On Beyond-Birthday-Bound Security: Revisiting the Development of ISO/IEC 9797-1 MACs

Yaobin Shen[1] and Lei Wang[1,2,3*]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
yb_shen@sjtu.edu.cn,wanglei_hb@sjtu.edu.cn
[2] University College Oxford Blockchain Research Centre, Oxford, United Kingdom
[3] Oxford-Hainan Blockchain Research Institute, Hainan, China

Abstract. ISO/IEC 9797-1 is an international standard for block-cipher-based Message Authentication Code (MAC). The current version ISO/IEC 9797-1:2011 specifies six single-pass CBC-like MAC structures that are capped at the birthday bound security. For a higher security that is beyond-birthday bound, it recommends to use the concatenation combiner of two single-pass MACs. In this paper, we reveal the invalidity of the suggestion, by presenting a birthday bound forgery attack on the concatenation combiner, which is essentially based on Joux's multi-collision. Notably, our new forgery attack for the concatenation of two MAC Algorithm 1 with padding scheme 2 only requires 3 queries. Moreover, we look for patches by revisiting the development of ISO/IEC 9797-1 with respect to the beyond-birthday bound security. More specifically, we evaluate the XOR combiner of single-pass CBC-like MACs, which was used in previous version of ISO/IEC 9797-1.

Keywords: ISO/IEC 9797-1 · Beyond Birthday Bound Security · XOR Combiner

1 Introduction

A Message Authentication Code (MAC) is a fundamental symmetric-key primitive to provide integrity and authenticity of messages between two parties. A MAC is usually proven to be Pseudo-Random-Function (PRF), which in turn implies its unforgeability security. The advantage of PRF-security is often measured by n the block size, q the total number of queries, ℓ the length of the longest message and σ the total number of blocks of all queries. There are several approaches to build a MAC, among which a popular way is to iterate a block cipher [BKR00]. A block-cipher-based MAC is typically proven or analyzed under the assumption that underlying block ciphers are Pseudo-Random Permutations (PRP). Most block cipher-based MACs achieve the so-called birthday security, i.e., against up to $O(q^2/2^n)$ adversarial queries.

It is not always sufficient for block-cipher-based MACs with just a birthday-bound security. An example is that a lightweight block cipher such as PRESENT [BKL+07] and PRINCE [BCG+12] has a short block size, e.g. $n = 64$. In such case, the birthday bound becomes 2^{32} and is vulnerable in certain practical applications. For instance, Bhargavan and Leurent [BL16] have demonstrated two practical attacks called Sweet32 that exploit collision on short block ciphers. Hence, performing MACs with beyond birthday security in practical devices is of great importance.

ISO/IEC 9797-1 is an international standard that specifies MACs based on a block cipher. ISO/IEC 9797-1:2011 [ISO11] provides six different mechanisms of CBC MACs,

---

Table 1: Comparison of XMAC1, XMAC5 and other CBC-type MACs with beyond birthday security. XMAC1 resp. XMAC5 denotes the XORing of two MAC Algorithm 1 resp. 5 in ISO/IEC 9797-1:2011. n denotes the input size of the blockcipher, q denotes the total number of queries, ℓ denotes the maximum number of blocks among these q queries, σ denotes the total number of blocks of these q queries. The security bounds of SUM-ECBC, XMAC1 with pad3, XMAC5 with pad4 are conditioned on $\ell \leq 2^{n/3}$.

Algorithm	#keys	Security	Ref.
SUM-ECBC	4	$O(q^3\ell^3/2^{2n})$	[Yas10]
3kf9	3	$O(q^3\ell^3/2^{2n} + q\ell/2^n)$	[ZWSW12]
XMAC1 with pad3	2	$O(\sigma q^2\ell/2^{2n})$	Sect. 5
XMAC1 with pad2	2	$O(\sigma^2/2^n)$	App. A
XMAC5 with pad4	2	$O(\sigma q^2\ell/2^{2n})$	Sect. 6

called MAC Algorithm 1-6. Each MAC is defined by specifying the final iteration and output transformation. These MACs have been widespread implemented in practical devices and thus are of great importance. Since each of these MACs uses single CBC-MAC pass, they all suffer from birthday forgery attacks as explained in [PvO95, PvO99]. It is suggested to improve the security level by concatenating outputs of two MACs in Annex C, C.2 Rationale, ISO/IEC 9797-1:2011:

> if a MAC algorithm with a higher security level is needed, it is recommended to perform two MAC calculations with independent keys and concatenate the results (rather than XORing them).

Our contributions. Firstly, we reveal that the suggestion is unfortunately wrong. More precisely, we present a forgery attack with a complexity of birthday bound on the concatenation combiner of two CBC-like MACs. More surprisingly, our forgery attack for the concatenation of two MAC Algorithm 1 with padding scheme 2 only requires 3 queries. These attacks well demonstrate that the recommended combiner in ISO/IEC 9797-1: 2011 [ISO11] provides a security of at most birthday bound, and cannot amplify the security of underlying single-pass CBC-like MAC for a higher security bound.

Secondly, for finding a patch, we revisit the development of this standard. *Surprisingly* the last version ISO/IEC 9797-1: 1999 [ISO99] suggests XORing two single-pass MACs. More precisely, it recommends two MACs claiming beyond birthday bound security, which are named as $\mathtt{MAC_5}$ and $\mathtt{MAC_6}$ in the standard document. Moreover, each MAC may use one of three padding algorithms denoted as $\mathtt{Pad_1}$, $\mathtt{Pad_2}$ and $\mathtt{Pad_3}$. Hence there are in total 6 dedicated MACs in ISO/IEC 9797-1:1999 claiming beyond-birthday bound security. Joux et al. published a birthday-bound forgery attack on $\mathtt{MAC_5}$ with $\mathtt{Pad_1}$, which is simply XORing two plain CBC MACs. This is the only known birthday-bound attack on XOR combiner of CBC-like MACs back then [JPS03], which might be the main reason that ISO/IEC 97971-1:2011 explicitly object to XORing two single-pass CBC-like MACs for beyond-birthday bound security. On the other hand, Yasuda proved that $\mathtt{MAC_6}$ with any of the three padding indeed achieves beyond-birthday-bound security [Yas10]. Thus, the security of $\mathtt{MAC_5}$ with $\mathtt{Pad_2}$ or with $\mathtt{Pad_3}$ leaves unsolved, which is highlighted by Rogaway in his report for CRYPTOREC.

We revisit the impact of XORing of two MACs on ISO/IEC 9797-1:2011 and show this operation can lift up the security level. We give the first two provable-security bounds for XORing of two MAC Algorithm 1 denoted as XMAC1 (i.e., $\mathtt{MAC_5}$ in ISO/IEC 9797-1:1999) in ISO/IEC 9797-1:2011 with either padding scheme 2 or 3. We prove that XMAC1 with padding scheme 3 is secure beyond birthday bound with $O(\sigma q^2\ell/2^{2n})$. Note that our

result implies that this is the first CBC-type MAC that provably goes beyond birthday barrier with only two secret keys. When instantiated with padding scheme 2, we prove that XMAC1 is secure with birthday bound $O(\sigma^2/2^n)$. Illustrated with Joux et al.'s attack, this bound is tight up to a constant factor.

Moreover, compared with previous version, MAC Algorithm 5 in ISO/IEC 9797-1:2011 is a new introduced single-pass CBC-like MAC, often referred to as CMAC. We also prove that XORing of two MAC Algorithm 5 with padding scheme 4 denoted as XMAC5 is secure with a beyond birthday bound $O(\sigma q^2 \ell/2^{2n})$.

Organization.   In Sect. 2, we give essential notations and definitions. We give birthday forgery attacks on concatenations of two MACs of ISO/IEC 9797-1:2011 in Sect. 3. We present our main results on XMAC1, XMAC5 in Sect. 4. Then we give proofs of XMAC1 with padding scheme 3, XMAC1 with padding scheme 2, and XMAC5 with padding scheme 4 in Sect. 5, Appendix A, Sect. 6 respectively. Finally we conclude this paper in Sect. 7.

2   Preliminaries

2.1   Notation

If $\mathcal{X}$ is a set, then $X \xleftarrow{\$} \mathcal{X}$ denotes the operation of drawing X from $\mathcal{X}$ uniformly at random. $\{0,1\}^*$ denotes all bit strings including the empty string. The bit length of a string X is written by $|X|$. Concatenation of strings X and Y is written as either $X\|Y$ or simply XY. We denote $X \oplus Y$ the bitwise exclusive-or of two equal-length strings. For a string $X \in \{0,1\}^{n\ell}$ with $\ell \geq 1$, we divide X into n-bit blocks as $X = X[1]\|\ldots\|X[\ell]$ where $|X[1]| = \cdots = |X[\ell]| = n$. If ℓ is a non-negative integer such that $\ell < 2^n$, we write $\mathsf{bin}_n(\ell)$ for the n-bit binary representation of ℓ.

We write $\mathrm{Perm}(n)$ for the set of all permutations over $\{0,1\}^n$, and $\mathrm{Rand}(n)$ for the set of all functions mapping $\{0,1\}^*$ to $\{0,1\}^n$. We often perform lazy sampling for specifying a random permutation $P \xleftarrow{\$} \mathrm{Perm}(n)$. We denote $\mathrm{Dom}(P)$ and $\mathrm{Ran}(P)$ the sets of already-defined domain points and range points of P respectively, and $\overline{\mathrm{Dom}(P)}$ and $\overline{\mathrm{Ran}(P)}$ for the complementary sets. A block cipher E is a family of permutations $\{E_K : K \in \mathcal{K}\}$, where $E_K(\cdot) = E(K, \cdot)$ is a permutation over $\{0,1\}^n$ specified by a key K. $\mathcal{K}$ is the key space and n is the block length.

A MAC is an algorithm that takes two inputs a key K and a message M then outputs a fixed-length tag T. K, M and T are all binary strings. The CBC MAC is built from cipher block chaining some underlying block cipher. Let $M_i = M_i[1]\|M_i[2]\|\cdots\|M_i[m_i]$ be a message, where $|M_i[1]| = |M_i[2]| = \cdots = |M_i[m_i]| = n$ and m_i is the block length. Then $\mathrm{CBC}[E_K](M_i)$, the CBC MAC of M_i, is defined as $y^i_{m_i}$, where

$$y^i_j = E_K(M_i[j] \oplus y^i_{j-1})$$

for $j = 1, \ldots, m_i$ and $y^i_0 = 0^n$.

There are total four padding schemes specified in ISO/IEC 9797-1:2011. As padding scheme 1 allows a trivial forgery, we only consider padding scheme 2, 3 and 4 in this paper. Note that MAC Algorithm 5 shall only be used with padding scheme 4 as mandated by the standard.

pad2 the message M is always right-padded with a single '1' bit then right-padded with i bits '0' where i is the least non-negative integer such that $|M| + i + 1$ is a positive multiple of n.

pad3 the message M is mapped to $\mathsf{bin}_n(|M|)\|M0^i$ where i is the least non-negative integer such that $|M| + i$ is a positive multiple of n.

pad4 if the message has length that is positive multiple of n, then no padding shall be applied. Otherwise, the message shall be right-padded with a single '1' bit then right-padded with i bits '0' where i is the least non-negative integer such that $|M| + i + 1$ is a positive multiple of n.

After padded via padding scheme 3, the messages list would become prefix-free, meaning that for any two messages M_i and M_j, M_i is not a prefix of M_j. We simply denote **pad2**(M), **pad3**(M) and **pad4**(M) the operations of mapping M to a sequence of n-bit blocks with padding scheme 2, 3 and 4 respectively.

In MAC Algorithm 5, there are two masking values L_1 and L_2 that are computed by the operation multx(L). For an n-bit string L, multx(L) is defined as $L \cdot x$, where L is treated as an element in the Galois field $\mathrm{GF}(2^n)$ and is multiplied by the monomial x in $\mathrm{GF}(2^n)$. It can be processed as follows:

$$\mathrm{multx}(L) = \begin{cases} L \ll 1, & \text{if } \mathrm{msb}(L) = 0 \\ (L \ll 1) \oplus \mathrm{const} & \text{if } \mathrm{msb}(L) = 1 \end{cases}$$

where $\mathrm{msb}(L)$ denotes the most significant bit of L, $\ll$ denotes the one-bit left shift operation, and const is some constant specified in the standard ISO/IEC 9797-1:2011. Then $L_1 = \mathrm{multx}(E_K(0^n))$ and $L_2 = \mathrm{multx}(L_1)$.

2.2 Security Notions

An adversary $\mathcal{A}$ is an algorithm that always outputs a bit. We write $\mathcal{A}^{O(\cdot)} \Rightarrow 1$ to denote the event that $\mathcal{A}$ outputs 1 after interacting with oracle $O(\cdot)$. We focus on the information-theoretic setting, namely, all keyed block ciphers are replaced with random permutations. Throughout this paper, an adversary $\mathcal{A}$ is allowed to unbounded computational power and assumed to be deterministic without loss of generality. Its complexity is measured by the number of queries, the maximum block length of messages, the total number of blocks of messages. Recalling that any pseudo-random function (PRF) is a secure MAC [BKR00], our goal is to prove $F[P]$ is a secure PRF, where $F[P]$ is an interested function based on random permutations. We say that $F[P]$ is a secure PRF if it is indistinguishable from a random function $\mathcal{R} \xleftarrow{\$} \mathrm{Rand}(n)$. Formally, we define

$$\mathbf{Adv}_{F[P]}^{\mathrm{prf}}(\mathcal{A}) \overset{\mathrm{def}}{=} \Pr[P \xleftarrow{\$} \mathrm{Perm}(n) : \mathcal{A}^{F[P](\cdot)} \Rightarrow 1] - \Pr[\mathcal{R} \xleftarrow{\$} \mathrm{Rand}(n) : \mathcal{A}^{\mathcal{R}(\cdot)} \Rightarrow 1].$$

Note that the probabilities are taken over $P, \mathcal{R}$, and $\mathcal{A}'s$ coins.

3 Forgery Attacks on The Concatenation of Two MACs in ISO/IEC 9797-1:2011

In this section, we present forgery attacks on the concatenation of the MACs in the standard ISO/IEC 9797-1:2011, which are depicted in Fig. 1. We only present the attack for the concatenation of a MAC Algorithm with two independent keys here and the attacks for the concatenation of two different MAC Algorithms is the same. Significantly, the forgery attack for the concatenation of two MAC Algorithm 1 only requires 3 queries with padding scheme 2, while forgery attacks for other MACs need a complexity of birthday bound. We detail our attacks as follows. We denote MAC$i_K(M)$ the MAC for a message M computed using the MAC Algorithm i specified in ISO/IEC 9797-1:2011 for $1 \leq i \leq 6$. We denote MAC$i_{K_1}(M) \| \mathrm{MAC}i_{K_2}(M)$ the concatenation of two MAC Algorithms i with independent keys K_1, K_2. Since padding scheme 1 allows trivial forgery attack, we only
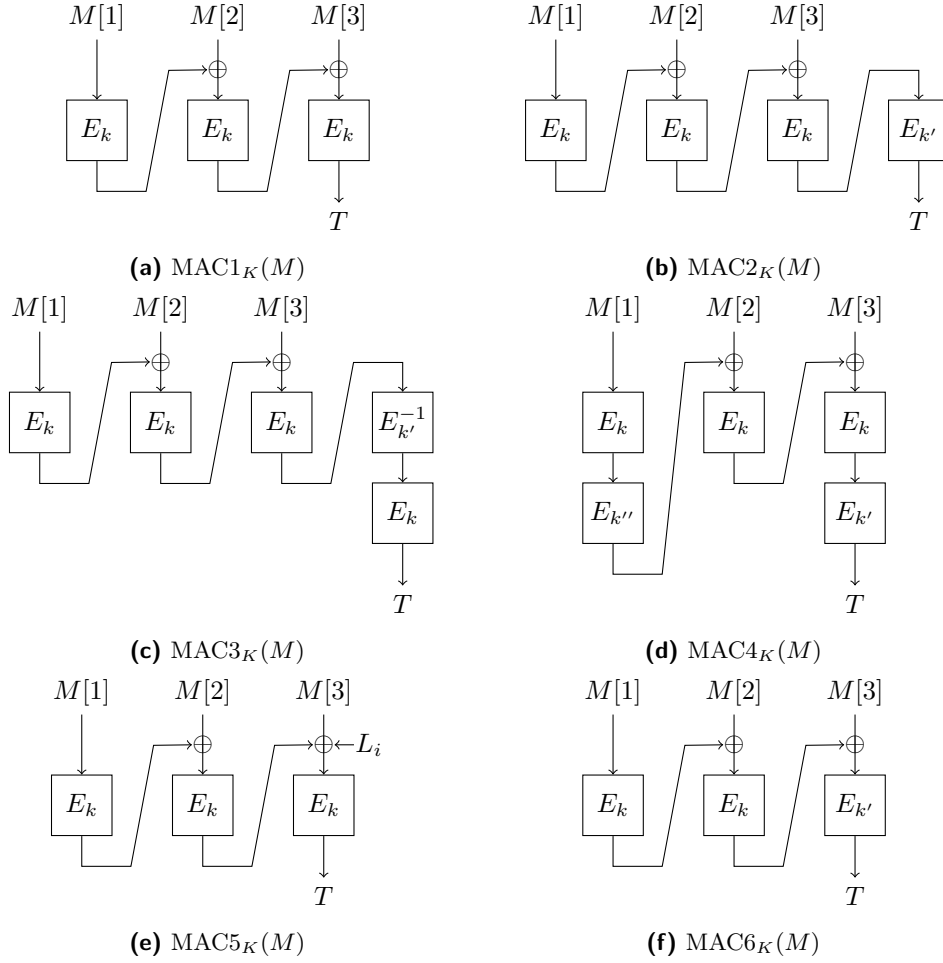
Figure 1: Illustration of the six ISO/IEC MACs. The message M has been padded with specific padding scheme. For $\text{MAC1}_K(M)$ and $\text{MAC5}_K(M)$, the underlying key is $K = k$. For $\text{MAC2}_K(M)$, MAC3 and $\text{MAC6}_K(M)$, the underlying key is $K = (k, k')$. For $\text{MAC4}_K(M)$, the underlying key is $K = (k, k', k'')$. For $\text{MAC5}_K(M)$, if the message M before padded has length (in bits) that is positive multiple of n, then $L_i = L_1$, otherwise $L_i = L_2$.

consider padding scheme 2, 3 and 4 here. In addition, MAC Algorithm 5 shall only be used with padding scheme 4 as mandated by the standard.

At first, we consider the attack when messages are padded with scheme 2. For $\text{MAC1}_{K_1}(M) \| \text{MAC1}_{K_2}(M)$, i.e., the concatenation of two MAC Algorithm 1, we first query a single block message M_1 and obtain $\text{MAC1}_{K_1}(M_1) \| \text{MAC1}_{K_2}(M_1)$. Then we query two strings $M_1 \oplus \text{MAC1}_{K_1}(M_1)$ and $M_1 \oplus \text{MAC1}_{K_2}(M_1)$, and receive $\text{MAC1}_{K_2}(M_1 \oplus \text{MAC1}_{K_1}(M_1))$ (the right half of concatenation) and $\text{MAC1}_{K_1}(M_1 \oplus \text{MAC1}_{K_2}(M_1))$ (the left half of concatenation) respectively. In this stage, we can forge a MAC of the message $M_1 \| M_1 \oplus \text{MAC1}_{K_1}(M_1) \oplus \text{MAC1}_{K_2}(M_1)$ with probability 1 without querying this message since

$$\text{MAC1}_{K_1}(M_1 \| M_1 \oplus \text{MAC1}_{K_1}(M_1) \oplus \text{MAC1}_{K_2}(M_1)) = \text{MAC1}_{K_1}(M_1 \oplus \text{MAC1}_{K_2}(M_1))$$

and

$$\mathrm{MAC1}_{K_2}(M_1 \| M_1 \oplus \mathrm{MAC1}_{K_1}(M_1) \oplus \mathrm{MAC1}_{K_2}(M_1)) = \mathrm{MAC1}_{K_2}(M_1 \oplus \mathrm{MAC1}_{K_1}(M_1)).$$

This attack only requires 3 queries.

For $i \in \{2, 3, 4, 6\}$, we adopt the idea of multicollisions attack in iterated hash functions proposed by Joux [Jou04]. For any $i \in \{2, 3, 4, 6\}$, we first focus on collisions of the left half of concatenation, i.e., $\mathrm{MAC}i_{K_1}(M)$. Because two blocks are needed to produce a collision in CBC-like MACs, we first search two two-block messages $a_{1,1} \| r_{1,1}$ and $a_{2,1} \| r_{2,1}$ such that $\mathrm{MAC}i_{K_1}(a_{1,1} \| r_{1,1}) = \mathrm{MAC}i_{K_1}(a_{2,1} \| r_{2,1})$. This requires about $2^{n/2}$ MAC computations due to birthday paradox. Fixing $a_{1,1} \| r_{1,1}$ and $a_{2,1} \| r_{2,1}$, we then search $a_{1,2} \| r_{1,2}$ and $a_{2,2} \| r_{2,2}$ such that

$$\mathrm{MAC}i_{K_1}(a_{1,1} \| r_{1,1} \| a_{1,2} \| r_{1,2}) = \mathrm{MAC}i_{K_1}(a_{2,1} \| r_{2,1} \| a_{2,2} \| r_{2,2}).$$

This also requires about $2^{n/2}$ MAC computations. We do this until we find two $2t$-block $(t \geq n/2)$ messages $a_{1,1} \| r_{1,1} \| \ldots \| a_{1,t} \| r_{1,t}$ and $a_{2,1} \| r_{2,1} \| \ldots \| a_{2,t} \| r_{2,t}$ such that

$$\mathrm{MAC}i_{K_1}(a_{1,1} \| r_{1,1} \| \ldots \| a_{1,t} \| r_{1,t}) = \mathrm{MAC}i_{K_1}(a_{2,1} \| r_{2,1} \| \ldots \| a_{2,t} \| r_{2,t}).$$

This yields 2^t different messages $a_{i_1,1} \| r_{i_1,1} \| \ldots \| a_{i_t,t} \| r_{i_t,t}$ for $i_1, \ldots, i_t \in \{1, 2\}$ with the same MAC value on the left half of concatenation. Assume $t \geq n/2$, then with high probability there exists a collision among these 2^t elements such that the MAC value on the right half are also equal. Assume the collided messages are M_1 and M_2. Then the two MAC values for $M_1 \| A$ and $M_2 \| A$ are also a collision for any n-bit block A. This attack requires in total about $(1 + n) \cdot 2^{n/2}$ MAC computations.

Secondly, we consider messages after padded with scheme 3 for $i \in \{1, 2, 3, 4, 6\}$. For any $i \in \{1, 2, 3, 4, 6\}$, we first focus on collisions of the left half of concatenation, i.e., $\mathrm{MAC}i_{K_1}(M)$. Let $t \geq n/2$ and each message has same block-length $2t$. We search two $2t$-block messages $a_{1,1} \| r_{1,1} \| 0^n \| \ldots \| 0^n$ and $a_{2,1} \| r_{2,1} \| 0^n \| \ldots \| 0^n$ (the last $2t - 2$ blocks are all zero) such that

$$\mathrm{MAC}i_{K_1}(\mathsf{bin}_n(2t) \| a_{1,1} \| r_{1,1} \| 0^n \| \ldots \| 0^n) = \mathrm{MAC}i_{K_1}(\mathsf{bin}_n(2t) \| a_{2,1} \| r_{2,1} \| 0^n \| \ldots \| 0^n).$$

This requires about $2^{n/2}$ MAC computations due to birthday paradox. Fixing $a_{1,1} \| r_{1,1}$ and $a_{2,1} \| r_{2,1}$, we then search $a_{1,2} \| r_{1,2}$ and $a_{2,2} \| r_{2,2}$ such that

$$\begin{aligned} &\mathrm{MAC}i_{K_1}(\mathsf{bin}_n(2t) \| a_{1,1} \| r_{1,1} \| a_{1,2} \| r_{1,2} \| 0^n \| \ldots \| 0^n) \\ =\ &\mathrm{MAC}i_{K_1}(\mathsf{bin}_n(2t) \| a_{2,1} \| r_{2,1} \| a_{2,2} \| r_{2,2} \| 0^n \| \ldots \| 0^n). \end{aligned}$$

This also requires about $2^{n/2}$ MAC computations. We do this until find two $2t$-block $(t \geq n/2)$ messages $a_{1,1} \| r_{1,1} \| \ldots \| a_{1,t} \| r_{1,t}$ and $a_{2,1} \| r_{2,1} \| \ldots \| a_{2,t} \| r_{2,t}$ such that

$$\mathrm{MAC}i_{K_1}(\mathsf{bin}_n(2t) \| a_{1,1} \| r_{1,1} \| \ldots \| a_{1,t} \| r_{1,t}) = \mathrm{MAC}i_{K_1}(\mathsf{bin}_n(2t) \| a_{2,1} \| r_{2,1} \| \ldots \| a_{2,t} \| r_{2,t}).$$

This yields 2^t different messages $a_{i_1,1} \| r_{i_1,1} \| \ldots \| a_{i_t,t} \| r_{i_t,t}$ for $i_1, \ldots, i_t \in \{1, 2\}$ with the same MAC value on the left half of concatenation. Assume $t \geq n/2$, then with high probability there exists a collision among these 2^t elements such that the MAC value on the right half are also equal. Assume the collided messages are $a_{i_1,1} \| r_{i_1,1} \ldots \| a_{i_t,t} \| r_{i_t,t}$ and $a_{j_1,1} \| r_{j_1,1} \ldots \| a_{j_t,t} \| r_{j_t,t}$, then the two MAC values for $a_{i_1,1} \| r_{i_1,1} \ldots \| a_{i_t,t} \| r_{i_t,t} \oplus A$ and $a_{j_1,1} \| r_{j_1,1} \ldots \| a_{j_t,t} \| r_{j_t,t} \oplus A$ are also a collision for any n-bit block A. This attack requires in total about $(1 + n) \cdot 2^{n/2}$ MAC computations. The forgery attack for the concatenation of two MAC Algorithm 5 with padding scheme 4 is exactly the same as above and hence we omit the details here.

4 Main Results on XMAC1 and XMAC5

In the information-theoretic setting, we simply denote XMAC1$[P]$ the XORing of two MAC Algorithm 1 based on random permutations. Similarly, we denote XMAC5$[P]$ the XORing of two MAC Algorithm 5 based on random permutations. We consider an adversary $\mathcal{A}$ that makes at most q queries to its oracle, each query being at most ℓ blocks, and the total number of blocks of all queries being at most σ.

As for XMAC1, we have the following two results.

Theorem 1. *With padding scheme 3, if $\ell \leq 2^{n/3}$, one has*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathrm{XMAC1}[P]}(\mathcal{A}) \leq \frac{844\sigma q^2 \ell}{2^{2n}}.$$

Theorem 2. *With padding scheme 2, one has*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathrm{XMAC1}[P]}(\mathcal{A}) \leq \frac{2\sigma^2}{2^n} + \frac{2\sigma q}{2^n} + \frac{0.5q^2}{2^n}.$$

As for XMAC5, we have

Theorem 3. *For $\ell \leq 2^{n/3}$, one has*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathrm{XMAC5}[P]}(\mathcal{A}) \leq \frac{4}{2^n} + \frac{58\sigma^2 q}{2^{2n}} + \frac{841\sigma q^2 \ell}{2^{2n}}.$$

The proof of Theorem 1 is given in Sec. 5, the proof of Theorem 2 is given in Appendix A, and the proof of Theorem 3 is given in Sec. 6.

5 Security of XMAC1 with Padding Scheme 3
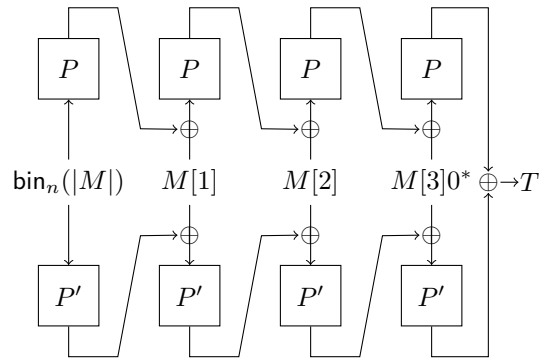
Figure 2: Illustration of XMAC1$[P]$ with padding scheme 3 for $M = M[1]\|M[2]\|M[3]$, where $|M[1]| = |M[2]| = n$ and $1 \leq |M[3]| \leq n$.

In this section, we adopt the framework used in proofs for SUM-ECBC [Yas10] and PMAC_Plus [Yas11], and prove that XMAC1$[P]$ instantiated with padding scheme 3 (described in Fig. 2) is an $O(2^{2n/3})$-secure PRF. Note that in the rest of this section, we always consider the messages list after padded with padding scheme 3, i.e., $M_i = \mathbf{pad3}(M_i)$ and denote by $m_i = |\mathbf{pad3}(M_i)|$ the length of message for $1 \leq i \leq q$. This messages list is easily seen to be prefix-free.

5.1 Main Ideas

We focus on the last input of random permutations P and P' at each query, denoted by $x_{m_i}^i$ and $u_{m_i}^i$ for $1 \leq i \leq q$. We consider an adversary $\mathcal{A}$ that aims at distinguishing XMAC1$[P]$ from a random function $\mathcal{R} : \{0,1\}^* \to \{0,1\}^n$. $\mathcal{A}$ is allowed to unlimited computational power but can make at most q queries to its oracle, each query being at most ℓ blocks, and the total number of blocks of all queries being at most σ. Without loss of generality, $\mathcal{A}$ is assumed to be deterministic and never to repeat a query. The main game is presented in Fig. 3 and codes of the four cases are given in Fig. 4, Fig. 5 and Fig. 6 respectively. Depending on the behavior after bad events, this game can simulate either XMAC1$[P]$ or a random function $\mathcal{R}$. These two games are identical until bad events occur, so by the fundamental lemma of game-playing [BR06] we have

$$\Pr[\mathcal{A}^{\text{XMAC1}[P](\cdot)} \Rightarrow 1] - \Pr[\mathcal{A}^{\mathcal{R}(\cdot)} \Rightarrow 1] \leq \Pr[\mathcal{A}^{\mathcal{R}(\cdot)} \text{ sets } \textbf{bad}].$$

Note that in the game simulating random function $\mathcal{R}$, the respond returning to the adversary is always a random n-bit string, unrelated to adversary's query or the setting of $\textbf{bad}$. Thus even if $\mathcal{A}$ prepares all of its queries $M_1, \ldots, M_q$ in advance, the probability that $\mathcal{A}$ sets a $\textbf{bad}$ flag is not made smaller, therewith the interaction being vacuous in this game. We write bad events in more detail:

$$\Pr[\mathcal{A} \text{ sets } \textbf{bad}]$$

$$= \sum_{i=1}^{q} \big(\Pr[x_{m_i}^i \notin \text{Dom}(P) \wedge u_{m_i}^i \notin \text{Dom}(P')] \cdot \Pr[\mathcal{A} \text{ sets } \textbf{bad} \mid \text{Case A}]$$

$$+ \Pr[x_{m_i}^i \in \text{Dom}(P) \wedge u_{m_i}^i \notin \text{Dom}(P')] \cdot \Pr[\mathcal{A} \text{ sets } \textbf{bad} \mid \text{Case B}]$$

$$+ \Pr[x_{m_i}^i \notin \text{Dom}(P) \wedge u_{m_i}^i \in \text{Dom}(P')] \cdot \Pr[\mathcal{A} \text{ sets } \textbf{bad} \mid \text{Case C}]$$

$$+ \Pr[x_{m_i}^i \in \text{Dom}(P) \wedge u_{m_i}^i \in \text{Dom}(P')] \cdot \Pr[\mathcal{A} \text{ sets } \textbf{bad} \mid \text{Case D}]\big)$$

$$\leq \sum_{i=1}^{q} \Pr[\mathcal{A} \text{ sets } \textbf{bad} \mid \text{Case A}] + \sum_{i=1}^{q} \Pr[x_{m_i}^i \in \text{Dom}(P)] \cdot \Pr[\mathcal{A} \text{ sets } \textbf{bad} \mid \text{Case B}]$$

$$+ \sum_{i=1}^{q} \Pr[u_{m_i}^i \in \text{Dom}(P')] \cdot \Pr[\mathcal{A} \text{ sets } \textbf{bad} \mid \text{Case C}]$$

$$+ \sum_{i=1}^{q} \Pr[x_{m_i}^i \in \text{Dom}(P) \wedge u_{m_i}^i \in \text{Dom}(P')].$$

These four terms are relevant to four cases and will be bounded in following subsections.

5.2 Analysis of Case A

We handle this case via the technique of fair sets developed by Lucks [Luc00], which has also been used in proofs of SUM-ECBC [Yas10] and PMAC_Plus [Yas11].

Lemma 1. *In Case A, we have*

$$\sum_{i=1}^{q} \Pr[\mathcal{A} \text{ sets } \textbf{bad} \mid \text{Case A}] \leq \frac{4\sigma^2 q}{2^{2n}},$$

for $\sigma \leq 2^{n-1}$.

```
 1: for i = 1 to q do
 2:     x^i_{m_i} ← P(x^i_{m_i-1}) ⊕ M_i[m_i]
 3:     u^i_{m_i} ← P'(u^i_{m_i-1}) ⊕ M_i[m_i]
 4:     if x^i_{m_i} ∉ Dom(P) and u^i_{m_i} ∉ Dom(P') then
 5:         go to Case A
 6:     end if
 7:     if x^i_{m_i} ∈ Dom(P) and u^i_{m_i} ∉ Dom(P') then
 8:         go to Case B
 9:     end if
10:     if x^i_{m_i} ∉ Dom(P) and u^i_{m_i} ∈ Dom(P') then
11:         go to Case C
12:     end if
13:     if x^i_{m_i} ∈ Dom(P) and u^i_{m_i} ∈ Dom(P') then
14:         go to Case D
15:     end if
16: end for
```

Figure 3: We omit the internal computations and present the computation on the last input of permutation P and P' at each query. $x^i_{m_i}$ and $u^i_{m_i}$ respectively denote the last input to P and P' at i-th query. $x^i_{m_i-1}$ and $u^i_{m_i-1}$ respectively denote $(m_i - 1)$-th input to P and P' at i-th query.

```
 1: Choose a fair set U ⊂ Ran(P)‾ × Ran(P')‾                    XMAC1[P]/ R
 2: (y^i_{m_i}, w^i_{m_i}) ←$ Ran(P)‾ × Ran(P')‾
 3: if (y^i_{m_i}, w^i_{m_i}) ∉ U then
 4:     bad←true   (y^i_{m_i}, w^i_{m_i}) ←$ U
 5: end if
 6: T_i ← y^i_{m_i} ⊕ w^i_{m_i}
 7: return T_i
```

Figure 4: Case A

Proof. We consider the game as described in Fig. 4. The code without the boxed statement faithfully simulates $P(x^i_{m_i}) \oplus P'(u^i_{m_i})$ for $1 \leq i \leq q$, while the code with boxed statement always returns a n-bit random string T_i. We choose a fair set U as follows. Enumerate $\mathrm{Ran}(P)$ as $\{y_1, \ldots, y_\alpha\}$ and $\mathrm{Ran}(P')$ as $\{w_1, \ldots, w_\beta\}$. For each $y_i \in \{y_1, \ldots, y_\alpha\}$ and $w_j \in \{w_1, \ldots, w_\beta\}$, we choose arbitrarily representatives $(y'_i, w'_j) \in \overline{\mathrm{Ran}(P)} \times \overline{\mathrm{Ran}(P')}$ such that $y'_i \oplus w'_j = y_i \oplus w_j$ for $1 \leq i \leq \alpha$ and $1 \leq j \leq \beta$. We remove these $\alpha\beta$ pairs (y'_i, w'_j) from $\overline{\mathrm{Ran}(P)} \times \overline{\mathrm{Ran}(P')}$ and obtain U. For each value $T \in \{0,1\}^n$, we have

$$|\{(y, w) \in U \mid y \oplus w = T\}| = 2^n - \alpha - \beta,$$

i.e., the chance to induce T from U is equal. Let α_i and β_i respectively denote the number

of new defined domain points of P and P' at i-th query. Then

$$\sum_{i=1}^{q} \Pr[\mathcal{A} \text{ sets } \mathbf{bad} \mid \text{Case A}]$$

$$\leq \sum_{i=1}^{q} \frac{|(\overline{\text{Ran}(P)} \times \overline{\text{Ran}(P')}) \setminus U|}{|\overline{\text{Ran}(P)} \times \overline{\text{Ran}(P')}|}$$

$$= \sum_{i=1}^{q} \frac{(\alpha_1 + \cdots + \alpha_i - 1)(\beta_1 + \cdots + \beta_i - 1)}{(2^n - \alpha_1 - \cdots - \alpha_i + 1)(2^n - \beta_1 - \cdots - \beta_i + 1)}$$

$$\leq \sum_{i=1}^{q} \frac{\sigma^2}{(2^n - \sigma)^2} \leq \frac{4\sigma^2 q}{2^{2n}}$$

under the condition $\sigma \leq 2^{n-1}$ and concludes the proof of Lemma 1. $\qquad \square$

5.3 Analysis of Case B

In this case, $x_{m_i}^i$ collides with previous inputs to P. The output string is $T_i = y_{m_i}^i \oplus w_{m_i}^i$. That is, either $y_{m_i}^i$ or $w_{m_i}^i$ being random may make T_i a random string. Our goal is to bound the probability that $x_{m_i}^i$ collides with previous inputs of P and subsequently $w_{m_i}^i$ deviates from a random n-bit string.

We first use the following *full collision probability* lemma proved in [BPR05b, BPR05a, JN16] to bound the probability of $x_{m_i}^i$ colliding with previous inputs of P. For any two prefix-free messages M_i and M_j, the full collision probability $\mathbf{FCP}_n(M_i, M_j)$ is the probability of the event $x_{m_j}^j \in \{x_1^i, \ldots, x_{m_i}^i,$
$x_1^j, \ldots, x_{m_j-1}^j\}$ where for each $b \in \{i, j\}$, we have $x_k^b = P(x_{k-1}^b) \oplus M_b[k]$ for $2 \leq k \leq m_b$ and $x_1^b = M_b[1]$.

Lemma 2 (Full Collision Probability). *For any two prefix-free messages $M_i \in \{0,1\}^{m_i n}$ and $M_j \in \{0,1\}^{m_j n}$, we have*

$$\mathbf{FCP}_n(M_i, M_j) \leq \frac{3(m_i + m_j)}{2^n - m_i - m_j} + \frac{(m_i + m_j)^4}{2^{2n}}.$$

Remark on This Lemma. The full collision probability lemma is first proved by Bellare *et al.* in [BPR05b] and then refined in its full version [BPR05a]. Recently Jha and Nandi [JN16] pointed out a flaw in the previous proof.

We denote by $\mathbf{FCP}_n(\emptyset, M_1)$ the probability of the special case $x_{m_1}^1 \in \{x_1^1, \ldots, x_{m_1-1}^1\}$ and it is easily seen that $\mathbf{FCP}_n(\emptyset, M_1) \leq \frac{3m_1}{2^n - m_1} + \frac{m_1^4}{2^{2n}}$. Then at i-th query, we have

$$\Pr[x_{m_i}^i \in \text{Dom}(P)] \leq \sum_{j=1}^{i-1} \mathbf{FCP}_n(M_j, M_i).$$

We next utilize game-playing techniques to examine the randomness of string $w_{m_i}^i$. Note that if we pick $w_{m_i}^i$ as $w_{m_i}^i \xleftarrow{\$} \{0,1\}^n$, then the distribution of $T_i = y_{m_i}^i \oplus w_{m_i}^i$ would be uniformly random. We consider the game presented in Fig. 5. The code with the boxed statement is the simulation of $P(x_{m_i}^i) \oplus P'(u_{m_i}^i)$ while the code without boxed statement corresponds to a random function. Without a bad event occurring, the responses that $\mathcal{A}$ receives from the oracle are uniform and independent binary strings. We see that the bad event occurs with a probability of $|\text{Ran}(P')|/2^n$ for each sampling operation, which is at most $\sigma/2^n$.
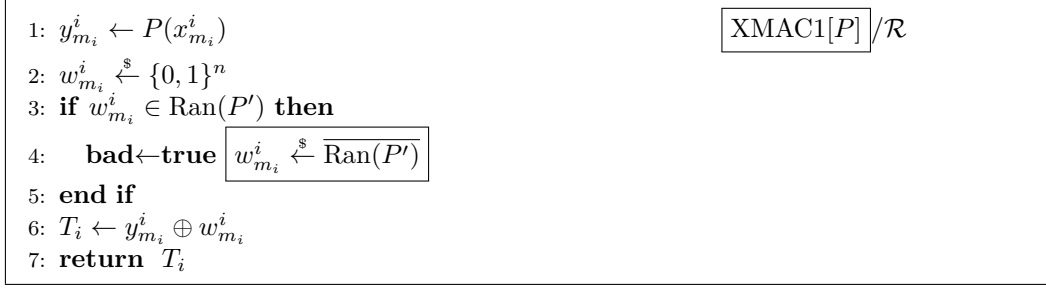
$$
\boxed{
\begin{array}{ll}
\text{1: } & y^i_{m_i} \leftarrow P(x^i_{m_i}) \hspace{5cm} \boxed{\text{XMAC1}[P]}/\mathcal{R} \\[4pt]
\text{2: } & w^i_{m_i} \xleftarrow{\$} \{0,1\}^n \\[4pt]
\text{3: } & \textbf{if } w^i_{m_i} \in \mathrm{Ran}(P') \textbf{ then} \\[4pt]
\text{4: } & \quad \textbf{bad} \leftarrow \textbf{true} \boxed{\; w^i_{m_i} \xleftarrow{\$} \overline{\mathrm{Ran}(P')} \;} \\[4pt]
\text{5: } & \textbf{end if} \\[4pt]
\text{6: } & T_i \leftarrow y^i_{m_i} \oplus w^i_{m_i} \\[4pt]
\text{7: } & \textbf{return } T_i
\end{array}
}
$$

Figure 5: Case B

Let $M_1, M_2, \ldots, M_q$ be a sequence of messages, then

$$
\sum_{i=1}^{q} \Pr[x^i_{m_i} \in \mathrm{Dom}(P)] \cdot \Pr[\mathcal{A} \text{ sets } \mathbf{bad} \mid \text{Case B}]
$$

$$
\leq \sum_{i=1}^{q} \Pr[x^i_{m_i} \in \mathrm{Dom}(P)] \cdot \frac{|\mathrm{Ran}(P')|}{2^n}
$$

$$
\leq \frac{\sigma}{2^n} \cdot \left(\mathbf{FCP}_n(\emptyset, M_1) + \sum_{i=2}^{q} \sum_{j=1}^{i-1} \mathbf{FCP}_n(M_j, M_i) \right)
$$

$$
\leq \frac{\sigma}{2^n} \cdot \left(\frac{6m_1}{2^n} + \frac{m_1^4}{2^{2n}} + \sum_{i=2}^{q} \sum_{j=1}^{i-1} \left(\frac{6(m_i + m_j)}{2^n} + \frac{(m_i + m_j)^4}{2^{2n}} \right) \right)
$$

$$
\leq \frac{\sigma}{2^n} \cdot \left(\frac{12\sigma q}{2^n} + \frac{16\sigma q \ell^3}{2^{2n}} \right) \leq \frac{28\sigma^2 q}{2^{2n}},
$$

if $\ell \leq 2^{n/3}$.

5.4   Analysis of Case C

The analysis of Case C is identical to Case B since P and P' are two independent random permutations. We obtain the same upper bound in this case:

$$
\sum_{i=1}^{q} \Pr[u^i_{m_i} \in \mathrm{Dom}(P')] \cdot \Pr[\mathcal{A} \text{ sets } \mathbf{bad} \mid \text{Case C}] \leq \frac{28\sigma^2 q}{2^{2n}},
$$

if $\ell \leq 2^{n/3}$.

5.5   Analysis of Case D

In this case, $y^i_{m_i}$ and $w^i_{m_i}$ both have appeared before and T_i is not a random string anymore. As shown in Fig.6, we always set **bad** flag in this case. The code with boxed statements simulates $P(x^i_{m_i}) \oplus P'(u^i_{m_i})$ for $1 \leq i \leq q$ while the code without boxed statements is the simulation of a random function.

```
1: T_i ←$ {0,1}^n                                          XMAC1[P]/R
2: bad←true
   | y^i_{m_i} ← P(x^i_{m_i})
   | w^i_{m_i} ← P(u^i_{m_i})
   | T_i ← y^i_{m_i} ⊕ w^i_{m_i}
3: return T_i
```

Figure 6: Case D

Let $M_1, \ldots, M_q$ be a sequence of messages, then by using Lemma 2, we have

$$\sum_{i=1}^{q} \Pr[x^i_{m_i} \in \mathrm{Dom}(P) \wedge u^i_{m_i} \in \mathrm{Dom}(P')]$$

$$\leq \mathbf{FCP}_n(\emptyset, M_1)^2 + \sum_{i=2}^{q} \sum_{j=1}^{i-1} \sum_{k=1}^{i-1} \mathbf{FCP}_n(M_j, M_i) \cdot \mathbf{FCP}_n(M_k, M_i)$$

$$\leq \left(\frac{6m_1}{2^n} + \frac{m_1^4}{2^{2n}}\right)^2 + \sum_{i=2}^{q} \sum_{j=1}^{i-1} \sum_{k=1}^{i-1} \left(\frac{6(m_i + m_j)}{2^n} + \frac{(m_i + m_j)^4}{2^{2n}}\right) \cdot \left(\frac{6(m_i + m_k)}{2^n} + \frac{(m_i + m_k)^4}{2^{2n}}\right)$$

$$\leq \frac{144\sigma q^2 \ell}{2^{2n}} + \frac{384\sigma q^2 \ell^4}{2^{3n}} + \frac{256\sigma q^2 \ell^7}{2^{4n}} \leq \frac{784\sigma q^2 \ell}{2^{2n}},$$

if $\ell \leq 2^{n/3}$.

5.6   Summation

Finally we sum up the probabilities over above four cases and obtain

$$\Pr[\mathcal{A}^{\mathcal{R}(\cdot)} \text{ sets } \mathbf{bad}]$$

$$\leq \frac{4\sigma^2 q}{2^{2n}} + \frac{28\sigma^2 q}{2^{2n}} + \frac{28\sigma^2 q}{2^{2n}} + \frac{784\sigma q^2 \ell}{2^{2n}}$$

$$\leq \frac{844\sigma q^2 \ell}{2^{2n}}$$

under the condition $\ell \leq 2^{n/3}$, which completes the proof of Theorem 1.

6   Security of XMAC5

The proof of XMAC5 is similar to the proof of XMAC1 with padding scheme 3 and we only outline their main differences here. We use exactly the same main game except that we need to set **bad** flag in following additional events:

- $L_1 \cdot u = 0^n$, $L_2 \cdot u = 0^n$, $L_1 \cdot u^2 = 0^n$, $L_2 \cdot u^2 = 0^n$, i.e., $L_1 = 0^n$ or $L_2 = 0^n$. $\Pr[L_1 = 0^n \vee L_2 = 0^n] = \Pr[P, P' \xleftarrow{\$} \mathrm{Perm}(n) : P(0^n) = 0^n \vee P'(0^n) = 0^n] = \frac{2}{2^n}$;

- $L_1 \cdot u = L_1 \cdot u^2$, $L_2 \cdot u = L_2 \cdot u_2$, i.e., $L_1 = \mathrm{constant}_1$, $L_2 = \mathrm{constant}_2$. $\Pr[L_1 = \mathrm{constant}_1 \vee L_2 = \mathrm{constant}_2] = \Pr[P, P' \xleftarrow{\$} \mathrm{Perm}(n) : P(0^n) = \mathrm{constant}_1 \vee P'(0^n) = \mathrm{constant}_2] = \frac{2}{2^n}$.

These account to a term $\frac{4}{2^n}$. The definitions of four cases (A,B,C,D) are the same as in Section 5. Note that $P(0^n)$ and $P'(0^n)$ have been defined at the beginning and thus $P(0^n) \in \mathrm{Ran}(P)$ and $P'(0^n) \in \mathrm{Ran}(P')$. We always consider a padded sequence of
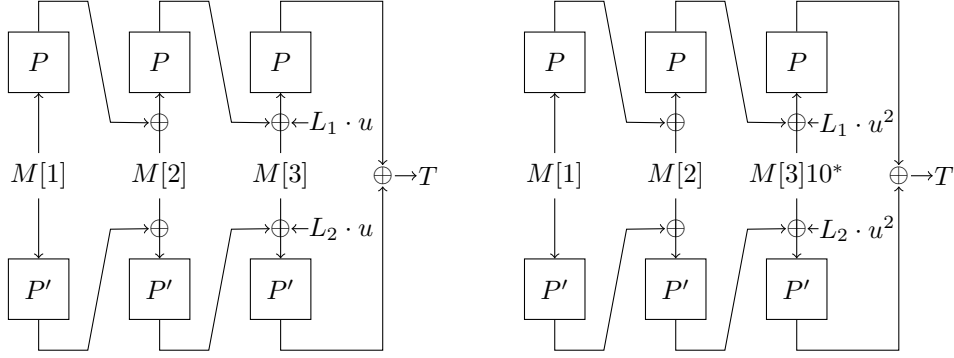
Figure 7: Illustration of XMAC5[P]. The left is the case where the message length is a positive multiple of n while the right is the case where the message length is not a positive multiple of n. $L_1 = P(0^n)$ and $L_2 = P'(0^n)$, u is some non-zero constant, and $\cdot$ is field multiplication.

messages $M_1, \ldots, M_q$, that is, each message has been padded with 10^* when the length is not a positive multiple of n and further the last block of each message has been bitwise exclusive-or with either $L_i \cdot u$ or $L_i \cdot u^2$ depending on the length, and $i = 1$ when analyzing P, $i = 2$ when analyzing P'.

- In the Case A, we have

$$\sum_{i=1}^{q} \Pr[\mathcal{A} \text{ sets } \mathbf{bad} \mid \text{Case A}]$$

$$\leq \sum_{i=1}^{q} \frac{|(\overline{\mathrm{Ran}(P)} \times \overline{\mathrm{Ran}(P')}) \setminus U|}{|\overline{\mathrm{Ran}(P)} \times \overline{\mathrm{Ran}(P')}|}$$

$$= \sum_{i=1}^{q} \frac{(\alpha_1 + \cdots + \alpha_i)(\beta_1 + \cdots + \beta_i)}{(2^n - \alpha_1 - \cdots - \alpha_i)(2^n - \beta_1 - \cdots - \beta_i)}$$

$$\leq \sum_{i=1}^{q} \frac{\sigma^2}{(2^n - \sigma)^2} \leq \frac{4\sigma^2 q}{2^{2n}}$$

  under the condition $\sigma \leq 2^{n-1}$.

- In the Case B, the probability that $x_{m_i}^i$ collides with the values in $\mathrm{Dom}(P)$ would be slightly enlarged as M_i may be a prefix of previous messages (we consider the messages list that after padded with 10^* when the length is not a positive multiple of n and further the last block of each message has been bitwise exclusive-or with either $L_1 \cdot u$ or $L_1 \cdot u^2$ depending on the length of message). The probability of M_i being a prefix of previous messages or $M_i = 0^n$ is at most $\frac{i}{2^n}$, as if M_i is a prefix of M_j for $1 \leq j \leq i-1$ or $M_i = 0^n$:

  - $m_i = m_j$, then we have $M_i[m_i] \oplus L_1 \cdot u = M_j[m_j] \oplus L_1 \cdot u^2$, which happens with probability of $\frac{1}{2^n}$;
  - if $m_i < m_j$, then we have $M_i[1]\| \ldots \|M_i[m_i - 1] = M_j[1]\| \ldots \|M_j[m_i - 1]$ and $M_i[m_i] \oplus L_1 \cdot u = M_j[m_i]$ (or $M_i[m_i] \oplus L_1 \cdot u^2 = M_j[m_i]$), which happens with probability of $\frac{1}{2^n}$;
  - if $m_i > m_j$, then it is impossible;
  - if $M_i = 0^n$, then $M_i[1] \oplus L_1 \cdot u = 0^n$ (or $M_i[1] \oplus L_1 \cdot u^2 = 0^n$), which happens with probability of $\frac{1}{2^n}$.

Let $M_0 = 0^n$ and $M_1, \ldots, M_q$ be a sequence of messages, then

$$\sum_{i=1}^{q} \Pr[x_{m_i}^i \in \mathrm{Dom}(P)] \cdot \Pr[\mathcal{A} \text{ sets } \mathbf{bad} \mid \text{Case B}]$$

$$\leq \sum_{i=1}^{q} \Pr[x_{m_i}^i \in \mathrm{Dom}(P)] \cdot \frac{|\mathrm{Ran}(P')|}{2^n}$$

$$\leq \frac{\sigma}{2^n} \cdot \left(\sum_{i=2}^{q} \Pr[M_i \text{ is a prefix} \vee M_i = 0^n] + \sum_{i=1}^{q} \sum_{j=0}^{i-1} \mathbf{FCP}_n(M_j, M_i) \right)$$

$$\leq \frac{\sigma}{2^n} \cdot \left(\sum_{i=1}^{q} \frac{i}{2^n} + \sum_{i=1}^{q} \sum_{j=0}^{i-1} (\frac{6(m_i + m_j)}{2^n} + \frac{(m_i + m_j)^4}{2^{2n}}) \right)$$

$$\leq \frac{\sigma}{2^n} \cdot \left(\frac{q(q+1)}{2^{n+1}} + \frac{12\sigma q}{2^n} + \frac{16\sigma q \ell^3}{2^{2n}} \right) \leq \frac{29\sigma^2 q}{2^{2n}},$$

if $\ell \leq 2^{n/3}$.

- In the Case C, the analysis is exactly the same as in the Case B, and thus

$$\sum_{i=1}^{q} \Pr[u_{m_i}^i \in \mathrm{Dom}(P')] \cdot \Pr[\mathcal{A} \text{ sets } \mathbf{bad} \mid \text{Case C}] \leq \frac{29\sigma^2 q}{2^{2n}},$$

if $\ell \leq 2^{n/3}$.

- In the Case D, we have

$$\sum_{i=1}^{q} \Pr[x_{m_i}^i \in \mathrm{Dom}(P) \wedge u_{m_i}^i \in \mathrm{Dom}(P')]$$

$$\leq \sum_{i=1}^{q} \left(\frac{i}{2^n} + \sum_{j=0}^{i-1} \mathbf{FCP}_n(M_j, M_i) \right) \cdot \left(\frac{i}{2^n} + \sum_{k=0}^{i-1} \mathbf{FCP}_n(M_k, M_i) \right)$$

$$\leq \sum_{i=1}^{q} \sum_{j=0}^{i-1} \sum_{k=0}^{i-1} \mathbf{FCP}_n(M_j, M_i) \cdot \mathbf{FCP}_n(M_k, M_i)$$

$$+ \frac{2q}{2^n} \sum_{i=1}^{q} \sum_{j=0}^{i-1} \mathbf{FCP}_n(M_j, M_i) + \sum_{i=1}^{q} \frac{i^2}{2^{2n}}$$

$$\leq \sum_{i=1}^{q} \sum_{j=0}^{i-1} \sum_{k=0}^{i-1} \left(\frac{6(m_i + m_j)}{2^n} + \frac{(m_i + m_j)^4}{2^{2n}} \right) \cdot \left(\frac{6(m_i + m_k)}{2^n} + \frac{(m_i + m_k)^4}{2^{2n}} \right)$$

$$+ \frac{2q}{2^n} \sum_{i=1}^{q} \sum_{j=0}^{i-1} (\frac{6(m_i + m_j)}{2^n} + \frac{(m_i + m_j)^4}{2^{2n}}) + \frac{q(q+1)^2}{3 \cdot 2^{2n}}$$

$$\leq \frac{784\sigma q^2 \ell}{2^{2n}} + \frac{2q}{2^n} \cdot \left(\frac{12\sigma q}{2^n} + \frac{16\sigma q \ell^3}{2^{2n}} \right) + \frac{q(q+1)^2}{3 \cdot 2^{2n}} \leq \frac{841\sigma q^2 \ell}{2^{2n}}$$

if $\ell \leq 2^{n/3}$.

Summing up the above possibilities, the PRF-security bound of XMAC5 can be bounded by

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathrm{XMAC5}[P]}(q, \ell, \sigma) \leq \frac{4}{2^n} + \frac{58\sigma^2 q}{2^{2n}} + \frac{841\sigma q^2 \ell}{2^{2n}}.$$

7 Conclusion

In this paper, we present forgery attacks for the concatenation of MACs in the standard ISO/IEC 9797-1:2011. More concretely, we give a 3-queries forgery attack for the concatenation of MAC Algorithm 1 with padding scheme 2, and give birthday-bound-complexity forgery attacks for other concatenating MACs. We then turn to consider how to lift the security bound of these MACs to a higher level. We prove that XMAC1 can achieve beyond birthday bound security with padding scheme 3, and achieve birthday bound security with padding scheme 2. We also prove that XMAC5 can achieve $2n/3$-bit security. Our results imply that XORing of any two CBC-like MACs in ISO/IEC 9797-1:2011 can lift up the security bound to a higher level while the previous suggestion by concatenating two MACs in this standard does not. It seems unlikely that $O(2^{2n/3})$ is a tight bound for either XMAC1 or XMAC5. A future work is to further improve the provable-security bounds of these constructions.

Acknowledgments

References

[BCG+12] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, et al. Prince–a low-latency block cipher for pervasive computing applications. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 208–225. Springer, 2012.

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

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

[BL16] Karthikeyan Bhargavan and Gaëtan Leurent. On the practical (in-)security of 64-bit block ciphers: Collision attacks on HTTP over TLS and openvpn. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 456–467, 2016.

[BPR05a] M Bellare, K Pietrzak, and P Rogaway. Improved security analyses for cbc macs. https://cseweb.ucsd.edu/~mihir/papers/cbc-improved.pdf, 2005.

[BPR05b] Mihir Bellare, Krzysztof Pietrzak, and Phillip Rogaway. Improved security analyses for CBC macs. In *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, pages 527–545, 2005.

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, pages 409–426, 2006.

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

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

[ISO99] ISO/IEC:Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher. Iso/iec 9797-1:1999, 1999.

[ISO11] ISO/IEC:Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher. Iso/iec 9797-1:2011, 2011.

[JN16] Ashwin Jha and Mridul Nandi. Revisiting structure graph and its applications to CBC-MAC and EMAC. *IACR Cryptology ePrint Archive*, 2016:161, 2016.

[Jou04] Antoine Joux. Multicollisions in iterated hash functions. application to cascaded constructions. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, pages 306–316, 2004.

[JPS03] Antoine Joux, Guillaume Poupard, and Jacques Stern. New attacks against standardized macs. In *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers*, pages 170–181, 2003.

[Luc00] Stefan Lucks. The sum of prps is a secure PRF. In *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, pages 470–484, 2000.

[Pat08] Jacques Patarin. A proof of security in o(2n) for the xor of two random permutations. In *Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada, August 10-13, 2008, Proceedings*, pages 232–248, 2008.

[PvO95]   Bart Preneel and Paul C. van Oorschot. Mdx-mac and building fast macs from hash functions. In *Advances in Cryptology - CRYPTO '95, 15th Annual International Cryptology Conference, Santa Barbara, California, USA, August 27-31, 1995, Proceedings*, pages 1–14, 1995.

[PvO99]   Bart Preneel and Paul C. van Oorschot. On the security of iterated message authentication codes. *IEEE Trans. Information Theory*, 45(1):188–199, 1999.

[Yas10]   Kan Yasuda. The sum of CBC macs is a secure PRF. In *Topics in Cryptology - CT-RSA 2010, The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010. Proceedings*, pages 366–381, 2010.

[Yas11]   Kan Yasuda. A new variant of PMAC: beyond the birthday bound. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 596–609, 2011.

[ZWSW12] Liting Zhang, Wenling Wu, Han Sui, and Peng Wang. 3kf9: Enhancing 3gpp-mac beyond the birthday bound. In *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, pages 296–312, 2012.

A   Security of XMAC1 with Padding Scheme 2
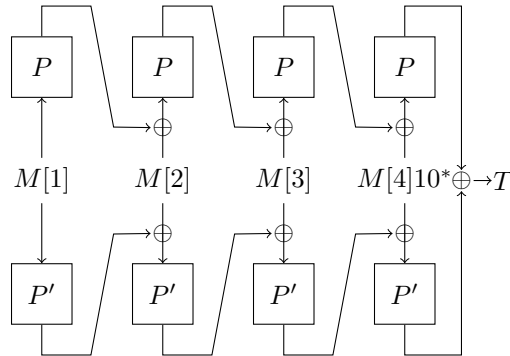
Figure 8:   Illustration of XMAC1$[P]$ with padding scheme 2 for $M = M[1]\|M[2]\|M[3]\|M[4]$, where $|M[1]| = |M[2]| = |M[3]| = n$ and $0 \leq |M[4]| \leq n - 1$.

We always consider the messages list after padded with padding scheme 2, i.e., $M_i = \mathbf{pad2}(M_i)$ and denote the message length $m_i = |\mathbf{pad2}(M_i)|$. Note that padding scheme 2 is not a prefix-free encoding, and if the adversary asks a long message $M_1 = M_2\|*$ then asks M_2 a prefix of M_1, the internal values during computing M_2 have all been defined before. Hence we cannot use the same proof methods in section 5 or section 6. In this section, we resort to the well-known H-coefficient technique [CS14, Pat08] and prove that XMAC1$[P]$ with padding scheme 3 is a $O(2^{n/2})$-secure PRF.

A.1   Double Collision Attack

For the sake of completeness, we sketch the double collision attack [JPS03] here for XMAC1 with padding scheme 2. First we search for two one-block messages M_1 and M_2 that yield the same tag $T = E_{K_1}(M_1) \oplus E_{K_2}(M_1) = E_{K_1}(M_2) \oplus E_{K_2}(M_2)$. Then we compute the tag values for two one block longer messages of the form $M_1\|A$ and $M_2\|B$ for random

A, B. If it holds that $A \oplus B = E_{K_1}(M_1) \oplus E_{K_1}(M_2) = E_{K_2}(M_1) \oplus E_{K_2}(M_2)$, then we get collisions for both the upper chain and lower chain and thus a collision on the MAC values of these two extended messages. We can check this kind of double collisions by adding a same block at the end of both extended messages and the resulting messages still collide. This attack requires about $2^{1+n/2}$ MAC computations. However, such a attack does not work out for either XMAC5 or XMAC1 with padding scheme 3, as the padded messages list is always prefix-free except with a negligible probability.

A.2 The H-coefficient Technique

We briefly introduce the H-coefficient technique [CS14, Pat08] here as the following part of this section adopts this method. A view ν is the query-response tuples that $\mathcal{A}$ receives when interacting with either $F[P]$ (real world) or $\mathcal{R}$ (ideal world). We denote X_{re}, resp. X_{id}, the probability distribution of the ν when $\mathcal{A}$ interacts with $F[P]$, resp. $\mathcal{R}$. We also denote $\Theta = \{\nu \mid \Pr[X_{\mathrm{id}} = \nu] > 0\}$ the set of all attainable views ν while $\mathcal{A}$ interacting with $\mathcal{R}$. The H-coefficient technique evaluates the upper bound of $\mathbf{Adv}_{F[P]}^{\mathrm{prf}}(\mathcal{A})$ by using the following lemma. The proof of this lemma can be found in [CLL+14, CS14].

Lemma 3. *Let Θ_{good} and Θ_{bad} be two disjoint subsets of Θ satisfying $\Theta = \Theta_{\mathrm{good}} \sqcup \Theta_{\mathrm{bad}}$. If there exists ϵ_1 such that $\Pr[X_{\mathrm{id}} \in \Theta_{\mathrm{bad}}] \leq \epsilon_1$ and for each view $\nu \in \Theta_{\mathrm{good}}$, it has*

$$\frac{\Pr[X_{\mathrm{re}} = \nu]}{\Pr[X_{\mathrm{id}} = \nu]} \geq 1 - \epsilon_2.$$

Then $\mathbf{Adv}_{F[P]}^{\mathrm{prf}}(\mathcal{A}) \leq \epsilon_1 + \epsilon_2$.

A.3 Preparations for the H-coefficient Technique

We replace P and P' in XMAC1$[P]$ by random functions F and F', respectively. We write the resulting algorithm as XMAC1$[\mathcal{R}]$. Using the PRP/PRF switching lemma [BR06], we obtain

$$\mathbf{Adv}_{\mathrm{XMAC1}[P]}^{\mathrm{prf}}(q, \ell, \sigma) \leq \frac{\sigma^2}{2^n} + \mathbf{Adv}_{\mathrm{XMAC1}[\mathcal{R}]}^{\mathrm{prf}}(q, \ell, \sigma).$$

We define the following functions from F, F', rand_i and rand_i', where $\mathsf{rand}_i, \mathsf{rand}_i' \xleftarrow{\$} \{0,1\}^n$ for $1 \leq i \leq \ell - 1$:
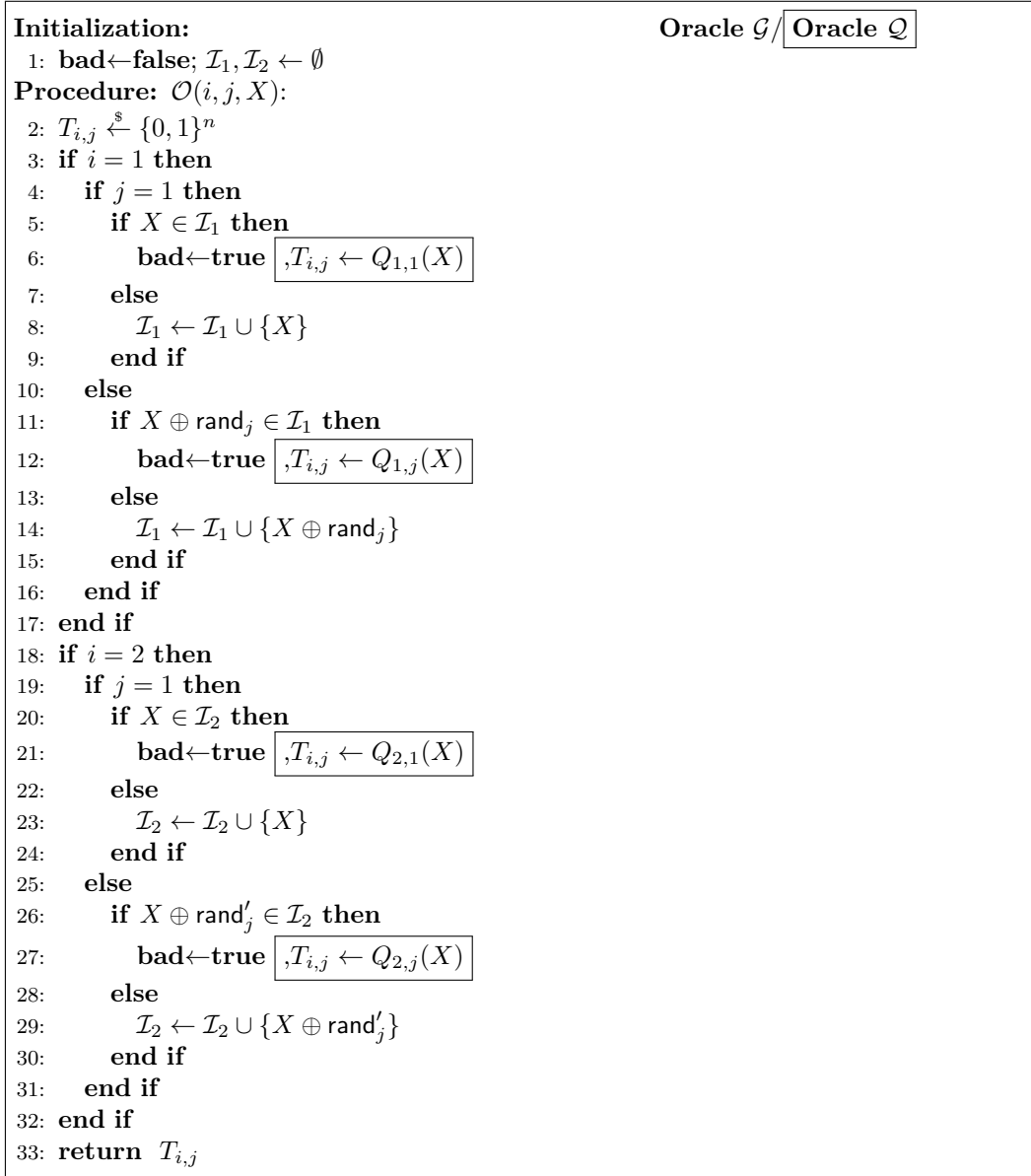
$$\begin{cases} Q_{1,1}(X) & = & F(X) \oplus \mathsf{rand}_1 \\ Q_{1,i}(X) & = & F(X \oplus \mathsf{rand}_{i-1}) \oplus \mathsf{rand}_i \text{ for } 2 \leq i \leq \ell - 1 \\ Q_{1,\ell}(X) & = & F(X \oplus \mathsf{rand}_{\ell-1}) \\ Q_{2,1}(X) & = & F'(X) \oplus \mathsf{rand}_1' \\ Q_{2,i}(X) & = & F'(X \oplus \mathsf{rand}_{i-1}') \oplus \mathsf{rand}_i' \text{ for } 2 \leq i \leq \ell - 1 \\ Q_{2,\ell}(X) & = & F'(X \oplus \mathsf{rand}_{\ell-1}') \end{cases}$$

We write $\mathcal{Q}$ for the set of these functions. Let $G_{i,j}$ be 2ℓ independent random functions, for $1 \leq i \leq 2$ and $1 \leq j \leq \ell$. We write $\mathcal{G}$ for the set of these functions. For an adversary $\mathcal{B}$, we define

$$\mathbf{Adv}_{\mathcal{Q}}^{\mathrm{prf}}(\mathcal{B}) \overset{\mathrm{def}}{=} \Pr[\mathcal{B}^{\mathcal{Q}(\cdot)} \Rightarrow 1] - \Pr[\mathcal{B}^{\mathcal{G}(\cdot)} \Rightarrow 1],$$

where in the right-hand side of the equation, the first probability is taken over F, F', $\mathsf{rand}_i, \mathsf{rand}_i'$ and $\mathcal{B}$'s coin, and the second one is over random functions in $\mathcal{G}$ and $\mathcal{B}$'s coin. $\mathcal{B}$ makes queries of the form $(i, j, X) \in \{1, 2\} \times \{1, 2, \cdots, \ell\} \times \{0, 1\}^n$, and receives $Q_{i,j}(X)$ or $G_{i,j}(X)$. We prove that $\mathcal{Q}$ is indistinguishable from $\mathcal{G}$ by the following lemma.

Lemma 4. *Let $\mathcal{B}$ be an adversary that makes at most q queries. Then we have $\mathbf{Adv}_{\mathcal{Q}}^{\mathrm{prf}}(\mathcal{B}) \leq 0.5q^2/2^n$.*

Initialization: Oracle $\mathcal{G}$/ | Oracle $\mathcal{Q}$ |
 1: **bad**←**false**; $\mathcal{I}_1, \mathcal{I}_2 \leftarrow \emptyset$
Procedure: $\mathcal{O}(i, j, X)$:
 2: $T_{i,j} \xleftarrow{\$} \{0,1\}^n$
 3: **if** $i = 1$ **then**
 4:    **if** $j = 1$ **then**
 5:       **if** $X \in \mathcal{I}_1$ **then**
 6:          **bad**←**true** $\boxed{,T_{i,j} \leftarrow Q_{1,1}(X)}$
 7:       **else**
 8:          $\mathcal{I}_1 \leftarrow \mathcal{I}_1 \cup \{X\}$
 9:       **end if**
10:    **else**
11:       **if** $X \oplus \mathsf{rand}_j \in \mathcal{I}_1$ **then**
12:          **bad**←**true** $\boxed{,T_{i,j} \leftarrow Q_{1,j}(X)}$
13:       **else**
14:          $\mathcal{I}_1 \leftarrow \mathcal{I}_1 \cup \{X \oplus \mathsf{rand}_j\}$
15:       **end if**
16:    **end if**
17: **end if**
18: **if** $i = 2$ **then**
19:    **if** $j = 1$ **then**
20:       **if** $X \in \mathcal{I}_2$ **then**
21:          **bad**←**true** $\boxed{,T_{i,j} \leftarrow Q_{2,1}(X)}$
22:       **else**
23:          $\mathcal{I}_2 \leftarrow \mathcal{I}_2 \cup \{X\}$
24:       **end if**
25:    **else**
26:       **if** $X \oplus \mathsf{rand}'_j \in \mathcal{I}_2$ **then**
27:          **bad**←**true** $\boxed{,T_{i,j} \leftarrow Q_{2,j}(X)}$
28:       **else**
29:          $\mathcal{I}_2 \leftarrow \mathcal{I}_2 \cup \{X \oplus \mathsf{rand}'_j\}$
30:       **end if**
31:    **end if**
32: **end if**
33: **return** $T_{i,j}$

Figure 9: Game used to prove Lemma 4

Proof. When $\mathcal{B}$ interacts with the oracle $\mathcal{Q}$, we define two sets. $\mathcal{I}_1$ is the set of input values of F in $Q_{1,i}$ and $\mathcal{I}_2$ is the set of input values of F' in $Q_{2,i}$, for $1 \le i \le \ell$. We set a **bad** flag if $\mathcal{I}_1$ has a collision or $\mathcal{I}_2$ has a collision. The code with boxed statements simulates $\mathcal{Q}$ while the code without boxed statements simulates $\mathcal{G}$. By the fundamental lemma of game-playing [BR06], we have

$$\mathbf{Adv}_{\mathcal{Q}}^{\mathrm{prf}}(\mathcal{B}) \le \Pr[\mathcal{B}^{\mathcal{G}(\cdot)} \text{ sets } \mathbf{bad}].$$

Before bad event occurs, $\mathcal{B}$ learns nothing from the values returned by the oracle except a random n-bit string. Hence We only need to consider a fixed sequence of queries made by $\mathcal{B}$. Suppose that $\mathcal{B}$ makes total q_1 queries to $Q_{1,i}$ and makes total q_2 queries to $Q_{2,i}$ for $1 \le i \le \ell$. $\mathcal{I}_1$ has a collision if and only if $X = X' \oplus \mathsf{rand}_i$ or $X \oplus \mathsf{rand}_i = X' \oplus \mathsf{rand}_j$ for $1 \le i, j \le \ell - 1$. Since rand_i is a random string, we have $\Pr[\mathcal{I}_1 \text{ has a collision}] \le 0.5 q_1^2 / 2^n$.

1: $M_i[1]\|\dots\|M_i[m_i] \xleftarrow{n} \mathbf{pad2}(M_i)$
2: $y_0^i \leftarrow 0^n$
3: $w_0^i \leftarrow 0^n$
4: **for** $j = 1$ to m_i **do**
5:     $y_j^i = Q_{1,j}(y_{j-1}^i \oplus M_i[j])$
6:     $w_j^i = Q_{2,j}(w_{j-1}^i \oplus M_i[j])$
7: **end for**
8: $T_i \leftarrow y_{m_i}^i \oplus w_{m_i}^i$
9: **return** T_i

Figure 10: Definition of $\text{XMAC1}[\mathcal{Q}](M_i)$

Similarly, $\Pr[\mathcal{I}_2 \text{ has a collision}] \leq 0.5q_2^2/2^n$. Therefore, we can bound the overall probability of bad event occurring as

$$\Pr[\mathcal{B}^{\mathcal{G}(\cdot)} \text{ sets } \mathbf{bad}] \leq \frac{0.5q_1^2}{2^n} + \frac{0.5q_2^2}{2^n} \leq \frac{0.5q^2}{2^n},$$

which concludes the proof. $\qquad\square$

We consider an algorithm $\text{XMAC1}[\mathcal{Q}]$, manipulating messages based on $\mathcal{Q}$. Its definition is presented in Fig. 10. We can see that $\text{XMAC1}[\mathcal{Q}]$ is exactly the same as $\text{XMAC1}[\mathcal{R}]$ since all the internal values rand_i and rand_i' are canceled during the computation. We next consider another algorithm $\text{XMAC1}[\mathcal{G}]$ based on $\mathcal{G}$. Its definition is presented in Fig. 11. It is obtained from $\text{XMAC1}[\mathcal{Q}]$ by replacing $Q_{i,j}$ with $G_{i,j}$, for $i \in \{1,2\}$ and $1 \leq j \leq \ell$. By using the Lemma 4, we have

$$\mathbf{Adv}_{\text{XMAC1}[\mathcal{R}]}^{\text{prf}}(q,\ell,\sigma) = \mathbf{Adv}_{\text{XMAC1}[\mathcal{Q}]}^{\text{prf}}(q,\ell,\sigma) \leq \mathbf{Adv}_{\text{XMAC1}[\mathcal{G}]}^{\text{prf}}(q,\ell,\sigma) + \frac{\sigma^2}{2^n}.$$

To upper bound $\mathbf{Adv}_{\text{XMAC1}[\mathcal{G}]}^{\text{prf}}(q,\ell,\sigma)$, we next resort to the H-coefficient technique [CS14, Pat08], which has been briefly introduced in Sect. A.2.
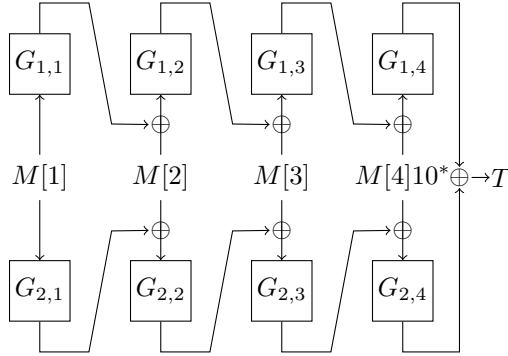
Figure 11: Definition of $\text{XMAC1}[\mathcal{G}]$, $G_{i,j}$ are independent random functions for $i \in \{1,2\}$ and $1 \leq j \leq 4$.

A.4   Analysis of Bad Views

In the real world, the corresponding oracle is $\text{XMAC1}[\mathcal{G}]$ while in the ideal world it is a random function. We note that the adversary $\mathcal{A}$ can make at most q queries to its oracle,

each query being at most ℓ blocks, the total number of blocks of queries being at most σ, and outputs a single bit. Let a view

$$\nu = ((M_1, T_1), \ldots, (M_q, T_q))$$

be a list of queries and corresponding answers. We start by defining bad views and good views.

Definition 1. A bad view is an attainable view $\nu = ((M_1, T_1), \ldots, (M_q, T_q))$ that there exists a collision in ν such that

$$T_i = T_j, \text{ where } 1 \leq i < j \leq q.$$

Otherwise, we call it a good view. We denote Θ_{bad} the set of bad views and Θ_{good} the set of good views.

Now we upper bound the probability to get a bad view in the ideal world.

Lemma 5. *For any integer q, we have*

$$\Pr[X_{\text{id}} \in \Theta_{\text{bad}}] \leq \frac{0.5q^2}{2^n}.$$

Proof. In the ideal world, T_i is simply a random n-bit string and $\Pr[T_i = T_j] = \frac{1}{2^n}$ for any $i \neq j$. Thus,

$$\Pr[X_{\text{id}} \in \Theta_{\text{bad}}] \leq \binom{q}{2} \frac{1}{2^n} \leq \frac{0.5q^2}{2^n}.$$

$\square$

A.5 Analysis of Good Views

We now analyze good views and prove the following lemma.

Lemma 6. *For any good view ν, we have*

$$\frac{\Pr[X_{\text{re}} = \nu]}{\Pr[X_{\text{id}} = \nu]} \geq 1 - \frac{2\sigma q}{2^n}.$$

Proof. Let $\nu = ((M_1, T_1), \ldots, (M_q, T_q))$ be a good view. Since in the ideal world the oracle is a random function, we simply have

$$\Pr[X_{\text{id}} = \nu] = \frac{1}{2^{qn}}. \tag{1}$$

Now we proceed to lower bound the probability of obtaining ν in the real world. The key point is to count the number of functions that induce ν. From the definition, we have

$$\Pr[X_{\text{re}} = \nu] = \frac{\#\text{functions inducing } \nu}{\#\text{total functions}}.$$

For a message $M_i = M_i[1] \| \cdots \| M_i[m_i]$, we denote x_j^i and y_j^i the input and corresponding output of $G_{1,j}$, u_j^i and w_j^i the input and corresponding output of $G_{2,j}$ in XMAC1$[\mathcal{G}]$ for $1 \leq i \leq m_i$. Since our goal is to compute the lower bound of $\Pr[X_{\text{re}} = \nu]$, we can ignore some troublesome functions and merely count the number of ones that induce ν and satisfy the following **condition** to ease the analysis:

$$\text{if } M_i[1] \| \cdots \| M_i[t] \neq M_j[1] \| \cdots \| M_j[t], \text{ then } x_t^i \neq x_t^j \text{ and } u_t^i \neq u_t^j.$$

We denote σ_i the number of messages that have block length at least i. We first compute the probability of $G_{1,1}$ and $G_{2,1}$ satisfying the requirements. We divide these q messages into several groups according to the first block. Messages in each group have the identical first block. After such a classification, we will obtain r groups. Assume the ith group contains q_i messages and denote $M_j^i (1 \le j \le q_i)$ the jth message in ith group, then we have $q_1 + q_2 + \cdots + q_r = \sigma_1$. By abusing notation, in the ith group, we denote x_1^i and y_1^i the input and output of $G_{1,1}$, u_1^i and w_1^i the input and output of $G_{2,1}$. We note that if there exists a message in the ith group outputting a tag T_i after computation of the first block, then there is a relation between y_1^i and w_1^i as $y_1^i \oplus w_1^i$. Note that each group has at most one relation, otherwise this group has a pair of identical messages. Then we count the number of choices y_1^i and w_1^i in ith group for $1 \le i \le r$ in turn, which will be affected by whether there exists a relation in this group or not:

- For the first group,

    - no relation: there are both 2^n possibilities for y_1^1 and w_1^1, thus total $(2^n)^2$;

    - a relation: there are 2^n possibilities for y_1^1 and once y_1^1 is determined so does w_1^1 as $w_1^1 = T_1 \oplus y_1^1$.

- For the 2nd group, once y_1^1 and w_1^1 are fixed,

    - no relation: due to the additional **condition**, we have $y_1^2 \oplus M_j^2[2] \ne y_1^1 \oplus M_i^1[2]$ and $w_1^2 \oplus M_j^2[2] \ne w_1^1 \oplus M_i^1[2]$ for $1 \le i \le q_1$ and $1 \le j \le q_2$. Therefore, there are both at least $2^n - q_1 q_2$ possibilities for y_1^2 and w_1^2, thus total at lest $(2^n - q_1 q_2)^2$.

    - a relation: there are at least $2^n - 2q_1 q_2$ possibilities for y_1^2 as $y_1^2 \oplus M_j^2[2] \ne y_1^1 \oplus M_i^1[2]$ and $y_1^2 \oplus T_2 \oplus M_j^2[2] = w_1^2 \oplus M_j^2[2] \ne w_1^1 \oplus M_i^1[2]$ for $1 \le i \le q_1$ and $1 \le j \le q_2$.

- $\ldots$

- For the k-th group, once $y_1^1, \ldots, y_1^{k-1}$ and $w_1^1, \ldots, w_1^{k-1}$ are fixed,

    - no relation: due to the additional **condition**, we have $y_1^k \oplus M_j^k[2] \ne y_1^t \oplus M_i^t[2]$ and $w_1^k \oplus M_j^k[2] \ne w_1^t \oplus M_i^t[2]$ for $1 \le t \le k-1, 1 \le i \le q_t$ and $1 \le j \le q_k$. Therefore, there are both at least $2^n - q_k(q_1 + q_2 + \cdots + q_{k-1})$ possibilities for y_1^k and w_1^k, thus total at least $(2^n - q_k(q_1 + q_2 + \cdots + q_{k-1}))^2$.

    - a relation: there are at least $2^n - 2q_k(q_1 + q_2 + \cdots + q_{k-1})$ possibilities for y_1^k since $y_1^k \oplus M_j^k[2] \ne y_1^t \oplus M_i^t[2]$ and $y_1^k \oplus T_k \oplus M_j^k[2] = w_1^k \oplus M_j^k[2] \ne w_1^t \oplus M_i^t[2]$ for $1 \le t \le k-1, 1 \le i \le q_t$ and $1 \le j \le q_k$.

Hence, the number of tuples $(y_1^1, w_1^1, \ldots, x_1^r, w_1^r)$ is at least

$$\prod_{i=1}^{r} C^i$$

where for $1 \le i \le r$, either $C^i = (2^n - q_i(q_1 + \cdots + q_{i-1}))^2 \ge (2^n - q_i q)^2 \ge 2^n(2^n - 2q_i q)$ without a relation, or $C^i = 2^n - 2q_i(q_1 + \cdots + q_{i-1}) \ge 2^n - 2q_i q$ with a relation. We denote s the total number of relations among these r groups, then the probability that $G_{1,1}$ and

$G_{2,1}$ meet the requirements is at least

$$\frac{\prod_{i=1}^{r} C^i ((2^n)^{2^n-r})^2}{((2^n)^{2^n})^2} = \prod_{i=1}^{r} \frac{C^i}{2^{2n}}$$

$$\geq \frac{1}{2^{ns}} \prod_{i=1}^{r} (1 - \frac{2q_i q}{2^n})$$

$$\geq \frac{1}{2^{ns}} (1 - \frac{2\sigma_1 q}{2^n}).$$

Then we proceed to analyze the remaining blocks and apply the same analysis to the rest of random functions $G_{1,i}$ and $G_{2,i}$ as they are independent from each other for $1 \leq i \leq \ell$. There are at most ℓ pairs of $(G_{1,i}, G_{2,i})$, so the probability of inducing ν in the real world can be bounded by

$$\Pr[X_{\mathrm{re}} = \nu] \geq \prod_{i=1}^{\ell} \frac{1}{2^{ns}} (1 - \frac{2\sigma_i q}{2^n})$$

$$= \frac{1}{2^{nq}} \prod_{i=1}^{\ell} (1 - \frac{2\sigma_i q}{2^n})$$

$$\geq \frac{1}{2^{nq}} (1 - \frac{2\sigma q}{2^n}). \tag{2}$$

Combining (1) and (2) together, we obtain

$$\frac{\Pr[X_{\mathrm{re}} = \nu]}{\Pr[X_{\mathrm{id}} = \nu]} \geq 1 - \frac{2\sigma q}{2^n}, \tag{3}$$

and this completes the proof of Lemma 6.   □

Following Lemma 3 and using the results of Lemma 5 and Lemma 6, we have

$$\mathbf{Adv}_{\mathrm{XMAC1}[\mathcal{G}]}^{\mathrm{prf}}(q, \ell, \sigma) \leq \frac{2\sigma q}{2^n} + \frac{0.5q^2}{2^n}. \tag{4}$$

Finally we obtain the claimed bound in Theorem 2 as

$$\mathbf{Adv}_{\mathrm{XMAC1}[P]}^{\mathrm{prf}}(q, \ell, \sigma) \leq \frac{\sigma^2}{2^n} + \frac{\sigma^2}{2^n} + \frac{2\sigma q}{2^n} + \frac{0.5q^2}{2^n}$$

$$\leq \frac{2\sigma^2}{2^n} + \frac{2\sigma q}{2^n} + \frac{0.5q^2}{2^n}. \tag{5}$$

We emphasize that this bound is tight up to a constant factor due to the double collision attack as mentioned in Appendix A.1 for XMAC1 with padding scheme 2.