3GPP 무결성 알고리즘 $f9$의 증명가능 안전성

홍 도 원† · 신 상 욱† · 류 희 수† · 정 교 일††

요 약

3GPP의 비동기식 IMT-2000 시스템의 보안 구조에는 표준 무결성 알고리즘 $f9$가 있다. $f9$는 비동기식(W-CDMA) IMT-2000의 무선 구간에서 데이터 무결성과 시그널링 데이터의 출처를 인증하기 위한 메시지 인증 코드(MAC)를 계산하는 알고리즘으로 블록 암호 KASUMI에 기반한 CBC-MAC의 변형이다. 이 논문은 $f9$의 증명 가능한 안전성을 제공한다. 기반이 되는 블록 암호가 유사 랜덤 순열이면 어떤 공격자에 대해서도 $f9$가 안전함을 증명한다.

Provable Security of 3GPP Integrity Algorithm $f9$

Dowon Hong† · Sang Uk Shin† · Heuisu Ryu† · Kyo-Il Chung††

ABSTRACT

Within the security architecture of the 3GPP system there is a standardised integrity algorithm $f9$. The integrity algorithm $f9$ computes a MAC to authenticate the data integrity and data origin of signalling data over a radio access link of W-CDMA IMT-2000. $f9$ is a variant of the standard CBC MAC based on the block cipher KASUMI. In this paper we provide the provable security of $f9$. We prove that $f9$ is secure by giving concrete bound on an adversary's inability to forge in terms of her inability to distinguish the underlying block cipher from a pseudorandom permutation.

키워드 : 메시지 인증 코드(MAC), 증명가능 안전성(Provabile Security), $f9$, 3GPP

1. Introduction

There is a standardised integrity algorithm $f9$ within the security architecture of the 3GPP (3rd Generation Partnership Project) system and this algorithm is a variant of the standard CBC (Cipher Block Chaining) MAC(Message Authentication Code) based on the block cipher KASUMI [11]. To protect the data integrity and guarantee the data origin authentication over a radio access link of W-CDMA IMT-2000, the integrity algorithm $f9$ was proposed. We call this authentication scheme as "3GPP MAC". The purpose of this work is to investigate the provable security of 3GPP MAC.

The provable security treatment of MAC based on the block cipher was started by Bellare et al. [1]. They showed that CBC MAC is secure in the sense of reduction-based cryptography. But their proof depends on the assumption that it is only messages of one fixed length that are being MACed. It is well known that CBC MAC is not secure when message lengths can vary [1]. Petrank and Rackoff [8] were the first to rigorously address issues of message length variability. They provided the provable security of EMAC (Encrypted CBC MAC) which handles messages of variable unknown lengths. Black and Rogaway [2] introduced three refinements to EMAC with optimizing efficiency on arbitrary bit strings. They also showed the provable security of them by using new techniques which regard EMAC as an instance of the Carter-Wegman paradigm [3, 10]. Recently, several new modes, PMAC (Parallelizable MAC) of Rogaway [9] and XECB-MAC of Gligor and Donescu [4] etc., which are provably secure, are proposed in Modes of Operation Workshop of NIST.

3GPP MAC is similar to EMAC, except that the MAC scheme uses a pair of key K and K', where K' is derived from K, instead of two independent keys K and K', and the input of the final block computation is the exclusive or of the outputs from all block computations instead of CBC MAC value. These make main difficulties in the proof of security of 3GPP MAC. In this paper we prove that 3GPP MAC is secure in the sense of reduction-based crypto-

† 정 회 원 : 한국전자통신연구원 선임연구원
†† 정 회 원 : 한국전자통신연구원 정보보호기반연구부장/책임연구원
논문접수 : 2002년 1월 29일, 심사완료 : 2002년 5월 24일

graphy. Specifically, we prove that 3GPP MAC is a pseudo-random function which means that no attacker with polynomially many encryption queries can distinguish between the 3GPP MAC and the perfect random function. Using this fact we show that 3GPP MAC is a secure MAC under the assumption that the underlying block cipher is a pseudorandom permutation. This assumption is reasonable since the pseudorandomness of the 3GPP block cipher KASUMI was lately proved by Kang et al. [5, 6].

2. Preliminaries

2.1 Notations

A family of functions is a function $F: \boldsymbol{K} \times A \rightarrow B$ where $\boldsymbol{K}$ is a finite set of strings and $A, B \subseteq \{0,1\}^*$. To pick a function f at random from a family F means to pick a key K uniformly at random from $\boldsymbol{K}$ and let $f \leftarrow F(K, \cdot)$; we write $f \xleftarrow{R} F$ for this operation. Let $R_{n \rightarrow n}$ be the family of all functions from $\{0,1\}^n$ to $\{0,1\}^n$ and $R_{n^* \rightarrow l}$ be the family of all functions from $(\{0,1\}^n)^*$ to $\{0,1\}^l$. Similarly, let $Perm_n$ be the all permutations on n-bit strings. A block cipher can be regarded as a family of permutations on a message space indexed by a secret key. We consider a block cipher as a function $E: \boldsymbol{K} \times \{0,1\}^n \rightarrow \{0,1\}^n$ where $E_K(\cdot) = E(K, \cdot)$ is a permutation on $\{0,1\}^n$.

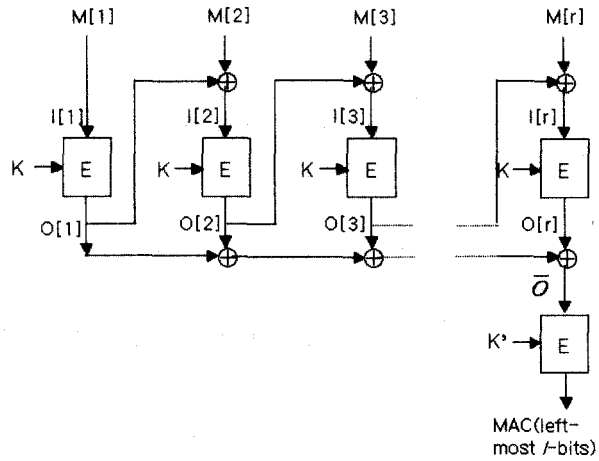
Let F, G be two function families. An adversary $\boldsymbol{A}$ distinguishing between F and G has access to an oracle f and, at the end of its computation, outputs one bit. The oracle f will be chosen either from F or from G, and the purpose of the adversary $\boldsymbol{A}$ is to distinguish whether the oracle f obtained from F or G. The advantage of $\boldsymbol{A}$ in distinguishing F from G is defined by

$$Adv_A(F, G) = |P_{f \xleftarrow{R} F}[A = 1] - P_{f \xleftarrow{R} G}[A = 1]|.$$

Here $P_{f \xleftarrow{R} F(\text{or } G)}[A = 1]$ is the probability that $\boldsymbol{A}$ output 1 when f is selected at random from F (or G).

2.2 3GPP MAC scheme

3GPP MAC scheme operates as follows. Suppose the underlying block cipher E has n - bit blocks. A message M is first padded and split into a sequence of r n-bit blocks : $M[1], \cdots, M[r]$. Throughout this paper we assume that length of each message is a multiple of n. The 3GPP MAC scheme uses a pair of keys K and K', where $K' = K \oplus Const$ and $Const = 0xAA\cdots A$. For any r-block message $M = M[1]\cdots M[r]$ the 3GPP MAC computation is as follows. Also see (Figure 1).



(Figure 1) 3GPP Integrity Algorithm $f9$

$O_K[0] \leftarrow 0^n$
for $i = 1, \cdots, r$ do
$I[i] \leftarrow O_K[i-1] \oplus M[i]$
$O_K[i] \leftarrow E_K(I[i])$
$\overline{O}_K \leftarrow O_K[1] \oplus O_K[2] \oplus \cdots \oplus O_K[r]$
$3GPPMAC_K(M) \leftarrow$ the left-most l bits of $E_{K'}(\overline{O}_K)$
return $3GPPMAC_K(M)$

In this structure the exclusive or of the outputs from all block computations is xored to the input of the final block computation. The MAC on the input message consists of the left-most l bits of the output n-bit from this computation. Note that the 3GPP integrity algorithm f9 specified in the 3GPP technical specification[11] provide the KASUMI that produces a 64-bit output from a 64-bit input under the control of an 128-bit key, as the underlying block cipher and use left-most 32 bits as the 3GPP MAC value.

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

$3GPPMAC_f(M) \leftarrow$ the left-most l bits of $\overline{f}(\overline{O}_f)$,
where $\overline{O}_f = O_f[1] \oplus O_f[2] \oplus \cdots \oplus O_f[r]$,
$O_f[i] = f(I[i])$, and $I[i] = O_f[i-1] \oplus M[i]$ for $1 \le i \le r$.

We consider two function families related to 3GPP MAC. A family $3GPPMAC_{Perm_n}$ is a set of functions $3GPPMAC_f$ for all $f \in Perm_n$ and a family $3GPPMAC_F$ for a block cipher F is a set of functions $3GPPMAC_f$ for all $f \in F$.

3. Security of 3GPP MAC

3.1 Main results

In this section we show that the security of $3GPPMAC_F$ is reduced to the security of the underlying block cipher F. This reduction-based approach on the security of MAC was introduced by Bellare et al. [1]. We call a block cipher is secure if it is a pseudorandom permutation which means that no attacker with polynomially many encryption queries can distinguish between block cipher and perfect random permutation. This approach for modeling the security of a block cipher was suggested by Luby and Rackoff [7].

To present the theorem we define what it means by that the adversary succeeds in breaking the 3GPP MAC scheme. Throughout the paper we denote by n the length of blocks in block cipher F. Also, we denote by $|M|$ the number of blocks in the string M.

[Definition 1]

Let $F : \boldsymbol{K} \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a block cipher and $\boldsymbol{A}$ be a probabilistic oracle Turing machine (the adversary). Consider the following stochastic experiment. First, a function f is selected randomly from F. Then $\boldsymbol{A}$ gets to see the $3GPPMAC_f$ value of messages $M_1, M_2, \cdots, M_{q-1}$ which it chooses adaptively, i.e., M_i is chosen based upon the values $3GPPMAC_f(M_j)$, $j = 1, \cdots, i-1$ and upon the random tape of $\boldsymbol{A}$. We say that an adversary $\boldsymbol{A}(t, q, \sigma, \varepsilon)$ breaks the $3GPPMAC_F$ scheme if for an adversary $\boldsymbol{A}$ who runs in time at most t and makes at most q oracle queries which satisfying $\sum_{i=1}^{q} |M_i| \leq \sigma$,

$$P[\ A \text{ outputs } (M_q, 3GPPMAC_f(M_q))\ ] \geq \varepsilon$$

where M_q is different from all the previous queries $M_1, M_2, \cdots, M_{q-1}$. Here the probability is defined over the random choice of $f \in F$ and the random tape of $\boldsymbol{A}$.

We first start by checking the possibility of distinguishing a random function in $R_{n* \rightarrow l}$ from a random function in $3GPPMAC_{Perm_n}$. We show that even a computationally unbounded adversary cannot gain too much advantage.

[Theorem 1]

Let $\boldsymbol{A}$ be an adversary that makes queries to a random function chosen either from $3GPPMAC_{Perm_n}$ or from $R_{n* \rightarrow}$. Suppose that $\boldsymbol{A}$ asks its oracle q queries, these queries having aggregate length of σ blocks. Then

$$Adv_A(3GPPMAC_{Perm_n}, R_{n* \rightarrow l}) \leq \frac{1.5\sigma^2 + q^2}{2^{n-1}}$$

The above theorem gives the information-theoretic bound on the security of 3GPP MAC. The proof of Theorem 1 is in Section 3.2. As we know if MAC algorithm preserves pseudorandomness, the MAC resists existential forgery under adaptively chosen message attack[1]. Using this fact and Theorem 1 we can obtain the following main result.

[Theorem 2]

Let $F \subseteq Perm_n$ be a family of permutations. Let $\varepsilon, t \geq 0$ be real numbers and q, σ be positive integers. If there exists an adversary $\boldsymbol{A}$ that $(t, q, \sigma, \varepsilon)$ breaks $3GPPMAC_F$, then there exists an adversary $\boldsymbol{A}'$ distinguishing F from $Perm_n$ with the following property.

$$Adv_{A'}(F, Perm_n) \geq \varepsilon - \frac{1.5\sigma^2 + q^2}{2^{n-1}} - \frac{1}{2^l}.$$

Here $\boldsymbol{A}'$ makes at most σ queries and runs in time at most t', where $t' = t + O(\sigma n)$.

The block cipher KASUMI, which is the underlying block cipher of 3GPP MAC scheme, is a pseudorandom permutation family [5, 6]. This implies an inability to forge 3GPP MAC with good probability.

Proof of Theorem 2 : We are given an adversary $\boldsymbol{A}$ that $(t, q, \sigma, \varepsilon)$ breaks $3GPPMAC_F$. From this adversary we can easily build an adversary $\boldsymbol{A}''$ that distinguishing $3GPPMAC_F$ from $R_{n* \rightarrow l}$ and having

$$Adv_{A''}(3GPPMAC_F, R_{n* \rightarrow l}) \geq \varepsilon - 2^{-l} \qquad (3.1)$$

Let $\boldsymbol{A}''$ be given an oracle for a function $g : (\{0,1\}^n)^* \rightarrow \{0,1\}^l$. The adversary $\boldsymbol{A}''$ uses its oracle to answer $\boldsymbol{A}$'s queries. When $\boldsymbol{A}$ makes its first oracle query M_1, $\boldsymbol{A}''$ pauses and computes $g(M_1)$ using its own oracle g. The value $g(M_1)$ is returned to $\boldsymbol{A}$ and the execution of the latter continues in this way until all its oracle queries are answered. Finally $\boldsymbol{A}''$ takes the output of $\boldsymbol{A}$, (M_q, γ), and checks if the forgery is successful by asking its oracle whether $g(M_q) = \gamma$. If the forgery is successful, $\boldsymbol{A}''$ outputs

1, and otherwise 0. Since $A(t, q, \sigma, \varepsilon)$ breaks $3GPPMAC_F$,

$$P_{g \xleftarrow{R} 3GPPMAC_F}[A'' = 1] \geq \varepsilon$$

and

$$P_{g \xleftarrow{R} R_{n^* \to l}}[A'' = 1] \geq 2^{-l}$$

which complete the equation (3.1). The aggregate length of all queries of A'' is at most σ and A'' runs in time at most $t + O(n\sigma)$.

Now we show that if we are given an adversary A'' distinguishing $3GPPMAC_F$ from $R_{n^* \to l}$, then we can build an adversary A' distinguishing F from $Perm_n$ such that

$$\begin{aligned} &Adv_{A'}(F, Perm_n) \\ &\geq Adv_{A''}(3GPPMAC_F, R_{n^* \to l}) \\ &\quad - Adv_{A''}(3GPPMAC_{Perm_n}, R_{n^* \to l}) \end{aligned} \tag{3.2}$$

and, furthermore, A' makes at most σ queries and runs in time at most t', where $t' = t + O(n\sigma)$.

Adversary A' gets an oracle $f: \{0,1\}^n \to \{0,1\}^n$. It will run A'' as a subroutine, using f to simulate the oracle $g \leftarrow 3GPPMAC_f$ that A'' expects. That is, A' will itself provide the answers to oracle queries of A'' by appropriately using f. When A'' makes its first oracle query M_1, algorithm A' pauses and computes $3GPPMAC_f(M_1)$. The value $3GPPMAC_f(M_1)$ is returned to A'' and the execution of the latter continues in this way until all its oracle queries are answered. Now A'' will output its guess bit b. Adversary A' simply returns the same as its own guess bit. We know that A' makes at most σ oracle queries and runs in time at most t'.

We now proceed to the analysis.

$$\begin{aligned} &Adv_{A'}(F, Perm_n) \\ &= |P_{f \xleftarrow{R} F}[A' = 1] - P_{f \xleftarrow{R} Perm_n}[A' = 1]| \\ &= |P_{g \xleftarrow{R} 3GPPMAC_F}[A'' = 1] \\ &\quad - P_{g \xleftarrow{R} 3GPPMAC_{Perm_n}}[A'' = 1]|. \end{aligned}$$

On the other hand,

$$\begin{aligned} &Adv_{A''}(3GPPMAC_{Perm_n}, R_{n^* \to l}) \\ &= |P_{g \xleftarrow{R} 3GPPMAC_{Perm_n}}[A'' = 1] \\ &\quad - P_{g \xleftarrow{R} R_{n^* \to l}}[A'' = 1]|. \end{aligned}$$

Take the sum of the two equation above, we obtain the equation (3.2).

Combining (3.1), (3.2) and Theorem 1 we have

$$\begin{aligned} &Adv_{A'}(F, Perm_n) \\ &\geq Adv_{A''}(3GPPMAC_F, R_{n^* \to l}) - \frac{(1.5\sigma^2 + q^2)}{2^{n-1}} \\ &\geq \varepsilon - \frac{(1.5\sigma^2 + q^2)}{2^{n-1}} - 2^{-l} \end{aligned}$$

which completes proof. □

3.2 Proof of Theorem 1

Note that the second permutation $\overline{f}$ in $3GPPMAC_f(\cdot)$ is derived from f. In order to prove Theorem 1 we first will prove the result of theorem when the second permutation $\overline{f}$ is not related to the first permutation f in the 3GPP MAC scheme. Let f and f' be chosen independently from $Perm_n$. For any r-block message $M = M[1] \cdots M[r]$ we now write

$$3GPPMAC^1_{f,f'}(M) = \text{the left-most } l \text{ bits of } f'(\overline{O}_f),$$

where $\overline{O}_f = O_f[1] \oplus O_f[2] \oplus \cdots \oplus O_f[r]$, $O_f[i] = f(I[i])$, and $I[i] = O_f[i-1] \oplus M[i]$ for $1 \leq i \leq r$. We also set that a family $3GPPMAC^1_{Perm_n}$ is a set of functions $3GPPMAC^1_{f,f'}$ for all $f, f' \in Perm_n$ where f and f' are chosen independently from $Perm_n$.

The following Lemma 1 gives the information-theoretic bound on the security of $3GPPMAC^1_{Perm_n}$.

[Lemma 1]

Let A be an adversary that makes queries to a random function chosen either from $3GPPMAC^1_{Perm_n}$ or from $R_{n^* \to l}$. Suppose that A asks its oracle q queries, these queries having aggregate length of σ blocks. Then

$$Adv_A(3GPPMAC^1_{Perm_n}, R_{n^* \to l}) \leq \frac{(1.5\sigma^2 + q^2)}{2^n}$$

To prove the Lemma 1 we apply the idea from the proof of PMAC's security in [8]. Rogaway started new proof method by measuring the pseudorandomness of PMAC in terms of two other functions : the collision probability of single messages and the collision probability of a fixed pair of messages. Second, he obtained collision bounds of two collision probabilities. We follow these steps for $3GPPMAC^1$. The main difficulty is in the proof of the first step.

Proof of Lemma 1. Let A be an adversary distinguishing $3GPPMAC^1_{Perm_n}$ from $R_{n^* \to l}$. Since the adversary A is not

limited in computational power we may assume it is deterministic. One can imagine $\boldsymbol{A}$ interacting with a $3GPPMAC^{1}_{Perm_n}$ oracle as $\boldsymbol{A}$ playing Game 1 (see (Figure 2)).

1 *unusual* ← false ; for all $x \in \{0,1\}^n$ do $f(x) \leftarrow$ undefined, $f'(x) \leftarrow$ undefined
2 When **A** makes its t-th query, $M_t = M_t[1] \cdots M_t[r_t]$ where $t \in \{1, \cdots, q\}$
3 $I_t[1] \leftarrow M_t[1]$
4 For $i = 1, \cdots, r_t$ do
5 $A \leftarrow \{I_t[j] \mid 1 \le j \le i-1\} \cup \{I_s[j] \mid 1 \le s \le t-1, 1 \le j \le r_s\}$
6 if $I_t[i] \in A$ then $O_t[i] \leftarrow f(I_t[i])$
7 else $O_t[i] \xleftarrow{R} \{0,1\}^n$
8 $A_f \leftarrow \{f(I_t[j]) \mid 1 \le j \le i-1\} \cup \{f(I_s[j]) \mid 1 \le s \le t-1, 1 \le j \le r_s\}$
9 if $O_t[i] \in A_f$ then [*unusual* ← true; $O_t[i] \xleftarrow{R} A_f^C$]
10 $f(I_t[i]) \leftarrow O_t[i]$
11 if $i < r_t$ then $I_t[i+1] \leftarrow O_t[i] \oplus M_t[i+1]$
12 $\overline{O_t} \leftarrow O_t[1] \oplus \cdots \oplus O_t[r_t]$
13 $B \leftarrow \{\overline{O_s} \mid 1 \le s \le t-1\}$
14 If $\overline{O_t} \in B$ then [*unusual* ← true ; $MAC_t \leftarrow f'(\overline{O_t})$]
15 else $MAC_t \xleftarrow{R} \{0,1\}^n$
16 $B_f \leftarrow \{f'(\overline{O_s}) \mid 1 \le s \le t-1\}$
17 if $MAC_t \in B_f$ then [*unusual* ← true; $MAC_t \xleftarrow{R} B_f^C$]
18 $f'(\overline{O_t}) \leftarrow MAC_t$
19 $3GPPMAC^{1}_{f,f}(M_t) \leftarrow$ the left-most l-bit of MAC_t
20 Return $3GPPMAC^{1}_{f,f}(M_t)$

(Figure 2) Game 1 : Simulation of $3GPPMAC^{1}_{Perm_n}$

Here we use A_f^C (or $B_{f'}^C$) to denote $\{0,1\}^n - A_f$ (or $B_{f'}$). Two particular permutations f and f' are equally likely among all permutations from $\{0,1\}^n$ to $\{0,1\}^n$. In our analysis, we view the selection of f and f' as an incremental procedure. This is equivalent to selection f and f' uniformly at random. This game perfectly simulates the behavior of $3GPPMAC^{1}_{Perm_n}$.

We observe that if the *unusual* is not set to true in an execution of the game, then the returned value $3GPPMAC^{1}_{f,f}(M_t)$ at line 20 is random since the left-most l-bit of the string randomly selected at line 15. Hence we have that

$$Adv_A(3GPPMAC^{1}_{Perm_n}, R_{n^* \to l}) \le P[\textit{unusual} \leftarrow \text{true in Game 1}] \quad (3.3)$$

First we consider the probability that *unusual* ← 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

$$P[\textit{unusual} \leftarrow \text{true in line 9 or 17 in Game 1}] \le \frac{1+2+\cdots+(\sigma-1)+1+\cdots+(q-1)}{2^n} \le \frac{\sigma^2+q^2}{2^{n+1}} \quad (3.4)$$

Now we can modify Game 1 by changing the behavior when *unusual* ← true, and adding as a compensating factor the bound given by the equation (3.4). The modified game is as Game 2 (see (Figure 3)). By the equation (3.4) we have that

$$P[\textit{unusual} \leftarrow \text{true in Game 1}] \le P[\textit{unusual} \leftarrow \text{true in Game 2}] + \frac{\sigma^2+q^2}{2^{n+1}} \quad (3.5)$$

1 *unusual* ← false ; for all $x \in \{0,1\}^n$ do $f(x) \leftarrow$ undefined, $f'(x) \leftarrow$ undefined
2 When **A** makes its t-th query, $M_t = M_t[1] \cdots M_t[r_t]$ where $t \in \{1, \cdots, q\}$
3 $I_t[1] \leftarrow M_t[1]$
4 For $i = 1, \cdots, r_t$ do
5 $A \leftarrow \{I_t[j] \mid 1 \le j \le i-1\} \cup \{I_s[j] \mid 1 \le s \le t-1, 1 \le j \le r_s\}$
6 if $I_t[i] \in A$ then $O_t[i] \leftarrow f(I_t[i])$
7 else [$O_t[i] \xleftarrow{R} \{0,1\}^n$; $f(I_t[i]) \leftarrow O_t[i]$]
8 if $i < r_t$ then $I_t[i+1] \leftarrow O_t[i] \oplus M_t[i+1]$
9 $\overline{O_t} \leftarrow O_t[1] \oplus \cdots \oplus O_t[r_t]$
10 $B \leftarrow \{\overline{O_s} \mid 1 \le s \le t-1\}$
11 If $\overline{O_t} \in B$ then *unusual* ← true
12 $MAC_t \xleftarrow{R} \{0,1\}^n$
13 $f'(\overline{O_t}) \leftarrow MAC_t$
14 $3GPPMAC^{1}_{f,f}(M_t) \leftarrow$ the left-most l-bit of MAC_t
15 Return $3GPPMAC^{1}_{f,f}(M_t)$

(Figure 3) Game 2 : Simplification of Game 1

We note that in Game 2 the value $3GPPMAC^{1}_{f,f}(M_t)$ returned in response to a query M_t is a random l-bit string. We can defer all but the selection of MAC_t values in Game 2. This does not change the probability that *unusual* ← true. This modified game is called Game 3, and it is depicted in (Figure 4).

Now we want to show that the probability of *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. We show something stronger : even if one arbitrarily fixes the values of $MAC_1, \cdots, MAC_q \in \{0,1\}^n$, the probability that *unusual* ← true is still small. Since the oracle answers have now been fixed and the adversary is deterministic, the

queries $MAC_1, \cdots, MAC_q$ that the adversary will make have likewise been fixed. The new game is called Game 4(C)(see Figure 5). It depends on constants $C = (q, MAC_1, \cdots, MAC_q, M_1, \cdots, M_q)$.

```
1  unusual ← false ; for all x ∈ {0,1}^n do f(x) ← undefined,
   f'(x) ←undefined
2  When A makes its t-th query, M_t = M_t[1] ··· M_t[r_t] where
   t ∈ {1, ···, q}
3     MAC_t ←R {0,1}^n
4     3GPPMAC^1_{f,f'}(M_t) ←the left-most l-bit of MAC_t
5     Return 3GPPMAC^1_{f,f'}(M_t)
6  When A is done making its q queries
7     For t = 1, ···, q do
8        I_t[1] ← M_t[1]
9        For i = 1, ···, r_t do
10          A ← {I_t[j] | 1 ≤ j ≤ i−1} ∪ {I_s[j] | 1 ≤ s ≤ t−1,
                1 ≤ j ≤ r_s}
11          if I_t[i] ∈ A then O_t[i] ← f(I_t[i])
12          else [ O_t[i] ←R {0,1}^n ; f(I_t[i]) ← O_t[i] ]
13          if i < r_t then I_t[i+1] ← O_t[i] ⊕ M_t[i+1]
14       O_t ← O_t[1] ⊕ ··· ⊕ O_t[r_t]
15       B ← { O_s | 1 ≤ s ≤ t−1 }
16       If O_t ∈ B then unusual ← true
17       f'( O_t ) ← MAC_t
```

(Figure 4) Game 3 : Modification of Game 2

```
1  unusual ← false ; for all x ∈ {0,1}^n do f(x) ← undefined,
   f'(x) ←undefined
2  For t = 1, ···, q do
3     I_t[1] ← M_t[1]
4     For i = 1, ···, r_t do
5        A ← {I_t[j] | 1 ≤ j ≤ i−1} ∪ {I_s[j] | 1 ≤ s ≤ t−1, 1 ≤ j ≤ r_s}
6        if I_t[i] ∈ A then O_t[i] ← f(I_t[i])
7        else [ O_t[i] ←R {0,1}^n ; f(I_t[i]) ← O_t[i]]
8        if i < r_t then I_t[i+1] ← O_t[i] ⊕ M_t[i+1]
9     O_t ← O_t[1] ⊕ ··· ⊕ O_t[r_t]
10    B ← { O_s | 1 ≤ s ≤ t−1 }
11    If O_t ∈ B then unusual ← true
12    f'( O_t ) ← MAC_t
```

(Figure 5) Game 4(C)

At this point we know that

$$P[\,unusual \leftarrow \text{true in Game 3}] \le \max_{C} \{\, P[\,unusual \leftarrow \text{true in Game 4}(C)]\} \quad (3.6)$$

Thus, by (3.3), (3.5), and (3.6) we have that

$$Adv_A(3GPPMAC^1_{Perm_n}, R_{n^* \to l}) \le \max_{C} \{\, P[\,unusual \leftarrow \text{true in Game 4}(C)]\,\} + \frac{\sigma^2 + q^2}{2^{n+1}} \quad (3.7)$$

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

Finally, we modify Game 4(C) that the flag *unusual* is set to true every case that Game 4(C), plus some additional cases. This game is called Game 5(C) (see (Figure 6)).

```
1  unusual ← false ; for all x ∈ {0,1}^n do f(x) ←undefined,
   f'(x) ←undefined
2  For t = 1, ···, q do
3     For i = 1, ···, r_t do
4        I_t[1] ← M_t[1] ; O_t[i] ←R {0,1}^n
5        A ← {I_t[j] | 1 ≤ j ≤ i−1} ∪ {I_s[j] | 1 ≤ s ≤ t−1, 1 ≤ j ≤ r_s}
6        if M_t[1] = M_s[1] for some s < t then O_t[1] ← f(I_t[1])
7        else if I_t[i] ∈ A then unusual ← true
8           else f(I_t[i]) ← O_t[i]
9        if i < r_t then I_t[i+1] ← O_t[i] ⊕ M_t[i+1]
10    O_t ← O_t[1] ⊕ ··· ⊕ O_t[r_t]
11    B ← { O_s | 1 ≤ s ≤ t−1 }
12    If O_t ∈ B then unusual ← true
13    f'( O_t ) ← 0^n
```

(Figure 6) Game 5(C)

Notice that in Game 5, we choose a random $O_t[i]$ value in line 4. First, look at line 6 in Game 4(C). The value $I_t[i]$ belongs to A for the trivial reason that $i = 1$ and $M_t[1] = M_s[1]$ for some $s < t$, or for other non-trivial reasons that $I_t[i] = I_t[j]$ for $j < i$, or $I_t[i] = I_s[j]$ for some $s < t$ except $i = j = 1$. If $I_t[i]$ belongs to A for a non-trivial reason, we effectively give up, setting *unusual*← true. To avoid that the game depend on the MAC_t-values, we also set $f'(\overline{O_t})$ 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 have that

$$P[\,unusual \leftarrow \text{true in Game 4}(C)] \le P[\,unusual \leftarrow \text{true in Game 5}(C)]. \quad (3.8)$$

The coins used in Game 5 are $O_1 = O_1[1] \cdots O_1[r_1], \cdots, O_q = O_q[1] \cdots O_q[r_q]$, where $O_t[i]$ are random coins or are $O_s[1]$ where s is the least number such that $s < t$, and $M_t[1] = M_s[1]$ when $i = 1$. Run Game 5 on $M_1, \cdots, M_q$ and the indicated vector of coins. Suppose that *unusual* gets true on this execution. Then we have the following three cases. Let $t \in \{1, \cdots, q\}$ be the particular value when *unusual* first set to true.

Case 1 : $I_t[i] = I_t[j]$ for some $j \in \{1,\cdots,i-1\}$.
Set $O_t[0] = 0$. Observe that

$$P[I_t[i] = I_t[j]] = P[O_t[i-1] \oplus O_t[j-1] = M_t[i] \oplus M_t[j]] = 2^{-n}$$

since one of two values, $O_t[i-1]$ and $O_t[j-1]$, is uniformly random, and there are r_t blocks in M_t. Thus we have

$$P[\text{ Case 1 occurs for } M_t] \le \binom{r_t}{2} 2^{-n}.$$

Case 2 : $I_t[i] = I_s[j]$ for some $s \in \{1,\cdots,t-1\}$. In this case we obtain that

$$P[I_t[i] = I_s[j]] = P[O_t[i-1] \oplus O_s[j-1] = M_t[i] \oplus M_s[j]] = 2^{-n}$$

and

$$P[\text{ Case 2 occurs for } M_t \text{ and } M_s] \le r_t \cdot r_s \cdot 2^{-n}$$

since $|M_t| \cdot |M_s| \le r_t r_s$.

Case 3 : $\overline{O_t} = \overline{O_s}$ for some $s \in \{1,\cdots,t-1\}$. Observe that

$$P[\overline{O_t} = \overline{O_s}] = P[O_t[1] \oplus \cdots \oplus O_t[r_t] = O_s[1] \oplus \cdots \oplus O_s[r_s]] = 2^{-n}$$

because $\overline{O_t}$ is uniformly random. Thus we have that

$$P[\text{ Case 3 occurs for } M_t \text{ and } M_s] \le 2^{-n}$$

Hence we obtain that

$$\begin{aligned}
&\max_C \{P[unusual \leftarrow \text{ true in Game } 5(C)]\} \\
&\le \max_{\sigma = \sum r_i} \left\{ \sum_{1 \le t \le q} \binom{r_t}{2} + \sum_{1 \le s < t \le q} r_s r_t + \sum_{1 \le s < t \le q} 1 \right\} \cdot 2^{-n} \\
&\le \left(\frac{\sigma(\sigma-1)}{2} + \frac{\sigma^2}{2} + q\frac{(q-1)}{2} \right) \cdot \frac{1}{2^n} \qquad (3.9) \\
&\le \frac{2\sigma^2 + q^2}{2^{n+1}}. \qquad (3.10)
\end{aligned}$$

Here (3.9) follows because the first sum is maximized with a single message of length σ, while the second sum is maximized by q messages of length σ/q.

Combining (3.7), (3.8) and (3.10) we have that

$$Adv_A(3GPPMAC^1_{Perm_n}, R_{n^* \to l}) \le \frac{2\sigma^2 + q^2}{2^{n+1}} + \frac{\sigma^2 + q^2}{2^{n+1}}.$$

This completes the proof of Lemma 1. □

We now check the possibility of distinguishing a random function in original $3GPPMAC_{Perm_n}$ from a random function in $3GPPMAC^1_{Perm_n}$.

[Lemma 2]

*Let **A** be an adversary that makes queries to a random function chosen either from $3GPPMAC_{Perm_n}$ or from $3GPPMAC^1_{Perm_n}$. Suppose that **A** asks its oracle q queries, these queries having aggregate length of σ blocks. Then*

$$Adv_A(3GPPMAC_{Perm_n}, 3GPPMAC^1_{Perm_n}) \le \frac{1.5\sigma^2 + q^2}{2^n}.$$

Proof of Lemma 2. Let Col be the event that there is a collision among the messages in the original $3GPPMAC$ scheme and let Col^1 be the event that there is a collision among the messages in the $3GPPMAC^1$ scheme. Observe that since the second function in both schemes is a permutation, collision probabilities in both scheme do not depend on the final computations. Thus the following equation holds :

$$P_{g \xleftarrow{R} 3GPPMAC_{Perm_n}}(\text{Col}) = P_{h \xleftarrow{R} 3GPPMAC^1_{Perm_n}}(\text{Col}^1) \qquad (3.11)$$

For the same reason, if no collision occurs, the adversary outputs 1 with same probability for $3GPPMAC$ and $3GPPMAC^1$ because she sees outputs of a random permutation on distinct points. Namely, the following holds :

$$\begin{aligned}
&P_{g \xleftarrow{R} 3GPPMAC_{Perm_n}}(A = 1 | \overline{\text{Col}}) \\
&= P_{h \xleftarrow{R} 3GPPMAC^1_{Perm_n}}(A = 1 | \overline{\text{Col}^1})
\end{aligned} \qquad (3.12)$$

where the event $\overline{\text{Col}}$ and $\overline{\text{Col}^1}$ are the complements of Col and Col^1, respectively. Therefore, by using the equation (3.11) and (3.12), we can write the adversary's advantage as follows.

$$Adv_A(3GPPMAC_{Perm_n}, 3GPPMAC^1_{Perm_n}) \le P_{g \xleftarrow{R} 3GPPMAC_{Perm_n}}(\text{Col}^1)$$

To bound this quantity, we now reconsider the proof of Lemma 1. In the proof of Lemma 1, the Game 1 perfectly simulates the behavior of $3GPPMAC^1_{Perm_n}$ and we have that

$$\begin{aligned}
&P_{g \xleftarrow{R} 3GPPMAC_{Perm_n}}(\text{Col}^1) \\
&\le P(unusual \leftarrow \text{ true in Game } 1) \\
&\le \frac{1.5\sigma^2 + q^2}{2^n}
\end{aligned}$$

which completes the proof of Lemma 2. □

Proof of Theorem 1 : From Lemma 1 and 2, Theorem 1 is proved straightforwardly. □

4. Conclusion

In this work we examined the provable security of 3GPP integrity algorithm $f9$. We proved that if there is an existential forgery attack on this MAC scheme, then the underlying block cipher can be attacked with comparable parameters. It might be seen as highly unlikely for a 3GPP block cipher KASUMI.

References

[1] M. Bellare, J. Kilian, P. Rogaway, "The security of cipher block chaining," Advances in Cryptology-Crypto'94, Springer-Verlag, LNCS 839, pp.341-358, 1994. An updated version can be found in the personal URLs of the authors. See, for example, http://www-cse.ucsd.edu/users/mihir/.

[2] J. Black and P. Rogaway, "CBC MACs for arbitrary-length messages : the three-key constructions," Advances in Cryptology-Crypto'2000, Springer-Verlag, LNCS 1880, pp.197-215, 2000.

[3] L. Carter and M. Wegman, "Universal hash functions," J. of Computer and System Sciences, Vol.18, pp.143-154, 1979.

[4] V. Gligor and P. Donescu, "Fast encryption and authentication : XCBC encryption and XECB authentication modes," Contribution to NIST, Available at http://csrc.nist.gov/encryption/modes/, April, 2001.

[5] J. Kang, S. Shin, D. Hong and O. Yi, "Provable security of KASUMI and 3GPP encryption mode $f8$," Advances in Cryptology-ASIACRYPT '2001, Springer-Verlag, LNCS 2248, pp.255-271, 2001.

[6] J. Kang, O. Yi, D. Hong, and H. Cho, "Pseudorandomness of MISTY-type transformations and the block cipher KASUMI," ACISP 2001, Springer-Verlag, LNCS 2119, pp.60-73, 2001.

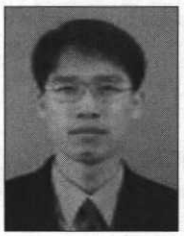
[7] M. Luby and C. Rackoff, "How to construct pseudorandom permutations and pseudorandom functions," SIAM J. Comput, Vol.17, pp.189-203, 1988.

[8] E. Petrank, C. Rackoff, "CBC MAC for Real-Time Data Source," Journal of Cryptology, Vol.13, pp.315-338, 2000.

[9] P. Rogaway, "PMAC : A parallelizable message authentication code," Contribution to NIST, Available at http://csrc.nist.gov/encryption/modes/, April, 2001.

[10] M. Wegman and L. Carter, "New hash functions and their use in authentication and set equality," J. of Computer and System Sciences, Vol.22, pp.265-279, 1981.

[11] 3G TS 35.201 "Specification of the 3GPP confidentiality and integrity algorithm ; Document 1 : f8 and f9 specifications,".



홍 도 원

e-mail : dwhong@etri.re.kr

1994년 고려대학교 이과대학 수학과(학사)

1996년 고려대학교 수학과(석사)

2000년 고려대학교 수학과(박사)

2000년~현재 한국전자통신연구원 선임연구원

관심분야 : 정보보호 이론, 이동통신 정보보호

신 상 욱

e-mail : shinsu@etri.re.kr

1995년 부산수산대학교(현 부경대학교) 전자계산학(학사)

1997년 부경대학교 전자계산학과(석사)

2000년 부경대학교 전자계산학과(박사)

2000년~현재한국전자통신연구원 선임연구원

관심분야 : 암호 이론, 정보보호론

류 희 수

e-mail : hsryu@etri.re.kr

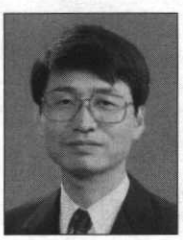
1990년 고려대학교 이과대학 수학과(학사)

1992년 고려대학교 수학과(석사)

1999년 Johns Hopkins University 수학과(박사)

2000년~현재 한국전자통신연구원 선임연구원

관심분야 : 정보보호 이론, 타원곡선암호



정 교 일

e-mail : kyoil@etri.re.kr

1981년 한양대학교 전자공학과(학사)

1983년 한양대학교 산업대학원 전자계산학과(석사)

1997년 한양대학교 전자공학과(박사)

1981년~현재 한국전자통신연구원 정보보호 기반연구부장/책임연구원

관심분야 : 정보보호, IC 카드, 생체인식, 신호처리