/4}$ on DbHtS. That PMAC_Plus provides security for roughly up to $2^{3n/4}$ queries is a consequence of this result. In this paper, we propose a public permutation-based variable input-length PRF called pPMAC_Plus. We show that pPMAC_Plus is secure against all adversaries that make at most $2^{2n/3}$ queries. We also show that the bound is essentially tight. It is of note here that instantiation of each block cipher of pPMAC_Plus with the two-round iterated Even-Mansour cipher can yield a beyond the birthday bound secure PRF based on public permutations. Altogether, the solution incurs $(2\ell + 4)$ permutation calls, whereas our proposal requires only $(\ell + 2)$ permutation calls, ℓ being the maximum number of message blocks.

Keywords: PMAC_Plus · Public Permutation · Sum-Capture Lemma · H-Coefficient Technique

1 Introduction

Background. A Pseudo-Random Function (PRF) is a fundamental primitive in symmetric key cryptography. It is useful in providing solutions like authentication of messages, encryption of any arbitrary-length messages, etc. Most of the PRFs are built on top of a block cipher in some mode of operation. Some of the commonly used block cipher-based PRFs are CBC-MAC [BKR00], PMAC [BR02], OMAC [IK03], LightMAC [LPTY16], etc. However, all of these block cipher-based PRF constructions provide security up to $2^{n/2}$ adversarial queries, where n is the block size of the block cipher. In cryptography, this bound is typically known as the **birthday bound**.

Birthday bound secure constructions are often acceptable in practice when they are instantiated with block ciphers having a large block size (e.g., AES-128). To justify the above statement, consider PMAC construction whose PRF advantage is roughly $5\ell q^2/2^n$ [NM08], where ℓ is the upper limit on message size in terms of the number of blocks. When it is instantiated with AES-128, it gives a security of roughly up to 2^{48} adversarial queries, provided the longest message size is 2^{16} blocks and the success probability of breaking the scheme is restricted to 2^{-10}. However, with a growing trend of designing and standardizing lightweight block ciphers (NIST light-weight competition) like PRESENT [BKL+07], GIFT [BPP+17], LED [GPPR12], etc. that are particularly suitable for a resource-constrained environment, birthday bound-secure constructions are not suitable for use in practice. For example, PMAC instantiated with the PRESENT block cipher (a 64-bit block cipher) gives security up to 2^{16} adversarial queries when the longest message size is 2^{16} blocks and the success probability of breaking the scheme is 2^{-10}. Thus, it is not safe to use birthday bound-secure PRFs when they are instantiated with lightweight block ciphers. Although using AES-128 in a birthday bound-secure mode provides 64-bit security (which is adequate for the present-day), it may not be so in the future due to technological advancement. In such a situation, the feasible option would be to use a mode that gives higher security than the usual birthday one instead of replacing the cipher with a larger block size.

Table 1: Comparison table for permutation-based PRFs and MACs. n denotes the state size of the permutation, which we also call block size. The first column denotes the number of input blocks versus the number of output blocks. i/p (resp. o/p) size denotes the bit size of the input (resp. output) to the construction. Constructions with a dagger symbol use keyed hash functions and the number of keys they require includes the hash key as well; they also take nonce as one of their inputs. Security bounds mentioned in green denote lower bounds for which a matching upper bound isn't yet proven, while blue denotes tight bounds and red denotes upper bounds.

Constructions	# of permutations	# of keys	(i/p, o/p) size	Security
SoEM1 [CLM19]	1	2	(n, n)	$\mathbf{n/2}$
SoEM21 [CLM19]	2	1	(n, n)	$\mathbf{n/2}$
SoEM22 [CLM19]	2	2	(n, n)	$\mathbf{2n/3}$
SoKAC1 [CLM19]	1	2	(n, n)	$\mathbf{2n/3}$ [CNTY20]
SoKAC21 [CLM19]	2	1	(n, n)	$\mathbf{n/2}$ [Nan20a]
pEDM [DNT21a]	1	2	(n, n)	$\mathbf{2n/3}$ [DNT21a]
PDMMAC [CNTY20]	1	1	(n, n)	$\mathbf{2n/3}$
DS-SoEM [BDLN20]	1	2	$(n-1, n)$	$\mathbf{2n/3}$
CENCPP* [BDLN20]	$w+1$	2	(n, wn)	$\mathbf{2n/3} - \log(\mathbf{w^2})$
DS-CENCPP* [BDLN20]	1	2	$(n - \log(w+1), wn)$	$\mathbf{2n/3} - \log(\mathbf{w^4})$
(†) nEHtM$_p$ [DN20a]	1	2	$(n - 1 + \ell n, n)$	$\mathbf{2n/3}$
(†) PDM*MAC [CNTY20]	1	2	$(n + \ell n, n)$	$\mathbf{2n/3}$
(†) 1K-PDM*MAC [CNTY20]	1	1	$(n + \ell n, n)$	$\mathbf{2n/3}$
Chaskey [MMVH+14]	1	1	$(\ell n, t)$	$\mathbf{n/2 + 2^{-t}}$
pPMAC_Plus [Our Construction]	1	3	$(\ell n, n)$	$\mathbf{2n/3}$

Beyond the Birthday Bound PRFs. Over the years, there have been many proposals of beyond the birthday bound secure PRFs. In [Yas10], Yasuda proposed SUM-ECBC, a beyond the birthday bound-secure PRF. SUM-ECBC is a rate-1/2 sequential mode of construction with four block cipher keys that offers about $2n/3$-bit security. Yasuda, in [Yas11], proposed another beyond the birthday bound secure PRF, called PMAC_Plus that also offers about $2n/3$-bit security. However, unlike SUM-ECBC, it is a rate-1 and parallel mode of construction with three block cipher keys. In the following year, Zhang et al. [ZWSW12] proposed another candidate for a beyond the birthday bound-secure PRF, called 3kf9, which is a rate-1 sequential mode of construction with three block cipher keys and offers $2n/3$-bit security. Following these works, Naito proposed LightMAC_Plus in [Nai17], the first beyond the birthday bound secure PRF which is proven to have an ℓ independent beyond the birthday bound and hence effectively offers a better security than that of all the earlier three proposals. Datta et al. [DDN+17] proposed a single-keyed variant of the PMAC_Plus that offers a better security bound than that of PMAC_Plus. In [DDNP18], Datta et al. unified the design of all four beyond the birthday bound secure PRFs (i.e., SUM-ECBC, PMAC_Plus, 3kf9, LightMAC_Plus) and gave a common security proof for all of them. They also proposed a two-keyed version of SUM-ECBC, PMAC_Plus, 3kf9, LightMAC_Plus and have shown that all of them achieve roughly $2n/3$-bit security. Interestingly, all these constructions share a similar structural design and offer the same level of security. All this motivated the unification of these designs and the provision of a common security proof for all of them in [DDNP18].

Double block-Hash-then-Sum. DbHtS [DDNP18] is a generic methodology for designing block cipher-based beyond the birthday bound secure PRFs. It is a composition of two constituent elements: (i) a double block hash function that outputs a $2n$-bit hash value of the input message and (ii) a sum function used in the finalization phase that generates the final tag by XORing the encryption (via two independent block ciphers) of two n-bit hash values. The authors have shown that if the cover-free advantage (refers to the probability that for a triplet of messages M_i, M_j, M_k, the first halves (i.e. the leftmost n bits) of the hash values of M_i and M_j collide and the second halves (i.e. the rightmost n bits) of the hash values of M_i and M_k collide) and the block-wise universal advantage (refers to the probability of collision of either of the halves of the hash values of any pair of distinct messages) of the underlying double-block hash function is sufficiently low, then DbHtS is secure up to $2^{2n/3}$ adversarial queries. The authors have also shown the applicability of their result by instantiating the two-keyed variants of SUM-ECBC, PMAC_Plus, 3kf9, LightMAC_Plus and have proven $2n/3$-bit security for all of them. Using the generic result, authors have also improved the security bound for SUM-ECBC and PMAC_Plus.
In [LNS18], Leurent et al. have shown attacks on all these constructions with $2^{3n/4}$-query complexity. Recently, Kim et al. [KLL20] have proven $3n/4$-bit security of DbHtS and hence established the tightness of the bound for SUM-ECBC, PMAC_Plus, 3kf9 and LightMAC_Plus.

Permutation-based cryptography. A block cipher is generally designed to be efficient in evaluating the input in both forward and backward directions. However, a closer inspection reveals that all the block cipher-based PRFs discussed so far do not require the inverse mapping of the block ciphers. Thus, a block cipher is an over-engineered primitive for block cipher-based PRF constructions that do not require the inverse function of their underlying primitives.

Concurrently with block ciphers, cryptographic permutations have evolved as useful primitives. The primary feature of a cryptographic permutation is that it does not use any key and hence does not require any separate processing for it. The use of cryptographic permutations gained popularity during the SHA-3 competition [RBB03] as several submitted candidates in the competition were based on this type of primitive. The selection of the permutation-based Keccak sponge function as the SHA-3 standard has further boosted the level of confidence of the community in using this primitive. Today, permutation-based sponge-based constructions have become a successful and full-fledged alternative to block cipher-based modes. In fact, in the first round of the ongoing NIST lightweight competition [NIS18], 24 out of the 57 submitted constructions are based on cryptographic permutations, and out of these 24, 16 permutation-based proposals have qualified for round 2. These statistics depict the wide adoption of permutation-based designs [CDNY18, BKL+17, BCDM19, CN19, DHP+19, DEMS19] in the community. A long line of research has also been carried out in the study of designing block ciphers and tweakable block ciphers out of public random permutations. Iterated Even Mansour (IEM) [CS14] and Tweakable Even-Mansour (TEM) [CLS15] ciphers are notable approaches in this direction.

PRFs Built from Public Permutations. Variable input-length PRFs built using public permutations mostly follow sponge-type constructions. Inherent drawbacks of such designs are that (i) they do not use the full size of the permutation for guaranteeing security and (ii) they attain only birthday bound security in the size of their capacity c, (except Bettle [CDNY18], whose security bound is roughly the size of its capacity). It is obvious that the sponge-type designs offering $c/2$-bit security are good in practice when they are instantiated with large permutations such as Keccak [BDPA13]. However, just like large block ciphers, large permutations are not suitable for a resource-constrained environment. In such a scenario, lightweight permutations such as SPONGENT [BKL+13] and PHOTON [GPP11] (whose state sizes go as low as 88 and 100 bits respectively) are preferred over large ones. The use of these lightweight permutations in birthday bound secure sponge constructions offers a practically inadequate security. Thus, to utilize lightweight permutations in practice, the natural choice would be to design a beyond the birthday bound secure mode. In this regard, Chen et al. [CLM19] have proposed two instances of public permutation-based pseudo-random functions, namely SoEM22 and SoKAC1. Both of them map an n-bit input to an n-bit output and offer beyond the birthday bound security with respect to the state size of the permutation. However, Nandi [Nan20b] has shown a birthday bound attack on SoKAC1 and hence invalidated its beyond the birthday bound security claim. Bhattacharjee et al. [BDLN20] have shown a public permutation-based fixed input-length to variable output-length PRF called XORPP* and its domain-separated variant called DS-XORPP*. Both of these constructions are built with a CENC [IMV16]-style design and both of them have $2n/3$-bit security [BDLN20]. Chakraborti et al. [CNTY20] have proposed a beyond the birthday bound secure public permutation-based fixed input-length PRF, called PDMMAC, a variable input-length PRF PDM*MAC and its single-keyed variant. Recently, Dutta et al. [DNT21b] have proposed another candidate for public permutation-based PRFs, called pEDM, and have shown a tight $2n/3$-bit security. This line of research has been further extended in [DN20b] by Dutta and Nandi, where they have proposed a beyond the birthday bound secure nonce-based MAC build on top of public permutations.

Our Contribution. Given the state of the art in permutation-based cryptography, it is natural to wonder whether we can design a variable input-length PRF based on some lower-level primitive like public permutations instead of block ciphers that offer beyond the birthday bound security. In this paper, we provide an answer in the positive. To this end, we propose a permutation-based PMAC_Plus construction, which we call pPMAC_Plus. The permutation-based variant of PMAC_Plus is exactly similar to PMAC_Plus with the following exception: in the block cipher-based PMAC_Plus construction, the output t is defined as follows:

$$t = \mathsf{E}_{k_1}(\Sigma) \oplus \mathsf{E}_{k_2}(\Theta),$$

where (Σ, Θ) is the $2n$-bit output value of the underlying double-block hash function PMAC_Plus-Hash. For pPMAC_Plus, we mask Σ and Θ with k_2 and follow by a domain separation through $\mathsf{chop}_{\mathsf{LSB}}(\cdot)\|0$, $\mathsf{chop}_{\mathsf{LSB}}(\cdot)\|1$, respectively. Next, we replace both $\mathsf{E}_{k_1}(\cdot)$ and $\mathsf{E}_{k_2}(\cdot)$ by an n-bit public random permutation $\pi(\cdot)$ (where k_1 and k_2 are two independently sampled block cipher keys). While PMAC_Plus-Hash is built from a block cipher E_k (independent from E_{k_1} and E_{k_2}), E_k is also replaced by π in pPMAC_Plus-Hash, the

α^{th} block of the input message masked with the string $(2^{\alpha}k_0 \oplus 2^{2\alpha}k_1)$, where k_0, k_1 and k_2 are three independently sampled n-bit strings.

One can directly replace each block cipher of PMAC_Plus with the two-round iterated Even-Mansour cipher [CLL+14b] or Mennnink's SoEM22 construction [CLM19] and obtain a beyond the birthday bound secure PRF based on public permutations. While both the solutions incur $(2\ell + 4)$ permutation calls, our proposal requires only $(\ell + 2)$ permutation calls, where ℓ is the maximum number of message blocks. Furthermore, unlike PMAC_Plus which has a tight $3n/4$-bit security, we have shown that pPMAC_Plus achieves a tight security bound of the order of $2^{2n/3}$.

2 Preliminaries

General Notations. For a positive integer q, $[q]$ denotes the set $\{1, \ldots q\}$ and for two natural numbers q_1, q_2 such that $q_2 > q_1$, $[q_1, q_2]$ denotes the set $\{q_1, \ldots, q_2\}$. We write $[n)$ to denote the set $[n] \cup \{0\}$. For a fixed positive integer n, we write $\{0,1\}^n$ to denote the set of all binary strings of length n and $\{0,1\}^* = \cup_{i \geq 0}\{0,1\}^i$ to denote the set of all binary strings with arbitrary finite length. We refer to the elements of $\{0,1\}^n$ as *blocks*. For any element $x \in \{0,1\}^*$, $|x|$ denotes the number of bits in x and for $x, y \in \{0,1\}^*$, $x\|y$ denotes the concatenation of y to x. A function $\mathsf{chop}_{\mathsf{LSB}} : \{0,1\}^n \to \{0,1\}^{n-1}$ removes the least significant bit of a string $x \in \{0,1\}^n$. We denote the bitwise XOR operation of $x, y \in \{0,1\}^n$ by $x \oplus y$. We parse $x \in \{0,1\}^*$ as $x = x_1\|x_2\| \ldots \|x_l$ where for each $i = 1, \ldots, l-1$, x_i is a block and $1 \leq |x_l| \leq n$. For a tuple $\widetilde{x} := (x_1, \ldots, x_q)$ of length q, an element x_i of $\widetilde{x}$ is called *fresh* if for all $j \neq i$, $x_i \neq x_j$. Otherwise, we say x_i is *not fresh* or *repeated* in $\widetilde{x}$. Sometimes we denote tuple $\widetilde{x}$ as $(x_i)_{i \in [q]}$. $\widetilde{x}$ is said to be *distinct* if each of its elements is fresh. Otherwise, we say it is *not a fresh tuple*. We call $\widetilde{x}$ a *block-tuple*, if each of its element is a member of $\{0,1\}^n$. Concatenation of two tuples $\widetilde{x}$ and $\widetilde{y}$ is denoted by $(\widetilde{x}, \widetilde{y})$. For a set $\mathcal{X}$, $\mathcal{X}^{(q)} := \{(x_1, \ldots, x_q) : x_i \in \mathcal{X}, \forall i \neq j \in [q], x_i \neq x_j\}$ denotes the set of all distinct tuples over $\mathcal{X}$ of length q. $(\{0,1\}^n)^{(q)}$ denotes the set of all block-wise distinct tuples of length q. For a finite subset $\mathcal{S}$ of $\mathbb{N}$, $\max \mathcal{S}$ denotes the maximum valued elements of $\mathcal{S}$.

Given a finite set $\mathcal{S}$ and a random variable X, we write $X \leftarrow\!\!\$\, \mathcal{S}$ to denote that X is sampled uniformly at random from $\mathcal{S}$. We say that $X_1, X_2, \ldots, X_q$ are without replacement (wor) sampled from $\mathcal{S}$, which we denote as $X_1, X_2, \ldots, X_q \xleftarrow{\text{wor}} \mathcal{S}$, if for each $i \in [q], X_i \leftarrow\!\!\$\, \mathcal{S} \setminus \{X_1, \ldots, X_{i-1}\}$. Note that when $i = 1$, then $X_1 \leftarrow\!\!\$\, \mathcal{S}$. We say that $X_1, X_2, \ldots, X_q$ are with replacement (wr) sampled from $\mathcal{S}$, which we denote as $X_1, X_2, \ldots X_q \leftarrow\!\!\$\, \mathcal{S}$, if for each $i \in [q], X_i \leftarrow\!\!\$\, \mathcal{S}$. We also use this notion to denote that these random variables are sampled uniformly and independently from $\mathcal{S}$. ϕ denotes the empty set. We write $\mathcal{S} \leftarrow \phi$ to denote that $\mathcal{S}$ is defined to be an empty set. We use the same notation $\Phi \leftarrow \phi$ to denote that the function Φ is undefined at every point of its domain. Moreover, the same notation $Y \leftarrow X$ is used to denote assignment of the variable X to Y.

The set of all functions from $\mathcal{X}$ to $\mathcal{Y}$ is denoted as $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$. Similarly, the set of all permutations over $\mathcal{X}$ is represented by $\mathsf{Perm}(\mathcal{X})$. A function Φ is said to be a *block function* if it maps elements from an arbitrary domain to $\{0,1\}^n$. Set of all block functions with domain $\mathcal{X}$ is denoted as $\mathsf{Func}(\mathcal{X})$. [1] A permutation over $\{0,1\}^n$ is called a *block permutation* and the set of all block permutations is denoted as Perm. If Φ maps to $(\{0,1\}^n)^2$, then we call it a *double-block function*. We write a double block function as $\Phi = (\Phi_1, \Phi_2)$, where Φ_1 and Φ_2 are block functions. For integers $1 \leq b \leq a$, we write $(a)_b$ to denote $a(a-1)\ldots(a-b+1)$, where $(a)_0 = 1$ by convention.

[1] When $\mathcal{X} = \{0,1\}^n$ then we write Func to denote $\mathsf{Func}(\{0,1\}^n)$.

2.1 Distinguishing Advantage

An adversary A is modeled as a randomized algorithm with access to some external oracle $\mathcal{O}$. Such an adversary is called an *oracle adversary.* An oracle $\mathcal{O}$ is an algorithm itself which could be a cryptographic scheme being analyzed. The interaction between A and $\mathcal{O}$, denoted by $\mathsf{A}^{\mathcal{O}}$, generates a transcript $\tau = \{(x_1, y_1), (x_2, y_2), \ldots, (x_q, y_q)\}$, where $x_1, x_2, \ldots, x_q$ are q queries of A to oracle $\mathcal{O}$ and $y_1, y_2, \ldots, y_q$ be the corresponding responses, where $y_i = \mathcal{O}(x_i)$. We assume that A is **adaptive** which means that x_i is dependent on the previous $i - 1$ responses.

Distinguishing Game. Let F and G be two random systems and an adversary A is given oracle access to either of F or G. After interaction with an oracle $\mathcal{O} \in \{\mathsf{F}, \mathsf{G}\}$, A outputs 1, which is denoted as $\mathsf{A}^{\mathcal{O}} \Rightarrow 1$. Such an adversary is called a *distinguisher* and the game is called a *distinguishing game.* The task of the distinguisher in a distinguishing game is to tell which of the two systems it has interacted with. The advantage of distinguisher A in distinguishing the random system F from G is defined as

$$\mathbf{Adv}_{\mathsf{G}}^{\mathsf{F}}(\mathsf{A}) := |\ \Pr[\mathsf{A}^{\mathsf{F}} \Rightarrow 1] - \Pr[\mathsf{A}^{\mathsf{G}} \Rightarrow 1]\ |,$$

where the above probability is defined over the probability spaces of A and $\mathcal{O}$. The maximum advantage in distinguishing F from G is defined as the

$$\max_{\mathsf{A} \in \mathcal{A}} \mathbf{Adv}_{\mathsf{G}}^{\mathsf{F}}(\mathsf{A}),$$

where $\mathcal{A}$ is the class of all possible distinguishers. One can easily generalize this setting when the distinguisher interacts with multiple oracles, which are separated by commas. For example, $\mathbf{Adv}_{\mathsf{G}_1,\ldots,\mathsf{G}_m}^{\mathsf{F}_1,\ldots,\mathsf{F}_m}(\mathsf{A})$ denotes the advantage of A in distinguishing $(\mathsf{F}_1, \ldots, \mathsf{F}_m)$ from $(\mathsf{G}_1, \ldots, \mathsf{G}_m)$.

2.2 PRF Security in the Random Permutation Model

A *keyed function* with the key space $\mathcal{K}$, the domain $\mathcal{X}$ and the range $\mathcal{Y}$ is a function $\mathsf{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. We denote $\mathsf{F}(k, x)$ by $\mathsf{F}_k(x)$. A random function RF from $\mathcal{X}$ to $\mathcal{Y}$ is a uniform random variable over the set $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$, i.e., $\mathsf{RF} \leftarrow_{\$} \mathsf{Func}(\mathcal{X}, \mathcal{Y})$. We define the pseudorandom security of F under the random permutation model. We assume that F makes internal public-random-permutation calls to π (F can make calls to multiple random permutations when all of them are independent and uniform over the set Perm). For simplicity, we write F_k^{π} to denote F with a uniformly random permutation $\pi \leftarrow_{\$} \mathsf{Perm}$ and a uniformly random key $k \leftarrow_{\$} \mathcal{K}$. The distinguisher A is given access to either $(\mathsf{F}_k^{\pi}, \pi)$ for $k \leftarrow_{\$} \mathcal{K}$ or (RF, π) where *pi* is an n-bit uniform random permutation. We define the prf-advantage of A against a keyed function F in the random permutation model as

$$\mathbf{Adv}_{\mathsf{F}}^{\mathrm{PRF}}(\mathsf{A}) := \mathbf{Adv}_{(\mathsf{RF},\pi)}^{(\mathsf{F}_k^{\pi},\pi)}(\mathsf{A}).$$

We say F is a (q, p, ϵ, t)-PRF if $\mathbf{Adv}_{\mathsf{F}}^{\mathrm{PRF}}(\mathsf{A}) \leq \epsilon$ for all adversaries A that makes q queries to F, p offline queries to π and runs for at most time t.

2.3 Lazy Sampling of Random Permutations

Consider a distinguisher A interacting with an n-bit random permutation $\pi \leftarrow_{\$} \{0, 1\}^n$. We simulate this interaction by a simulator $\mathcal{S}$ that maintains a partial function Ψ. Ψ is initially defined to be an empty function (a function with empty domain), i.e., $\Psi \leftarrow \phi$. We consider two dynamically growing sets $\mathsf{Dom}(\Psi)$ and $\mathsf{Ran}(\Psi)$ associated to Ψ, such that the points at which Ψ has already been defined gets included in $\mathsf{Dom}(\Psi)$ and their respective defined values get included in $\mathsf{Ran}(\Psi)$. Initially, $\mathsf{Dom}(\Psi), \mathsf{Ran}(\Psi) \leftarrow \phi$. On the i^{th} query x_i, the simulator checks whether $x_i \in \mathsf{Dom}(\Psi)$. If so, the corresponding response is $y_i \leftarrow \mathsf{Psi}(x_i)$. Else, the response is sampled uniformly from $\{0,1\}^n \setminus \mathsf{Ran}(\Psi)$ and x_i, y_i are added to $\mathsf{Dom}(\Psi)$ and $\mathsf{Ran}(\Psi)$ respectively, i.e.,

$$\mathsf{Dom}(\Psi) \leftarrow \mathsf{Dom}(\Psi) \cup \{x_i\},\ \mathsf{Ran}(\Psi) \leftarrow \mathsf{Ran}(\Psi) \cup \{y_i\}.$$

Note that at any point of time $\mathsf{Dom}(\Psi), \mathsf{Ran}(\Psi) \subseteq \{0,1\}^n$.

2.4 H-Coefficients Technique

The H-Coefficients Technique [Pat08, CLL+14b] was introduced by Patarin [Pat08] and recently regained attention since Chen and Steinberger used it to analyze the iterated Even-Mansour cipher [CS14]. This technique gives a systematic way to upper bound the statistical distance between the answers of the distributions of two interactive random systems and is typically used to prove the information theoretic pseudo-randomness of constructions. In this setting, we consider a computationally unbounded and hence deterministic distinguisher A that interacts with the oracles in either of the two worlds: (a) oracles in the real world, which happens to be the construction of our interest, or (b) the oracles in the ideal world, which is usually considered a uniform random function or permutation. The collection of all queries and responses that A makes and receives to and from the oracles in either of the two worlds, is called the *attack transcript* of A, denoted as τ. In both worlds, the oracle sometimes releases more internal information to A after it completes all its queries and responses, but before outputs its decision. In this case, the attack transcript of A includes the additional information, and clearly, the maximum distinguishing advantage of A in this setting can not be less than the previous one. Observe that the transcript τ is a random variable and the randomness of its distribution only comes from that of the oracles present in either of the two worlds with which A has interacted.

Let D_{re} and D_{id} be two random variables that takes the transcript τ induced in the real world and the ideal world respectively. The probability of realizing a transcript τ in the ideal world, i.e., $\Pr[\mathsf{D}_{\text{id}} = \tau]$) is called the *ideal interpolation probability.* Similarly, one can define the *real interpolation probability.* A transcript τ is said to be *attainable* with respect to A if its ideal interpolation probability is non-zero (i.e., $\Pr[\mathsf{D}_{\text{id}} = \tau] > 0$). We denote the set of all attainable transcripts by $\mathcal{V}$. Following these notations, we state the main theorem of H-Coefficients Technique [Pat08, CLL+14b] as follows:

Theorem 1 (**H-Coefficients Technique**). *Let* A *be a fixed deterministic distinguisher that has access to oracles either in the real world, i.e.,* $\mathcal{O}_{\text{re}}$ *or the oracles in the ideal world, i.e.,* $\mathcal{O}_{\text{id}}$*. Let* $\mathcal{V} = \mathsf{GoodT} \sqcup \mathsf{BadT}$ *(disjoint union) be some partition of the set of all attainable transcripts of* A*. Suppose there exists* $\epsilon_{\text{ratio}} \geq 0$ *such that for any* $\tau \in \mathsf{GoodT}$*,*

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

and there exists $\epsilon_{\text{bad}} \geq 0$ *such that* $\Pr[\mathsf{D}_{\text{id}} \in \mathsf{BadT}] \leq \epsilon_{\text{bad}}$*. Then,*

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

Note that when $\mathcal{O}_{\text{id}}$ *is a uniform random function and* $\mathcal{O}_{\text{re}}$ *is some keyed construction defined over the same domain, then Eqn.* (1) *says that* $\mathbf{Adv}^{\text{prf}}_{\mathcal{O}_{\text{re}}}(\mathsf{A}) \leq \epsilon_{\text{ratio}} + \epsilon_{\text{bad}}$*.*

3 Some Useful Mathematical Results

This section presents three important results with later use in the security analysis of our proposed construction. The first result is concerned with the sum of two independent random permutations under conditional distribution. The second one bounds the probability of multicollision of the sum of two uniform random variables, while the last one gives an upper bound on the number of solutions to a system of equations with unknown variables that are supposed to take distinct values.

3.1 Sum of Two Independent Random Permutations Under a Conditional Distribution

Let E be a block cipher over n-bits. Based on E, we define the *sum function* as follows:

$$\mathsf{sum}_{k_1,k_2}(x) := \mathsf{E}_{k_1}(x) \oplus \mathsf{E}_{k_2}(x),\ \ x \in \{0,1\}^n.$$

The security of the sum of two identical random permutations (i.e., when $k_1 = k_2$) under conditional distribution has been studied in [DDN+17]. This paper requires the same result with the change that instead of two identical random permutations, it considers the permutations to be independent (i.e., k_1 and k_2 are independently sampled). Proof of the lemma is straightforward and similar to that of Theorem 2 of [DDN+17]. Hence we omit the proof.

Lemma 1. *Let $\mathcal{Y}_1 \subseteq \{0,1\}^n$ and $\mathcal{Y}_2 \subseteq \{0,1\}^n$ be two sets of size s_1 and s_2 respectively. Let $\widetilde{t} := (t_1, \dots, t_r)$ be a block tuple of length r. We define the following set:*

$$\mathcal{H} := \{(h_i^1, h_i^2)_i : h_i^1 \oplus h_i^2 = t_i\ \forall i \in [r],\ (h_i^b)_i \in (\{0,1\}^n \setminus \mathcal{Y}_b)^{(r)}\ \forall b \in [2]\}.$$

Then we have the following lower bound on the cardinality of $\mathcal{H}$:

$$|\mathcal{H}| \geq \frac{(2^n - s_1)_r(2^n - s_2)_r}{2^{nr}}\left(1 - \frac{rs_1s_2 + r^2(s_1 + s_2) + r^3}{(2^n - s_1 - r)(2^n - s_2 - r)}\right).$$

Moreover, if $s_1 + r \leq 2^{n-1}$ and $s_2 + r \leq 2^{n-1}$, then we have

$$|\mathcal{H}| \geq \frac{(2^n - s_1)_r(2^n - s_2)_r}{2^{nr}}\left(1 - \frac{4rs_1s_2 + 4r^2(s_1 + s_2) + 4r^3}{2^{2n}}\right).$$

3.2 A Sum-Capture Lemma

In this section, we state a variant of the sum-capture lemma [Bab] used in [CLL+14a]. Informally, the result states that when choosing a random subset $\mathcal{A}$ of $\{0,1\}^n$ (or more generally any abelian group) of size q, the value

$$\mu(\mathcal{A}) := \max_{\mathcal{B},\mathcal{C} \subseteq \{0,1\}^n} |\{(a,b,c) \in \mathcal{A} \times \mathcal{B} \times \mathcal{C} : a = b \oplus c\}|,$$

is at most $q|\mathcal{B}||\mathcal{C}|/2^n$, except with negligible probability. Chen et al. [CLL+14a] proved the result for a different setting, in which $\mathcal{A}$ arises from the interaction of an adversary with a random permutation P, namely $\mathcal{A} = x \oplus y : (x,y) \in \mathcal{Q}$, where $\mathcal{Q}$ is the transcript of the interaction between the adversary and the permutation. Cogliati and Seurin [CS16] used this result in a slightly different setting. We state below the result as stated in [CS16], the proof of which can be found in [CS16].

Lemma 2. *Let T^* be a multiset of $q \geq 1$ uniformly random and independently chosen elements of $\{0,1\}^n$. Then assuming $9n \leq q \leq 2^{n-1}$, we have*

$$\Pr_{T^*}\left[\exists \mathcal{U}, \mathcal{V} \subseteq \{0,1\}^n : \mu(T^*, \mathcal{U}, \mathcal{V}) \geq \frac{q|\mathcal{U}||\mathcal{V}|}{2^n} + 3\sqrt{nq|\mathcal{U}||\mathcal{V}|}\right] \leq \frac{2}{2^n}, \tag{2}$$

where the probability is taken over the uniform distribution of the multiset T^*.

3.3 Some Results on Linear Algebra

Let A be a matrix of dimension $s \times t$ defined over $\{0,1\}^n$. A_{ij} denotes the element in its i^{th} row and j^{th} column. For a column vector $\widetilde{C}$ of dimension $s \times 1$, $A \| \widetilde{C}$ denotes the augmented matrix of dimension $s \times (t+1)$. For any row vector $\widetilde{R} \coloneqq (r_1, \ldots, r_t)$ of dimension $1 \times t$, transpose of row vector $\widetilde{R}$, denoted as $\widetilde{R}^{\mathsf{T}}$, denotes the column vector

$$\widetilde{R}^{\mathsf{T}} \coloneqq \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_t \end{pmatrix}$$

of dimension $t \times 1$. One can represent any system of s linear equations with t unknowns $\widetilde{Y} \coloneqq (Y_1, \ldots, Y_t)$ defined over $\{0,1\}^n$, denoted as $\mathcal{L}$, as a matrix A of dimension $s \times t$, where the i^{th} equation $\mathcal{L}_i \coloneqq a_{i1} \cdot Y_1 \oplus \ldots \oplus a_{it} \cdot Y_t = c_i$, where $c_i \in \{0,1\}^n$, corresponds to the i^{th} row vector of A as $\widetilde{a}_i \coloneqq (a_{i1}, \ldots, a_{it})$. We say $\mathcal{L}$ is *consistent* if it has at least one solution, otherwise we call it *inconsistent*. For $\mathcal{L}$ to be consistent, one must have $\mathtt{rank}(A) = \mathtt{rank}(A \| \widetilde{C})$, where the rank of a matrix A is defined as the maximum number of linearly independent columns of A and $\widetilde{C} = (c_1, \ldots, c_s)^{\mathsf{T}}$. $\mathcal{L}$ has a unique solution if $\mathtt{rank}(A) = t$ and it has many solutions if $\mathtt{rank}(A) < t$.

Let $A \cdot \widetilde{Y}^{\mathsf{T}} = \widetilde{C}$ represent a system of s linear equations with t unknowns $\widetilde{Y}$ defined over $\{0,1\}^n$, where $\mathtt{rank}(A) = r$ and the elements of A are from $\{0,1\}^n$. Let $\widetilde{Y} \xleftarrow{\text{wor}} \mathcal{Y} \subseteq \{0,1\}^n$ and $\widetilde{C}$ is any arbitrary column vector of dimension $s \times 1$ with its elements from $\{0,1\}^n$. Thus, the probability of realizing a particular solution is at most $\frac{1}{(|\mathcal{Y}|-t+r)_r}$ as stated formally in the following lemma, proof of which can be found in [DDN+17].

Lemma 3. *Let $\widetilde{Y} \coloneqq (Y_1, \ldots, Y_t)$ be without replacement samples from a set $\mathcal{Y} \subseteq \{0,1\}^n$ and A be a matrix of dimension $s \times t$ defined over $\{0,1\}^n$. Then, for any given column vector $\widetilde{C}$ of dimension $s \times 1$ over $\{0,1\}^n$, we have*

$$\Pr[(A)_{s \times t} \cdot \widetilde{Y}^{\mathsf{T}} = \widetilde{C}] \leq \frac{1}{(|\mathcal{Y}| - t + r)_r},$$

where $r = \mathtt{rank}(A)$.

4 pPMAC_Plus: A Public Permutation-Based BBB Secure MAC

In this section, we propose pPMAC_Plus, a public permutation-based beyond the birthday bound secure MAC. It takes an n-bit independent public permutation π and three independent n-bit keys k_0, k_1 and k_2. For processing a message $M \in \{0,1\}^*$, the padding function $\mathsf{pad} : \{0,1\}^* \rightarrow (\{0,1\}^n)^+$ is applied on M that parses M into l blocks $(M[1], M[2], \ldots, M[l])$ by concatenating $10*$ to the right so that for each $i \in [l-1]$, $|M[i]| = n$ and $1 \leq |M[l]| \leq n$.

For each $\alpha \in [l]$, the message block $M[\alpha]$ of M is masked with $2^{\alpha} k_0 \oplus 2^{2\alpha} k_1$ before passing it through the permutation π. Output blocks of the permutation are then XORed together, followed by masking with another key k_2 to generate an n-bit value Σ. Each output block of the hash permuation instances is simply XORed in one case and multiplied by 2 before XORing in another, and both are masked with the key k_2 to generate output values Σ and Θ, respectively. Finally, $\mathsf{chop}_{\mathsf{LSB}}\,(\Sigma \oplus k_2)\,\|0$ and $\mathsf{chop}_{\mathsf{LSB}}\,(\Theta \oplus k_2)\,\|1$ are passed through two copies of the same permutation π (as used in the hash function) and the XOR of their outputs produces the MAC T. An algorithmic description of pPMAC_Plus is given in Fig. 4.1, and a pictorial illustration in Fig. 4.2.

```
$\text{pPMAC\_Plus}^{\pi}_{k_0,k_1,k_2}(M)$
$(\Sigma, \Theta) \leftarrow \mathsf{pPMAC\_Plus\text{-}Hash}(M)$.
return $T \leftarrow \pi(\Sigma) \oplus \pi(\Theta)$.
```

```
$\text{pPMAC\_Plus-Hash}^{\pi}_{k_0,k_1,k_2}(M)$
$(M[1], M[2], \ldots, M[\ell]) \leftarrow \mathsf{pad}(M)$.
$(\Sigma, \Theta) \leftarrow (0^n, 0^n)$.
for $(i = 1; i \leq \ell; i++)$
  $X \leftarrow 2^i k_0 \oplus 2^{2i} k_1 \oplus M[i]$.
  $Y_i \leftarrow \pi_0(X)$.
  $\Sigma \leftarrow \Sigma \oplus Y_i, \Theta \leftarrow \Theta \oplus 2^{l-i+1} \cdot Y_i$.
return
$(\mathsf{chop_{LSB}}(\Sigma \oplus k_2) \| 0, \mathsf{chop_{LSB}}(\Theta \oplus k_2) \| 1)$.
```

Figure 4.1: pPMAC_Plus is depicted on the left, while a permutation-based DbH function of pPMAC_Plus is shown on the right.

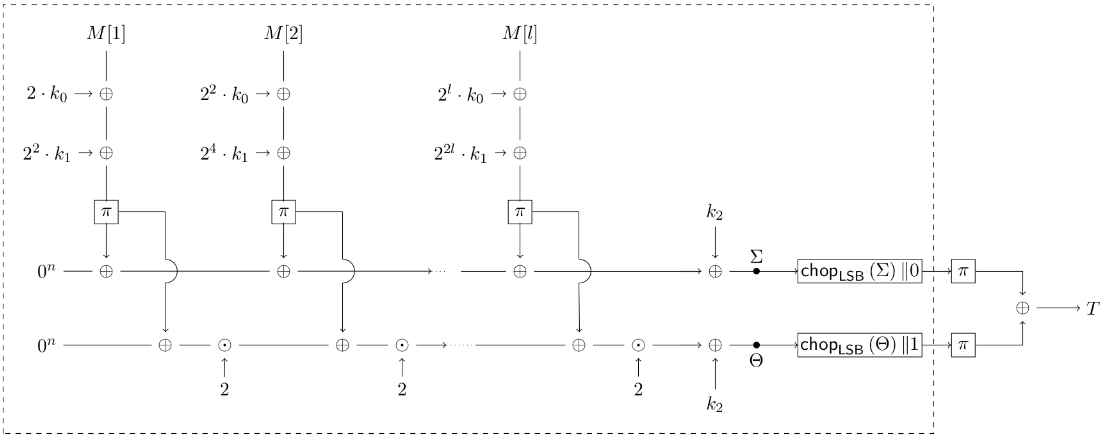

Figure 4.2: A message with ℓ blocks (after padding) is input into $\mathsf{pPMAC\_Plus} - \mathsf{Hash}^{\pi}_{k_0,k_1,k_2}$, which produces outputs Σ and Θ. These outputs are masked with a random key k_2 before going through a domain-separating function, and the final pPMAC_Plus output is produced by passing these two values through two permutations and adding the resultants.

Remark 1. Note that the structural design of pPMAC_Plus is similar to that of PMAC_Plus. The only difference of the former with the latter is that PMAC_Plus uses a block cipher E with three independent block cipher keys, whereas pPMAC_Plus replaces E by an n-bit random permutation π alongwith some masking elements and domain separation. It is easy to see that directly replacing E in PMAC_Plus by the-two round iterated Even-Mansour cipher or SoEM22 construction [CLL+14b] immediately leads the security of the resulting construction to beyond the birthday bound. However, both solutions pay a price for invoking the underlying permutation twice to process a single message block. Therefore, processing an ℓ-block message requires $2\ell + 4$ permutation calls in the former approach and only $\ell + 2$ permutation calls in ours.

4.1 Security of pPMAC_Plus

In this section, we state that pPMAC_Plus is secure against any information theoretic adversary that makes roughly $2^{2n/3}$ online and offline queries.

Theorem 2 (**Security of pPMAC_Plus**). *Let $\mathcal{M}$ be a non-empty finite set and π a uniformly sampled n-bit public permutation. Let A be any distinguisher that makes at most q construction queries and at most p primitive queries, and runs for at most time t. Then*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{pPMAC\_Plus}}(\mathsf{A}) \leq \frac{2\sqrt{3nqp_1p_2}+4}{2^n} + \frac{q^3(5\ell^3+3\ell^2+8\ell+4)}{2^{2n}} + \frac{6qp^2l^2+2q^2pl+2q^2l}{2^{2n}} + \frac{4q^3+45q^2p+20qp^2+5q^2}{2^{2n}}.$$

The PRF security of pPMAC_Plus is roughly at most $2^{2n/3}$ when $q \approx p$.

5 A Key-Recovery Attack on pPMAC_Plus

In this section, we show a matching key-recovery attack on pPMAC_Plus with a total of $2^{2n/3+1}$ of each of construction and primitive queries. We refer the readers to the full attack in Fig. 5.1.

Backward attack. The attack proceeds by first making $2^{2n/3}$ construction queries of two-block messages $M_i[1]\|M_i[2]$ for $i \in [2^{2n/3}]$, and collects the responses T_i. Next, it makes two sets of $2^{2n/3}$ offline forward queries – one with least significant bit (LSB) 0 and the other with LSB 1 – to the primitive permutation π, and collects their corresponding responses in lists $\mathcal{L}_0$ and $\mathcal{L}_1$, respectively. All these $2p = 2^{2n/3+1}$ forward queries and their responses are also collected into a list of pairs $\mathcal{L} = \{(\tilde{x}_1, \tilde{y}_1), (\tilde{x}_2, \tilde{y}_2), \ldots, (\tilde{x}_a, \tilde{y}_a), \ldots, (\tilde{x}_{2p}, \tilde{y}_{2p})\}$. A check of pairs $(\tilde{v}_b, \tilde{z}_c) \in (\mathcal{L}_0 \times \mathcal{L}_1)$ such that $\tilde{v}_b \oplus \tilde{z}_c = T_i$ provides triples (i, b, c) collected in a set $\mathcal{S}_1$. Computing pairs $(\hat{\Sigma}, \hat{\Theta})$ for all pairs of second-coordinates $(\tilde{y}_{a_1}, \tilde{y}_{a_2}) \in \mathcal{L}\big|_2 \times \mathcal{L}\big|_2$ helps filter the elements of $\mathcal{S}_1$ by checking whether

$$\hat{\Sigma} \oplus \tilde{u}_b = \hat{\Theta} \oplus \tilde{w}_c,$$

where $\tilde{u}_b$ and $\tilde{w}_c$ are the preimages of $\tilde{v}_b$ and $\tilde{z}_c$ respectively. If this check passes, then the attack computes a candidate key $\hat{k}_2$, stores the corresponding triple in $\mathcal{S}_2$ and then for all elements of $\mathcal{S}_2$, computes a pair of candidate keys $(\hat{k}_0, \hat{k}_1)$.

Removing false positives. In order to remove the false positive keys from the set of candidates, the attack makes another $2^{2n/3}$ construction queries with messages of two blocks $M'_i[1]\|M'_i[2]$, where $M'_i[1] = (M_i[1] \oplus 1)$ and $M'_i[2] = M_i[2]$, and collects their corresponding responses T'_i for $i \in [2^{2n/3}]$. Next, it evaluates pPMAC_Plus on messages $M'_i[1]\|M'_i[2]$, $i \in [2^{2n/3}]$, with the candidate key-triple $\left(\hat{k}_0, \hat{k}_1, \hat{k}_2\right)$. If the computed values match with the received responses T'_i, then this triple of keys $(\hat{k}_0, \hat{k}_1, \hat{k}_2)$ stays in the candidate key-list, otherwise, it is removed. We show that the true key belongs to the set of potential candidate keys with a high probability and that the size of the set of the candidate keys is not very large. We have thus described a deterministic adversary A that recovers the key of pPMAC_Plus by making a total of $2^{2n/3+1}$ construction queries and $2^{2n/3+1}$ primitive queries as shown in Fig. 5.1.

5.1 Analysis of the Attack

First observe that for internal values $x_i[1] = M_i[1] \oplus 2 \cdot k_0 \oplus 2^2 \cdot k_1$ and $x_i[2] = M_i[2] \oplus 2^2 \cdot k_0 \oplus 2^4 \cdot k_1$ ($M_i = M_i[1]\|M_i[2]$), $i \in [2^{2n/3}]$,

$$\mathbf{E}\left[\left|\{(i, a_1, a_2) \in [2^{2n/3}] \times [2^{2n/3+1}] \times [2^{2n/3+1}]\}\right| \ : \ (x_i[1] = \tilde{x}_{a_1}) \wedge (x_i[2] = \tilde{x}_{a_2})\right] = \mathcal{O}(1).$$

```
Construction and Primitive Queries
choose distinct $(M_i[1]\|M_i[2]) \in \{0,1\}^{2n}\ \forall i \in [2^{2n/3}]$
$T_i \leftarrow \mathsf{pPMAC\_Plus}\,(M_i[1]\|M_i[2])\, \forall i \in [2^{2n/3}]$.
choose distinct $\tilde{u}_b \in \{0,1\}^n\ \forall b \in [2^{2n/3}]$
$\tilde{v}_b \leftarrow \pi_1\,(\tilde{u}_b)\, \forall b \in [2^{2n/3}]$.
choose distinct $\tilde{w}_c \in \{0,1\}^n\ \forall c \in [2^{2n/3}]$
$\tilde{z}_c \leftarrow \pi_2\,(\tilde{w}_c)\, \forall c \in [2^{2n/3}]$.
$\{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{2^{2n/3+1}}\} \leftarrow \{\tilde{u}_b\}_{b=1}^{2^{2n/3}} \cup \{\tilde{w}_c\}_{c=1}^{2^{2n/3}}$.
$\tilde{y}_a \leftarrow \pi_0\,(\tilde{x}_a)\,, a \in [2^{2n/3+1}]$.
```

```
Backward Attack
$\mathcal{S}_1 \leftarrow \phi$.
$\forall (i,b,c) \in [2^{2n/3}] \times [2^{2n/3}] \times [2^{2n/3}]$, if $\tilde{v}_b \oplus \tilde{z}_c = T_i$ then $\mathcal{S}_1 \leftarrow \mathcal{S}_1 \cup \{(i,b,c)\}$
$\mathcal{S}_2 \leftarrow \phi$.
$\forall (a_1, a_2, (i,b,c)) \in [2^{2n/3+1}] \times [2^{2n/3+1}] \times \mathcal{S}_1$,
    compute $\hat{\Sigma}^{(a_1,a_2,(i,b,c))} \leftarrow \tilde{y}_{a_1} \oplus \tilde{y}_{a_2}$
    compute $\hat{\Theta}^{(a_1,a_2,(i,b,c))} \leftarrow 2^2 \cdot \tilde{y}_{a_1} \oplus 2 \cdot \tilde{y}_{a_2}$.
if $\hat{\Sigma}_2^{(a_1,a_2,(i,b,c))} \oplus \tilde{u}_b = \hat{\Theta}_2^{(a_1,a_2,(i,b,c))} \oplus \tilde{w}_c$ then
    $\hat{k}_2^{(a_1,a_2,(i,b,c))} \leftarrow \hat{\Sigma}_2^{(a_1,a_2,(i,b,c))} \oplus \tilde{u}_b$
    $\mathcal{S}_2 \leftarrow \mathcal{S}_2 \cup \{(a_1, a_2, (i,b,c))\}$.
 $\forall (a_1, a_2, (i,b,c)) \in \mathcal{S}_2$
    compute $\hat{k}_0^{(a_1,a_2,(i,b,c))} \leftarrow (2^3 \oplus 2^4)^{-1}\,(2 \cdot M_i[1] \oplus M_i[2] \oplus 2 \cdot \tilde{x}_{a_1} \oplus \tilde{x}_{a_2})\,$,
    compute $\hat{k}_1^{(a_1,a_2,(i,b,c))} \leftarrow (2^2 \oplus 2^3)^{-1}\left(2^2 \cdot M_i[1] \oplus M_i[2] \oplus 2^2 \cdot \tilde{x}_{a_1} \oplus \tilde{x}_{a_2}\right)$.
```

```
Removing False Positives
$T_i' \leftarrow \mathsf{pPMAC\_Plus}\,((M_i[1] \oplus 1)\|M_i[2])\ \ \forall i \in [2^{2n/3}]$.
$\mathcal{K} \leftarrow \phi$
$\forall (a_1, a_2, (i,b,c)) \in \mathcal{S}_2$,
    if $T_i' = \mathsf{pPMAC\_Plus}^{\left(\hat{k}_0^{(a_1,a_2,(i,b,c))}, \hat{k}_1^{(a_1,a_2,(i,b,c))}, \hat{k}_2^{(a_1,a_2,(i,b,c))}\right)}\,((M_i[1] \oplus 1)\|M_i[2])\, \forall i \in [2^{2n/3}]$,
    then $\mathcal{K} \leftarrow \mathcal{K} \cup \{\hat{k}_0^{(a_1,a_2,(i,b,c))}, \hat{k}_1^{(a_1,a_2,(i,b,c))}, \hat{k}_2^{(a_1,a_2,(i,b,c))}\}$.
return $\mathcal{K}$.
```

Figure 5.1: An attack on pPMAC_Plus, where a computationally unbounded adversary makes $\mathcal{O}\left(2^{2n/3}\right)$ queries to the construction and primitives.

Next, for internal values $u_i = y_i[1] \oplus y_i[2] \oplus k_2$ and $w_i = 2^2 \cdot y_i[1] \oplus 2 \cdot y_i[2] \oplus k_2$, $i \in [2^{2n/3}]$,

$$\mathbf{E}\left[\left|\{(b,c) \in [2^{2n/3}] \times [2^{2n/3}]\}\right| \ :\ (u_i = \tilde{u}_b) \wedge (w_i = \tilde{w}_c)\right] = \mathcal{O}(1).$$

Thus, bounding the number of queries to the construction and each of the primitives by $\mathcal{O}\left(2^{2n/3}\right)$ ensures the presence of at least one tuple (a_1, a_2, b, c) of primitive query indices that matches with true internal values corresponding to a construction index i with high probability.

The backward attack checks for the validity of the equations induced by the construction. First consider the set $\mathcal{S}_1$. It is computed over sets of sizes q, p and p with a restriction of two conditions on n-bit strings. Therefore, $\mathbf{E}\left[|\mathcal{S}_1|\right] = \frac{qp^2}{2^n}$. Similarly, $\mathbf{E}\left[|\mathcal{S}_2|\right] = \frac{qp^4}{2^{2n}}$. Note here that only the indices b, c that appear in tuples $(i, b, c) \in \mathcal{S}_1$ are considered for the check in step 7 of the backward attack, and the corresponding construction query-index i is used next for computing guess values $(\hat{k}_0, \hat{k}_1)$ of the key-pair. Observe that the probability depends on the sampling of values $\tilde{y}_a$, and not on the keys, as the hash computation of the message is not even considered so far.

By the same formula, the expected size of $\mathcal{K}$ is $|\mathcal{S}_2| \times q \times \frac{1}{2^{2n}} = \frac{q^2p^4}{2^{4n}}$. Since q and p both have the same order $\mathcal{O}\left(2^{2n/3}\right)$, $\mathbf{E}\left[|\mathcal{K}|\right]$ is $\mathcal{O}(1)$ when $q = \mathcal{O}\left(2^{2n/3}\right)$. Finally, since the true key is in $\mathcal{K}$ with very high probability due to the choice of lengths of the query-lists, the true key (k_0, k_1, k_2) must belong to $\mathcal{K}$ with very high probability. This demonstrates that the above is indeed an $\mathcal{O}\left(2^{2n/3}\right)$ attack on pPMAC_Plus.

6 Proof of Theorem 2

In this section, we prove Theorem 2. We often denote pPMAC_Plus$[\pi, k_0, k_1, k_2]$ simply by pPMAC_Plus* when the primitives and the underlying keys are understood. We consider any information theoretic deterministic distinguisher A that has access to a triplet of oracles in the real and the ideal worlds: In the real world, it has access to the oracles $\mathbf{O}_{\text{re}} := (\text{pPMAC\_Plus}^*, \pi^+, \pi^-)$, where π is a uniformly chosen random n-bit permutation and k_0, k_1, k_2 are three independently and uniformly chosen random n-bit keys. In the ideal world, it has access to the oracles $\mathbf{O}_{\text{id}} := (\$, \pi^+, \pi^-)$, where π is again a uniformly chosen n-bit random permutation. Queries to the first oracle in either of the two worlds are called *construction queries* and queries to the remaining oracles are called *primitive queries.* Note that as the primitive π is a permutation, an adversary can make queries in the forward direction, which we call *forward primitive queries*, as well as in the inverse direction, which we call *backward primitive queries.* Throughout the proof, we assume that neither does an adversary A make duplicate or redundant queries nor does it make queries whose responses can be constructed from the previous query-responses. We call such an adversary a **non-trivial adversary**. We also assume that A makes q construction queries and p (forward and backward) primitive queries in either of the two worlds.

Once an adversary has finished making all its queries, the keys k_0, k_1, k_2 in the real world, and corresponding dummy values in the ideal world are released to the adversary. Furthermore, the intermediate values $\left((x_i[1], x_i[2], \ldots, x_i\left[l_i\right]), (y_i[1], y_i[2], \ldots, y_i\left[l_i\right]), u_i, v_i, w_i, z_i\right)$ for each construction query $i \in [q]$ are also released. These values represent the following:

$$\begin{aligned} x_i[\alpha] = M_i \oplus 2^{\alpha} k_0 \oplus 2^{1\alpha} k_1 \;\forall\, \alpha \in [l_i] \quad &, \quad y_i[\alpha] = \pi(x_i[\alpha]) \;\forall\, \alpha \in [l_i] \\ u_i = \mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_i \oplus k_2\right) \| 0 \quad &, \quad v_i = \pi(u_i) \\ w_i = \mathsf{chop}_{\mathsf{LSB}}\left(\Theta_i \oplus k_2\right) \| 1 \quad &, \quad z_i = \pi(w_i). \end{aligned} \tag{3}$$

6.1 An Outline of the Proof

We begin the proof by providing well-defined algorithms for the interaction of an adversary with the real and ideal worlds. While the adversarial interaction with the real world only involves an online phase (since its responses are true to the construction), the ideal world also requires an offline phase for computation of certain output values so as to mimic the real world more closely. These algorithms are detailed in Fig.s 6.1–6.5.

Stage I of the offline phase of the ideal world (Fig. 6.3) lists certain events (which we call *bad events*), for which, the algorithm aborts; the probability of occurrence of these events is computed next. This is the bad event analysis, and can be found in Sect. 6.5.

The remaining cases are analyzed in the good transcript analysis (Sect. 6.6) by proving that the ideal interpolation probability is very close to the real interpolation probability. The computation for the real case is quite straightforward, and the bound is given by Eqn. (54).

For the ideal world, all queries made by the adversary to the online and offline oracles are indexed according to the respective algorithms. These indices are first split into those corresponding to free (non-repeating hash output blocks) and single-colliding (collision in exactly one block of the hash output) indices. An equivalence relation is defined according to the collisions of the hash function outputs of the second category of indices so as to classify the output definitions for both inputs. These steps are detailed in Stages II (Fig. 6.4) and III (Fig. 6.5) of the offline phase of the ideal world. This partitions all queried indices into the following sets:

1. $\mathcal{F}$ is the set of indices corresponding to free queries,
2. $\mathcal{I}$ contains the indices corresponding to queries with one hash output block colliding with a primitive query input,
3. $\mathcal{P}^c$ is the set of indices corresponding to queries with one of their hash output blocks colliding with one of the hash-primitive inputs, and
4. $\mathcal{Q}^c$ is the set of indices corresponding to queries with one of their hash output blocks colliding with the corresponding block of the hash output of another query.

6.2 Real World and Ideal World

In the real world, when an adversary A makes a construction query with message M to $\mathsf{pPMAC\_Plus}^*$, it receives the tag $T \leftarrow \mathsf{pPMAC\_Plus}^*(M)$. In the ideal world, when A makes a construction query with message M to \$, it samples an n-bit tag $T \leftarrow_{\$} \{0,1\}^n$ and returns it to A. In both the worlds, A is allowed to make forward as well as backward primitive queries to π. When A makes the a^{th} forward query $\tilde{x}_a$ to π for $a \in [2p]$, it samples $\tilde{y}_a \leftarrow_{\$} \{0,1\}^n \setminus \{\tilde{y}_1, \ldots, \tilde{y}_{a-1}\}$ and returns it to the adversary. Similarly, for the a^{th} backward query $\tilde{y}_a$ to π, it returns $\tilde{x}_a \leftarrow_{\$} \{0,1\}^n \setminus \{\tilde{x}_1, \ldots, \tilde{x}_{a-1}\}$ and returns it to the adversary.
The behavior of the oracles in the real and ideal worlds is detailed in Fig.s 6.1 and 6.2. This is showing the correct figure numbers – 6.1 and 6.2 – but it takes the reader to figures 4.1 and 4.2 on clicking on the pdf; why is this happening? When all the queries and responses are finished, the real world returns the key (k_0, k_1, k_2) to A, whereas the ideal world behaves as depicted in Fig.s 6.3, 6.4 and 6.5.

6.3 Offline Phase of the Ideal World

After the query-response phase, the ideal world samples three n-bit dummy keys (k_0, k_1, k_2), uniformly and independently of all the previously sampled random variables. Then it starts computing the hash value of $\mathsf{pPMAC\_Plus\text{-}Hash}^*$ for all the q queried messages. During this hash computation, if any of the events mentioned in stage 1 of the game (shown in Fig. 6.3) occur, it is aborted. The first event Coll addresses collisions between two inputs to the hash-permutations of a particular construction query and inputs to any forward primitive query. $\mathsf{3\text{-}Coll}$ takes care of collisions of a hash-permutation input from one construction query with one input block each of hash-permutations involved in two other contruction queries. $(\mathsf{Bad}_1\text{-}\mathsf{Bad}_3)$ occur when there is a collision in both invocations

```
Real-Online
$\pi^0 \leftarrow \pi|_{\hat{u}\|0}\ \forall \hat{u} \in \{0,1\}^{n-1},\ \pi^1 \leftarrow \pi|_{\hat{w}\|1}\ \forall \hat{w} \in \{0,1\}^{n-1}.$
$\mathsf{Dom}(\pi^0) \leftarrow \phi, \mathsf{Dom}(\pi^1) \leftarrow \phi, \mathsf{Ran}(\pi^0) \leftarrow \phi, \mathsf{Ran}(\pi^1) \leftarrow \phi.$
$\forall i \in [q]$, on query $M_i$, output $T_i \leftarrow \mathsf{pPMAC\_Plus}^*(M_i)$.
$\forall b \in [p]$ such that $\tilde{u}_b = \hat{u}_b\|0$, on query $(\tilde{u}_b, +)$ to $\pi^0$,
  output $\tilde{v}_b \leftarrow_{\$} \{0,1\}^n \setminus \mathsf{Ran}(\pi^0) \cup \mathsf{Ran}(\pi^1)$.
  $\pi^0(\tilde{u}_b) \leftarrow \tilde{v}_b$.
  $\mathsf{Dom}(\pi^0) \leftarrow \mathsf{Dom}(\pi^0) \cup \{\tilde{u}_b\}$.
  $\mathsf{Ran}(\pi^0) \leftarrow \mathsf{Ran}(\pi^0) \cup \{\tilde{v}_b\}$.
$\forall c \in [p]$ such that $\tilde{w}_c = \hat{w}_c\|1$, on query $(\tilde{w}_c, +)$ to $\pi^1$,
  output $\tilde{z}_c \leftarrow_{\$} \{0,1\}^n \setminus \mathsf{Ran}(\pi^0) \cup \mathsf{Ran}(\pi^1)$.
  $\pi^0(\tilde{w}_c) \leftarrow \tilde{z}_c$.
  $\mathsf{Dom}(\pi^1) \leftarrow \mathsf{Dom}(\pi^1) \cup \{\tilde{w}_c\}$.
  $\mathsf{Ran}(\pi^1) \leftarrow \mathsf{Ran}(\pi^1) \cup \{\tilde{z}_c\}$.
$\forall a \in [p]$, on query $(\tilde{y}_a, -)$ to $\pi$ such that
$\tilde{y}_a \notin \mathsf{Ran}(\pi^0) \cup \mathsf{Ran}(\pi^1)$, output $\tilde{x}_a \leftarrow_{\$} \{0,1\}^n \setminus \left(\mathsf{Dom}(\pi^0) \cup \mathsf{Dom}(\pi^1)\right)$.
    if $\mathsf{LSB}(\tilde{x}_a) = 0$, then $\mathsf{Dom}(\pi^0) \leftarrow \mathsf{Dom}(\pi^0) \cup \{\tilde{x}_a\}, \mathsf{Ran}(\pi^0) \leftarrow \mathsf{Ran}(\pi^0) \cup \{\tilde{y}_a\}$.
    else $\mathsf{Dom}(\pi^1) \leftarrow \mathsf{Dom}(\pi^1) \cup \{\tilde{x}_a\}, \mathsf{Ran}(\pi^1) \leftarrow \mathsf{Ran}(\pi^1) \cup \{\tilde{y}_a\}$.
 $\mathsf{Dom}(\pi) \leftarrow \mathsf{Dom}(\pi^0) \sqcup \mathsf{Dom}(\pi^1), \mathsf{Ran}(\pi) \leftarrow \mathsf{Ran}(\pi^0) \sqcup \mathsf{Ran}(\pi^1)$.
```

Figure 6.1: Description of the online phase of the real world. π^0 is the restriction of the permutation π to the domain $\{\hat{u}\|0 : \hat{u} \in \{0,1\}^{n-1}\}$, and similarly, π^1 is the restriction of the permutation π to the domain $\{\hat{w}\|1 : \hat{w} \in \{0,1\}^{n-1}\}$.

of π involved in the sum function. Note that $\overline{\mathsf{Bad}_2}$ and $\overline{\mathsf{Bad}_3}$ guarantee that a collision of the value Σ_i of the i^{th} construction query with a primitive query $\tilde{x}_a$ ensures freshness of Θ_i, and by symmetry, the same for Σ_i due to a primitive-value collision of Θ_i.
This makes certain that the output $T_i \oplus \tilde{y}_a$ of Θ_i through π remains fresh. However, if $T_i \oplus \tilde{y}_a$ collides with any $\tilde{y}_{a'}$ due to the sampling of T_i, then permutation compatibility is violated. A similar violation arises when $\mathsf{chop}_{\mathsf{LSB}}\left(\Theta_i \oplus k_2\right)\|0$ collides with a primitive query $\tilde{x}_a$, but the output of Σ_i is not fresh. This event is captured in Bad_4. The events $\overline{\mathsf{Bad}_1}$ and $\overline{\mathsf{Bad}_3}$ guarantee that a collision in exactly one half of the hash blocks of two construction queries implies freshness of the other half. This also means that their tags do not collide with each other. However, if they do happen to collide with each other through sampling of the tags, permutation compatibility is again violated, as captured in Bad_5. If the game is not aborted in stage I, it proceeds to stage II.

In this stage, there may exist a set of indices for which exactly one hash block collides with a primitive query. For example, if $\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_i \oplus k_2\right)\|0$ collides with $\tilde{u}_b$ for some $i \in [q]$ and for some $b \in [p]$, then we remove i from $\mathcal{I}$ and add Σ_i to $\widetilde{\Sigma}$ and Θ_i to $\widetilde{\Theta}$, as well as $\mathsf{chop}_{\mathsf{LSB}}\left(\Theta_i \oplus k_2\right)\|1$ to the domain of π and $T_i \oplus \tilde{v}_b$ to the range of π. Similarly, if $\mathsf{chop}_{\mathsf{LSB}}\left(\Theta_i \oplus k_2\right)\|0$ collides with $\tilde{w}_c$ for some $i \in [q]$ and for some $c \in [p]$, we remove i from $\mathcal{I}$ and add Σ_i and Θ_i to $\widetilde{\Sigma}$ and $\widetilde{\Theta}$ respectively, as well as $\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_i \oplus k_2\right)\|0$ to the domain of π and $T_i \oplus \tilde{z}_c$ to the range of π. Note that if $\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_i \oplus k_2\right)\|0$ collides with $\tilde{u}_b$, then Θ_i is fresh as Bad_2 and Bad_3 do not occur. Moreover, $T_i \oplus \tilde{y}_a$ is also fresh as Bad_4 does not occur. Hence, the inclusion of Θ_i in the set $\mathsf{Dom}(\pi^1)$ and $T_i \oplus \tilde{y}_a$ in $\mathsf{Ran}(\pi^1)$ is sound. One can similarly argue that the inclusion of Σ_i in $\mathsf{Dom}(\pi^0)$ and $T_i \oplus y_j^2$ in $\mathsf{Ran}(\pi^0)$ is also sound.

```
Ideal-Online
$\mathsf{Dom}(\pi^0) \leftarrow \phi, \mathsf{Dom}(\pi^1) \leftarrow \phi, \mathsf{Ran}(\pi^0) \leftarrow \phi, \mathsf{Ran}(\pi^1) \leftarrow \phi.$
$\forall i \in [q],$ on query $M_i,$ output $T_i \leftarrow\!\!\$ \{0,1\}^n.$    /* Construction query. */
$\forall b \in [p]$ such that $\tilde{u}_b = \hat{u}_b \| 0,$ on query $(\tilde{u}_b, +)$ to $\pi,$    /* Forward primitive query. */
  output $\tilde{v}_b \leftarrow\!\!\$ \{0,1\}^n \setminus \mathsf{Ran}(\pi^0) \cup \mathsf{Ran}(\pi^1).$
  $\mathsf{Dom}(\pi^0) \leftarrow \mathsf{Dom}(\pi^0) \cup \{\tilde{u}_b\}.$
  $\mathsf{Ran}(\pi^0) \leftarrow \mathsf{Ran}(\pi^0) \cup \{\tilde{v}_b\}.$
$\forall c \in [p]$ such that $\tilde{w}_c = \hat{w}_c \| 1,$ on query $(\tilde{w}_c, +)$ to $\pi,$    /* Backward primitive query. */
  output $\tilde{z}_c \leftarrow\!\!\$ \{0,1\}^n \setminus \mathsf{Ran}(\pi^0) \cup \mathsf{Ran}(\pi^1).$
  $\mathsf{Dom}(\pi^1) \leftarrow \mathsf{Dom}(\pi^1) \cup \{\tilde{w}_c\}.$
  $\mathsf{Ran}(\pi^1) \leftarrow \mathsf{Ran}(\pi^1) \cup \{\tilde{z}_c\}.$
$\forall a \in [p],$ on query $(\tilde{y}_a, -)$ to $\pi$ such that
$\tilde{y}_a \notin \mathsf{Ran}(\pi^0) \cup \mathsf{Ran}(\pi^1),$ output $\tilde{x}_a \leftarrow\!\!\$ \{0,1\}^n \setminus \left(\mathsf{Dom}(\pi^0) \cup \mathsf{Dom}(\pi^1)\right).$
    if $\mathsf{LSB}(\tilde{x}_a) = 0,$ then $\mathsf{Dom}(\pi^0) \leftarrow \mathsf{Dom}(\pi^0) \cup \{\tilde{x}_a\}, \mathsf{Ran}(\pi^0) \leftarrow \mathsf{Ran}(\pi^0) \cup \{\tilde{y}_a\}.$
    else $\mathsf{Dom}(\pi^1) \leftarrow \mathsf{Dom}(\pi^1) \cup \{\tilde{x}_a\}, \mathsf{Ran}(\pi^1) \leftarrow \mathsf{Ran}(\pi^1) \cup \{\tilde{y}_a\}.$
$\mathsf{Dom}(\pi) \leftarrow \mathsf{Dom}(\pi^0) \sqcup \mathsf{Dom}(\pi^1), \mathsf{Ran}(\pi) \leftarrow \mathsf{Ran}(\pi^0) \sqcup \mathsf{Ran}(\pi^1).$
```

Figure 6.2: Description of the online phase of the ideal world. π^0 is the restriction of the permutation π to the domain $\{\hat{u}\|0 : \hat{u} \in \{0,1\}^{n-1}\}$, and similarly, π^1 is the restriction of the permutation π to the domain $\{\hat{w}\|1 : \hat{w} \in \{0,1\}^{n-1}\}$.

For the remaining $q - |\mathcal{I}|$ indices, there may exist a set of free indices $\mathcal{F}$ for which both blocks of the hash value are fresh in the set of $2(q - |\mathcal{I}|)$ hash block values. The oracle samples outputs for these fresh hash values without replacement such that for any $i \in \mathcal{F}$, the sampled outputs v_i and z_i sum up to T_i.

The cases remaining in stage III are those for which exactly one block of the hash value collides with that of another construction query. For all $i \in [q] \setminus (\mathcal{F} \sqcup \mathcal{I})$, if the output of the colliding hash value, say Σ_i, is not yet sampled, then the oracle samples its output without replacement, say v_i and sets the output of the remaining block, i.e., the output of Θ_i as the sum of v_i and T_i (see line 2 of stage III). Else, the oracle sets the output of Σ_i to the already defined element and adjusts the output of the other block accordingly (see line 3 of stage III). Note that in the latter case, the oracle does not sample the output. If the output of Θ_i (i.e., $T_i \oplus v_i$) happens to collide with any previously sampled output or any element of $\mathsf{Ran}(\pi^1)$ in the above argument, then RC_Σ is set to 1 (see line 4 of stage III) and aborts the game. Similarly, the oracle sets RC_Θ to 1 if the adjustment of the output of Σ_i causes a collision with any previously sampled output or any element of $\mathsf{Ran}(\pi^0)$. Note that these events cannot hold for the real oracle as at least one of Θ_i or Σ_i is always fresh in the tuple of $2(q - |\mathcal{I}|)$ hash block values. Finally, it returns all these sampled values along with the sampled hash key to the distinguisher A.

6.4 Attack transcript

Let $\tau_c := \{(M_1, T_1), (M_2, T_2), \ldots, (M_q, T_q)\}$ be the list of construction queries and responses made by A. We call τ_c the *construction query transcript.* Let $\tau_p := \{(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_{2p}, \tilde{y}_{2p})\}$ be the list of primitive queries and responses made to π by A. The pair (τ_c, τ_p) constitutes the query transcript of the attack. For convenience, we slightly modify the experiment by revealing the keys (k_0, k_1, k_2) and internal or random values to the distinguisher A (only after it completes making all its queries but before it outputs its decision) in addition to

```
IDEAL-OFFLINE: STAGE I
$(k_0, k_1, k_2) \leftarrow_{\$} (\{0,1\}^n)^3$.
if $\exists i \in [q], \alpha \neq \beta$ in $[l_i]$ and $a_1 \neq a_2$ for which $\tilde{x}_{a_1}, \tilde{x}_{a_2} \in \mathsf{Dom}(\pi)$ :
    $\big(M_i[\alpha] \oplus 2^{\alpha} k_0 \oplus 2^{2\alpha} k_1 = \tilde{x}_{a_1}\big) \wedge \big(M_i[\beta] \oplus 2^{\beta} k_0 \oplus 2^{2\beta} k_1 = \tilde{x}_{a_2}\big)$ , then $\boxed{\mathsf{Coll} \leftarrow 1,}$ $\perp$ .
if $\exists i_1, i_2, i_3 \in [q]$, and distinct $\alpha_1 \in [l_{i_1}], \alpha_2 \in [l_{i_2}], \alpha_3 \in [l_{i_3}]$ :
    $\big(M_{i_1}[\alpha_1] \oplus 2^{\alpha_1} k_0 \oplus 2^{2\alpha_1} k_1 = M_{i_2}[\alpha_2] \oplus 2^{\alpha_2} k_0 \oplus 2^{2\alpha_2} k_1\big)$
    $\wedge \big(M_{i_1}[\alpha_1] \oplus 2^{\alpha_1} k_0 \oplus 2^{2\alpha_1} k_1 = M_{i_3}[\alpha_3] \oplus 2^{\alpha_3} k_0 \oplus 2^{2\alpha_3} k_1\big)$ , then $\boxed{\mathsf{3\text{-}Coll} \leftarrow 1,}$ $\perp$ .
$\forall i \in [q],\ (\Sigma_i, \Theta_i) \leftarrow \mathsf{pPMAC\_Plus\text{-}Hash}^{\pi}_{k_0,k_1,k_2}(M_i)$.          / * See subroutine 4.1 * /
$\widetilde{\Sigma} \leftarrow \{\Sigma_1, \ldots, \Sigma_q\}$ , $\widetilde{\Theta} \leftarrow \{\Theta_1, \ldots, \Theta_q\}$ .
if $\exists i_1, i_2, i_3 \in [q]$ with $i_2 \neq i_1, i_3 \neq i_1$ :
    $(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2) \| 0 = \mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_2} \oplus k_2) \| 0)$
    $\wedge (\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2) \| 1 = \mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_3} \oplus k_2) \| 1)$ , then $\boxed{\mathsf{Bad}_1 \leftarrow 1,}$ $\perp$ .
 if $\exists i \in [q]$ :
    $\big(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2) \| 0 \in \mathsf{Dom}(\pi^0)\big) \wedge \big(\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2) \| 1 \in \mathsf{Dom}(\pi^1)\big)$ ,
    then $\boxed{\mathsf{Bad}_2 \leftarrow 1,}$ $\perp$ .
 if $\exists i_1 \neq i_2 \in [q]$ :
    $\big[(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2) \| 0 = \mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_2} \oplus k_2) \| 0) \wedge \big(\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2) \| 1 \in \mathsf{Dom}(\pi^1)\big)\big]$
    $\vee \big[(\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2) \| 1 = \mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_2} \oplus k_2) \| 1) \wedge \big(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2) \| 0 \in \mathsf{Dom}(\pi^0)\big)\big]$ ,
   then $\boxed{\mathsf{Bad}_3 \leftarrow 1,}$ $\perp$ .
 if $\exists i \in [q], b, c \in [p]$ :
    $[(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2) \| 0 = \tilde{u}_b) \wedge (T_i \oplus \tilde{v}_b = \tilde{z}_c)]$
    $\vee [(\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2) \| 1 = \tilde{w}_c) \wedge (T_i \oplus \tilde{z}_c = \tilde{v}_b)]$ , then $\boxed{\mathsf{Bad}_4 \leftarrow 1,}$ $\perp$ .
 if $\exists$ distinct $i_1, i_2 \in [q]$ :
    $[(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2) \| 0 = \mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_2} \oplus k_2) \| 0) \wedge (T_{i_1} = T_{i_2})]$
    $\vee [(\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2) \| 1 = \mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_2} \oplus k_2) \| 1) \wedge (T_{i_1} = T_{i_2})]$ , then $\boxed{\mathsf{Bad}_5 \leftarrow 1,}$ $\perp$ .
 if $\exists i_1, i_2, i_3 \in [q], b, c \in [p]$ and $\alpha in [l_{i_2}], \beta in [l_{i_3}]$ :
    $([\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2) \| 0 = x_{i_2}[\alpha]] \wedge [\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2) \| 1 = x_{i_3}[\beta]]) \vee$
    $([\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2) \| 0 = x_{i_2}[\alpha]] \wedge [\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2) \| 1 = \tilde{w}_c]) \vee$
    $([\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2) \| 0 = x_{i_2}[\alpha]] \wedge [\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2) \| 1 = \mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_3} \oplus k_2) \| 1]) \vee$
    $([\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2) \| 0 = \tilde{u}_b] \wedge [\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2) \| 1 = x_{i_3}[\beta]]) \vee$
    $([\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2) \| 0 = \mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2) \| 0] \wedge [\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2) \| 1 = x_{i_3}[\beta]])$ ,
    then $\boxed{\mathsf{Bad}_6 \leftarrow 1,}$ $\perp$ .
 go to stage II .
```

Figure 6.3: Stage I of the offline phase of the ideal oracle. The internal values $x_i[\alpha]$ are as defined in Eqn. (3).

responses to the queries it makes. If A interacts with the real world, then the actual key of the construction is revealed along with the permutation outputs of the hash output blocks Σ and Θ, whereas for the ideal world, a triplet of dummy n-bit keys (k_0, k_1, k_2) is revealed. The construction query transcript of the attack is thus

$$\hat{\tau}_c = ((M_1, T_1, v_1, z_1), (M_2, T_2, v_2, z_2), \ldots, (M_q, T_q, v_q, z_q), k_0, k_1, k_2)\,.$$

Ideal-Offline: Stage II

```
$\forall i \in [q]$ if $(\exists b \in [p] : \mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2)\|0 = \tilde{u}_b) \vee$
$(\exists i_2 \in [q]$ and $\alpha \in [l_{i_2}] : \mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2)\|0 = x_{i_2}[\alpha])$, then
    $\widetilde{\Sigma} \leftarrow \widetilde{\Sigma} \setminus \Sigma_i$ and $\mathcal{I} \leftarrow \mathcal{I} \cup \{i\}$.
    $\mathsf{Dom}(\pi^1) \leftarrow \mathsf{Dom}(\pi^1) \cup \{\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2)\|1\}$.
    $\mathsf{Ran}(\pi^1) \leftarrow \mathsf{Ran}(\pi^1) \cup \{T_i \oplus \tilde{v}_b\}$.
$\forall i \in [q]$if $(\exists c \in [p] : \mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2)\|1 = \tilde{w}_c) \vee$
$(\exists i_2 \in [q]$ and $\alpha \in [l_{i_2}] : \mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2)\|1 = x_{i_2}[\alpha])$, then
    $\widetilde{\Theta} \leftarrow \widetilde{\Theta} \setminus \Theta_i$ and $\mathcal{I} \leftarrow \mathcal{I} \cup \{i\}$.
    $\mathsf{Dom}(\pi^0) \leftarrow \mathsf{Dom}(\pi^0) \cup \{\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2)\|0\}$.
    $\mathsf{Ran}(\pi^0) \leftarrow \mathsf{Ran}(\pi^0) \cup \{T_i \oplus \tilde{z}_c\}$.
$\mathcal{F} \leftarrow \{i \in [q] \setminus \mathcal{I} : (\Sigma_i \neq \Sigma_{i'}) \wedge (\Theta_i \neq \Theta_{i''})$ for any $i', i'' \neq i$ in $[q] \setminus \mathcal{I}\}$.
$f \leftarrow |\mathcal{F}|$.
$v_i \leftarrow\!\!\$\, \{0,1\}^n \setminus \mathsf{Ran}(\pi^0)\ \forall i \in \mathcal{F}$.
$\mathcal{S} \leftarrow \{(v_i, z_i) \in \{0,1\}^n \setminus \mathsf{Ran}(\pi^0) \times \{0,1\}^n \setminus \mathsf{Ran}(\pi^1) : v_i \oplus z_i = T_i\ \forall i \in \mathcal{F}\}$.
for $(v_i, z_i) \leftarrow\!\!\$\, \mathcal{S}$ :
 set $\Pi^0(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2)\|0) \leftarrow v_i, \Pi^1(\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2)\|1) \leftarrow z_i$.
    $\mathsf{Dom}(\Pi^0) \leftarrow \mathsf{Dom}(\Pi^0) \cup \{\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2)\|0,\}$, $\mathsf{Ran}(\Pi^0) \leftarrow \mathsf{Ran}(\Pi^0) \cup \{v_i\}$.
    $\mathsf{Dom}(\Pi^1) \leftarrow \mathsf{Ran}(\Pi^1) \cup \{\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2)\|1\}$, $\mathsf{Ran}(\Pi^0) \leftarrow \mathsf{Ran}(\Pi^0) \cup \{v_i\}$.
go to stage III.
```

Figure 6.4: Stage II of the offline phase of the ideal oracle.

Therefore, the query transcript of the attack is $\tau = (\hat{\tau}_c, \tau_p)$, where τ_p can further be partitioned into $\tau_p^0 := \{(\tilde{u}_b, \tilde{v}_b) : \tilde{u}_b = \hat{u}_b\|0$ where $\hat{u}_b \in \{0,1\}^{n-1}, \forall b \in [p]\}$ and $\tau_p^1 := \{(\tilde{w}_c, \tilde{z}_c) : \tilde{w}_c = \hat{w}_c\|1$ where $\hat{w}_c \in \{0,1\}^{n-1}, \forall c \in [p]\}$. Note that if A interacts with the real world, then

$$\begin{aligned}\forall i \in [q], v_i &:= \pi^0(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2)\|0) := \pi(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2)\|0),\\ z_i &:= \pi^1(\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2)\|1) := \pi(\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2)\|1),\end{aligned}$$

where $(\Sigma_i, \Theta_i) := \mathsf{pPMAC\_Plus\text{-}Hash}^{\pi}_{k_0,k_1,k_2}(M_i)$. Moreover, a transcript τ in the real world must satisfy the following conditions:

- $v_i \oplus z_i = T_i, \forall i \in [q]$.
- $\forall a \in [2p], \pi(\tilde{x}_a) = \tilde{y}_a$ such that $\forall b \in [p], \pi(\tilde{u}_b) = \tilde{v}_b$ and $\forall c \in [p], \pi(\tilde{w}_c) = \tilde{z}_c$, where $\tilde{u}_b = \hat{u}_b\|0$ and $\tilde{w}_c = \hat{w}_c\|1$ for $\hat{u}_b, \hat{w}_c \in \{0,1\}^{n-1}$.
- $\widetilde{\Sigma}$ is permutation compatible with $\widetilde{v}$ and $\widetilde{\Theta}$ is permutation compatible with $\widetilde{z}$ (note that $(\widetilde{\Sigma}, \widetilde{\Theta})$ is uniquely determined by the message tuple $(M_1, \ldots, M_q)$, the tuple of keys k_0, k_1, k_2 and the public random permutation π).

6.5 Definition and Probability of Bad Transcripts

Suppose $\mathcal{X}$ denotes the set of all attainable transcripts and D_{re} and D_{id} the random variables that take transcript τ induced in the real world and ideal world respectively. An attainable transcript $\tau \in \mathcal{X}$ is said to be *bad* if either of the following bad flags

$$\mathsf{Coll}, \mathsf{3\text{-}Coll}, \mathsf{Bad}_1, \mathsf{Bad}_2, \mathsf{Bad}_3, \mathsf{Bad}_4, \mathsf{Bad}_5, \mathsf{Bad}_6, \mathsf{RC}_\Sigma, \mathsf{RC}_\Theta$$

Ideal-Offline: Stage III

```
$\forall i \in [q],\ \mathsf{Dom}(\pi) \leftarrow \mathsf{Dom}(\pi^0) \cup \mathsf{Dom}(\pi^1) \cup \left\{M_i[\alpha] \oplus 2^{\alpha} k_0 \oplus 2^{2\alpha} k_1 \ :\ \alpha \in [l_i]\right\}$,
 $\mathsf{Ran}(\pi) \leftarrow \mathsf{Ran}(\pi^0) \cup \mathsf{Ran}(\pi^1) \cup \left\{\pi\left(M_i[\alpha] \oplus 2^{\alpha} k_0 \oplus 2^{2\alpha} k_1\right) \ :\ \alpha \in [l_i]\right\}$.
$\forall i \in [q] \setminus (\mathcal{F} \sqcup \mathcal{I})$ such that $\exists \Sigma_{i'} \in \widetilde{\Sigma}$ with $\Sigma_i = \Sigma_{i'}$,
  if $\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2) \| 0 \notin \mathsf{Dom}(\pi) \cup \mathsf{Dom}(\Pi^0)$,
  then $\Pi^0(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2) \| 0) \leftarrow v_i \leftarrow_{\$} \{0,1\}^n \setminus \mathsf{Ran}(\Pi^0) \cup \mathsf{Ran}(\pi^0)$ and $z_i \leftarrow T_i \oplus v_i$.
  else $v_i \leftarrow \Pi^0(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2 \| 0)$ and $z_i \leftarrow T_i \oplus v_i$.
    $\mathsf{Dom}\left(\Pi^0\right) \leftarrow \mathsf{Dom}\left(\Pi^0\right) \cup \{\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2) \| 0, \}$, $\mathsf{Ran}\left(\Pi^0\right) \leftarrow \mathsf{Ran}\left(\Pi^0\right) \cup \{v_i\}$.
    $\mathsf{Dom}\left(\Pi^1\right) \leftarrow \mathsf{Dom}\left(\Pi^1\right) \cup \{\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2) \| 1\}$, $\mathsf{Ran}\left(\Pi^1\right) \leftarrow \mathsf{Ran}\left(\Pi^1\right) \cup \{v_i\}$.
  if $z_i \in \mathsf{Ran}(\Pi^1) \cup \mathsf{Ran}(\pi^1)$, then $\boxed{\mathsf{RC}_{\Sigma} \leftarrow 1}$, $\Pi^1(\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2) \| 1) \leftarrow z_i, \perp$ .
$\forall i \in [q] \setminus (\mathcal{F} \sqcup \mathcal{I})$ such that $\exists \Theta_{i''} \in \widetilde{\Theta}$ with $\Theta_i = \Theta_{i''}$,
  if $\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2) \| 1 \notin \mathsf{Dom}(\pi) \cup \mathsf{Dom}(\Pi^1)$,
  then $\Pi^1(\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2) \| 1) \leftarrow z_i \leftarrow_{\$} \{0,1\}^n \setminus \mathsf{Ran}(\Pi^1) \cup \mathsf{Ran}(\pi^1)$ and $z_i \leftarrow T_i \oplus z_i$.
  else $z_i \leftarrow \Pi^1(\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2 \| 1)$ and $v_i \leftarrow T_i \oplus z_i$.
    $\mathsf{Dom}\left(\Pi^0\right) \leftarrow \mathsf{Dom}\left(\Pi^0\right) \cup \{\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2) \| 0, \}$, $\mathsf{Ran}\left(\Pi^0\right) \leftarrow \mathsf{Ran}\left(\Pi^0\right) \cup \{v_i\}$.
    $\mathsf{Dom}\left(\Pi^1\right) \leftarrow \mathsf{Ran}\left(\Pi^1\right) \cup \{\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2) \| 1\}$, $\mathsf{Ran}\left(\Pi^0\right) \leftarrow \mathsf{Ran}\left(\Pi^0\right) \cup \{v_i\}$.
  if $v_i \in \mathsf{Ran}(\Pi^0) \cup \mathsf{Ran}(\pi^0)$, then $\boxed{\mathsf{RC}_{\Theta} \leftarrow 1}$, $\Pi^0(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2) \| 0) \leftarrow v_i, \perp$ .
$\mathsf{Dom}(\pi) \leftarrow \mathsf{Dom}(\pi^0) \sqcup \mathsf{Dom}(\pi^1)$, $\mathsf{Ran}(\pi) \leftarrow \mathsf{Ran}(\pi^0) \sqcup \mathsf{Ran}(\pi^1)$.
$\mathsf{Dom}(\Pi) \leftarrow \mathsf{Dom}(\Pi^0) \sqcup \mathsf{Dom}(\Pi^1)$, $\mathsf{Ran}(\Pi) \leftarrow \mathsf{Ran}(\Pi^0) \sqcup \mathsf{Ran}(\Pi^1)$.
```

Figure 6.5: Stage III of the offline phase of the ideal oracle. Boxed statements denote bad events. Whenever a bad event is set to 1, the game gets immediately aborted (denoted $\perp$) and returns the remaining values of the transcript arbitrarily.

is set to 1 as defined in Fig. 6.3. We define the event Bad as

$$\mathsf{Coll} \vee \mathsf{3\text{-}Coll} \vee \left(\vee_{i=1}^{6} \underbrace{(\mathsf{Bad}_i \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}})}_{\mathsf{Bad}_i^*} \right) \vee \underbrace{(\mathsf{RC}_{\Sigma} \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}})}_{\mathsf{RC}_{\Sigma}^*} \vee \underbrace{(\mathsf{RC}_{\Theta} \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}})}_{\mathsf{RC}_{\Theta}^*} .$$

Thus, $\mathsf{BadT} := \{((M_i, T_i, v_i, z_i), (\tilde{x}_a, \tilde{y}_a)) \in (\hat{\tau}_c, \tau_p) \ :\ ((M_i, T_i, v_i, z_i), (\tilde{x}_a, \tilde{y}_a))$ satisfies at least one condition boxed in Fig. 6.3$\} \subseteq \mathcal{X}$ and $\mathsf{GoodT} := \mathcal{X} \setminus \mathsf{BadT}$ denote the set of bad and good transcripts, respectively. Having identified the bad transcripts, we bound the probability of realizing them in the ideal world in the following lemma.

Lemma 4. *Let* BadT *be the set of all attainable bad transcripts and* D_{id} *be the random variable that takes a transcript* τ *induced in the ideal world. Then*

$$\begin{aligned}\Pr[\mathsf{D}_{\mathrm{id}} \in \mathsf{BadT}] \leq \epsilon_{\mathrm{bad}} \ &= \ \frac{2\sqrt{3nqp^2} + 4}{2^n} + \frac{q^3(10\ell^3 + 5\ell^2 + 4\ell + 8)}{2^{2n}} \\ &\quad + \frac{q^2 p(2\ell + 9)}{2^{2n}} + \frac{qp^2(11\ell^2 + 4\ell + 8)}{2^{2n}} + \frac{q^2(2\ell + 5)}{2^{2n}}.\end{aligned}$$

Proof. Bounding the probability of the bad transcripts in the ideal world is equivalent to bounding the probability of the event Bad in the ideal world. Due to the union bound,

$$\Pr[\mathsf{Bad}] \quad \leq \quad \Pr[\mathsf{Coll}] + \Pr[\mathsf{3\text{-}Coll}] + \sum_{i=1}^{6} \Pr[\mathsf{Bad}_i^*] + \Pr[\mathsf{RC}_{\Sigma}^*] + \Pr[\mathsf{RC}_{\Theta}^*]. \tag{4}$$

In the following, we separately bound each of the above terms. By a slight abuse of notation, we use the flag names to identify the corresponding event. Before we bound the terms, we set up a few notations.

Notational set-up. Let $\mathcal{U} = \{\tilde{x}_a \in \{0,1\}^n : (\tilde{x}_a, \tilde{y}_a) \in \tau_p\}$ and $\mathcal{V} = \{\tilde{y}_a \in \{0,1\}^n : (\tilde{x}_a, \tilde{y}_a) \in \tau_p\}$ be the domain and range of the transcript of π. Let $(M_1, \ldots, M_q)$ be a tuple of q distinct messages such that the i^{th} message M_i has ℓ_i blocks with $\ell = \max\{l_1, \ldots, l_q\}$, being the maximum number of message blocks amongst all the q messages. For two distinct fixed indices $i_1, i_2 \in [q]$, we define the set

$$\mathsf{NEQ}_{i_1,i_2} = \{\alpha \in \min[l_{i_1}, l_{i_2}] : M_{i_1}[\alpha] \neq M_{i_2}[\alpha]\} \cup \{\alpha : \min[l_{i_1}, l_{i_2}] + 1 \leq \alpha \leq \max[l_{i_1}, l_{i_2}]\}.$$

In words, NEQ_{i_1,i_2} refers to the set of all positions at which inputs to the hash permutation π from message blocks of M_{i_1} and M_{i_2} differ. We denote the inputs (resp. outputs) of these permutation instances as x_i (resp. y_i). In particular, we write $x_i[\alpha]$ to denote the permutation input corresponding to the α^{th} block of the i^{th} message, i.e. $x_i[\alpha] = M_i[\alpha] \oplus 2^\alpha k_0 \oplus 2^{2\alpha} k_1$ and $y_i[\alpha] = \pi(x_i[\alpha])$.

Bounding Coll. For a fixed choice of $i \in [q], \alpha \neq \beta$ in $[l_i]$ and $a_1, a_2 \in [p]$, the system of equations

$$\begin{aligned} 2^\alpha k_0 \oplus 2^{2\alpha} k_1 &= M^i[\alpha] \oplus \tilde{x}_{a_1}, \\ 2^\beta k_0 \oplus 2^{2\beta} k_1 &= M^i[\beta] \oplus \tilde{x}_{a_2} \end{aligned}$$

has rank 2. Since k_0 and k_1 are two independent n-bit keys, varying over all possible choices of indices gives

$$\Pr[\mathsf{Coll}] \leq \frac{qp^2\ell^2}{2^{2n+1}}. \tag{5}$$

Bounding 3-Coll. For a fixed choice of $i_1, i_2, i_3 \in [q]$, and distinct $\alpha_1 \in [l_{i_1}], \alpha_2 \in [l_{i_2}], \alpha_3 \in [l_{i_3}]$, the system of equations

$$\begin{aligned} (2^{\alpha_1} \oplus 2^{\alpha_2}) k_0 \oplus (2^{2\alpha_1} \oplus 2^{2\alpha_2}) k_1 &= M_{i_1}[\alpha_1] \oplus M_{i_2}[\alpha_2], \\ (2^{\alpha_1} \oplus 2^{\alpha_3}) k_0 \oplus (2^{2\alpha_1} \oplus 2^{2\alpha_3}) k_1 &= M_{i_1}[\alpha_1] \oplus M_{i_3}[\alpha_3] \end{aligned}$$

has rank 2. Since k_0 and k_1 are two independent n-bit keys, varying over all possible choices of indices gives

$$\Pr[\mathsf{3\text{-}Coll}] \leq \frac{q^3\binom{3\ell}{3}}{2^{2n}} \leq \frac{5q^3\ell^3}{2^{2n}}. \tag{6}$$

Bounding Event Bad_1^*. We fix three messages M_{i_1}, M_{i_2} and M_{i_3} where $i_1 \neq i_2, i_1 \neq i_3$, such that M_{i_1} has l_{i_1} blocks, M_{i_2} has l_{i_2} blocks and M_{i_3} has l_{i_3} blocks. Consider the event

$$\mathsf{CollX}^{(1)} : \{\exists\, j_1, j_2 \in \{i_1, i_2, i_3\} \text{ and } \alpha \in [l_{j_1}], \beta \in [l_{j_2}], \text{ such that } x_{j_1}[\alpha] = x_{j_2}[\beta]\}.$$

Therefore,

$$\begin{aligned} \Pr[\mathsf{Bad}_1^*] \leq \sum_{i_1,i_2,i_3} \Big(& \underbrace{\Pr[\Theta_{i_1} = \Theta_{i_3} \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \mathsf{CollX}^{(1)}]}_{(1)} \\ & + \underbrace{\Pr[\Sigma_{i_1} = \Sigma_{i_2} \wedge \Theta_{i_1} = \Theta_{i_3} \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \overline{\mathsf{CollX}^{(1)}}]}_{(2)} \Big). \end{aligned} \tag{7}$$

Bounding (1): It is easy to see that for a fixed triplet of messages, the probability of $\mathsf{CollX}^{(1)}$ is at most $\binom{3\ell}{2}/2^n$. Under this condition, $\Theta_{i_1} = \Theta_{i_3}$ provides a non-trivial equation for some random variable $y_{i'}[\alpha']$. Assuming $l_{i_1} \leq l_{i_3}$, let $\alpha \in [l_{i_1}]$ (if it exists) be the largest index such that $M_{i_1}[\alpha] \neq M_{i_3}[\alpha]$. Then either $y_{i_1}[\alpha]$ or $y_{i_3}[\alpha]$ is fresh and the equation $\Theta_{i_1} = \Theta_{i_3}$ is non-trivial for this random variable. On the other hand, if no such index α exists(i.e. $M_{i_1}[\alpha] = M_{i_3}[\alpha]$ for all $\alpha \in [l_{i_1}]$ and $l_{i_1} < l_{i_3}$), we can obtain a freshly sampled random variable $y_{i_3}[\beta]$, for which $\Theta_{i_1} = \Theta_{i_3}$ becomes non-trivial. Therefore, the probability that this equation is satisfied is at most $1/(2^n - 2\ell) \leq 2/2^n$, assuming $\ell \leq 2^{n-2}$, giving (1) an upper bound of $\binom{3\ell}{2}/2^n \cdot 2/2^n \leq 9\ell^2/2^{2n}$:

$$\Pr[\Theta_i = \Theta_k \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \mathsf{CollX}^{(1)}] \leq \frac{9\ell^2}{2^{2n}}. \tag{8}$$

Bounding (2): We split this case into the following two subcases:

$i_2 = i_3$: Without loss of generality, assume $l_{i_1} \leq l_{i_2}$. Note that if $l_{i_1} = l_{i_2}$, then l_{i_2} must be at least 2 for $\Sigma_{i_1} = \Sigma_{i_2}$ to yield a non-trivial equation. In this case, we can easily find two freshly sampled random variables $y_{i_1}[\alpha]$ and $y_{i_2}[\beta]$ for which $(\Sigma_{i_1} = \Sigma_{i_2}) \wedge (\Theta_{i_1} = \Theta_{i_2})$ yields a system of equations of rank 2. Hence by the rank argument (i.e. Lemma 3),

$$\Pr[(\Sigma_{i_1} = \Sigma_{i_2}) \wedge (\Theta_{i_1} = \Theta_{i_2}) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \overline{\mathsf{CollX}^{(1)}}] \leq \frac{1}{(2^n - 2\ell)_2}. \tag{9}$$

In the particular case when $l_{i_1} + 1 = l_{i_2}$ and $\mathsf{NEQ}_{i_1 i_2} = \{l_{i_2}\}$, if $x_{i_2}[l_{i_2}] = \tilde{x}_a$ for some $a \in [p]$, then $\Sigma_{i_1} = \Sigma_{i_2}$ and $\Theta_{i_1} = \Theta_{i_2}$ would boil down to

$$\begin{aligned} &\tilde{y}_a = 0^n, \\ &\left(2^{l_{i_1}} \oplus 2^{l_{i_1}+1}\right) y_{i_1}[1] \oplus \ldots \left(2 \oplus 2^2\right) y_{i_1}[l_{i_1}] \oplus 2\tilde{y}_a = 0^n. \end{aligned} \tag{10}$$

As the second equation in (10) is non-trivial, and $x_{i_2}[l_{i_2}] = \tilde{x}_a$ holds with probability at most $1/2^n$ (the number of choices for $\tilde{x}_a$ is 1),

$$\Pr[(\Sigma_{i_1} = \Sigma_{i_2}) \wedge (\Theta_{i_1} = \Theta_{i_2}) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \overline{\mathsf{CollX}^{(1)}}] \leq \frac{1}{2^n(2^n - 2\ell)}. \tag{11}$$

In case $l_{i_2} \geq l_{i_1} + 2$, we either determine $\beta_1, \beta_2 \in \{l_{i_1} + 1, \ldots, l_{i_2}\}$ or $\beta_1 \in [l_{i_1}], \beta_2 \in \{l_{i_1} + 1, \ldots, l_{i_2}\}$ such that $y_{i_2}[\beta_1]$ and $y_{i_2}[\beta_2]$ are freshly sampled. In both instances, $(\Sigma_{i_1} = \Sigma_{i_2}) \wedge (\Theta_{i_1} = \Theta_{i_2})$ would result in a system of equations having rank 2, and hence by the rank argument (i.e. Lemma 3),

$$\Pr[(\Sigma_{i_1} = \Sigma_{i_2}) \wedge (\Theta_{i_1} = \Theta_{i_2}) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \overline{\mathsf{CollX}^{(1)}}] \leq \frac{1}{(2^n - 2\ell)_2}. \tag{12}$$

Combining Eqn.s (9), (11) and (12), and assuming $\ell + 1 \leq 2^{n-2}$, we have

$$\Pr[(\Sigma_{i_1} = \Sigma_{i_2}) \wedge (\Theta_{i_1} = \Theta_{i_2}) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \overline{\mathsf{CollX}^{(1)}}] \leq \frac{10}{2^{2n}}. \tag{13}$$

$i_2 \neq i_3$: We approach this in five parts, the first four addressing cases when either M_{i_1} is a prefix of one of M_{i_2} and M_{i_3}, or one of M_{i_2} and M_{i_3} is a prefix of M_{i_1}, and the fifth when neither of the first four occur.

M_{i_1} is a prefix of M_{i_2}: Let $l_{i_2} = l_{i_1} + 1$ and $x_{i_2}[l_{i_2}] = \tilde{x}_a$ for some $a \in [p]$. Then $\Theta_{i_1} = \Theta_{i_3}$ becomes a non-trivial equation, contributing a term $1/(2^n - 3\ell)$ to the bound. An additional $1/2^n$ is contributed by the event $x_{i_2}[l_{i_2}] = \tilde{x}_a$ (as the number of choices for $\tilde{x}_a$ is 1). Assuming $\ell \leq 2^{n-1}/3$, the bound is thus $2/2^{2n}$. On the other hand, if $x_{i_2}[l_{i_2}]$ is fresh, then a freshly sampled random variable $y_{i_1}[\star]$ can be found such that $\Theta_{i_1} = \Theta_{i_3}$ becomes a non-trivial equation. Therefore,

$\Sigma_{i_1} = \Sigma_{i_2}, \Theta_{i_1} = \Theta_{i_3}$ becomes a system of equations of rank 2 (in $y_{i_2}[l_{i_2}]$ and $y_{i_1}[\star]$), and hence by the rank argument (i.e. Lemma 3), we bound the probability of the event by $1/(2^n - 3\ell)_2 \leq 4/2^{2n}$, assuming $3\ell + 1 \leq 2^{n-1}$.

If $l_{i_2} \geq l_{i_1} + 2$, then it is easy to find an index $\beta \in \{l_{i_1} + 1, \ldots, l_{i_2}\}$ such that $y_{i_2}[\beta]$ is freshly sampled. Moreover, we can find another index $\alpha \in [l_{i_1}]$ (or $\alpha \in [l_{i_3}]$) such that $y_{i_1}[\alpha]$ (or $y_{i_3}[\alpha]$) is freshly sampled. In both cases, $\Sigma_{i_1} = \Sigma_{i_2}, \Theta_{i_1} = \Theta_{i_3}$ becomes a system of equations of rank 2. Therefore, by the rank argument (i.e. Lemma 3) and assuming $3\ell + 1 \leq 2^{n-1}$, the probability of the event becomes at most $4/2^{2n}$. Thus,

$$\Pr[\Sigma_{i_1} = \Sigma_{i_2} \wedge \Theta_{i_1} = \Theta_{i_2} \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \overline{\mathsf{CollX}^{(1)}}] \leq \frac{10}{2^{2n}}. \tag{14}$$

The other subcases can be argued similarly and their probabilities bounded above by $10/2^{2n}$.

We now assume that neither is M_{i_1} a prefix of M_{i_2} or M_{i_3}, and nor the reverse. In this case, we can find an index α such that $M_{i_1}[\alpha] \neq M_{i_2}[\alpha]$ and $y_{i_1}[\alpha]$ is freshly sampled. Moreover, we can find another index β such that $M_{i_1}[\beta] \neq M_{i_3}[\beta]$ and $y_{i_3}[\beta]$ is freshly sampled. $\Sigma_{i_1} = \Sigma_{i_2}, \Theta_{i_1} = \Theta_{i_3}$ is a system of equations of rank 2 in these two variables, and hence by the rank argument (i.e. Lemma 3) and by assuming $3\ell + 1 \leq 2^{n-1}$,

$$\Pr[\Sigma_{i_1} = \Sigma_{i_2} \wedge \Theta_{i_1} = \Theta_{i_2} \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \overline{\mathsf{CollX}^{(1)}}] \leq \frac{4}{2^{2n}}. \tag{15}$$

Therefore, combining Eqn.s (14) and (15), the assumption $3\ell + 1 \leq 2^{n-1}$ gives

$$\Pr[\Sigma_{i_1} = \Sigma_{i_2} \wedge \Theta_{i_1} = \Theta_{i_2} \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \overline{\mathsf{CollX}^{(1)}}] \leq \frac{14}{2^{2n}}. \tag{16}$$

Finally, varying over all choices of $i_1, i_2, i_3 \in [q]$ and comibining Eqn.s (7), (8), (13) and (16) with the assumption that $3\ell + 1 \leq 2^{n-1}$, we have

$$\Pr[\mathsf{Bad}_1^*] \leq \frac{3q^3\ell^2}{2^{2n+1}} + \frac{4q^3}{2^{2n}}. \tag{17}$$

<u>Bounding Event Bad_2^*.</u> For fixed indices $i \in [q]$ and $b, c \in [p]$, the event

$$(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2) \| 0 = \tilde{u}_b) \wedge (\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2) \| 1 = \tilde{w}_c)$$

can be reduced to the following system of equations:

$$\begin{aligned} y_i[1] \oplus y_i[2] \oplus \ldots \oplus y_i[l_i] \oplus k_2 &= \tilde{u}_b, \\ 2^{l_i} y_i[1] \oplus 2^{l_i - 1} y_i[2] \oplus \ldots \oplus 2y_i[l_i] \oplus k_2 &= \tilde{w}_c. \end{aligned} \tag{18}$$

We split Bad_2^* into the following two cases:

Case (1): Suppose $l_i = 1$ and $M_i[1] \oplus 2k_0 \oplus 2^2 k_1$ collides with a primitive query input $\tilde{x}_{a'}$ for some $a' \in [2p]$. In this case, (18) boils down to $\{\tilde{y}_{a'} \oplus k_2 = \tilde{u}_b, 2\tilde{y}_{a'} \oplus k_2 = \tilde{w}_c\}$. The probability of occurrence of Bad_2^* can now be bounded using the events $M_i[1] \oplus 2k_0 \oplus 2^2 k_1 = \tilde{x}_{a'}$ and $k_2 = \tilde{y}_{a'} \oplus \tilde{u}_b$; the probability of the first event is bounded by 2^{-n} through the randomness of k_0, and the probability of the latter is bounded by 2^{-n} through the randomness of k_2. Note that the number of choices for a' is $2p$, that for b, c (each) is p, and that for i is q. For each of these choices of $\tilde{x}_{a'}$ and $\tilde{u}_b$, the number of choices for $\tilde{w}_c$ is 1. Hence,

$$\Pr[\mathsf{Bad}_2^*] \leq \frac{2qp^2}{2^{2n}}. \tag{19}$$

On the other hand, if $M_i[1] \oplus 2k_0 \oplus 2^2k_1$ does not collide with any primitive query, then $y_i[1]$ is fresh. Thus, (18) boils down to $\{y_i[1] \oplus k_2 = \tilde{u}_b,\ 2y_i[1] \oplus k_2 = \tilde{w}_c\}$. Note that the rank of this system of equations is 2. Varying over all possible choices of $b, c \in [p]$ and $i \in [q]$ gives

$$\Pr[\mathsf{Bad}_2^*] \leq \frac{qp^2}{2^n(2^n - \ell)}. \tag{20}$$

Case (2): In this case, we assume $l_i > 1$. Let $\mathsf{CollX}^{(2)}$ be the event that refers to the collision of any two input blocks, i.e.

$$\mathsf{CollX}^{(2)} \ : \{\exists\, \alpha_1, \alpha_2 \in [l_i],\ \text{such that}\ \alpha_1 \neq \alpha_2, x_i[\alpha_1] = x_i[\alpha_2]\}\,.$$

Therefore, we write

$$\Pr[\mathsf{Bad}_2^*] \leq \sum_{i=1}^{q} \left(\underbrace{\Pr[\text{Eqn.s (18) hold } \wedge \mathsf{CollX}^{(2)}]}_{(1)} + \underbrace{\Pr[\text{Eqn.s (18) hold } \wedge \overline{\mathsf{CollX}^{(2)}}]}_{(2)} \right).$$

The joint event in (1) holds with probability at most $\binom{\ell}{2}/2^{2n}$ (in which the event $\mathsf{CollX}^{(2)}$ contrinutes the term $\binom{\ell}{2}/2^n$ and the randomness of k_2 contributes the term $1/2^n$). The event in (2) ensures the freshness of at least one of the variables $y_i[1], \ldots, y_i[l_i]$. Without loss of generality, let us assume $y_i[1]$ is fresh. Given the values of all the other random variables $y_i[\star]$ in (18), the reduced system of equations $\{y_i[1] \oplus k_2 = c,\ 2y_i[1] \oplus k_2 = c'\}$ with rank 2 results in an upper bound of $1/2^n(2^n - \ell)$. Varying (1) and (2) over all choices of $b, c \in [p]$ and $i \in [q]$ gives

$$\Pr[\mathsf{Bad}_2^*] \leq \frac{qp^2\ell^2}{2^{2n+1}} + \frac{qp^2}{2^n(2^n - \ell)}. \tag{21}$$

From Eqn.s (19), (20) and (21), and with the assumption that $\ell \leq 2^{n-1}$, we obtain

$$\Pr[\mathsf{Bad}_2^*] \leq \frac{5qp^2\ell^2}{2^{2n}} + \frac{2qp^2}{2^{2n}}. \tag{22}$$

Bounding Event Bad_3^*. This event can be split into the following two sub-events:

$$\begin{aligned}
(1) \quad : \quad & \big\{\exists\, i_1 \neq i_2 \text{ in } [q] \text{ such that } \left(\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i_1} \oplus k_2\right) \| 0 = \mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i_2} \oplus k_2\right) \| 0\right) \\
& \wedge \left(\mathsf{chop}_{\mathsf{LSB}}\left(\Theta_{i_1} \oplus k_2\right) \| 1 \in \mathsf{Dom}(\pi^1)\right) \wedge \overline{\mathsf{Coll} \vee \mathsf{3\text{-}Coll}}\big\}, \\
(2) \quad : \quad & \big\{\exists\, i_1 \neq i_2 \text{ in } [q] \text{ such that } \left(\mathsf{chop}_{\mathsf{LSB}}\left(\Theta_{i_1} \oplus k_2\right) \| 1 = \mathsf{chop}_{\mathsf{LSB}}\left(\Theta_{i_2} \oplus k_2\right) \| 1\right) \\
& \wedge \left(\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i_1} \oplus k_2\right) \| 0 \in \mathsf{Dom}(\pi^0)\right) \wedge \overline{\mathsf{Coll} \vee \mathsf{3\text{-}Coll}}\big\}.
\end{aligned}$$

Bounding (1): For fixed $i_1 \neq i_2$ in $[q]$ and a fixed $c \in [p]$, the event is $(\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i_1} \oplus k_2\right) \| 0 = \mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i_2} \oplus k_2\right) \| 0) \wedge (\mathsf{chop}_{\mathsf{LSB}}\left(\Theta_{i_1} \oplus k_2\right) \| 1 = \tilde{w}_c) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}}$. Without loss of generality, assume $l_{i_1} \geq l_{i_2}$. Since the probability of (1) is zero for $l_{i_1} \leq 1$, assume $l_{i_1} \geq 2$. As before, we determine an index $\beta \in [l_{i_1} - 1]$: If $l_{i_1} > l_{i_2}$, then $\beta = l_{i_1}$; if $l_{i_1} = l_{i_2}$ and $\mathsf{NEQ}_{i_1 i_2} = \{l_{i_1}\}$, then the probabilty becomes zero – so we set $\beta = \max\{\alpha \in \mathsf{NEQ}_{i_1 i_2}\}$ $(\neq l_{i_1})$ when $l_{i_1} = l_{i_2}$. Let

$$\mathsf{CollX}_\beta^{(3)} : \{(\exists\, \beta_1 \in [l_{i_1}] : \beta_1 \neq \beta, x_{i_1}[\beta] = x_{i_1}[\beta_1]) \vee (\exists\, \beta_2 \in [l_{i_2}] : x_{i_1}[\beta] = x_{i_2}[\beta_2])\}$$

be the event that denotes the collision of $x_{i_1}[\beta]$ with atleast one of the remaining input blocks. Also let E_β denote the event $\{\exists\ a \in [p] : x_{i_1}[\beta] = \tilde{x}_a\}$. Therefore, we write

$$\begin{aligned}
\Pr\Big[&(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2)\,\|0 = \mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_2} \oplus k_2)\,\|0) \wedge (\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2)\,\|1 = \tilde{w}_c) \\
&\wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}}\Big] \leq \Pr\Big[(\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2)\,\|1 = \tilde{w}_c) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \mathsf{CollX}_\beta^{(3)}\Big] \\
&+ \Pr\Big[(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2)\,\|0 = \mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_2} \oplus k_2)\,\|0) \wedge \\
&(\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2)\,\|1 = \tilde{w}_c) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \overline{\mathsf{CollX}_\beta^{(3)}}\Big] \\
&\leq \Pr\Big[(\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2)\,\|1 = \tilde{w}_c) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \mathsf{CollX}_\beta^{(3)}\Big] \\
&+ \Pr\Big[(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2)\,\|0 = \mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_2} \oplus k_2)\,\|0) \wedge \\
&(\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2)\,\|1 = \tilde{w}_c) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \overline{\mathsf{CollX}_\beta^{(3)}} \wedge \mathsf{E}_\beta\Big] \\
&+ \Pr\Big[(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2)\,\|0 = \mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_2} \oplus k_2)\,\|0) \wedge \\
&(\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2)\,\|1 = \tilde{w}_c) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \overline{\mathsf{CollX}_\beta^{(3)}} \wedge \overline{\mathsf{E}_\beta}\Big]. \qquad (23)
\end{aligned}$$

We break this down into three manageable chunks:

$$\begin{aligned}
\mathsf{E.1} &:= \Pr\Big[(\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2)\,\|1 = \tilde{w}_c) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \mathsf{CollX}_\beta^{(3)}\Big], \\
\mathsf{E.2} &:= \Pr\Big[(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2)\,\|0 = \mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_2} \oplus k_2)\,\|0) \\
&\quad \wedge (\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2)\,\|1 = \tilde{w}_c) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \overline{\mathsf{CollX}_\beta^{(3)}} \wedge \mathsf{E}_\beta\Big] \text{ and} \\
\mathsf{E.3} &:= \Pr\Big[(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2)\,\|0 = \mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_2} \oplus k_2)\,\|0) \\
&\quad \wedge (\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2)\,\|1 = \tilde{w}_c) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \overline{\mathsf{CollX}_\beta^{(3)}} \wedge \overline{\mathsf{E}_\beta}\Big].
\end{aligned}$$

1. In the sub-event $(\mathsf{E.1})$, since the equation $\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2)\,\|1 = \tilde{w}_c$ is non-trivial, it can be bound by probability $2/2^n$ using the randomness of k_2, and $\mathsf{CollX}_\beta^{(3)}$ holds with probability at most $2\ell/2^n$. Thus, $(\mathsf{E.1})$ can be bound by
$$4\ell/2^{2n}. \qquad (24)$$

2. We first consider the case when $l_{i_1} = l_{i_2} + 1$ and $\mathsf{NEQ}_{i_1 i_2} = \{l_{i_1}\}$ in $(\mathsf{E.2})$. Since $x_{i_1}[l_{i_1}] = \tilde{x}_a$, it boils the event $(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2)\,\|0 = \mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_2} \oplus k_2)\,\|0) \wedge (\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2)\,\|1 = \tilde{w}_c)$ down to the following system of equations:
$$\begin{aligned}
&\tilde{y}_a = 0^n, \\
&2^{l_{i_1}} y_{i_1}[1] \oplus 2^{l_{i_1}-1} y_{i_1}[2] \oplus \ldots \oplus 2 y_{i_1}[l_{i_1}] \oplus k_2 = \tilde{w}_c. \qquad (25)
\end{aligned}$$
As the equation $\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2)\,\|1 = \tilde{w}_c$ is non-trivial, its probability can be at most $2/2^n$. Moreover, the probability that $x_{i_1}[l_{i_1}] = \tilde{x}_a$ is bounded above by $1/2^n$

(since the number of choices for a is 1). Thus, this case of $(\mathsf{E}.2)$ can be bound by $2/2^{2n}$.

We next consider the case when $l_{i_1} > l_{i_2}$ in $(\mathsf{E}.2)$. As E_β holds, $y_{i_1}[l_{i_1}]$ is not fresh. However, as Coll, $\mathsf{3\text{-}Coll}$ and $\mathsf{CollX}_\beta^{(3)}$ also do not hold, at least one of the variables $y_{i_1}[\star]$ must be fresh, i.e. $\exists\, \alpha \in \mathsf{NEQ}_{i_1 i_2} \setminus \{l_{i_1}\}$ such that $y_{i_1}[\alpha]$ is fresh. Without loss of generality, let us assume that $y_{i_1}[1]$ is fresh. Give all other random variables $y_{i_1}[\star]$ and $y_{i_2}[\star]$ in

$$\begin{aligned} y_{i_1}[1] \oplus y_{i_1}[2] \oplus \ldots y_{i_1}[l_{i_1}] \oplus y_{i_2}[1] \oplus y_{i_2}[2] \oplus \ldots \oplus y_{i_2}[l_{i_2}] &= 0^n, \\ 2^{l_{i_1}} y_{i_1}[1] \oplus 2^{l_{i_1}-1} y_{i_1}[2] \oplus \ldots 2 y_{i_1}[l_{i_1}] \oplus k_2 &= \tilde{w}_c, \end{aligned} \tag{26}$$

we obtain $y_{i_1}[1] = d$ and $2^{l_{i_1}} y_{i_1}[1] \oplus k_2 = d'$, for constants d and d'. Hence,

$$\Pr[\text{Eqn. 26 holds}] \leq \frac{1}{2^n(2^n - 2\ell)} \leq \frac{8}{2^{2n}}, \quad \text{assuming } \ell \leq 2^{n-2}.$$

Combining the above bounds the probability of $(\mathsf{E}.2)$ by

$$\frac{10}{2^{2n}}. \tag{27}$$

3. In the event $(\mathsf{E}.3)$, it is easy to see that $y_{i_1}[l_{i_1}]$ is fresh. Hence, given all other random variables $y_{i_1}[\star]$ and $y_{i_2}[\star]$ in Eqn. (26), the system is reduced to $y_{i_1}[l_{i_1}] = d$, $2y_{i_1}[l_{i_1}] \oplus k_2 = d'$ for some constants d and d'. Hence, the probability of $(\mathsf{E}.3)$ has an upper bound of
$$\frac{4}{2^n(2^n - 2\ell)} \leq \frac{8}{2^{2n}}, \tag{28}$$
where the last inequality follows as $\ell \leq 2^{n-1}$.

Varying over all possible choices of $i_1 \neq i_2$ in $[q]$ and $c \in [p]$ and combining Eqn.s (23), (24), (27) and (28) gives
$$\Pr[(1)] \leq \frac{(4.5 + l)q^2 p}{2^{2n}}. \tag{29}$$

Bounding (2): This is symmetric to (1). Hence, it can be similarly bounded:

$$\Pr[(2)] \leq \frac{(4.5 + l)q^2 p}{2^{2n}}. \tag{30}$$

Therefore, from Eqn.s (29) and (30),

$$\Pr[\mathsf{Bad}_3^*] = \Pr[(1)] + \Pr[(2)] \leq \frac{(9 + 2l)q^2 p}{2^{2n}}. \tag{31}$$

Bounding Bad_4^*. This event can be split into the following two sub-events:

$$\begin{aligned} (1) \quad : \quad & \{\exists\, i \in [q],\, b, c \in [p] \text{ such that } (\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2) \| 0 = \mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_2} \oplus k_2) \| 0) \\ & \wedge (T_i \oplus \tilde{v}_b = \tilde{z}_c) \wedge \overline{\mathsf{Coll} \vee \mathsf{3\text{-}Coll}}\}, \\ (2) \quad : \quad & \{\exists\, i \in [q],\, b, c \in [p] \text{ such that } (\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2) \| 1 = \mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_2} \oplus k_2) \| 1) \\ & \wedge (T_i \oplus \tilde{z}_c = \tilde{v}_b) \wedge \overline{\mathsf{Coll} \vee \mathsf{3\text{-}Coll}}\}. \end{aligned}$$

Bounding (1): We fix a message M_i consisting of l_i blocks. We also fix the indices b and c. Now, we analyze the probability of the event in two cases: (I) The i^{th} construction query occurs after the b^{th} and c^{th} primitive queries. (II) At least one of the primitive queries appears after the i^{th} construction query.

Case I: As T_i is distributed uniformly at random and since the distribution of k_2 is independent of all the other random variables, we bound the probability of the event by $1/2^{2n}$. Varying over all possible choices of $i \in [q]$ and $b, c \in [p]$, we have

$$\Pr[\mathsf{Bad}_4^*] \leq \frac{qp^2}{2^{2n}} \text{ in case (I).} \tag{32}$$

Case II: Suppose the b^{th} primitive query is the latest.

(a) If the primitive query is in the forward direction, then $\tilde{v}_b$ is randomly distributed. Hence by the randomness of k_2 and $\tilde{v}_b$, we bound the probability of the event to at most $2/2^n$. Varying over all possible choices of $i \in [q]$ and $b, c \in [p]$, we have

$$\Pr[\mathsf{Bad}_4^*] \leq \frac{2qp^2}{2^{2n}} \text{ in case (IIa).} \tag{33}$$

(b) If the b^{th} primitive query is in the inverse direction, then $\tilde{u}_b$ is random. We bound the event Bad_4^* given the complement of the event

$$\mathsf{E} : \left\{ |\{(T_i, \tilde{v}_b, \tilde{z}_c) \in [q] \times [p] \times [p] : T_i = \tilde{v}_b \oplus \tilde{z}_c\}| \geq \frac{qp^2}{2^n} + \sqrt{3nqp^2} \right\}.$$

As $\Pr[\mathsf{Bad}_4^*] \leq \Pr[\mathsf{Bad}_4^* \mid \overline{\mathsf{E}}] + \Pr[\mathsf{E}]$ and as $\Pr[\mathsf{E}] \leq 2/2^n$ according to the sum-capture Lemma 2, for a fixed choice of i, b and c such that $T_i = \tilde{v}_b \oplus \tilde{z}_c$, the probability of the event $\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2) \| 0 = \tilde{u}_b$ is at most $1/2^n$ by the randomness of k_2. As the number of choices for i, b and c is at most $qp^2/2^n + \sqrt{3nqp^2}$,

$$\Pr[\mathsf{Bad}_4^*] \leq \frac{qp^2}{2^{2n}} + \frac{\sqrt{3nqp^2}}{2^n} + \frac{2}{2^n} \text{ in case (IIb).} \tag{34}$$

The analysis is exactly the same when the c^{th} primitive query is the latest. Therefore,

$$\Pr[\mathsf{Bad}_4^*] \leq \frac{qp^2}{2^{2n}} + \frac{\sqrt{3nqp^2}}{2^n} + \frac{2}{2^n}. \tag{35}$$

Bounding (2): The analysis for bounding this sub-event is exactly identical to that of Bad_4^*. Thus

$$\Pr[\mathsf{Bad}_4^*] \leq \frac{2qp^2}{2^{2n}} + \frac{2\sqrt{3nqp^2}}{2^n} + \frac{4}{2^n}. \tag{36}$$

Bounding Bad_5^*. We again begin by partitioning the event into two sub-events:

$$\begin{aligned}
(1) \quad : \quad & \{\exists\, i_1 \neq i_2 \text{ in } [q] \text{ such that } (\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2) \| 0 = \mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_2} \oplus k_2) \| 0) \\
& \wedge (T_{i_1} = T_{i_2}) \wedge \overline{\mathsf{Coll} \vee \mathsf{3\text{-}Coll}}\} \\
(2) \quad : \quad & \{\exists\, i_1 \neq i_2 \text{ in } [q] \text{ such that } (\mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_1} \oplus k_2) \| 0 = \mathsf{chop}_{\mathsf{LSB}}(\Theta_{i_2} \oplus k_2) \| 1) \\
& \wedge (T_{i_1} = T_{i_2}) \wedge \overline{\mathsf{Coll} \vee \mathsf{3\text{-}Coll}}\}.
\end{aligned}$$

Bounding (1): For the two fixed distinct messages M_{i_1} and M_{i_2}, the event $\mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_1} \oplus k_2) \| 0 = \mathsf{chop}_{\mathsf{LSB}}(\Sigma_{i_2} \oplus k_2) \| 0$ is reduced to the following equations:

$$y_{i_1}[1] \oplus y_{i_1}[2] \oplus \ldots \oplus y_{i_1}[l_{i_1}] \oplus y_{i_2}[1] \oplus y_{i_2}[2] \oplus \ldots \oplus y_{i_2}[l_{i_2}] = 0^n. \tag{37}$$

Without loss of generality, assume $l_{i_1} \geq l_{i_2}$. The probability of the event is zero for $l_{i_1} \leq 1$. Thus, we assume $l_{i_1} \geq 2$. As before, we determine an index $\beta \in [l_{i_1} - 1]$ as follows: if

$l_{i_1} > l_{i_2}$, then $\beta = l_{i_1}$. If $l_{i_1} = l_{i_2}$ and $\mathsf{NEQ}_{i_1 i_2} = \{l_{i_1}\}$, then the probabilty of the event is again zero. So we set $\beta = \max \mathsf{NEQ}_{i_1 i_2}$ when $l_{i_1} = l_{i_2}$. Note the following event:

$$\mathsf{CollX}_{\beta}^{(4)} : \big\{ (\exists\, \beta_1 \in [l_{i_1}] \text{ such that } \beta_1 \neq \beta, x_{i_1}[\beta] = x_{i_1}[\beta_1]) \\ \vee (\exists\, \beta_2 \in [l_{i_2}] \text{ such that } x_{i_1}[\beta] = x_{i_2}[\beta_2]) \big\}.$$

$$\text{Therefore, } \Pr\left[\mathsf{chop_{LSB}}\left((\Sigma_{i_1} \oplus k_2)\right)\|0 = \mathsf{chop_{LSB}}\left(\Sigma_{i_2} \oplus k_2\right)\|0) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}}\right] \leq \\ \underbrace{\Pr\left[\mathsf{CollX}_{\beta}^{(5)}\right]}_{\mathsf{E.4}} + \underbrace{\Pr\left[\mathsf{chop_{LSB}}\left(\Sigma_{i_1} \oplus k_2\right)\|0 = \mathsf{chop_{LSB}}\left(\Sigma_{i_2} \oplus k_2\right)\|0 \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \overline{\mathsf{CollX}_{\beta}^{(5)}}\right]}_{\mathsf{E.5}} \tag{38}$$

Due to the randomness of k_0 and k_1, the first term (i.e. $\mathsf{E.4}$) in Eqn. (38) is bound by $(\ell - 1 + \ell)/2^n \leq 2\ell/2^n$. We split the analysis of $\mathsf{E.5}$ into the following two cases:

Case I: When $l_{i_1} = l_{i_2} + 1$ and $\mathsf{NEQ}_{i_1 i_2} = \{l_{i_1}\}$, if $x_{i_1}[l_{i_1}] = \tilde{x}_a$ for some $a \in [p]$, then $\tilde{y}_a = 0^n$. Therefore, the event occurs with a probability of at most $1/2^n$ due to the randomness of k_0 and k_1 (note that the number of choices for $\tilde{x}_a$ is 1). On the other hand, if $x_{i_1}[l_{i_1}]$ is fresh, then $y_{i_1}[l_{i_1}]$ is freshly sampled and hence for this random variable, the rank 1 equation $\mathsf{chop_{LSB}}\left(\Sigma_{i_1} \oplus k_2\right)\|0 = \mathsf{chop_{LSB}}\left(\Sigma_{i_2} \oplus k_2\right)\|0$ ensures a probability bound of $1/(2^n - 2\ell)$, by the rank argument (i.e. Lemma 3).

Case-II: When $l_{i_1} \geq l_{i_2} + 2$, at least one $\beta \in \{l_{i_2} + 1, \ldots, l_{i_1}\}$ can be certainly found such that $x_{i_1}[\beta]$ is fresh and hence $y_{i_1}[\beta]$ is freshly sampled. For this random variable $y_{i_1}[\beta]$, the rank 1 equation $\mathsf{chop_{LSB}}\left(\Sigma_{i_1} \oplus k_2\right)\|0 = \mathsf{chop_{LSB}}\left(\Sigma_{i_2} \oplus k_2\right)\|0$ ensures a probability bound of $1/(2^n - 2\ell)$, by the rank argument (i.e. Lemma 3).

Combining the above two cases and by assuming $\ell \leq 2^{n-2}$gives

$$\Pr\left[\mathsf{chop_{LSB}}\left(\Sigma_{i_1} \oplus k_2\right)\|0 = \mathsf{chop_{LSB}}\left(\Sigma_{i_2} \oplus k_2\right)\|0 \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \wedge \overline{\mathsf{CollX}_{\beta}^{(5)}}\right] \leq \frac{5}{2^n}. \tag{39}$$

Therefore from Eqn.s (38) and (39), and by the assumption $\ell \leq 2^{n-2}$, we have

$$\Pr\left[\mathsf{chop_{LSB}}\left(\Sigma_{i_1} \oplus k_2\right)\|0 = \mathsf{chop_{LSB}}\left(\Sigma_{i_2} \oplus k_2\right)\|0 \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}}\right] \leq \frac{2\ell + 5}{2^n}. \tag{40}$$

Finally, from Eqn. (41), the fact that the event $T_{i_1} = T_{i_2}$ is independent of the event $(\mathsf{chop_{LSB}}\left(\Sigma_{i_1} \oplus k_2\right)\|0 = \mathsf{chop_{LSB}}\left(\Sigma_{i_2} \oplus k_2\right)\|0) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}}$, and that for a fixed choice of i_1 and i_2, the probability that $T_{i_1} = T_{i_2}$ holds is 2^{-n}, we have

$$\begin{aligned} & \Pr[(1)] \\ = & \sum_{i_1, i_2} \big(\Pr[T_{i_1} = T_{i_2}] \cdot \\ & \Pr\left[(\mathsf{chop_{LSB}}\left(\Sigma_{i_1} \oplus k_2\right)\|0 = \mathsf{chop_{LSB}}\left(\Sigma_{i_2} \oplus k_2\right)\|0) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}}\right] \big) \\ \leq & \frac{\frac{1}{2}q^2(2\ell + 5)}{2^{2n}} \leq \frac{q^2\ell + 2.5q^2}{2^{2n}}. \end{aligned} \tag{41}$$

Bounding (2): This event is symmetric to the first, and thus has the same bound:

$$\Pr[(2)] \leq \frac{q^2\ell + 2.5q^2}{2^{2n}}. \tag{42}$$

Therefore, from Eqn.s (41) and (42), we have

$$\Pr[\mathsf{Bad}_5^*] = \Pr[(1)] + \Pr[(2)] \leq \frac{2q^2\ell + 5q^2}{2^{2n}}. \tag{43}$$

Bounding Bad_6^*. Consider the sub-event $(\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i_1} \oplus k_2\right) \| 0 = x_{i_1}[\alpha]) \wedge (\mathsf{chop}_{\mathsf{LSB}}\left(\Theta_{i_1} \oplus k_2\right) \| 1 = x_{i_3}[\beta])$ $\wedge\ \overline{\mathsf{Coll} \vee \mathsf{3\text{-}Coll}}$. This event can be expanded in terms of XOR operations on the hash permutation outputs as follows (where $\alpha \in [l_{i_2}]$ and $\beta \in [l_{i_3}]$ are aribitrary indices):

$$\begin{aligned}
\Pr\left[\mathsf{Bad}_6^*\right] &= \Pr\Big[\left(\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i_1} \oplus k_2\right) \| 0 = M_{i_2}[\alpha] \oplus 2^{\alpha} k_0 \oplus 2^{2\alpha} k_1\right) \\
&\quad \wedge \left(\mathsf{chop}_{\mathsf{LSB}}\left(\Theta_{i_1} \oplus k_2\right) \| 1 = M_{i_3}[\beta] \oplus 2^{\beta} k_0 \oplus 2^{2\beta} k_1\right) \wedge \overline{\mathsf{Coll} \vee \mathsf{3\text{-}Coll}}\Big] \\
&\leq \Pr\left[\pi\left(M_{i_1}[1]\right) \oplus \ldots \oplus \pi\left(M_{i_1}[l_i]\right) = M_{i_2}[\alpha] \oplus 2^{\alpha} k_0 \oplus 2^{2\alpha} k_1\right] \times \\
&\quad \Pr\left[2^{l_i} \pi\left(M_{i_1}[1]\right) \oplus \ldots \oplus 2\pi\left(M_{i_1}[l_i]\right) = M_{i_3}[\beta] \oplus 2^{\beta} k_0 \oplus 2^{2\beta} k_1\right].
\end{aligned}$$

For fixed indices i_1, i_2, i_3, the above probability is clearly $\left(2^{-n}\right)^2$, by the randomness of keys k_0 and k_1. Similarly, the probability of occurrence of the remaining four sub-events is also $\left(2^{-n}\right)^2$. Counting the choices for each index thus gives

$$\Pr[\mathsf{Bad}_6^*] \leq \frac{q^3 l^2 + 2qp^2 + 2q^3}{2^{2n}}. \tag{44}$$

Bounding RC_{Σ}^*. Recall the offline phase of the ideal oracle (Fig.s 6.3-6.5). Denote the number of elements removed from the construction transcript of an adversary in step 3 of stage II by s_1, and the number of elements removed in step 2 of stage III by s_2. Thus $\widehat{q^0} := q - (s_1 + s_2 + f)$ denotes the number of elements left in $\widetilde{\Sigma}$ at the end of the offline phase, f as in step 10 of stage III. Also let $\widehat{p^0} := \left|\mathsf{Dom}\left(\pi^0\right)\right|$, where the set $\mathsf{Dom}\left(\pi^0\right)$ is as it stands at the end of the offline phase. Thus $\widehat{p^0} = p + (s_1 + s_2)$ (since p is the number of primitive queries with LSB 0). $\widehat{q^1}$ and $\widehat{p^1}$ can be similarly defined. The bad event occurs if for some $i' \neq i$ in $[\widehat{q^0}]$, one of the following occurs:

$$\begin{aligned}
(1) &\quad : \quad \Big\{\left(\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_i \oplus k_2\right) \| 0 = \mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i'} \oplus k_2\right) \| 0\right) \wedge \left(z_i = \tilde{z}_c\right) \text{ for some } c \in [\widehat{p^0}]\Big\} \\
(2) &\quad : \quad \Big\{\left(\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_i \oplus k_2\right) \| 0 = \mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i'} \oplus k_2\right) \| 0\right) \wedge \left(z_i = z_j\right) \text{ for some } j \in [\widehat{q^0}]\Big\},
\end{aligned}$$

where $v_i \leftarrow_{\$} \{0,1\}^n \setminus \mathsf{Ran}(\pi^0)$.

Bounding (1): The sub-event $z_i = \tilde{z}_c$, i.e. $v_i = T_i \oplus \tilde{z}_c$ is a result of the lazy sampling of v_i, independent of the sub-event $\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_i \oplus k_2\right) \| 0 = \mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i'} \oplus k_2\right) \| 0$. For a particular choice of i,i' and c,

$$\begin{aligned}
&\Pr\left[\left(\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_i \oplus k_2\right) \| 0 = \mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i'} \oplus k_2\right) \| 0\right) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}}\right] \times \Pr\left[z_i = \tilde{z}_c\right] \\
\leq\ & \frac{2\ell + 5}{2^n} \times \Pr\left[z_i = \tilde{z}_c\right] \ \text{ (as already computed in Eqn. (41))} \\
\leq\ & \frac{2\ell + 5}{2^n} \cdot \frac{1}{2^n - \widehat{q^0}},
\end{aligned}$$

where ℓ denotes the maximum number of message blocks amongst all q queries. Summing over all choices of i, i' and c bounds the probability to

$$\widehat{q^0}(\widehat{q^0} - 1)\widehat{p^1} \cdot \frac{(2\ell + 5)}{2^{2n}}. \tag{45}$$

Bounding (2): We split this bad event into the following cases:

Case I: Suppose $i' \neq j$. As in (1), the sub-event $z_i = z_j$ is a result of the lazy sampling of z_i, independent of the sub-event $\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_i \oplus k_2\right) \| 0 = \mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i'} \oplus k_2\right) \| 0$. Thus, the probability of this case for a particular choice of i,i' and c is

$$\begin{aligned}
\mathsf{P}_{i,i',c} &= \Pr[\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_i \oplus k_2\right) \| 0 = \mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i'} \oplus k_2\right) \| 0 \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}}] \times \Pr[z_i = z_j] \\
&\leq \frac{2\ell + 5}{2^n} \times \frac{1}{2^n - \widehat{q^0}} \text{ (as already computed in Eqn. (41))} \\
&\leq \frac{4\ell + 10}{2^{2n}} \text{ (since } \widehat{q^0} \leq 2^{n-1}\text{).}
\end{aligned}$$

Summing over all possible choices of i,i' and c, we obtain an upper bound

$$\widehat{q^0}(\widehat{q^0} - 1)(\widehat{q^0} - 2) \cdot \frac{4\ell + 10}{2^{2n}}. \tag{46}$$

Case II: Now suppose $i' = i$. $v_i \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathsf{Ran}(\pi^0)$ is thus sampled first and $z_{i'}$ is then set to v_i. This case eventually boils down to the joint event $\big(\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_i \oplus k_2\right) \| 0 = \mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i'} \oplus k_2\right) \| 0\big) \wedge (z_i = z_j)$. If $T_i = T_{i'}$, then $z_i = z_j$ is implied by the first sub-event. Therefore,

$$\begin{aligned}
& \Pr[T_i = T_{i'} \wedge (\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_i \oplus k_2\right) \| 0 = \mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i'} \oplus k_2\right) \| 0) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}}] \\
= \; & \Pr\left[(\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_i \oplus k_2\right) \| 0 = \mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i'} \oplus k_2\right) \| 0) \wedge \overline{\mathsf{Coll}} \wedge \overline{\mathsf{3\text{-}Coll}} \mid T_i = T_{i'}\right] \\
& \cdot \Pr\left[T_i = T_{i'}\right] \\
\leq \; & \frac{2\ell + 5}{2^n} \times \Pr\left[T_i = T_{i'}\right] \text{ (as already computed in Eqn. (41))} \\
\leq \; & \frac{2\ell + 5}{2^{2n}},
\end{aligned} \tag{47}$$

as all the $\widehat{q^0}$ messages are fixed given $T_1, \ldots, T_{\widehat{q^0}}$. On the other hand, if $T_i \neq T_{i'}$ then $z_i \neq v_{i'} \oplus T_{i'}$ and hence the probability becomes zero.

Summing over all (i, i', j) with $i < i'$, the probability for this case is bounded by

$$\frac{\widehat{q^0}(\widehat{q^0} - 1)}{2} \cdot \frac{2\ell + 5}{2^{2n}}. \tag{48}$$

Combining cases I and II, we have

$$\Pr[(2)] \leq \widehat{q^0}(\widehat{q^0} - 1)(2\widehat{q^0} - 3) \cdot \frac{2\ell + 5}{2^{2n}}. \tag{49}$$

$$\begin{aligned}
\text{Therefore, } \Pr[\mathsf{RC}^*_\Sigma] \leq \Pr[(1)] + \Pr[(2)] &\leq \frac{\widehat{q^0}(\widehat{q^0} - 1) \cdot (2\ell + 5)}{2^{2n}} \left(2\widehat{q^0} - 3 + \widehat{p^1}\right) \\
&\leq \frac{2q(q - 1)(2\ell + 5)}{2^{2n}} \left(q + p\right),
\end{aligned} \tag{50}$$

since $\widehat{q^0} \leq q$ and $\widehat{p^1} \leq 2p$.

Bounding RC^*_Θ. The event RC^*_Θ can be bound identically as RC^*_Σ. Hence,

$$\Pr[\mathsf{RC}^*_\Theta] \leq \frac{2q(q - 1)(2\ell + 5)}{2^{2n}} \left(q + p\right). \tag{51}$$

The final bound follows from Eqn.s (4)–(51). □

6.6 Analysis of Good Transcripts

In this section, we show that realizing a good transcript $\tau = (\hat{\tau}_c, \tau_p)$ is almost as likely in the real world as in the ideal world. For each $i \in \mathcal{F}$, both $\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_i \oplus k_2\right) \| 0$ and $\mathsf{chop}_{\mathsf{LSB}}\left(\Theta_i \oplus k_2\right) \| 1$ are fresh for elements (Σ_i, Θ_i) in $\widetilde{\Sigma} \times \widetilde{\Theta}$, as shown in line 9 of stage II of Fig. 6.4. Due to the changes made in lines 2 and 6 of the same stage, repeating elements of $\widetilde{\Sigma}$ (resp. $\widetilde{\Theta}$) are moved to τ_p, and each such index i is added to $\mathcal{I}$. Since these alterations do not create any inconsistencies, the cardinality of τ_p increases. Assuming that $s_1 + s_2$ elements are added to τ_p in step 3 and $t_1 + t_2$ elements in step 7, the size of the modified transcript τ'_p is $p' := 2p + s_1 + s_2 + t_1 + t_2 = p'_0 + p'_1$ (where $p'_0 := p + s_1 + s_2$ and $p'_1 := p + t_1 + t_2$). Therefore, the number of elements in the modified collections $\widetilde{\Sigma}$ and $\widetilde{\Theta}$, which we denote by $\widetilde{\Sigma}_*$ and $\widetilde{\Theta}_*$ (resp.), is $q' := q - s_1 - s_2 - t_1 - t_2$ at the end of stage II.

Moreover, as the transcript τ is good, for every $i \notin \mathcal{F} \sqcup \mathcal{I}$, exactly one of $\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_i \oplus k_2\right) \| 0$ and $\mathsf{chop}_{\mathsf{LSB}}\left(\Theta_i \oplus k_2\right) \| 1$ is fresh in $(\widetilde{\Sigma}_*, \widetilde{\Theta}_*)$. Thus, there are exactly $(q' + f)$ fresh blocks ($2f$ fresh blocks corresponding to all indices belonging to $\mathcal{F}$ and $(2q' - 2f)/2$ additional fresh blocks), and $q' - f$ repeated blocks.

Let $\mathcal{P}^c$ be the set of all indices corresponding to queries with one of their hash output blocks colliding with one of the hash primitive inputs. We define a relation $\sim$ on $\mathcal{Q}^c := [q] \setminus \mathcal{F} \sqcup \mathcal{I} \sqcup \mathcal{P}^c$ as $i_1 \sim i_2$ if $\big(\mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i_1} \oplus k_2\right) \| 0 = \mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i_2} \oplus k_2\right) \| 0\big) \vee \big(\mathsf{chop}_{\mathsf{LSB}}\left(\Theta_{i_1} \oplus k_2\right) \| 1 = \mathsf{chop}_{\mathsf{LSB}}\left(\Theta_{i_2} \oplus k_2\right) \| 1\big)$, where Σ_{i_1} is an element of $\widetilde{\Sigma}_*$ and Θ_{i_1} is an element of $\widetilde{\Theta}_*$. Note that as τ is good, for any $i_1 \sim i_2$, exactly one of the following two occurs:

$$\begin{aligned} &\text{(i)} \qquad \mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i_1} \oplus k_2\right) \| 0 = \mathsf{chop}_{\mathsf{LSB}}\left(\Sigma_{i_2} \oplus k_2\right) \| 0, \\ &\text{(ii)} \qquad \mathsf{chop}_{\mathsf{LSB}}\left(\Theta_{i_1} \oplus k_2\right) \| 1 = \mathsf{chop}_{\mathsf{LSB}}\left(\Theta_{i_2} \oplus k_2\right) \| 1. \end{aligned}$$

Furthermore, if i_1 and i_2 are related through (i), then any other index $j \in \mathcal{Q}^c$ cannot be related to i_1 or i_2 through (ii), and vice versa. Clearly, $\sim$ is an equivalence relation. Thus, it partitions $\mathcal{Q}^c$, which in turn induces a partition on $\widetilde{\Sigma}_*$ and $\widetilde{\Theta}_*$. Let r_0 be the number of equivalence classes of $\widetilde{\Sigma}_*$ and r_1 the number of equivalence classes of $\widetilde{\Theta}_*$. Let d_i^0 be the number of elements in the i^{th} equivalence class of $\widetilde{\Sigma}_*$ and d_i^1 the number of elements in the i^{th} equivalence class of $\widetilde{\Theta}_*$. For each equivalence class of $\widetilde{\Sigma}_*$ or $\widetilde{\Theta}_*$, we sample an output for the least-indexed element, thus determining the (common) output for all other elements in that class (see lines 4 and 11 of stage III in Fig. 6.5). Due to the definition of $\mathcal{S}$ in line 12 of stage II, and due to lines 4, 5, 11 and 12 of stage III $\forall\, i \in [q']$, $v_i \oplus z_i = T_i$ holds. Also, RC_Σ or RC_Θ are not set to $\mathtt{1}$ (as τ is good), ensuring no range collision for two different inputs. This proves the following result:

For a good transcript τ, the q' tuples of input and output blocks of π^0 and π^1 are permutation compatible, i.e. $\widetilde{\Sigma}_*$ is permutation compatible with $\mathsf{Ran}(\Pi^0) \cup \mathsf{Ran}(\pi^0)$ and $\widetilde{\Theta}_*$ is permutation compatible with $\mathsf{Ran}(\Pi^1) \cup \mathsf{Ran}(\pi^1)$.

This is useful for computing the ratio of the real to ideal interpolation probabilities of a good transcript τ through the following lemma:

Lemma 5. *Let $\tau = (\hat{\tau}_q, \tau_p)$ be a good transcript. Then*

$$\frac{\Pr\left[\mathsf{D}_{\mathrm{re}} = \tau\right]}{\Pr\left[\mathsf{D}_{\mathrm{id}} = \tau\right]} \geq 1 - \frac{16qp^2 + 16q^2p + 4q^3}{2^{2n}}.$$

Proof. Ideal Interpolation Probability. Observe that the keys (k_0, k_1, k_2), the response tuple $\widetilde{T}$, and the (lazily sampled) π^0, π^1, Π^0 and Π^1 are jointly independent as each T_i is distributed independent of all the previously sampled values of T, all outputs of π^0 and π^1, the keys k_0, k_1 and k_2 as well as Π^0 and Π^1 (in the offline phase of the game). Let B denote the event $\{(\mathsf{\Pi}^0(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2)\,\|0) = v_i) \wedge (\mathsf{\Pi}^1(\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2)\,\|1) = z_i) \;\; \forall i \in \mathcal{F}\}$. Therefore,

$$\begin{aligned}
\Pr[\mathsf{D}_{\mathrm{id}} = \tau] &= \frac{1}{2^{3n}} \cdot \frac{1}{2^{nq}} \cdot \frac{1}{(2^n)_{p'_0}} \cdot \frac{1}{(2^n)_{p'_1}} \cdot \Pr\big[\, (\mathsf{\Pi}^0(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2)\,\|0) = v_i) \wedge \\
&\quad (\mathsf{\Pi}^1(\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2)\,\|1) = z_i) \;\; \forall i \in [q]\big] \\
&= \frac{1}{2^{3n}} \cdot \frac{1}{2^{nq}} \cdot \frac{1}{(2^n)_{p'_0}} \cdot \frac{1}{(2^n)_{p'_1}} \cdot \Pr[\mathsf{B}] \cdot \Pr\big[\, (\mathsf{\Pi}^0(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2)\,\|0) = v_i) \wedge \\
&\quad (\mathsf{\Pi}^1(\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2)\,\|1) = z_i) \;\; \forall i \in \mathcal{Q}^c \mid \mathsf{B}\big] \\
&= \frac{1}{2^{3n}} \cdot \frac{1}{2^{nq}} \cdot \frac{1}{(2^n)_{p'_0}} \cdot \frac{1}{(2^n)_{p'_1}} \cdot \frac{1}{|\mathcal{S}|} \cdot \frac{1}{(2^n - f - p'_0)_{r_0}} \cdot \frac{1}{(2^n - f - p'_1)_{r_1}}. \qquad (52)
\end{aligned}$$

Recall here that $\mathsf{\Pi}^0$ and $\mathsf{\Pi}^1$ are defined in two steps:

1. Elements of $\mathcal{S}$ are sampled randomly for all free indices $i \in \mathcal{F}$ (line 13 of stage II in Fig. 6.4) and thus $\Pr[\mathsf{B}] = |\mathcal{S}|^{-1}$.
2. The remaining input-output values of $\mathsf{\Pi}^0$ and $\mathsf{\Pi}^1$ are defined through lazy sampling (lines $4, 5, 11$ and 12 of stage III in Fig. 6.5).

In the second step of the sampling process, the oracle samples permutation outputs for r_0 and r_1 distinct values in such a manner that neither do they collide with the values sampled in the first step, nor with the values in the modified list τ'_p. Hence, we have

$$\begin{aligned}
\Pr\big[(\mathsf{\Pi}^0(\mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2)\,\|0) = v_i) \wedge (\mathsf{\Pi}^1(\mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2)\,\|1) = z_i) \;\forall i \in \mathcal{Q}^c \,|\, \mathsf{B}\big] \\
= \frac{1}{(2^n - f - p'_0)_{r_0}} \cdot \frac{1}{(2^n - f - p'_1)_{r_1}}.
\end{aligned}$$

Real Interpolation Probability. From the claim (6.6) stated previously in this section, it is obvious that $\widetilde{\Sigma}_*$ is permutation compatible with $\mathsf{Ran}(\mathsf{\Pi}^0) \cup \mathsf{Ran}(\pi^0)$ and $\widetilde{\Theta}_*$ is permutation compatible with $\mathsf{Ran}(\mathsf{\Pi}^1) \cup \mathsf{Ran}(\pi^1)$. Therefore,

$$\sum_{i=1}^{r_0} d_i^0 + \sum_{i=1}^{r_1} d_i^1 = |\mathcal{Q}^c| = (q' - f), \qquad (53)$$

since the number of non-fresh blocks is $(q' - f)$. We define two sets:

$$\begin{aligned}
\mathcal{U}_0 &:= \{i \in \mathcal{Q}^c : \mathsf{chop}_{\mathsf{LSB}}(\Sigma_i \oplus k_2)\,\|0 \text{ is fresh in } \widetilde{\Sigma}_*\}, \\
\mathcal{U}_1 &:= \{i \in \mathcal{Q}^c : \mathsf{chop}_{\mathsf{LSB}}(\Theta_i \oplus k_2)\,\|1 \text{ is fresh in } \widetilde{\Theta}_*\}.
\end{aligned}$$

$$\text{Clearly, } u_0 := |\mathcal{U}_0| = r_0 + f + \sum_{i=1}^{r_1} d_i^1, \;\; u_1 := |\mathcal{U}_1| = r_1 + f + \sum_{i=1}^{r_0} d_i^0.$$

One can easily verify that the number of distinct inputs to π^b $(b \in \{0,1\})$ is $\overline{u}_b := u_b + p'_b$. Hence,

$$\Pr[\mathsf{D}_{\mathrm{re}} = \tau] \;=\; \frac{1}{2^{3n}} \cdot \frac{1}{(2^n)_{\overline{u}_0}} \cdot \frac{1}{(2^n)_{\overline{u}_1}}. \qquad (54)$$

Computing the ratio. From Eqn.s (54) and (52),

$$\begin{aligned}
\frac{\Pr[\mathsf{D}_{\mathrm{re}} = \tau]}{\Pr[\mathsf{D}_{\mathrm{id}} = \tau]} &\overset{(2)}{=} \frac{2^{nq} \cdot ((2^n)_p)^2 \cdot (2^n - f - p'_0)_{r_0} \cdot (2^n - f - p'_1)_{r_1} \cdot |\mathcal{S}|}{(2^n)_{\overline{u}_0} \cdot (2^n)_{\overline{u}_1}} \\
&\overset{(3)}{\geq} 2^{n(q-f)} \cdot \mathsf{A}_1 \cdot \mathsf{A}_2 \cdot \left(1 - \frac{4fp'_0p'_1 + 4f^2(p'_0 + p'_1) + 4f^3}{2^{2n}}\right), \qquad (55)
\end{aligned}$$

where $\mathsf{A}_1 := \left(\frac{(2^n - p'_0)_f \cdot (2^n - f - p'_0)_{r_0}}{(2^n - p)_{u_1+s_1}}\right)$, $\mathsf{A}_2 := \left(\frac{(2^n - p'_1)_f \cdot (2^n - f - p'_1)_{r_1}}{(2^n - p)_{u_2+t_1}}\right)$.

Note that (3) follows from $p'_0 = p + s_1$ and $p'_1 = p + t_1$ and the following result from Lemma 1:

$$|\mathcal{S}| \geq \frac{(2^n - p'_0)_f \cdot (2^n - p'_1)_f}{2^{nf}} \cdot \underbrace{\left(1 - \frac{4fp'_0p'_1 + 4f^2(p'_0 + p'_1) + 4f^3}{2^{2n}}\right)}_{\Delta},$$

where we assume that $f + p'_0 \leq 2^{n-1}$ and $f + p'_1 \leq 2^{n-1}$.

Furthermore,

$$\mathsf{A}_1 = \left(\frac{(2^n - p'_0)_{f+r_0}}{(2^n - p)_{s_1} \cdot (2^n - p'_0)_{f+r_0} \cdot (2^n - p'_0 - f - r_0)_{\sum_{i=1}^{r_1} d^1_i}}\right)$$

and

$$\mathsf{A}_2 = \left(\frac{(2^n - p'_1)_{f+r_1}}{(2^n - p)_{t_1} \cdot (2^n - p'_1)_{f+r_1} \cdot (2^n - p'_1 - f - r_1)_{\sum_{i=1}^{r_0} d^0_i}}\right)$$

Therefore, from Eqn. (55),

$$\mathsf{P} = \frac{2^{n(q-f)} \cdot \Delta}{(2^n - p)_{s_1} \cdot (2^n - p'_0 - f - r_0)_{\sum_{i=1}^{r_1} d^1_i} \cdot (2^n - p)_{t_1} \cdot (2^n - p'_1 - f - r_1)_{\sum_{i=1}^{r_0} d^0_i}}.$$

Due to Eqn. (53), the total number of terms in the denominator of P is

$$\sum_{i=1}^{r_0} d^0_i + \sum_{i=1}^{r_1} d^1_i + s_1 + t_1 = q' - f + s_1 + t_1 = q - f,$$

as $q' = q - s_1 - t_1$. Not only does this number match exactly with the number of terms in its numerator (except the constant Δ), but also each term of the numerator (except Δ) is greater than each term of the denominator. Thus the term-by-term ratio is at least 1 and hence $\mathsf{P} \geq \Delta$. Finally, the inequalities $f \leq q$, $p'_0 \leq 2p$ and $p'_1 \leq 2p$ prove the result. □

7 Conclusion and Future Work

In this paper, we have shown a tight security bound of the public permutation-based pPMAC_Plus construction. Unlike PMAC_Plus, which is tightly secure for $2^{3n/4}$ queries, the public permutation-based pPMAC_Plus is tightly secure for $2^{2n/3}$ queries. Similar to pPMAC_Plus, analysing the security of the public permutation-based LightMAC_Plus construction is an interesting open problem.

References

[Bab] László Babai. The fourier transform and equations over finite abelian groups: An introduction to the method of trigonometric sums.

[BCDM19] Tim Beyne, Yu Long Chen, Christoph Dobraunig, and Bart Mennink. Elephant. *NIST LWC*, 2019.

[BDLN20] Arghya Bhattarcharjee, Avijit Dutta, Eik List, and Mridul Nandi. CENCPP - beyond-birthday-secure encryption from public permutations. *IACR Cryptol. ePrint Arch.*, 2020:602, 2020.

[BDPA13] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 313–314, 2013.

[BKL$^+$07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, pages 450–466, 2007.

[BKL$^+$13] Andrey Bogdanov, Miroslav Knezevic, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede. SPONGENT: the design space of lightweight cryptographic hashing. *IEEE Trans. Computers*, 62(10):2041–2053, 2013.

[BKL$^+$17] Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. Gimli : A cross-platform permutation. In *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, pages 299–320, 2017.

[BKR00] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of the cipher block chaining message authentication code. *J. Comput. Syst. Sci.*, 61(3):362–399, 2000.

[BPP$^+$17] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT: A small present - towards reaching the limit of lightweight encryption. In *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, pages 321–345, 2017.

[BR02] John Black and Phillip Rogaway. A block-cipher mode of operation for parallelizable message authentication. In *EUROCRYPT 2002*, pages 384–397, 2002.

[CDNY18] Avik Chakraborti, Nilanjan Datta, Mridul Nandi, and Kan Yasuda. Beetle family of lightweight and secure authenticated encryption ciphers. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):218–241, 2018.

[CLL$^+$14a] Shan Chen, Rodolphe Lampe, Jooyoung Lee, Yannick Seurin, and John Steinberger. Minimizing the two-round even-mansour cipher. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014*, pages 39–56, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[CLL$^+$14b] Shan Chen, Rodolphe Lampe, Jooyoung Lee, Yannick Seurin, and John P. Steinberger. Minimizing the two-round even-mansour cipher. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 39–56, 2014.

[CLM19] Yu Long Chen, Eran Lambooij, and Bart Mennink. How to build pseudorandom functions from public random permutations. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 266–293. Springer, 2019.

[CLS15] Benoit Cogliati, Rodolphe Lampe, and Yannick Seurin. Tweaking even-mansour ciphers. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 189–208. Springer, 2015.

[CN19] Bishwajit Chakraborty and Mridul Nandi. Orange. *NIST LWC*, 2019.

[CNTY20] Avik Chakraborti, Mridul Nandi, Suprita Talnikar, and Kan Yasuda. On the composition of single-keyed tweakable even-mansour for achieving BBB security. *IACR Trans. Symmetric Cryptol.*, 2020(2):1–39, 2020.

[CS14] Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 327–350, 2014.

[CS16] Benoît Cogliati and Yannick Seurin. Ewcdm: An efficient, beyond-birthday secure, nonce-misuse resistant mac. Cryptology ePrint Archive, Report 2016/525, 2016. https://ia.cr/2016/525.

[DDN$^+$17] 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.

[DDNP18] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Goutam Paul. Double-block hash-then-sum: A paradigm for constructing bbb secure prf. *IACR Transactions on Symmetric Cryptology*, 2018(3):36–92, 2018.

[DEMS19] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2. *NIST LWC*, 2019.

[DHP$^+$19] Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Xoodyak, a lightweight cryptographic scheme. *NIST LWC*, 2019.

[DN20a] Avijit Dutta and Mridul Nandi. BBB secure nonce based MAC using public permutations. In Abderrahmane Nitaj and Amr M. Youssef, editors, *Progress in Cryptology - AFRICACRYPT 2020 - 12th International Conference on Cryptology in Africa, Cairo, Egypt, July 20-22, 2020, Proceedings*, volume 12174 of *Lecture Notes in Computer Science*, pages 172–191. Springer, 2020.

[DN20b] Avijit Dutta and Mridul Nandi. Bbb secure nonce based mac using public permutations. Cryptology ePrint Archive, Report 2020/509, 2020. https://eprint.iacr.org/2020/509.

[DNT21a] Avijit Dutta, Mridul Nandi, and Suprita Talnikar. Permutation based edm: An inverse free bbb secure prf. *IACR Transactions on Symmetric Cryptology*, pages 31–70, 06 2021.

[DNT21b] Avijit Dutta, Mridul Nandi, and Suprita Talnikar. Permutation based EDM: an inverse free BBB secure PRF. *IACR Trans. Symmetric Cryptol.*, 2021(2):31–70, 2021.

[GPP11] Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON family of lightweight hash functions. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 222–239, 2011.

[GPPR12] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. *IACR Cryptology ePrint Archive*, 2012:600, 2012.

[IK03] Tetsu Iwata and Kaoru Kurosawa. OMAC: one-key CBC MAC. In *Fast Software Encryption, 2003*, pages 129–153, 2003.

[IMV16] Tetsu Iwata, Bart Mennink, and Damian Vizár. CENC is optimally secure. *IACR Cryptology ePrint Archive*, 2016:1087, 2016.

[KLL20] Seongkwang Kim, ByeongHak Lee, and Jooyoung Lee. Tight security bounds for double-block hash-then-sum macs. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 435–465. Springer, 2020.

[LNS18] Gaetan Leurent, Mridul Nandi, and Ferdinand Sibleyras. Generic attacks against beyond-birthday-bound macs. volume 2018, page 541, 2018.

[LPTY16] Atul Luykx, Bart Preneel, Elmar Tischhauser, and Kan Yasuda. A MAC mode for lightweight block ciphers. *IACR Cryptology ePrint Archive*, 2016:190, 2016.

[MMVH$^+$14] Nicky Mouha, Bart Mennink, Anthony Van Herrewege, Dai Watanabe, Bart Preneel, and Ingrid Verbauwhede. Chaskey: An efficient mac algorithm for 32-bit microcontrollers. In Antoine Joux and Amr Youssef, editors, *Selected Areas in Cryptography – SAC 2014*, pages 306–323, Cham, 2014. Springer International Publishing.

[Nai17] Yusuke Naito. Blockcipher-based macs: Beyond the birthday bound without message length. Cryptology ePrint Archive, Report 2017/852, 2017.

[Nan20a] Mridul Nandi. Mind the composition: Birthday bound attacks on EWCDMD and sokac21. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 203–220. Springer, 2020.

[Nan20b] Mridul Nandi. Mind the composition: Birthday bound attacks on EWCDMD and sokac21. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 203–220. Springer, 2020.

[NIS18] NIST. Lightweight cryptography, 2018. Online: https://csrc.nist.gov/Projects/Lightweight-Cryptography. Accessed: August 01, 2019.

[NM08] Mridul Nandi and Avradip Mandal. Improved security analysis of PMAC. *J. Mathematical Cryptology*, 2(2):149–162, 2008.

[Pat08] Jacques Patarin. The "Coefficients H" Technique. In *Selected Areas in Cryptography, SAC*, pages 328–345, 2008.

[RBB03] Phillip Rogaway, Mihir Bellare, and John Black. Sha-3 standard. *ACM Transactions on Information and System Security (TISSEC)*, 6(3):365–403, 2003.

[Yas10] Kan Yasuda. The sum of CBC macs is a secure PRF. In *CT-RSA 2010*, pages 366–381, 2010.

[Yas11] Kan Yasuda. A new variant of PMAC: beyond the birthday bound. In *CRYPTO 2011*, pages 596–609, 2011.

[ZWSW12] Liting Zhang, Wenling Wu, Han Sui, and Peng Wang. 3kf9: Enhancing 3gpp-mac beyond the birthday bound. In *ASIACRYPT 2012*, pages 296–312, 2012.