A Concrete Security Analysis for 3GPP-MAC

Dowon Hong[1] Ju-Sung Kang[1] Bart Preneel[2] Heuisu Ryu[1]

[1]Information Security Technology Division, ETRI
161 Kajong-Dong, Yusong-Gu, Taejon, 305-350, KOREA
{dwhong,jskang,hsryu}@etri.re.kr

[2]Katholieke Universiteit Leuven, ESAT/COSIC
Kasteelpark Arenberg 10, B-3001, Leuven-Heverlee, Belgium
Bart.Preneel@esat.kuleuven.ac.be

Abstract. The standardized integrity algorithm $f9$ of the 3GPP algorithm computes a MAC (Message Authentication Code) to establish the integrity and the data origin of the signalling data over a radio access link of W-CDMA IMT-2000. The function $f9$ is based on the block cipher KASUMI and it can be considered as a variant of CBC-MAC. In this paper we examine the provable security of $f9$. We prove that $f9$ is a secure pseudorandom function by giving a concrete bound on an adversary's inability to forge a MAC value in terms of her inability to distinguish the underlying block cipher from a random permutation.

Keywords: Message authentication code, 3GPP-MAC, Provable security, Pseudo-randomness.

1 Introduction

Within the security architecture of 3GPP (the 3rd Generation Partnership Project) a standardized data authentication algorithm $f9$ has been defined; this MAC (Message Authentication Code) algorithm is a variant of the standard CBC-MAC (Cipher Block Chaining) based on the block cipher KASUMI [22]. We refer to this MAC algorithm as "3GPP-MAC." The purpose of this work is to provide a proof of security for the 3GPP-MAC algorithm.

Providing a security proof in the sense of reduction-based cryptography intuitively means that one proves the following statement: if there exists an adversary $\mathcal{A}$ that breaks a given MAC built from a block cipher E, then there exists a corresponding adversary $\mathcal{A}'$ that breaks the block cipher E. The provable security treatment of MACs based on a block cipher started by Bellare *et al.* [1]. They have provided such a security proof for CBC-MAC. However, their proof is restricted to the case where the input messages are of fixed length. It is well known that CBC-MAC is not secure when the message length is variable [1]. A matching birthday attack has been described by Preneel and van Oorschot in [17]. Petrank and Rackoff [16] were the first to rigorously address the issue of message length variability. They provided a security proof for EMAC (Encrypted

CBC-MAC) which handles messages of variable unknown lengths. Black and Rogaway [3] introduced three refinements to EMAC that improve the efficiency. They also provided a new security proof by using new techniques which treat EMAC as an instance of the Carter-Wegman paradigm [5, 20]. Jaulmes, Joux, and Valette [7] proposed RMAC (Randomized MAC) which is an extension of EMAC. They showed that the security of RMAC improves over the birthday bound of [17] in the ideal-cipher model. This is not a reduction-based provable security result. Note that RMAC is currently being considered for standardization by NIST. However, recently it has been demonstrated that RMAC is vulnerable to related-key attacks [12–14]. Furthermore, it has been shown that it is not possible to provide a proof of security for the salted variant of RMAC [19]. Black and Rogaway [4, 18] have proposed a parallelizable block cipher mode of operation for message authentication (PMAC) together with a security proof. Several other new modes, such as XECB-MAC [6] and TMAC [8] have been submitted to NIST for consideration, but they will probably not be included in the standard [24].

The security evaluation of 3GPP-MAC has primarily been performed by the 3GPP SAGE group (Security Algorithms Group of Experts) [21]. Based on some ad hoc analysis, the general conclusion of [21] is that 3GPP-MAC does not exhibit any security weaknesses. Recently, Knudsen and Mitchell [11] analyzed 3GPP-MAC from the viewpoint of a birthday attack. They have described several types of forgery and key recovery attacks for 3GPP-MAC; they have also shown that key recovery attacks are infeasible: the most efficient attack requires around 3×2^{48} chosen messages. We believe that it is important to provide a security proof for a MAC algorithm based on an information theoretic and a complexity theoretic analysis. Such a security proof can be considered as a theoretical evidence of the soundness of the overall structure of a MAC algorithm. However so far no security proof has been provided in the literature for 3GPP-MAC. This observation motivates this paper.

In this paper we prove that 3GPP-MAC is secure in the sense of reduction-based cryptography. More specifically, we prove that 3GPP-MAC is a pseudorandom function which means that no attacker with polynomially many queries can distinguish 3GPP-MAC from a perfect random function; by using this fact, we show that 3GPP-MAC is a secure MAC algorithm under the assumption that the underlying block cipher is a pseudorandom permutation. This assumption is a reasonable one since the pseudorandomness of the 3GPP block cipher KASUMI has recently been investigated by Kang *et al.* [9, 10]. We do not address the question whether the distinguishing bound we have obtained is sufficiently tight or not. We leaves this as an open problem.

2 Preliminaries

2.1 Notation

Let $\{0,1\}^n$ denote the set of all n-bit strings, and $\{0,1\}^{n*}$ be the set of all binary strings whose bit-lengths are positive multiples of n. Let $\mathcal{R}_{n* \to l}$ be the set of all functions $\lambda : \{0,1\}^{n*} \to \{0,1\}^l$, $\mathcal{P}_n$ be the set of all permutations $\pi : \{0,1\}^n \to \{0,1\}^n$, and $\mathcal{K}$ be the key space which is the set of all possible key values K.

For any given key space $\mathcal{K}$, message space $\{0,1\}^{n*}$, and codomain $\{0,1\}^l$, a MAC is a map $\mathcal{F} : \mathcal{K} \times \{0,1\}^{n*} \to \{0,1\}^l$. A MAC $\mathcal{F}$ can be regarded as a family of functions from $\{0,1\}^{n*}$ to $\{0,1\}^l$ indexed by a key $K \in \mathcal{K}$. In fact, $\mathcal{F}$ is a multiset since two or more different keys may define the same function.

Let $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher; then $E_K(X) = Y$ denotes that E uses a key $K \in \mathcal{K}$ to encrypt an n-bit string X to an n-bit ciphertext Y.

2.2 The 3GPP-MAC algorithm

The 3GPP-MAC algorithm operates as follows. Suppose the underlying block cipher E has n-bit input and output blocks. Every message M in 3GPP-MAC is first padded such that the length is a multiple of n. The padding string in 3GPP-MAC is appended even if the size of the message is already a multiple of n; it is of the following form: $Count \,||\, Fresh \,||\, Message \,||\, Direction \,||\, 1 \,||\, 00 \cdots 0$, where $Count$, $Fresh$, and $Direction$ are system dependent parameters. Throughout this paper we assume that the lengths of all messages are multiples of n since the details of the padding scheme are not relevant for our proof of security.

The 3GPP-MAC algorithm uses a pair of 128-bit keys K and K', where $K' = K \oplus Const$ and $Const = 0xAA \cdots A$. For any r-block message $M = M[1] \cdots M[r]$, 3GPP-MAC is computed as follows:

$$
\begin{aligned}
& O[0] \leftarrow 0 \\
& \text{for } i = 1, \cdots, r \text{ do} \\
& \qquad I[i] \leftarrow O[i-1] \oplus M[i] \\
& \qquad O[i] \leftarrow E_K(I[i]) \\
& \overline{O}_K(M) \leftarrow O[1] \oplus O[2] \oplus \cdots \oplus O[r] \\
& \mathcal{M}_K(M) \leftarrow \text{the leftmost } \; l \text{ bits of } E_{K'}(\overline{O}_K(M)) \\
& \text{return } \mathcal{M}_K(M)
\end{aligned}
$$

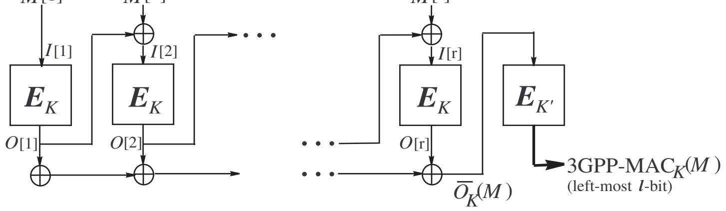

Fig. 1. The 3GPP-MAC algorithm

Here $\mathcal{M}_K(M)$ is the 3GPP-MAC value of the message M. The 3GPP-MAC algorithm is also depicted in Fig. 1. The 3GPP integrity algorithm $f9$ in the 3GPP technical specification [22] states that the underlying block cipher is KASUMI: KASUMI is a 64-bit block cipher with a 128-bit key. The 3GPP-MAC value consists of the leftmost 32 bits of the final encryption or $l = 32$.

Note that in the 3GPP-MAC algorithm, K and K' should be distinct to handle variable length messages. In fact, it is easy to break the 3GPP-MAC algorithm if $K = K'$. For example, if an adversary requests $\mathcal{M}_K(X)$ for a 1-block message X, obtaining T, and requests $\mathcal{M}_K(0)$ of a 1-block message 0, obtaining S, then she can compute the MAC $\mathcal{M}_K(X||0||T \oplus X||0||T) = S$. In other words, from the MACs of X and 0, one can forge the MAC of $X||0||\mathcal{M}_K(X) \oplus X||0||\mathcal{M}_K(X)$ without knowing the key K.

2.3 Comparison between CBC-MAC, EMAC, and 3GPP-MAC

The basic CBC-MAC algorithm [23] works as follows: for any r-block message $M = M[1] \cdots M[r]$, the CBC-MAC of M under the key K is defined as $CBC_{E_K}(M) = C_r$, where $C_i = E_K(M[i] \oplus C_{i-1})$ for $i = 1, \cdots, r$ and $C_0 = 0$. The CBC-MAC is illustrated in Fig. 2.

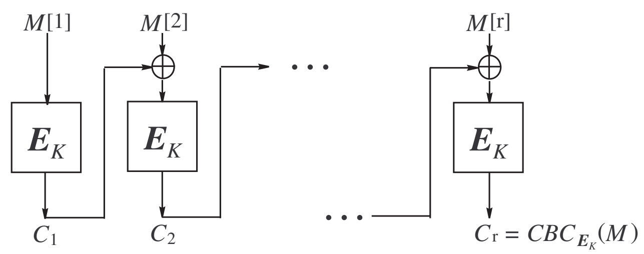

Fig. 2. The CBC-MAC algorithm

It is well known that CBC-MAC is secure for messages of constant length, while it is insecure for arbitrary variable length messages [1]. There have been several efforts to design a variant of CBC-MAC for variable length messages. Bellare *et al.* [1] have suggested three variants of CBC-MAC, *Input-length key separation*, *Length-prepending*, and *Encrypt last block*, to handle variable length messages. Out of these three variants the most attractive method is the last one, since the length of message is not needed until the end of the computation. The method of encrypting the last block is called the EMAC; it has been proposed by the RIPE project in 1993 [2] and subsequently included in the ISO standard [23]; its security has been rigorously analyzed by Petrank and Rackoff [16].

For any r-block message $M = M[1] \cdots M[r]$, EMAC of M is defined as $EMAC_{E_{K_1}, E_{K_2}}(M) = E_{K_2}(CBC_{E_{K_1}}(M))$, where K_1 and K_2 are two different keys in $\mathcal{K}$. The EMAC algorithm is depicted in Fig. 3. In fact, Petrank and Rackoff [16] used one secret key K to produce two secret keys $K_1 = E_K(0)$ and $K_2 = E_K(1)$, and they regarded E_{K_1} and E_{K_2} as two independently chosen random functions f_1 and f_2 for the proof of security.

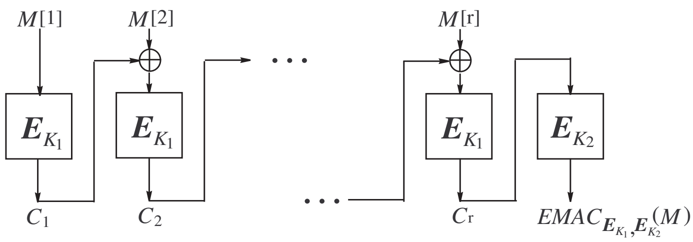

Fig. 3. The EMAC algorithm

In order to optimize efficiency for constructions that accept arbitrary bit strings, Black and Rogaway [3] refined EMAC in three methods which they called ECBC, FCBC, and XCBC, respectively.

On the other hand, 3GPP-MAC can be seen as a variant of EMAC. There are two differences between 3GPP-MAC and EMAC. First, 3GPP-MAC uses a pair of keys K and K' such that K' is straightforwardly derived from K by XORing the fixed constant, but in EMAC, two keys K_1 and K_2 are obtained by encrypting two plaintexts 0 and 1 with the same key K. Thus we cannot regard E_K and $E_{K'}$ as two independently chosen random functions f and f' for the proof of security. This situation of 3GPP-MAC is different from that of EMAC. Second, while 3GPP-MAC uses $CBC_{E_K}(M) \oplus (C_1 \oplus C_2 \oplus \cdots \oplus C_{r-1})$ as the input of the final block computation $E_{K'}$, EMAC uses $CBC_{E_{K_1}}(M)$ without XORing C_i's as the input of the final computation E_{K_2}. These two distinct points give rise to a different security proof for EMAC and 3GPP-MAC.

2.4 Security model

We consider the following security model. Let $\mathcal{A}$ be an adversary and $\mathcal{A}^{\mathcal{O}}$ denote that $\mathcal{A}$ can access an oracle $\mathcal{O}$. Without loss of generality, adversaries are assumed to never ask a query outside the domain of the oracle, and to never repeat a query. For any $g \in \mathcal{F}$, we say that $\mathcal{A}$ forges g if $\mathcal{A}$ outputs $g(x)$ for some $x \in \{0,1\}^{n*}$ where $\mathcal{A}^g$ never queried x to its oracle g. Define

$$Adv_{\mathcal{F}}^{mac}(\mathcal{A}) = \Pr\left(\mathcal{A} \text{ forges } g \mid g \stackrel{R}{\leftarrow} \mathcal{F}\right) ,$$

where $g \stackrel{R}{\leftarrow} \mathcal{F}$ denotes the experiment of choosing a random element from $\mathcal{F}$.

Assume that for any random function $\lambda \in \mathcal{R}_{n*\to l}$, the value of $\lambda(x)$ is a uniformly chosen l-bit string from $\{0,1\}^l$, for each $x \in \{0,1\}^{n*}$. That is, for any $\lambda \in \mathcal{R}_{n*\to l}$, $x \in \{0,1\}^{n*}$, and $y \in \{0,1\}^l$, $\Pr(\lambda(x) = y) = 2^{-l}$. This is a reasonable assumption since for any uniformly chosen function $g : \{0,1\}^m \to \{0,1\}^l$, $\Pr(g(x) = y) = 2^{-l}$ for each $x \in \{0,1\}^m$ and $y \in \{0,1\}^l$, regardless of the input length m. We define the advantage of an adversary $\mathcal{A}$ to distinguish a MAC $\mathcal{F}$ from the family of random functions $\mathcal{R}_{n*\to l}$ as

$$Adv_{\mathcal{F}}^{\mathcal{R}_{n*\to l}}(\mathcal{A}) = \Pr\left(\mathcal{A}^g \text{ outputs } 1 \mid g \xleftarrow{R} \mathcal{F}\right) - \Pr\left(\mathcal{A}^\lambda \text{ outputs } 1 \mid \lambda \xleftarrow{R} \mathcal{R}_{n*\to l}\right) .$$

We overload the notation defined above and write that

$$Adv_{\mathcal{F}}^{mac}(t, q, \sigma) = \max_{\mathcal{A}}\{Adv_{\mathcal{F}}^{mac}(\mathcal{A})\}$$

and

$$Adv_{\mathcal{F}}^{\mathcal{R}_{n*\to l}}(t, q, \sigma) = \max_{\mathcal{A}}\{Adv_{\mathcal{F}}^{\mathcal{R}_{n*\to l}}(\mathcal{A})\} ,$$

where the maximum is over all adversaries $\mathcal{A}$ who run in time at most t and ask its oracle q queries having aggregate length of σ blocks.

On the other hand, we regard the block cipher Λ_n as a family of permutations from $\{0,1\}^n$ to itself indexed by a secret key $K \in \mathcal{K}$. Define

$$Adv_{\Lambda_n}^{\mathcal{P}_n}(\mathcal{A}) = \Pr\left(\mathcal{A}^f \text{ outputs } 1 \mid f \xleftarrow{R} \Lambda_n\right) - \Pr\left(\mathcal{A}^\pi \text{ outputs } 1 \mid \pi \xleftarrow{R} \mathcal{P}_n\right)$$

and

$$Adv_{\Lambda_n}^{\mathcal{P}_n}(t, q) = \max_{\mathcal{A}}\{Adv_{\Lambda_n}^{\mathcal{P}_n}(\mathcal{A})\} ,$$

where the maximum is over all distinguishers $\mathcal{A}$ that run in time at most t and make at most q queries.

In what follows it will be convenient for us to think of 3GPP-MAC as using two functions f and f' instead of E_K and $E_{K'}$, respectively. We do this by denoting f to be E_K for a randomly chosen key K and f' to be $E_{K'}$ for a second key K'. Note that f' is derived from f. Now, we may write

$$\mathcal{M}_f(M) = \text{the leftmost } \ l \text{ bits of } f'(\bar{O}_f(M)) ,$$

where $\bar{O}_f(M) = O[1] \oplus O[2] \oplus \cdots \oplus O[r]$, $O[i] = f(I[i])$, $I[i] = O[i-1] \oplus M[i]$ for $1 \le i \le r$, and $O[0] = 0$.

We consider two function families related to 3GPP-MAC. A family $\mathcal{M}_{\Lambda_n}$ for a block cipher Λ_n is the set of all functions $\mathcal{M}_f$ for all $f \in \Lambda_n$ and a family $\mathcal{M}_{\mathcal{P}_n}$ is the set of all functions $\mathcal{M}_\pi$ for all $\pi \in \mathcal{P}_n$. The $\mathcal{M}_\pi$ is similarly defined as $\mathcal{M}_f$ by considering π and π' instead of f and f', that is, for any message M,

$$\mathcal{M}_\pi(M) = \text{the leftmost } \ l \text{ bits of } \pi'(\bar{O}_\pi(M)) ,$$

where $\pi' \in \mathcal{P}_n - \{\pi\}$ is automatically determind by π. Note that our result in the next section have nothing to do with the method of determining π' from π.

3 The security of 3GPP-MAC

3.1 Main results

In this section we prove that the security of $\mathcal{M}_{\Lambda_n}$ is implied by the security of the underlying block cipher Λ_n. We call a block cipher secure if it is a pseudorandom permutation: this means that no attacker with polynomially many encryption queries can distinguish the block cipher from a perfect random permutation. This approach to modeling the security of a block cipher was introduced by Luby and Rackoff [15].

We first give the following information-theoretic bound on the security of 3GPP-MAC. We start by checking the possibility of distinguishing a random function in $\mathcal{R}_{n*\to l}$ from a random function in $\mathcal{M}_{\mathcal{P}_n}$. We show that even a computationally unbounded adversary cannot obtain a too large advantage.

Theorem 1 *Let $\mathcal{A}$ be an adversary that makes queries to a random function chosen either from $\mathcal{M}_{\mathcal{P}_n}$ or from $\mathcal{R}_{n*\to l}$. Suppose that $\mathcal{A}$ asks its oracle q queries, these queries having aggregate length of σ blocks. Then*

$$Adv_{\mathcal{M}_{\mathcal{P}_n}}^{\mathcal{R}_{n*\to l}}(\mathcal{A}) \leq \frac{(\sigma^2 + 2q^2)}{2^n} .$$

The proof of Theorem 1 will be given in Sect. 3.2. It is a well-known result that if a MAC algorithm preserves pseudorandomness, it resists an existential forgery under adaptive chosen message attacks [1]. By using this fact and Theorem 1, we can obtain the main result:

Theorem 2 *Let $\Lambda_n : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be a family of permutations obtained from a block cipher. Then*

$$Adv_{\mathcal{M}_{\Lambda_n}}^{\mathcal{R}_{n*\to l}}(t, q, \sigma) \leq Adv_{\Lambda_n}^{\mathcal{P}_n}(t', \sigma) + \frac{(\sigma^2 + 2q^2)}{2^n} \tag{3.1}$$

and

$$Adv_{\mathcal{M}_{\Lambda_n}}^{mac}(t, q, \sigma) \leq Adv_{\Lambda_n}^{\mathcal{P}_n}(t', \sigma) + \frac{(\sigma^2 + 2q^2)}{2^n} + \frac{1}{2^l} , \tag{3.2}$$

where $t' = t + O(\sigma n)$.

Proof. Let $\mathcal{A}$ be an adversary distinguishing $\mathcal{M}_{\Lambda_n}$ from $\mathcal{R}_{n*\to l}$ which makes at most q oracle queries having aggregate length of σ blocks and runs in time at most t. In order to prove equation (3.1), we first show that there exists an adversary $\mathcal{B}_{\mathcal{A}}$ which distinguishes Λ_n from $\mathcal{P}_n$ such that

$$Adv_{\Lambda_n}^{\mathcal{P}_n}(\mathcal{B}_{\mathcal{A}}) = Adv_{\mathcal{M}_{\Lambda_n}}^{\mathcal{R}_{n*\to l}}(\mathcal{A}) - Adv_{\mathcal{M}_{\mathcal{P}_n}}^{\mathcal{R}_{n*\to l}}(\mathcal{A}) ,$$

where $\mathcal{B}_{\mathcal{A}}$ makes at most σ queries and runs in time at most $t' = t + O(\sigma n)$. The adversary $\mathcal{B}_{\mathcal{A}}$ gets an oracle $f : \{0,1\}^n \to \{0,1\}^n$, a permutation chosen from Λ_n or $\mathcal{P}_n$. It will run $\mathcal{A}$ as a subroutine, using f to simulate the oracle $h : \{0,1\}^{n*} \to \{0,1\}^l$ that $\mathcal{A}$ expects.

Adversary $\mathcal{B}_{\mathcal{A}}^f$
  for $i = 1, \cdots, q$ do
    when $\mathcal{A}$ asks its oracle a query M_i, answer with $\mathcal{M}_f(M_i)$
  end for
  $\mathcal{A}$ outputs a bit b
  return b

The oracle supplied to $\mathcal{A}$ by $\mathcal{B}_{\mathcal{A}}$ is $\mathcal{M}_f$, where f is $\mathcal{B}_{\mathcal{A}}$'s oracle, and hence

$$\begin{aligned} Adv_{\Lambda_n}^{\mathcal{P}_n}(\mathcal{B}_{\mathcal{A}}) &= \Pr\left(\mathcal{B}_{\mathcal{A}}^f = 1 \mid f \overset{R}{\leftarrow} \Lambda_n\right) - \Pr\left(\mathcal{B}_{\mathcal{A}}^f = 1 \mid f \overset{R}{\leftarrow} \mathcal{P}_n\right) \\ &= \Pr\left(\mathcal{A}^h = 1 \mid h \overset{R}{\leftarrow} \mathcal{M}_{\Lambda_n}\right) - \Pr\left(\mathcal{A}^h = 1 \mid h \overset{R}{\leftarrow} \mathcal{M}_{\mathcal{P}_n}\right) \ . \end{aligned}$$

However

$$Adv_{\mathcal{M}_{\mathcal{P}_n}}^{\mathcal{R}_{n*\to l}}(\mathcal{A}) = \Pr\left(\mathcal{A}^h = 1 \mid h \overset{R}{\leftarrow} \mathcal{M}_{\mathcal{P}_n}\right) - \Pr\left(\mathcal{A}^h = 1 \mid h \overset{R}{\leftarrow} \mathcal{R}_{n*\to l}\right) \ .$$

Therefore by taking the sum of the two equations above, we obtain that

$$\begin{aligned} & Adv_{\Lambda_n}^{\mathcal{P}_n}(\mathcal{B}_{\mathcal{A}}) + Adv_{\mathcal{M}_{\mathcal{P}_n}}^{\mathcal{R}_{n*\to l}}(\mathcal{A}) \\ &= \Pr\left(\mathcal{A}^h = 1 \mid h \overset{R}{\leftarrow} \mathcal{M}_{\Lambda_n}\right) - \Pr\left(\mathcal{A}^h = 1 \mid h \overset{R}{\leftarrow} \mathcal{R}_{n*\to l}\right) \\ &= Adv_{\mathcal{M}_{\Lambda_n}}^{\mathcal{R}_{n*\to l}}(\mathcal{A}) \ . \end{aligned}$$

From this equation and the result of Theorem 1, we get

$$Adv_{\Lambda_n}^{\mathcal{P}_n}(\mathcal{B}_{\mathcal{A}}) \geq Adv_{\mathcal{M}_{\Lambda_n}}^{\mathcal{R}_{n*\to l}}(\mathcal{A}) - \frac{(\sigma^2 + 2q^2)}{2^n} \ ,$$

and the equation (3.1) follows, since

$$\begin{aligned} Adv_{\mathcal{M}_{\Lambda_n}}^{\mathcal{R}_{n*\to l}}(t, q, \sigma) &= \max_{\mathcal{A}} \left\{ Adv_{\mathcal{M}_{\Lambda_n}}^{\mathcal{R}_{n*\to l}}(\mathcal{A}) \right\} \\ &\leq \max_{\mathcal{A}} \left\{ Adv_{\Lambda_n}^{\mathcal{P}_n}(B_{\mathcal{A}}) + \frac{(\sigma^2 + 2q^2)}{2^n} \right\} \\ &\leq Adv_{\Lambda_n}^{\mathcal{P}_n}(t', \sigma) + \frac{(\sigma^2 + 2q^2)}{2^n} \ . \end{aligned}$$

Using Proposition 2.7 of [1], we can easily show that

$$Adv_{\mathcal{M}_{\Lambda_n}}^{mac}(t, q, \sigma) \leq Adv_{\mathcal{M}_{\Lambda_n}}^{\mathcal{R}_{n*\to l}}(t', q, \sigma) + \frac{1}{2^l} \ , \tag{3.3}$$

where $t' = t + O(\sigma n)$. Combining (3.1) and (3.3) we obtain the equation (3.2) which completes the proof. $\square$

3.2 Proof of Theorem 1

Remember that the second permutation $\bar{\pi}$ in $\mathcal{M}_\pi(\cdot)$ is derived from π. In order to prove Theorem 1 we first prove the result under the condition that the second permutation $\bar{\pi}$ is not related with the first permutation π in 3GPP-MAC. Assume that π and π' are chosen independently from $\mathcal{P}_n$. For any r-block message $M = M[1] \cdots M[r]$, we set

$$\mathcal{M}_{\pi,\pi'}(M) = \text{the leftmost } \ l \text{ bits of } \pi'\left(\bar{O}_\pi(M)\right),$$

where $\bar{O}_\pi(M) = O[1] \oplus \cdots \oplus O[r]$, $O[i] = \pi(I[i])$, and $I[i] = O[i-1] \oplus M[i]$ for $1 \le i \le r$. Let $\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}$ be the set of all functions $\mathcal{M}_{\pi,\pi'}$, where π and π' are chosen independently from $\mathcal{P}_n$.

Lemma 1 below provides an information-theoretic bound on the security of $\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}$.

Lemma 1 *Let $\mathcal{A}$ be an adversary that makes queries to a random function chosen either from $\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}$ or from $\mathcal{R}_{n*\to l}$. Suppose that $\mathcal{A}$ asks its oracle q queries, these queries having aggregate length of σ blocks. Then*

$$Adv^{\mathcal{R}_{n*\to l}}_{\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}}(\mathcal{A}) \le \frac{(\sigma^2 + 2q^2)}{2^{n+1}}.$$

Proof. To prove Lemma 1 we apply the idea from the proof of PMAC's security in [18]. Let $\mathcal{A}$ be an adversary distinguishing $\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}$ from $\mathcal{R}_{n*\to l}$. Since the adversary $\mathcal{A}$ is not limited in computational power, we may assume it is deterministic. One can imagine $\mathcal{A}$ interacting with a $\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}$ oracle as $\mathcal{A}$ playing the following game, denoted Game 1.

Game 1: Simulation of $\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}$

unusual $\leftarrow$ false; for all $x \in \{0,1\}^n$ do $\pi(x) \leftarrow$ undefined, $\pi'(x) \leftarrow$ undefined
When $\mathcal{A}$ makes its t-th query, $M_t = M_t[1] \cdots M_t[r_t]$ where $t \in \{1, \cdots, q\}$
  $I_t[1] \leftarrow M_t[1]$
  For $i = 1, \cdots, r_t$ do
    $A \leftarrow \{I_t[j] \mid 1 \le j \le i-1\} \cup \{I_s[j] \mid 1 \le s \le t-1,\ 1 \le j \le r_s\}$
    if $I_t[i] \in A$ then $O_t[i] \leftarrow \pi(I_t[i])$
    else $O_t[i] \xleftarrow{R} \{0,1\}^n$
      $A_\pi \leftarrow \{\pi(I_t[j]) \mid 1 \le j \le i-1\} \cup \{\pi(I_s[j]) \mid 1 \le s \le t-1,\ 1 \le j \le r_s\}$
      if $O_t[i] \in A_\pi$ then [*unusual* $\leftarrow$ true; $O_t[i] \xleftarrow{R} A_\pi^C$]
      $\pi(I_t[i]) \leftarrow O_t[i]$
    if $i < r_t$ then $I_t[i+1] \leftarrow O_t[i] \oplus M_t[i+1]$
  $\bar{O}_t(M_t) \leftarrow O_t[1] \oplus \cdots \oplus O_t[r_t]$
  $B \leftarrow \{\bar{O}_s(M_s) \mid 1 \le s \le t-1\}$
  If $\bar{O}_t(M_t) \in B$ then [*unusual* $\leftarrow$ true; $MAC_t \leftarrow \pi'(\bar{O}_t)$]
  else $MAC_t \xleftarrow{R} \{0,1\}^n$
    $B_{\pi'} \leftarrow \{\pi'(\bar{O}_s(M_s)) \mid 1 \le s \le t-1\}$
    if $MAC_t \in B_{\pi'}$ then [*unusual* $\leftarrow$ true; $MAC_t \xleftarrow{R} B_{\pi'}^C$]

18 $\quad\quad \pi'(\bar{O}_t(M_t)) \leftarrow MAC_t$
19 $\quad \mathcal{M}_{\pi,\pi'}(M_t) \leftarrow$ the leftmost l-bit of MAC_t
20 $\quad$ Return $\mathcal{M}_{\pi,\pi'}(M_t)$

Here we use A_π^C and $B_{\pi'}^C$ to denote the complements of A_π and $B_{\pi'}$, respectively. Two particular permutations π and π' are equally likely among all permutations from $\{0,1\}^n$ to $\{0,1\}^n$. In our simulation, we will view the selection of π and π' as an incremental procedure. This will be equivalent to selecting π and π' uniformly at random. This game perfectly simulates the behavior of $\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}$.

Let $UNUSUAL$ be the event that the flag *unusual* is set to true in Game 1. In the absence of event $UNUSUAL$, the returned value $\mathcal{M}_{\pi,\pi'}(M_t)$ at line 20 is random since the leftmost l bits of the string randomly selected at line 15. That is, the adversary sees the returned random values on distinct points. Therefore we get that

$$Adv_{\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}}^{\mathcal{R}_{n*\to l}}(\mathcal{A}) \le \Pr(UNUSUAL)\,. \tag{3.4}$$

First we consider the probability that the flag *unusual* is set to true in line 9 or 17. In both cases, we have just chosen a random n-bit string and then we check whether it is a element in a set or not. We have that

$$\begin{aligned} &\Pr(unusual = \text{true in lines 9 or 17 in Game 1}) \\ &\quad \le \frac{1+2+\cdots+(\sigma-1)+1+\cdots+(q-1)}{2^n} \\ &\quad \le \frac{\sigma^2+q^2}{2^{n+1}}\,. \end{aligned} \tag{3.5}$$

Now we can modify Game 1 by changing the behavior when $unusual = \text{true}$, and adding as a compensating factor the bound given by equation (3.5). We omit lines 8, 9, 16 and 17, and the last statement in line 14. The modified game is as follows.

Game 2: Simplification of Game 1
1 $\;$ *unusual* $\leftarrow$ false; for all $x \in \{0,1\}^n$ do $\pi(x) \leftarrow$ undefined, $\pi'(x) \leftarrow$ undefined
2 $\;$ When $\mathcal{A}$ makes its t-th query, $M_t = M_t[1]\cdots M_t[r_t]$ where $t \in \{1,\cdots,q\}$
3 $\quad I_t[1] \leftarrow M_t[1]$
4 $\quad$ For $i = 1,\cdots,r_t$ do
5 $\quad\quad A \leftarrow \{I_t[j] \mid 1 \le j \le i-1\} \cup \{I_s[j] \mid 1 \le s \le t-1,\ 1 \le j \le r_s\}$
6 $\quad\quad$ if $I_t[i] \in A$ then $O_t[i] \leftarrow \pi(I_t[i])$
7 $\quad\quad$ else $[O_t[i] \xleftarrow{R} \{0,1\}^n;\ \pi(I_t[i]) \leftarrow O_t[i]]$
8 $\quad\quad$ if $i < r_t$ then $I_t[i+1] \leftarrow O_t[i] \oplus M_t[i+1]$
9 $\quad \bar{O}_t(M_t) \leftarrow O_t[1] \oplus \cdots \oplus O_t[r_t]$
10 $\quad B \leftarrow \{\bar{O}_s(M_s) \mid 1 \le s \le t-1\}$
11 $\quad$ If $\bar{O}_t(M_t) \in B$ then *unusual* $\leftarrow$ true
12 $\quad MAC_t \xleftarrow{R} \{0,1\}^n$
13 $\quad \pi'(\bar{O}_t(M_t)) \leftarrow MAC_t$
14 $\quad \mathcal{M}_{\pi,\pi'}(M_t) \leftarrow$ the leftmost l-bit of MAC_t

Return $\mathcal{M}_{\pi,\pi'}(M_t)$

By the equation (3.5) we have that

$$\Pr(\mathit{UNUSUAL}) \le \Pr(\mathit{unusual} = \text{true in Game 2}) + \frac{\sigma^2 + q^2}{2^{n+1}} \,. \tag{3.6}$$

In Game 2 the value $\mathcal{M}_{\pi,\pi'}(M_t)$ returned in response to a query M_t is a random l-bit string. Thus we can first select these MAC_t values in Game 2. This does not change the view of the adversary that interacts with the game and the probability that *unusual* is set to true. This modified game is called Game 3, and it is depicted as follows.

Game 3: Modification of Game 2

1 $\mathit{unusual} \leftarrow$ false; for all $x \in \{0,1\}^n$ do $\pi(x) \leftarrow$ undefined, $\pi'(x) \leftarrow$ undefined
2 When $\mathcal{A}$ makes its t-th query, $M_t = M_t[1] \cdots M_t[r_t]$ where $t \in \{1, \cdots, q\}$
3    $MAC_t \stackrel{R}{\leftarrow} \{0,1\}^n$
4    $\mathcal{M}_{\pi,\pi'}(M_t) \leftarrow$ the leftmost l-bit of MAC_t
5   Return $\mathcal{M}_{\pi,\pi'}(M_t)$
6 When $\mathcal{A}$ is done making its q queries
7   For $t = 1, \cdots, q$ do
8       $I_t[1] \leftarrow M_t[1]$
9   For $i = 1, \cdots, r_t$ do
10          $A \leftarrow \{I_t[j] \mid 1 \le j \le i-1\} \cup \{I_s[j] \mid 1 \le s \le t-1,\ 1 \le j \le r_s\}$
11   if $I_t[i] \in A$ then $O_t[i] \leftarrow \pi(I_t[i])$
12   else $[O_t[i] \stackrel{R}{\leftarrow} \{0,1\}^n;\ \pi(I_t[i]) \leftarrow O_t[i]]$
13   if $i < r_t$ then $I_t[i+1] \leftarrow O_t[i] \oplus M_t[i+1]$
14       $\bar{O}_t(M_t) \leftarrow O_t[1] \oplus \cdots \oplus O_t[r_t]$
15       $B \leftarrow \{\bar{O}_s(M_s) \mid 1 \le s \le t-1\}$
16   If $\bar{O}_t(M_t) \in B$ then $\mathit{unusual} \leftarrow$ true
17       $\pi'(\bar{O}_t(M_t)) \leftarrow MAC_t$

We note that

$$\Pr(\mathit{unusual} = \text{true in Game 3}) = \Pr(\mathit{unusual} = \text{true in Game 2}) \,. \tag{3.7}$$

Now we want to show that the probability of $\mathit{unusual} =$ true in Game 3, over the random MAC_t values selected at line 3 and the random $O_t[i]$ values selected at line 12, is small. In fact, we will show something stronger: even if one arbitrarily fixes the values of $MAC_1, \cdots, MAC_q \in \{0,1\}^n$, the probability that *unusual* will be set to true is still small. Since the oracle answers have now been fixed and the adversary is deterministic, the queries $M_1, \cdots, M_q$ that the adversary will make have likewise been fixed. The new game is called Game 4(C). It depends on constants $C = (q, MAC_1, \cdots, MAC_q, M_1, \cdots, M_q)$.

Game 4(C)

$\mathit{unusual} \leftarrow$ false; for all $x \in \{0,1\}^n$ do $\pi(x) \leftarrow$ undefined, $\pi'(x) \leftarrow$ undefined
For $t = 1, \cdots, q$ do

3 $\quad I_t[1] \leftarrow M_t[1]$
4 $\quad$ For $i = 1, \cdots, r_t$ do
5 $\qquad A \leftarrow \{I_t[j] \mid 1 \le j \le i-1\} \cup \{I_s[j] \mid 1 \le s \le t-1,\ 1 \le j \le r_s\}$
6 $\qquad$ if $I_t[i] \ \in\ A$ then $O_t[i] \leftarrow \pi(I_t[i])$
7 $\qquad$ else $[O_t[i] \stackrel{R}{\leftarrow} \{0,1\}^n;\ \pi(I_t[i]) \leftarrow O_t[i]]$
8 $\qquad$ if $i < r_t$ then $I_t[i+1] \leftarrow O_t[i] \oplus M_t[i+1]$
9 $\quad \bar{O}_t(M_t) \leftarrow O_t[1] \oplus \cdots \oplus O_t[r_t]$
10 $\quad B \leftarrow \{\bar{O}_s(M_s) \mid 1 \le s \le t-1\}$
11 $\quad$ If $\bar{O}_t(M_t) \ \in\ B$ then *unusual* $\leftarrow$ true
12 $\quad \pi'(\bar{O}_t(M_t)) \leftarrow MAC_t$

We know that

$$\begin{aligned} &\Pr(\mathit{unusual} = \text{true in Game 3})] \\ &\quad \le \max_C\{\Pr(\mathit{unusual} = \text{true in Game } 4(C))\}\,. \end{aligned} \tag{3.8}$$

Thus, by (3.4) and (3.6)-(3.8) we have that

$$Adv^{\mathcal{R}_{n*\to l}}_{\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}}(\mathcal{A}) \le \max_C\{\Pr(\mathit{unusual} = \text{true in Game 4(C)})\} + \frac{\sigma^2 + q^2}{2^{n+1}}\ , \tag{3.9}$$

where, if $\mathcal{A}$ is limited to q queries of aggregate length σ, then C specifies q, message strings $M_1, \cdots, M_q$ of aggregate block length σ, and $MAC_1, \cdots, MAC_q \ \in \{0,1\}^n$.

Finally, we modify Game $4(C)$ by changing the order of choosing a random $O_t[i]$ in line 7. This game is called Game $5(C)$.

Game $5(C)$
1 $\ $ *unusual* $\leftarrow$ false; for all $x \in \{0,1\}^n$ do $\ \pi(x) \leftarrow$ undefined, $\ \pi'(x) \leftarrow$ undefined
2 $\ $ For $t = 1, \cdots, q$ do
3 $\quad$ For $i = 1, \cdots, r_t$ do
4 $\qquad I_t[1] \leftarrow M_t[1]\ ;\ O_t[i] \stackrel{R}{\leftarrow} \{0,1\}^n$
5 $\qquad A \leftarrow \{I_t[j] \mid 1 \le j \le i-1\} \cup \{I_s[j] \mid 1 \le s \le t-1,\ 1 \le j \le r_s\}$
6 $\qquad$ if $I_t[i] \ \in\ A$ then $O_t[i] \leftarrow \pi(I_t[i])$
7 $\qquad$ else $\pi(I_t[i]) \leftarrow O_t[i]$
8 $\qquad$ if $i < r_t$ then $I_t[i+1] \leftarrow O_t[i] \oplus M_t[i+1]$
9 $\quad \bar{O}_t(M_t) \leftarrow O_t[1] \oplus \cdots \oplus O_t[r_t]$
10 $\quad B \leftarrow \{\bar{O}_s(M_s) \mid 1 \le s \le t-1\}$
11 $\quad$ If $\bar{O}_t(M_t) \ \in\ B$ then *unusual* $\leftarrow$ true
12 $\quad \pi'(\bar{O}_t(M_t)) \leftarrow 0^n$

Notice that in Game 5, we choose a random $O_t[i]$ value in line 4. To avoid that the game depends on the MAC_t-values, we also set $\pi'(\bar{O}_t(M_t))$ to some particular value, 0^n, instead of to MAC_t in the last line. The particular value associated to this point is not used unless *unusual* has already been set to true. Thus we obtain that

$$\Pr(unusual = \text{true in Game 4(C)})$$
$$= \Pr(unusual = \text{true in Game 5(C)})\,. \qquad (3.10)$$

The coins used in Game 5 are $O_1(M_1) = O_1[1]\cdots O_1[r_1], \cdots, O_q(M_q) = O_q[1]\cdots O_q[r_q]$, where either $O_s[i]$'s are random coins or are a synonym $O_u[j]$. Here we set $O_t[0] = 0$ and $I_t[k] \leftarrow O_t[k-1] \oplus M_t[k]$ for $1 \le t \le q$ and $1 \le k \le r_t$, and if there exists the smallest number $u < s$ such that $I_s[i] = I_u[j]$ then $O_s[i] = O_u[j]$, else if there exists the smallest number $j < i$ such that $I_s[i] = I_s[j]$ then $O_s[i] = O_s[j]$, else $O_s[i]$ is a random coin.

Run Game 5 on $M_1, \cdots, M_q$ and the indicated vector of coins. Suppose that *unusual* gets set to true on this execution. Let $s \in \{1, \cdots, q\}$ be the particular value of t when *unusual* first get set to true. Then

$$\bar{O}_s(M_s) = \bar{O}_u(M_u) \text{ for some } u \in \{1, \cdots, s-1\}\,.$$

In this case, if we had run Game 5 using coins O_u and O_s and restricting the execution of line 2 to $t \in \{u, s\}$, then *unusual* still would have been set to true. In this restricted Game 5, we get

$$\Pr\left(\bar{O}_s(M_s) = \bar{O}_u(M_u)\right) = \Pr\left(O_s[1] \oplus \cdots \oplus O_s[r_s] = O_u[1] \oplus \cdots \oplus O_u[r_u]\right)$$
$$= 2^{-n}$$

because $O_u[1]$ in $\bar{O}_u(M_u)$ is a random string in $\{0,1\}^n$. Thus we obtain that

$$\max_C \{\Pr(unusual \leftarrow \text{true in Game } 5(C))\}$$
$$\le \max_{\substack{r_1,\cdots,r_q \\ \sigma=\sum r_i}} \left\{ \sum_{1\le u<s\le q} 2^{-n} \right\}$$
$$\le \left(\frac{q(q-1)}{2}\right) \cdot \frac{1}{2^n}$$
$$\le \frac{q^2}{2^{n+1}}\,. \qquad (3.11)$$

Combining (3.9)-(3.11), we get that

$$Adv^{\mathcal{R}_{n*\to l}}_{\mathcal{M}_{\mathcal{P}_n\times\mathcal{P}_n}}(\mathcal{A}) \le \frac{\sigma^2 + q^2}{2^{n+1}} + \frac{q^2}{2^{n+1}}\,.$$

This completes the proof of Lemma 1. □

Now we check the possibility of distinguishing a random function in the original $\mathcal{M}_{\mathcal{P}_n}$ from a random function in $\mathcal{M}_{\mathcal{P}_n\times\mathcal{P}_n}$. To obtain the result, we first need to define what inner collisions are in $\mathcal{M}_{\mathcal{P}_n}$ and $\mathcal{M}_{\mathcal{P}_n\times\mathcal{P}_n}$.

Definition 1 *Let $M_1, \cdots, M_q$ be q strings in $\{0,1\}^{n*}$, and let π, π' be two random permutations in $\mathcal{P}_n$. We say that there occurs an inner collision of $\mathcal{M}_\pi$ on the queries $M_1, \cdots, M_q$ if the collision occurs before invoking the second permutation $\bar{\pi}$ which is derived from π. Namely, if there exists a pair of indices $1 \le i < j \le q$ for which $\bar{O}_\pi(M_i) = \bar{O}_\pi(M_j)$. Similarly, we say that there exists an inner collision of $\mathcal{M}_{\pi,\pi'}$ on the queries $M_1, \cdots, M_q$ if the collision occurs before invoking the second permutation π'.*

Lemma 2 *Let $\mathcal{A}$ be an adversary that makes queries to a random function chosen either from $\mathcal{M}_{\mathcal{P}_n}$ or from $\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}$. Suppose that $\mathcal{A}$ asks its oracle q queries, these queries having aggregate length of σ blocks. Then*

$$Adv_{\mathcal{M}_{\mathcal{P}_n}}^{\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}}(\mathcal{A}) \le \frac{\sigma^2 + 2q^2}{2^{n+1}} .$$

Proof. Let $ICol(\mathcal{M}_{\mathcal{P}_n})$ be the event that there is an inner collision among the messages in $\mathcal{M}_{\mathcal{P}_n}$ and let $ICol(\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n})$ be the event that there is an inner collision among the messages in $\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}$. Observe that since both algorithms are the same before invoking the second permutation, the inner collision probabilities in both algorithms are the same. Thus the following equation holds:

$$\Pr\left(ICol(\mathcal{M}_{\mathcal{P}_n}) \mid \pi \xleftarrow{R} \mathcal{P}_n\right) = \Pr\left(ICol(\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}) \mid \pi, \pi' \xleftarrow{R} \mathcal{P}_n\right) . \tag{3.12}$$

For the same reason, if no inner collisions occur, the adversary outputs 1 with the same probability for $\mathcal{M}_{\mathcal{P}_n}$ and $\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}$ because she sees outputs of permutations on distinct points and the second permutations of $\mathcal{M}_{\mathcal{P}_n}$ and $\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}$ are independent. Let $\Pr_1(\cdot)$ denote the probability that $\mathcal{A}^{\mathcal{M}_\pi}$ outputs 1 under the experiment $\pi \xleftarrow{R} \mathcal{P}_n$ and $\Pr_2(\cdot)$ denote the probability that $\mathcal{A}^{\mathcal{M}_{\pi,\pi'}}$ outputs 1 under the experiment $\pi, \pi' \xleftarrow{R} \mathcal{P}_n$. Then the following holds:

$$\Pr_1\left(\mathcal{A}^{\mathcal{M}_\pi} = 1 \mid \overline{ICol(\mathcal{M}_{\mathcal{P}_n})}\right) = \Pr_2\left(\mathcal{A}^{\mathcal{M}_{\pi,\pi'}} = 1 \mid \overline{ICol(\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n})}\right) , \tag{3.13}$$

where $\overline{ICol(\mathcal{M}_{\mathcal{P}_n})}$ and $\overline{ICol(\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n})}$ are complements of $ICol(\mathcal{M}_{\mathcal{P}_n})$ and $ICol(\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n})$, respectively. Therefore, by using the equation (3.12) and (3.13), we can write the adversary's advantage as follows:

$$\begin{aligned}
&Adv_{\mathcal{M}_{\mathcal{P}_n}}^{\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}}(\mathcal{A}) \\
&\le \left|\Pr_1\left(\mathcal{A}^{\mathcal{M}_\pi} = 1\right) - \Pr_2(\mathcal{A}^{\mathcal{M}_{\pi,\pi'}} = 1)\right| \\
&= \Big|\Pr_1\left(\mathcal{A}^{\mathcal{M}_\pi} = 1 | ICol(\mathcal{M}_{\mathcal{P}_n})\right) \cdot \Pr_1\left(ICol(\mathcal{M}_{\mathcal{P}_n})\right) \\
&\quad + \Pr_1\left(\mathcal{A}^{\mathcal{M}_\pi} = 1 | \overline{ICol(\mathcal{M}_{\mathcal{P}_n})}\right) \cdot \Pr_1\left(\overline{ICol(\mathcal{M}_{\mathcal{P}_n})}\right) \\
&\quad - \Pr_2\left(\mathcal{A}^{\mathcal{M}_{\pi,\pi'}} = 1 | ICol(\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n})\right) \cdot \Pr_2\left(ICol(\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n})\right) \\
&\quad - \Pr_2\left(\mathcal{A}^{\mathcal{M}_{\pi,\pi'}} = 1 | \overline{ICol(\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n})}\right) \cdot \Pr_2\left(\overline{ICol(\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n})}\right)\Big| \\
&= \big|\Pr_2\left(ICol(\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n})\right) \\
&\quad \cdot \left\{\Pr_1\left(\mathcal{A}^{\mathcal{M}_\pi} = 1 | ICol(\mathcal{M}_{\mathcal{P}_n})\right) - \Pr_2\left(\mathcal{A}^{\mathcal{M}_{\pi,\pi'}} = 1 | ICol(\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n})\right)\right\}\big| \\
&\le \Pr_2\left(ICol(\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n})\right) .
\end{aligned}$$

To bound this quantity, we consider again the proof of Lemma 1. In the proof of Lemma 1, Game 1 perfectly simulates the behavior of $\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}$. Observe that when an inner collision occurs in $\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n}$, the flag *unusual* is set to true in Game 1. Thus by Lemma 1, we obtain that

$$\begin{aligned}\Pr_2\left(ICol(\mathcal{M}_{\mathcal{P}_n \times \mathcal{P}_n})\right) &\le \Pr\left(\mathit{unusual} = \text{true in Game 1}\right)\\ &\le \frac{\sigma^2 + 2q^2}{2^{n+1}}\ ,\end{aligned}$$

which completes the proof of Lemma 2. □

Proof of Theorem 1: From Lemma 1 and 2, the proof of Theorem 1 follows immediately. □

4 Conclusion

In this work we have examined the provable security of the 3GPP-MAC algorithm $f9$. We have provided a proof of security for 3GPP-MAC in the sense of reduction-based cryptography. More specifically, we have shown that if there is an existential forgery attack on 3GPP-MAC, then the underlying block cipher can be attacked with comparable parameters. It might be seen as highly unlikely that a realistic attack exists on the 3GPP block cipher KASUMI. If that is indeed the case, our results establish the soundness of the 3GPP-MAC algorithm.

References

1. M. Bellare, J. Kilian, and P. Rogaway, *The security of cipher block chaining*, Crypto'94, LNCS 839, Springer-Verlag, 1994, pp. 341-358. An updated version can be found in the personal URLs of the authors. See, for example, http://www-cse.ucsd.edu/users/mihir/.
2. A. Berendschot *et al.*, *Integrity Primitives for Secure Information Systems. Final Report of RACE Integrity Primitives Evaluation (RIPE-RACE 1040),* LNCS 1007, Springer-Verlag, 1995
3. J. Black and P. Rogaway, *CBC-MACs for arbitrary-length messages: the three-key constructions*, Crypto 2000, LNCS 1880, Springer-Verlag, 2000, pp. 197-215.
4. J. Black and P. Rogaway, *A Block-Cipher Mode of Operation for Parallelizable Message Authentication*, EUROCRYPT 2002, LNCS 2332, Springer-Verlag, 2002, pp. 384-397.
5. L. Carter and M. Wegman, *Universal hash functions*, J. of Computer and System Sciences, 18, 1979, pp. 143-154
6. V. Gligor and P. Donescu, *Fast encryption and authentication: XCBC encryption and XECB authentication modes*, Contribution to NIST, April 20, 2001. Available at http://csrc.nist.gov/encryption/modes/.
7. É. Jaulmes, A. Joux, and F. Valette, *On the security of Randomized CBC-MAC Beyond the Birthday Paradox Limit: A New Construction*, FSE 2002, LNCS 2365, Springer-Verlag, 2002, pp. 237-251.

8. K. Kurosawa and T. Iwata, *TMAC: Two-Key CBC-MAC*, Contribution to NIST, June 21, 2002. Available at http://csrc.nist.gov/encryption/modes/.
9. J. Kang, S. Shin, D. Hong and O. Yi, *Provable security of KASUMI and 3GPP encryption mode* $f8$, ASIACRYPT 2001, LNCS 2248, Springer-Verlag, 2001, pp. 255-271.
10. J. Kang, O. Yi, D. Hong, and H. Cho, *Pseudorandomness of MISTY-type transformations and the block cipher KASUMI*, ACISP 2001, LNCS 2119, Springer-Verlag, 2001, pp. 60-73.
11. L. R. Knudsen and C. J. Mitchell, *Analysis of 3gpp-MAC and two-key 3gpp-MAC*, Discrete Applied Mathematics, to appear.
12. L. Knudsen, *Analysis of RMAC*, Contribution to NIST, November 10, 2002. Available at http://csrc.nist.gov/CryptoToolkit/modes/comments/.
13. T. Kohno, *Related-Key and Key-Collision Attacks Against RMAC*, Contribution to NIST, 2002. Available at http://csrc.nist.gov/CryptoToolkit/modes/comments/.
14. J. Lloyd, *An Analysis of RMAC*, Contribution to NIST, November 18, 2002. Available at http://csrc.nist.gov/CryptoToolkit/modes/comments/.
15. M. Luby and C. Rackoff, *How to construct pseudorandom permutations and pseudorandom functions*, SIAM J. Comput., 17, 1988, pp. 189-203.
16. E. Petrank, C. Rackoff, *CBC-MAC for Real-Time Data Source*, J. of Cryptology, 13, 2000, pp. 315-338.
17. B. Preneel, P.C. van Oorschot, *MDx-MAC and building fast MACs from hash functions,* Crypto'95, LNCS 963, Springer-Verlag, 1995, pp. 1–14.
18. P. Rogaway, *PMAC: A parallelizable message authentication code*, Contribution to NIST, April 17, 2001. Available at http://csrc.nist.gov/encryption/modes/.
19. P. Rogaway, *Comments on NIST's RMAC Proposal*, Contribution to NIST, December 2, 2002. Available at http://csrc.nist.gov/CryptoToolkit/modes/comments/.
20. M. Wegman and L. Carter, *New hash functions and their use in authentication and set equality*, J. of Computer and System Sciences, 22, 1981, pp. 265-279.
21. 3GPP TR 33.909, *Report on the evaluation of 3GPP standard confidentiality and integrity algorithms*, V1.0.0, 2002-12.
22. 3GPP TS 35.201 *Specification of the 3GPP confidentiality and integrity algorithm; Document 1: f8 and f9 specifications.*
23. ISO/IEC 9797-1:1999(E) *Information technology - Security techniques - Message Authentication Codes(MACs) - Part 1.*
24. http://csrc.nist.gov/encryption/modes/