OMAC: One-Key CBC MAC

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

December 20, 2002

Abstract

In this paper, we present One-key CBC MAC (OMAC) and prove its security for arbitrary length messages. OMAC takes only one key, K (k bits) of a block cipher E. Previously, XCBC requires three keys, $(k + 2n)$ bits in total, and TMAC requires two keys, $(k + n)$ bits in total, where n denotes the block length of E.

1 Introduction

1.1 Background

The CBC MAC [6, 7] is a well-known method to generate a message authentication code (MAC) based on a block cipher. Bellare, Kilian, and Rogaway proved the security of the CBC MAC for fixed message length mn, where n is the block length of the underlying block cipher E [1]. However, it is well known that the CBC MAC is *not* secure unless the message length is fixed.

Therefore, several variants of CBC MAC have been proposed for variable length messages.

First Encrypted MAC (EMAC) was proposed. It is obtained by encrypting the CBC MAC value by E again with a new key K_2. That is,

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

where M is a message and K_1 is the key of the CBC MAC and $\mathrm{CBC}_{K_1}(M)$ is the CBC MAC value of M. EMAC was originally developed for the RACE project [2]. Petrank and Rackoff then proved that EMAC is secure if the message length is a multiple of n, that is, if the domain is $(\{0\ 1\}^n)^+$ [11] (Vaudenay showed another proof by using decorrelation theory [14, 15]). Note that, however, EMAC requires two key schedulings of the underlying block cipher E.

Next Black and Rogaway proposed XCBC which requires only one key scheduling of the underlying block cipher E [3]. XCBC takes three keys: one block cipher key K_1, and two n-bit keys K_2 and K_3.

- If $M \in (\{0\ 1\})^+$ then XCBC computes exactly the same as the CBC MAC, except for XORing an n-bit key K_2 before encrypting the last block.
- If $M \notin (\{0\ 1\})^+$ then 10^i padding ($i \quad n-1-|M| \bmod n$) 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.

See Fig. 1.

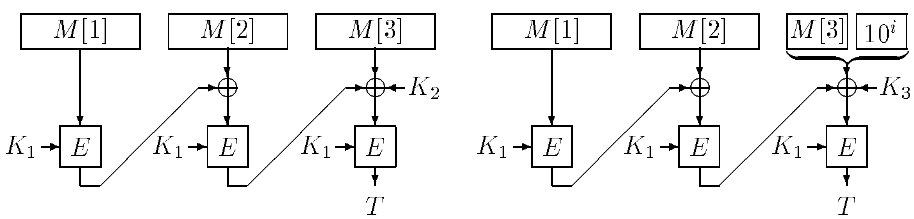

Figure 1: Illustration of XCBC.

A drawback of XCBC is, however, that it requires three keys, $(k + 2n)$ bits in total.

Finally Kurosawa and Iwata proposed Two-key CBC MAC (TMAC) [10]. TMAC takes two keys, $(k + n)$ bits in total: a block cipher key K_1 and an n-bit key K_2. TMAC is obtained from XCBC by replacing (K_2, K_3) with $(K_2 \cdot \mathtt{u}, K_2)$, where $\mathtt{u}$ is some constant in $\mathrm{GF}(2^n)$.

1.2 Our Contribution

In this paper, we present One-key CBC MAC (OMAC) and prove its security for arbitrary length messages. OMAC takes only one key, K of a block cipher E. The key length, k bits, is the minimum because the underlying block

Table 1: Comparison of key length.

	XCBC [3]	TMAC [10]	OMAC (This paper)
key length	$(k+2n)$ bits	$(k+n)$ bits	k bits

cipher must have a k-bit key K anyway. See Table 1 for comparison with XCBC and TMAC. OMAC is obtained from XCBC by replacing (K_2,K_3) with $(L\cdot\mathtt{u},L\cdot\mathtt{u}^{-1})$ for some constant $\mathtt{u}$ in GF(2^n), where L is given by

$$L \quad E_K(0^n) \ .$$

$L\cdot\mathtt{u}$ and $L\cdot\mathtt{u}^{-1}$ can be computed efficiently from L by one shift and one conditional XOR. OMAC is described as follows (see Fig. 2).

- If $M \in (\{0 \ 1\})^+$, then OMAC computes exactly the same as the CBC MAC, except for XORing $L\cdot\mathtt{u}$ before encrypting the last block.
- If $M \in (\{0 \ 1\})^+$, then 10^i padding ($i \quad n-1-|M| \bmod n$) is appended to M and OMAC computes exactly the same as the CBC MAC for the padded message, except for XORing $L\cdot\mathtt{u}^{-1}$ before encrypting the last block.

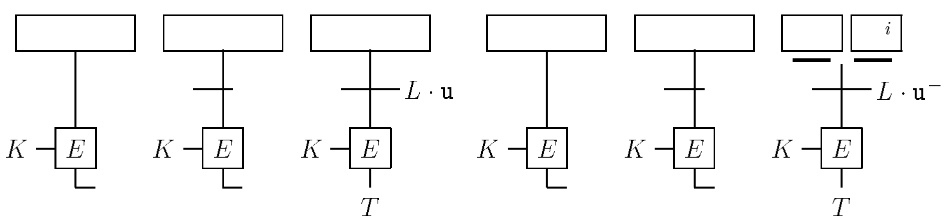

Figure 2: Illustration of OMAC. Note that $L \quad E_K(0^n)$.

Note that in TMAC, K_2 is a part of the key while in OMAC, L is not a part of the key and is generated from K.

This saving of the key length makes the security proof of OMAC much harder than that of TMAC substantially as shown below. In Fig. 2, suppose that $M[1] \quad 0^n$. Then the output of the first E_K is L. The same L appears again at the last block always. In general, such reuse of L would get one into trouble in the security proof. Indeed, the security proof of OMAC is substantially harder than the those of XCBC and TMAC due to this reuse of L.

(In OCB mode [13] and PMAC [5], $L \quad E_K(0^n)$ is also used as a key of a universal hash function. However, L appears as an output of some internal block cipher only with negligible probability.)

Nevertheless we prove that OMAC is as secure as XCBC, where the security analysis is in the concrete-security paradigm [1]. Further OMAC has all other nice properties which XCBC (and TMAC) has. That is, the domain of OMAC is $\{0\ 1\}^*$, it requires one key scheduling of the underlying block cipher E and $\max\{1\ \lceil |M|/n \rceil\}$ block cipher invocations.

1.3 Other Related Work

Jaulmes, Joux and Valette proposed RMAC [9] which is an extension of EMAC. RMAC encrypts the CBC MAC value with $K_2 \oplus R$, where R is an n-bit random string and it is a part of the tag. That is,

$$\mathrm{RMAC}_{K_1,K_2}(M) = \ (E_{K_2 \oplus R}(\mathrm{CBC}_{K_1}(M)), R\ )$$

They showed that the security of RMAC is beyond the birthday paradox limit.

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 $a \quad a_{n-1} \cdots a_1 a_0 \in \{0\ 1\}^n$, let $a\ < 1 = \ a_{n-2} \cdots a_1 a_0 0$ denote the n-bit string which is a left shift of a by 1 bit, while $a\ > 1 =$ $0 a_{n-1} \cdots a_2 a_1$ 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 $|a| \leq n$, we let

$$\mathtt{pad}_n(a) = \begin{cases} a 10^{n-|a|-1} & \text{if } |a| < n, \\ a & \text{if } |a| \quad n. \end{cases} \tag{1}$$

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

2.2 CBC MAC

The block cipher E is a function $E : \mathcal{K}_E \times \{0\ 1\}^n \rightarrow \{0\ 1\}^n$, where each $E(K, \cdot) = E_K(\cdot)$ is a permutation on $\{0\ 1\}^n$, $\mathcal{K}_E$ is the set of possible keys and n is the block length.

The CBC MAC [,7] is the simplest and most well-known algorithm to make a MAC from a block cipher E. Let $M \quad M[1] \circ M[2] \circ \cdots \circ M[m]$ be a message string, where $|M[1]| \quad |M[2]| \quad \cdots \quad |M[m]| \quad n$. Then $\mathrm{CBC}_K(M)$, the CBC MAC of M under key K, is defined as $Y[m]$, where

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

for $i = 1 \qquad , m$ and $Y[0] \quad 0^n$. Bellare, Kilian and Rogaway proved the security of the CBC MAC for fixed message length mn-bits [1].

2.3 The Field with 2^n Points

The field with 2^n points is denoted $\mathrm{GF}(2^n)$. 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 $a \oplus b$.

Multiplication. 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.

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

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 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 $n =1\ 28$, where $f(\mathtt{u})= \mathtt{u}^{128} + \mathtt{u}\ \ + \mathtt{u}^2 + \mathtt{u} + 1$. Then multiplying $a \quad a_{127} \cdots a_1 a_0$ by $\mathtt{u}$ yields a product $a_{127}\mathtt{u}^{128} + a_{126}\mathtt{u}^{12}\ \ + \cdots + a_1\mathtt{u}^2 + a_0\mathtt{u}$. Thus, if $a_{127} \quad 0$, then $a \cdot \mathtt{u} \quad a<\ \ 1$. If $a_{127} \quad 1$, then we must add $\mathtt{u}^{128}$ to $a<\ \ 1$. Since $\mathtt{u}^{12}\ \ + \mathtt{u}\ \ + \mathtt{u}^2 + \mathtt{u} + 1 \quad 0$ we have $\mathtt{u}^{128} \quad \mathtt{u}\ \ + \mathtt{u}^2 + \mathtt{u} + 1$, so adding $\mathtt{u}^{128}$ means to xor by $0^{120}10000111$. In summary, when $n =1\ 28$,

$$a \cdot \mathtt{u} \qquad \begin{array}{ll} a<\ \ 1 & \text{if } a_{127} =0\ , \\ (a<\ \ 1) \oplus 0^{120}10000111 & \text{otherwise.} \end{array} \tag{2}$$

Division. 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 $n =1\ 28$. Then multiplying $a \quad a_{127} \cdots a_1 a_0$ 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 $a \cdot \mathtt{u}^{-1} \quad a>\ \ 1$. If $a_0 =1$, then we must add $\mathtt{u}^{-1}$ to $a>\ \ 1$. Since $\mathtt{u}^{128} + \mathtt{u}\ \ + \mathtt{u}^2 + \mathtt{u} + 1 \quad 0$ we have $\mathtt{u}^{12} \quad \mathtt{u}\ \ + \mathtt{u} + 1 + \mathtt{u}^{-1}$, so adding $\mathtt{u}^{-1} \quad \mathtt{u}^{127} + \mathtt{u}\ \ + \mathtt{u} + 1$ means to xor by $10^{120}1000011$. In summary, when $n \quad 128$,

$$a \cdot \mathtt{u}^{-1} \qquad \begin{array}{ll} a>\ \ 1 & \text{if } a_0 =0\ , \\ (a>\ \ 1) \oplus 10^{120}1000011 & \text{otherwise.} \end{array} \tag{3}$$

3 Basic Construction

In this section, we show a basic construction of OMAC-family.

OMAC-family is defined by a block cipher $E : \mathcal{K}_E \quad 0\ \ 1\}^n \quad 0\ \ 1\}^n$, $\times\{ \qquad \to\{$ an n-bit constant $\mathtt{Cst}$, a universal hash function $H : \{0\ \ 1\}^n \quad X \quad 0\ \ 1\}^n$, $\to\{$ and two distinct constants $\mathtt{Cst}_1\ \ \mathtt{Cst}_2 \in X$, where X is the finite domain of H.

H, $\mathtt{Cst}_1$ and $\mathtt{Cst}_2$ must satisfy the following conditions while $\mathtt{Cst}$ is arbitrary. We write $H_L(\cdot)$ for $H(L, \cdot)$.

1. For any $y \in \{0\ \ 1\}^n$, the number of $L \in \{0\ \ 1\}^n$ such that $H_L(\mathtt{Cst}_1)= y$ is at most $\epsilon_1 \cdot 2^n$ for some sufficiently small ϵ_1.

2. For any $y \in \{0\ \ 1\}^n$, the number of $L \in \{0\ \ 1\}^n$ such that $H_L(\mathtt{Cst}_2)= y$ is at most $\epsilon_2 \cdot 2^n$ for some sufficiently small ϵ_2.

3. For any $y \in \{0\ 1\}^n$, the number of $L \in \{0\ 1\}^n$ such that $H_L(\texttt{Cst}_1) \oplus H_L(\texttt{Cst}_2) = \ y$ is at most $\epsilon_3 \cdot 2^n$ for some sufficiently small ϵ_3.

4. For any $y \in \{0\ 1\}^n$, the number of $L \in \{0\ 1\}^n$ such that $H_L(\texttt{Cst}_1) \oplus L \quad y$ is at most $\epsilon \ \cdot 2^n$ for some sufficiently small ϵ .

5. For any $y \in \{0\ 1\}^n$, the number of $L \in \{0\ 1\}^n$ such that $H_L(\texttt{Cst}_2) \oplus L \quad y$ is at most $\epsilon \ \cdot 2^n$ for some sufficiently small ϵ .

6. For any $y \in \{0\ 1\}^n$, the number of $L \in \{0\ 1\}^n$ such that $H_L(\texttt{Cst}_1) \oplus H_L(\texttt{Cst}_2) \oplus L \quad y$ is at most $\epsilon \ \cdot 2^n$ for some sufficiently small ϵ .

Remark. Property 1 and 2 says that $H_L(\texttt{Cst}_1)$ and $H_L(\texttt{Cst}_2)$ are almost uniformly distributed. Property 3 is satisfied by AXU (almost XOR universal) hash functions [12]. Property 4, 5, are new requirements introduced here.

The algorithm of OMAC-family is described in Fig. 3 and illustrated in Fig. 4, where $\texttt{pad}_n(\cdot)$ is defined in (1).

The key space $\mathcal{K}$ of OMAC-family is $\mathcal{K} \quad \mathcal{K}_E$. It takes a key $K \in \mathcal{K}_E$ and a message $M \in \{0\ 1\}^*$, and returns a string in $\{0\ 1\}^n$.

4 Proposed Specification

In this section, we show our proposed specification of OMAC-family. Our choice is; $\texttt{Cst} = 0\ ^n$, $H_L(x) = \ L \cdot x$, $\texttt{Cst}_1 \quad \texttt{u}$ and $\texttt{Cst}_2 \quad \texttt{u}^{-1}$, where "$\cdot$" denotes multiplication over GF(2^n). It is easy to see that the conditions in Sec. 3 are satisfied for $\epsilon_i = 2\ ^{-n}$ for $i = 1 \qquad 6$.

Equivalently, $L \quad E_K(0^n)$, $H_L(\texttt{Cst}_1) = \ L \cdot \texttt{u}$ and $H_L(\texttt{Cst}_2) = \ L \cdot \texttt{u}^{-1}$, where $L \cdot \texttt{u}$ and $L \cdot \texttt{u}^{-1}$ can be computed efficiently from L by one shift and one conditional XOR, respectively, as shown in (2) and (3).

We call this algorithm OMAC specifically. OMAC is defined in Fig. 5 and illustrated in Fig. 2.

5 Security of OMAC

5.1 Security Definitions

Let 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 Perm(n).

Algorithm OMAC-family$_K(M)$
$L \leftarrow E_K(\mathtt{Cst})$
$Y[0] \leftarrow 0^n$
Partition M into $M[1] \cdots M[m]$
for $i \leftarrow 1$ **to** $m-1$ **do**
  $X[i] \leftarrow M[i] \oplus Y[i-1]$
  $Y[i] \leftarrow E_K(X[i])$
$X[m] \leftarrow \mathtt{pad}_n(M[m]) \oplus Y[m-1]$
if $|M[m]| \quad n$ **then** $X[m] \leftarrow X[m] \oplus H_L(\mathtt{Cst}_1)$
  else $X[m] \leftarrow X[m] \oplus H_L(\mathtt{Cst}_2)$
$T \leftarrow E_K(X[m])$
return T

Figure 3: Definition of OMAC-family.

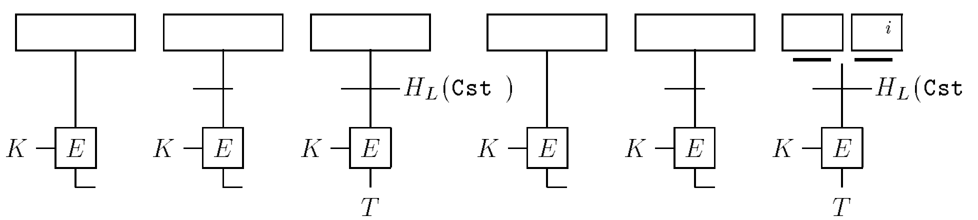

Figure 4: Illustration of OMAC-family.

Algorithm OMAC$_K(M)$
$L \leftarrow E_K(0^n)$
$Y[0] \leftarrow 0^n$
Partition M into $M[1] \cdots M[m]$
for $i \leftarrow 1$ **to** $m-1$ **do**
  $X[i] \leftarrow M[i] \oplus Y[i-1]$
  $Y[i] \leftarrow E_K(X[i])$
$X[m] \leftarrow \mathtt{pad}_n(M[m]) \oplus Y[m-1]$
if $|M[m]| \quad n$ **then** $X[m] \leftarrow X[m] \oplus L \cdot \mathtt{u}$
  else $X[m] \leftarrow X[m] \oplus L \cdot \mathtt{u}^{-1}$
$T \leftarrow E_K(X[m])$
return T

Figure 5: Definition of OMAC.

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 $P(\cdot)$, when allowed computation time t and q queries to an oracle (which is either $E_K(\cdot)$ or $P(\cdot)$). This advantage is defined as follows.

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

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

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

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

where the maximum is over all adversaries who run in time at most t, make at most q queries, and each query is at most μ-bits. We say that a MAC algorithm is secure if $\mathtt{Adv}_F^{\mathsf{mac}}(t,q,\mu)$ 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

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

where the maximum is over all adversaries who run in time at most t, make at most q queries, and each query is at most μ-bits. We say that a MAC algorithm is pseudorandom if $\mathtt{Adv}_F^{\mathsf{viprf}}(t,q,\mu)$ is sufficiently small.

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

5.2 Theorem Statements

We first prove that OMAC is pseudorandom if the underlying block cipher is a random permutation P (information-theoretic result).

Lemma 5.1 (Main Lemma) *Suppose that a random permutation $P \in$ Perm(n) is used in OMAC as the underlying block cipher. Let $\mathcal{A}$ be an adversary which asks at most q queries, and each query is at most nm-bits (m is the maximum number of blocks in each query). Assume $m \quad 2^n/4$. Then*

$$\begin{aligned}&\Pr(P \overset{R}{\leftarrow} \mathit{Perm}(n) : \mathcal{A}^{\mathrm{OMAC}_P(\cdot)} = 1\ ) \\ &\quad - \Pr(R \overset{R}{\leftarrow} \mathit{Rand}(*, n\ ) : \mathcal{A}^{R(\cdot)} = 1\ ) \quad \frac{(5m^2+1)q^2}{2^n}\end{aligned} \tag{4}$$

A proof is given in the next section.

We next show that OMAC is pseudorandom if the underlying block cipher E is secure. It is standard to pass to this complexity-theoretic result from Lemma 5.1. For example, see [1, Section 3.2] for the proof technique.

Corollary 5.1 *Let $E : \mathcal{K}_E \times \{0\ 1\}^n \to \{0\ 1\}^n$ be the underlying block cipher used in OMAC. Then*

$$\mathtt{Adv}^{\mathsf{viprf}}_{\mathrm{OMAC}}(t, q\ \ nm) \quad \frac{(5m^2+1)q^2}{2^n} + \mathtt{Adv}^{\mathsf{prp}}_{E}(t', q')$$

where $t' \quad t + O(mq)$ and $q' \quad mq$.

Finally we show that OMAC is secure as a MAC algorithm from Corollary 5.1 in the usual way. For example, see [1, Proposition 2.7] for the proof technique.

Theorem 5.1 *Let $E : \mathcal{K}_E \times \{0\ 1\}^n \to \{0\ 1\}^n$ be the underlying block cipher used in OMAC. Then*

$$\mathtt{Adv}^{\mathsf{mac}}_{\mathrm{OMAC}}(t, q\ \ nm) \quad \frac{(5m^2+1)q^2+1}{2^n} + \mathtt{Adv}^{\mathsf{prp}}_{E}(t', q')$$

where $t' \quad t + O(mq)$ and $q' \quad mq$.

5.3 Proof of Main Lemma

For a random permutation $P \in \mathrm{Perm}(n)$ and a random n-bit string $\mathtt{Rnd} \in \{0\ 1\}^n$, define

$$\begin{array}{ll} Q_1(x) \overset{\mathrm{ef}}{} P(x) \oplus \mathtt{Rnd} & Q_2(x) \overset{\mathrm{def}}{} P(x \oplus \mathtt{Rnd}) \oplus \mathtt{Rnd} \\ Q_3(x) \overset{\mathrm{def}}{} P(x \oplus \mathtt{Rnd} \oplus L \cdot \mathtt{u}) & Q\ \ (x) \overset{\mathrm{def}}{} P(x \oplus \mathtt{Rnd} \oplus L \cdot \mathtt{u}^{-1}) \\ Q\ \ (x) \overset{\mathrm{def}}{} P(x \oplus L \cdot \mathtt{u}) \text{ and} & Q\ \ (x) \overset{\mathrm{def}}{} P(x \oplus L \cdot \mathtt{u}^{-1}) \end{array} \tag{5}$$

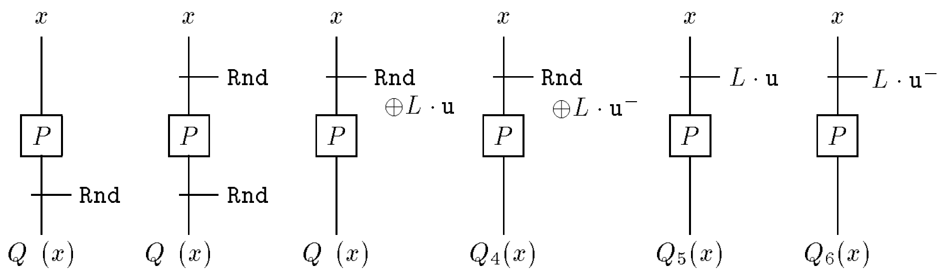

Figure 6: Illustrations of Q_1, Q_2 Q_3, Q , Q and Q . Note that L $P(\texttt{Cst})$.

where L $P(\texttt{Cst})$ and $\texttt{Cst} = 0^{\ \, n}$. See Fig. for illustrations. We first show that $Q_1(\cdot)$, $Q_2(\cdot)$, $Q_3(\cdot)$, $Q\ (\cdot)$, $Q\ (\cdot)$, $Q\ (\cdot)$ are indistinguishable from a pair of six independent random permutations $P_1(\cdot)$, $P_2(\cdot)$, $P_3(\cdot)$, $P\ (\cdot)$, $P\ (\cdot)$, $P\ (\cdot)$.

Lemma 5.2 *Let $\mathcal{A}$ be an adversary which asks at most q queries in total. Then*

$$\Pr(P \overset{R}{\leftarrow} Perm(n); \texttt{Rnd} \overset{R}{\leftarrow} \{0\ \ 1\}^n : \mathcal{A}^{Q_1(\cdot),\ldots,Q_6(\cdot)} = 1\ )$$
$$- \Pr(P_1 \qquad ,P \ \overset{R}{\leftarrow} Perm(n) : \mathcal{A}^{P_1(\cdot),\ldots,P_6(\cdot)} = 1\ ) \qquad \frac{3q^2}{2^n}$$

A proof is given in Appendix A.

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

We prove that MOMAC is pseudorandom.

Lemma 5.3 *Let $\mathcal{A}$ be an adversary which asks at most q queries, and each query is at most nm-bits. Assume $m \quad 2^n/4$. Then*

$$\Pr(P_1 \qquad ,P \ \overset{R}{\leftarrow} Perm(n) : \mathcal{A}^{\mathrm{MOMAC}_{P_1,\ldots,P_6}(\cdot)} = 1\ )$$
$$- \Pr(R \overset{R}{\leftarrow} Rand(*,n\ ) : \mathcal{A}^{R(\cdot)} = 1\ ) \qquad \frac{(2m^2+1)q^2}{2^n}$$

A proof is given in Appendix B.

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

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 $|M[m]| \quad n$ **then** $T \leftarrow P_3(X[m])$
        else $T \leftarrow P\ (X[m])$
if $m =1$ **then**
    $X[m] \leftarrow \mathtt{pad}_n(M[m])$
    if $|M[m]| \quad n$ **then** $T \leftarrow P\ (X[m])$
        else $T \leftarrow P\ (X[m])$
return T

Figure 7: Definition of MOMAC.

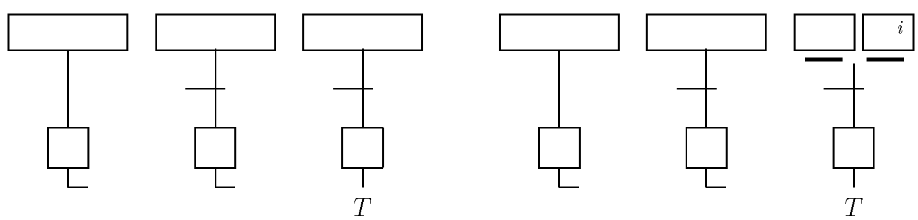

Figure 8: Illustration of MOMAC for $|M|$. $>n$

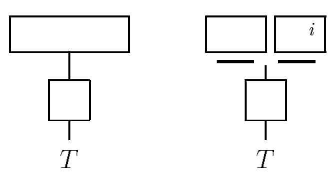

Figure 9: Illustration of MOMAC for $|M| \leq n$.

Lemma 5.4 *Let $\mathcal{A}$ be an adversary which asks at most q queries, and each query is at most nm-bits. Assume $m \quad 2^n/4$. Then*

$$\begin{aligned}&\Pr(P \stackrel{R}{\leftarrow} \mathit{Perm}(n) : \mathcal{A}^{\mathrm{OMAC}_P(\cdot)} = 1\ ) \\ &\qquad - \Pr(P_1 \quad , P \ \stackrel{R}{\leftarrow} \mathit{Perm}(n) : \mathcal{A}^{\mathrm{MOMAC}_{P_1} \quad {}_{P_6}(\cdot)} = 1\ ) \qquad \frac{3m^2q^2}{2^n}\end{aligned}$$

Proof. Suppose that there exists an adversary $\mathcal{A}$ such that

$$\begin{aligned}&\Pr(P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\mathrm{OMAC}_P(\cdot)} = 1\ ) \\ &\qquad - \Pr(P_1 \quad , P \ \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\mathrm{MOMAC}_{P_1} \quad {}_{P_6}(\cdot)} = 1\ ) \qquad \frac{3m^2q^2}{2^n}\end{aligned}$$

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

- $\mathcal{B}_{\mathcal{A}}$ asks at most mq queries, and
- $\Pr(P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{B}_{\mathcal{A}}^{Q_1(\cdot) \quad , Q_6(\cdot)} = 1\ )$
  $\qquad - \Pr(P_1 \quad , P \ \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{B}_{\mathcal{A}}^{P_1(\cdot) \quad , P_6(\cdot)} = 1\ ) \qquad \frac{3m^2q^2}{2^n},$

which contradicts Lemma 5.2.

Let $\mathcal{O}_1(\cdot) \qquad \mathcal{O}\ (\cdot)$ be $\mathcal{B}_{\mathcal{A}}$'s oracles. The construction of $\mathcal{B}_{\mathcal{A}}$ is given in Fig. 10.

Algorithm $\mathcal{B}_{\mathcal{A}}^{\mathcal{O}_1 \quad \mathcal{O}_6}$
When $\mathcal{A}$ asks its r-th query $M^{(r)}$:
    $T^{(r)} \leftarrow \mathrm{MOMAC}_{\mathcal{O}_1 \quad \mathcal{O}_6}(M^{(r)})$
    return $T^{(r)}$
When $\mathcal{A}$ halts and outputs b:
    output b

Figure 10: Algorithm $\mathcal{B}_{\mathcal{A}}$. Note that for $1 \quad i \quad 6$, $\mathcal{O}_i$ is either P_i or Q_i

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

Now we see that:

- $\mathcal{B}_{\mathcal{A}}$ asks at most mq queries to its oracles, since $\mathcal{A}$ asks at most q queries, and each query is at most nm-bits.

- $\Pr(P_1 \quad ,P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{B}_{\mathcal{A}}^{P_1(\cdot) \quad ,P_6(\cdot)} = 1\ )$
  $= \Pr(P_1 \quad ,P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\mathrm{MOMAC}_{P_1} \quad P_6(\cdot)} \quad 1)$,
  since $\mathcal{B}_{\mathcal{A}}$ gives $\mathcal{A}$ a perfect simulation of $\mathrm{MOMAC}_{P_1 \quad ,P_6}(\cdot)$ if $\mathcal{O}_i(\cdot) = P_i(\cdot)$ for $1 \quad i \quad 6$.

- $\Pr(P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{B}_{\mathcal{A}}^{Q_1(\cdot) \quad ,Q_6(\cdot)} = 1\ )$
  $= \Pr(P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\mathrm{OMAC}_P(\cdot)} \quad 1)$,
  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 $1 \quad i \quad 6$. See Fig. 11 and Fig. 12 for illustrations of $\mathcal{B}_{\mathcal{A}}$'s computation. Note that `Rnd` is canceled in Fig. 11.

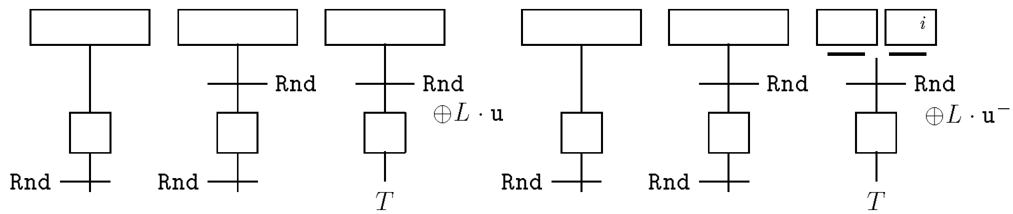

Figure 11: Computation of $\mathcal{B}_{\mathcal{A}}$ when $\mathcal{O}_i \quad Q_i$ for $1 \quad i \quad 6$, and $|M| \quad$.

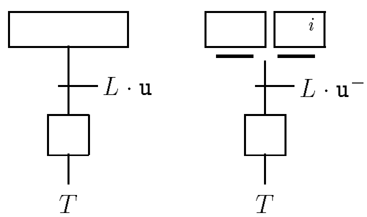

Figure 12: Computation of $\mathcal{B}_{\mathcal{A}}$ when $\mathcal{O}_i \quad Q_i$ for $1 \quad i \quad 6$, and $|M| \leq n$.

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

We finally give a proof of Main Lemma.

Proof (of Lemma 5.1). By the triangle inequality, the left hand side of (4) is at most

$$\Pr(P_1 \quad P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\mathrm{MOMAC}_{P_1} \quad ,P_6(\cdot)} = 1\ ) - \Pr(R \stackrel{R}{\leftarrow} \mathrm{Rand}(*,n) : \mathcal{A}^{R(\cdot)} = 1\ ) \tag{6}$$

$$+ \Pr(P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\mathrm{OMAC}_P(\cdot)} = 1\ ) - \Pr(P_1 \quad P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{\mathrm{MOMAC}_{P_1} \quad ,P_6(\cdot)} = 1\ ) \tag{7}$$

Lemma 5.3 gives us an upper bound on (6) and Lemma 5.4 gives us an upper bound on (7). Therefore the bound follows since

$$\frac{(2m^2+1)q^2}{2^n} + \frac{3m^2q^2}{2^n} \quad \frac{(5m^2+1)q^2}{2^n}$$

Q.E.D.

6 Discussions

6.1 Summary of Properties

We give a summary of properties of OMAC in Table 2.

Table 2: Summary of properties of OMAC.

Security function	Message authentication code.
Error propagation	Not applicable.
Synchronization	Not applicable.
Parallelizability	Sequential.
Keying material	Single block cipher key.
Ctr/IV/Nonce requirements	No counter/IV/nonce is used.
Memory requirements	Very modest.
Pre-processing capability	$L \quad E_K(\texttt{Cst})$, $L \cdot \texttt{u}$ and $L \cdot \texttt{u}^{-1}$ can be pre-processed.
Message-length requirements	Arbitrarily length.
Ciphertext expansion	Not applicable.

6.2 Advantages

Minimum key length. The key length of OMAC is k-bits, while the key length of XCBC is $(k+2n)$-bits and the key length of TMAC is $(k+n)$-bits.

Arbitrarily length messages. The domain of OMAC is $\{0\ 1\}^*$ and $|M|$ need not be a multiple of the block length n.

Optimal number of block cipher invocations. To generate a tag for any non-empty message $M \in \{0\ 1\}^*$, OMAC requires $\lceil |M|/n \rceil$ block cipher invocations (The empty string is an exception, and it requires one block cipher invocation).

Optimal number of block cipher key schedulings. OMAC needs only one block cipher key scheduling.

Provable security. We prove that OMAC is a variable input length pseudorandom function (VIPRF) with fixed output length assuming that the underlying block cipher is a pseudorandom permutation (PRP).

No decryption. As for any CBC MAC variant, OMAC does not use decryption of the block cipher.

Simplicity. Because OMAC is simple, it is easily implemented in both software and hardware.

6.3 Limitations

Sequential block cipher invocations. The CBC MAC and its variants, including OMAC, are not parallelizable.

Limited pre-processing capability. For OMAC, key scheduling of the underlying block cipher, $L \quad E_K(\mathtt{Cst})$, $L \cdot \mathtt{u}$ and $L \cdot \mathtt{u}^{-1}$ can be preprocessed. Additional pre-processing is not possible.

6.4 Design Rationale

Our choice for OMAC is $\mathtt{Cst} = 0^n$, $H_L(x) = L \cdot x$, $\mathtt{Cst}_1 \quad \mathtt{u}$ and $\mathtt{Cst}_2 \quad \mathtt{u}^{-1}$, where "$\cdot$" denotes multiplication over $\mathrm{GF}(2^n)$. Or equivalently, $L \quad E_K(0^n)$, $H_L(\mathtt{Cst}_1) = L \cdot \mathtt{u}$ and $H_L(\mathtt{Cst}_2) = L \cdot \mathtt{u}^{-1}$. Below, we list reasons of this choice.

- We adopted multiplications in $\mathrm{GF}(2^n)$ since it is simple, easy to understand, and easy to implement for appropriate constants.
- We adopted $\mathtt{u}$ and $\mathtt{u}^{-1}$ as constants, since $L \cdot \mathtt{u}$ and $L \cdot \mathtt{u}^{-1}$ can be computed efficiently from L by one shift and one conditional XOR, respectively, as shown in (2) and (3).
- One might try to use $\mathtt{Cst}_1 \quad 1$ instead of $\mathtt{Cst}_1 \quad \mathtt{u}$. In this case, the fourth condition in Sec. 3 is not satisfied, and in fact, the scheme can be easily attacked. Similarly, if one uses $\mathtt{Cst}_2 \quad 1$ instead of $\mathtt{Cst}_2 \quad \mathtt{u}^{-1}$, the fifth condition in Sec. 3 is not satisfied, and the scheme can be easily attacked. Therefore, we can not use "1" as a constant.

6.5 On Standard Key Separation Technique

For XCBC, assume that we want to use a single key K of E, where E is the AES.

Then the following key separation technique is suggested in [4]. Let K be a k-bit AES key. Then

$$\begin{aligned} &K_1 \quad \text{the first } k \text{ bits of } \mathrm{AES}_K(C_{1a}) \circ \mathrm{AES}_K(C_{1b}), \\ &K_2 = \mathrm{A\ ES}_K(C_2) \quad \text{and} \\ &K_3 = \mathrm{A\ ES}_K(C_3) \end{aligned}$$

for some distinct constants C_{1a}, C_{1b}, C_2 and C_3. We call it XCBC+`kst` (key separation technique). XCBC+`kst` uses one k-bit key. However, it requires *additional* one key scheduling of AES and *additional* 3 or 4 AES invocations during the pre-processing time.

Similar discussion can be applied to TMAC. For example, we can let

$$\begin{aligned} &K_1 \quad \text{the first } k \text{ bits of } \mathrm{AES}_K(C_{1a}) \circ \mathrm{AES}_K(C_{1b}), \text{ and} \\ &K_2 = \mathrm{A\ ES}_K(C_2) \end{aligned}$$

for some distinct constants C_{1a}, C_{1b} and C_2. We call it TMAC+`kst`.

We note that OMAC does *not* need such a key separation technique since its key length is k bits in its own form (without using any key separation technique). This saves storage space and pre-processing time compared to XCBC+`kst` and TMAC+`kst`.

6.6 Comparison

Let $E : \{0\ 1\}^k \quad 0\ 1\}^n \quad 0\ 1\}^n$ be a block cipher, and $M \in \{0\ 1\}^*$ be a message. We show a comparison of CBC MAC and its variants in Table 3, where

- "K len." denotes the key length.
- "#K sche." denotes the number of block cipher key schedulings. For RMAC, it requires one block cipher key scheduling each time generating a tag.
- "#M" denotes the number messages which the sender has MACed.
- "#E invo." denotes the number of block cipher invocations to generate a tag for a message M, assuming $|M| \quad 0$.

- "$\#E$ pre." denotes the number of block cipher invocations during the pre-processing time. These block cipher invocations can be done without the message. For XCBC+`kst` and TMAC+`kst`, the block cipher is assumed to be the AES.

Table 3: A comparison of CBC MAC and its variants.

Name	Domain	K len.	$\#K$ sche.	$\#E$ invo.	$\#E$ pre.
CBC MAC	$(\{\ ,\ \}^n)^m$	k		$\|\ \ \|/n$	
EMAC	$(\{\ ,\ \}^n)^+$	k		$1+\|\ \ \|/n$	
RMAC	$\{\ ,\ \}^*$	k	$1+\ \#$	$1+\lceil(\|\ \ \|+1\ )/n\rceil$	
CBC	$\{\ ,\ \}^*$	$k+2\ n$		$\lceil\|\ \ \|/n\rceil$	
TMAC	$\{\ ,\ \}^*$	$k+n$		$\lceil\|\ \ \|/n\rceil$	
CBC+`kst`	$\{\ ,\ \}^*$	k		$\lceil\|\ \ \|/n\rceil$	3o r 4
TMAC+`kst`	$\{\ ,\ \}^*$	k		$\lceil\|\ \ \|/n\rceil$	r
OMAC	$\{\ ,\ \}^*$	k		$\lceil\|\ \ \|/n\rceil$	

6.7 MAC Truncation

It is possible to reduce the output length by truncating the value of $\mathrm{OMAC}_K(M)$. That is, let

$$\mathrm{OMAC}[\tau]_K(M) \quad \text{the first } \tau\text{-bits of } \mathrm{OMAC}_K(M)$$

Then we can prove a security bound similar to Theorem 5.1.

Corollary 6.1 *Let $E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$ be the underlying block cipher used in OMAC$[\tau]$. Then*

$$\mathtt{Adv}^{\mathsf{mac}}_{\mathrm{OMAC}[\tau]}(t, q\ \ nm) \quad \frac{(5m^2+1)q^2}{2^n} + \frac{1}{2^\tau} + \mathtt{Adv}^{\mathsf{prp}}_E(t', q')$$

where $t' \quad t + O(mq)$ and $q' \quad mq$.

(Lemma 5.1 and Corollary 5.1 for OMAC also hold for OMAC$[\tau]$.)

References

[1] M. Bellare, J. Kilian, and P. Rogaway. The security of the cipher block chaining message authentication code. *JCSS,* vol. 1, no. 3, 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] J. Black and P. Rogaway. Comments to NIST concerning AES modes of operations: A suggestion for handling arbitrary-length messages with the CBC MAC. *Second Modes of Operation Workshop.* Available at `http://www.cs.ucdavis.edu/~rogaway/`.

[5] J. Black and P. Rogaway. A block-cipher mode of operation for parallelizable message authentication. *Advances in Cryptology — EUROCRYPT 2002, LNCS 2332,* pp. 384–397, Springer-Verlag, 2002.

[6] 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.

[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. Cryptology ePrint Archive, Report 2001/180, November 25, 2002, `http://eprint.iacr.org/`.

[9] É. Jaulmes, A. Joux, and F. Valette. On the security of randomized CBC-MAC beyond the birthday paradox limit: A new construction. *Fast Software Encryption, FSE 2002, LNCS 2365,* pp. 237–251, Springer-Verlag, 2002. Full version is available at Cryptology ePrint Archive, Report 2001/074, `http://eprint.iacr.org/`.

[10] K. Kurosawa and T. Iwata. TMAC: Two-Key CBC MAC. Cryptology ePrint Archive, Report 2001/092, July 10, 2002, `http://eprint.iacr.org/`. To appear in *CT-RSA 2003, LNCS 2612,* pp. 33–49, Springer-Verlag, 2003.

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

[12] P. Rogaway. Bucket hashing and its application to fast message authentication. *Advances in Cryptology — CRYPTO '95, LNCS 963,* pp. 29–42, Springer-Verlag, 1995.

[13] P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: a block-cipher mode of operation for efficient authenticated encryption. *Proceedings of ACM Conference on Computer and Communications Security, ACM CCS 2001,* ACM, 2001.

[14] S. Vaudenay. Decorrelation over infinite domains: The encrypted CBC-MAC case. *Selected Areas in Cryptography, SAC 2000, LNCS 2012,* pp. 57–71, Springer-Verlag, 2001.

[15] S. Vaudenay. Decorrelation over infinite domains: The encrypted CBC-MAC case. *Communications in Information and Systems (CIS),* vol. 1, pp. 75–85, 2001.

Proof of Lemma 5.2

If A is a finite multiset then $\#A$ denotes the number of elements in A.

Let $\{a, b, c, \quad\}$ be a finite multiset of bit strings. That is, $a \in \{0\ 1\}^*$,$b \in \{0\ 1\}^*$,$c \in \{0\ 1\}^*$ hold. We say "$\{a, b, c, \quad\}$ are distinct" if there exists no element occurs twice or more. Equivalently, $\{a, b, c, \quad\}$ are distinct if any two elements in $\{a, b, c, \quad\}$ are distinct.

Before proving Lemma 5.2, we need the following lemma.

Lemma A.1 *Let q_1,q_2,q_3,q ,q ,q be six non-negative integers. For 1 i , let $x_i^{(1)}$,$x_i^{(q_i)}$ be fixed n-bit strings such that $\{x_i^{(1)} \quad x_i^{(q_i)}\}$ are distinct. Similarly, for 1 i , let $y_i^{(1)} \quad y_i^{(q_i)}$ be fixed n-bit strings such that $\{y_1^{(1)} \quad ,y_1^{(q_1)}\}$ $\{y_2^{(1)} \quad ,y_2^{(q_2)}\}$ are distinct, and $\{y_3^{(1)} \quad ,y_3^{(q_3)}\}$ $\{y^{(1)} \quad ,y^{(q_4)}\} \cup \{y^{(1)} \quad ,y^{(q_5)}\} \cup \{y^{(1)} \quad ,y^{(q_6)}\}$ are distinct. Let $P \in$*

Perm(n) *and* $\texttt{Rnd} \in \{0\ 1\}^n$. *Then the number of* $(P, \texttt{Rnd})$ *which satisfies*

$$
\begin{array}{lll}
Q_1(x_1^{(i)})= y_1^{(i)} & \text{for } 1\quad {}^\forall i \quad q_1, \\
Q_2(x_2^{(i)})= y_2^{(i)} & \text{for } 1\quad {}^\forall i \quad q_2, \\
Q_3(x_3^{(i)})= y_3^{(i)} & \text{for } 1\quad {}^\forall i \quad q_3, \\
Q\ \ (x^{(i)})= y^{(i)} & \text{for } 1\quad {}^\forall i \quad q\ , \\
Q\ \ (x^{(i)})= y^{(i)} & \text{for } 1\quad {}^\forall i \quad q \ \text{ and} \\
Q\ \ (x^{(i)})= y^{(i)} & \text{for } 1\quad {}^\forall i \quad q
\end{array}
\tag{8}
$$

is at least $(2^n - 2q - q^2) \cdot (2^n - q)!$, *where* $q \quad q_1 + \cdots + q$.

Proof (of Lemma A.1). At the top level, we consider two cases: $\texttt{Cst} \in \{x_1^{(1)} \quad , x_1^{(q_1)}\}$ and $\texttt{Cst} \in \{x_1^{(1)} \quad , x_1^{(q_1)}\}$.

Case 1: $\texttt{Cst} \in \{x_1^{(1)} \quad , x_1^{(q_1)}\}$**.** Let c be a unique integer such that $1 \quad c \quad q_1$ and $\texttt{Cst} \quad x_1^{(c)}$. Let l be an n-bit variable. First, observe that:

$$
\begin{array}{l}
\#\{l \mid 1 \quad {}^\exists i \quad q_1\ 1 \quad {}^\exists j \quad q_2, x_1^{(i)} \quad x_2^{(j)} \oplus y_1^{(c)} \oplus l\} \leq q_1 q_2, \\
\#\{l \mid 1 \quad {}^\exists i \quad q_1\ 1 \quad {}^\exists j \quad q_3, x_1^{(i)} \quad x_3^{(j)} \oplus y_1^{(c)} \oplus l \oplus l \cdot \texttt{u}\} \leq q_1 q_3, \\
\#\{l \mid 1 \quad {}^\exists i \quad q_1\ 1 \quad {}^\exists j \quad q\ , x_1^{(i)} \quad x^{(j)} \oplus y_1^{(c)} \oplus l \oplus l \cdot \texttt{u}^{-1}\} \leq q_1 q\ , \\
\#\{l \mid 1 \quad {}^\exists i \quad q_1\ 1 \quad {}^\exists j \quad q\ , x_1^{(i)} \quad x^{(j)} \oplus l \cdot \texttt{u}\} \leq q_1 q\ , \\
\#\{l \mid 1 \quad {}^\exists i \quad q_1\ 1 \quad {}^\exists j \quad q\ , x_1^{(i)} \quad x^{(j)} \oplus l \cdot \texttt{u}^{-1}\} \leq q_1 q\ , \\
\#\{l \mid 1 \quad {}^\exists i \quad q_2\ 1 \quad {}^\exists j \quad q_3, x_2^{(i)} \quad x_3^{(j)} \oplus l \cdot \texttt{u}\} \leq q_2 q_3, \\
\#\{l \mid 1 \quad {}^\exists i \quad q_2\ 1 \quad {}^\exists j \quad q\ , x_2^{(i)} \quad x^{(j)} \oplus l \cdot \texttt{u}^{-1}\} \leq q_2 q\ , \\
\#\{l \mid 1 \quad {}^\exists i \quad q_2\ 1 \quad {}^\exists j \quad q\ , x_2^{(i)} \oplus y_1^{(c)} \oplus l \quad x^{(j)} \oplus l \cdot \texttt{u}\} \leq q_2 q\ , \\
\#\{l \mid 1 \quad {}^\exists i \quad q_2\ 1 \quad {}^\exists j \quad q\ , x_2^{(i)} \oplus y_1^{(c)} \oplus l \quad x^{(j)} \oplus l \cdot \texttt{u}^{-1}\} \leq q_2 q\ , \\
\#\{l \mid 1 \quad {}^\exists i \quad q_3\ 1 \quad {}^\exists j \quad q\ , x_3^{(i)} \oplus l \cdot \texttt{u} \quad x^{(j)} \oplus l \cdot \texttt{u}^{-1}\} \leq q_3 q\ , \\
\#\{l \mid 1 \quad {}^\exists i \quad q_3\ 1 \quad {}^\exists j \quad q\ , x_3^{(i)} \oplus y_1^{(c)} \oplus l \quad x^{(j)}\} \leq q_3 q\ , \\
\#\{l \mid 1 \quad {}^\exists i \quad q_3\ 1 \quad {}^\exists j \quad q\ , x_3^{(i)} \oplus y_1^{(c)} \oplus l \oplus l \cdot \texttt{u} \quad x^{(j)} \oplus l \cdot \texttt{u}^{-1}\} \leq q_3 q\ , \\
\#\{l \mid 1 \quad {}^\exists i \quad q\ \ 1 \quad {}^\exists j \quad q\ , x^{(i)} \oplus y_1^{(c)} \oplus l \oplus l \cdot \texttt{u}^{-1} \quad x^{(j)} \oplus l \cdot \texttt{u}\} \leq q\ q\ , \\
\#\{l \mid 1 \quad {}^\exists i \quad q\ \ 1 \quad {}^\exists j \quad q\ , x^{(i)} \oplus y_1^{(c)} \oplus l \quad x^{(j)}\} \leq q\ q\ , \\
\#\{l \mid 1 \quad {}^\exists i \quad q\ \ 1 \quad {}^\exists j \quad q\ , x^{(i)} \oplus l \cdot \texttt{u} \quad x^{(j)} \oplus l \cdot \texttt{u}^{-1}\} \leq q\ q\ , \\
\#\{l \mid 1 \quad {}^\exists i \quad q_1\ 1 \quad {}^\exists j \quad q_3, y_1^{(i)} \oplus y_1^{(c)} \oplus l \quad y_3^{(j)}\} \leq q_1 q_3, \\
\#\{l \mid 1 \quad {}^\exists i \quad q_1\ 1 \quad {}^\exists j \quad q\ , y_1^{(i)} \oplus y_1^{(c)} \oplus l \quad y^{(j)}\} \leq q_1 q\ , \\
\#\{l \mid 1 \quad {}^\exists i \quad q_1\ 1 \quad {}^\exists j \quad q\ , y_1^{(i)} \oplus y_1^{(c)} \oplus l \quad y^{(j)}\} \leq q_1 q\ ,
\end{array}
$$

$$
\begin{aligned}
&\#\{l \mid 1 \quad {}^{\exists}i \quad q_1\ 1 \quad {}^{\exists}j \quad q\ , y_1^{(i)} \oplus y_1^{(c)} \oplus l \quad y^{(j)}\} \le\ q_1 q\ ,\\
&\#\{l \mid 1 \quad {}^{\exists}i \quad q_2\ 1 \quad {}^{\exists}j \quad q_3, y_2^{(i)} \oplus y_1^{(c)} \oplus l \quad y_3^{(j)}\} \le\ q_2 q_3,\\
&\#\{l \mid 1 \quad {}^{\exists}i \quad q_2\ 1 \quad {}^{\exists}j \quad q\ , y_2^{(i)} \oplus y_1^{(c)} \oplus l \quad y^{(j)}\} \le\ q_2 q\ ,\\
&\#\{l \mid 1 \quad {}^{\exists}i \quad q_2\ 1 \quad {}^{\exists}j \quad q\ , y_2^{(i)} \oplus y_1^{(c)} \oplus l \quad y^{(j)}\} \le\ q_2 q\ , \text{ and}\\
&\#\{l \mid 1 \quad {}^{\exists}i \quad q_2\ 1 \quad {}^{\exists}j \quad q\ , y_2^{(i)} \oplus y_1^{(c)} \oplus l \quad y^{(j)}\} \le\ q_2 q\ .
\end{aligned}
$$

Here we used the fact that we are working in a field (We will continue to use this without mention).

We now fix any l which is *not* included in any of the above twenty-three sets. We have at least $(2^n - (q_1q_2 + 2q_1q_3 + 2q_1q\ + 2q_1q\ + 2q_1q\ + 2q_2q_3 + 2q_2q\ + 2q_2q\ + 2q_2q\ + q_3q\ + q_3q\ + q_3q\ + q\ q\ + q\ q\ + q\ q\ )) \ge (2^n - q^2)$ choice of such l.

Now we let $L \leftarrow l$ and $\texttt{Rnd} \leftarrow l \oplus y_1^{(c)}$. Then we have

$$
\begin{aligned}
\{&x_1^{(1)} \qquad , x_1^{(q_1)}\\
&x_2^{(1)} \oplus \texttt{Rnd} \qquad , x_2^{(q_2)} \oplus \texttt{Rnd}\\
&x_3^{(1)} \oplus \texttt{Rnd} \oplus L\cdot\texttt{u} \qquad , x_3^{(q_3)} \oplus \texttt{Rnd} \oplus L\cdot\texttt{u}\\
&x^{(1)} \oplus \texttt{Rnd} \oplus L\cdot\texttt{u}^{-1} \qquad , x^{(q_4)} \oplus \texttt{Rnd} \oplus L\cdot\texttt{u}^{-1}\\
&x^{(1)} \oplus L\cdot\texttt{u} \qquad x^{(q_5)} \oplus L\cdot\texttt{u}\\
&x^{(1)} \oplus L\cdot\texttt{u}^{-1} \qquad , x^{(q_6)} \oplus L\cdot\texttt{u}^{-1}\}
\end{aligned}
$$

(which are inputs to P) are distinct. Also, the corresponding outputs

$$
\begin{aligned}
\{&y_1^{(1)} \oplus \texttt{Rnd} \qquad , y_1^{(q_1)} \oplus \texttt{Rnd}\\
&y_2^{(1)} \oplus \texttt{Rnd} \qquad , y_2^{(q_2)} \oplus \texttt{Rnd}\\
&y_3^{(1)} \qquad , y_3^{(q_3)}\\
&y^{(1)} \qquad , y^{(q_4)}\\
&y^{(1)} \qquad , y^{(q_5)}\\
&y^{(1)} \qquad , y^{(q_6)}\}
\end{aligned}
$$

are distinct. In other words, for P, the above $q_1 + q_2 + q_3 + q\ + q\ + q\ $ input-output pairs are determined. The remaining $2^n - (q_1 + q_2 + q_3 + q\ + q\ + q\ )$ input-output pairs are undetermined. Therefore we have $(2^n - (q_1 + q_2 + q_3 + q\ + q\ + q\ ))! \quad (2^n - q)!$ possible choice of P for any such fixed $(L, \texttt{Rnd})$.

Case 2: $\texttt{Cst} \in \{x_1^{(1)} \quad , x_1^{(q_1)}\}$**.** In this case, we count the number of $\texttt{Rnd}$ and L independently. Then similar to Case 1, observe that:

$$\#\{\mathtt{Rnd} \mid 1 \quad \exists i \quad q_2 \;\; \mathtt{Cst} \quad x_2^{(i)} \oplus \mathtt{Rnd}\} \leq \; q_2,$$
$$\#\{\mathtt{Rnd} \mid 1 \quad \exists i \quad q_1 \;\; 1 \quad \exists j \quad q_2, x_1^{(i)} \quad x_2^{(j)} \oplus \mathtt{Rnd}\} \leq \; q_1 q_2,$$
$$\#\{\mathtt{Rnd} \mid 1 \quad \exists i \quad q_3 \;\; 1 \quad \exists j \quad q \;\;, x_3^{(i)} \oplus \mathtt{Rnd} \quad x^{(j)}\} \leq \; q_3 q \;\;,$$
$$\#\{\mathtt{Rnd} \mid 1 \quad \exists i \quad q \;\; 1 \quad \exists j \quad q \;\;, x^{(i)} \oplus \mathtt{Rnd} \quad x^{(j)}\} \leq \; q \; q \;\;,$$
$$\#\{\mathtt{Rnd} \mid 1 \quad \exists i \quad q_1 \;\; 1 \quad \exists j \quad q_3, y_1^{(i)} \oplus \mathtt{Rnd} \quad y_3^{(j)}\} \leq \; q_1 q_3,$$
$$\#\{\mathtt{Rnd} \mid 1 \quad \exists i \quad q_1 \;\; 1 \quad \exists j \quad q \;\;, y_1^{(i)} \oplus \mathtt{Rnd} \quad y^{(j)}\} \leq \; q_1 q \;\;,$$
$$\#\{\mathtt{Rnd} \mid 1 \quad \exists i \quad q_1 \;\; 1 \quad \exists j \quad q \;\;, y_1^{(i)} \oplus \mathtt{Rnd} \quad y^{(j)}\} \leq \; q_1 q \;\;,$$
$$\#\{\mathtt{Rnd} \mid 1 \quad \exists i \quad q_1 \;\; 1 \quad \exists j \quad q \;\;, y_1^{(i)} \oplus \mathtt{Rnd} \quad y^{(j)}\} \leq \; q_1 q \;\;,$$
$$\#\{\mathtt{Rnd} \mid 1 \quad \exists i \quad q_2 \;\; 1 \quad \exists j \quad q_3, y_2^{(i)} \oplus \mathtt{Rnd} \quad y_3^{(j)}\} \leq \; q_2 q_3,$$
$$\#\{\mathtt{Rnd} \mid 1 \quad \exists i \quad q_2 \;\; 1 \quad \exists j \quad q \;\;, y_2^{(i)} \oplus \mathtt{Rnd} \quad y^{(j)}\} \leq \; q_2 q \;\;,$$
$$\#\{\mathtt{Rnd} \mid 1 \quad \exists i \quad q_2 \;\; 1 \quad \exists j \quad q \;\;, y_2^{(i)} \oplus \mathtt{Rnd} \quad y^{(j)}\} \leq \; q_2 q \;\;, \text{ and}$$
$$\#\{\mathtt{Rnd} \mid 1 \quad \exists i \quad q_2 \;\; 1 \quad \exists j \quad q \;\;, y_2^{(i)} \oplus \mathtt{Rnd} \quad y^{(j)}\} \leq \; q_2 q \;\;.$$

We fix any `Rnd` which is *not* included in any of the above twelve sets. We have at least $(2^n - (q_2 + q_1 q_2 + q_3 q \;\; + q \; q \;\; + q_1 q_3 + q_1 q \;\; + q_1 q \;\; + q_1 q \;\; + q_2 q_3 + q_2 q \;\; + q_2 q \;\; + q_2 q \;\;)) \geq (2^n - q - q^2/2)$ choice of such `Rnd`.

Next we see that:

$$\#\{L \mid 1 \quad \exists i \quad q_3 \;\; \mathtt{Cst} \quad x_3^{(i)} \oplus \mathtt{Rnd} \oplus L \cdot \mathtt{u}\} \leq \; q_3,$$
$$\#\{L \mid 1 \quad \exists i \quad q \;\; \mathtt{Cst} \quad x^{(i)} \oplus \mathtt{Rnd} \oplus L \cdot \mathtt{u}^{-1}\} \leq \; q \;\;,$$
$$\#\{L \mid 1 \quad \exists i \quad q \;\; \mathtt{Cst} \quad x^{(i)} \oplus L \cdot \mathtt{u}\} \leq \; q \;\;,$$
$$\#\{L \mid 1 \quad \exists i \quad q \;\; \mathtt{Cst} \quad x^{(i)} \oplus L \cdot \mathtt{u}^{-1}\} \leq \; q \;\;,$$
$$\#\{L \mid 1 \quad \exists i \quad q_1 \;\; 1 \quad \exists j \quad q_3, x_1^{(i)} \quad x_3^{(j)} \oplus \mathtt{Rnd} \oplus L \cdot \mathtt{u}\} \leq \; q_1 q_3,$$
$$\#\{L \mid 1 \quad \exists i \quad q_1 \;\; 1 \quad \exists j \quad q \;\;, x_1^{(i)} \quad x^{(j)} \oplus \mathtt{Rnd} \oplus L \cdot \mathtt{u}^{-1}\} \leq \; q_1 q \;\;,$$
$$\#\{L \mid 1 \quad \exists i \quad q_1 \;\; 1 \quad \exists j \quad q \;\;, x_1^{(i)} \quad x^{(j)} \oplus L \cdot \mathtt{u}\} \leq \; q_1 q \;\;,$$
$$\#\{L \mid 1 \quad \exists i \quad q_1 \;\; 1 \quad \exists j \quad q \;\;, x_1^{(i)} \quad x^{(j)} \oplus L \cdot \mathtt{u}^{-1}\} \leq \; q_1 q \;\;,$$
$$\#\{L \mid 1 \quad \exists i \quad q_2 \;\; 1 \quad \exists j \quad q_3, x_2^{(i)} \quad x_3^{(j)} \oplus L \cdot \mathtt{u}\} \leq \; q_2 q_3,$$
$$\#\{L \mid 1 \quad \exists i \quad q_2 \;\; 1 \quad \exists j \quad q \;\;, x_2^{(i)} \quad x^{(j)} \oplus L \cdot \mathtt{u}^{-1}\} \leq \; q_2 q \;\;,$$
$$\#\{L \mid 1 \quad \exists i \quad q_2 \;\; 1 \quad \exists j \quad q \;\;, x_2^{(i)} \oplus \mathtt{Rnd} \quad x^{(j)} \oplus L \cdot \mathtt{u}\} \leq \; q_2 q \;\;,$$
$$\#\{L \mid 1 \quad \exists i \quad q_2 \;\; 1 \quad \exists j \quad q \;\;, x_2^{(i)} \oplus \mathtt{Rnd} \quad x^{(j)} \oplus L \cdot \mathtt{u}^{-1}\} \leq \; q_2 q \;\;,$$
$$\#\{L \mid 1 \quad \exists i \quad q_3 \;\; 1 \quad \exists j \quad q \;\;, x_3^{(i)} \oplus L \cdot \mathtt{u} \quad x^{(j)} \oplus L \cdot \mathtt{u}^{-1}\} \leq \; q_3 q \;\;,$$
$$\#\{L \mid 1 \quad \exists i \quad q_3 \;\; 1 \quad \exists j \quad q \;\;, x_3^{(i)} \oplus \mathtt{Rnd} \oplus L \cdot \mathtt{u} \quad x^{(j)} \oplus L \cdot \mathtt{u}^{-1}\} \leq \; q_3 q \;\;,$$
$$\#\{L \mid 1 \quad \exists i \quad q \;\; 1 \quad \exists j \quad q \;\;, x^{(i)} \oplus \mathtt{Rnd} \oplus L \cdot \mathtt{u}^{-1} \quad x^{(j)} \oplus L \cdot \mathtt{u}\} \leq \; q \; q \;\;,$$
$$\#\{L \mid 1 \quad \exists i \quad q \;\; 1 \quad \exists j \quad q \;\;, x^{(i)} \oplus L \cdot \mathtt{u} \quad x^{(j)} \oplus L \cdot \mathtt{u}^{-1}\} \leq \; q \; q \;\;,$$
$$\#\{L \mid 1 \quad \exists i \quad q_1, L \quad y_1^{(i)} \oplus \mathtt{Rnd}\} \leq \; q_1,$$

$$
\begin{aligned}
&\#\{L \mid 1 \quad {}^{\exists}i \quad q_2, L \quad y_2^{(i)} \oplus \mathtt{Rnd}\} \leq \; q_2,\\
&\#\{L \mid 1 \quad {}^{\exists}i \quad q_3, L \quad y_3^{(i)}\} \leq \; q_3,\\
&\#\{L \mid 1 \quad {}^{\exists}i \quad q\;, L \quad y^{(i)}\} \leq \; q\;,\\
&\#\{L \mid 1 \quad {}^{\exists}i \quad q\;, L \quad y^{(i)}\} \leq \; q\;, \text{ and}\\
&\#\{L \mid 1 \quad {}^{\exists}i \quad q\;, L \quad y^{(i)}\} \leq \; q\;.
\end{aligned}
$$

We now fix any L which is *not* included in any of the above twenty-two sets. We have at least $(2^n-(q_1q_3+q_1q\; +q_1q\; +q_1q\; +q_2q_3+q_2q\; +q_2q\; +q_2q\; + q_3q\; +q_3q\; +q\;q\; +q\;q\; +q_1+q_2+2q_3+2q\; +2q\; +2q\;)) \geq (2^n-2q-q^2/2)$ choice of such L.

Then we have

$$
\begin{aligned}
\{&\mathtt{Cst}\\
&x_1^{(1)} \qquad ,x_1^{(q_1)}\\
&x_2^{(1)} \oplus \mathtt{Rnd} \qquad ,x_2^{(q_2)} \oplus \mathtt{Rnd}\\
&x_3^{(1)} \oplus \mathtt{Rnd} \oplus L\cdot\mathtt{u} \qquad ,x_3^{(q_3)} \oplus \mathtt{Rnd} \oplus L\cdot\mathtt{u}\\
&x^{(1)} \oplus \mathtt{Rnd} \oplus L\cdot\mathtt{u}^{-1} \qquad ,x^{(q_4)} \oplus \mathtt{Rnd} \oplus L\cdot\mathtt{u}^{-1}\\
&x^{(1)} \oplus L\cdot\mathtt{u} \qquad x^{(q_5)} \oplus L\cdot\mathtt{u}\\
&x^{(1)} \oplus L\cdot\mathtt{u}^{-1} \qquad ,x^{(q_6)} \oplus L\cdot\mathtt{u}^{-1}\}
\end{aligned}
$$

(which are inputs to P) are distinct. Also, the corresponding outputs

$$
\begin{aligned}
\{&L,\\
&y_1^{(1)} \oplus \mathtt{Rnd} \qquad ,y_1^{(q_1)} \oplus \mathtt{Rnd}\\
&y_2^{(1)} \oplus \mathtt{Rnd} \qquad ,y_2^{(q_2)} \oplus \mathtt{Rnd}\\
&y_3^{(1)} \qquad ,y_3^{(q_3)}\\
&y^{(1)} \qquad ,y^{(q_4)}\\
&y^{(1)} \qquad ,y^{(q_5)}\\
&y^{(1)} \qquad ,y^{(q_6)}\}
\end{aligned}
$$

are distinct. In other words, for P, the above $1+q_1+q_2+q_3+q\; +q\; +q\;$ input-output pairs are determined. The remaining $2^n-(1+q_1+q_2+q_3+q\; +q\; +q\;)$ input-output pairs are undetermined. Therefore we have $(2^n-(1+q_1+q_2+q_3+q\; +q\; +q\;))! \quad (2^n-(1+q))!$ possible choice of P for any such fixed $(L, \mathtt{Rnd})$.

Completing the Proof. In Case 1, we have at least $(2^n-q^2)\cdot(2^n-q)!$ choice of $(P, \mathtt{Rnd})$ which satisfies (8).

In Case 2, we have at least $(2^n - q - q^2/2) \cdot (2^n - 2q - q^2/2) \cdot (2^n - (1+q))!$ choice of $(P, \texttt{Rnd})$ which satisfies (8). This bound is at least $(2^n - 2q - q^2) \cdot (2^n - q)!$.

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

We now prove Lemma 5.2.

Proof (of Lemma 5.2). For $1 \quad i \quad 6$, let $\mathcal{O}_i$ be either Q_i or P_i. The adversary $\mathcal{A}$ has oracle access to $\mathcal{O}_1 \quad \mathcal{O}$. Since $\mathcal{A}$ is computationally unbounded, there is no loss of generality to assume that $\mathcal{A}$ is deterministic.

There are six types of queries $\mathcal{A}$ can make: $(\mathcal{O}_j, x)$ which denotes the query "what is $\mathcal{O}_j(x)$?" For the i-th query $\mathcal{A}$ makes to $\mathcal{O}_j$, define the query-answer pair $(x_j^{(i)}, y_j^{(i)}) \in \{0 \ 1\}^n \quad 0 \ 1\}^n$, where $\mathcal{A}$'s query was $(\mathcal{O}_j, x_j^{(i)})$ and the answer it got was $y_j^{(i)}$.

Suppose that we run $\mathcal{A}$ with oracles $\mathcal{O}_1 \quad \mathcal{O}$. For this run, assume that $\mathcal{A}$ made q_j queries to $\mathcal{O}_j(\cdot)$, where $q_1 + \cdots + q \quad q$. For this run, we define view v of $\mathcal{A}$ as

$$\begin{array}{ll} v \ ^{\text{ef}} & \langle (y_1^{(1)} \quad y_1^{(q_1)}) \ (y_2^{(1)} \quad y_2^{(q_2)}) \ (y_3^{(1)} \quad , y_3^{(q_3)}) \\ & (y^{(1)} \quad y^{(q_4)}) \ (y^{(1)} \quad y^{(q_5)}) \ (y^{(1)} \quad , y^{(q_6)}) \rangle \end{array} \tag{9}$$

For this view, we always have:

$$\text{For } 1 \quad j \quad 6, \ \{y_j^{(1)} \quad , y_j^{(q_j)}\} \text{ are distinct.}$$

We note that since $\mathcal{A}$ never repeats a query, for the corresponding queries, we have:

$$\text{For } 1 \quad j \quad 6, \ \{x_j^{(1)} \quad , x_j^{(q_j)}\} \text{ are distinct.}$$

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 oracles), then

- $\mathcal{A}$'s queries are uniquely determined,
- $q_1 \quad , q$ are uniquely determined,
- the parsing of V into the format defined in (9) 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 $\quad {}^{\forall}i < {}^{\forall}j \quad q$, the i-th n-bit block of V $\quad$ the j-th n-bit block of V.

Note that if $V \in \boldsymbol{V}_{good}$ then the corresponding parsing v satisfies:

- $\{y_1^{(1)} \quad , y_1^{(q_1)}\} \quad \{y_2^{(1)} \quad , y_2^{(q_2)}\}$ are distinct, and
- $\{y_3^{(1)} \quad , y_3^{(q_3)}\} \cup \{y^{(1)} \quad , y^{(q_4)}\} \cup \{y^{(1)} \quad , y^{(q_5)}\} \cup \{y^{(1)} \quad , y^{(q_6)}\}$ 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

$$\#\{V \mid V \in (\boldsymbol{V}_{one} \cap \boldsymbol{V}_{good})\} \geq N_{one} - \binom{q}{2}\frac{2^{qn}}{2^n} \tag{10}$$

Evaluation of p_{rand}. We first evaluate

$$\begin{aligned} p_{rand} \quad \text{ef} \quad & \Pr(P_1 \quad , P \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{P_1(\cdot) \quad , P_6(\cdot)} = 1 \;) \\ & \frac{\#\{(P_1 \quad , P \;) \mid \mathcal{A}^{P_1(\cdot) \quad , P_6(\cdot)} = 1 \;\}}{\{(2^n)!\}} \end{aligned}$$

For each $V \in \boldsymbol{V}_{one}$, the number of $(P_1 \quad , P \;)$ such that

$$\text{For } 1 \quad j \quad 6,\; P_j(x_j^{(i)}) = \; y_j^{(i)} \text{ for } 1 \quad {}^{\forall}i \quad q_j, \tag{11}$$

is exactly $\prod_{1 \leq j \leq} (2^n - q_j)!$, which is at most $(2^n - q)! \cdot \{(2^n)!\}$. Therefore, we have

$$\begin{aligned} p_{rand} \quad & \sum_{V \in \boldsymbol{V}_{one}} \frac{\#\{(P_1 \quad , P \;) \mid (P_1 \quad , P \;) \text{ satisfying (11)}\}}{\{(2^n)!\}} \\ & \sum_{V \in \boldsymbol{V}_{one}} \frac{(2^n - q)!}{(2^n)!} \\ & N_{one} \cdot \frac{(2^n - q)!}{(2^n)!} \end{aligned}$$

Evaluation of p_{real}. We next evaluate

$$
\begin{aligned}
p_{real} \quad &{}^{\mathrm{ef}} \quad \Pr(P \xleftarrow{R} \mathrm{Perm}(n); \mathtt{Rnd} \xleftarrow{R} \{0\ \ 1\}^n : \mathcal{A}^{Q_1(\cdot) \quad ,Q_6(\cdot)} =1\ ) \\
&\quad \frac{\#\{(P, \mathtt{Rnd}) \mid \mathcal{A}^{Q_1(\cdot) \quad ,Q_6(\cdot)} =1\ \}}{(2^n)! \cdot 2^n}
\end{aligned}
$$

Then from Lemma A.1, we have

$$
\begin{aligned}
p_{real} \quad &\geq \quad \mathop{}_{V\in(\boldsymbol{V}_{one}\cap\boldsymbol{V}_{good})} \frac{\#\,\{(P, \mathtt{Rnd}) \mid (P, \mathtt{Rnd}) \text{ satisfying } (8)\}}{(2^n)! \cdot 2^n} \\
&\geq \quad \mathop{}_{V\in(\boldsymbol{V}_{one}\cap\boldsymbol{V}_{good})} \frac{(2^n-q)!}{(2^n)!} \cdot \quad 1 - \frac{2q+q^2}{2^n}
\end{aligned}
$$

Completing the Proof. From (10) we have

$$
\begin{aligned}
p_{real} \quad &\geq \quad N_{one} - \quad \begin{matrix} q \\ 2 \end{matrix} \ \frac{2^{qn}}{2^n} \quad \cdot \frac{(2^n-q)!}{(2^n)!} \cdot \quad 1 - \frac{2q+q^2}{2^n} \\
&\geq \quad p_{rand} - \quad \begin{matrix} q \\ 2 \end{matrix} \ \frac{2^{qn}}{2^n} \cdot \frac{(2^n-q)!}{(2^n)!} \quad \cdot \quad 1 - \frac{2q+q^2}{2^n} \qquad (12)
\end{aligned}
$$

Since $2^{qn} \cdot \frac{(2\ \ -q)!}{(2\ \ )!} \geq 1$, from (12), we have

$$
\begin{aligned}
p_{real} \quad &\geq \quad \left(p_{rand} - \frac{q(q-1)}{2 \cdot 2^n}\right) \cdot \quad 1 - \frac{2q+q^2}{2^n} \\
&\geq \quad p_{rand} - \frac{3q^2+3q}{2 \cdot 2^n} \\
&\geq \quad p_{rand} - \frac{3q^2}{2^n} \qquad (13)
\end{aligned}
$$

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

$$
1 - p_{real} \geq 1 - p_{rand} - \frac{3q^2}{2^n} \qquad (14)
$$

Finally, (13) and (14) give $|p_{real} - p_{rand}| \leq \frac{3q^2}{2}$. Q.E.D.

B Proo o Lemma 5.3

Let S and S' be distinct bit strings such that $|S| \quad sn$ for some $s \geq 1$, and $|S'| \quad s'n$ for some $s' \geq 1$. Define $V_n(S, S') \stackrel{\text{ef}}{=} \text{P r}(P_2 \stackrel{R}{\leftarrow} \text{Perm}(n) : \text{CBC}_{P_2}(S) = \text{CBC}_{P_2}(S'))$. Then the following proposition is known [3].

Proposition B.1 (Black and Rogaway [3]) *Let S and S' be distinct bit strings such that $|S| \quad sn$ for some $s \geq 1$, and $|S'| \quad s'n$ for some $s' \geq 1$. Assume that $s, s' \quad 2^n/4$. Then*

$$V_n(S, S') \quad \frac{(s + s')^2}{2^n}$$

Now let M and M' be distinct bit strings such that $|M| \quad mn$ for some $m \geq 2$, and $|M'| \quad m'n$ for some $m' \geq 2$. Define $W_n(M, M') \stackrel{\text{ef}}{}$ $\text{Pr}(P_1 \quad , P \stackrel{R}{\leftarrow} \text{Perm}(n) : \text{MOMAC}_{P_1 \quad , P_6}(M) = \text{M OMAC}_{P_1 \ldots, P_6}(M'))$. We note that P and P are irrelevant in the event $\text{MOMAC}_{P_1 \ldots, P_6}(M) =$ $\text{MOMAC}_{P_1 \quad , P_6}(M')$ since M and M' are both longer than n-bits. Also, P is irrelevant in the above event since $|M|$ and $|M'|$ are both multiples of n. Further, P_3 is irrelevant in the above event since it is invertible, and thus, there is a collision if and only if there is a collision at the input to the last encryption.

We show the following lemma.

Lemma B.1 (MOMAC Collision Bound) *Let M and M' be distinct bit strings such that $|M| \quad mn$ for some $m \geq 2$, and $|M'| \quad m'n$ for some $m' \geq 2$. Assume that $m, m' \quad 2^n/4$. Then*

$$W_n(M, M') \quad \frac{(m + m')^2}{2^n}$$

Proof. Let $M[1] \cdots M[m]$ and $M'[1] \cdots M'[m']$ be partitions of M and M' respectively. We consider two cases: $M[1] \quad M'[1]$ and $M[1] \quad M'[1]$.

Case 1: $M[1] \quad M'[1]$**.** In this case, Let P_1 be any permutation in $\text{Perm}(n)$, and let $S \leftarrow (P_1(M[1]) \oplus M[2]) \circ M[3] \circ \cdots \circ M[m]$ and $S' \leftarrow (P_1(M'[1]) \oplus M'[2]) \circ M'[3] \circ \cdots \circ M'[m']$. Observe that $\text{MOMAC}_{P_1 \ldots, P_6}(M) =$ $\text{MOMAC}_{P_1 \quad , P_6}(M')$ if and only if $\text{CBC}_{P_2}(S) = \text{CBC}_{P_2}(S')$, since we may ignore the last encryptions in $\text{CBC}_{P_2}(S)$ and $\text{CBC}_{P_2}(S')$. Therefore

$$W_n(M, M') \quad V_n(S, S') \quad \frac{(m + m' - 2)^2}{2^n}$$

Case 2: $M[1] \quad M'[1]$**.** In this case, we split into two cases: $P_1(M[1]) \oplus M[2] \quad P_1(M'[1]) \oplus M'[2]$ and $P_1(M[1]) \oplus M[2] \quad P_1(M'[1]) \oplus M'[2]$. The former event will occur with probability at most 1. The later one will occur with probability at most $\frac{1}{2 \; -1}$, which is at most $\frac{2}{2}$. Then it is not hard to see that

$$W_n(M, M') \quad 1 \cdot V_n(S, S') + \frac{2}{2^n} \quad \frac{(m+m-2)^2}{2^n} + \frac{2}{2^n} \quad \frac{(m+m')^2}{2^n}$$

by applying the similar argument as in Case 1. Q.E.D.

Let m be an integer such that $m \quad 2^n/4$. We consider the following four sets.

$$\begin{aligned}
D_1 &\stackrel{\text{def}}{} \{M \mid M \in \{0 \;\, 1\}^*,\ n< \ |M| \leq \ mn \text{ and } |M| \text{ is a multiple of } n\} \\
D_2 &\stackrel{\text{def}}{} \{M \mid M \in \{0 \;\, 1\}^*,\ n< \ |M| \leq \ mn \text{ and } |M| \text{ is not a multiple of } n\} \\
D_3 &\stackrel{\text{ef}}{} \{M \mid M \in \{0 \;\, 1\}^* \text{ and } |M| \quad n\} \\
D \; &\stackrel{\text{ef}}{} \{M \mid M \in \{0 \;\, 1\}^* \text{ and } |M| \quad \} \qquad <n
\end{aligned}$$

We next show the following lemma.

Lemma B.2 *Let* q_1, q_2, q_3, q *be four non-negative integers. For* $1 \quad i \quad 4$*, let* $M_i^{(1)} \quad , M_i^{(q_i)}$ *be fixed bit strings such that* $M_i^{(j)} \in D_i$ *for* $1 \quad j \quad q_i$ *and* $\{M_i^{(1)} \quad , M_i^{(q_i)}\}$ *are distinct. Similarly, for* $1 \quad i \quad 4$*, let* $T_i^{(1)} \quad , T_i^{(q_i)}$ *be fixed n-bit strings such that* $\{T_i^{(1)} \quad , T_i^{(q_i)}\}$ *are distinct. Then the number of* $P_1 \quad , P \ \in$ *Perm*(n) *such that*

$$\begin{aligned}
&MOMAC_{P_1 \quad , P_6}(M_1^{(i)}) = \ T_1^{(i)} \text{ for } 1 \quad \forall i \quad q_1, \\
&MOMAC_{P_1 \quad , P_6}(M_2^{(i)}) = \ T_2^{(i)} \text{ for } 1 \quad \forall i \quad q_2, \\
&MOMAC_{P_1 \quad , P_6}(M_3^{(i)}) = \ T_3^{(i)} \text{ for } 1 \quad \forall i \quad q_3 \text{ and} \\
&MOMAC_{P_1 \quad , P_6}(M^{(i)}) = \ T^{(i)} \text{ for } 1 \quad \forall i \quad q
\end{aligned} \tag{15}$$

is at least $\{(2^n)!\} \ \left(1 - \frac{2q^2m^2}{2}\right) \cdot \frac{1}{2}$*, where* $q \quad q_1 + \cdots + q$ *.*

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

$$\text{MOMAC}_{P_1 \, .., P_6}(M_1^{(i)}) = \text{M OMAC}_{P_1 \quad , P_6}(M_1^{(j)}) \text{ for } 1 \quad \exists i < \ \exists j \quad q_1$$

is at most $\{(2^n)!\}^2 \cdot \ \frac{q_1}{2} \ \cdot \frac{m^2}{2}$ from Lemma B.1. Note that $P_3 \quad , P$ are irrelevant in the above event.

We next consider $M_2^{(1)} \quad ,M_2^{(q_2)}$. The number of (P_1,P_2) such that

$$\mathrm{MOMAC}_{P_1\ ..,P_6}(M_2^{(i)}) = \mathrm{M\ OMAC}_{P_1 \quad ,P_6}(M_2^{(j)}) \text{ for } 1 \quad {}^{\exists}i < {}^{\exists}j \quad q_2$$

is at most $\{(2^n)!\}^2 \cdot \ {q_2 \atop 2} \ \cdot \frac{m^2}{2}$ from Lemma B.1.

Now we fix any (P_1,P_2) which is not like the above. We have at least $\{(2^n)!\}^2 \ \ 1 - \ {q_1 \atop 2} \ \cdot \frac{m^2}{2} - \ {q_2 \atop 2} \ \cdot \frac{m^2}{2}$ choice.

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 are distinct. Also, the corresponding outputs $\{T_3^{(1)} \quad ,T_3^{(q_3)}\}$ are distinct, and $\{T^{(1)} \quad ,T^{(q_4)}\}$ are distinct. We know that the inputs to P are distinct, and the corresponding outputs $\{T_3^{(1)} \quad ,T_3^{(q_3)}\}$ are distinct. Also, the inputs to P are distinct, and and the corresponding outputs $\{T^{(1)} \quad ,T^{(q_4)}\}$ are distinct. Therefore, we have at least $\{(2^n)!\}^2 \ \ 1 - \ {q_1 \atop 2} \ \cdot \frac{m^2}{2} - \ {q_2 \atop 2} \ \cdot \frac{m^2}{2} \ \cdot (2^n - q_1)! \cdot (2^n - q_2)! \cdot (2^n - q_3)! \cdot (2^n - q \ )!$ choice of $P_1 \quad ,P$ which satisfies (15). This bound is at least $\{(2^n)!\} \quad 1 - \frac{2q^2m^2}{2} \ \cdot \frac{1}{2}$ since $(2^n - q_i)! \geq \frac{(2\ )!}{2\ i}$.

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

We now prove Lemma 5.3.

Proof (of Lemma 5.3). Let $\mathcal{O}$ be either $\mathrm{MOMAC}_{P_1 \quad ,P_6}$ or R. Since $\mathcal{A}$ is computationally unbounded, there is no loss of generality to assume that $\mathcal{A}$ is deterministic.

For the query $\mathcal{A}$ makes to the oracle $\mathcal{O}$, define the query-answer pair $(M_j^{(i)},T_j^{(i)}) \in D_j \quad 0\ 1\}^n$, 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 $1 \quad j \quad 4$ and $q_1 + \cdots + q \quad q$. For this run, we define view v of $\mathcal{A}$ as

$$v \ {}^{\mathrm{ef}} \quad \begin{array}{l} (T_1^{(1)} \quad ,T_1^{(q_1)}) \ (T_2^{(1)} \quad ,T_2^{(q_2)}) \\ (T_3^{(1)} \quad ,T_3^{(q_3)}) \ (T^{(1)} \quad ,T^{(q_4)}) \end{array} \tag{16}$$

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,

- $q_1 \quad ,q$ are uniquely determined,
- the parsing of V into the format defined in (16) 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 $N_{one} \stackrel{\mathrm{def}}{=} \# \boldsymbol{V}_{one}$. Also, let $\boldsymbol{V}_{good}$ be a set of all qn-bit strings V such that:

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

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

$$\#\{V \mid V \in (\boldsymbol{V}_{one} \cap \boldsymbol{V}_{good})\} \geq N_{one} - \binom{q}{2} \frac{2^{qn}}{2^n} \tag{17}$$

Evaluation of p_{rand}. We first evaluate

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

Then it is not hard to see

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

Evaluation of p_{real}. We next evaluate

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

Then from Lemma B.2, we have

$$\begin{aligned} p_{real} & \geq \sum_{V \in (\boldsymbol{V}_{one} \quad \boldsymbol{V}_{good})} \frac{\#\{(P_1 \quad ,P) \mid (P_1 \quad ,P) \text{ satisfying (15)}\}}{\{(2^n)!\}} \\ & \geq \sum_{V \in (\boldsymbol{V}_{one} \quad \boldsymbol{V}_{good})} \left(1 - \frac{2q^2m^2}{2^n}\right) \cdot \frac{1}{2^{qn}} \end{aligned}$$

Completing the Proof. From (17) we have

$$
\begin{aligned}
p_{real} \quad \geq \quad & N_{one} - \binom{q}{2} \frac{2^{qn}}{2^n} \cdot 1 - \frac{2q^2m^2}{2^n} \cdot \frac{1}{2^{qn}} \\
& p_{rand} - \binom{q}{2} \frac{1}{2^n} \cdot 1 - \frac{2q^2m^2}{2^n} \\
\geq \quad & p_{rand} - \binom{q}{2} \frac{1}{2^n} - \frac{2q^2m^2}{2^n} \\
\geq \quad & p_{rand} - \frac{2q^2m^2 + q^2}{2^n} \qquad (18)
\end{aligned}
$$

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

$$1 - p_{real} \geq 1 - p_{rand} - \frac{2q^2m^2 + q^2}{2^n} \qquad (19)$$

Finally, (18) and (19) give $|p_{real} - p_{rand}| \leq \frac{2q^2m^2+q^2}{2}$. Q.E.D.

C Document History

- November 25, 2002. First version of the OMAC document submitted to IACR ePrint [8].
- December 20, 2002. Second version of the OMAC document submitted to NIST. Section 6, Appendix C and Appendix D are added.

D Intellectual Propert Statement

We do not have any intellectual property claims related to OMAC.