Stronger Security Bounds for OMAC, TMAC and XCBC

Tetsu Iwata Kaoru Kurosawa

Department of Computer and Information Sciences,
Ibaraki University
4–12–1 Nakanarusawa, Hitachi, Ibaraki 316-8511, Japan
{iwata, kurosawa}@cis.ibaraki.ac.jp

April 30, 2003

Abstract. OMAC, TMAC and XCBC are CBC-type MAC schemes which are provably secure for arbitrary message length. In this paper, we present a more tight upper bound on $\mathtt{Adv}^{\mathsf{mac}}$ for each scheme, where $\mathtt{Adv}^{\mathsf{mac}}$ denotes the maximum success (forgery) probability of adversaries. Our bounds are expressed in terms of the *total length* of all queries of an adversary to the MAC generation oracle while the previous bounds are expressed in terms of the *maximum length* of each query. In particular, a significant improvement occurs if the lengths of queries are heavily unbalanced.

Key words: OMAC, TMAC, XCBC, modes of operation, block cipher, provable security.

Contents

1 Introduction

1.1 Background

The CBC MAC [5, 7] is a well-known method to generate a message authentication code (MAC) based on a block cipher E. We denote the CBC MAC value of a message M by

$$\mathrm{CBC}_K(M),$$

where K is the key of E. While Bellare, Kilian, and Rogaway proved that the CBC MAC is secure for fixed length messages [1], it is *not* secure for variable length messages.

Therefore, several variants of CBC MAC have been proposed which are provably secure for variable length messages. They include EMAC, XCBC, TMAC and then OMAC.

EMAC (Encrypted MAC) is obtained by encrypting $\mathrm{CBC}_{K_1}(M)$ by E again with a new key K_2 [2]. That is,

$$\mathrm{EMAC}_{K_1,K_2}(M) = E_{K_2}(\mathrm{CBC}_{K_1}(M)).$$

Petrank and Rackoff proved that EMAC is secure if the message length is a multiple of n, where n is the block length of E [12].

For arbitrary length messages, we can simply append the minimal 10^i to a message M so that the length is a multiple of n. In this method, however, we must append an entire extra block 10^{n-1} if the size of the message is already a multiple of n. This is a "wasting" of one block cipher invocation.

Black and Rogaway next proposed XCBC to solve the above problem [3]. XCBC takes *three* keys: K_1 for E, and K_2 and K_3. In XCBC, we do not append 10^{n-1} if the size of the message is already a multiple of n. Only if this is not the case, we append the minimal 10^i. In order to distinguish them, K_2 or K_3 is XORed before encrypting the last block. XCBC is now described as follows (see Fig. 1).

- If $|M| = mn$ for some $m > 0$, then XCBC computes exactly the same as the CBC MAC, except for XORing an n-bit key K_2 before encrypting the last block.
- Otherwise, 10^i padding ($i = n - |M| - 1 \bmod n$) is appended to M and XCBC computes exactly the same as the CBC MAC for the padded message, except for XORing another n-bit key K_3 before encrypting the last block.

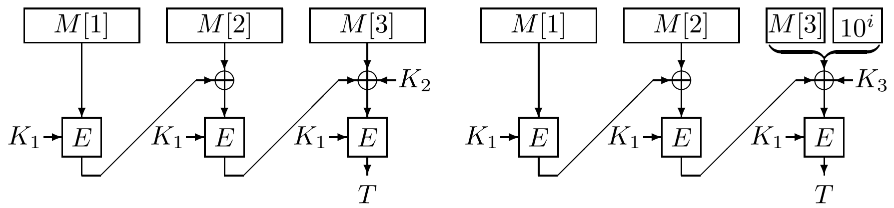

Fig. 1. Illustration of XCBC.

Kurosawa and Iwata then proposed TMAC which requires *two* keys, K_1 and K_2 [9]. TMAC is obtained from XCBC by replacing (K_2, K_3) with $(K_2 \cdot \mathtt{u}, K_2)$, where $\mathtt{u}$ is some non-zero constant and "$\cdot$" denotes multiplication in $\mathrm{GF}(2^n)$.

Finally, Iwata and Kurosawa proposed OMAC which requires only *one* key K of the block cipher E [8]. OMAC is a generic name for OMAC1 and OMAC2. Let $L = E_K(0^n)$. Then

Table 1. Comparison of the key lengths.

	XCBC [3]	TMAC [9]	OMAC [8]
key length	$(k+2n)$ bits	$(k+n)$ bits	k bits

OMAC1 is obtained by replacing (K_2, K_3) with $(L \cdot \mathtt{u}, L \cdot \mathtt{u}^2)$ in XCBC. Similarly, OMAC2 is obtained from XCBC by replacing (K_2, K_3) with $(L \cdot \mathtt{u}, L \cdot \mathtt{u}^{-1})$.

See Table 1 for the comparison of the key lengths, where k denotes the key length of E.

1.2 Our Contribution

XCBC, TMAC and OMAC are all provably secure against chosen message attack. Indeed, the authors showed an upper bound on $\mathtt{Adv}^{\mathsf{mac}}$ for each scheme, where $\mathtt{Adv}^{\mathsf{mac}}$ denotes the maximum success (forgery) probability of adversaries.

In this paper, we present a more tight upper bound on $\mathtt{Adv}^{\mathsf{mac}}$ for each scheme by using a more specific parameter. Consider adversaries who run in time at most t and query at most q messages to the MAC generation oracle.

1. The previous bounds are expressed in terms of the maximum length of each query.
2. Our bounds are expressed in terms of the total length of all queries.

More precisely,

1. Table 2 shows the previous bounds on $\mathtt{Adv}_F^{\mathsf{mac}}(t, q, m)$ which is defined as the maximum forgery probability of adversaries such that each query is at most m blocks, where 1 block is n bits, and
2. Table 3 shows our bounds on $\mathtt{Adv}_F^{\mathsf{mac}}(t, q, \sigma)$ which is defined as the maximum forgery probability of adversaries such that the total length of all queries are at most σ blocks,

where F is XCBC, TMAC or OMAC and n is the block length of the underlying block cipher E. In these tables, $\mathtt{Adv}_E^{\mathsf{prp}}(t', q')$ is the the maximum distinguishing probability between the block cipher E and a randomly chosen permutation, where the maximum is over all adversaries who run in time at most t' and make at most q' queries.

Table 2. Previous security bounds of XCBC, TMAC and OMAC.

Name	Security Bound
XCBC [3, Corollary 2]	$\mathtt{Adv}_{\mathrm{XCBC}}^{\mathsf{mac}}(t, q, m) \le \frac{(4m^2+1)q^2+1}{2^n} + 3 \cdot \mathtt{Adv}_E^{\mathsf{prp}}(t', q')$, where $t' = t + O(mq)$ and $q' = mq$.
TMAC [9, Theorem 5.1]	$\mathtt{Adv}_{\mathrm{TMAC}}^{\mathsf{mac}}(t, q, m) \le \frac{(3m^2+1)q^2+1}{2^n} + \mathtt{Adv}_E^{\mathsf{prp}}(t', q')$, where $t' = t + O(mq)$ and $q' = mq$.
OMAC [8, Theorem 5.1]	$\mathtt{Adv}_{\mathrm{OMAC}}^{\mathsf{mac}}(t, q, m) \le \frac{(5m^2+1)q^2+1}{2^n} + \mathtt{Adv}_E^{\mathsf{prp}}(t', q')$, where $t' = t + O(mq)$ and $q' = mq+1$.

In general, $\sigma \le mq$, where σ is the total block length of all queries, q is the number of queries, and m is the the maximum block length among all queries.

Table 3. Security bounds of XCBC, TMAC and OMAC obtained in this paper.

Name	Security Bound
XCBC	$\mathtt{Adv}^{\mathtt{mac}}_{\mathrm{XCBC}}(t,q,\sigma) \le \frac{3\sigma^2+1}{2^n} + \mathtt{Adv}^{\mathtt{prp}}_{E}(t',q')$, where $t' = t + O(\sigma)$ and $q' = \sigma$.
TMAC	$\mathtt{Adv}^{\mathtt{mac}}_{\mathrm{TMAC}}(t,q,\sigma) \le \frac{3\sigma^2+1}{2^n} + \mathtt{Adv}^{\mathtt{prp}}_{E}(t',q')$, where $t' = t + O(\sigma)$ and $q' = \sigma$.
OMAC	$\mathtt{Adv}^{\mathtt{mac}}_{\mathrm{OMAC}}(t,q,\sigma) \le \frac{4\sigma^2+1}{2^n} + \mathtt{Adv}^{\mathtt{prp}}_{E}(t',q')$, where $t' = t + O(\sigma)$ and $q' = \sigma + 1$.

A significant improvement occurs if all queries are very short (say, 1 block) except for one very long query (m blocks). For example, suppose that $n = 64$ (for example, Triple DES [4]), $m = 2^{16}$ and $q = 2^{16} + 1$. It is easy to see that $\sigma = 2^{16} + 2^{16} = 2^{17}$. In this case, our bounds shown in Table 3 are still meaningful while the previous bounds shown in Table 2 are useless because they become larger than one.

1.3 Our Collision Bound

To show our security bounds, we derive upper bounds on some collision probabilities. For q distinct messages $M^{(1)}, \ldots, M^{(q)}$ such that each $|M^{(i)}|$ is a multiple of n, let

$$\sigma = |M^{(1)}| + \cdots + |M^{(q)}|.$$

For XCBC and TMAC, we consider a collision such that

$$\mathrm{CBC}_P(M^{(i)}) = \mathrm{CBC}_P(M^{(j)})$$

for some $i \neq j$, where CBC_P denotes the CBC MAC with a randomly chosen permutation P as the underlying block cipher E. We then prove that

$$\Pr(1 \le {}^{\exists}i < {}^{\exists}j \le q, \mathrm{CBC}_P(M^{(i)}) = \mathrm{CBC}_P(M^{(j)})) \le \frac{\sigma^2}{2^n}$$

for any $M^{(1)}, \ldots, M^{(q)}$. It is formally stated in Lemma 5.2 and proved in Sec. 5.2.

For OMAC, we consider MOMAC-E, a variant of the CBC MAC, as follows. Let a message be $M = M[1] \circ M[2] \circ \cdots \circ M[m]$, where $|M[1]| = |M[2]| = \cdots = |M[m]| = n$ and $m \ge 2$. Let P_1 and P_2 be two independent randomly chosen permutations. Then

1. Let $Y[1] = P_1(M[1])$
2. For $i = 2, \ldots, m-1$, compute
$$Y[i] = P_2(M[i] \oplus Y[i-1])$$
3. Finally define
$$\text{MOMAC-E}_{P_1,P_2}(M) = M[m] \oplus Y[m-1].$$

We show that

$$\Pr(1 \le {}^{\exists}i < {}^{\exists}j \le q, \text{MOMAC-E}_{P_1,P_2}(M^{(i)}) = \text{MOMAC-E}_{P_1,P_2}(M^{(j)})) \le \frac{(\sigma - q)^2}{2^n}.$$

It is formally stated in Lemma 4.2 and proved in Sec. 4.2.

2 Preliminaries

2.1 Notation

For a set A, $x \stackrel{R}{\leftarrow} A$ means that x is chosen from A uniformly at random. If $a, b \in \{0,1\}^*$ are equal-length strings then $a \oplus b$ is their bitwise XOR. If $a, b \in \{0,1\}^*$ are strings then $a \circ b$ denote their concatenation. For simplicity, we sometimes write ab for $a \circ b$ if there is no confusion.

For an n-bit string $a = a_{n-1} \cdots a_1 a_0 \in \{0,1\}^n$, let $a \ll 1 = a_{n-2} \cdots a_1 a_0 0$ denote the n-bit string which is a left shift of a by 1 bit, while $a \gg 1 = 0 a_{n-1} \cdots a_2 a_1$ denote the n-bit string which is a right shift of a by 1 bit.

If $a \in \{0,1\}^*$ is a string then $|a|$ denotes its length in bits. For any bit string $a \in \{0,1\}^*$ such that $|a| \le n$, we let

$$\mathtt{pad}_n(a) = \begin{cases} a10^{n-|a|-1} & \text{if } |a| < n, \\ a & \text{if } |a| = n. \end{cases} \tag{1}$$

Define $\|a\|_n = \max\{1, \lceil |a|/n \rceil\}$, where the empty string counts as one block. In pseudocode, we write "Partition M into $M[1] \cdots M[m]$" as shorthand for "Let $m = \|M\|_n$, and let $M[1], \ldots, M[m]$ be bit strings such that $M[1] \cdots M[m] = M$ and $|M[i]| = n$ for $1 \le i < m$."

2.2 CBC MAC

A block cipher E is a function

$$E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n,$$

where $\mathcal{K}_E$ is the set of keys and $E(K, \cdot) = E_K(\cdot)$ is a permutation on $\{0,1\}^n$. n is called the block length of E.

The CBC MAC [5, 7] is the simplest and most well-known MAC scheme based on block ciphers E. For a message $M = M[1] \circ M[2] \circ \cdots \circ M[m]$ such that

$$|M[1]| = |M[2]| = \cdots = |M[m]| = n,$$

let $Y[0] = 0^n$ and

$$Y[i] = E_K(M[i] \oplus Y[i-1])$$

for $i = 1, \ldots, m$. Then the CBC MAC of M under key K is defined as

$$\mathrm{CBC}_K(M) = Y[m].$$

Bellare, Kilian, and Rogaway proved that the CBC MAC is secure for fixed length messages [1]. However, it is well known that CBC MAC is *not* secure for variable length messages.

2.3 XCBC, TMAC and OMAC

XCBC, TMAC and OMAC are CBC-type MAC schemes which are provably secure for arbitrary message length.

- Each scheme takes a message $M \in \{0,1\}^*$ and produces a tag in $\{0,1\}^n$.
- Each scheme is defined by using a block cipher $E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$.

Algorithm $\mathrm{XCBC}_{K_1,K_2,K_3}(M)$
$Y[0] \leftarrow 0^n$
Partition M into $M[1] \cdots M[m]$
for $i \leftarrow 1$ **to** $m-1$ **do**
  $X[i] \leftarrow M[i] \oplus Y[i-1]$
  $Y[i] \leftarrow E_{K_1}(X[i])$
$X[m] \leftarrow \mathtt{pad}_n(M[m]) \oplus Y[m-1]$
if $|M[m]| = n$ **then** $X[m] \leftarrow X[m] \oplus K_2$
  else $X[m] \leftarrow X[m] \oplus K_3$
$T \leftarrow E_{K_1}(X[m])$
return T

Fig. 2. Definition of XCBC.

2.3.1 XCBC

XCBC takes three keys $(K_1, K_2, K_3) \in \mathcal{K}_E \times \{0,1\}^n \times \{0,1\}^n$. The algorithm of XCBC is described in Fig. 2 and illustrated in Fig. 1, where $\mathtt{pad}_n(\cdot)$ is defined in (1).

2.3.2 TMAC-family and TMAC

TMAC takes two keys $(K_1, K_2) \in \mathcal{K}_E \times \{0,1\}^n$. In general, TMAC-family is defined by not only a block cipher E but also (1) a universal hash function

$$H : \mathcal{K}_H \times X \to \{0,1\}^n$$

where $\mathcal{K}_H$ is the set of keys and X is the domain and (2) two distinct constants $\mathtt{Cst}_1, \mathtt{Cst}_2 \in X$. They must satisfy the following three conditions for sufficiently small $\epsilon_1, \epsilon_2, \epsilon_3$. (We write $H_K(\cdot)$ for $H(K, \cdot)$.)

1. $\forall y \in \{0,1\}^n$, $\#\{K \in \mathcal{K}_H \mid H_K(\mathtt{Cst}_1) = y\} \le \epsilon_1 \cdot \#\mathcal{K}_H$
2. $\forall y \in \{0,1\}^n$, $\#\{K \in \mathcal{K}_H \mid H_K(\mathtt{Cst}_2) = y\} \le \epsilon_2 \cdot \#\mathcal{K}_H$
3. $\forall y \in \{0,1\}^n$, $\#\{K \in \mathcal{K}_H \mid H_K(\mathtt{Cst}_1) \oplus H_K(\mathtt{Cst}_2) = y\} \le \epsilon_3 \cdot \#\mathcal{K}_H$

The algorithm of TMAC-family is described in Fig. 3 and illustrated in Fig. 4.

TMAC is obtained by letting $\mathcal{K}_H = \{0,1\}^n$, $H_K(x) = K \cdot x$, $\mathtt{Cst}_1 = \mathtt{u}$ and $\mathtt{Cst}_2 = 1$, where "$\cdot$" denotes multiplication over $\mathrm{GF}(2^n)$ (See Appendix A for details). Equivalently, TMAC is obtained by letting

$$H_{K_2}(\mathtt{Cst}_1) = K_2 \cdot \mathtt{u} \text{ and } H_{K_2}(\mathtt{Cst}_2) = K_2.$$

The above three conditions are satisfied with $\epsilon_1 = \epsilon_2 = \epsilon_3 = 2^{-n}$.

2.3.3 OMAC-family, OMAC1 and OMAC2

OMAC is a generic name for OMAC1 and OMAC2, where OMAC1 and OMAC2 take just one key $K \in \mathcal{K}_E$. In general, OMAC-family is defined by not only a block cipher E but also (1) a universal hash function

$$H : \{0,1\}^n \times X \to \{0,1\}^n$$

Algorithm TMAC-family$_{K_1,K_2}(M)$
$Y[0] \leftarrow 0^n$
Partition M into $M[1] \cdots M[m]$
for $i \leftarrow 1$ **to** $m-1$ **do**
    $X[i] \leftarrow M[i] \oplus Y[i-1]$
    $Y[i] \leftarrow E_{K_1}(X[i])$
$X[m] \leftarrow \mathtt{pad}_n(M[m]) \oplus Y[m-1]$
if $|M[m]| = n$ **then** $X[m] \leftarrow X[m] \oplus H_{K_2}(\mathtt{Cst}_1)$
    else $X[m] \leftarrow X[m] \oplus H_{K_2}(\mathtt{Cst}_2)$
$T \leftarrow E_{K_1}(X[m])$
return T

Fig. 3. Definition of TMAC-family.

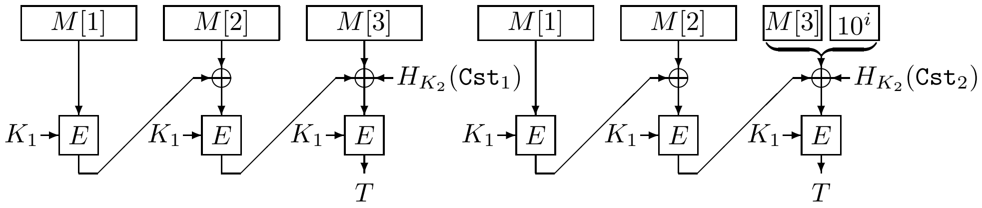

Fig. 4. Illustration of TMAC-family.

where X is the domain, (2) two distinct constants $\mathtt{Cst}_1, \mathtt{Cst}_2 \in X$ and (3) an arbitrary n-bit constant $\mathtt{Cst} \in \{0,1\}^n$. (The set of keys of H is $\{0,1\}^n$.) They must satisfy the following six conditions for sufficiently small $\epsilon_1, \epsilon_2, \ldots, \epsilon_6$.

1. $\forall y \in \{0,1\}^n$, $\#\{L \in \{0,1\}^n \mid H_L(\mathtt{Cst}_1) = y\} \le \epsilon_1 \cdot 2^n$
2. $\forall y \in \{0,1\}^n$, $\#\{L \in \{0,1\}^n \mid H_L(\mathtt{Cst}_2) = y\} \le \epsilon_2 \cdot 2^n$
3. $\forall y \in \{0,1\}^n$, $\#\{L \in \{0,1\}^n \mid H_L(\mathtt{Cst}_1) \oplus H_L(\mathtt{Cst}_2) = y\} \le \epsilon_3 \cdot 2^n$
4. $\forall y \in \{0,1\}^n$, $\#\{L \in \{0,1\}^n \mid H_L(\mathtt{Cst}_1) \oplus L = y\} \le \epsilon_4 \cdot 2^n$
5. $\forall y \in \{0,1\}^n$, $\#\{L \in \{0,1\}^n \mid H_L(\mathtt{Cst}_2) \oplus L = y\} \le \epsilon_5 \cdot 2^n$
6. $\forall y \in \{0,1\}^n$, $\#\{L \in \{0,1\}^n \mid H_L(\mathtt{Cst}_1) \oplus H_L(\mathtt{Cst}_2) \oplus L = y\} \le \epsilon_6 \cdot 2^n$

The algorithm of OMAC-family is described in Fig. 5 and illustrated in Fig. 6.

OMAC1 is obtained by letting $\mathtt{Cst} = 0^n$, $H_L(x) = L \cdot x$, $\mathtt{Cst}_1 = \mathtt{u}$ and $\mathtt{Cst}_2 = \mathtt{u}^2$, where "$\cdot$" denotes multiplication over GF(2^n). Equivalently, OMAC1 is obtained by letting

$$L = E_K(0^n),\ H_L(\mathtt{Cst}_1) = L \cdot \mathtt{u} \text{ and } H_L(\mathtt{Cst}_2) = L \cdot \mathtt{u}^2.$$

OMAC2 is the same as OMAC1 except for $\mathtt{Cst}_2 = \mathtt{u}^{-1}$. Equivalently, OMAC2 is obtained by letting

$$L = E_K(0^n),\ H_L(\mathtt{Cst}_1) = L \cdot \mathtt{u} \text{ and } H_L(\mathtt{Cst}_2) = L \cdot \mathtt{u}^{-1}.$$

The above six conditions are satisfied with $\epsilon_1 = \cdots = \epsilon_6 = 2^{-n}$ for both OMAC1 and OMAC2.

```
Algorithm OMAC-family_K(M)
L ← E_K(Cst)
Y[0] ← 0^n
Partition M into M[1] ··· M[m]
for i ← 1 to m − 1 do
        X[i] ← M[i] ⊕ Y[i − 1]
        Y[i] ← E_K(X[i])
X[m] ← pad_n(M[m]) ⊕ Y[m − 1]
if |M[m]| = n then X[m] ← X[m] ⊕ H_L(Cst_1)
               else X[m] ← X[m] ⊕ H_L(Cst_2)
T ← E_K(X[m])
return T
```

Fig. 5. Definition of OMAC-family.

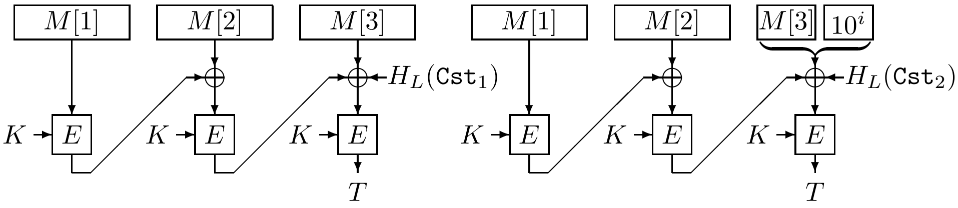

Fig. 6. Illustration of OMAC-family.

3 Stronger Security Bounds

3.1 Definitions of Security

Our definitions follow from [1, 6, 11]. Let $\mathrm{Perm}(n)$ denote the set of all permutations on $\{0,1\}^n$. We say that P is a random permutation if P is randomly chosen from $\mathrm{Perm}(n)$.

The security of a block cipher E can be quantified as $\mathtt{Adv}_E^{\mathsf{prp}}(t,q)$, the maximum advantage that an adversary $\mathcal{A}$ can obtain when trying to distinguish $E_K(\cdot)$ (with a randomly chosen key K) from a random permutation $P(\cdot)$, where the maximum is over all adversaries who run in time at most t, and make at most q queries to an oracle (which is either $E_K(\cdot)$ or $P(\cdot)$). This advantage is defined as follows.

$$\begin{cases} \mathtt{Adv}_E^{\mathsf{prp}}(\mathcal{A}) \stackrel{\mathrm{def}}{=} \left|\Pr(K \stackrel{R}{\leftarrow} \mathcal{K}_E : \mathcal{A}^{E_K(\cdot)} = 1) - \Pr(P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{P(\cdot)} = 1)\right| \\ \mathtt{Adv}_E^{\mathsf{prp}}(t,q) \stackrel{\mathrm{def}}{=} \max_{\mathcal{A}} \{\mathtt{Adv}_E^{\mathsf{prp}}(\mathcal{A})\} \end{cases}$$

We say that a block cipher E is secure if $\mathtt{Adv}_E^{\mathsf{prp}}(t,q)$ is sufficiently small (prp stands for PseudoRandom Permutation).

Similarly, a MAC algorithm is a map $F : \mathcal{K}_F \times \{0,1\}^* \to \{0,1\}^n$, where $\mathcal{K}_F$ is a set of keys and we write $F_K(\cdot)$ for $F(K,\cdot)$. We say that an adversary $\mathcal{A}^{F_K(\cdot)}$ *forges* if $\mathcal{A}$ outputs $(M, F_K(M))$ where $\mathcal{A}$ never queried M to its oracle $F_K(\cdot)$. Then we define the advantage as

$$\begin{cases} \mathtt{Adv}_F^{\mathsf{mac}}(\mathcal{A}) \stackrel{\mathrm{def}}{=} \Pr(K \stackrel{R}{\leftarrow} \mathcal{K}_F : \mathcal{A}^{F_K(\cdot)} \text{ forges}) \\ \mathtt{Adv}_F^{\mathsf{mac}}(t,q,\sigma) \stackrel{\mathrm{def}}{=} \max_{\mathcal{A}} \{\mathtt{Adv}_F^{\mathsf{mac}}(\mathcal{A})\} \end{cases}$$

where the maximum is over all adversaries who run in time at most t, and make at most q queries, having aggregate length of at most σ blocks, where the aggregate length of q queries

$M^{(1)}, \ldots, M^{(q)}$ is $\sigma = \sum_{1 \le i \le q} \|M^{(i)}\|_n$. We say that a MAC algorithm is secure if $\mathtt{Adv}_F^{\mathsf{mac}}(t, q, \sigma)$ is sufficiently small.

Let $\mathrm{Rand}(*, n)$ denote the set of all functions from $\{0,1\}^*$ to $\{0,1\}^n$. This set is given a probability measure by asserting that a random element R of $\mathrm{Rand}(*, n)$ associates to each string $M \in \{0,1\}^*$ a random string $R(M) \in \{0,1\}^n$. Then we define the advantage as

$$\begin{cases} \mathtt{Adv}_F^{\mathsf{viprf}}(\mathcal{A}) \stackrel{\mathrm{def}}{=} \left|\Pr(K \stackrel{R}{\leftarrow} \mathcal{K}_F : \mathcal{A}^{F_K(\cdot)} = 1) - \Pr(R \stackrel{R}{\leftarrow} \mathrm{Rand}(*, n) : \mathcal{A}^{R(\cdot)} = 1)\right| \\ \mathtt{Adv}_F^{\mathsf{viprf}}(t, q, \sigma) \stackrel{\mathrm{def}}{=} \max_{\mathcal{A}} \left\{\mathtt{Adv}_F^{\mathsf{viprf}}(\mathcal{A})\right\} \end{cases}$$

where the maximum is over all adversaries who run in time at most t, make at most q queries, having aggregate length of at most σ blocks. We say that a MAC algorithm is pseudorandom if $\mathtt{Adv}_F^{\mathsf{viprf}}(t, q, \sigma)$ is sufficiently small (viprf stands for Variable-length Input PseudoRandom Function).

Without loss of generality, adversaries are assumed to never ask a query outside the domain of the oracle, and to never repeat a query.

3.2 Theorem Statements

We first prove that OMAC-family, TMAC-family and XCBC are pseudorandom if the underlying block cipher is a random permutation P (information-theoretic result).

Lemma 3.1 (Main Lemma for OMAC-family) *Suppose that H, $\mathtt{Cst}_1$ and $\mathtt{Cst}_2$ satisfy the conditions in Sec. 2.3.3 for some sufficiently small $\epsilon_1, \ldots, \epsilon_6$, and let $\mathtt{Cst}$ be an arbitrarily n-bit constant. Suppose that a random permutation $P \in \mathit{Perm}(n)$ is used in OMAC-family as the underlying block cipher. Let $\mathcal{A}$ be an adversary which asks at most q queries, having aggregate length of at most σ blocks. Assume $\sigma \le 2^n/2$. Then*

$$\begin{aligned} &\Big|\Pr(P \stackrel{R}{\leftarrow} \mathit{Perm}(n) : \mathcal{A}^{\text{OMAC-family}_P(\cdot)} = 1) \\ &\qquad - \Pr(R \stackrel{R}{\leftarrow} \mathit{Rand}(*, n) : \mathcal{A}^{R(\cdot)} = 1)\Big| \le \frac{\sigma^2}{2} \cdot \left(\frac{5}{2^n} + 3\epsilon\right) \quad , \end{aligned} \tag{2}$$

where $\epsilon = \max\{\epsilon_1, \ldots, \epsilon_6\}$.

Lemma 3.2 (Main Lemma for TMAC-family) *Suppose that H, $\mathtt{Cst}_1$ and $\mathtt{Cst}_2$ satisfy the conditions in Sec. 2.3.2 for some sufficiently small $\epsilon_1, \epsilon_2, \epsilon_3$. Suppose that a random permutation $P \in \mathit{Perm}(n)$ is used in TMAC-family as the underlying block cipher. Let $\mathcal{A}$ be an adversary which asks at most q queries, having aggregate length of at most σ blocks. Assume $\sigma \le 2^n/2$. Then*

$$\begin{aligned} &\Big|\Pr(P \stackrel{R}{\leftarrow} \mathit{Perm}(n), K_2 \stackrel{R}{\leftarrow} \mathcal{K}_H : \mathcal{A}^{\text{TMAC-family}_{P,K_2}(\cdot)} = 1) \\ &\qquad - \Pr(R \stackrel{R}{\leftarrow} \mathit{Rand}(*, n) : \mathcal{A}^{R(\cdot)} = 1)\Big| \le \frac{\sigma^2}{2} \cdot \left(\frac{5}{2^n} + \epsilon\right) \quad , \end{aligned} \tag{3}$$

where $\epsilon = \max\{\epsilon_1, \epsilon_2, \epsilon_3\}$.

Lemma 3.3 (Main Lemma for XCBC) *Suppose that a random permutation $P \in \mathit{Perm}(n)$ is used in XCBC as the underlying block cipher. Let $\mathcal{A}$ be an adversary which asks at most q queries, having aggregate length of at most σ blocks. Assume $\sigma \le 2^n/2$. Then*

$$\begin{aligned} &\Big|\Pr(P \stackrel{R}{\leftarrow} \mathit{Perm}(n), K_2, K_3 \stackrel{R}{\leftarrow} \{0,1\}^n : \mathcal{A}^{\mathrm{XCBC}_{P,K_2,K_3}(\cdot)} = 1) \\ &\qquad - \Pr(R \stackrel{R}{\leftarrow} \mathit{Rand}(*, n) : \mathcal{A}^{R(\cdot)} = 1)\Big| \le \frac{3\sigma^2}{2^n} \quad . \end{aligned} \tag{4}$$

Proofs are given in Sec. 4, Sec. 5, and Sec. 6, respectively.

Given the above three lemmas, it is standard to pass to the following complexity-theoretic result (For example, see [1, Section 3.2]). It shows that OMAC, TMAC and XCBC are pseudorandom if the underlying block cipher is secure.

Corollary 3.1 *Let $E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ be the underlying block cipher used in OMAC, TMAC and XCBC. Then*

- $\mathtt{Adv}^{\mathsf{viprf}}_{\mathrm{OMAC}}(t, q, \sigma) \le \dfrac{4\sigma^2}{2^n} + \mathtt{Adv}^{\mathsf{prp}}_E(t', q')$, *where* $t' = t + O(\sigma)$ *and* $q' = \sigma + 1$,
- $\mathtt{Adv}^{\mathsf{viprf}}_{\mathrm{TMAC}}(t, q, \sigma) \le \dfrac{3\sigma^2}{2^n} + \mathtt{Adv}^{\mathsf{prp}}_E(t', q')$, *where* $t' = t + O(\sigma)$ *and* $q' = \sigma$, *and*
- $\mathtt{Adv}^{\mathsf{viprf}}_{\mathrm{XCBC}}(t, q, \sigma) \le \dfrac{3\sigma^2}{2^n} + \mathtt{Adv}^{\mathsf{prp}}_E(t', q')$, *where* $t' = t + O(\sigma)$ *and* $q' = \sigma$.

Finally, we obtain the following theorem in the usual way (For example, see [1, Proposition 2.7]). It shows that OMAC, TMAC and XCBC are secure as MACs if the underlying block cipher is secure.

Theorem 3.1 *Let $E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ be the underlying block cipher used in OMAC, TMAC and XCBC. Then*

- $\mathtt{Adv}^{\mathsf{mac}}_{\mathrm{OMAC}}(t, q, \sigma) \le \dfrac{4\sigma^2 + 1}{2^n} + \mathtt{Adv}^{\mathsf{prp}}_E(t', q')$, *where* $t' = t + O(\sigma)$ *and* $q' = \sigma + 1$,
- $\mathtt{Adv}^{\mathsf{mac}}_{\mathrm{TMAC}}(t, q, \sigma) \le \dfrac{3\sigma^2 + 1}{2^n} + \mathtt{Adv}^{\mathsf{prp}}_E(t', q')$, *where* $t' = t + O(\sigma)$ *and* $q' = \sigma$, *and*
- $\mathtt{Adv}^{\mathsf{mac}}_{\mathrm{XCBC}}(t, q, \sigma) \le \dfrac{3\sigma^2 + 1}{2^n} + \mathtt{Adv}^{\mathsf{prp}}_E(t', q')$, *where* $t' = t + O(\sigma)$ *and* $q' = \sigma$.

4 Proof for OMAC-family

4.1 $Q_1, \ldots, Q_6$ and MOMAC [8]

Let H, $\mathtt{Cst}_1$ and $\mathtt{Cst}_2$ satisfy the conditions in Sec. 2.3.3 for some sufficiently small $\epsilon_1, \ldots, \epsilon_6$, and $\mathtt{Cst}$ be an arbitrarily n-bit constant. For a random permutation $P \in \mathrm{Perm}(n)$ and a random n-bit string $\mathtt{Rnd} \in \{0,1\}^n$, define

$$\begin{cases} Q_1(x) \stackrel{\mathrm{def}}{=} P(x) \oplus \mathtt{Rnd}, & Q_2(x) \stackrel{\mathrm{def}}{=} P(x \oplus \mathtt{Rnd}) \oplus \mathtt{Rnd}, \\ Q_3(x) \stackrel{\mathrm{def}}{=} P(x \oplus \mathtt{Rnd} \oplus H_L(\mathtt{Cst}_1)), & Q_4(x) \stackrel{\mathrm{def}}{=} P(x \oplus \mathtt{Rnd} \oplus H_L(\mathtt{Cst}_2)), \\ Q_5(x) \stackrel{\mathrm{def}}{=} P(x \oplus H_L(\mathtt{Cst}_1)) \text{ and} & Q_6(x) \stackrel{\mathrm{def}}{=} P(x \oplus H_L(\mathtt{Cst}_2)), \end{cases} \tag{5}$$

where $L = P(\mathtt{Cst})$.

The following proposition shows that $Q_1(\cdot)$, $Q_2(\cdot)$, $Q_3(\cdot)$, $Q_4(\cdot)$, $Q_5(\cdot)$, $Q_6(\cdot)$ are indistinguishable from a pair of six independent random permutations $P_1(\cdot)$, $P_2(\cdot)$, $P_3(\cdot)$, $P_4(\cdot)$, $P_5(\cdot)$, $P_6(\cdot)$.

Proposition 4.1 *Let $\mathcal{A}$ be an adversary which asks at most q queries in total. Then*

$$\Big| \Pr(P \stackrel{R}{\leftarrow} \mathit{Perm}(n); \mathtt{Rnd} \stackrel{R}{\leftarrow} \{0,1\}^n : \mathcal{A}^{Q_1(\cdot),\ldots,Q_6(\cdot)} = 1) - \Pr(P_1, \ldots, P_6 \stackrel{R}{\leftarrow} \mathit{Perm}(n) : \mathcal{A}^{P_1(\cdot),\ldots,P_6(\cdot)} = 1) \Big| \le \frac{3q^2}{2} \cdot \left(\frac{1}{2^n} + \epsilon \right) ,$$

where $\epsilon = \max\{\epsilon_1, \ldots, \epsilon_6\}$.

Algorithm $\mathrm{MOMAC}_{P_1,P_2,P_3,P_4,P_5,P_6}(M)$
Partition M into $M[1]\cdots M[m]$
if $m \geq 2$ **then**
  $X[1] \leftarrow M[1]$
  $Y[1] \leftarrow P_1(X[1])$
  for $i \leftarrow 2$ **to** $m-1$ **do**
    $X[i] \leftarrow M[i] \oplus Y[i-1]$
    $Y[i] \leftarrow P_2(X[i])$
  $X[m] \leftarrow \mathtt{pad}_n(M[m]) \oplus Y[m-1]$
  if $|M[m]| = n$ **then** $T \leftarrow P_3(X[m])$
    else $T \leftarrow P_4(X[m])$
if $m = 1$ **then**
  $X[m] \leftarrow \mathtt{pad}_n(M[m])$
  if $|M[m]| = n$ **then** $T \leftarrow P_5(X[m])$
    else $T \leftarrow P_6(X[m])$
return T

Fig. 7. Definition of MOMAC.

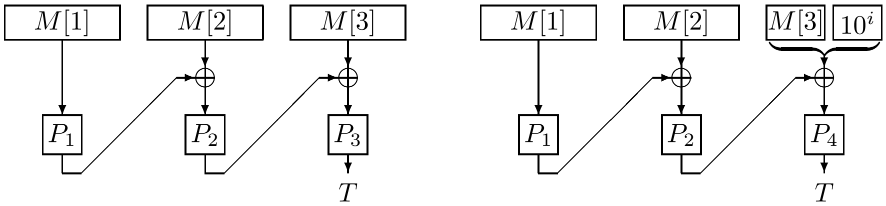

Fig. 8. Illustration of MOMAC for $|M| > n$.

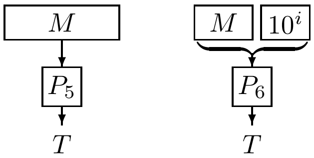

Fig. 9. Illustration of MOMAC for $|M| \leq n$.

A proof is given in [8].

Next, we recall MOMAC (Modified OMAC) [8]. It uses six independent random permutations $P_1, P_2, P_3, P_4, P_5, P_6 \in \mathrm{Perm}(n)$. The algorithm $\mathrm{MOMAC}_{P_1,\ldots,P_6}(\cdot)$ is described in Fig. 7 and illustrated in Fig. 8 and Fig. 9.

4.2 MOMAC is Pseudorandom

We prove that MOMAC is pseudorandom (information-theoretic result).

Lemma 4.1 *Let $\mathcal{A}$ be an adversary which asks at most q queries, having aggregate length of at most σ blocks. Assume $\sigma \leq 2^n/2$. Then*

$$\left|\Pr(P_1,\ldots,P_6 \stackrel{R}{\leftarrow} \mathit{Perm}(n) : \mathcal{A}^{\mathrm{MOMAC}_{P_1,\ldots,P_6}(\cdot)} = 1) - \Pr(R \stackrel{R}{\leftarrow} \mathit{Rand}(*,n) : \mathcal{A}^{R(\cdot)} = 1)\right| \leq \frac{\sigma^2}{2^n} .$$

To prove Lemma 4.1, we first define MOMAC-E (MOMAC without final encryption). It takes a message M such that $|M| = mn$ for some $m \geq 2$. It is obtained from MOMAC by removing the final encryption, that is, it uses two independent random permutations $P_1, P_2 \in \mathrm{Perm}(n)$. More precisely, the algorithm MOMAC-$\mathrm{E}_{P_1,P_2}(\cdot)$ is described in Fig. 10.

```
Algorithm MOMAC-E_{P1,P2}(M)
Partition M into M[1] ··· M[m]
 X[1] ← M[1]
 Y[1] ← P1(X[1])
 for i ← 2 to m − 1 do
        X[i] ← M[i] ⊕ Y[i − 1]
        Y[i] ← P2(X[i])
 X[m] ← M[m] ⊕ Y[m − 1]
return X[m]
```

Fig. 10. Definition of MOMAC-E. Note that $|M| = mn$ for some $m \geq 2$.

We first show the following lemma.

Lemma 4.2 (MOMAC-E Collision Bound) *Let q, $m_1, \ldots, m_q$ and σ be integers such that $m_i \geq 2$, $\sigma = m_1 + \cdots + m_q$, and $\sigma \leq 2^n/2$. Let $M^{(1)}, \ldots, M^{(q)}$ be fixed and distinct bit strings such that $|M^{(i)}| = m_i n$. Then the probability of collision,*

$$\Pr(P_1, P_2 \xleftarrow{R} \mathit{Perm}(n) : 1 \leq {}^{\exists}i < {}^{\exists}j \leq q, \mathit{MOMAC\text{-}E}_{P_1,P_2}(M^{(i)}) = \mathit{MOMAC\text{-}E}_{P_1,P_2}(M^{(j)}))$$

is at most $\frac{(\sigma-q)^2}{2^n}$.

Proof. We view the computation of MOMAC-$\mathrm{E}_{P_1,P_2}(M^{(i)})$ as playing the game given in Fig. 11.

In Fig. 11, $M^{(i)}[1] \cdots M^{(i)}[m_i]$ is a partition of $M^{(i)}$. We initially set each range point of P_1 and P_2 as `undefined`. The notation $\mathrm{Domain}(P_i)$ denotes the set of points x where $P_i(x)$ is no longer `undefined`. We use $\mathrm{Range}(P_i)$ to denote the set of points $P_i(x)$ which are no longer `undefined`. We use $\overline{\mathrm{Range}}(P_i)$ to denote $\{0,1\}^n \setminus \mathrm{Range}(P_i)$.

During the game, the $X^{(i)}[j]$ are those values produced after XORing with the current message block $M^{(i)}[j]$, $Y^{(i)}[1]$ values are $P_1(X^{(i)}[1])$ and, for $j \geq 2$, $Y^{(i)}[j]$ values are $P_2(X^{(i)}[j])$. The game has two parts: computation of $X^{(1)}[2], \ldots, X^{(q)}[2]$ (line 11–23) and computation of $X^{(1)}[m_1], \ldots, X^{(q)}[m_q]$ (line 31–45).

We examine the probability that P_1 and P_2 cause a collision, which will occur in our game if and only if $X^{(i)}[m_i] = X^{(j)}[m_j]$ for some $1 \leq i < j \leq q$. This condition will set $\mathtt{bad}_1$ or $\mathtt{bad}_2$ to `true`. However, we set $\mathtt{bad}_i$ to `true` in many other cases in order to simplify the analysis.

The idea behind the variable $\mathtt{bad}_i$ is as follows: throughout the game (line 13 and 35), we randomly choose a range value for P_1 and P_2 at some `undefined` domain point. Since P_1 and P_2 have not yet been determined at this point, the choice of our range value will be an independent uniform selection: there is no dependence on any prior choice. If the range value for P_i were already determined by some earlier choice, the analysis would become more involved. We avoid the latter condition by setting $\mathtt{bad}_i$ to `true` whenever such interdependencies are detected.

The detection mechanism works as follows: throughout the processing of $M^{(1)}, \ldots, M^{(q)}$, we will require P_1 be evaluated at q domain point $X^{(1)}[1], \ldots, X^{(q)}[1]$ and P_2 be evaluated at $\sigma - q$ domain point $X^{(1)}[2], \ldots, X^{(1)}[m_1], \ldots, X^{(q)}[2], \ldots, X^{(q)}[m_q]$ (ignoring duplications due to any common prefix of $M^{(1)}, \ldots, M^{(q)}$), we can rest assured that we are free to assign their

Initialization:

for $i \leftarrow 1$ **to** q **do** $X^{(i)}[1] \leftarrow M^{(i)}[1]$;
for all $x \in \{0,1\}^n$ **do** $P_1(x), P_2(x) \leftarrow$ `undefined`;
$\mathtt{bad}_1, \mathtt{bad}_2 \leftarrow$ `false`; $BAD \leftarrow \emptyset$;

Computation of $X^{(1)}[2], \ldots, X^{(q)}[2]$**:**

for $i \leftarrow 1$ **to** q **do**
  if $X^{(i)}[1] \notin \mathrm{Domain}(P_1)$ **then**
    $Y^{(i)}[1] \stackrel{R}{\leftarrow} \overline{\mathrm{Range}}(P_1)$;
    $P_1(X^{(i)}[1]) \leftarrow Y^{(i)}[1]$;
    $X^{(i)}[2] \leftarrow Y^{(i)}[1] \oplus M^{(i)}[2]$;
    $BAD' \leftarrow \{X^{(i)}[2]\}$;
    $Index \leftarrow \{k \mid i+1 \le k \le q \text{ and } X^{(i)}[1] = X^{(k)}[1]\}$;
    for all $k \in Index$ **do**
      $Y^{(k)}[1] \leftarrow Y^{(i)}[1]$;
      $X^{(k)}[2] \leftarrow Y^{(k)}[1] \oplus M^{(k)}[2]$;
      $BAD' \leftarrow BAD' \cup \{X^{(k)}[2]\}$;
    if $BAD' \cap BAD \neq \emptyset$ **then** $\mathtt{bad}_1 \leftarrow$ `true`;
    else $BAD \leftarrow BAD' \cup BAD$;

Computation of $X^{(1)}[m_1], \ldots, X^{(q)}[m_q]$**:**

for $j \leftarrow 2$ **to** σ **do**
  for $i \leftarrow 1$ **to** q **do**
    if $j < m_i$ **then**
      if $X^{(i)}[j] \notin \mathrm{Domain}(P_2)$ **then**
        $Y^{(i)}[j] \stackrel{R}{\leftarrow} \overline{\mathrm{Range}}(P_2)$;
        $P_2(X^{(i)}[j]) \leftarrow Y^{(i)}[j]$;
        $X^{(i)}[j+1] \leftarrow Y^{(i)}[j] \oplus M^{(i)}[j+1]$;
        $BAD' \leftarrow \{X^{(i)}[j+1]\}$;
        $Index \leftarrow \{k \mid i+1 \le k \le q,\ j < m_k \text{ and } X^{(i)}[j] = X^{(k)}[j]\}$;
        for all $k \in Index$ **do**
          $Y^{(k)}[j] \leftarrow Y^{(i)}[j]$;
          $X^{(k)}[j+1] \leftarrow Y^{(k)}[j] \oplus M^{(k)}[j+1]$;
          $BAD' \leftarrow BAD' \cup \{X^{(k)}[j+1]\}$;
        if $BAD' \cap BAD \neq \emptyset$ **then** $\mathtt{bad}_2 \leftarrow$ `true`;
        else $BAD \leftarrow BAD' \cup BAD$;

Fig. 11. Game used in the proof of Lemma 4.2.

corresponding range points without constraint. We maintain a set BAD to track which domain points of P_2 have already been determined. Next we begin randomly choosing range points for $X^{(i)}[j]$; if any such choice leads to a value already contained in BAD, we set $\mathtt{bad}_i$ to $\mathtt{true}$. Note that the choice of $Y^{(i)}[j]$ for $X^{(i)}[j]$ may automatically determines some other $Y^{(k)}[j]$ for $X^{(k)}[j]$ due to common prefix of $M^{(1)}, \ldots, M^{(q)}$. We maintain sets *Index* and BAD' to track such points.

We now bound the probability of the event that $\mathtt{bad}_1 \leftarrow \mathtt{true}$ and $\mathtt{bad}_2 \leftarrow \mathtt{true}$ by analyzing our game.

Bounding the probability of $\mathtt{bad}_1 \leftarrow \mathtt{true}$**.** In line 22, it is required that some $Y^{(i)}[1]$ was selected in line 13 such that $Y^{(i)}[1] \oplus M^{(i)}[2] \in BAD$, or $Y^{(i)}[1] \oplus M^{(k)}[2] \in BAD$ for some $k \in \mathit{Index}$. The set BAD begins with the empty set and then grows by the number of points in BAD' with each random choice of $Y^{(i)}[1]$. Now, suppose that for the t-th process of line 13, the corresponding BAD' after line 21 has l_t points, assuming that $\mathtt{bad}_1$ is $\mathtt{false}$ for the first $t-1$ process of line 13. Define

$$V(t) \stackrel{\text{def}}{=} \Pr_{\text{line 13}}(\mathtt{bad}_1 \leftarrow \mathtt{true} \text{ at the } t\text{-th choice of } Y^{(i)}[1] \mid \mathtt{bad}_1 \text{ is } \mathtt{false} \text{ before choosing } Y^{(i)}[1]) ,$$

where $\Pr_{\text{line 13}}(\cdot)$ shows that the probability is taken over the random choice in line 13. Then we have

$$V(t) = \frac{(l_1 + \cdots + l_{t-1})l_t}{2^n - (t-1)} ,$$

since P_1 has $2^n - (t-1)$ $\mathtt{undefined}$ domain points, BAD has $(l_1 + \cdots + l_{t-1})$ points, and BAD' has l_t points.

Also, suppose that line 11–23 terminates after s process of line 13. Then we have

$$\Pr_{\text{line 13}}(\mathtt{bad}_1 \leftarrow \mathtt{true}) \le \sum_{1 \le t \le s} V(t) = \sum_{1 \le t \le s} \frac{(l_1 + \cdots + l_{t-1})l_t}{2^n - (t-1)} .$$

Now we can bound the above by

$$\sum_{1 \le t \le s} \frac{(l_1 + \cdots + l_{t-1})l_t}{2^n - (t-1)} \le \frac{2}{2^n} \sum_{1 \le t \le s} (l_1 + \cdots + l_{t-1})l_t = \frac{2}{2^n} \cdot \frac{{l'_0}^2 - l_1^2 - \cdots - l_s^2}{2} \le \frac{{l'_0}^2}{2^n} ,$$

where $l'_0 \stackrel{\text{def}}{=} l_1 + \cdots + l_s$. The first inequality follows since s is at most q, which is at most $2^n/2$.

Bounding the probability of $\mathtt{bad}_2 \leftarrow \mathtt{true}$**.** Next, in line 44, it is required that some $Y^{(i)}[j]$ was selected in line 35 such that $Y^{(i)}[j] \oplus M^{(i)}[j+1] \in BAD$, or $Y^{(i)}[j] \oplus M^{(k)}[j+1] \in BAD$ for some $k \in \mathit{Index}$. The set BAD begins with l'_0 points. It grows by the number of points in BAD' with each random choice of $Y^{(i)}[j]$. Now, suppose that for the t'-th process of line 35, the corresponding BAD' after line 43 has $l'_{t'}$ points, assuming that $\mathtt{bad}_2$ is $\mathtt{false}$ for the first $t'-1$ process of line 35. Define

$$V'(t') \stackrel{\text{def}}{=} \Pr_{\text{line 35}}(\mathtt{bad}_2 \leftarrow \mathtt{true} \text{ at the } t'\text{-th choice of } Y^{(i)}[j] \mid \mathtt{bad}_2 \text{ is } \mathtt{false} \text{ before choosing } Y^{(i)}[j]) ,$$

where $\Pr_{\text{line 35}}(\cdot)$ shows that the probability is taken over the random choice in line 35. Then we have

$$V'(t') = \frac{(l'_0 + l'_1 + \cdots + l'_{t'-1})l'_{t'}}{2^n - (t'-1)} ,$$

since P_2 has $2^n - (t'-1)$ `undefined` domain points, BAD has $(l'_0 + l'_1 + \cdots + l'_{t'-1})$ points, and BAD' has $l'_{t'}$ points.

Also, suppose that the game terminates after s' process of line 35. Then we have

$$\Pr_{\text{line 35}}(\mathtt{bad}_2 \leftarrow \mathtt{true}) \leq \sum_{1 \leq t' \leq s'} V'(t') = \sum_{1 \leq t' \leq s'} \frac{(l'_0 + l'_1 + \cdots + l'_{t'-1})l'_{t'}}{2^n - (t'-1)} \ .$$

Now we can bound the above by

$$\sum_{1 \leq t' \leq s'} \frac{(l'_0 + l'_1 + \cdots + l'_{t'-1})l'_{t'}}{2^n - (t'-1)} \leq \frac{2}{2^n} \sum_{1 \leq t' \leq s'} (l'_0 + l'_1 + \cdots + l'_{t-1})l'_{t'} \leq \frac{(\sigma - q)^2 - {l'_0}^2}{2^n} \ ,$$

where the first inequality follows since s' is at most σ, which is at most $2^n/2$, and the second inequality follows since $\sigma - q \geq l'_0 + l'_1 + \cdots + l'_{s'}$ and

$$\sum_{1 \leq t' \leq s'} (l'_0 + l'_1 + \cdots + l'_{t'-1})l'_{t'} \leq \frac{(\sigma - q)^2 - {l'_0}^2 - {l'_1}^2 - \cdots - {l'_{s'}}^2}{2} \leq \frac{(\sigma - q)^2 - {l'_0}^2}{2} \ .$$

Completing the Proof. Finally, we obtain the stated bound since

$$\Pr_{\text{line 13}}(\mathtt{bad}_1 \leftarrow \mathtt{true}) + \Pr_{\text{line 35}}(\mathtt{bad}_2 \leftarrow \mathtt{true}) \leq \frac{{l'_0}^2}{2^n} + \frac{(\sigma - q)^2 - {l'_0}^2}{2^n} = \frac{(\sigma - q)^2}{2^n} \ .$$

Q.E.D.

We next consider the following four sets.

$$\begin{cases} D_1 \stackrel{\text{def}}{=} \{M \mid M \in \{0,1\}^*,\, n < |M| \text{ and } |M| \text{ is a multiple of } n\} \\ D_2 \stackrel{\text{def}}{=} \{M \mid M \in \{0,1\}^*,\, n < |M| \text{ and } |M| \text{ is not a multiple of } n\} \\ D_3 \stackrel{\text{def}}{=} \{M \mid M \in \{0,1\}^* \text{ and } |M| = n\} \\ D_4 \stackrel{\text{def}}{=} \{M \mid M \in \{0,1\}^* \text{ and } |M| < n\} \end{cases}$$

We show the following lemma.

Lemma 4.3 *Let q_1, q_2, q_3, q_4 be four non-negative integers. For $1 \leq i \leq 4$, let $M_i^{(1)}, \ldots, M_i^{(q_i)}$ be fixed bit strings such that $M_i^{(j)} \in D_i$ for $1 \leq j \leq q_i$ and $\{M_i^{(1)}, \ldots, M_i^{(q_i)}\}$ are distinct. Similarly, for $1 \leq i \leq 4$, let $T_i^{(1)}, \ldots, T_i^{(q_i)}$ be fixed n-bit strings such that $\{T_i^{(1)}, \ldots, T_i^{(q_i)}\}$ are distinct. Then the number of $P_1, \ldots, P_6 \in Perm(n)$ such that*

$$\begin{cases} MOMAC_{P_1,\ldots,P_6}(M_1^{(i)}) = T_1^{(i)} \text{ for } 1 \leq {}^\forall i \leq q_1, \\ MOMAC_{P_1,\ldots,P_6}(M_2^{(i)}) = T_2^{(i)} \text{ for } 1 \leq {}^\forall i \leq q_2, \\ MOMAC_{P_1,\ldots,P_6}(M_3^{(i)}) = T_3^{(i)} \text{ for } 1 \leq {}^\forall i \leq q_3 \text{ and} \\ MOMAC_{P_1,\ldots,P_6}(M_4^{(i)}) = T_4^{(i)} \text{ for } 1 \leq {}^\forall i \leq q_4 \end{cases} \qquad (6)$$

is at least $\{(2^n)!\}^6 \left(1 - \frac{(\sigma-q)^2}{2^n}\right) \cdot \frac{1}{2^{qn}}$, where $q = q_1 + \cdots + q_4$, $\sigma_i = \sum_{1 \leq j \leq q_i} \|M_i^{(j)}\|_n$ and $\sigma = \sigma_1 + \cdots + \sigma_4$.

Proof. We first consider $M_1^{(1)}, \ldots, M_1^{(q_1)}$. The number of (P_1, P_2) such that

$$\text{MOMAC-E}_{P_1,P_2}(M_1^{(i)}) = \text{MOMAC-E}_{P_1,P_2}(M_1^{(j)}) \text{ for } 1 \le {}^{\exists}i < {}^{\exists}j \le q_1$$

is at most $\{(2^n)!\}^2 \cdot \frac{(\sigma_1 - q_1)^2}{2^n}$ from Lemma 4.2.

We next consider $M_2^{(1)}, \ldots, M_2^{(q_2)}$. Let $M_2'^{(i)}$ denote the padded message of $M_2^{(i)}$. Then the number of (P_1, P_2) such that

$$\text{MOMAC-E}_{P_1,P_2}(M_2'^{(i)}) = \text{MOMAC-E}_{P_1,P_2}(M_2'^{(j)}) \text{ for } 1 \le {}^{\exists}i < {}^{\exists}j \le q_2$$

is at most $\{(2^n)!\}^2 \cdot \frac{(\sigma_2 - q_2)^2}{2^n}$ from Lemma 4.2.

Therefore, we have at least

$$\{(2^n)!\}^2 \left(1 - \frac{(\sigma_1 - q_1)^2}{2^n} - \frac{(\sigma_2 - q_2)^2}{2^n}\right)$$

choice of (P_1, P_2) such that

$$\begin{cases} \text{MOMAC-E}_{P_1,P_2}(M_1^{(i)}) \ne \text{MOMAC-E}_{P_1,P_2}(M_1^{(j)}) \text{ for } 1 \le {}^{\forall}i < {}^{\forall}j \le q_1 \text{ and} \\ \text{MOMAC-E}_{P_1,P_2}(M_2'^{(i)}) \ne \text{MOMAC-E}_{P_1,P_2}(M_2'^{(j)}) \text{ for } 1 \le {}^{\forall}i < {}^{\forall}j \le q_2 \end{cases} \tag{7}$$

We fix any (P_1, P_2) which satisfies (7).

Now P_1 and P_2 are fixed in such a way that the inputs to P_3 are distinct and the inputs to P_4 are distinct. Also, the corresponding outputs $\{T_1^{(1)}, \ldots, T_1^{(q_1)}\}$ are distinct, and $\{T_2^{(1)}, \ldots, T_2^{(q_2)}\}$ are distinct. We know that the inputs to P_5 are distinct, and the corresponding outputs $\{T_3^{(1)}, \ldots, T_3^{(q_3)}\}$ are distinct. Also, the inputs to P_6 are distinct, and and the corresponding outputs $\{T_4^{(1)}, \ldots, T_4^{(q_4)}\}$ are distinct. Therefore, we have at least

$$\{(2^n)!\}^2 \left(1 - \frac{(\sigma_1 - q_1)^2}{2^n} - \frac{(\sigma_2 - q_2)^2}{2^n}\right) \cdot (2^n - q_1)! \cdot (2^n - q_2)! \cdot (2^n - q_3)! \cdot (2^n - q_4)!$$

choice of $P_1, \ldots, P_6$ which satisfies (6). This bound is at least $\{(2^n)!\}^6 \left(1 - \frac{(\sigma - q)^2}{2^n}\right) \cdot \frac{1}{2^{qn}}$ since $(\sigma - q)^2 \ge (\sigma_1 - q_1)^2 + (\sigma_2 - q_2)^2$ and $(2^n - q_i)! \ge \frac{(2^n)!}{2^{q_i n}}$.

This concludes the proof of the lemma. Q.E.D.

We now prove Lemma 4.1.

Proof (of Lemma 4.1). Let $\mathcal{O}$ be either $\text{MOMAC}_{P_1,\ldots,P_6}$ or R. Since $\mathcal{A}$ is computationally unbounded, there is no loss of generality to assume that $\mathcal{A}$ is deterministic.

Now for the query $\mathcal{A}$ makes to the oracle $\mathcal{O}$, define the query-answer pair $(M_j^{(i)}, T_j^{(i)}) \in D_j \times \{0,1\}^n$, where $\mathcal{A}$'s i-th query in D_j was $M_j^{(i)} \in D_j$ and the answer it got was $T_j^{(i)} \in \{0,1\}^n$.

Suppose that we run $\mathcal{A}$ with the oracle. For this run, assume that $\mathcal{A}$ made q_j queries in D_j, where $1 \le j \le 4$ and $q_1 + \cdots + q_4 = q$. Also, for $1 \le i \le 4$, let $\sigma_i = \sum_{1 \le j \le q_i} \|M_i^{(j)}\|_n$ (therefore, $q_3 = \sigma_3$ and $q_4 = \sigma_4$). For this run, we define view v of $\mathcal{A}$ as

$$\begin{aligned} v \stackrel{\text{def}}{=} \langle &(T_1^{(1)}, \ldots, T_1^{(q_1)}), (T_2^{(1)}, \ldots, T_2^{(q_2)}), \\ &(T_3^{(1)}, \ldots, T_3^{(q_3)}), (T_4^{(1)}, \ldots, T_4^{(q_4)}) \rangle \ . \end{aligned} \tag{8}$$

Since $\mathcal{A}$ is deterministic, the i-th query $\mathcal{A}$ makes is fully determined by the first $i-1$ query-answer pairs. This implies that if we fix some qn-bit string V and return the i-th n-bit block as the answer for the i-th query $\mathcal{A}$ makes (instead of the oracle), then

- $\mathcal{A}$'s queries are uniquely determined,
- $q_1, \ldots, q_4$ are uniquely determined,
- $\sigma_1, \ldots, \sigma_4$ are uniquely determined,
- the parsing of V into the format defined in (8) is uniquely determined, and
- the final output of $\mathcal{A}$ (0 or 1) is uniquely determined.

Let $\boldsymbol{V}_{one}$ be a set of all qn-bit strings V such that $\mathcal{A}$ outputs 1. We let $N_{one} \stackrel{\text{def}}{=} \#\boldsymbol{V}_{one}$. Also, let $\boldsymbol{V}_{good}$ be a set of all qn-bit strings V such that:

For $1 \leq {}^\forall i < {}^\forall j \leq q$, the i-th n-bit block of $V \neq$ the j-th n-bit block of V.

Note that if $V \in \boldsymbol{V}_{good}$, then the corresponding parsing v of V satisfies that: $\{T_1^{(1)}, \ldots, T_1^{(q_1)}\}$ are distinct, $\{T_2^{(1)}, \ldots, T_2^{(q_2)}\}$ are distinct, $\{T_3^{(1)}, \ldots, T_3^{(q_3)}\}$ are distinct and $\{T_4^{(1)}, \ldots, T_4^{(q_4)}\}$ are distinct. Now observe that the number of V which is *not* in the set $\boldsymbol{V}_{good}$ is at most $\binom{q}{2}\frac{2^{qn}}{2^n}$. Therefore, we have

$$\#\{V \mid V \in (\boldsymbol{V}_{one} \cap \boldsymbol{V}_{good})\} \geq N_{one} - \binom{q}{2}\frac{2^{qn}}{2^n} \ . \tag{9}$$

Evaluation of p_{rand}. We first evaluate

$$p_{rand} \stackrel{\text{def}}{=} \Pr(R \stackrel{R}{\leftarrow} \mathrm{Rand}(*, n) : \mathcal{A}^{R(\cdot)} = 1) \ .$$

Then it is not hard to see

$$p_{rand} = \sum_{V \in \boldsymbol{V}_{one}} \frac{1}{2^{qn}} = \frac{N_{one}}{2^{qn}} \ .$$

Evaluation of p_{real}. We next evaluate

$$\begin{aligned} p_{real} &\stackrel{\text{def}}{=} \Pr(P_1, \ldots, P_6 \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\mathrm{MOMAC}_{P_1,\ldots,P_6}(\cdot)} = 1) \\ &= \frac{\#\{(P_1, \ldots, P_6) \mid \mathcal{A}^{\mathrm{MOMAC}_{P_1,\ldots,P_6}(\cdot)} = 1\}}{\{(2^n)!\}^6} \ . \end{aligned}$$

Then from Lemma 4.3, we have

$$\begin{aligned} p_{real} &\geq \sum_{V \in (\boldsymbol{V}_{one} \cap \boldsymbol{V}_{good})} \frac{\#\{(P_1, \ldots, P_6) \mid (P_1, \ldots, P_6) \text{ satisfying } (6)\}}{\{(2^n)!\}^6} \\ &\geq \sum_{V \in (\boldsymbol{V}_{one} \cap \boldsymbol{V}_{good})} \left(1 - \frac{(\sigma - q)^2}{2^n}\right) \cdot \frac{1}{2^{qn}} \ . \end{aligned}$$

Completing the Proof. From (9) we have

$$\begin{aligned}
p_{real} &\geq \left(N_{one} - \binom{q}{2}\frac{2^{qn}}{2^n}\right)\cdot\left(1-\frac{(\sigma-q)^2}{2^n}\right)\cdot\frac{1}{2^{qn}} \\
&= \left(p_{rand} - \binom{q}{2}\frac{1}{2^n}\right)\cdot\left(1-\frac{(\sigma-q)^2}{2^n}\right) \\
&\geq p_{rand} - \binom{q}{2}\frac{1}{2^n} - \frac{(\sigma-q)^2}{2^n} \\
&\geq p_{rand} - \frac{q^2+(\sigma-q)^2}{2^n} \\
&\geq p_{rand} - \frac{\sigma^2}{2^n} \ . \qquad (10)
\end{aligned}$$

Applying the same argument to $1-p_{real}$ and $1-p_{rand}$ yields that

$$1-p_{real} \geq 1-p_{rand} - \frac{\sigma^2}{2^n} \ . \qquad (11)$$

Finally, (10) and (11) give $|p_{real}-p_{rand}| \leq \frac{\sigma^2}{2^n}$. Q.E.D.

4.3 From MOMAC to OMAC-family

The next lemma shows that OMAC-family$_P(\cdot)$ and MOMAC$_{P_1,\ldots,P_6}(\cdot)$ are indistinguishable.

Lemma 4.4 *Let $\mathcal{A}$ be an adversary which asks at most q queries, having aggregate length of at most σ blocks. Assume $\sigma \leq 2^n/2$. Then*

$$\Big|\Pr(P \stackrel{R}{\leftarrow} Perm(n) : \mathcal{A}^{\text{OMAC-family}_P(\cdot)} = 1) \\ - \Pr(P_1,\ldots,P_6 \stackrel{R}{\leftarrow} Perm(n) : \mathcal{A}^{\text{MOMAC}_{P_1,\ldots,P_6}(\cdot)} = 1)\Big| \leq \frac{3\sigma^2}{2}\cdot\left(\frac{1}{2^n}+\epsilon\right) \ .$$

Proof. We prove through a contradiction argument. Suppose that there exists an adversary $\mathcal{A}$ such that

$$\Big|\Pr(P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\text{OMAC-family}_P(\cdot)} = 1) \\ - \Pr(P_1,\ldots,P_6 \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\text{MOMAC}_{P_1,\ldots,P_6}(\cdot)} = 1)\Big| > \frac{3\sigma^2}{2}\cdot\left(\frac{1}{2^n}+\epsilon\right) \ .$$

By using $\mathcal{A}$, we show a construction of an adversary $\mathcal{B}_\mathcal{A}$ such that:

- $\mathcal{B}_\mathcal{A}$ asks at most σ queries, and
- $\Big|\Pr(P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{B}_\mathcal{A}^{Q_1(\cdot),\ldots,Q_6(\cdot)} = 1) \\ - \Pr(P_1,\ldots,P_6 \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{B}_\mathcal{A}^{P_1(\cdot),\ldots,P_6(\cdot)} = 1)\Big| > \frac{3\sigma^2}{2}\cdot\left(\frac{1}{2^n}+\epsilon\right),$

which contradicts Proposition 4.1.

Let $\mathcal{O}_1(\cdot),\ldots,\mathcal{O}_6(\cdot)$ be $\mathcal{B}_\mathcal{A}$'s oracles. The construction of $\mathcal{B}_\mathcal{A}$ is given in Fig. 12.

When $\mathcal{A}$ asks $M^{(r)}$, then $\mathcal{B}_\mathcal{A}$ computes $T^{(r)} = \mathrm{MOMAC}_{\mathcal{O}_1,\ldots,\mathcal{O}_6}(M^{(r)})$ as if the underlying random permutations are $\mathcal{O}_1,\ldots,\mathcal{O}_6$, and returns $T^{(r)}$. When $\mathcal{A}$ halts and outputs b, then $\mathcal{B}_\mathcal{A}$ outputs b.

Now we see that:

Algorithm $\mathcal{B}_{\mathcal{A}}^{\mathcal{O}_1,\ldots,\mathcal{O}_6}$

When $\mathcal{A}$ **asks its** r**-th query** $M^{(r)}$**:**

  $T^{(r)} \leftarrow \mathrm{MOMAC}_{\mathcal{O}_1,\ldots,\mathcal{O}_6}(M^{(r)})$

  return $T^{(r)}$

When $\mathcal{A}$ **halts and outputs** b**:**

  output b

Fig. 12. Algorithm $\mathcal{B}_{\mathcal{A}}$. Note that for $1 \le i \le 6$, $\mathcal{O}_i$ is either P_i or Q_i

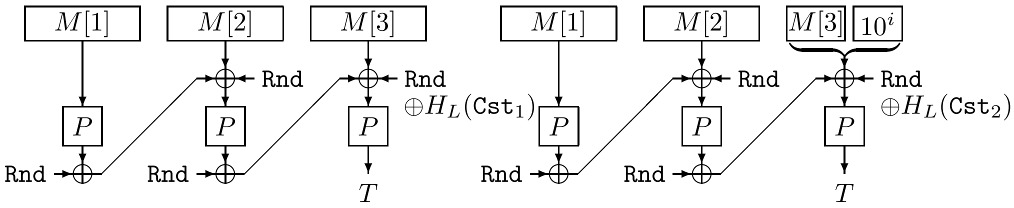

Fig. 13. Computation of $\mathcal{B}_{\mathcal{A}}$ when $\mathcal{O}_i = Q_i$ for $1 \le i \le 6$, and $|M| > n$.

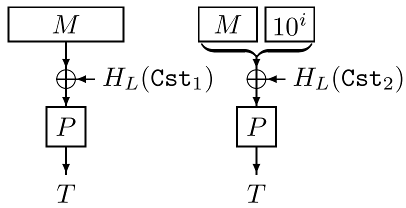

Fig. 14. Computation of $\mathcal{B}_{\mathcal{A}}$ when $\mathcal{O}_i = Q_i$ for $1 \le i \le 6$, and $|M| \le n$.

- $\mathcal{B}_{\mathcal{A}}$ asks at most σ queries to its oracles, since $\mathcal{A}$ asks at most q queries having aggregate length of at most σ blocks.
- $\Pr(P_1, \ldots, P_6 \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{B}_{\mathcal{A}}^{P_1(\cdot),\ldots,P_6(\cdot)} = 1)$
  $= \Pr(P_1, \ldots, P_6 \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\mathrm{MOMAC}_{P_1,\ldots,P_6}(\cdot)} = 1)$,
  since $\mathcal{B}_{\mathcal{A}}$ gives $\mathcal{A}$ a perfect simulation of $\mathrm{MOMAC}_{P_1,\ldots,P_6}(\cdot)$ if $\mathcal{O}_i(\cdot) = P_i(\cdot)$ for $1 \le i \le 6$.
- $\Pr(P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{B}_{\mathcal{A}}^{Q_1(\cdot),\ldots,Q_6(\cdot)} = 1)$
  $= \Pr(P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\mathrm{OMAC}_P(\cdot)} = 1)$,
  since $\mathcal{B}_{\mathcal{A}}$ gives $\mathcal{A}$ a perfect simulation of $\mathrm{OMAC}_P(\cdot)$ if $\mathcal{O}_i(\cdot) = Q_i(\cdot)$ for $1 \le i \le 6$. See Fig. 13 and Fig. 14. Note that `Rnd` is canceled in Fig. 13.

This concludes the proof of the lemma. Q.E.D.

4.4 Proof of Main Lemma for OMAC-family

We finally give a proof of Main Lemma for OMAC-family.

Proof (of Lemma 3.1). By the triangle inequality, the left hand side of (2) is at most

$$\Big| \Pr(P_1, \ldots, P_6 \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\mathrm{MOMAC}_{P_1,\ldots,P_6}(\cdot)} = 1) - \Pr(R \stackrel{R}{\leftarrow} \mathrm{Rand}(*, n) : \mathcal{A}^{R(\cdot)} = 1) \Big| \qquad (12)$$

$$+\Big|\Pr(P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\text{OMAC-family}_P(\cdot)} = 1)$$
$$- \Pr(P_1, \ldots, P_6 \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\mathrm{MOMAC}_{P_1,\ldots,P_6}(\cdot)} = 1)\Big| \ . \tag{13}$$

Lemma 4.1 gives us an upper bound on (12) and Lemma 4.4 gives us an upper bound on (13). Therefore the bound follows since

$$\frac{\sigma^2}{2^n} + \frac{3\sigma^2}{2} \cdot \left(\frac{1}{2^n} + \epsilon\right) = \frac{\sigma^2}{2} \cdot \left(\frac{5}{2^n} + 3\epsilon\right) \ .$$

This concludes the proof of the lemma. Q.E.D.

5 Proof for TMAC-family

5.1 Q_1, Q_2, Q_3 [9] and FCBC [3]

Let H, $\mathtt{Cst}_1$ and $\mathtt{Cst}_2$ satisfy the conditions in Sec. 2.3.2 for some sufficiently small $\epsilon_1, \epsilon_2, \epsilon_3$. For a random permutation $P \in \mathrm{Perm}(n)$ and a random string $K_2 \in \mathcal{K}_H$, define

$$\begin{cases} Q_1(x) \stackrel{\text{def}}{=} P(x), \\ Q_2(x) \stackrel{\text{def}}{=} P(x \oplus H_{K_2}(\mathtt{Cst}_1)), \\ Q_3(x) \stackrel{\text{def}}{=} P(x \oplus H_{K_2}(\mathtt{Cst}_2)). \end{cases} \tag{14}$$

The following proposition shows that $Q_1(\cdot)$, $Q_2(\cdot)$, $Q_3(\cdot)$ are indistinguishable from a pair of three independent random permutations $P_1(\cdot)$, $P_2(\cdot)$, $P_3(\cdot)$.

Proposition 5.1 *Let $\mathcal{A}$ be an adversary which asks at most q queries in total. Then*

$$\Big|\Pr(P \stackrel{R}{\leftarrow} \mathit{Perm}(n); K_2 \stackrel{R}{\leftarrow} \mathcal{K}_H : \mathcal{A}^{Q_1(\cdot),Q_2(\cdot),Q_3(\cdot)} = 1)$$
$$- \Pr(P_1, P_2, P_3 \stackrel{R}{\leftarrow} \mathit{Perm}(n) : \mathcal{A}^{P_1(\cdot),P_2(\cdot),P_3(\cdot)} = 1)\Big| \le \frac{q^2}{2} \cdot \left(\frac{1}{2^n} + \epsilon\right) \ ,$$

where $\epsilon = \max\{\epsilon_1, \epsilon_2, \epsilon_3\}$.

A proof is given in [9].

Next we recall FCBC [3]. It uses three independent random permutations $P_1, P_2, P_3 \in \mathrm{Perm}(n)$. The algorithm $\mathrm{FCBC}_{P_1,P_2,P_3}(\cdot)$ is described in Fig. 15 and illustrated in Fig. 16.

5.2 FCBC is Pseudorandom

We prove that FCBC is pseudorandom (information-theoretic result).

Lemma 5.1 *Let $\mathcal{A}$ be an adversary which asks at most q queries, having aggregate length of at most σ blocks. Assume $\sigma \le 2^n/2$. Then*

$$\Big|\Pr(P_1, P_2, P_3 \stackrel{R}{\leftarrow} \mathit{Perm}(n) : \mathcal{A}^{\mathrm{FCBC}_{P_1,P_2,P_3}(\cdot)} = 1) - \Pr(R \stackrel{R}{\leftarrow} \mathit{Rand}(*, n) : \mathcal{A}^{R(\cdot)} = 1)\Big| \le \frac{2\sigma^2}{2^n} \ .$$

To prove Lemma 5.1, we define CBC-E (CBC MAC without final encryption). It takes a message M such that $|M| = mn$ for some $m \ge 1$. It is obtained from the CBC MAC by removing the final encryption. More precisely, the algorithm $\text{CBC-E}_P(\cdot)$ is described in Fig. 17, where $P \in \mathrm{Perm}(n)$ is a random permutation.

We first show the following lemma.

Algorithm $\mathrm{FCBC}_{P_1,P_2,P_3}(M)$
$Y[0] \leftarrow 0^n$
Partition M into $M[1] \cdots M[m]$
for $i \leftarrow 1$ **to** $m-1$ **do**
  $X[i] \leftarrow M[i] \oplus Y[i-1]$
  $Y[i] \leftarrow P_1(X[i])$
$X[m] \leftarrow \mathtt{pad}_n(M[m]) \oplus Y[m-1]$
if $|M[m]| = n$ **then** $T \leftarrow P_2(X[m])$
  else $T \leftarrow P_3(X[m])$
return T

Fig. 15. Definition of FCBC.

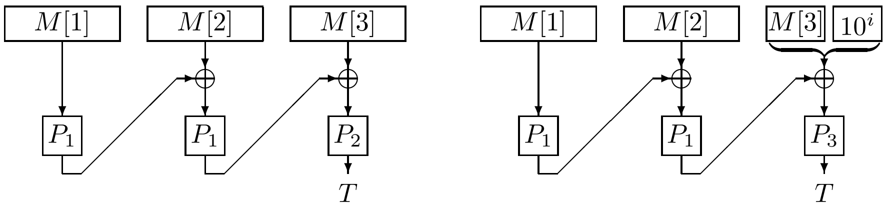

Fig. 16. Illustration of FCBC.

Algorithm $\text{CBC-E}_P(M)$
$Y[0] \leftarrow 0^n$
Partition M into $M[1] \cdots M[m]$
for $i \leftarrow 1$ **to** $m-1$ **do**
  $X[i] \leftarrow M[i] \oplus Y[i-1]$
  $Y[i] \leftarrow P(X[i])$
$X[m] \leftarrow M[m] \oplus Y[m-1]$
return $X[m]$

Fig. 17. Definition of CBC-E.

Lemma 5.2 (CBC-E Collision Bound) *Let q, $m_1, \ldots, m_q$ and σ be integers such that $m_i \geq 1$, $\sigma = m_1 + \cdots + m_q$, and $\sigma \leq 2^n/2$. Let $M^{(1)}, \ldots, M^{(q)}$ be fixed and distinct bit strings such that $|M^{(i)}| = m_i n$. Then*

$$\Pr(P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : 1 \leq {}^{\exists}i < {}^{\exists}j \leq q, \text{CBC-E}_P(M^{(i)}) = \text{CBC-E}_P(M^{(j)})) \leq \frac{\sigma^2}{2^n} \ .$$

Proof. We view the computation of $\text{CBC-E}_P(M^{(i)})$ as playing the game given in Fig. 18.

Initialization:
1: **for** $i \leftarrow 1$ **to** q **do** $X^{(i)}[1] \leftarrow M^{(i)}[1]$;
2: **for all** $x \in \{0,1\}^n$ **do** $P(x) \leftarrow$ `undefined`;
3: `bad` $\leftarrow$ `false`; $BAD \leftarrow \{X^{(1)}[1], \ldots, X^{(q)}[q]\}$;
Computation of $X^{(1)}[m_1], \ldots, X^{(q)}[m_q]$**:**
11: **for** $j \leftarrow 1$ **to** σ **do**
12:     **for** $i \leftarrow 1$ **to** q **do**
13:         **if** $j < m_i$ **then**
14:             **if** $X^{(i)}[j] \notin \mathrm{Domain}(P)$ **then**
15:                 $Y^{(i)}[j] \stackrel{R}{\leftarrow} \overline{\mathrm{Range}}(P)$;
16:                 $P(X^{(i)}[j]) \leftarrow Y^{(i)}[j]$;
17:                 $X^{(i)}[j+1] \leftarrow Y^{(i)}[j] \oplus M^{(i)}[j+1]$;
18:                 $BAD' \leftarrow \{X^{(i)}[j+1]\}$;
19:                 $Index \leftarrow \{k \mid i+1 \leq k \leq q,\ j < m_k \text{ and } X^{(i)}[j] = X^{(k)}[j]\}$;
20:                 **for all** $k \in Index$ **do**
21:                     $Y^{(k)}[j] \leftarrow Y^{(i)}[j]$;
22:                     $X^{(k)}[j+1] \leftarrow Y^{(k)}[j] \oplus M^{(k)}[j+1]$;
23:                     $BAD' \leftarrow BAD' \cup \{X^{(k)}[j+1]\}$;
24:                 **if** $BAD' \cap BAD \neq \emptyset$ **then** `bad` $\leftarrow$ `true`;
25:                 **else** $BAD \leftarrow BAD' \cup BAD$;

Fig. 18. Game used in the proof of Lemma 5.2.

Similarly to the proof of Lemma 4.2, it is enough to bound the probability of the event that `bad` $\leftarrow$ `true`.

In line 24, it is required that some $Y^{(i)}[j]$ was selected in line 15 such that $Y^{(i)}[j] \oplus M^{(i)}[j+1] \in BAD$, or $Y^{(i)}[j] \oplus M^{(k)}[j+1] \in BAD$ for some $k \in Index$. Suppose that the set BAD begins with l_0 points. Then it grows by the number of points in BAD' with each random choice of $Y^{(i)}[j]$. Now, suppose that for the t-th process of line 15, the corresponding BAD' after line 23 has l_t points, assuming that `bad` is `false` for the first $t-1$ process of line 15. Define

$$V(t) \stackrel{\text{def}}{=} \Pr_{\text{line 15}}(\texttt{bad} \leftarrow \texttt{true} \text{ at the } t\text{-th choice of } Y^{(i)}[j] \mid \texttt{bad} \text{ is } \texttt{false} \text{ before choosing } Y^{(i)}[j]) \ .$$

Then we have

$$V(t) = \frac{(l_0 + l_1 + \cdots + l_{t-1})l_t}{2^n - (t-1)} \ ,$$

since P has $2^n - (t-1)$ `undefined` domain points, BAD has $(l_0 + l_1 + \cdots + l_{t-1})$ points, and BAD' has l_t points.

Also, suppose that the game terminates after s process of line 15. Then we have

$$\Pr_{\text{line 15}}(\texttt{bad} \leftarrow \texttt{true}) \leq \sum_{1\leq t\leq s} V(t) = \sum_{1\leq t\leq s} \frac{(l_0+l_1+\cdots+l_{t-1})l_t}{2^n-(t-1)} \ .$$

Now we can bound the above by

$$\sum_{1\leq t\leq s} \frac{(l_0+l_1+\cdots+l_{t-1})l_t}{2^n-(t-1)} \leq \frac{2}{2^n}\sum_{1\leq t\leq s}(l_0+l_1+\cdots+l_{t-1})l_t \leq \frac{\sigma^2}{2^n} \ ,$$

where the first inequality follows since s is at most σ, which is at most $2^n/2$, and the second inequality follows since $\sigma \geq l_0+l_1+\cdots+l_s$ and

$$\sum_{1\leq t\leq s}(l_0+l_1+\cdots+l_{t-1})l_t \leq \frac{\sigma^2-{l_0}^2-{l_1}^2-\cdots-{l_s}^2}{2} \leq \frac{\sigma^2}{2} \ .$$

Q.E.D.

We next consider the following two sets.

$$\begin{cases} D_1 \stackrel{\text{def}}{=} \{M \mid M\in\{0,1\}^* \text{ and } |M| \text{ is a positive multiple of } n\} \\ D_2 \stackrel{\text{def}}{=} \{M \mid M\in\{0,1\}^* \text{ and } |M| \text{ is not a positive multiple of } n\} \end{cases}$$

We show the following lemma.

Lemma 5.3 *Let q_1, q_2 be two non-negative integers. For $1\leq i\leq 2$, let $M_i^{(1)},\ldots,M_i^{(q_i)}$ be fixed bit strings such that $M_i^{(j)}\in D_i$ for $1\leq j\leq q_i$ and $\{M_i^{(1)},\ldots,M_i^{(q_i)}\}$ are distinct. Similarly, for $1\leq i\leq 2$, let $T_i^{(1)},\ldots,T_i^{(q_i)}$ be fixed n-bit strings such that $\{T_i^{(1)},\ldots,T_i^{(q_i)}\}$ are distinct. Then the number of $P_1,P_2,P_3\in Perm(n)$ such that*

$$\begin{cases} FCBC_{P_1,P_2,P_3}(M_1^{(i)}) = T_1^{(i)} \text{ for } 1\leq {}^\forall i\leq q_1 \text{ and} \\ FCBC_{P_1,P_2,P_3}(M_2^{(i)}) = T_2^{(i)} \text{ for } 1\leq {}^\forall i\leq q_2 \end{cases} \tag{15}$$

is at least $\{(2^n)!\}^3\left(1-\frac{\sigma^2}{2^n}\right)\cdot\frac{1}{2^{qn}}$, where $q=q_1+q_2$, $\sigma_i=\sum_{1\leq j\leq q_i}\|M_i^{(j)}\|_n$ and $\sigma=\sigma_1+\sigma_2$.

Proof. We first consider $M_1^{(1)},\ldots,M_1^{(q_1)}$. The number of P_1 such that

$$\text{CBC-E}_{P_1}(M_1^{(i)}) = \text{CBC-E}_{P_1}(M_1^{(j)}) \text{ for } 1\leq {}^\exists i < {}^\exists j\leq q_1$$

is at most $\{(2^n)!\}\cdot\frac{\sigma_1^2}{2^n}$ from Lemma 5.2.

We next consider $M_2^{(1)},\ldots,M_2^{(q_2)}$. Let ${M'_2}^{(i)}$ denote the padded message of $M_2^{(i)}$. Then the number of P_1 such that

$$\text{CBC-E}_{P_1}({M'_2}^{(i)}) = \text{CBC-E}_{P_1}({M'_2}^{(j)}) \text{ for } 1\leq {}^\exists i < {}^\exists j\leq q_2$$

is at most $\{(2^n)!\}\cdot\frac{\sigma_2^2}{2^n}$ from Lemma 5.2.

Therefore, we have at least

$$\{(2^n)!\}\left(1-\frac{\sigma_1^2}{2^n}-\frac{\sigma_2^2}{2^n}\right)$$

choice of P_1 such that

$$\begin{cases} \text{CBC-E}_{P_1}(M_1^{(i)}) \neq \text{CBC-E}_{P_1}(M_1^{(j)}) \text{ for } 1 \leq {}^\forall i < {}^\forall j \leq q_1 \text{ and} \\ \text{CBC-E}_{P_1}(M_2'^{(i)}) \neq \text{CBC-E}_{P_1}(M_2'^{(j)}) \text{ for } 1 \leq {}^\forall i < {}^\forall j \leq q_2 \end{cases} \tag{16}$$

We fix any P_1 which satisfies (16).

Now P_1 is fixed in such a way that the inputs to P_2 are distinct and the inputs to P_3 are distinct. Also, the corresponding outputs $\{T_1^{(1)}, \ldots, T_1^{(q_1)}\}$ are distinct, and $\{T_2^{(1)}, \ldots, T_2^{(q_2)}\}$ are distinct. Therefore, we have at least

$$\{(2^n)!\} \left(1 - \frac{\sigma_1^2}{2^n} - \frac{\sigma_2^2}{2^n}\right) \cdot (2^n - q_1)! \cdot (2^n - q_2)!$$

choice of P_1, P_2, P_3 which satisfies (15). This bound is at least $\{(2^n)!\}^3 \left(1 - \frac{\sigma^2}{2^n}\right) \cdot \frac{1}{2^{qn}}$ since $\sigma^2 \geq \sigma_1^2 + \sigma_2^2$ and $(2^n - q_i)! \geq \frac{(2^n)!}{2^{q_i n}}$.

This concludes the proof of the lemma. Q.E.D.

We now prove Lemma 5.1

Proof (of Lemma 5.1). We proceed similarly to the proof of Lemma 4.1.

Let $\mathcal{O}$ be either $\text{FCBC}_{P_1,P_2,P_3}$ or R. Since $\mathcal{A}$ is computationally unbounded, there is no loss of generality to assume that $\mathcal{A}$ is deterministic.

Now for the query $\mathcal{A}$ makes to the oracle $\mathcal{O}$, define the query-answer pair $(M_j^{(i)}, T_j^{(i)}) \in D_j \times \{0,1\}^n$, where $\mathcal{A}$'s i-th query in D_j was $M_j^{(i)} \in D_j$ and the answer it got was $T_j^{(i)} \in \{0,1\}^n$.

Suppose that we run $\mathcal{A}$ with the oracle. For this run, assume that $\mathcal{A}$ made q_j queries in D_j, where $1 \leq j \leq 2$ and $q_1 + q_2 = q$. Also, for $1 \leq i \leq 2$, let $\sigma_i = \sum_{1 \leq j \leq q_i} \|M_i^{(j)}\|_n$. For this run, we define view v of $\mathcal{A}$ as

$$v \stackrel{\text{def}}{=} \langle (T_1^{(1)}, \ldots, T_1^{(q_1)}), (T_2^{(1)}, \ldots, T_2^{(q_2)}) \rangle \ . \tag{17}$$

Since $\mathcal{A}$ is deterministic, the i-th query $\mathcal{A}$ makes is fully determined by the first $i-1$ query-answer pairs. This implies that if we fix some qn-bit string V and return the i-th n-bit block as the answer for the i-th query $\mathcal{A}$ makes (instead of the oracle), then

- $\mathcal{A}$'s queries are uniquely determined,
- q_1, q_2 are uniquely determined,
- σ_1, σ_2 are uniquely determined,
- the parsing of V into the format defined in (17) is uniquely determined, and
- the final output of $\mathcal{A}$ (0 or 1) is uniquely determined.

Let $\boldsymbol{V}_{one}$ be a set of all qn-bit strings V such that $\mathcal{A}$ outputs 1. We let $N_{one} \stackrel{\text{def}}{=} \#\boldsymbol{V}_{one}$. Also, let $\boldsymbol{V}_{good}$ be a set of all qn-bit strings V such that:

For $1 \leq {}^\forall i < {}^\forall j \leq q$, the i-th n-bit block of $V \neq$ the j-th n-bit block of V.

Note that if $V \in \boldsymbol{V}_{good}$, then the corresponding parsing v of V satisfies that: $\{T_1^{(1)}, \ldots, T_1^{(q_1)}\}$ are distinct and $\{T_2^{(1)}, \ldots, T_2^{(q_2)}\}$ are distinct. Now observe that the number of V which is *not* in the set $\boldsymbol{V}_{good}$ is at most $\binom{q}{2}\frac{2^{qn}}{2^n}$. Therefore, we have

$$\#\{V \mid V \in (\boldsymbol{V}_{one} \cap \boldsymbol{V}_{good})\} \geq N_{one} - \binom{q}{2}\frac{2^{qn}}{2^n} \ . \tag{18}$$

Evaluation of p_{rand}. We first evaluate

$$p_{rand} \stackrel{\text{def}}{=} \Pr(R \stackrel{R}{\leftarrow} \mathrm{Rand}(*, n) : \mathcal{A}^{R(\cdot)} = 1) \ .$$

Then it is not hard to see

$$p_{rand} = \sum_{V \in \boldsymbol{V}_{one}} \frac{1}{2^{qn}} = \frac{N_{one}}{2^{qn}} \ .$$

Evaluation of p_{real}. We next evaluate

$$\begin{aligned} p_{real} &\stackrel{\text{def}}{=} \Pr(P_1, P_2, P_3 \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\mathrm{FCBC}_{P_1,P_2,P_3}(\cdot)} = 1) \\ &= \frac{\#\{(P_1, P_2, P_3) \mid \mathcal{A}^{\mathrm{FCBC}_{P_1,P_2,P_3}(\cdot)} = 1\}}{\{(2^n)!\}^3} \ . \end{aligned}$$

Then from Lemma 5.3, we have

$$\begin{aligned} p_{real} &\geq \sum_{V \in (\boldsymbol{V}_{one} \cap \boldsymbol{V}_{good})} \frac{\#\{(P_1, P_2, P_3) \mid (P_1, P_2, P_3) \text{ satisfying (15)}\}}{\{(2^n)!\}^3} \\ &\geq \sum_{V \in (\boldsymbol{V}_{one} \cap \boldsymbol{V}_{good})} \left(1 - \frac{\sigma^2}{2^n}\right) \cdot \frac{1}{2^{qn}} \ . \end{aligned}$$

Completing the Proof. From (18) we have

$$\begin{aligned} p_{real} &\geq \left(N_{one} - \binom{q}{2} \frac{2^{qn}}{2^n}\right) \cdot \left(1 - \frac{\sigma^2}{2^n}\right) \cdot \frac{1}{2^{qn}} \\ &= \left(p_{rand} - \binom{q}{2} \frac{1}{2^n}\right) \cdot \left(1 - \frac{\sigma^2}{2^n}\right) \\ &\geq p_{rand} - \binom{q}{2} \frac{1}{2^n} - \frac{\sigma^2}{2^n} \\ &\geq p_{rand} - \frac{q^2 + \sigma^2}{2^n} \\ &\geq p_{rand} - \frac{2\sigma^2}{2^n} \ . \end{aligned} \tag{19}$$

Applying the same argument to $1 - p_{real}$ and $1 - p_{rand}$ yields that

$$1 - p_{real} \geq 1 - p_{rand} - \frac{2\sigma^2}{2^n} \ . \tag{20}$$

Finally, (19) and (20) give $|p_{real} - p_{rand}| \leq \frac{2\sigma^2}{2^n}$. Q.E.D.

5.3 From FCBC to TMAC-family

The next lemma shows that $\text{TMAC-family}_{P,K_2}(\cdot)$ and $\mathrm{FCBC}_{P_1,P_2,P_3}(\cdot)$ are indistinguishable.

Lemma 5.4 *Let $\mathcal{A}$ be an adversary which asks at most q queries, having aggregate length of at most σ blocks. Assume $\sigma \le 2^n/2$. Then*

$$\Big|\Pr(P \stackrel{R}{\leftarrow} \mathit{Perm}(n), K_2 \stackrel{R}{\leftarrow} \mathcal{K}_H : \mathcal{A}^{\text{TMAC-family}_{P,K_2}(\cdot)} = 1)$$
$$- \Pr(P_1, P_2, P_3 \stackrel{R}{\leftarrow} \mathit{Perm}(n) : \mathcal{A}^{\text{FCBC}_{P_1,P_2,P_3}(\cdot)} = 1)\Big| \le \frac{\sigma^2}{2} \cdot \left(\frac{1}{2^n} + \epsilon\right) \;.$$

By using Proposition 5.1, it can be proved similarly to the proof of Lemma 4.4.

5.4 Proof of Main Lemma for TMAC-family

We finally give a proof of Main Lemma for TMAC-family.

Proof (of Lemma 3.2). By the triangle inequality, the left hand side of (3) is at most

$$\Big|\Pr(P_1, P_2, P_3 \stackrel{R}{\leftarrow} \text{Perm}(n) : \mathcal{A}^{\text{FCBC}_{P_1,P_2,P_3}(\cdot)} = 1)$$
$$- \Pr(R \stackrel{R}{\leftarrow} \text{Rand}(*, n) : \mathcal{A}^{R(\cdot)} = 1)\Big| \tag{21}$$

$$+ \Big|\Pr(P \stackrel{R}{\leftarrow} \text{Perm}(n), K_2 \stackrel{R}{\leftarrow} \mathcal{K}_H : \mathcal{A}^{\text{TMAC-family}_{P,K_2}(\cdot)} = 1)$$
$$- \Pr(P_1, P_2, P_3 \stackrel{R}{\leftarrow} \text{Perm}(n) : \mathcal{A}^{\text{FCBC}_{P_1,P_2,P_3}(\cdot)} = 1)\Big| \;. \tag{22}$$

Lemma 5.1 gives us an upper bound on (21) and Lemma 5.4 gives us an upper bound on (22). Therefore the bound follows since

$$\frac{2\sigma^2}{2^n} + \frac{\sigma^2}{2} \cdot \left(\frac{1}{2^n} + \epsilon\right) = \frac{\sigma^2}{2} \cdot \left(\frac{5}{2^n} + \epsilon\right) \;.$$

This concludes the proof of the lemma. Q.E.D.

6 Proof for XCBC

6.1 Q_1, Q_2, Q_3

For a random permutation $P \in \text{Perm}(n)$ and two random n-bit strings $K_2, K_3 \in \{0,1\}^n$, define

$$\begin{cases} Q_1(x) \stackrel{\text{def}}{=} P(x), \\ Q_2(x) \stackrel{\text{def}}{=} P(x \oplus K_2), \\ Q_3(x) \stackrel{\text{def}}{=} P(x \oplus K_3). \end{cases} \tag{23}$$

The following proposition shows that $Q_1(\cdot)$, $Q_2(\cdot)$, $Q_3(\cdot)$ are indistinguishable from a pair of three independent random permutations $P_1(\cdot)$, $P_2(\cdot)$, $P_3(\cdot)$.

Proposition 6.1 *Let $\mathcal{A}$ be an adversary which asks at most q queries in total. Then*

$$\Big|\Pr(P \stackrel{R}{\leftarrow} \mathit{Perm}(n); K_2, K_3 \stackrel{R}{\leftarrow} \{0,1\}^n : \mathcal{A}^{Q_1(\cdot),Q_2(\cdot),Q_3(\cdot)} = 1)$$
$$- \Pr(P_1, P_2, P_3 \stackrel{R}{\leftarrow} \mathit{Perm}(n) : \mathcal{A}^{P_1(\cdot),P_2(\cdot),P_3(\cdot)} = 1)\Big| \le \frac{q^2}{2^n} \;,$$

where $\epsilon = \max\{\epsilon_1, \epsilon_2, \epsilon_3\}$.

It can be proved by extending the proof of [3, Lemma 4]. Also, it can be proved similar to Proposition 5.1.

6.2 From FCBC to XCBC

The next lemma shows that $\mathrm{XCBC}_{P,K_2,K_3}(\cdot)$ and $\mathrm{FCBC}_{P_1,P_2,P_3}(\cdot)$ are indistinguishable.

Lemma 6.1 *Let $\mathcal{A}$ be an adversary which asks at most q queries, having aggregate length of at most σ blocks. Assume $\sigma \le 2^n/2$. Then*

$$\Big|\Pr(P \stackrel{R}{\leftarrow} \mathit{Perm}(n), K_2, K_3 \stackrel{R}{\leftarrow} \{0,1\}^n : \mathcal{A}^{\mathrm{XCBC}_{P,K_2,K_3}(\cdot)} = 1) \\ - \Pr(P_1, P_2, P_3 \stackrel{R}{\leftarrow} \mathit{Perm}(n) : \mathcal{A}^{\mathrm{FCBC}_{P_1,P_2,P_3}(\cdot)} = 1)\Big| \le \frac{\sigma^2}{2^n} \ .$$

By using Proposition 6.1, it can be proved similarly to the proof of Lemma 4.4.

6.3 Proof of Main Lemma for XCBC

We finally give a proof of Main Lemma for XCBC.

Proof (of Lemma 3.3). By the triangle inequality, the left hand side of (4) is at most

$$\Big|\Pr(P_1, P_2, P_3 \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\mathrm{FCBC}_{P_1,P_2,P_3}(\cdot)} = 1) \\ - \Pr(R \stackrel{R}{\leftarrow} \mathrm{Rand}(*, n) : \mathcal{A}^{R(\cdot)} = 1)\Big| \tag{24}$$

$$+ \Big|\Pr(P \stackrel{R}{\leftarrow} \mathrm{Perm}(n), K_2, K_3 \stackrel{R}{\leftarrow} \{0,1\}^n : \mathcal{A}^{\mathrm{XCBC}_{P,K_2,K_3}(\cdot)} = 1) \\ - \Pr(P_1, P_2, P_3 \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\mathrm{FCBC}_{P_1,P_2,P_3}(\cdot)} = 1)\Big| \ . \tag{25}$$

Lemma 5.1 gives us an upper bound on (24) and Lemma 6.1 gives us an upper bound on (25). Therefore the bound follows since

$$\frac{2\sigma^2}{2^n} + \frac{\sigma^2}{2^n} = \frac{3\sigma^2}{2^n} \ .$$

This concludes the proof of the lemma. Q.E.D.

References

[1] M. Bellare, J. Kilian, and P. Rogaway. The security of the cipher block chaining message authentication code. *JCSS,* vol. 61, no. 3, pp. 362–399, 2000. Earlier version in *Advances in Cryptology — CRYPTO '94, LNCS 839,* pp. 341–358, Springer-Verlag, 1994.

[2] A. Berendschot, B. den Boer, J. P. Boly, A. Bosselaers, J. Brandt, D. Chaum, I. Damgård, M. Dichtl, W. Fumy, M. van der Ham, C. J. A. Jansen, P. Landrock, B. Preneel, G. Roelofsen, P. de Rooij, and J. Vandewalle. Final Report of RACE Integrity Primitives. *LNCS 1007,* Springer-Verlag, 1995.

[3] J. Black and P. Rogaway. CBC MACs for arbitrary-length messages: The three key constructions. *Advances in Cryptology — CRYPTO 2000, LNCS 1880,* pp. 197–215, Springer-Verlag, 2000.

[4] FIPS Publication 46-3. Data Encryption Standard (DES). U. S. Department of Commerce / National Institute of Standards and Technology, October 25, 1999.

[5] FIPS 113. Computer data authentication. Federal Information Processing Standards Publication 113, U. S. Department of Commerce / National Bureau of Standards, National Technical Information Service, Springfield, Virginia, 1994.

[6] O. Goldreigh, S. Goldwasser and S. Micali. How to construct random functions. *J. ACM,* vol. 33, no. 4, pp. 792–807, October 1986.

[7] ISO/IEC 9797-1. Information technology — security techniques — data integrity mechanism using a cryptographic check function employing a block cipher algorithm. International Organization for Standards, Geneva, Switzerland, 1999. Second edition.

[8] T. Iwata and K. Kurosawa. OMAC: One-Key CBC MAC. Pre-proceedings of *Fast Software Encryption, FSE 2003,* pp. 137–161, 2003. To appear in *LNCS,* Springer-Verlag.

[9] K. Kurosawa and T. Iwata. TMAC: Two-Key CBC MAC. *Topics in Cryptology — CT-RSA 2003, LNCS 2612,* pp. 33–49, Springer-Verlag, 2003.

[10] R. Lidl and H. Niederreiter. Introduction to finite fields and their applications, revised edition. Cambridge University Press, 1994.

[11] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.,* vol. 17, no. 2, pp. 373–386, April 1988.

[12] E. Petrank and C. Rackoff. CBC MAC for real-time data sources. *J.Cryptology,* vol. 13, no. 3, pp. 315–338, Springer-Verlag, 2000.

A The Field with 2^n Points

We interchangeably think of a point a in $\mathrm{GF}(2^n)$ in any of the following ways:

1. as an abstract point in a field;
2. as an n-bit string $a_{n-1} \cdots a_1 a_0 \in \{0,1\}^n$;
3. as a formal polynomial $a(\mathtt{u}) = a_{n-1}\mathtt{u}^{n-1} + \cdots + a_1\mathtt{u} + a_0$ with binary coefficients.

To add two points in $\mathrm{GF}(2^n)$, take their bitwise XOR. We denote this operation by $a \oplus b$.

To multiply two points, fix some irreducible polynomial $f(\mathtt{u})$ having binary coefficients and degree n. To be concrete, choose the lexicographically first polynomial among the irreducible degree n polynomials having a minimum number of coefficients. We list some indicated polynomials (See [10, Chapter 10] for other polynomials).

$$\begin{cases} f(\mathtt{u}) = \mathtt{u}^{64} + \mathtt{u}^4 + \mathtt{u}^3 + \mathtt{u} + 1 & \text{for } n = 64, \\ f(\mathtt{u}) = \mathtt{u}^{128} + \mathtt{u}^7 + \mathtt{u}^2 + \mathtt{u} + 1 & \text{for } n = 128, \text{ and} \\ f(\mathtt{u}) = \mathtt{u}^{256} + \mathtt{u}^{10} + \mathtt{u}^5 + \mathtt{u}^2 + 1 & \text{for } n = 256. \end{cases}$$

To multiply two points $a \in \mathrm{GF}(2^n)$ and $b \in \mathrm{GF}(2^n)$, regard a and b as polynomials $a(\mathtt{u}) = a_{n-1}\mathtt{u}^{n-1} + \cdots + a_1\mathtt{u} + a_0$ and $b(\mathtt{u}) = b_{n-1}\mathtt{u}^{n-1} + \cdots + b_1\mathtt{u} + b_0$, form their product $c(\mathtt{u})$ where one adds and multiplies coefficients in $\mathrm{GF}(2)$, and take the remainder when dividing $c(\mathtt{u})$ by $f(\mathtt{u})$.

Note that it is particularly easy to multiply a point $a \in \{0,1\}^n$ by $\mathtt{u}$. We show a method for $n = 128$, where $f(\mathtt{u}) = \mathtt{u}^{128} + \mathtt{u}^7 + \mathtt{u}^2 + \mathtt{u} + 1$. Then multiplying $a = a_{127} \cdots a_1 a_0$ by $\mathtt{u}$ yields a product $a_{127}\mathtt{u}^{128}+a_{126}\mathtt{u}^{127}+\cdots+a_1\mathtt{u}^2+a_0\mathtt{u}$. Thus, if $a_{127}=0$, then $a\cdot\mathtt{u}=a\ll 1$. If $a_{127}=1$, then we must add $\mathtt{u}^{128}$ to $a\ll 1$. Since $\mathtt{u}^{128}+\mathtt{u}^7+\mathtt{u}^2+\mathtt{u}+1=0$ we have $\mathtt{u}^{128}=\mathtt{u}^7+\mathtt{u}^2+\mathtt{u}+1$, so adding $\mathtt{u}^{128}$ means to xor by $0^{120}10000111$. In summary, when $n=128$,

$$a\cdot\mathtt{u}=\begin{cases} a\ll 1 & \text{if } a_{127}=0,\\ (a\ll 1)\oplus 0^{120}10000111 & \text{otherwise.}\end{cases} \tag{26}$$

Also, note that it is easy to divide a point $a\in\{0,1\}^n$ by $\mathtt{u}$, meaning that one multiplies a by the multiplicative inverse of $\mathtt{u}$ in the field: $a\cdot\mathtt{u}^{-1}$. We show a method for $n=128$. Then multiplying $a=a_{127}\cdots a_1a_0$ by $\mathtt{u}^{-1}$ yields a product $a_{127}\mathtt{u}^{126}+a_{126}\mathtt{u}^{125}+\cdots+a_2\mathtt{u}+a_1+a_0\mathtt{u}^{-1}$. Thus, if $a_0=0$, then $a\cdot\mathtt{u}^{-1}=a\gg 1$. If $a_0=1$, then we must add $\mathtt{u}^{-1}$ to $a\gg 1$. Since $\mathtt{u}^{128}+\mathtt{u}^7+\mathtt{u}^2+\mathtt{u}+1=0$ we have $\mathtt{u}^{127}=\mathtt{u}^6+\mathtt{u}+1+\mathtt{u}^{-1}$, so adding $\mathtt{u}^{-1}=\mathtt{u}^{127}+\mathtt{u}^6+\mathtt{u}+1$ means to xor by $10^{120}1000011$. In summary, when $n=128$,

$$a\cdot\mathtt{u}^{-1}=\begin{cases} a\gg 1 & \text{if } a_0=0,\\ (a\gg 1)\oplus 10^{120}1000011 & \text{otherwise.}\end{cases} \tag{27}$$