Towards Tight Security Bounds for OMAC, XCBC and TMAC

Soumya Chattopadhyay[1](✉), Ashwin Jha[2], and Mridul Nandi[1]

[1] Indian Statistical Institute, Kolkata, India
s.c.2357@gmail.com
[2] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
ashwin.jha@cispa.de

Abstract. OMAC — a single-keyed variant of CBC-MAC by Iwata and Kurosawa — is a widely used and standardized (NIST FIPS 800-38B, ISO/IEC 29167-10:2017) message authentication code (MAC) algorithm. The best security bound for OMAC is due to Nandi who proved that OMAC's pseudorandom function (PRF) advantage is upper bounded by $O(q^2\ell/2^n)$, where n, q, and ℓ, denote the block size of the underlying block cipher, the number of queries, and the maximum permissible query length (in terms of n-bit blocks), respectively. In contrast, there is no attack with matching lower bound. Indeed, the best known attack on OMAC is the folklore birthday attack achieving a lower bound of $\Omega(q^2/2^n)$. In this work, we close this gap for a large range of message lengths. Specifically, we show that OMAC's PRF security is upper bounded by $O(q^2/2^n + q\ell^2/2^n)$. In practical terms, this means that for a 128-bit block cipher, and message lengths up to 64 GB, OMAC can process up to 2^{64} messages before rekeying (same as the birthday bound). In comparison, the previous bound only allows 2^{48} messages. As a side-effect of our proof technique, we also derive similar tight security bounds for XCBC (by Black and Rogaway) and TMAC (by Kurosawa and Iwata). As a direct consequence of this work, we have established tight security bounds (in a wide range of ℓ) for all the CBC-MAC variants, except for the original CBC-MAC.

Keywords: OMAC · CMAC · XCBC · TMAC · CBC-MAC · PRF · Tight security

1 Introduction

Message Authentication Code (or, MAC) algorithms are symmetric-key primitives which are used for data authenticity and integrity. The sender generates a short tag based on message and a secret key which can be recomputed by any authorized receiver. MACs are commonly designed either based on a hash function or a block cipher. CBC-MAC is a block cipher-based MAC (message

A. Jha carried out this work in the framework of the French-German-Center for Cybersecurity, a collaboration of CISPA and LORIA.

S. Agrawal and D. Lin (Eds.): ASIACRYPT 2022, LNCS 13791, pp. 348–378, 2022.
https://doi.org/10.1007/978-3-031-22963-3_12

authentication code) which is based on the CBC mode of operation invented by Ehrsam et al. [11]. Given an n-bit block cipher E instantiated with a key K, the CBC-MAC construction is defined recursively as follows: for any $x \in \{0,1\}^n$, $\mathsf{CBC}_{E_K}(x) := E_K(x)$. For all $m = (m[1], \ldots, m[\ell]) \in (\{0,1\}^n)^\ell$ where $\ell \geq 2$, we define

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

It was an international standard, and has been proven secure for fixed-length messages or prefix-free message spaces (i.e., no message is a prefix to another message). Simple length extension attacks prohibit its usage for arbitrary length messages. However, appropriately chosen operations to process the last block can resist these attacks. One such idea was first applied in EMAC [2,4], where the CBC-MAC output was encrypted using an independently keyed block cipher. It worked for all messages with lengths that are divisible by the block size of the underlying block cipher. Black and Rogaway proposed [5] three-keyed constructions, ECBC, FCBC, and XCBC, which are proven to be secure against adversaries querying arbitrary length messages. Later, in back-to-back works, Iwata and Kurosawa proposed two improved constructions (in terms of the key size), namely, TMAC [17] that uses two keys, and OMAC[1] [12] that requires just a single key. Nandi proposed [20] GCBC1 and GCBC2, a slight improvement over OMAC in terms of the number of block cipher calls for multi-block messages.

1.1 Related Works and Motivation

It is well-established [1] that the security of any deterministic MAC can be quantified via the pseudorandom function (or PRF[2]) security. Consequently, most of the works on CBC-MAC variants analyze their PRF security. For constructions like ECBC, FCBC and EMAC, Pietrzak [25] showed a PRF bound of $O(q^2/2^n)$ for $\ell < 2^{n/8}$, where q and ℓ denote the number of messages and the maximum permissible length (no. of n-bit blocks) of the messages. Later, Jha and Nandi [15] discovered a flaw in the proof of the earlier bound and showed a bound of $O(q/2^{n/2})$ up to $\ell < 2^{n/4}$. However, in these constructions an extra (independent) block cipher is called at the end. Considering the number of block cipher calls, XCBC, TMAC and OMAC are better choices. XCBC uses two independent masking keys for the last block which are used depending on whether the last block is padded or not. In case of TMAC, the two masking keys are derived from a single n-bit key. OMAC optimized the key derivation further. Here, both the keys are derived using the underlying block cipher itself. Thus, it is much better in this respect. Classical bound for these constructions was $O(\sigma^2/2^n)$ [5,17], σ being the total number blocks among all the messages. Later, in a series of work [13,19,21,22], the improved bounds for XCBC, TMAC, and OMAC were shown to be in the form of $O(q^2\ell/2^n)$, $O(\sigma^2/2^n)$ and $O(\sigma q/2^n)$. Interestingly, it has also

[1] This is same as CMAC [10] — a NIST recommended AES based MAC — for appropriate choice of constants.

[2] A keyed construction is called a PRF if it is computationally infeasible to distinguish it from a random function.

been shown in [14] that if we use a PRF, instead of a block cipher in these constructions, there is an attack with roughly $\Omega(q^2\ell/2^n)$ advantage, which is tight. No such attack is known in the presence of a block cipher. This gives an implicit motivation to study the exact security of these constructions in the presence of block ciphers. In this paper, we aim to show birthday-bound security for these block cipher based MACs for a suitable range of message lengths.

In a different paradigm but with similar motivations, recently Chattopadhyay et al. [8] showed birthday-bound security for another standardized MAC called LightMAC [18]. However, similar result for original PMAC [6] is still an open problem (although a result is available for its variant in [7]). In addition to the improved bound for LightMAC, Chattopadhyay et al. proposed a new proof approach called the reset-sampling method. They also hinted (via a very brief discussion) that this method could be useful for proving better security for OMAC. However, the discussion in [8] is overly simplistic and contains no formal analysis of bad events. Indeed, the reset-sampling is more involved than anticipated in [8], giving rise to some crucial and tricky bad events (see Sect. 4). To their credit, they do say that

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

In this paper, we take up this topic and give a complete and rigorous analysis.

1.2 Our Contributions

In Sect. 3, we show that the PRF advantages for OMAC, XCBC and TMAC are upper bounded by $O\left(q^2/2^n\right) + O\left(q\ell^2/2^n\right)$, which is almost tight in terms of the number of queries q while $\ell \ll 2^{n/4}$. This bound is not exactly the birthday bound $O\left(q^2/2^n\right)$, but for any fixed target advantage, in terms of the limit on q it behaves almost like the birthday bound for a fairly good range of ℓ (see the following discussion). The proof of our security bound is given in Sect. 4 and follows the recently introduced reset-sampling approach [8]. These improved bounds, in combination with previous results [15,16] for EMAC, ECBC and FCBC, completely characterize (see Table 1) the security landscape of CBC-MAC variants for message lengths up to $2^{n/4}$ blocks.

A Note on the Tightness and Improvement in Bounds: In Fig. 1, we present a graph[3] comparing the best known bound for OMAC [21], i.e., $B_1(\ell, q) = 10q^2\ell/2^n$, the ideal birthday bound, i.e., $B_{\mathsf{id}} = q^2/2^n$, and the bound shown in this paper (see Theorem 3.1), i.e., $B_2(\ell, q) \approx \frac{16q^2}{2^n} + \frac{2q\ell^2}{2^n}$ (as the remaining terms are dominated by these two terms). In the graph, we show the trade-off curve for the parameters $X = \log \ell$ and $Y = \log q$, where log denotes "log base 2", for a fixed choice of advantage value, say $\epsilon = 2^{-a}$ for some $a \in \mathbb{N}$. Let $n_a := n - a$. Then, we have

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

Table 1. Summary of security (PRF advantage) bounds for the CBC-MAC family. Here n, q, ℓ, and σ denote the block size, number of queries, maximum permissible message length, and sum of message lengths of all q queries, respectively.

Scheme	**State-of-the-art**		**This paper**	
	Bound	Restriction	Bound	Restriction
CBC-MAC [11]	$O\left(\sigma q/2^n\right)$ [15,16]	$\ell = o\left(2^{n/3}\right)$	-	-
EMAC [2,4]	$O\left(q^2/2^n + q\ell^2/2^n\right)$ [15,16]	-	-	-
ECBC,FCBC [5]	$O\left(q^2/2^n + q\ell^2/2^n\right)$ [15,16]	-	-	-
XCBC [5], TMAC [17]	$O\left(q^2\ell/2^n\right)$ [19][a]	$\ell = o\left(2^{n/3}\right)$	$O\left(q^2/2^n + q\ell^2/2^n\right)$	-
	$O\left(\sigma^2/2^n\right)$ [13][a]	-		
OMAC [12]	$O\left(\sigma q/2^n\right)$ [21]	$\ell = o\left(2^{n/3}\right)$	$O\left(q^2/2^n + q\ell^2/2^n\right)$	-

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

$$B_{\mathsf{id}} : Y = \frac{n_a}{2} \quad B_1 : X + 2Y = n_a - \log 10 \quad B_2 : \log(16 \cdot 2^{2Y} + 2 \cdot 2^{2X+Y}) = n_a.$$

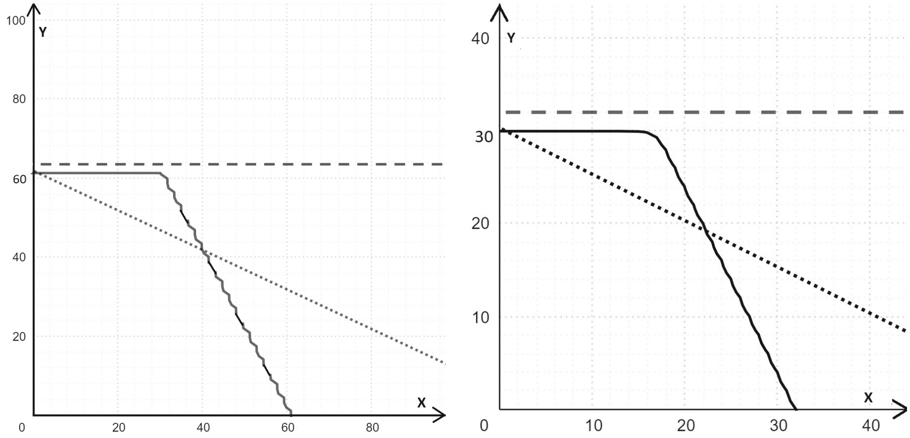
Looking at the equation related to the bound B_2 we can see that it is actually a combination of two linear equations: $2Y = n_a - 4$ and $2X + Y = n_a - 1$, the choice depending on whether $16q^2/2^n$ or $2q\ell^2/2^n$ dominates. Precisely, the curve expressing the relation between $\log \ell$ and $\log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

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

For example, if we take $(n, a) = (128, 32)$, the bound proved in this paper is very close to the birthday bound for $\ell \le 2^{25}$ and even after degrading, it remains better than the bound in [21] till $\ell \le 2^{32}$. Moreover, if we take $(n, a) = (128, 64)$, q remains 2^{30} until $\ell \le 2^{16}$ and degrades below the existing bound only after $\ell \ge 2^{22}$. Thus, if we consider the advantage in general terms, we can always take the minimum among the advantage proved in this paper and that proved in [21].

2 Preliminaries

For $n \in \mathbb{N}$, $[n]$ and $(n]$ denote the sets $\{1, 2, \ldots, n\}$ and $\{0\} \cup [n]$, respectively. The set of all bit strings (including the empty string $\perp$) is denoted $\{0, 1\}^*$. The length



1.1: For $\epsilon = 2^{-1}$

1.2: For $\epsilon = 2^{-64}$

Fig. 1. $(\log \ell, \log q)$-Trade-off Graph for the bounds of OMAC. For $n = 128$, and two different choices of the target advantage, $\epsilon = 2^{-1}$ (on the left), and $\epsilon = 2^{-64}$ (on the right), the above graphs show the relation between $X = \log \ell$ and $Y = \log q$. The *dashed*, *dotted* and *continuous* curves represent the equations B_{id}, B_1, and B_2, respectively.

of any bit string $x \in \{0,1\}^*$, denoted $|x|$, is the number of bits in x. For $n \in \mathbb{N}$, $\{0,1\}^n$ denotes the set of all bit strings of length n, and $\{0,1\}^{\leq n} := \bigcup_{i=0}^{n} \{0,1\}^i$. For $x, y \in \{0,1\}^*$, $z = x \| y$ denotes the concatenation of x and y. Additionally, x (resp. y) is called the *prefix* (resp. *suffix*) of z. For $x, y \in \{0,1\}^*$, let $\mathsf{Prefix}(x, y)$ denote the length of the largest possible common prefix of x and y. For $1 \leq k \leq n$, we define the falling factorial $(n)_k := n!/(n-k)! = n(n-1)\cdots(n-k+1)$. Any pair of q-tuples $\widetilde{x} = (x_1, \ldots, x_q)$ and $\widetilde{y} = (y_1, \ldots, y_q)$, are said to be *permutation compatible*, denoted $\widetilde{x} \leftrightsquigarrow \widetilde{y}$, if $(x_i = x_j) \iff (y_i = y_j)$, for all $i \neq j$. By an abuse of notation, we also use $\widetilde{x}$ to denote the set $\{x_i : i \in [q]\}$ for any $\widetilde{x}$.

2.1 Security Definitions

Distinguishers: A (q, T)-distinguisher $\mathscr{A}$ is an oracle Turing machine, that makes at most q oracle queries, runs in time at most T, and outputs a single bit. For any oracle $\mathcal{O}$, we write $\mathscr{A}^{\mathcal{O}}$ to denote the output of $\mathscr{A}$ after its interaction with $\mathcal{O}$. By convention, $T = \infty$ denotes computationally unbounded (information-theoretic) and deterministic distinguishers. In this paper, we assume that the distinguisher is non-trivial, i.e., it never makes a duplicate query. Let $\mathbb{A}(q, T)$ be the class of all non-trivial distinguishers limited to q queries and T computations.

Primitives and Their Security: The set of all functions from $\mathcal{X}$ to $\mathcal{Y}$ is denoted $\mathcal{F}(\mathcal{X}, \mathcal{Y})$, and the set of all permutations of $\mathcal{X}$ is denoted $\mathcal{P}(\mathcal{X})$. We simply write $\mathcal{F}(a, b)$ and $\mathcal{P}(a)$, whenever $\mathcal{X} = \{0,1\}^a$ and $\mathcal{Y} = \{0,1\}^b$. For a finite set $\mathcal{X}$, $\mathsf{X} \leftarrow_{\$} \mathcal{X}$ denotes the uniform at random sampling of X from $\mathcal{X}$.

Pseudorandom Function: A $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$-*keyed function* F with key space $\mathcal{K}$, domain $\mathcal{X}$, and range $\mathcal{Y}$ is a function $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. We write $F_k(x)$ for $F(k, x)$.

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

$$\mathbf{Adv}_F^{\mathsf{prf}}(\mathscr{A}) = \mathbf{Adv}_{F;\Gamma}(\mathscr{A}) := \left| \Pr_{\mathsf{K} \leftarrow\$ \mathcal{K}} \left(\mathscr{A}^{F_{\mathsf{K}}} = 1\right) - \Pr_{\Gamma \leftarrow\$ \mathcal{F}(\mathcal{X},\mathcal{Y})} \left(\mathscr{A}^{\Gamma} = 1\right) \right|. \quad (2)$$

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

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

Pseudorandom Permutation: For some $n \in \mathbb{N}$, a $(\mathcal{K}, \mathcal{B})$-*block cipher* E with key space $\mathcal{K}$ and block space $\mathcal{B} := \{0,1\}^n$ is a $(\mathcal{K}, \mathcal{B}, \mathcal{B})$-keyed function, such that $E(k, \cdot)$ is a permutation over $\mathcal{B}$ for any key $k \in \mathcal{K}$. We write $E_k(x)$ for $E(k, x)$.

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

$$\mathbf{Adv}_E^{\mathsf{prp}}(\mathscr{A}) = \mathbf{Adv}_{E;\Pi}(\mathscr{A}) := \left| \Pr_{\mathsf{K} \leftarrow\$ \mathcal{K}} \left(\mathscr{A}^{E_{\mathsf{K}}} = 1\right) - \Pr_{\Pi \leftarrow\$ \mathcal{P}(n)} \left(\mathscr{A}^{\Pi} = 1\right) \right|. \quad (3)$$

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

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

2.2 H-coefficient Technique

Let $\mathscr{A}$ be a computationally unbounded and deterministic distinguisher that's trying to distinguish the real oracle $\mathcal{O}_1$ from the ideal oracle $\mathcal{O}_0$. The collection of all queries and responses that $\mathscr{A}$ made and received to and from the oracle, is called the *transcript* of $\mathscr{A}$, denoted as ν. Let V_1 and V_0 denote the transcript random variable induced by $\mathscr{A}$'s interaction with $\mathcal{O}_1$ and $\mathcal{O}_0$, respectively. Let $\mathcal{V}$ be the set of all transcripts. A transcript $\nu \in \mathcal{V}$ is said to be *attainable* if $\Pr(\mathsf{V}_0 = \nu) > 0$, i.e., it can be realized by $\mathscr{A}$'s interaction with $\mathcal{O}_0$.

Following these notations, we state the main result of the so-called H-coefficient technique [23,24] in Theorem 2.1. A proof of this result is available in [24].

Theorem 2.1 [H-coefficient]. *For $\epsilon_1, \epsilon_2 \geq 0$, suppose there is a set $\mathcal{V}_{\mathsf{bad}} \subseteq \mathcal{V}$, referred as the set of all bad transcripts, such that the following conditions hold:*

- $\Pr\left(\mathsf{V}_0 \in \mathcal{V}_{\mathsf{bad}}\right) \leq \epsilon_1$*; and*
- *For any $\nu \in \mathcal{V} \setminus \mathcal{V}_{\mathsf{bad}}$, ν is attainable and* $\dfrac{\Pr\left(\mathsf{V}_1 = \nu\right)}{\Pr\left(\mathsf{V}_0 = \nu\right)} \geq 1 - \epsilon_2$.

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

$$\mathbf{Adv}_{\mathcal{O}_1;\mathcal{O}_0}(\mathscr{A}) \leq \epsilon_1 + \epsilon_2.$$

Reset-Sampling Method: In H-coefficient based proofs, often we release additional information to the adversary in order to make it easy to define the bad transcripts. In such scenarios, one has to define how this additional information is sampled, and naturally the sampling mechanism is construction specific. The reset-sampling method [8] is a sampling philosophy, within this highly mechanized setup of H-coefficient technique, where some of the variables are reset/resampled (hence the name) depending upon the consistency requirement for the overall transcript. We employ this sampling approach in our proof.

3 The CBC-MAC Family

Throughout, n denotes the *block size*, $\mathcal{B} := \{0,1\}^n$, and any $x \in \mathcal{B}$ is referred as a *block*. For any non-empty $m \in \{0,1\}^*$, $(m[1], \ldots, m[\ell_m]) \xleftarrow{n} m$ denotes the *block parsing* of m, where $|m[i]| = n$ for all $1 \leq i \leq \ell_m - 1$ and $1 \leq |m[\ell_m]| \leq n$. In addition, we associate a boolean flag δ_m to each $m \in \{0,1\}^*$, which is defined as

$$\delta_m := \begin{cases} -1 & \text{if } |m| = n\ell_m, \\ 0 & \text{otherwise.} \end{cases}$$

For any $m \in \{0,1\}^{\leq n}$, we define

$$\overline{m} := \begin{cases} m\|10^{n-|m|-1} & \text{if } |m| < n, \\ m & \text{otherwise.} \end{cases}$$

CBC FUNCTION: The CBC function, based on a permutation[4] $\pi \in \mathcal{P}(n)$, takes as input a non-empty message $m \in \mathcal{B}^*$ and computes the output $\mathsf{CBC}_\pi(m) := y_m^\pi[\ell_m]$ inductively as described below:

$y_m^\pi[0] = 0^n$ and for $1 \leq i \leq \ell_m$, we have

$$\begin{aligned} x_m^\pi[i] &:= y_m^\pi[i-1] \oplus m[i], \\ y_m^\pi[i] &:= \pi(x_m^\pi[i]), \end{aligned} \tag{4}$$

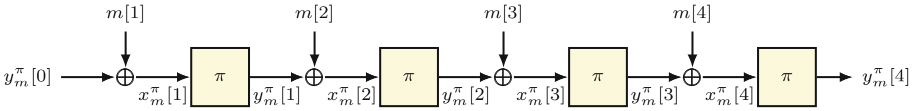


Fig. 2. Evaluation of CBC function over a 4-block message m.

where $(m[1],\ldots,m[\ell_m]) \xleftarrow{n} m$. For empty message, we define the CBC output as the constant 0^n. Figure 2 illustrates the evaluation of CBC function over a 4-block message m.

Given the definition of CBC_π, one can easily define all the variants of $\mathsf{CBC\text{-}MAC}$. Here, we define XCBC, TMAC and OMAC— the three constructions that we study in this paper.

XCBC: The XCBC algorithm is a three-key construction, based on a permutation $\pi \in \mathcal{P}(n)$ and keys $(L_{-1}, L_0) \in \mathcal{B}^2$, that takes as input a non-empty message $m \in \{0,1\}^*$, and computes the output

$$\mathsf{XCBC}_{\pi,L_{-1},L_0}(m) := t = \pi\left(\mathsf{CBC}_\pi\left(m^*\right) \oplus \overline{m[\ell_m]} \oplus L_{\delta_m}\right), \tag{5}$$

where $(m[1],\ldots,m[\ell_m]) \xleftarrow{n} m$, and $m^* := m[1]\|\cdots\|m[\ell_m-1]$.

TMAC: The TMAC algorithm is a two-key construction, based on a permutation $\pi \in \mathcal{P}(n)$ and key $L \in \mathcal{B}$, that takes as input a non-empty message $m \in \{0,1\}^*$, and computes the output

$$\mathsf{TMAC}_{\pi,L}(m) := t = \pi\left(\mathsf{CBC}_\pi\left(m^*\right) \oplus \overline{m[\ell_m]} \oplus \mu_{\delta_m} \odot L\right), \tag{6}$$

where $(m[1],\ldots,m[\ell_m]) \xleftarrow{n} m$, $m^* := m[1]\|\cdots\|m[\ell_m-1]$, μ_{-1} and μ_0 are constants chosen from $\mathrm{GF}(2^n)$ (viewing $\mathcal{B}$ as $\mathrm{GF}(2^n)$), such that $\mu_{-1}, \mu_0, 1\oplus\mu_{-1}, 1\oplus\mu_0$ are all distinct and not equal to either 0 or 1, and $\odot$ denotes the field multiplication operation over $\mathrm{GF}(2^n)$ with respect to a fixed primitive polynomial. For the sake of uniformity, we define $L_{\delta_m} := \mu_{\delta_m} \odot L$ in context of TMAC.

OMAC: The OMAC algorithm is a single-keyed construction, based on a permutation $\pi \in \mathcal{P}(n)$, that takes as input a non-empty message $m \in \{0,1\}^*$, and computes the output

$$\mathsf{OMAC}_{\pi}(m) := t = \pi\left(\mathsf{CBC}_\pi\left(m^*\right) \oplus \overline{m[\ell_m]} \oplus \mu_{\delta_m} \odot \pi(0^n)\right), \tag{7}$$

where $(m[1],\ldots,m[\ell_m]) \xleftarrow{n} m$, $m^* := m[1]\|\cdots\|m[\ell_m-1]$, μ_{-1} and μ_0 are constants chosen analogously as in the case of TMAC. For the sake of uniformity, we define $L_{\delta_m} := \mu_{\delta_m} \odot \pi(0^n)$ in context of OMAC.

[4] Instantiated with a block cipher in practical applications.

Input and Output Tuples: In the context of CBC evaluation within OMAC, we refer to $x_m^\pi := (x_m^\pi[1], \ldots, x_m^\pi[\ell_m - 1])$ and $y_m^\pi := (y_m^\pi[0], \ldots, y_m^\pi[\ell_m - 1])$ as the *intermediate input* and *output* tuples, respectively, associated to π and m. We define the final input variable as $x_m^\pi[\ell_m] := y_m^\pi[\ell_m - 1] \oplus \overline{m[\ell_m]} \oplus \mu_{\delta_m} \odot \pi(0^n)$. Clearly, the input and output tuples (including the final input) are well defined for OMAC. Analogous definitions are possible (and useful in proof) for XCBC and TMAC as well. It is worth noting that the intermediate input tuple x_m^π is uniquely determined by the intermediate output tuple y_m^π and the message m, and it is independent of the permutation π. Going forward, we drop π from the notations, whenever it is clear from the context.

3.1 Tight Security Bounds for OMAC, XCBC and TMAC

The main technical result of this paper, given in Theorem 3.1, is a tight security bound for OMAC for a wide range of message lengths. The proof of this theorem is postponed to Sect. 4. In addition, we also provide similar result for XCBC and TMAC in Theorem 3.2. We skip the proof since it is almost identical to the one for Theorem 3.1, and has slightly less relevance given that a more efficient and standardized algorithm OMAC already achieves similar security. In what follows we define

$$\begin{aligned}\epsilon'(q,\ell) := {} & \frac{16q^2 + q\ell^2}{2^n} + \frac{8q^2\ell^4 + 32q^3\ell^2 + 2q^2\ell^3}{2^{2n}} \\ & + \frac{3q^3\ell^5 + 143q^3\ell^6 + 11q^4\ell^3}{2^{3n}} + \frac{17q^4\ell^6 + 5462q^4\ell^8}{2^{4n}}.\end{aligned}$$

Theorem 3.1 (OMAC bound). *Let $q, \ell, \sigma, T > 0$. For $q + \sigma \leq 2^{n-1}$, the PRF insecurity of OMAC, based on block cipher E_{K}, against $\mathbb{A}(q,T)$ is given by*

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathit{OMAC}_{E_{\mathsf{K}}}}(q,\ell,\sigma,T) \leq \mathbf{Adv}^{\mathsf{prp}}_{E}(q+\sigma,T') + \frac{4\sigma}{2^n} + \epsilon'(q,\ell), \tag{8}$$

where q denotes the number of queries, ℓ denotes an upper bound on the number of blocks per query, σ denotes the total number of blocks present in all q queries, $T' = T + \sigma O(T_E)$ and T_E denotes the runtime of E.

Theorem 3.2 (XCBC-TMAC bound). *Let $q, \ell, \sigma, T > 0$. For $q + \sigma \leq 2^{n-1}$, the PRF insecurity of XCBC and TMAC, based on block cipher E_{K} and respective masking keys $(\mathsf{L}, \mathsf{L}_{-1}, \mathsf{L}_0)$, against $\mathbb{A}(q,T)$ is given by*

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathit{XCBC}_{E_{\mathsf{K}},\mathsf{L}_{-1},\mathsf{L}_0}}(q,\ell,\sigma,T) \leq \mathbf{Adv}^{\mathsf{prp}}_{E}(q+\sigma,T') + \epsilon'(q,\ell) \tag{9}$$

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathit{TMAC}_{E_{\mathsf{K}},\mathsf{L}}}(q,\ell,\sigma,T) \leq \mathbf{Adv}^{\mathsf{prp}}_{E}(q+\sigma,T') + \epsilon'(q,\ell) \tag{10}$$

where q denotes the number of queries, ℓ denotes an upper bound on the number of blocks per query, σ denotes the total number of blocks present in all q queries, $T' = T + \sigma O(T_E)$ and T_E denotes the runtime of E.

Proof of this theorem is almost same as that of Theorem 3.1. The bad event on a collision on zero block input is redundant and hence dropped here. Rest of the proof remains the same and so we skip the details.

Remark 3.1. Note that the actual advantage cannot exceed 1. Let us denote $\frac{q^2}{2^n} = \alpha$ and $\frac{q\ell^2}{2^n} = \beta$. Looking at $\epsilon(q,\ell)$ (where $\epsilon(q,\ell) = \epsilon'(q,\ell) + \frac{4\sigma}{2^n}$ in case of OMAC and $\epsilon(q,\ell) = \epsilon'(q,\ell)$ in case of XCBC,TMAC), we see that any term in the expression is upper bounded by $c \cdot \alpha^s \beta^t$ for some constant c and $s, t \geq 0$ such that at least one of s and t is at least 1. As we can assume both α, β to be less than 1, each $\alpha^s\beta^t$ will be less than or equal to α or β. Thus, the above PRF-advantage expressions for $\mathsf{MAC} \in \{\mathsf{OMAC}, \mathsf{XCBC}, \mathsf{TMAC}\}$ can be written as

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{MAC}}(q,\ell,\sigma) = O\left(\frac{q^2}{2^n}\right) + O\left(\frac{q\ell^2}{2^n}\right).$$

Indeed, under the assumption that $\ell \leq 2^{n/4-0.5}$ and $q \leq 2^{n/2-1}$, one can simplify the above bounds to $20q^2/2^n + 23q\ell^2/2^n$.

A Note on The Proof Approach: In the analysis of OMAC, XCBC and TMAC, we have to handle the case that the final input collides with some intermediate input, the so-called *full collision* event. In earlier works the probability of this event is shown to be $q^2\ell/2^n$ (as there are less than $q\ell$ many intermediate inputs and q final inputs and any such collision happens with roughly $1/2^n$ probability). So, in a way they avoid handling this tricky event by disallowing it all together. In this work, we allow full collisions as long as the next intermediate input is not colliding with some other input (intermediate or final). Looking ahead momentarily, this is captured in `BadW3`. We can do this via the application of reset-sampling, resulting in a more amenable $(q^2/2^n + q\ell^2/2^n)$ bound.

4 Proof of Theorem 3.1

First, using the standard hybrid argument, we get

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{OMAC}_{E_\mathsf{K}}}(q,\ell,\sigma,T) \leq \mathbf{Adv}^{\mathsf{prp}}_{E}(q+\sigma,T') + \mathbf{Adv}^{\mathsf{prf}}_{\mathsf{OMAC}_\Pi}(q,\ell,\sigma,\infty). \tag{11}$$

Now, it is sufficient to bound $\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{OMAC}_\Pi}(q,\ell,\sigma,\infty)$, where the corresponding distinguisher $\mathscr{A}$ is computationally unbounded and deterministic. To bound this term, we employ the H-coefficient technique (see Sect. 2.2), and the recently introduced *reset-sampling* method [8]. The remaining steps of the proof are given in the remainder of this section.

4.1 Oracle Description and Corresponding Transcripts

Real Oracle: The real oracle corresponds to OMAC_Π. It responds faithfully to all the queries made by $\mathscr{A}$. Once the query-response phase is over, it releases all the intermediate inputs and outputs, as well as the masking keys L_{-1} and L_0 to $\mathscr{A}$. We write $\mathsf{L} = \Pi(0^n)$.

In addition, the real oracle releases three binary variables, namely, FlagT, FlagW and FlagX, all of which are degenerately set to 0. These flags are more of a technical requirement, and their utility will become apparent from the description of ideal oracle. For now, it is sufficient to note that these flags are degenerate in the real world.

Formally, we have $\mathsf{V}_1 := (\widetilde{\mathsf{M}}, \widetilde{\mathsf{T}}, \widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*, \widetilde{\mathsf{Y}}, \mathsf{L}_{-1}, \mathsf{L}_0, \mathsf{FlagT}, \mathsf{FlagW}, \mathsf{FlagX})$, where

- $\widetilde{\mathsf{M}} = (\mathsf{M}_1, \ldots, \mathsf{M}_q)$, the q-tuple of queries made by $\mathscr{A}$, where $\mathsf{M}_i \in \{0,1\}^*$ for all $i \in [q]$. In addition, for all $i \in [q]$, let $\ell_i := \left\lceil \frac{|\mathsf{M}_i|}{n} \right\rceil$.
- $\widetilde{\mathsf{T}} = (\mathsf{T}_1, \ldots, \mathsf{T}_q)$, the q-tuple of final outputs received by $\mathscr{A}$, where $\mathsf{T}_i \in \mathcal{B}$.
- $\widetilde{\mathsf{X}} = (\mathsf{X}_1, \ldots, \mathsf{X}_q)$, where X_i denotes the intermediate input tuple for the i-th query.
- $\widetilde{\mathsf{X}}^* = (\mathsf{X}_1[\ell_1], \ldots, \mathsf{X}_q[\ell_q])$, where $\mathsf{X}_i[\ell_i]$ denotes the final input for the i-th query.
- $\widetilde{\mathsf{Y}} = (\mathsf{Y}_1, \ldots, \mathsf{Y}_q)$, where Y_i denotes the intermediate output tuple for the i-th query.
- L_{-1} and L_0 denote the two masking keys. Note that L_{-1} and L_0 are easily derivable from L. So we could have simply released L. The added redundancy is to aid the readers in establishing an analogous connection between this proof and the proof for XCBC and TMAC.
- $\mathsf{FlagT} = \mathsf{FlagW} = \mathsf{FlagX} = 0$.

From the definition of OMAC, we know that $\Pi(\mathsf{X}_i[a]) = \mathsf{Y}_i[a]$ for all $(i, a) \in [q] \times [\ell_i]$. So, *in the real world we always have* $(0^n, \widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*) \leftrightsquigarrow (\mathsf{L}, \widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$, *i.e.,* $(0^n, \widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*)$ *is permutation compatible with* $(\mathsf{L}, \widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$. We keep this observation in our mind when we simulate the ideal oracle.

Ideal Oracle: By reusing notations from the real world, we represent the ideal oracle transcript as $\mathsf{V}_0 := (\widetilde{\mathsf{M}}, \widetilde{\mathsf{T}}, \widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*, \widetilde{\mathsf{Y}}, \mathsf{L}_{-1}, \mathsf{L}_0, \mathsf{FlagT}, \mathsf{FlagW}, \mathsf{FlagX})$. This should not cause any confusion, as we never consider the random variables V_1 and V_0 jointly, whence the probability distributions of the constituent variables will always be clear from the context.

The ideal oracle transcript is described in three phases, each contingent on some predicates defined over the previous stages. Specifically, the ideal oracle first initializes $\mathsf{FlagT} = \mathsf{FlagW} = \mathsf{FlagX} = 0$, and then follows the sampling mechanism given below:

PHASE I (QUERY-RESPONSE PHASE): In the query-response phase, the ideal oracle faithfully simulates $\Gamma \leftarrow\!\!\$\, \mathcal{F}(\{0,1\}^*, \mathcal{B})$. Formally, for $i \in [q]$, at the i-th query $\mathsf{M}_i \in \{0,1\}^*$, the ideal oracle outputs $\mathsf{T}_i \leftarrow\!\!\$\, \mathcal{B}$. The partial transcript generated at the end of the query-response phase is given by $(\widetilde{\mathsf{M}}, \widetilde{\mathsf{T}})$, where

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

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

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

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

Phase II (Offline Initial Sampling Phase):Onward, we must have $\mathsf{T}_i \neq \mathsf{T}_j$ whenever $i \neq j$, and $\mathsf{FlagT} = 0$, since this phase is only executed when BadT is false. In the offline phase, the ideal oracle's initial goal is to sample the input and output tuples in such a way that the intermediate input and output tuples are permutation compatible. For now we use notations W and Z, respectively, instead of X and Y, to denote the input and output tuples. This is done to avoid any confusions in the next step where we may have to reset some of these variables. To make it explicit, W and Z respectively denote the input and output tuples before resetting, and X and Y denote the input and output tuples after resetting.

Let P be a key-value table representing a partial permutation of $\mathcal{B}$, which is initialized to empty, i.e., the corresponding permutation is undefined on all points. We write P.domain and P.range to denote the set of all keys and values utilized till this point, respectively. The ideal oracle uses this partial permutation P to maintain permutation compatibility between intermediate input and output tuples, in the following manner:

```
Initial sampling
$\mathsf{L} \leftarrow_\$ \mathcal{B} \setminus \widetilde{\mathsf{T}}$
$\mathsf{L}_{-1} \leftarrow \mu_{-1} \odot \mathsf{L}$
$\mathsf{L}_0 \leftarrow \mu_0 \odot \mathsf{L}$
$\mathsf{P}(0^n) \leftarrow \mathsf{L}$
for $i = 1$ to $q$ do
  $\mathsf{Z}_i[0] \leftarrow 0^n$
  for $a = 1$ to $\ell_i - 1$ do
    $\mathsf{W}_i[a] \leftarrow \mathsf{Z}_i[a-1] \oplus \mathsf{M}_i[a]$
    if $\mathsf{W}_i[a] \in \mathsf{P.domain}$
      $\mathsf{Z}_i[a] \leftarrow \mathsf{P}(\mathsf{W}_i[a])$
    else
      $\mathsf{Z}_i[a] \leftarrow_\$ \mathcal{B} \setminus \left(\widetilde{\mathsf{T}} \cup \mathsf{P.range}\right)$
      $\mathsf{P}(\mathsf{W}_i[a]) \leftarrow \mathsf{Z}_i[a]$
  $\mathsf{W}_i[\ell_i] \leftarrow \mathsf{Z}_i[\ell_i - 1] \oplus \overline{\mathsf{M}}_i[\ell_i] \oplus \mathsf{L}_{\delta_{\mathsf{M}_i}}$
```

At this stage we have $\mathsf{Z}_i[a] = \mathsf{Z}_j[b]$ if and only if $\mathsf{W}_i[a] = \mathsf{W}_j[b]$ for all $(i,a) \in [q] \times [\ell_i - 1]$ and $(j,b) \in [q] \times [\ell_j - 1]$. In other words, $(0^n, \widetilde{\mathsf{W}}) \rightsquigarrow (\mathsf{L}, \widetilde{\mathsf{Z}})$. But it is obvious to see that the same might not hold between $(0^n, \widetilde{\mathsf{W}}, \widetilde{\mathsf{W}}^*)$ and $(\mathsf{L}, \widetilde{\mathsf{Z}}, \widetilde{\mathsf{T}})$. In the next stage our goal will be to reset some of the Z variables in such a way that the resulting input tuple is compatible with the resulting output tuple. However, in order to reset, we have to identify and avoid certain contentious input-output tuples.

Identifying Contentious Input-Outptut Tuples: We define several predicates on $(\widetilde{\mathsf{W}}, \widetilde{\mathsf{W}}^*)$, each of which represents some undesirable property of the sampled input and output tuples.

First, observe that L is chosen outside the set $\widetilde{\mathsf{T}}$. This leads to the first predicate:

$$\texttt{BadW1}: \ \exists (i,a) \in [q] \times [\ell_i], \text{ such that } (\mathsf{W}_i[a] = 0^n) \text{ and } (\ell_i > 1 \implies a > 1).$$

since, if BadW1 is true, then $(0^n, \widetilde{\mathsf{W}}^*)$ is not compatible with $(\mathsf{L}, \widetilde{\mathsf{T}})$. In fact, ¬BadW1 implies that none of the inputs, except the first input which is fully in adversary's control, can possibly be 0^n. This stronger condition will simplify the analysis greatly. The second predicate simply states that the final input tuple is not permutation compatible with the tag tuple, i.e., we have

$$\texttt{BadW2}: \ \exists i \neq j \in [q], \text{ such that } \mathsf{W}_i[\ell_i] = \mathsf{W}_j[\ell_j].$$

At this point, assuming $\neg(\texttt{BadW1} \vee \texttt{BadW2})$ holds true, the only way we can have permutation incompatibility is if $\mathsf{W}_i[a] = \mathsf{W}_j[\ell_j]$, for some $i, j \in [q]$ and $a \in [\ell_i - 1]$. A simple solution will be to reset $\mathsf{Z}_i[a]$ to T_j, for all such (i, a, j). In order to do this, we need that the following predicates must be false:

$$\texttt{BadW3}: \ \exists i, j, k \in [q], a \in [\ell_i - 1], b \in [\ell_k], \text{ such that}$$

$$(\mathsf{W}_i[a] = \mathsf{W}_j[\ell_j]) \wedge (\mathsf{W}_i[a+1] = \mathsf{W}_k[b]) \wedge \mathsf{Prefix}(\mathsf{M}_i, \mathsf{M}_k) < \max\{a+1, b\}.$$

$$\texttt{BadW4}: \ \exists i, j, k \in [q], a \neq b \in [\ell_i - 1], \text{ such that}$$

$$(\mathsf{W}_i[a] = \mathsf{W}_j[\ell_j]) \wedge (\mathsf{W}_i[b] = \mathsf{W}_k[\ell_k]).$$

$$\texttt{BadW5}: \ \exists i, j, k \in [q], a \in [\ell_i - 1], b \in [\ell_j - 1], \text{ such that}$$

$$(\mathsf{W}_i[a] = \mathsf{W}_j[\ell_j]) \wedge (\mathsf{W}_j[b] = \mathsf{W}_k[\ell_k]).$$

If BadW3 is true, then once $\mathsf{Z}_i[a]$ is reset, we lose the permutation compatibility since, the reset next input, i.e., $\mathsf{X}_i[a+1] = \mathsf{W}_i[a+1] \oplus \mathsf{Z}_i[a] \oplus \mathsf{T}_j = \mathsf{M}_i[a+1] \oplus \mathsf{T}_j \neq \mathsf{W}_k[b]$ with high probability, whereas $\mathsf{Z}_i[a+1] = \mathsf{Z}_k[b]$ with certainty. BadW4 simply represents the scenario where we may have to apply the initial resetting to two indices in a single message. Looking ahead momentarily, this may lead to contradictory *induced* resettings. Avoiding this predicate makes the resetting operation much more manageable. Similarly, avoiding BadW5, is just proactive prevention of contradictory resetting at $\mathsf{Z}_i[a]$, since if BadW5 occurs, then we may have a case where $\mathsf{X}_j[\ell_j]$ is reset due to induced resetting, leading to the case, $\mathsf{X}_i[a] \neq \mathsf{X}_j[\ell_j]$ and $\mathsf{Y}_i[a] = \mathsf{T}_j$, where recall that $\mathsf{Y}_i[a]$ is the resetting value of $\mathsf{Z}_i[a]$. We write

$$\texttt{BadW} := \texttt{BadW1} \vee \texttt{BadW2} \vee \texttt{BadW3} \vee \texttt{BadW4} \vee \texttt{BadW5}.$$

If BadW is true, then FlagW is set to 1, and $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*, \widetilde{\mathsf{Y}})$ is again defined degenerately, as in the case of BadT. Otherwise, the ideal oracle proceeds to the next and the final phase, i.e., the resetting phase.

Phase III.a Initial Resetting Phase: At this stage we must have $\neg(\mathtt{BadT} \vee \mathtt{BadW})$, i.e., $\mathsf{FlagW} = \mathsf{FlagT} = 0$. We describe the resetting phase in two sub-stages. First, we identify the indices affected by the initial resetting operation.

Definition 4.1 [full collision index]. *Any $(i, a, j) \in [q] \times [\ell_i - 1] \times [q]$ is called a full collision index (FCI) if $\mathsf{W}_i[a] = \mathsf{W}_j[\ell_j]$. Additionally, let*

$$\mathsf{FCI} := \{(i, a, j) : i, j \in [q], a \in [\ell_i - 1],\ \textit{such that } (i, a, j) \textit{ is an FCI}\}$$
$$\widetilde{\mathsf{FCI}} := \{(i, a) \in [q] \times [\ell_i - 1] : \exists j \in [q],\ \textit{such that } (i, a, j) \textit{ is an FCI}\}$$

The first sub-stage, executes a resetting for full collision indices in the following manner:

1. For all $(i, a, j) \in \mathsf{FCI}$, define $\mathsf{Y}_i[a] := \mathsf{T}_j$;
2. For all $(i, a, j) \in \mathsf{FCI}$, define

$$\mathsf{X}_i[a+1] := \mathsf{W}_i[a+1] \oplus \mathsf{Z}_i[a] \oplus \mathsf{Y}_i[a] = \overline{\mathsf{M}}_i[a+1] \oplus \mathsf{T}_j \oplus 1_{a=\ell_i-1} \odot \mathsf{L}_{\delta_{\mathsf{M}_i}},$$

where $1_{a=\ell_i-1}$ is an indicator variable that evaluates to 1 when $a = \ell_i - 1$, and 0 otherwise.

Once the initial resetting is executed, it may result in new permutation incompatibilities. This necessitates further resettings, referred as *induced resettings*, which require that the following predicates are false:

$\mathtt{BadX1}:\ \exists (i, a, j) \in \mathsf{FCI}, k \in [q], b \in [\ell_k] \setminus \{1\},$ such that

$$(\mathsf{X}_i[a+1] = \mathsf{W}_k[b]) \vee (\mathsf{X}_i[a+1] = 0^n).$$

$\mathtt{BadX2}:\ \exists (i, a, j) \in \mathsf{FCI}, k \in [q],$ such that

$$(\mathsf{X}_i[a+1] = \mathsf{M}_k[1]) \wedge (\mathsf{M}_i[a+2, \ldots, \ell_i] = \mathsf{M}_k[2, \ldots, \ell_k]).$$

$\mathtt{BadX3}:\ \exists (i, a, j), (k, b, l) \in \mathsf{FCI},$ such that $(\mathsf{X}_i[a+1] = \mathsf{M}_k[1])$.

$\mathtt{BadX4}:\ \exists (i, a, k), (j, b, l) \in \mathsf{FCI},$ such that

$$(\mathsf{X}_i[a+1] = \mathsf{X}_j[b+1]) \wedge (\mathsf{Prefix}(\mathsf{M}_i, \mathsf{M}_j) < \max\{a+1, b+1\}).$$

Here, the variable highlighted in red denotes the update after initial resetting. Let's review these predicates in slightly more details. First, $\mathtt{BadX1}$, represents the situation where after resetting the next input (highlighted text) collides with some intermediate input or 0^n. This would necessitate induced resetting at $\mathsf{Z}_i[a+1]$. In other words, if $\mathtt{BadX1}$ is false then no induced resettings occur, unless the next input collides with some first block input. This case is handled in the next two predicates. $\mathtt{BadX2}$ represents the situation when the next input collides with a first block and the subsequent message blocks are all same. This would induce a chain of resetting going all the way to the final input. As BadT is false, this would immediately result in a permutation incompatibility since tags are distinct. If BadX2 is false, then the chain of induced resetting must end at some point. BadX3 is used to avoid circular or contradictory resettings. It is analogous to BadW5 defined earlier. If it is false, then we know that the k-th message is free from resetting, so the induced resetting will be manageable. Finally, BadX4 represents the situation when two newly reset variables collide. We write

$$\texttt{BadX1234} := \texttt{BadX1} \vee \texttt{BadX2} \vee \texttt{BadX3} \vee \texttt{BadX4}$$

If BadX1234 is true, then FlagX is set to 1, and $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*, \widetilde{\mathsf{Y}})$ is again defined degenerately, as in the cases of BadT and BadW. Otherwise, the ideal oracle proceeds to the second and the final sub-stage of resetting.

<u>Phase III.b Induced Resetting Phase</u>: Here, the goal is to execute the induced resettings necessitated by the initial resetting operation.

First, we define the *index of induced resetting* for each $(i, a) \in \widetilde{\mathsf{FCI}}$, as the smallest index j such that $\mathsf{X}_i[a+1] = \mathsf{M}_j[1]$ and

$$\mathsf{Prefix}(\mathsf{M}_i[a+2, \ldots, \ell_i], \mathsf{M}_j[2, \ldots, \ell_j]) = \max\{\mathsf{Prefix}(\mathsf{M}_i[a+2, \ldots, \ell_i], \mathsf{M}_{j'}[2, \ldots, \ell_{j'}]) : j' \in [q]\},$$

i.e., $\mathsf{Prefix}(\mathsf{M}_i[a+2, \ldots, \ell_i], \mathsf{M}_j[2, \ldots, \ell_j])$ maximizes.

Definition 4.2 [induced collision sequence]. *A sequence of tuples* $((i, a+1, j, 1), \ldots, (i, a+p+1, j, p+1))$ *is called an induced collision sequence (ICS), if* $(i, a) \in \widetilde{\mathsf{FCI}}$, *and* j *is the index of induced resetting for* (i, a), *where* $p := \mathsf{Prefix}(\mathsf{M}_i[a+2, \ldots, \ell_i], \mathsf{M}_j[2, \ldots, \ell_j])$. *The individual elements of an ICS are referred as induced collision index (ICI). Additionally, we let*

$$\mathsf{ICI} := \{(i, a, j, b) : i, j \in [q], a \in [\ell_i - 1], b \in [\ell_j - 1],\ \textit{and } (i, a, j, b) \textit{ is an ICI.}\}$$
$$\widetilde{\mathsf{ICI}} := \{(i, a) \in [q] \times [\ell_i - 1] : \exists (j, b) \in [q] \times [\ell_j - 1],\ \textit{and } (i, a, j, b) \textit{ is an ICI.}\}$$

Now, as anticipated, in the second sub-stage of resetting, we reset the induced collision indices in the following manner:

1. For all $(i, a, j, b) \in \mathsf{ICI}$, define $\mathsf{Y}_i[a] := \mathsf{Z}_j[b]$;
2. For all $(i, a, j, b) \in \mathsf{ICI}$, define

$$\mathsf{X}_i[a+1] := \mathsf{W}_i[a+1] \oplus \mathsf{Z}_i[a] \oplus \mathsf{Y}_i[a] = \overline{\mathsf{M}}_i[a+1] \oplus \mathsf{Z}_j[b] \oplus 1_{a=\ell_i-1} \odot \mathsf{L}_{\delta_{\mathsf{M}_i}},$$

where $1_{a=\ell_i-1}$ is an indicator variable that evaluates to 1 when $a = \ell_i - 1$, and 0 otherwise.

Given $\neg$BadX1234, we know that the induced resetting must stop at some point before the final input. Now, it might happen that once the first chain of induced resetting stops, the next input again collides which may result in nested resetting or permutation incompatibility. The predicates BadX5, BadX6, and BadX7 below represent these scenarios.

- $\texttt{BadX5}: \ \exists (i, a, k, b) \in \mathsf{ICI}, l \in [q], b \in [\ell_l - 1],$ such that

$$(\mathsf{X}_i[a+2+p] = \mathsf{W}_l[b]) \vee (\mathsf{X}_i[a+2+p] = 0^n),$$

  where $p := \mathsf{Prefix}(\mathsf{M}_i[a+2, \ldots, \ell_i], \mathsf{M}_k[2, \ldots, \ell_k])$.

- $\texttt{BadX6}: \ \exists (i, a) \in \widetilde{\mathsf{FCI}}, (j, b, k, c) \in \mathsf{ICI}$, such that $(\mathsf{X}_i[a+1] = \mathsf{X}_j[b+2+p])$, where $p := \mathsf{Prefix}(\mathsf{M}_j[b+2, \ldots, \ell_j], \mathsf{M}_k[2, \ldots, \ell_k])$.

- $\texttt{BadX7}: \ \exists (i, a, k, c), (j, b, l, d) \in \mathsf{ICI}$, such that

$$(\mathsf{X}_i[a+2+p] = \mathsf{X}_j[b+2+p']) \wedge (\mathsf{Prefix}(\mathsf{M}_i, \mathsf{M}_j) < \max\{a+2+p, b+2+p'\}),$$

  where $p := \mathsf{Prefix}(\mathsf{M}_i[a+2, \ldots, \ell_i], \mathsf{M}_k[2, \ldots, \ell_k])$, and $p' := \mathsf{Prefix}(\mathsf{M}_j[b+2, \ldots, \ell_j], \mathsf{M}_l[2, \ldots, \ell_l])$.

Here, the variables highlighted in red and blue denote the update after initial resetting and induced resetting, respectively. These predicates are fairly self-explanatory. First $\texttt{BadX5}$ represents the situation that the immediate input after induced resetting collides with some intermediate input or 0^n. This may cause permutation incompatibility and would lead to nested induced resetting at $\mathsf{Z}_i[a+2+p]$. $\texttt{BadX6}$ handles a similar collision with a full collision resetted variable, and $\texttt{BadX7}$ handles the only remaining case where the immediate inputs after two different induced resetting collides. Note that, $\neg(\texttt{BadX5} \vee \texttt{BadX6} \vee \texttt{BadX7})$ would imply that for each message resetting stops at some point before the final input, and the next input is fresh.[5] We write

$$\texttt{BadX} := \texttt{BadX1} \vee \texttt{BadX2} \vee \texttt{BadX3} \vee \texttt{BadX4} \vee \texttt{BadX5} \vee \texttt{BadX6} \vee \texttt{BadX7}.$$

If $\texttt{BadX}$ is true, then FlagX is set to 1, and $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*, \widetilde{\mathsf{Y}})$ is again defined degenerately, as in the case of $\texttt{BadT}$ and $\texttt{BadW}$. Otherwise, for any remaining index $(i, a) \in [q] \times (\ell_i - 1] \setminus (\widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}})$, the ideal oracle resets as follows:

1. define $\mathsf{Y}_i[a] := \mathsf{Z}_i[a]$;
2. define $\mathsf{X}_i[a+1] := \mathsf{W}_i[a+1]$.

At this point, the ideal oracle transcript is completely defined. Intuitively, if the ideal oracle is not sampling $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*, \widetilde{\mathsf{Y}})$ degenerately at any stage, then we must have $(0^n, \widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*) \leftrightsquigarrow (\mathsf{L}, \widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$. The following proposition justifies this intuition.

Proposition 4.1. *For* $\neg(\texttt{BadT} \vee \texttt{BadW} \vee \texttt{BadX})$, *we must have* $(0^n, \widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*) \leftrightsquigarrow (\mathsf{L}, \widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$.

Proof. Let $\neg(\texttt{BadT} \vee \texttt{BadW} \vee \texttt{BadX})$ hold. Recall that $(0^n, \widetilde{\mathsf{W}}, \widetilde{\mathsf{W}}^*)$ may not be permutation compatible with $(\mathsf{L}, \widetilde{\mathsf{Z}}, \widetilde{\mathsf{T}})$. For any $(i, a) \in \widetilde{\mathsf{FCI}}$, there exists $i' \in [q]$

[5] Does not collide with any other input.

such that $\mathsf{W}_i[a] = \mathsf{W}_{i'}[\ell_{i'}]$ but $\mathsf{Z}_i[a] \neq \mathsf{T}_{i'}$. We apply the initial resetting to solve this issue. However, as a result of initial resetting, induced resetting takes place. Our goal is to show that the non-occurrence of the bad events assures that the compatibility is attained in the final reset tuples $(0^n, \widetilde{\mathsf{X}}, \widetilde{\mathsf{X}}^*)$ and $(\mathsf{L}, \widetilde{\mathsf{Y}}, \widetilde{\mathsf{T}})$. We prove all possible cases as follows:

- $\mathsf{X}_i[a] = 0^n \iff \mathsf{Y}_i[a] = \mathsf{L}$: If $a = 1$ and $\mathsf{X}_i[a] = 0$, then $(i, a) \notin \widetilde{\mathsf{FCI}}$ due to $\neg\texttt{BadW1}$. Also, $(i, 1) \notin \widetilde{\mathsf{ICI}}$. Thus, $\mathsf{Y}_i[a] = \mathsf{Z}_i[a] = \mathsf{L}$ and the converse also holds. Otherwise, due to $\neg\texttt{BadX1}$, $\mathsf{X}_i[a]$ can not be equal to 0. Also, due to $\neg\texttt{BadW1}$, $\mathsf{Y}_i[a]$ can not be equal to L.

- $\mathsf{X}_i[a] = \mathsf{X}_{i'}[\ell_{i'}] \iff \mathsf{Y}_i[a] = \mathsf{T}_{i'}$: For $(i, a) \in \widetilde{\mathsf{FCI}}$, this equivalence holds. Otherwise, $\mathsf{X}_i[a] = \mathsf{X}_{i'}[\ell_{i'}]$ can not hold due to $\neg(\texttt{BadX1} \vee \texttt{BadX5})$. Also $\mathsf{Y}_i[a] = \mathsf{T}_{i'}$ can not hold due to definition of $\widetilde{\mathsf{T}}$ and $\neg\texttt{BadX2}$.

- $\mathsf{X}_i[a] = \mathsf{X}_j[b] \iff \mathsf{Y}_i[a] = \mathsf{Y}_j[b]$: To prove this part we divide it in the following subcases:

  - $\boxed{(i, a), (i, b) \notin \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}}}$: Since in this case the variables are simply renamed due to definitions of resetting and $\neg\texttt{BadW3}$, the result follows from $\widetilde{\mathsf{W}} \leftrightsquigarrow \widetilde{\mathsf{Z}}$.

  - $\boxed{(i, a), (j, b) \in \widetilde{\mathsf{FCI}}}$: Since $(i, a), (j, b) \in \widetilde{\mathsf{FCI}}$, there exists unique $i', j' \in [q]$, such that $\mathsf{W}_i[a] = \mathsf{W}_{i'}[\ell_{i'}]$ and $\mathsf{W}_j[b] = \mathsf{W}_{j'}[\ell_{j'}]$. Now, note that $\mathsf{X}_i[a] = \mathsf{W}_i[a]$ and $\mathsf{X}_j[b] = \mathsf{W}_j[b]$ since $\widetilde{\mathsf{FCI}} \cap \widetilde{\mathsf{ICI}} = \emptyset$ due to $\neg\texttt{BadW4}$; $\mathsf{W}_{i'}[\ell_{i'}] = \mathsf{X}_{i'}[\ell_{i'}]$ and $\mathsf{W}_{j'}[\ell_{j'}] = \mathsf{X}_{j'}[\ell_{j'}]$ due to $\neg\texttt{BadW5}$. Therefore, we must have $\mathsf{X}_{j'}[\ell_{j'}] = \mathsf{W}_{j'}[\ell_{j'}] = \mathsf{W}_j[b] = \mathsf{X}_j[b] = \mathsf{X}_i[a] = \mathsf{W}_i[a] = \mathsf{W}_{i'}[\ell_{i'}] = \mathsf{X}_{i'}[\ell_{i'}]$, which is possible if and only if $i' = j'$ (since $\neg\texttt{BadW2}$ holds).

  - $\boxed{(i, a), (j, b) \in \widetilde{\mathsf{ICI}}}$: Since $(i, a), (j, b) \in \widetilde{\mathsf{ICI}}$, there exists $i', j' \in [q]$ and $a' \in [\ell_{i'} - 1], b' \in [\ell_{j'} - 1]$, such that $\mathsf{X}_i[a] = \mathsf{W}_{i'}[a']$ and $\mathsf{X}_j[b] = \mathsf{W}_{j'}[b']$. Further, $(i', a'), (j', b') \notin \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}}$ (due to $\neg\texttt{BadX3}$). If $\mathsf{X}_j[b] = \mathsf{X}_i[a]$, then we have $\mathsf{W}_{j'}[b'] = \mathsf{W}_{i'}[a']$. This gives us $\mathsf{Y}_j[b] = \mathsf{Z}_{j'}[b'] = \mathsf{Z}_{i'}[a'] = \mathsf{Y}_i[a]$ (due to $\widetilde{\mathsf{W}} \leftrightsquigarrow \widetilde{\mathsf{Z}}$). Similarly, $\mathsf{X}_i[a] \neq \mathsf{X}_j[b]$ implies $\mathsf{Y}_i[a] \neq \mathsf{Y}_j[b]$.

  - $\boxed{(i, a) \in \widetilde{\mathsf{FCI}} \text{ and } (j, b) \in \widetilde{\mathsf{ICI}}}$: Since $(i, a) \in \widetilde{\mathsf{FCI}}$, there exists a unique $i' \in [q]$, such that $\mathsf{X}_i[a] = \mathsf{W}_i[a] = \mathsf{W}_{i'}[\ell_{i'}] = \mathsf{X}_{i'}[\ell_{i'}]$ (the first equality is due to $\neg\texttt{BadW4}$, the second equality is due to the definition of full collision, the third equality is due to $\neg\texttt{BadW5}$). Since $(j, b) \in \widetilde{\mathsf{ICI}}$, we also have $\mathsf{X}_j[b] = \mathsf{W}_{j'}[b']$. If $\mathsf{X}_i[a] = \mathsf{X}_j[b]$, then $\mathsf{W}_{j'}[b'] = \mathsf{W}_{i'}[\ell_{i'}]$. Thus, $(j', b') = (i', \ell_{i'})$ due to $\neg\texttt{BadX3}$. Now, we have $\mathsf{Y}_i[a] = \mathsf{T}_{i'}$. Also, $\mathsf{Y}_j[b] = \mathsf{Y}_{j'}[b'] = \mathsf{Y}_{i'}[\ell_{i'}] = \mathsf{T}_{i'}$. Therefore, $\mathsf{Y}_i[a] = \mathsf{Y}_j[b]$. Moreover, $\mathsf{X}_i[a] \neq \mathsf{X}_j[b]$ implies that $\mathsf{Y}_i[a] \neq \mathsf{Y}_j[b]$ due to similar arguments as above and also $\neg\texttt{BadT}$.

  - $\boxed{(i, a) \in \widetilde{\mathsf{ICI}} \text{ and } (j, b) \in \widetilde{\mathsf{FCI}}}$: Similar as the above case.

- $\boxed{(i,a) \in \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}} \text{ and } (j,b) \notin \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}}}$: Since $(j,b) \notin \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}}$, we have $\mathsf{X}_j[b] = \mathsf{W}_j[b]$ and $\mathsf{Y}_j[b] = \mathsf{Z}_j[b]$. Suppose, $(i,a) \in \widetilde{\mathsf{FCI}}$. Then $\mathsf{X}_i[a] = \mathsf{X}_j[b]$ is not possible since it would imply that $(j,b) \in \widetilde{\mathsf{FCI}}$. Also, $\mathsf{Y}_i[a] = \mathsf{Y}_j[b]$ is not possible since it would contradict the definition of $\widetilde{\mathsf{T}}$. Now, suppose, $(i,a) \in \widetilde{\mathsf{ICI}}$. Therefore, $\mathsf{X}_i[a] = \mathsf{W}_{i'}[a']$ for some $i' \in [q]$ and $a' \in [\ell_{i'} - 1]$. If $\mathsf{X}_i[a] = \mathsf{X}_j[b]$, then $\mathsf{W}_j[b] = \mathsf{X}_j[b] = \mathsf{X}_i[a] = \mathsf{W}_{i'}[a']$. So, $\mathsf{Y}_j[b] = \mathsf{Z}_j[b] = \mathsf{Z}_{i'}[a'] = \mathsf{Y}_i[a]$. Similarly, $\mathsf{X}_i[a] \neq \mathsf{X}_j[b]$ implies $\mathsf{Y}_i[a] \neq \mathsf{Y}_j[b]$.
- $\boxed{(i,a) \notin \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}} \text{ and } (j,b) \in \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}}}$: Similar as the above case.

4.2 Transcript Analysis

Set of Transcripts: Given the description of transcript random variable corresponding to the ideal oracle, we can now define the set of transcripts $\mathcal{V}$ as the set of all tuples $\nu = (\widetilde{m}, \widetilde{t}, \widetilde{x}, \widetilde{x}^*, \widetilde{y}, l_{-1}, l_0, \text{flagT}, \text{flagW}, \text{flagX})$, where

- $\widetilde{m} = (m_1, \ldots, m_q)$, where $m_i \in \{0,1\}^*$ for $i \in [q]$. Let $\ell_i = \left\lceil \frac{|m_i|}{n} \right\rceil$ for $i \in [q]$.
- $\widetilde{t} = (t_1, \ldots, t_q)$, where $t_i \in \mathcal{B}$ for $i \in [q]$.
- $\widetilde{x} = (x_1, \ldots, x_q)$, where $x_i = (x_i[1], \ldots, x_i[\ell_i - 1])$ for $i \in [q]$.
- $\widetilde{x}^* = (x_1[\ell_1], \ldots, x_q[\ell_q])$.
- $\widetilde{y} = (y_1, \ldots, y_q)$, where $y_i = (y_i[0] = 0^n, y_i[1], \ldots, y_i[\ell_i - 1])$ for $i \in [q]$.
- $l_{-1} = \mu_{-1} \odot l, l_0 = \mu_0 \odot l$ where $l \in \mathcal{B}$ and μ_{-1}, μ_0 are constants chosen from $\mathrm{GF}(2^n)$ as defined before.
- $\text{flagT}, \text{flagW}, \text{flagX} \in \{0,1\}$.

Furthermore, the following must always hold:

1. if $\text{flagI} = 1$ for some $\text{I} \in \{\text{T}, \text{W}\}$, then $x_i[a] = y_j[b] = 0^n$ for all $i, j \in [q]$, $a \in [\ell_i]$, and $b \in [\ell_j - 1]$.
2. if $\text{flagT} = 0$, then t_i's are all distinct.
3. if $\text{flagI} = 0$ for all $\text{I} \in \{\text{T}, \text{W}, \text{X}\}$, then $x_i[a] = y_i[a-1] \oplus \overline{m}_i[a]$ and $(0^n, \widetilde{x}, \widetilde{y}^{\oplus}) \leftrightsquigarrow (L, \widetilde{y}, \widetilde{t})$.

The first two conditions are obvious from the ideal oracle sampling mechanism. The last condition follows from Proposition 4.1 and the observation that in ideal oracle sampling for any $\mathsf{I} \in \{\mathsf{T}, \mathsf{Z}, \mathsf{X}\}$, $\mathsf{FlagI} = 1$ if and only if BadI is true. Note that, condition 3 is vacuously true for real oracle transcripts.

Bad Transcript: A transcript $\nu \in \mathcal{V}$ is called *bad* if and only if the following predicate is true:

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

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

$$\mathcal{V}_{\mathsf{bad}} := \{\nu \in \mathcal{V} : \nu \text{ is bad.}\}.$$

All other transcript $\nu' = (\widetilde{m}, \widetilde{t}, \widetilde{x}, \widetilde{x}^*, \widetilde{y}, l_{-1}, l_0, \text{flagT}, \text{flagW}, \text{flagX}) \in \mathcal{V} \setminus \mathcal{V}_{\mathsf{bad}}$ are called *good*. From the preceding characterization of the set of transcripts, we conclude that for any good transcript ν', we must have $(0^n, \widetilde{x}, \widetilde{x}^*) \leftrightsquigarrow (L, \widetilde{y}, \widetilde{t})$. Henceforth, we drop flagT, flagW, and flagX for any good transcript with an implicit understanding that flagT = flagW = flagX = 0.

Following the H-coefficient mechanism, we have to upper bound the probability $\Pr(\mathsf{V}_0 \in \mathcal{V}_{\mathsf{bad}})$ and lower bound the ratio $\Pr(\mathsf{V}_1 = \nu)/\Pr(\mathsf{V}_0 = \nu)$ for any $\nu \in \mathcal{V} \setminus \mathcal{V}_{\mathsf{bad}}$.

Lemma 4.1 (bad transcript analysis). *For $q + \sigma \leq 2^{n-1}$, we have*

$$\begin{aligned}\Pr(\mathsf{V}_0 \in \mathcal{V}_{\mathsf{bad}}) \leq & \frac{4\sigma}{2^n} + \frac{16q^2 + q\ell^2}{2^n} + \frac{8q^2\ell^4 + 32q^3\ell^2 + 2q^2\ell^3}{2^{2n}} \\ & + \frac{3q^3\ell^5 + 143q^3\ell^6 + 11q^4\ell^3}{2^{3n}} + \frac{17q^4\ell^6 + 5462q^4\ell^8}{2^{4n}}.\end{aligned}$$

The proof of this lemma is postponed to Sect. 5.

Good Transcript: Now, fix a good transcript $\nu = (\widetilde{m}, \widetilde{t}, \widetilde{x}, \widetilde{x}^*, \widetilde{y}, l_{-1}, l_0)$. Let σ be the total number of blocks (and one additional for 0^n) and $\sigma' := |\widetilde{x} \cup \{0^n\}|$. Since, ν is good, we have $(0^n, \widetilde{x}, \widetilde{x}^*) \leftrightsquigarrow (L, \widetilde{y}, \widetilde{t})$. Then, we must have $|\widetilde{x}^*| = q$. Further, let $|\widetilde{x} \cap \widetilde{x}^*| = r$. Thus, $|\{0^n\} \cup \widetilde{x} \cup \widetilde{x}^*| = q + \sigma' - r$.

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

$$\Pr(\mathsf{V}_1 = \nu) = \frac{1}{(2^n)_{q+\sigma'-r}}. \tag{12}$$

Ideal World: In the ideal world, we employed a two stage sampling. First of all, we have

$$\Pr\left(\widetilde{\mathsf{T}} = \widetilde{t}, \mathsf{P}(0^n) = L\right) \leq \frac{1}{2^{nq}}, \tag{13}$$

since each T_i is sampled uniformly from the set $\mathcal{B}$ independent of others. Now, observe that all the full collision and induced collision indices are fully determined from the transcript ν itself. In other words, we can enumerate the set $\widetilde{\mathsf{CI}} := \widetilde{\mathsf{FCI}} \cup \widetilde{\mathsf{ICI}}$. Now, since the transcript is good, we must have $|\widetilde{\mathsf{CI}}| = \sigma - \sigma' + |\widetilde{x} \cap \widetilde{x}^*| = \sigma - \sigma' + r$, and for all indices $(i, a) \notin \widetilde{\mathsf{CI}}$, we have $\mathsf{Y}_i[a] = \mathsf{Z}_i[a]$. Thus, we have

$$\begin{aligned}\Pr\left(\mathsf{Y}_i[a] = y^i_a \wedge (i, a) \notin \widetilde{\mathsf{CI}} \mid \widetilde{\mathsf{T}} = \widetilde{t}\right) &= \Pr\left(\mathsf{Z}_i[a] = y^i_a \wedge (i, a) \notin \widetilde{\mathsf{CI}} \mid \widetilde{\mathsf{T}} = \widetilde{t}\right) \\ &= \frac{1}{(2^n - q)_{\sigma'-r}},\end{aligned} \tag{14}$$

where the second equality follows from the fact that truncation[6] of a without replacement sample from a set of size $(2^n - q)$ is still a without replacement sample from the same set. We have

$$\begin{aligned}\Pr\left(\mathsf{V}_0 = \omega\right) &= \Pr\left(\widetilde{\mathsf{T}} = \widetilde{t}\right) \times \Pr\left(\widetilde{\mathsf{Y}} = \widetilde{y} \mid \widetilde{\mathsf{T}} = \widetilde{t}\right) \\ &\le \frac{1}{2^{nq}} \times \Pr\left(\mathsf{Y}_i[a] = y_i[a] \wedge (i,a) \notin \widetilde{\mathsf{CI}} \mid \widetilde{\mathsf{T}} = \widetilde{t}\right) = \frac{1}{2^{nq}(2^n - q)_{\sigma'-r}}. \end{aligned} \tag{15}$$

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

Lemma 4.2 *For any $\nu \in \mathcal{V} \setminus \mathcal{V}_{\mathsf{bad}}$, we have* $\dfrac{\Pr\left(\mathsf{V}_1 = \nu\right)}{\Pr\left(\mathsf{V}_0 = \nu\right)} \ge 1.$

Proof The proof follows from dividing (12) by (15).

Using Theorem 2.1, and Lemma 4.1 and 4.2, we get

$$\begin{aligned}\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{OMAC}_\Pi}(q,\ell,\sigma,\infty) \le \frac{4\sigma}{2^n} + \frac{16q^2 + q\ell^2}{2^n} + \frac{8q^2\ell^4 + 32q^3\ell^2 + 2q^2\ell^3}{2^{2n}} \\ + \frac{3q^3\ell^5 + 143q^3\ell^6 + 11q^4\ell^3}{2^{3n}} + \frac{17q^4\ell^6 + 5462q^4\ell^8}{2^{4n}}. \end{aligned} \tag{16}$$

Theorem 3.1 follows from (11) and (16).

5 Proof of Lemma 4.1

Our proof relies on a graph-based combinatorial tool, called structure graphs [3,15]. A concise and complete description of this tool and relevant results are available in the full version of this paper [9, Appendix A]. Our aim will be to bound the probability of bad events only when they occur in conjunction with some "manageable" structure graphs. In all other cases, we upper bound the probability by the probability of realizing an unmanageable structure graph. Formally, we say that the structure graph $\mathcal{G}_{\mathsf{P}}(\widetilde{\mathsf{M}})$ is manageable if and only if:

1. for all $i \in [q]$, we have $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i)) = 0$, i.e., each M_i-walk is a path.
2. for all distinct $i, j \in [q]$, we have $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j)) \le 1$.
3. for all distinct $i, j, k \in [q]$, we have $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) \le 2$.
4. for all distinct $i, j, k, l \in [q]$, we have $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k, \mathsf{M}_l)) \le 3$.

[6] Removing some elements from the tuple.

Let `unman` denote the event that $\mathcal{G}_{\mathsf{P}}(\widetilde{\mathsf{M}})$ is unmanageable. Then, using [9, Corollary A.1], we have

$$\begin{aligned}\Pr(\mathtt{unman}) &\le \Pr\left(\exists i \in [q] : \mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i)) \ge 1\right) + \Pr\left(\exists i < j \in [q] : \mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j)) \ge 2\right)\\ &\quad + \Pr\left(\exists i < j < k \in [q] : \mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) \ge 3\right)\\ &\quad + \Pr\left(\exists i < j < k < l \in [q] : \mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k, \mathsf{M}_l)) \ge 4\right)\\ &\le \sum_{i\in[q]} \frac{(\ell_i-1)^2}{2^n} + \sum_{i<j\in[q]} \frac{(\ell_i+\ell_j-2)^4}{2^{2n}} + \sum_{i<j<k\in[q]} \frac{(\ell_i+\ell_j+\ell_k-3)^6}{2^{3n}}\\ &\quad + \sum_{i<j<k<l\in[q]} \frac{(\ell_i+\ell_j+\ell_k+\ell_l-4)^8}{2^{4n}}\\ &\le \frac{q\ell^2}{2^n} + \frac{8q^2\ell^4}{2^{2n}} + \frac{121.5q^3\ell^6}{2^{3n}} + \frac{5461.34q^4\ell^8}{2^{4n}}. \qquad (17)\end{aligned}$$

From now on we only consider manageable graphs. Observe that apart from the fact that a manageable graph is just a union of M_i-paths, there is an added benefit that it has no zero collision. Let $\mathtt{TU} := \neg(\mathtt{BadT} \vee \mathtt{unman})$ and $\mathtt{TUW} := \neg(\mathtt{BadT} \vee \mathtt{unman} \vee \mathtt{BadW})$. Now, we have

$$\begin{aligned}\Pr(\mathsf{V}_0 \in \mathcal{V}_{\mathsf{bad}}) &= \Pr\left((\mathsf{FlagT}=1) \vee (\mathsf{FlagW}=1) \vee (\mathsf{FlagX}=1)\right)\\ &\overset{1}{\le} \Pr(\mathtt{BadT} \vee \mathtt{BadW} \vee \mathtt{BadX})\\ &\le \Pr(\mathtt{BadT}) + \Pr(\mathtt{BadW}|\neg\mathtt{BadT}) + \Pr(\mathtt{BadX}|\neg(\mathtt{BadT} \vee \mathtt{BadW}))\\ &\overset{2}{\le} \Pr(\exists i \ne j : \mathsf{T}_i = \mathsf{T}_j) + \Pr(\mathtt{BadW}|\neg\mathtt{BadT}) + \Pr(\mathtt{BadX}|\neg(\mathtt{BadT} \vee \mathtt{BadW}))\\ &\overset{3}{\le} \frac{q^2}{2^{n+1}} + \Pr(\mathtt{unman}) + \Pr(\mathtt{BadW}|\mathtt{TU}) + \Pr(\mathtt{BadX}|\mathtt{TUW})\\ &\overset{4}{\le} \frac{0.5q^2+q\ell^2}{2^n} + \frac{8q^2\ell^4}{2^{2n}} + \frac{122q^3\ell^6}{2^{3n}} + \frac{5462q^4\ell^8}{2^{4n}}\\ &\quad + \Pr(\mathtt{BadW}|\mathtt{TU}) + \Pr(\mathtt{BadX}|\mathtt{TUW}) \qquad (18)\end{aligned}$$

Here, inequalities 1 and 2 follow by definition; 3 follows from the fact that T_i is chosen uniformly at random from $\mathcal{B}$ for each i; and 4 follows from (17).

Bounding $\Pr(\mathtt{BadW}|\neg(\mathtt{BadT} \vee \mathtt{unman}))$: Let $\mathtt{Ei} = \neg(\mathtt{TU} \vee \mathtt{BadW1} \vee \cdots \vee \mathtt{BadWi})$. We have

$$\begin{aligned}\Pr(\mathtt{BadW}|\mathtt{TU}) \le \Pr(\mathtt{BadW1}|\mathtt{TU}) &+ \Pr(\mathtt{BadW2}|\mathtt{E1}) + \Pr(\mathtt{BadW3}|\mathtt{E2})\\ &+ \Pr(\mathtt{BadW4}|\mathtt{E3}) + \Pr(\mathtt{BadW5}|\mathtt{E4}) \qquad (19)\end{aligned}$$

We bound the individual terms on the right hand side as follows:

Bounding $\Pr(\mathtt{BadW1}|\mathtt{TU})$*:* Fix some $(i, a) \in [q] \times [\ell_i]$. The only way we can have $\mathsf{W}_i[a] = 0^n$, for $1 < a < \ell_i$, is if $\mathsf{Z}_i[a-1] = \mathsf{M}_i[a]$. This happens with probability at most $(2^n - q)^{-1}$. For $a = \ell_i$, the equation

$$\mu_{\delta_{\mathsf{M}_i}} \odot \mathsf{L} \oplus \mathsf{Z}_i[\ell_i - 1] \oplus \overline{\mathsf{M}}_i[\ell_i] = 0^n$$

must hold non-trivially. The probability that this equation holds is bounded by at most $(2^n - q - 1)^{-1}$. Assuming $q + 1 \leq 2^{n-1}$, and using the fact that there can be at most σ choices for (i, a), we have

$$\Pr\left(\texttt{BadW1}|\texttt{TU}\right) \leq \frac{2\sigma}{2^n}. \tag{20}$$

Bounding $\Pr\left(\texttt{BadW2}|\texttt{E1}\right)$*:* Fix some $i \neq j \in [q]$. Since $\neg\texttt{unman}$ holds, we know that $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j)) \leq 1$. We handle the two resulting cases separately:

(A) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j)) = 1$: Suppose the collision source of the only accident are (i, a) and (j, b). Then, we have the following system of two equations

$$\begin{aligned} \mathsf{Z}_i[a] \oplus \mathsf{Z}_j[b] &= \mathsf{M}_i[a+1] \oplus \mathsf{M}_j[b+1] \\ (\mu_{\delta_{\mathsf{M}_i}} \oplus \mu_{\delta_{\mathsf{M}_j}}) \odot \mathsf{L} \oplus \mathsf{Z}_i[\ell_i - 1] \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_i[\ell_i] \oplus \overline{\mathsf{M}}_j[\ell_j] \end{aligned}$$

Suppose $\delta_{\mathsf{M}_i} \neq \delta_{\mathsf{M}_j}$, i.e. $\mu_{\delta_{\mathsf{M}_i}} \oplus \mu_{\delta_{\mathsf{M}_j}} \neq 0^n$. Using the fact that $\neg\texttt{BadW1}$ holds, we infer that $\mathsf{L} \notin \{\mathsf{Z}_i[a], \mathsf{Z}_j[b], \mathsf{Z}_i[\ell_i - 1], \mathsf{Z}_j[\ell_j - 1]\}$. So, the two equations are linearly independent, whence the rank is 2 in this case. Again, using [9, Lemma A.4], and the fact that there are at most $q^2/2$ choices for i and j, and ℓ^2 choices for a and b, we get

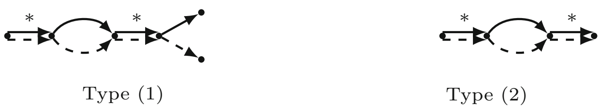
$$\Pr\left(\texttt{BadW2} \wedge \text{Case A} \wedge \delta_{\mathsf{M}_i} \neq \delta_{\mathsf{M}_j}|\texttt{E1}\right) \leq \frac{q^2\ell^2}{2(2^n - q - \sigma + 2)^2}.$$

Now, suppose $\delta_{\mathsf{M}_i} = \delta_{\mathsf{M}_j}$, i.e. $\mu_{\delta_{\mathsf{M}_i}} \oplus \mu_{\delta_{\mathsf{M}_j}} = 0^n$. Then, we can rewrite the system as

$$\begin{aligned} \mathsf{Z}_i[a] \oplus \mathsf{Z}_j[b] &= \mathsf{M}_i[a+1] \oplus \mathsf{M}_j[b+1] \\ \mathsf{Z}_i[\ell_i - 1] \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_i[\ell_i] \oplus \overline{\mathsf{M}}_j[\ell_j] \end{aligned}$$

We can have two types of structure graphs relevant to this case, as illustrated in Fig. 3. For type 1 all variables are distinct. So, the two equations are linearly independent, whence the rank is 2 in this case. Again, using [9, Lemma A.4], we get

$$\Pr\left(\texttt{BadW2} \wedge \text{Case A} \wedge \delta_{\mathsf{M}_i} = \delta_{\mathsf{M}_j} \wedge \text{Type 1}|\texttt{E1}\right) \leq \frac{q^2\ell^2}{2(2^n - q - \sigma + 2)^2}.$$



Type (1)

Type (2)

Fig. 3. Accident-1 manageable graphs for two messages. The solid and dashed lines correspond to edges in $\mathcal{W}_i$ and $\mathcal{W}_j$, respectively. $*$ denotes optional parts in the walk.

For type 2, it is clear that $\mathsf{Z}_j[\ell_j - 1] = \mathsf{Z}_i[\ell_i - 1]$. So, we can assume that the second equation holds trivially, thereby deriving a system in $\mathsf{Z}_i[a]$ and $\mathsf{Z}_j[b]$, with rank 1. Further, a and b are uniquely determined as $\ell_i - p$ and $\ell_j - p$, where p is the longest common suffix of M_i and M_j. So we have

$$\Pr\left(\mathtt{BadW2} \wedge \text{Case A} \wedge \delta_{\mathsf{M}_i} = \delta_{\mathsf{M}_j} \wedge \text{Type 2}|\mathtt{E1}\right) \leq \frac{q^2}{2(2^n - q - \sigma + 1)}.$$

(B) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j)) = 0$: In this case, we only have one equation of the form

$$(\mu_{\delta_{\mathsf{M}_i}} \oplus \mu_{\delta_{\mathsf{M}_j}}) \odot \mathsf{L} \oplus \mathsf{Z}_i[\ell_i - 1] \oplus \mathsf{Z}_j[\ell_j - 1] = \overline{\mathsf{M}}_i[\ell_i] \oplus \overline{\mathsf{M}}_j[\ell_j]$$

If $\delta_{\mathsf{M}_i} \neq \delta_{\mathsf{M}_j}$, we have an equation in three variables, namely L, $\mathsf{Z}_i[\ell_i - 1]$, and $\mathsf{Z}_j[\ell_j - 1]$; and if $\delta_{\mathsf{M}_i} = \delta_{\mathsf{M}_j}$, we have an equation in two variables, namely $\mathsf{Z}_i[\ell_i - 1]$, and $\mathsf{Z}_j[\ell_j - 1]$. In both the cases, the equation can only hold non-trivially, i.e., rank is 1. Using [9, Lemma A.4], we get

$$\Pr\left(\mathtt{BadW2} \wedge \text{Case B}|\mathtt{E1}\right) \leq \frac{q^2}{2(2^n - q - \sigma + 1)}.$$

On combining the three cases, we get

$$\Pr\left(\mathtt{BadW2}|\mathtt{E1}\right) \leq \frac{q^2}{2^n - q - \sigma + 1} + \frac{q^2\ell^2}{(2^n - q - \sigma + 2)^2}. \tag{21}$$

<u>*Bounding* $\Pr\left(\mathtt{BadW3}|\mathtt{E2}\right)$*:*</u> Fix some $i, j, k \in [q]$. Since $\neg\mathtt{unman}$ holds, we must have $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) \leq 2$. Accordingly, we have the following three cases:

(A) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 2$: Suppose (α_1, β_1) and (α_2, β_2) are collision source leading to one of the accident, and (α_3, β_3) and (α_4, β_4) are collision source leading to the other accident. Then, considering $\mathsf{W}_i[a] = \mathsf{W}_j[\ell_j]$, we have the following system of equations

$$\begin{aligned} \mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1] \\ \mathsf{Z}_{\alpha_3}[\beta_3] \oplus \mathsf{Z}_{\alpha_4}[\beta_4] &= \mathsf{M}_{\alpha_3}[\beta_3 + 1] \oplus \mathsf{M}_{\alpha_4}[\beta_4 + 1] \\ \mathsf{Z}_j[a - 1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a] \end{aligned}$$

The first two equations are independent by definition. Further, using $\neg\mathtt{BadW1}$, we can infer that the last equation is also independent of the first two equations. Thus the system has rank 3. There are at most $q^3/6$ choices for (i, j, k), and for each such choice we have 3 choices for $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ and at most ℓ^5 choices for $(\beta_1, \beta_2, \beta_3, \beta_4, a)$. Using [9, Lemma A.4], we have

$$\Pr\left(\mathtt{BadW3} \wedge \text{Case A}|\mathtt{E2}\right) \leq \frac{q^3\ell^5}{2(2^n - q - \sigma + 3)^3}.$$

(B) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i,\mathsf{M}_j,\mathsf{M}_k)) = 1$: Suppose (α_1,β_1) and (α_2,β_2) are collision source leading to the accident. First consider the case $a < \ell_i - 1$ and $b < \ell_k$. In this case, we have the following system of equations

$$\begin{aligned} \mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1+1] \oplus \mathsf{M}_{\alpha_2}[\beta_2+1] \\ \mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j-1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a] \\ \mathsf{Z}_i[a] \oplus \mathsf{Z}_k[b-1] &= \mathsf{M}_i[a+1] \oplus \mathsf{M}_k[b] \end{aligned}$$

The first two equations are clearly independent. Further, since $\mathsf{M}_i \neq \mathsf{M}_k$, the last equation must correspond to a true collision as a consequence of the accident. So, the rank of the above system is 2. Once we fix (i,j,k) and (a,b), we have at most 3 choices for (α_1,α_2), and β_1 and β_2 are uniquely determined as $a+1-p$ and $b-p$, where p is the largest common suffix of $\mathsf{M}_i[1,\ldots,a+1]$ and $\mathsf{M}_k[1,\ldots,b]$. So, we have

$$\Pr\left(\mathtt{BadW3} \wedge \text{Case B} \wedge a < \ell_i - 1 \wedge b < \ell_k | \mathtt{E2}\right) \leq \frac{q^3\ell^2}{2(2^n-q-\sigma+2)^2}.$$

Now, suppose $a = \ell_i - 1$. Then we can simply consider the first two equations

$$\begin{aligned} \mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1+1] \oplus \mathsf{M}_{\alpha_2}[\beta_2+1] \\ \mathsf{Z}_j[\ell_i-2] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j-1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[\ell_i-1] \end{aligned}$$

Clearly, the two equations are independent. We have at most q^3 choices for (i,j,k), 3 choices for (α_1,α_2), and ℓ^2 choices for (β_1,β_2). So we have

$$\Pr\left(\mathtt{BadW3} \wedge \text{Case B} \wedge a = \ell_i - 1 | \mathtt{E2}\right) \leq \frac{q^3\ell^2}{2(2^n-q-\sigma+2)^2}.$$

The case where $a < \ell_i - 1$ and $b = \ell_k$ can be handled similarly by considering the first and the third equations.

(C) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i,\mathsf{M}_j,\mathsf{M}_k)) = 0$: In this case, we know that the three paths, $\mathcal{W}_i$, $\mathcal{W}_j$, and $\mathcal{W}_k$ do not collide. This implies that we must have $a = \ell_i - 1$, or $b = \ell_k$ or both, in order for $\mathsf{W}_i[a+1] = \mathsf{W}_k[b]$ to hold. First, suppose both $a = \ell_i - 1$ and $b = \ell_k$. Then, we have the following system of equations:

$$\begin{aligned} \mathsf{Z}_j[\ell_i-2] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j-1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[\ell_i-2] \\ (\mu_{\delta_{\mathsf{M}_i}} \oplus \mu_{\delta_{\mathsf{M}_k}}) \odot \mathsf{L} \oplus \mathsf{Z}_i[\ell_i-1] \oplus \mathsf{Z}_k[\ell_k-1] &= \overline{\mathsf{M}}_i[\ell_i] \oplus \overline{\mathsf{M}}_k[\ell_k] \end{aligned}$$

Using the properties of μ_{-1} and μ_0, and $\neg\mathtt{BadW1}$, we can conclude that the above system has rank 2. There are at most $q^3/6$ choices for (i,j,k), and at most ℓ^2 choices for (a,b). So, we have

$$\Pr\left(\mathtt{BadW3} \wedge \text{Case C} \wedge a = \ell_i - 1 \wedge b = \ell_k | \mathtt{E2}\right) \leq \frac{q^3\ell^2}{6(2^n-q-\sigma+2)^2}.$$

The remaining two cases are similar. We handle the case $a = \ell_i - 1$ and $b < \ell_k$, and the other case can be handled similarly. We have the following system of equations

$$\begin{aligned} \mathsf{Z}_j[\ell_i - 2] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[\ell_i - 2] \\ \mu_{\delta_{\mathsf{M}_i}} \odot \mathsf{L} \oplus \mathsf{Z}_i[\ell_i - 1] \oplus \mathsf{Z}_k[b - 1] &= \overline{\mathsf{M}}_i[\ell_i] \oplus \mathsf{M}_k[b] \end{aligned}$$

If $\delta_{\mathsf{M}_i} \neq \delta_{\mathsf{M}_j}$, then using the same argument as above, we can conclude that the system has rank 2, and we get

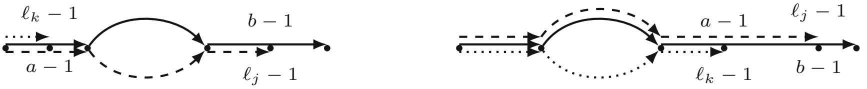
$$\Pr\left(\texttt{BadW3} \wedge \text{Case C} \wedge a = \ell_i - 1 \wedge b < \ell_k \wedge \delta_{\mathsf{M}_i} \neq \delta_{\mathsf{M}_j} | \texttt{E2}\right) \leq \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2}.$$

So, suppose $\delta_{\mathsf{M}_i} = \delta_{\mathsf{M}_j}$. Now, in order for the second equation to be a consequence of the first equation, we must have $\mathsf{Z}_i[\ell_i - 2] = \mathsf{Z}_j[\ell_j - 1]$ and $\mathsf{Z}_i[\ell_i - 1] = \mathsf{Z}_k[b]$. The only we way this happens trivially is if $\mathsf{M}_i[1, \ldots, \ell_i - 1] = \mathsf{M}_j[1, \ldots, \ell_j - 1]$ and $\mathsf{M}_i[1, \ldots, \ell_i - 1] = \mathsf{M}_k[1, \ldots, b]$. But, then we have $b = \ell_i - 1$, and once we fix (i, k) there's a unique choice for j, since $\mathsf{M}_j[1, \ldots, \ell_j - 1] = \mathsf{M}_i[1, \ldots, \ell_i - 1]$ and $\overline{\mathsf{M}}_j[\ell_j] = \overline{\mathsf{M}}_i[\ell_i] \oplus \mathsf{M}_i[\ell_i - 2] \oplus \mathsf{M}_k[b]$. So, we get

$$\Pr\left(\texttt{BadW3} \wedge \text{Case C} \wedge a = \ell_i - 1 \wedge b < \ell_k \wedge \delta_{\mathsf{M}_i} = \delta_{\mathsf{M}_j} | \texttt{E2}\right) \leq \frac{q^2}{2(2^n - q - \sigma + 1)}.$$

By combining all three cases, we have

$$\Pr\left(\texttt{BadW3} | \texttt{E2}\right) \leq \frac{q^3\ell^5}{2(2^n - q - \sigma + 3)^3} + \frac{2q^3\ell^2}{(2^n - q - \sigma + 2)^2} + \frac{q^2}{2(2^n - q - \sigma + 1)}. \tag{22}$$



Type (1) Type (2)

Fig. 4. Manageable graphs for case B.1. The solid, dashed and dotted lines correspond to edges in $\mathcal{W}_i$, $\mathcal{W}_j$, and $\mathcal{W}_k$, respectively.

Bounding $\Pr\left(\texttt{BadW4} | \texttt{E3}\right)$*:* Fix some $i, j, k \in [q]$. The analysis in this case is very similar to the one in case of $\texttt{BadW3} | \texttt{E2}$. So we will skip detailed argumentation whenever possible. Since $\neg\texttt{unman}$ holds, we must have $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) \leq 2$. Accordingly, we have the following three cases:

(A) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 2$: This can be bounded by using exactly the same argument as used in Case A for $\texttt{BadW3} | \texttt{E2}$. So, we have

$$\Pr\left(\texttt{BadW4} \wedge \text{Case A} | \texttt{E3}\right) \leq \frac{q^3\ell^5}{2(2^n - q - \sigma + 3)^3}.$$

(B) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 1$: Suppose (α_1, β_1) and (α_2, β_2) are collision source leading to the accident. Without loss of generality we assume $a < b$. Specifically, $b \le \ell_i - 1$ and $a \le b - 2$ due to $\neg(\mathtt{BadW2} \wedge \mathtt{BadW3})$. First consider the case $b = \ell_i - 1$. In this case, considering $\mathsf{W}_i[b] = \mathsf{W}_k[\ell_k]$, we have the following system of equations

$$\begin{aligned} \mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1] \\ \mathsf{Z}_i[b-1] \oplus \mu_{\delta_{\mathsf{M}_k}} \odot \mathsf{L} \oplus \mathsf{Z}_k[\ell_k - 1] &= \overline{\mathsf{M}}_k[\ell_k] \oplus \mathsf{M}_i[b] \end{aligned}$$

Using a similar argument as used in previous such cases, we establish that the two equations are independent. Now, once we fix (i, j, k), we have exactly one choice for b, at most 3 choices for (α_1, α_2), and ℓ^2 choices for (β_1, β_2). So, we have

$$\Pr\left(\mathtt{BadW4} \wedge \text{Case B} \wedge b = \ell_i - 1 | \mathtt{E3}\right) \le \frac{q^3 \ell^2}{2(2^n - q - \sigma + 2)^2}.$$

Now, suppose $b < \ell_i - 1$. Here we can have two cases:

(B.1) $\mathcal{W}_i$ *is involved in the accident*: Without loss of generality assume that $\alpha_1 = i$ and $\beta_1 \in [\ell_i - 1]$. Then, we have the following system of equations:

$$\begin{aligned} \mathsf{Z}_i[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_i[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1] \\ \mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a] \\ \mathsf{Z}_i[b-1] \oplus \mu_{\delta_{\mathsf{M}_k}} \odot \mathsf{L} \oplus \mathsf{Z}_k[\ell_k - 1] &= \overline{\mathsf{M}}_k[\ell_k] \oplus \mathsf{M}_i[b] \end{aligned}$$

Suppose $\mathsf{Z}_i[\beta_1] = \mathsf{Z}_i[a-1]$. Then, we must have $\beta_1 = a - 1$ as the graph is manageable.In this case, we consider the first two equations. It is easy to see that the two equations are independent, and once we fix i, j, k, there are at most 2 choices for α_2 and ℓ^2 choices for (β_1, β_2), which gives a unique choice for a. So, we have

$$\Pr\left(\mathtt{BadW4} \wedge \text{Case B.1} \wedge \beta_1 = a - 1 | \mathtt{E3}\right) \le \frac{q^3 \ell^2}{2(2^n - q - \sigma + 2)^2}.$$

We get identical bound for the case when $\mathsf{Z}_i[\beta_1] = \mathsf{Z}_i[b-1]$. Suppose $\mathsf{Z}_i[\beta_1] \notin \{\mathsf{Z}_i[a-1], \mathsf{Z}_i[b-1]\}$. Then, using the fact that there is only one accident in the graph and that accident is due to (i, β_1) and (α_2, β_2), we infer that $\mathsf{Z}_{\alpha_2}[\beta_2] \notin \{\mathsf{Z}_i[a-1], \mathsf{Z}_i[b-1]\}$. Now, the only way rank of the above system reduces to 2, is if $\mathsf{Z}_i[a-1] = \mathsf{Z}_k[\ell_k - 1]$ and $\mathsf{Z}_i[b-1] = \mathsf{Z}_j[\ell_j - 1]$ trivially. However, if this happens then a and b are uniquely determined by our choice of $(i, j, k, \beta_1, \alpha_2, \beta_2)$. See Fig. 4 for the two possible structure graphs depending upon the value of α_2. Basically, based on the choice of α_2, $a \in \{\ell_k, \ell_k - \beta_2 + \beta_1\}$. Similarly, $b \in \{\ell_j, \ell_j - \beta_2 + \beta_1\}$. So, using [9, Lemma A.4], we get

$$\Pr\left(\mathtt{BadW4} \wedge \text{Case B.1} \wedge \beta_1 \notin \{a-1, b-1\} | \mathtt{E3}\right) \le \frac{2q^3 \ell^2}{3(2^n - q - \sigma + 2)^2}.$$

(B.2) $\mathcal{W}_i$ *is not involved in the accident*: Without loss of generality assume $\alpha_1 = j$ and $\alpha_2 = k$. Then, we have the following system of equations:

$$\begin{aligned} \mathsf{Z}_j[\beta_1] \oplus \mathsf{Z}_k[\beta_2] &= \mathsf{M}_j[\beta_1 + 1] \oplus \mathsf{M}_k[\beta_2 + 1] \\ \mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a] \\ \mathsf{Z}_i[b-1] \oplus \mu_{\delta_{\mathsf{M}_k}} \odot \mathsf{L} \oplus \mathsf{Z}_k[\ell_k - 1] &= \overline{\mathsf{M}}_k[\ell_k] \oplus \mathsf{M}_i[b] \end{aligned}$$

Since the graph is manageable, $\{\mathsf{Z}_i[a-1], \mathsf{Z}_i[b-1]\} \cap \{\mathsf{Z}_j[\ell_j - 1], \mathsf{Z}_k[\ell_k - 1]\} \neq \emptyset$. Suppose $\{\mathsf{Z}_i[a-1], \mathsf{Z}_i[b-1]\} = \{\mathsf{Z}_j[\ell_j - 1], \mathsf{Z}_k[\ell_k - 1]\}$. Without loss of generality, assume $\mathsf{Z}_i[a-1] = \mathsf{Z}_k[\ell_k - 1]$ and $\mathsf{Z}_i[b-1] = \mathsf{Z}_j[\ell_j - 1]$. This can only happen if the resulting graph is of Type 2 form in Fig. 4, which clearly shows that we have unique choices for a and b when we fix the other indices. Now, suppose $|\{\mathsf{Z}_i[a-1], \mathsf{Z}_i[b-1]\} \cap \{\mathsf{Z}_j[\ell_j - 1], \mathsf{Z}_k[\ell_k - 1]\}| = 1$. Then, we must have $\mathsf{Z}_i[a-1] \in \{\mathsf{Z}_j[\beta_1], \mathsf{Z}_k[\beta_2]\}$ since $a < b$. Without loss of generality we assume that $\mathsf{Z}_i[a-1] = \mathsf{Z}_k[\beta_2]$ and $\mathsf{Z}_i[b-1] = \mathsf{Z}_j[\ell_j - 1]$. Using similar argument as before, we conclude that a and b are fixed once we fix all other indices. So using [9, Lemma A.4], we get

$$\Pr\left(\mathtt{BadW4} \wedge \text{Case B.2}|\mathtt{E3}\right) \leq \frac{2q^3\ell^2}{3(2^n - q - \sigma + 2)^2}.$$

(C) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 0$: In this case, we know that the three paths, $\mathcal{W}_i$, $\mathcal{W}_j$, and $\mathcal{W}_k$ do not collide. We have the following system of equations:

$$\begin{aligned} \mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a] \\ \mathsf{Z}_i[b-1] \oplus \mu_{\delta_{\mathsf{M}_k}}) \odot \mathsf{L} \oplus \mathsf{Z}_k[\ell_k - 1] &= \overline{\mathsf{M}}_i[\ell_k] \oplus \mathsf{M}_i[b] \end{aligned}$$

Using a similar analysis as in case C of $\mathtt{BadW3}|\mathtt{E2}$, we get

$$\Pr\left(\mathtt{BadW4} \wedge \text{Case C}|\mathtt{E3}\right) \leq \frac{q^3\ell^2}{6(2^n - q - \sigma + 2)^2} + \frac{q^2}{2(2^n - q - \sigma + 1)}.$$

By combining all three cases, we have

$$\Pr\left(\mathtt{BadW4}|\mathtt{E3}\right) \leq \frac{q^3\ell^5}{2(2^n - q - \sigma + 3)^3} + \frac{3q^3\ell^2}{(2^n - q - \sigma + 2)^2} + \frac{q^2}{2(2^n - q - \sigma + 1)}. \tag{23}$$

<u>*Bounding* $\Pr\left(\mathtt{BadW5}|\mathtt{E4}\right)$</u>*:* Fix some $i, j, k \in [q]$. The analysis in this case is again similar to the analysis of $\mathtt{BadW3}|\mathtt{E2}$ and $\mathtt{BadW4}|\mathtt{E3}$. We have the following three cases:

(A) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 2$: This can be bounded by using exactly the same argument as used in Case A for $\mathtt{BadW3}|\mathtt{E2}$. So, we have

$$\Pr\left(\mathtt{BadW5} \wedge \text{Case A}|\mathtt{E4}\right) \leq \frac{q^3\ell^5}{2(2^n - q - \sigma + 3)^3}.$$

(B) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 1$: Suppose (α_1, β_1) and (α_2, β_2) are collision source leading to the accident. In this case, we have the following system of equations

$$\begin{aligned}\mathsf{Z}_{\alpha_1}[\beta_1] \oplus \mathsf{Z}_{\alpha_2}[\beta_2] &= \mathsf{M}_{\alpha_1}[\beta_1 + 1] \oplus \mathsf{M}_{\alpha_2}[\beta_2 + 1]\\ \mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a]\\ \mathsf{Z}_j[b-1] \oplus \mu_{\delta_{\mathsf{M}_k}} \odot \mathsf{L} \oplus \mathsf{Z}_k[\ell_k - 1] &= \overline{\mathsf{M}}_k[\ell_k] \oplus \mathsf{M}_j[b]\end{aligned}$$

We can have two sub-cases:

(B.1) Suppose the third equation is simply a consequence of the second equation. Then, we must have $\delta_{\mathsf{M}_i} = \delta_{\mathsf{M}_j}$ and $\mathsf{Z}_i[a-1] = \mathsf{Z}_j[b-1]$ and $\mathsf{Z}_j[\ell_j - 1] = \mathsf{Z}_k[\ell_k - 1]$ must hold trivially, since the graph is manageable. We claim that $a = b = \mathsf{Prefix}(\mathsf{M}_i[1], \mathsf{M}_j[1]) + 1$. If not, then $\mathsf{M}_i[\ell_i] = \mathsf{M}_j[\ell_j]$ which in conjunction with $\mathsf{Z}_j[\ell_j - 1] = \mathsf{Z}_k[\ell_k - 1]$ implies that $\mathsf{W}_i[\ell_i] = \mathsf{W}_j[\ell_j]$ which contradicts `BadW2`. So, using [9, Lemma A.4], we get

$$\Pr\left(\mathtt{BadW5} \wedge \text{Case B.1} | \mathtt{E4}\right) \leq \frac{q^3 \ell^2}{2(2^n - q - \sigma + 2)^2}.$$

(B.2) The second and third equation are independent. Considering the subsystem consisting of these two equations, and using [9, Lemma A.4], we get

$$\Pr\left(\mathtt{BadW5} \wedge \text{Case B.2} | \mathtt{E4}\right) \leq \frac{q^3 \ell^2}{6(2^n - q - \sigma + 2)^2}.$$

(C) $\mathsf{Acc}(\mathcal{G}_{\mathsf{P}}(\mathsf{M}_i, \mathsf{M}_j, \mathsf{M}_k)) = 0$: We have the following system of equations:

$$\begin{aligned}\mathsf{Z}_i[a-1] \oplus \mu_{\delta_{\mathsf{M}_j}} \odot \mathsf{L} \oplus \mathsf{Z}_j[\ell_j - 1] &= \overline{\mathsf{M}}_j[\ell_j] \oplus \mathsf{M}_i[a]\\ \mathsf{Z}_i[b-1] \oplus \mu_{\delta_{\mathsf{M}_k}} \odot \mathsf{L} \oplus \mathsf{Z}_k[\ell_k - 1] &= \overline{\mathsf{M}}_i[\ell_k] \oplus \mathsf{M}_i[b]\end{aligned}$$

Let r denote the rank of the above system. Using a similar analysis as in case B.1 above, we conclude that $a = b = \mathsf{Prefix}(\mathsf{M}_i[1], \mathsf{M}_j[1]) + 1$ if $r = 1$. Using [9, Lemma A.4], we get

$$\begin{aligned}\Pr\left(\mathtt{BadW5} \wedge \text{Case C} \wedge r = 1 | \mathtt{E4}\right) &\leq \frac{q^2}{2(2^n - q - \sigma + 1)}.\\ \Pr\left(\mathtt{BadW5} \wedge \text{Case C} \wedge r = 2 | \mathtt{E4}\right) &\leq \frac{q^3 \ell^2}{6(2^n - q - \sigma + 2)^2}.\end{aligned}$$

By combining all three cases, we have

$$\Pr\left(\mathtt{BadW5} | \mathtt{E4}\right) \leq \frac{q^3 \ell^5}{2(2^n - q - \sigma + 3)^3} + \frac{5 q^3 \ell^2}{6(2^n - q - \sigma + 2)^2} + \frac{q^2}{2(2^n - q - \sigma + 1)}. \tag{24}$$

Further, from Eqs. (19)–(24), we have

$$\Pr(\texttt{BadW}|\texttt{TU}) \leq \frac{2\sigma}{2^n} + \frac{5q^2}{2(2^n - q - \sigma + 1)} + \frac{7q^3\ell^2}{(2^n - q - \sigma + 2)^2} + \frac{3q^3\ell^5}{2(2^n - q - \sigma + 3)^3}. \tag{25}$$

Bounding $\Pr(\texttt{BadX}|\texttt{TUW})$: In the full version [9, Appendix B] of this paper, we show that

$$\begin{aligned}\Pr(\texttt{BadX}|\texttt{TUW}) \leq \frac{2\sigma}{2^n} &+ \frac{10q^2}{2^n - q - \sigma + 1} + \frac{15q^3\ell^2 + q^2\ell^3}{(2^n - q - \sigma + 2)^2} \\ &+ \frac{12q^3\ell^6 + 6q^4\ell^3}{(2^n - q - \sigma + 3)^3} + \frac{8q^4\ell^6}{(2^n - q - \sigma + 4)^4}\end{aligned} \tag{26}$$

Combining Eqs. (18), (25), and (26), we have

$$\begin{aligned}\Pr(\mathsf{V}_0 \in \mathcal{V}_{\mathsf{bad}}) \leq \frac{4\sigma}{2^n} &+ \frac{16q^2 + q\ell^2}{2^n} + \frac{8q^2\ell^4 + 32q^3\ell^2 + 2q^2\ell^3}{2^{2n}} \\ &+ \frac{3q^3\ell^5 + 143q^3\ell^6 + 11q^4\ell^3}{2^{3n}} + \frac{17q^4\ell^6 + 5462q^4\ell^8}{2^{4n}}.\end{aligned} \tag{27}$$

6 Conclusion

In this paper we proved that OMAC, XCBC and TMAC are secure up to $q \leq 2^{n/2}$ queries, while the message length $\ell \leq 2^{n/4}$. As a consequence, we have proved that OMAC – a single-keyed CBC-MAC variant – achieves the same security level as some of the more elaborate CBC-MAC variants like EMAC and ECBC. This, in combination with the existing results [15,16], shows that the security is tight up to $\ell \leq 2^{n/4}$ for all CBC-MAC variants except for the original CBC-MAC. It could be an interesting future problem to extend our analysis and derive similar bounds for CBC-MAC over prefix-free message space. In order to prove our claims, we employed reset-sampling method by Chattopadhyay et al. [8], which seems to be a promising tool in reducing the length-dependency in single-keyed iterated constructions. Indeed, we believe that this tool might even be useful in obtaining better security bounds for single-keyed variants of many beyond-the-birthday-bound constructions.

References

1. Bellare, M., Goldreich, O., Mityagin, A.: The power of verification queries in message authentication and authenticated encryption. IACR Cryptol. ePrint Arch. **2004**, 309 (2004)
2. Bellare, M., Kilian, J., Rogaway, P.: The security of cipher block chaining. In: Proceedings of Advances in Cryptology - CRYPTO 1994, pp. 341–358 (1994)

3. Bellare, M., Pietrzak, K., Rogaway, P.: Improved security analyses for CBC macs. In: Proceedings of Advances in Cryptology - CRYPTO 2005, pp. 527–545 (2005)
4. Berendschot, A., et al.: Final Report of RACE Integrity Primitives, vol. 1007, LNCS, Springer-Verlag, Berlin (1995). https://doi.org/10.1007/3-540-60640-8
5. Black, J., Rogaway, P.: CBC macs for arbitrary-length messages: the three-key constructions. In: Proceedings of Advances in Cryptology - CRYPTO 2000, pp. 197–215 (2000)
6. Black, J., Rogaway, P.: A block-cipher mode of operation for parallelizable message authentication. In: Proceedings of Advances in Cryptology - EUROCRYPT 2002, pp. 384–397 (2002)
7. Chakraborty, B., Chattopadhyay, S., Jha, A., Nandi, M.: On length independent security bounds for the PMAC family. IACR Trans. Symmet. Cryptol. **2021**(2), 423–445 (2021)
8. Chattopadhyay, S., Jha, A., Nandi, M.: Fine-tuning the ISO/IEC Standard Lightmac. In: Proceedings of Advances in Cryptology - ASIACRYPT 2021, pp. 490–519 (2021)
9. Chattopadhyay, S., Jha, A., Nandi, M.: Towards tight security bounds for OMAC, XCBC and TMAC. IACR Cryptol. ePrint Arch. **2022**, 1234 (2022)
10. Dworkin, M.: Recommendation for block cipher modes of operation: the CMAC mode for authentication. NIST Special Publication 800–38b, National Institute of Standards and Technology, U. S. Department of Commerce (2005)
11. Ehrsam, W.F., Meyer, C.H.W., Smith, J.L., Tuchman, W.L.: Message verification and transmission error detection by block chaining. Patent 4,074,066, USPTO (1976)
12. Iwata, T., Kurosawa, K.: OMAC: One-Key CBC MAC. In: Fast Software Encryption - FSE 2003, Revised Papers, pp. 129–153 (2003)
13. Iwata, T., Kurosawa, K.: Stronger Security Bounds for OMAC, TMAC, and XCBC. In: Proceedings of Progress in Cryptology - INDOCRYPT 2003, pp. 402–415 (2003)
14. Jha, A., Mandal, A., Nandi, M.: On the exact security of message authentication using pseudorandom functions. IACR Trans. Symmetric Cryptol. **2017**(1), 427–448 (2017)
15. Jha, A., Nandi, M.: Revisiting structure graphs: applications to CBC-MAC and EMAC. J. Math. Cryptol. **10**(3–4), 157–180 (2016)
16. Jha, A., Nandi, M.: Revisiting structure graphs: applications to CBC-MAC and EMAC. IACR Cryptol. ePrint Arch. **2016**, 161 (2016)
17. Kurosawa, K., Iwata, T.: TMAC: two-key CBC MAC. In: Proceedings of Topics in Cryptology - CT-RSA 2003, pp. 33–49 (2003)
18. Luykx, A., Preneel, B., Tischhauser, E., Yasuda, K.: A MAC mode for lightweight block ciphers. In: Fast Software Encryption - FSE 2016, Revised Selected Papers, pp. 43–59 (2016)
19. Minematsu, K., Matsushima, T.: New bounds for PMAC, TMAC, and XCBC. In: Fast Software Encryption - FSE 2007, Revised Selected Papers, pp. 434–451 (2007)
20. Nandi, M.: Fast and secure CBC-type MAC algorithms. In: Fast Software Encryption - FSE 2009, Revised Selected Papers, pp. 375–393 (2009)
21. Nandi, M.: Improved security analysis for OMAC as a pseudorandom function. J. Math. Cryptol. **3**(2), 133–148 (2009)
22. Nandi, M., Mandal, A.: Improved security analysis of PMAC. J. Math. Cryptol. **2**(2), 149–162 (2008)

23. Patarin, J.: Etude des Générateurs de Permutations Pseudo-aléatoires Basés sur le Schéma du DES. Ph.D. thesis, Université de Paris (1991)
24. Patarin, J.: The "coefficients H" technique. In: Selected Areas in Cryptography - SAC 2008. Revised Selected Papers, pp. 328–345 (2008)
25. Pietrzak, K.: A tight bound for EMAC. In: Proceedings of Automata, Languages and Programming - ICALP 2006, Part II, pp. 168–179 (2006)