An extended abstract of this paper appears in *Fast Software Encryption, FSE 2004*, Lecture Notes in Computer Science, W. Meier and B. Roy editors, Springer-Verlag, 2004. This is the full version.

New Security Proofs for the 3GPP Confidentiality and Integrity Algorithms

Tetsu Iwata*   Tadayoshi Kohno†

January 26, 2004

Abstract

This paper analyses the 3GPP confidentiality and integrity schemes adopted by Universal Mobile Telecommunication System, an emerging standard for third generation wireless communications. The schemes, known as $f8$ and $f9$, are based on the block cipher KASUMI. Although previous works claim security proofs for $f8$ and $f9'$, where $f9'$ is a generalized versions of $f9$, it was recently shown that these proofs are incorrect. Moreover, Iwata and Kurosawa (2003) showed that it is *impossible* to prove $f8$ and $f9'$ secure under the standard PRP assumption on the underlying block cipher. We address this issue here, showing that it is possible to prove $f8'$ and $f9'$ secure if we make the assumption that the underlying block cipher is a secure PRP-RKA against a certain class of related-key attacks; here $f8'$ is a generalized version of $f8$. Our results clarify the assumptions necessary in order for $f8$ and $f9$ to be secure and, since no related-key attacks are known against the full eight rounds of KASUMI, lead us to believe that the confidentiality and integrity mechanisms used in real 3GPP applications are secure.

Keywords: Modes of operation, PRP-RKA, $f8$, $f9$, KASUMI, security proofs.

*Dept. of Computer and Information Sciences, Ibaraki University, 4–12–1 Nakanarusawa, Hitachi, Ibaraki 316-8511, Japan. E-mail: `iwata@cis.ibaraki.ac.jp`. URL: `http://crypt.cis.ibaraki.ac.jp/`.

†Dept. of Computer Science and Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. E-mail: `tkohno@cs.ucsd.edu`. URL: `http://www-cse.ucsd.edu/users/tkohno`. Supported by a National Defense Science and Engineering Graduate Fellowship.

Contents

1 Introduction

Background. Within the security architecture of the 3rd Generation Partnership Project (3GPP) system there are two standardized constructions: A confidentiality scheme $f8$, and an integrity scheme $f9$ [1]. 3GPP is the body standardizing the next generation of mobile telephony. Both $f8$ and $f9$ are modes of operations based on the block cipher KASUMI [2]. $f8$ is a symmetric encryption scheme which is a variant of the Output Feedback (OFB) mode with full feedback, and $f9$ is a Message Authentication Code (MAC) which is a variant of the CBC MAC.

Provable Security. Provable security is a standard security goal for block cipher modes of operations. Indeed, many of the block cipher modes of operations are provably secure assuming that the underlying block cipher is a secure pseudorandom permutation, or a super-pseudorandom permutation [21]. For example, we have: CTR mode [3] and CBC encryption mode [3] for symmetric encryption schemes, PMAC [8] and OMAC [14] for message authentication codes, and IAPM [17], OCB mode [22], CCM mode [23, 16], EAX mode [6] and CWC mode [20] for authenticated encryption schemes.

Therefore, it is natural to ask whether $f8$ and $f9$ are provably secure if the underlying block cipher is a secure pseudorandom permutation. Making this assumption, it was claimed that $f8$ is a secure symmetric encryption scheme in the sense of left-or-right indistinguishability [18] and that $f9'$ is a secure MAC [12], where $f9'$ is a generalized version of $f9$. However, these claims were disproven [15]. One of the remarkable aspects of $f8$ and $f9$ is the use of a non-zero constant called a "key modifier," or KM. In the $f8$ and $f9$ schemes, KASUMI is keyed with K and $K \oplus \mathrm{KM}$. The paper [15] constructs a secure pseudorandom permutation F with the following property: For any key K, the encryption function with key K is the decryption function with $K \oplus \mathrm{KM}$. That is, $F_K(\cdot) = F^{-1}_{K \oplus \mathrm{KM}}(\cdot)$. Then it was shown that $f8$ and $f9'$ are insecure if F is used as the underlying block cipher. This result shows that it is *impossible* to prove the security of $f8$ and $f9'$ even if the underlying block cipher is a secure pseudorandom permutation.

Our Contribution. Given the results in [15], it is logical to ask if there are assumptions under which $f8$ and $f9$ are actually secure and, if so, what those assumptions are. The answers to these questions would give us greater insights into the security of these two modes. Because of the constructions' use of keys related by fixed xor differences, the natural conjecture is that if the constructions are actually secure, then the minimum assumption on the block cipher must be that the block cipher is secure against some class of xor-restricted related-key attacks, as introduced in [7] and formalized in [5].

We prove that the above hypotheses are in fact correct and, in doing so, we clarify what assumptions are actually necessary in order for the $f8$ and $f9$ modes to be secure. In more detail, we first consider a generalized version of $f8$, which we call $f8'$. $f8'$ is a nonce-based symmetric encryption scheme, and is the natural nonce-based extension of the original $f8$. We then show that $f8'$ is a secure nonce-based deterministic symmetric encryption mode in the sense of indistinguishability from random strings if the underlying block cipher is secure against related-key attacks in which an adversary is able to obtain chosen-plaintext samples of the underlying block cipher using two keys related by a fixed known xor difference.

We next consider a generalized version of $f9$, which we call $f9'$. $f9'$ is a deterministic MAC, and is a natural extension of $f9$ that gives the user, or adversary, more liberty in controlling the input to the underlying CBC MAC core. We then show that $f9'$ is a secure pseudorandom function, which provably implies a secure MAC, if the underlying block cipher resists related-key attacks in which an adversary is able to obtain chosen-plaintext samples of the underlying block cipher using two keys related by a fixed known xor difference.

Since both $f8'$ and $f9'$ are generalized versions of $f8$ and $f9$, and, since the best known related-key attack against KASUMI breaks only six out of eight rounds [9], our results show that unless a novel new attack is discovered against KASUMI, the 3GPP confidentiality and integrity mechanisms are actually secure. We view this as an important practical corollary of our research since the 3GPP constructions are destined for use in future mobile telephony applications. Additionally, because our proofs explicitly quantify what properties of the underlying block cipher are necessary in order for $f8'$ and $f9'$ to be secure, our results can help others decide whether it is safe to instantiate the generalized 3GPP modes with block ciphers other than KASUMI. Of course, because the assumptions we make are stronger than the standard pseudorandomness assumptions, as proven necessary in [15], unless there is a significant reason to do otherwise, we suggest that future systems use more conventional modes such as CTR mode and OMAC.

For our proofs, rather than trying to find and re-use correct portions of the analyses in [18] and [12], we chose instead to prove the security of $f8'$ and $f9'$ directly. We did this in order to ensure the correctness of our results and to avoid presenting proofs covered with patches. We discuss some of problems with the previous analyses in more detail in Appendices A.1 and B.1.

An extended abstract of this paper appeared in [13].

Related Works. Initial security evaluation of KASUMI, $f8$ and $f9$ can be found in [11]. Knudsen and Mitchell analyzed the security of $f9'$ against forgery and key recovery attacks [19].

2 Preliminaries

Notation. If x is a string then $|x|$ denotes its length in bits. If x and y are two equal-length strings, then $x \oplus y$ denotes the xor of x and y. If x and y are strings, then $x \| y$ denotes their concatenation. Let $x \leftarrow y$ denote the assignment of y to x. If X is a set, let $x \stackrel{R}{\leftarrow} X$ denote the process of uniformly selecting at random an element from X and assigning it to x. If $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^m$ is a family of functions from $\{0,1\}^n$ to $\{0,1\}^m$ indexed by keys $\{0,1\}^k$, then we use the notation $F_K(D)$ as shorthand for $F(K, D)$. We say F is a family of permutations, i.e., a block cipher, if $n = m$ and $F_K(\cdot)$ is a permutation on $\{0,1\}^n$ for each $K \in \{0,1\}^k$. Let $\mathrm{Rand}(n, m)$ denote the set of all functions from $\{0,1\}^n$ to $\{0,1\}^m$. When we refer to the time of an algorithm or experiment in the provable security sections of this paper, we include the size of the code (in some fixed encoding). There is also an implicit big-$\mathcal{O}$ surrounding all such time references.

PRP-RKAs. The PRP-RKA notion was introduced in [5], and is based on the pseudorandomness notions introduced in [21] and later made concrete in [4]. The notion was designed to model block ciphers secure against related-key attacks [7].

Let $\mathrm{Perm}(k, n)$ denote the set of all block ciphers with domain $\{0,1\}^n$ and keys $\{0,1\}^k$. The notation $G \stackrel{R}{\leftarrow} \mathrm{Perm}(k, n)$ thus corresponds to selecting a random block-cipher, and comes down to defining G via

$$\text{For each } K \in \{0,1\}^k \text{ do: } \; G_K \stackrel{R}{\leftarrow} \mathrm{Perm}(n) \; ,$$

where $\mathrm{Perm}(n)$ is the set of all permutations on $\{0,1\}^n$.

Given a family of functions $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ and a key $K \in \{0,1\}^k$, we define the related-key oracle $F_{\mathrm{RK}(\cdot,K)}(\cdot)$ as an oracle that takes two arguments, a function $\phi : \{0,1\}^k \to \{0,1\}^k$

and an element $M \in \{0,1\}^n$, and that returns $F_{\phi(K)}(M)$, or the encipherment of M under the key $\phi(K)$. In this context, we shall refer to ϕ as a related-key-deriving (RKD) function.

The PRP-RKA notion, which we now describe, is parameterized by a set of RKD functions Φ. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a family of functions and let Φ be a set of RKD functions over $\{0,1\}^k$. Let $\mathcal{A}$ be an adversary with access to a related-key oracle, and restricted to queries of the form (ϕ, x) in which $\phi \in \Phi$ and $x \in \{0,1\}^n$, and let $\mathcal{A}$ return a bit. Then

$$\begin{aligned}\mathbf{Adv}^{\text{prp-rka}}_{\Phi,E}(\mathcal{A}) \stackrel{\text{def}}{=} & \Big| \Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k : \mathcal{A}^{E_{\text{RK}(\cdot,K)}(\cdot)} = 1) \\ & - \Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k \,;\, G \stackrel{R}{\leftarrow} \text{Perm}(k,n) : \mathcal{A}^{G_{\text{RK}(\cdot,K)}(\cdot)} = 1) \Big|\end{aligned}$$

is defined as the *PRP-RKA-advantage* of $\mathcal{A}$ in a Φ-restricted related-key attack (RKA) on E. Intuitively, we say that E is a *secure PRP-RKA under Φ-restricted related-key attacks* if the PRP-RKA-advantage of all adversaries using reasonable resources is small.

In this work we are primarily interested in keys that are related by some xor difference. For any $\Delta \in \{0,1\}^k$ we let $\text{XOR}_\Delta : \{0,1\}^k \to \{0,1\}^k$ denote the function which on input $K \in \{0,1\}^k$ returns $K \oplus \Delta$. We define $\Phi_k^\oplus$ as $\Phi_k^\oplus \stackrel{\text{def}}{=} \{ \text{XOR}_\Delta : \Delta \in \{0,1\}^k \}$. We briefly remark that modern block ciphers, e.g., AES [10], are designed to be secure PRP-RKAs under $\Phi_k^\oplus$-restricted related-key attacks. Additionally, the best-known $\Phi_k^\oplus$-restricted related-key attack against the block cipher KASUMI, which was designed for use with the 3GPP modes, only breaks six out of eight rounds [9].

3 Specifications of $f8$, $f8'$, $f9$ and $f9'$

3.1 3GPP Confidentiality Algorithm $f8$ [1]

$f8$ is a symmetric encryption scheme standardized by 3GPP[1]. It uses a block cipher KASUMI : $\{0,1\}^{128} \times \{0,1\}^{64} \to \{0,1\}^{64}$ as the underlying primitive. The $f8$ key generation algorithm returns a random 128-bit key K. The $f8$ encryption algorithm takes a 128-bit key K, a 32-bit counter COUNT, a 5-bit radio bearer identifier BEARER, a 1-bit direction identifier DIRECTION, and a message $M \in \{0,1\}^*$ to return a ciphertext C, which is the same length as M. Also, it uses a 128-bit constant KM $= (01)^{64}$ (or `0x55...55` in hexadecimal) called the key modifier. In more detail, the encryption algorithm is defined as follows:

Algorithm $\mathsf{f8\text{-}Encrypt}_K(\text{COUNT}, \text{BEARER}, \text{DIRECTION}, M)$
- $m \leftarrow \lceil |M|/64 \rceil$
- $Y[0] \leftarrow 0^{64}$
- $A \leftarrow \text{COUNT} \| \text{BEARER} \| \text{DIRECTION} \| 0^{26}$
- $A \leftarrow \text{KASUMI}_{K \oplus \text{KM}}(A)$
- For $i = 1$ to m do:
  - $X[i] \leftarrow A \oplus [i-1]_{64} \oplus Y[i-1]$
  - $Y[i] \leftarrow \text{KASUMI}_K(X[i])$
- $C \leftarrow M \oplus$ (the leftmost $|M|$ bits of $Y[1] \| \cdots \| Y[m]$)
- Return C

In the above description, $[i-1]_{64}$ denotes the 64-bit binary representation of $i-1$. The decryption algorithm, which takes COUNT, BEARER, DIRECTION, and a ciphertext C as input and returns a plaintext M, is defined in the natural way.

[1] The original specification [1] refers $f8$ as a symmetric synchronous stream cipher. The specification presented here is fully compatible with the original one.

Since we analyze and prove results about a variant of $f8$ whose encryption algorithm takes a nonce as input in lieu of COUNT, BEARER, and DIRECTION, we do not describe the specifics of how COUNT, BEARER, and DIRECTION are used in real 3GPP applications. We do note that 3GPP applications will never invoke the $f8$ encryption algorithm twice with the same (COUNT, BEARER, DIRECTION) triple, which means that our nonce-based variant is appropriate.

3.2 A Generalized Version of $f8$: $f8'$

$f8'$ is a nonce-based deterministic symmetric encryption scheme, which is a generalized (and weakened) version of $f8$. It uses a block cipher $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ as the underlying primitive. Let $f8'[E, \Delta]$ be $f8'$, where E is used as the underlying primitive and Δ is a non-zero k-bit key modifier. The $f8'$ key generation algorithm returns a random k-bit key K. The $f8'[E, \Delta]$ encryption algorithm, which we call $\mathsf{f8'\text{-}Encrypt}$, takes an n-bit nonce N instead of COUNT, BEARER and DIRECTION. That is, the encryption algorithm takes a k-bit key K, an n-bit nonce N, and a message $M \in \{0,1\}^*$ to return a ciphertext C, which is the same length as M. Then the encryption algorithm proceeds as follows:

$$
\begin{array}{l}
\text{Algorithm } \mathsf{f8'\text{-}Encrypt}_K(N, M) \\
\quad m \leftarrow \lceil |M|/n \rceil \\
\quad Y[0] \leftarrow 0^n \\
\quad A \leftarrow N \\
\quad A \leftarrow E_{K \oplus \Delta}(A) \\
\quad \text{For } i = 1 \text{ to } m \text{ do:} \\
\qquad X[i] \leftarrow A \oplus [i-1]_n \oplus Y[i-1] \\
\qquad Y[i] \leftarrow E_K(X[i]) \\
\quad C \leftarrow M \oplus (\text{the leftmost } |M| \text{ bits of } Y[1] \| \cdots \| Y[m]) \\
\quad \text{Return } C
\end{array}
$$

In the above description, $[i-1]_n$ denotes n-bit binary representation of $i-1$. Decryption is done in an obvious way.

Notice that we treat COUNT, BEARER and DIRECTION as a nonce. That is, as we will define in Section 4, we allow the adversary to choose these values. Consequently, $f8'$ can be considered a weakened version of $f8$ since it gives the an adversary the ability to control the entire initial value of A, rather than only a subset of the bits as would be the case for an adversary attacking $f8$.

3.3 3GPP Integrity Algorithm $f9$ [1]

$f9$ is a message authentication code standardized by 3GPP. It uses KASUMI as the underlying primitive. The $f9$ key generation algorithm returns a random 128-bit key K. The $f9$ tagging algorithm takes a 128-bit key K, a 32-bit counter COUNT, a 32-bit random number FRESH, a 1-bit direction identifier DIRECTION, and a message $M \in \{0,1\}^*$ and returns a 32-bit tag T. It uses a 128-bit constant $\mathrm{KM} = (10)^{64}$ (or `0xAA...AA` in hexadecimal), called the key modifier.

Let $M = M[1] \| \cdots \| M[m]$ be a message, where each $M[i]$ $(1 \le i \le m-1)$ is 64 bits. The last block $M[m]$ may have fewer than 64 bits. We define $\mathsf{pad}_{64}(\mathrm{COUNT}, \mathrm{FRESH}, \mathrm{DIRECTION}, M)$ as follows: It concatenates COUNT, FRESH, M and DIRECTION, and then appends a single "1" bit, followed by between 0 and 63 "0" bits so that the total length is a multiple of 64 bits. More precisely,

$$
\begin{aligned}
&\mathsf{pad}_{64}(\mathrm{COUNT}, \mathrm{FRESH}, \mathrm{DIRECTION}, M) \\
&\qquad = \mathrm{COUNT} \| \mathrm{FRESH} \| M \| \mathrm{DIRECTION} \| 1 \| 0^{63-(|M|+1 \bmod 64)} \ .
\end{aligned}
$$

Then the tagging algorithm is defined as follows:

$$
\begin{array}{l}
\text{Algorithm } \mathsf{f9\text{-}Tag}_K(\text{COUNT}, \text{FRESH}, \text{DIRECTION}, M) \\
\quad M \leftarrow \mathsf{pad}_{64}(\text{COUNT}, \text{FRESH}, \text{DIRECTION}, M) \\
\quad \text{Break } M \text{ into 64-bit blocks } M[1] \| \cdots \| M[m] \\
\quad Y[0] \leftarrow 0^{64} \\
\quad \text{For } i = 1 \text{ to } m \text{ do:} \\
\quad\quad X[i] \leftarrow M[i] \oplus Y[i-1] \\
\quad\quad Y[i] \leftarrow \text{KASUMI}_K(X[i]) \\
\quad T \leftarrow \text{KASUMI}_{K \oplus \text{KM}}(Y[1] \oplus \cdots \oplus Y[m]) \\
\quad T \leftarrow \text{the leftmost 32 bits of } T \\
\quad \text{Return } T
\end{array}
$$

The $f9$ verification algorithm is defined in the natural way.

As with $f8$, since we analyze and prove the security of a generalized version of $f9$, we do not describe how COUNT, FRESH, and DIRECTION are used in real 3GPP applications.

3.4 A Generalized Version of $f9$: $f9'$ [12, 19, 15]

The message authentication code $f9'$ is a generalized (and weakened) version of $f9$ that gives the user (or adversary) almost complete control over the input the underlying CBC MAC core. It uses a block cipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ as the underlying primitive. Let $f9'[E, \Delta, l]$ be $f9'$, where E is used as the underlying block cipher, Δ is a non-zero k-bit key modifier, and the tag length is l, where $1 \leq l \leq n$. The key generation algorithm returns a random k-bit key K. The tagging algorithm, which we call $\mathsf{f9'\text{-}Tag}$, takes a k-bit key K and a message $M \in \{0,1\}^*$ as input and returns an l-bit tag T.

Let $M = M[1] \| \cdots \| M[m]$ be a message, where each $M[i]$ $(1 \leq i \leq m-1)$ is n bits. The last block $M[m]$ may have fewer than n bits. In $f9'$, we use $\mathsf{pad'}_n$ instead of pad_{64}. $\mathsf{pad'}_n(M)$ works as follows: It simply appends a single "1" bit, followed by between 0 and $n-1$ "0" bits so that the total length is a multiple of n bits. More precisely,

$$\mathsf{pad'}_n(M) = M \| 1 \| 0^{n-1-(|M| \bmod n)} \ .$$

Thus, we simply ignore COUNT, FRESH, and DIRECTION. Equivalently, we consider COUNT, FRESH, and DIRECTION as a part of the message. The rest of the tagging algorithm is the same as with $f9$. In pseudocode,

$$
\begin{array}{l}
\text{Algorithm } \mathsf{f9'\text{-}Tag}_K(M) \\
\quad M \leftarrow \mathsf{pad'}_n(M) \\
\quad \text{Break } M \text{ into } n\text{-bit blocks } M[1] \| \cdots \| M[m] \\
\quad Y[0] \leftarrow 0^{n} \\
\quad \text{For } i = 1 \text{ to } m \text{ do:} \\
\quad\quad X[i] \leftarrow M[i] \oplus Y[i-1] \\
\quad\quad Y[i] \leftarrow E_K(X[i]) \\
\quad T \leftarrow E_{K \oplus \Delta}(Y[1] \oplus \cdots \oplus Y[m]) \\
\quad T \leftarrow \text{the leftmost } l \text{ bits of } T \\
\quad \text{Return } T
\end{array}
$$

The verification algorithm is defined in the natural way.

As we will define in Section 5, our adversary is allowed to choose COUNT, FRESH, and DIRECTION since $f9'$ treats them as a part of the message. In this sense, $f9'$ can be considered as a weakened version of $f9$.

4 Security of $f8'$

Definitions. Before proving the security of $f8'$, we must first formally define what we mean by a nonce-based encryption scheme, and what it means for such an encryption scheme to be secure.

Mathematically, a nonce-based symmetric encryption scheme $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of three algorithms and is defined for some nonce length n. The randomized key generation algorithm $\mathcal{K}$ takes no input and returns a random key K. The stateless and deterministic encryption algorithm takes a key K, an nonce $N \in \{0,1\}^n$, and a message $M \in \{0,1\}^*$ as input and returns a ciphertext C such that $|C| = |M|$; we write $C \leftarrow \mathcal{E}_K(N, M)$. The stateless and deterministic decryption algorithm takes a key K, a nonce $N \in \{0,1\}^n$, and a ciphertext $C \in \{0,1\}^*$ as input and returns a message M such that $|M| = |C|$; we write $M \leftarrow \mathcal{D}_K(N, C)$. For consistency, we require that for all keys K, nonces N, and messages M, $\mathcal{D}_K(N, \mathcal{E}_K(N, M)) = M$.

We adopt the strong notion of privacy for nonce-based encryption schemes from [22]. This notion, which we call indistinguishability from random strings, provably implies the more standard notions given in [3]. Let $\$(\cdot,\cdot)$ denote an oracle that on input a pair of strings (N, M) returns a random string of length $|M|$. If $\mathcal{A}$ is an adversary with access to an oracle, then

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathcal{SE}}(\mathcal{A}) \stackrel{\mathrm{def}}{=} \left| \Pr(K \stackrel{R}{\leftarrow} \mathcal{K} : \mathcal{A}^{\mathcal{E}_K(\cdot,\cdot)} = 1) - \Pr(\mathcal{A}^{\$(\cdot,\cdot)} = 1) \right|$$

is defined as the *PRIV-advantage* of $\mathcal{A}$ in distinguishing the outputs of the encryption algorithm with a randomly selected key from random strings. We say that $\mathcal{A}$ is nonce-respecting if it never queries its oracle twice with the same nonce value. Intuitively, we say that an encryption scheme *preserves privacy under chosen-plaintext attacks* if the PRIV-advantage of all nonce-respecting adversaries $\mathcal{A}$ using reasonable resources is small.

Provable Security Results. Let $p8'[n]$ be a variant of $f8'$ that uses random functions on n-bits instead of E_K and $E_{K \oplus \Delta}$. Specifically, the key generation algorithm for $p8'[n]$ returns two randomly selected functions R_1, R_2 from $\mathrm{Rand}(n, n)$. The encryption algorithm for $p8'[n]$, $\mathsf{p8'}$-$\mathsf{Encrypt}$, takes R_1 and R_2 as "keys" and uses them instead of E_K and $E_{K \oplus \Delta}$. The decryption algorithm is defined in the natural way.

We first upper-bound the advantage of an adversary in breaking the privacy of $p8'[n]$. Let (N_i, M_i) denote a privacy adversary's i-th oracle query. If the adversary makes exactly q oracle queries, then we define the total number of blocks for the adversary's queries as $\sigma = \sum_{i=1}^{q} \lceil |M_i|/n \rceil$.

Lemma 4.1 *Let $p8'[n]$ be as described above and let $\mathcal{A}$ be a nonce-respecting privacy adversary which asks at most q queries totaling at most σ blocks. Then*

$$\mathbf{Adv}^{\mathrm{priv}}_{p8'[n]}(\mathcal{A}) \leq \frac{\sigma^2}{2^n} \ . \tag{1}$$

A proof is given in Appendix A.

We now present our main result for $f8'$ (Theorem 4.1 below). At a high level, our theorem shows that if a block cipher E is secure against Φ-restricted related key attacks, where Φ is a small subset of $\Phi_k^{\oplus}$, then the construction $f8'[E, \Delta]$ based on E will be a provably secure encryption scheme. In more detail, our theorem states that given any adversary $\mathcal{A}$ attacking the privacy of $f8'[E, \Delta]$ and making at most q oracle queries totaling at most σ blocks, we can construct a Φ-restricted PRP-RKA adversary $\mathcal{B}$ attacking E such that $\mathcal{B}$ uses similar resources as $\mathcal{A}$ and $\mathcal{B}$ has advantage $\mathbf{Adv}^{\text{prp-rka}}_{\Phi,E}(\mathcal{B}) \geq \mathbf{Adv}^{\mathrm{priv}}_{f8'[E,\Delta]}(\mathcal{A}) - (3\sigma^2 + q^2)/2^{n+1}$. If we assume that E is secure against Φ-restricted related-key attacks and that $\mathcal{A}$ (and therefore $\mathcal{B}$) uses reasonable resources, then $\mathbf{Adv}^{\text{prp-rka}}_{\Phi,E}(\mathcal{B})$ must be small by definition, and thus $\mathbf{Adv}^{\text{priv}}_{f8'[E,\Delta]}(\mathcal{A})$ must also be small. This means that under these assumptions on E, $f8'[E,\Delta]$ is provably secure.

Since many block ciphers, including AES and KASUMI, are believed to resist $\Phi_k^{\oplus}$-restricted related-key attacks, and since Φ is a small subset of $\Phi_k^{\oplus}$, this theorem means that $f8'$ constructions built from these block ciphers will be provably secure. Additionally, because Φ is a small subset of $\Phi_k^{\oplus}$, the $f8'$ construction actually requires a much weaker assumption on the underlying block cipher than resistance to the full class of $\Phi_k^{\oplus}$-restricted related-key attacks, meaning that it is more likely for the underlying block cipher to resist Φ-restricted related-key attacks than $\Phi_k^{\oplus}$-restricted related-key attacks. Of course, our results also suggest that if a block cipher is known to be insecure under Φ-restricted related-key attacks, that block cipher should not be used in the $f8'$ construction.

Since $f8'$ is a weakened version of the KASUMI-based $f8$ encryption scheme, and since KASUMI is currently believed to resist $\Phi_k^{\oplus}$-restricted related-key attacks, our result shows that $f8$ as designed for use in the 3GPP protocols is secure.

Our main theorem statement for $f8'$ is given below.

Theorem 4.1 (Main Theorem for $f8'$) *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher and let Δ be a non-zero k-bit constant. Let $f8'[E,\Delta]$ be as described in Sec. 3.2. Let* id *be the identity function on $\{0,1\}^k$ and let $\Phi = \{\mathrm{id}, \mathrm{XOR}_\Delta\} \subseteq \Phi_k^{\oplus}$ be a set of RKD functions over $\{0,1\}^k$. If $\mathcal{A}$ is a nonce-respecting privacy adversary which asks at most q queries totaling at most σ blocks, then we can construct a Φ-restricted PRP-RKA adversary $\mathcal{B}$ against E such that*

$$\mathbf{Adv}^{\text{priv}}_{f8'[E,\Delta]}(\mathcal{A}) \leq \frac{3\sigma^2 + q^2}{2^{n+1}} + \mathbf{Adv}^{\text{prp-rka}}_{\Phi,E}(\mathcal{B}) \ . \tag{2}$$

Furthermore, $\mathcal{B}$ makes at most $\sigma + q$ oracle queries and uses the same time as $\mathcal{A}$.

Proof. Let $\mathsf{f8'\text{-}Encrypt}$ denote the encryption algorithm for $f8'[E,\Delta]$ and let $\mathsf{p8'\text{-}Encrypt}$ denote the encryption algorithm for $p8'[n]$. Expanding the definition of $\mathbf{Adv}^{\text{priv}}_{f8'[E,\Delta]}(\mathcal{A})$, we get:

$$\begin{aligned}
\mathbf{Adv}^{\text{priv}}_{f8'[E,\Delta]}(\mathcal{A}) &= \Big| \Pr(K \xleftarrow{R} \{0,1\}^k : \mathcal{A}^{\mathsf{f8'\text{-}Encrypt}_K(\cdot,\cdot)} = 1) - \Pr(\mathcal{A}^{\$(\cdot,\cdot)} = 1) \Big| \\
&= \Big| \Pr(K \xleftarrow{R} \{0,1\}^k : \mathcal{A}^{\mathsf{f8'\text{-}Encrypt}_K(\cdot,\cdot)} = 1) \\
&\quad - \Pr(R_1, R_2 \xleftarrow{R} \mathrm{Rand}(n,n) : \mathcal{A}^{\mathsf{p8'\text{-}Encrypt}_{R_1,R_2}(\cdot,\cdot)} = 1) \\
&\quad + \Pr(R_1, R_2 \xleftarrow{R} \mathrm{Rand}(n,n) : \mathcal{A}^{\mathsf{p8'\text{-}Encrypt}_{R_1,R_2}(\cdot,\cdot)} = 1) - \Pr(\mathcal{A}^{\$(\cdot,\cdot)} = 1) \Big| \\
&\leq \Big| \Pr(K \xleftarrow{R} \{0,1\}^k : \mathcal{A}^{\mathsf{f8'\text{-}Encrypt}_K(\cdot,\cdot)} = 1) \\
&\quad - \Pr(R_1, R_2 \xleftarrow{R} \mathrm{Rand}(n,n) : \mathcal{A}^{\mathsf{p8'\text{-}Encrypt}_{R_1,R_2}(\cdot,\cdot)} = 1) \Big| + \mathbf{Adv}^{\text{priv}}_{p8'[n]}(\mathcal{A}) \ .
\end{aligned}$$

Applying Lemma 4.1 we get

$$\begin{aligned}
\mathbf{Adv}^{\text{priv}}_{f8'[E,\Delta]}(\mathcal{A}) \leq & \Big| \Pr(K \xleftarrow{R} \{0,1\}^k : \mathcal{A}^{\mathsf{f8'\text{-}Encrypt}_K(\cdot,\cdot)} = 1) \\
& - \Pr(R_1, R_2 \xleftarrow{R} \mathrm{Rand}(n,n) : \mathcal{A}^{\mathsf{p8'\text{-}Encrypt}_{R_1,R_2}(\cdot,\cdot)} = 1) \Big| + \frac{\sigma^2}{2^n} \ .
\end{aligned}$$

Let $\mathcal{B}$ be a Φ-restricted related-key adversary against E that runs $\mathcal{A}$ and that returns the same bit that $\mathcal{A}$ returns. Let $F_{\mathrm{RK}(\cdot,K)}(\cdot)$ denote $\mathcal{B}$'s related-key oracle. When $\mathcal{A}$ makes an oracle query (N, M) to its oracle, $\mathcal{B}$ essentially computes the $\mathsf{f8'\text{-}Encrypt}$ algorithm, except that it uses its related-key oracle in place of E_K and $E_{K\oplus\Delta}$. In pseudocode,

Algorithm $\mathcal{B}^{F_{\text{RK}(\cdot,K)}(\cdot)}$
Run $\mathcal{A}$, replying to $\mathcal{A}$'s oracle queries (N, M) as follows:
  $m \leftarrow \lceil |M|/n \rceil$
  $Y[0] \leftarrow 0^n$
  $A \leftarrow N$
  $A \leftarrow F_{\text{RK}(\text{XOR}_\Delta,K)}(A)$
  For $i = 1$ to m do:
    $X[i] \leftarrow A \oplus [i-1]_n \oplus Y[i-1]$
    $Y[i] \leftarrow F_{\text{RK}(\text{id},K)}(X[i])$
  $C \leftarrow M \oplus$ (the leftmost $|M|$ bits of $Y[1] \| \cdots \| Y[m]$)
  Return C to $\mathcal{A}$
When $\mathcal{A}$ outputs b:
  output b

We now observe that

$$\Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k : \mathcal{A}^{\mathsf{f8'\text{-}Encrypt}_K(\cdot,\cdot)} = 1) = \Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k : \mathcal{B}^{E_{\text{RK}(\cdot,K)}(\cdot)} = 1)$$

since $\mathcal{B}$, when given related-key oracle access to E with a randomly selected key K, responds to $\mathcal{A}$ exactly as the $\mathsf{f8'\text{-}Encrypt}_K(\cdot,\cdot)$ oracle would respond with a randomly selected key K.

Let $\text{Rand}(k,n,n)$ denote the set of all functions from $\{0,1\}^k \times \{0,1\}^n$ to $\{0,1\}^n$. Then the equation

$$\begin{aligned}\Pr(R_1, R_2 \stackrel{R}{\leftarrow} \text{Rand}(n,n) : \mathcal{A}^{\mathsf{p8'\text{-}Encrypt}_{R_1,R_2}(\cdot,\cdot)} = 1)& \\ = \Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k \,;\, G \stackrel{R}{\leftarrow} \text{Rand}(k,n,n) : \mathcal{B}^{G_{\text{RK}(\cdot,K)}(\cdot)} = 1)&\end{aligned}$$

follows from the fact that when G is randomly selected from $\text{Rand}(k,n,n)$, regardless of the key K and since we assume $\Delta \neq 0^k$, G_K and $G_{K\oplus\Delta}$ are both randomly selected functions from $\text{Rand}(n,n)$.

Combining the above equations, we have that

$$\begin{aligned}\mathbf{Adv}^{\text{priv}}_{f8'[E,\Delta]}(\mathcal{A}) \le \Big| &\Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k : \mathcal{B}^{E_{\text{RK}(\cdot,K)}(\cdot)} = 1) \\ &- \Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k \,;\, G \stackrel{R}{\leftarrow} \text{Rand}(k,n,n) : \mathcal{B}^{G_{\text{RK}(\cdot,K)}(\cdot)} = 1)\Big| + \frac{\sigma^2}{2^n} \\ = \Big| &\Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k : \mathcal{B}^{E_{\text{RK}(\cdot,K)}(\cdot)} = 1) \\ &- \Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k \,;\, H \stackrel{R}{\leftarrow} \text{Perm}(k,n) : \mathcal{B}^{H_{\text{RK}(\cdot,K)}(\cdot)} = 1) \\ &+ \Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k \,;\, H \stackrel{R}{\leftarrow} \text{Perm}(k,n) : \mathcal{B}^{H_{\text{RK}(\cdot,K)}(\cdot)} = 1) \\ &- \Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k \,;\, G \stackrel{R}{\leftarrow} \text{Rand}(k,n,n) : \mathcal{B}^{G_{\text{RK}(\cdot,K)}(\cdot)} = 1)\Big| + \frac{\sigma^2}{2^n} \;.\end{aligned}$$

Using the PRP-RKA definition and applying a variant of the PRF/PRP switching lemma from [5], we get

$$\mathbf{Adv}^{\text{priv}}_{f8'[E,\Delta]}(\mathcal{A}) \le \mathbf{Adv}^{\text{prp-rka}}_{\Phi,E}(\mathcal{B}) + \frac{\sigma(\sigma-1)}{2^{n+1}} + \frac{q(q-1)}{2^{n+1}} + \frac{\sigma^2}{2^n} \;.$$

For the application of the PRF/PRP switching lemma, we note that $\mathcal{B}$ queries its related-key oracle with the RKD function id at most σ times and the RKD function XOR_Δ at most q times. Rearranging the above equation and simplifying gives (2), as desired. Q.E.D.

5 Security of $f9'$

Definitions. Before proving the security of $f9'$, we must first formally define what we mean by a MAC, and what it means for a MAC to be secure.

Mathematically, a message authentication scheme or MAC $\mathcal{MA} = (\mathcal{K}, \mathcal{T}, \mathcal{V})$ consists of three algorithms and is defined for some tag length l. The randomized key generation algorithm $\mathcal{K}$ takes no input and returns a random key K. The stateless and deterministic tagging algorithm takes a key K and a message $M \in \{0,1\}^*$ as input and returns a tag $T \in \{0,1\}^l$; we write $T \leftarrow \mathcal{T}_K(M)$. The stateless and deterministic verification algorithm takes a key K, a message $M \in \{0,1\}^*$, and a candidate tag $T \in \{0,1\}^l$ as input and returns a bit b; we write $b \leftarrow \mathcal{V}_K(M, T)$. For consistency, we require that for all keys K and messages M, $\mathcal{V}_K(M, \mathcal{T}_K(M)) = 1$.

For security, we adopt a strong notion of security for MACs, namely pseudorandomness (PRF). In [4] it was proven that if a MAC is secure PRF, then it is also unforgeable. If $\mathcal{A}$ is an adversary with access to an oracle, then

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathcal{MA}}(\mathcal{A}) \stackrel{\mathrm{def}}{=} \left| \Pr(K \stackrel{R}{\leftarrow} \mathcal{K} : \mathcal{A}^{\mathcal{T}_K(\cdot)} = 1) - \Pr(g \stackrel{R}{\leftarrow} \mathrm{Rand}(*, l) : \mathcal{A}^{g(\cdot)} = 1) \right|$$

is defined as the *PRF-advantage* of $\mathcal{A}$ in distinguishing the outputs of the tagging algorithm with a randomly selected key from the outputs of a random function with the same domain and range. Intuitively, we say that a message authentication code is *pseudorandom* or secure if the PRF-advantage of all adversaries $\mathcal{A}$ using reasonable resources is small.

Provable Security Results. Let $p9'[n]$ be a variant of $f9'$ that always outputs a full n-bit tag and that uses random functions on n-bits instead of E_K and $E_{K\oplus\Delta}$. Specifically, the key generation algorithm for $p9'[n]$ returns two randomly selected functions R_1, R_2 from $\mathrm{Rand}(n, n)$. The tagging algorithm for $p9'[n]$, $\mathsf{p9'\text{-}Tag}$, takes R_1 and R_2 as "keys" and uses them instead of E_K and $E_{K\oplus\Delta}$. The verification algorithm is defined in the natural way.

We first upper-bound the advantage of an adversary in attacking the pseudorandomness of $p9'[n]$. Let M_i denote an adversary's i-th oracle query. If an adversary makes exactly q oracle queries, then we define the total number of blocks for the adversary's queries as $\sigma = \sum_{i=1}^{q} \lceil |M_i|/n \rceil$.

Lemma 5.1 *Let $p9'[n]$ be as described above and let $\mathcal{A}$ be an adversary which asks at most q queries totaling at most σ blocks. Then*

$$\mathbf{Adv}^{\mathrm{prf}}_{p9'[n]}(\mathcal{A}) \leq \frac{\sigma^2 + q^2}{2^{n+1}} \ . \tag{3}$$

A proof is given in Appendix B.

We now present our main result for $f9'$ (Theorem 5.1), which we interpret as follows: our theorem shows that if a block cipher E is secure against Φ-restricted related-key attacks, where Φ is a small subset of $\Phi^{\oplus}_k$, then the construction $f9'[E, \Delta, l]$ based on E will be a provably secure message authentication code. In more detail, we show that given any adversary $\mathcal{A}$ attacking $f9'[E, \Delta, l]$ and making at most q oracle queries totaling at most σ blocks, we can construct a Φ-restricted PRP-RKA adversary $\mathcal{B}$ against E such that $\mathcal{B}$ uses similar resources as $\mathcal{A}$ and $\mathcal{B}$ has advantage $\mathbf{Adv}^{\text{prp-rka}}_{\Phi,E}(\mathcal{B}) \geq \mathbf{Adv}^{\mathrm{prf}}_{f9'[E,\Delta,l]}(\mathcal{A}) - (3q^2 + 2\sigma^2 + 2\sigma q)/2^{n+1}$. If we assume that E is secure against Φ-restricted related-key attacks and that $\mathcal{A}$ (and therefore $\mathcal{B}$) uses reasonable resources, then $\mathbf{Adv}^{\text{prp-rka}}_{\Phi,E}(\mathcal{B})$ must be small by definition. Therefore $\mathbf{Adv}^{\mathrm{prf}}_{f9'[E,\Delta,l]}(\mathcal{A})$ must be small as well, proving that under these assumptions on E, $f9'[E, \Delta, l]$ is secure.

Since many block ciphers, including AES and KASUMI, are believed to resist $\Phi^{\oplus}_k$-restricted related-key attacks, and since Φ is a small subset of $\Phi^{\oplus}_k$, this theorem means that $f9'$ constructions built from these block ciphers will be provably secure. Furthermore, because $f9'$ is a weakened version of the KASUMI-based $f9$ message authentication code, our result shows that $f9$ as designed for use in the 3GPP protocols is secure.

The precise theorem statement is as follows:

Theorem 5.1 (Main Theorem for $f9'$) *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher, let Δ be a non-zero k-bit constant, and let l, $1 \le l \le n$, be a constant. Let $f9'[E,\Delta,l]$ be as described in Sec. 3.4. Let id be the identity function on $\{0,1\}^k$ and let $\Phi = \{\mathrm{id}, \mathrm{XOR}_\Delta\} \subseteq \Phi_k^{\oplus}$ be a set of RKD functions over $\{0,1\}^k$. If $\mathcal{A}$ is a PRF adversary which asks at most q queries totaling at most σ blocks, then we can construct a Φ-restricted PRP-RKA adversary $\mathcal{B}$ against E such that*

$$\mathbf{Adv}^{\mathrm{prf}}_{f9'[E,\Delta,l]}(\mathcal{A}) \le \frac{3q^2 + 2\sigma^2 + 2\sigma q}{2^{n+1}} + \mathbf{Adv}^{\text{prp-rka}}_{\Phi,E}(\mathcal{B}) \ . \tag{4}$$

Furthermore, $\mathcal{B}$ makes at most $\sigma + 2q$ oracle queries and uses the same time as $\mathcal{A}$.

Proof. We first note that given any PRF adversary $\mathcal{A}$ against $f9'[E,\Delta,l]$, we can construct a PRF adversary $\mathcal{C}$ against $f9'[E,\Delta,n]$ such that the following equation holds

$$\mathbf{Adv}^{\mathrm{prf}}_{f9'[E,\Delta,l]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{prf}}_{f9'[E,\Delta,n]}(\mathcal{C}) \ . \tag{5}$$

This standard result follows from the fact that the extra bits provided to the adversary can only improve its chance of success.

Our approach to upper-bounding $\mathbf{Adv}^{\mathrm{prf}}_{f9'[E,\Delta,n]}(\mathcal{C})$ is similar to the approach we used to upper-bound $\mathbf{Adv}^{\mathrm{priv}}_{f8'[E,\Delta]}(\mathcal{A})$ in the proof of Theorem 5.1. Let $\mathsf{f9'\text{-}Tag}$ denote the tagging algorithm for $f9'[E,\Delta,n]$ and let $\mathsf{p9'\text{-}Tag}$ denote the tagging algorithm for $p9'[n]$. Expanding the definition of $\mathbf{Adv}^{\mathrm{prf}}_{f9'[E,\Delta,n]}(\mathcal{C})$ and applying Lemma 5.1, we get:

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{prf}}_{f9'[E,\Delta,n]}(\mathcal{C}) &= \Big| \Pr(K \xleftarrow{R} \{0,1\}^k : \mathcal{C}^{\mathsf{f9'\text{-}Tag}_K(\cdot)} = 1) - \Pr(g \xleftarrow{R} \mathrm{Rand}(*,n) : \mathcal{C}^{g(\cdot)} = 1) \Big| \\
&= \Big| \Pr(K \xleftarrow{R} \{0,1\}^k : \mathcal{C}^{\mathsf{f9'\text{-}Tag}_K(\cdot)} = 1) \\
&\qquad - \Pr(R_1, R_2 \xleftarrow{R} \mathrm{Rand}(n,n) : \mathcal{C}^{\mathsf{p9'\text{-}Tag}_{R_1,R_2}(\cdot)} = 1) \\
&\qquad + \Pr(R_1, R_2 \xleftarrow{R} \mathrm{Rand}(n,n) : \mathcal{C}^{\mathsf{p9'\text{-}Tag}_{R_1,R_2}(\cdot)} = 1) \\
&\qquad - \Pr(g \xleftarrow{R} \mathrm{Rand}(*,n) : \mathcal{C}^{g(\cdot)} = 1) \Big| \\
&\le \Big| \Pr(K \xleftarrow{R} \{0,1\}^k : \mathcal{C}^{\mathsf{f9'\text{-}Tag}_K(\cdot)} = 1) \\
&\qquad - \Pr(R_1, R_2 \xleftarrow{R} \mathrm{Rand}(n,n) : \mathcal{C}^{\mathsf{p9'\text{-}Tag}_{R_1,R_2}(\cdot)} = 1) \Big| + \frac{\sigma^2 + q^2}{2^{n+1}} \ .
\end{aligned}$$

As with the proof of Lemma 5.1, let $\mathcal{B}$ be a Φ-restricted related-key adversary against E that runs $\mathcal{C}$ and that returns the same bit that $\mathcal{C}$ returns. Let $F_{\mathrm{RK}(\cdot,K)}(\cdot)$ denote $\mathcal{B}$'s related-key oracle. This time, when $\mathcal{C}$ makes an oracle query (N, M) to its oracle, $\mathcal{B}$ essentially computes the $\mathsf{f9'\text{-}Tag}$ algorithm, except that it uses its related-key oracle in place of E_K and $E_{K\oplus\Delta}$. In pseudocode,

Algorithm $\mathcal{B}^{F_{\mathrm{RK}(\cdot,K)}(\cdot)}$
Run $\mathcal{C}$, replying to $\mathcal{C}$'s oracle queries M as follows:
$M \leftarrow \mathsf{pad}'_n(M)$
Break M into n-bit blocks $M[1]\|\cdots\|M[m]$
$Y[0] \leftarrow 0^n$
For $i = 1$ to m do:
$X[i] \leftarrow M[i] \oplus Y[i-1]$
$Y[i] \leftarrow F_{\mathrm{RK}(\mathrm{id},K)}(X[i])$
$T \leftarrow F_{\mathrm{RK}(\mathrm{XOR}_\Delta,K)}(Y[1] \oplus \cdots \oplus Y[m])$
Return T to $\mathcal{C}$
When $\mathcal{C}$ outputs b:
output b

We first observe that when $\mathcal{B}$ is given related-key oracle access to E with key K, it replies to $\mathcal{C}$'s oracle queries exactly as $\mathsf{f9'\text{-}Tag}_K(\cdot)$ does. This means that the following equation holds:

$$\Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k : \mathcal{C}^{\mathsf{f9'\text{-}Tag}_K(\cdot)} = 1) = \Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k : \mathcal{B}^{E_{\mathrm{RK}(\cdot,K)}(\cdot)} = 1)\ .$$

We also observe that when $\mathcal{B}$ is given related-key oracle access to G with key K, where G is a randomly selected function family from $\mathrm{Rand}(k,n,n)$, the functions $G_K(\cdot)$ and $G_{K\oplus\Delta}(\cdot)$ are both randomly selected from $\mathrm{Rand}(n,n)$. This means that $\mathcal{B}$ replies to $\mathcal{C}$'s oracle queries exactly as $\mathsf{p9'\text{-}Tag}_{R_1,R_2}(\cdot)$ would with two randomly selected functions R_1, R_2 from $\mathrm{Rand}(n,n)$. Consequently, the following equation holds:

$$\begin{aligned}&\Pr(R_1, R_2 \stackrel{R}{\leftarrow} \mathrm{Rand}(n,n) : \mathcal{C}^{\mathsf{p9'\text{-}Tag}_{R_1,R_2}(\cdot)} = 1)\\ &\qquad = \Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k\ ;\ G \stackrel{R}{\leftarrow} \mathrm{Rand}(k,n,n) : \mathcal{B}^{G_{\mathrm{RK}(\cdot,K)}(\cdot)} = 1)\end{aligned}$$

Combining these equations, we have that

$$\begin{aligned}\mathbf{Adv}^{\mathrm{prf}}_{f9'[E,\Delta,n]}(\mathcal{C}) \le \Big|\, & \Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k : \mathcal{B}^{E_{\mathrm{RK}(\cdot,K)}(\cdot)} = 1)\\ & - \Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k\ ;\ H \stackrel{R}{\leftarrow} \mathrm{Perm}(k,n) : \mathcal{B}^{H_{\mathrm{RK}(\cdot,K)}(\cdot)} = 1)\\ & + \Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k\ ;\ H \stackrel{R}{\leftarrow} \mathrm{Perm}(k,n) : \mathcal{B}^{H_{\mathrm{RK}(\cdot,K)}(\cdot)} = 1)\\ & - \Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k\ ;\ G \stackrel{R}{\leftarrow} \mathrm{Rand}(k,n,n) : \mathcal{B}^{G_{\mathrm{RK}(\cdot,K)}(\cdot)} = 1)\Big| + \frac{\sigma^2 + q^2}{2^{n+1}}\ .\end{aligned}$$

Applying the PRP-RKA definition and a variant of the PRF/PRP switching lemma from [5], we get

$$\mathbf{Adv}^{\mathrm{prf}}_{f9'[E,\Delta,n]}(\mathcal{C}) \le \mathbf{Adv}^{\mathrm{prp\text{-}rka}}_{\Phi,E}(\mathcal{B}) + \frac{(\sigma+q)\cdot(\sigma+q-1)}{2^{n+1}} + \frac{q\cdot(q-1)}{2^{n+1}} + \frac{\sigma^2+q^2}{2^{n+1}}\ .$$

For the application of the PRF/PRP switching lemma, we note that $\mathcal{B}$ queries its related-key oracle with the RKD function id at most $\sigma + q$ times and the RKD function XOR_Δ at most q times. Combining the above with equation (5) and simplifying gives the theorem statement. Q.E.D.

References

[1] 3GPP TS 35.201 v 3.1.1. Specification of the 3GPP confidentiality and integrity algorithms, Document 1: $f8$ and $f9$ specification. Available at `http://www.3gpp.org/tb/other/algorithms.htm`.

[2] 3GPP TS 35.202 v 3.1.1. Specification of the 3GPP confidentiality and integrity algorithms, Document 2: KASUMI specification. Available at `http://www.3gpp.org/tb/other/algorithms.htm`.

[3] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. Proceedings of *The 38th Annual Symposium on Foundations of Computer Science, FOCS '97,* pp. 394–405, IEEE, 1997.

[4] M. Bellare, J. Kilian, and P. Rogaway. The security of the cipher block chaining message authentication code. *JCSS,* vol. 61, no. 3, pp. 362–399, 2000. Earlier version in Y. Desmedt, editor, *Advances in Cryptology – CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 341–358. Springer-Verlag, Berlin Germany, 1994.

[5] M. Bellare, and T. Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 491–506. Springer-Verlag, Berlin Germany, 2003.

[6] M. Bellare, P. Rogaway, and D. Wagner. The EAX mode of operation. In W. Meier and B. Roy, editors, *Fast Software Encryption, FSE 2004,* Springer-Verlag, 2004.

[7] E. Biham. New types of cryptanalytic attacks using related keys. In T. Helleseth, editor, *Advances in Cryptology – EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pages 398–409. Springer-Verlag, Berlin Germany, 1993.

[8] J. Black and P. Rogaway. A block-cipher mode of operation for parallelizable message authentication. In L.R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 384–397. Springer-Verlag, Berlin Germany, 2002.

[9] M. Blunden and A. Escott. Related key attacks on reduced round KASUMI. In M. Matsui, editor, *Fast Software Encryption, FSE 2001*, volume 2355 of *Lecture Notes in Computer Science*, pages 277–285. Springer-Verlag, Berlin Germany, 2002.

[10] J. Daemen and V. Rijmen. *The Design of Rijndael.* Springer-Verlag, Berlin Germany, 2002.

[11] Evaluation report (version 2.0). Specification of the 3GPP confidentiality and integrity algorithms, Report on the evaluation of 3GPP confidentiality and integrity algorithms. Available at `http://www.3gpp.org/tb/other/algorithms.htm`.

[12] D. Hong, J-S. Kang, B. Preneel and H. Ryu. A concrete security analysis for 3GPP-MAC. In T. Johansson, editor, *Fast Software Encryption, FSE 2003*, volume 2887 of *Lecture Notes in Computer Science*, pages 154–169. Springer-Verlag, Berlin Germany, 2003.

[13] T. Iwata and T. Kohno. New security proofs for the 3GPP confidentiality and integrity algorithms. In W. Meier and B. Roy, editors, *Fast Software Encryption, FSE 2004,* Springer-Verlag, 2004.

[14] T. Iwata and K. Kurosawa. OMAC: One-Key CBC MAC. In T. Johansson, editor, *Fast Software Encryption, FSE 2003*, volume 2887 of *Lecture Notes in Computer Science*, pages 129–153. Springer-Verlag, Berlin Germany, 2003.

[15] T. Iwata and K. Kurosawa. On the correctness of security proofs for the 3GPP confidentiality and integrity algorithms. In K.G. Paterson, editor, *Cryptography and Coding, Ninth IMA International Conference*, volume 2898 of *Lecture Notes in Computer Science*, pages 306–318. Springer-Verlag, Berlin Germany, 2003.

[16] J. Jonsson. On the Security of CTR + CBC-MAC. In K. Nyberg and H.M. Heys, editors, *Selected Areas in Cryptography, 9th Annual Workshop (SAC 2002)*, volume 2595 of *Lecture Notes in Computer Science*, pages 76–93. Springer-Verlag, Berlin Germany, 2002.

[17] C.S. Jutla. Encryption modes with almost free message integrity. In B. Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 529–544. Springer-Verlag, Berlin Germany, 2001.

[18] J-S. Kang, S-U. Shin, D. Hong and O. Yi. Provable security of KASUMI and 3GPP encryption mode $f8$. In C. Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 255–271. Springer-Verlag, Berlin Germany, 2001.

[19] L.R. Knudsen and C.J. Mitchell. Analysis of 3gpp-MAC and two-key 3gpp-MAC. *Discrete Applied Mathematics*, vol. 128, no. 1, pp. 181–191, 2003.

[20] T. Kohno, J. Viega, and D. Whiting. CWC: A high-performance conventional authenticated encryption mode. In W. Meier and B. Roy, editors, *Fast Software Encryption, FSE 2004,* Springer-Verlag, 2004.

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

[22] 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.

[23] D. Whiting, R. Housley, and N. Ferguson. Counter with CBC-MAC (CCM). Submission to NIST. Available at `http://csrc.nist.gov/CryptoToolkit/modes/`.

A Proof of Lemma 4.1

Notation. We fix some notation. For q and σ in Lemma 4.1, let $m_1, \ldots, m_q$ be integers such that $m_i \geq 1$ and $\sigma \geq m_1 + \cdots + m_q$. Let $N_1, \ldots, N_q$ be fixed and distinct bit strings such that $|N_i| = n$. Let $M_1, \ldots, M_q$ be arbitrarily fixed bit strings such that $|M_i| = m_i n$, and let $M_i = M_i[1] \| \cdots \| M_i[m_i]$, where $M_i[j] \in \{0,1\}^n$. Also, let $C_1, \ldots, C_q$ be fixed bit strings such that $|C_i| = m_i n$ and, let $C_i = C_i[1] \| \cdots \| C_i[m_i]$, where $C_i[j] \in \{0,1\}^n$. Assume $C_1, \ldots, C_q$ satisfy the following condition:

$$\begin{array}{l}\text{For any } i \ (1 \leq i \leq q),\\ \quad 0^n, M_i[1] \oplus C_i[1] \oplus [1]_n, \ldots, M_i[m_i - 1] \oplus C_i[m_i - 1] \oplus [m_i - 1]_n \text{ are distinct.}\end{array} \tag{6}$$

(there is no condition on $C_1[m_1], \ldots, C_q[m_q]$).

For (N_i, M_i) and functions R_1 and R_2, let $A_i = R_1(N_i)$, and $M_i[0] \oplus C_i[0] = 0^n$. For $1 \le j \le m_i$, let $X_i[j] = A_i \oplus M_i[j-1] \oplus C_i[j-1] \oplus [j-1]_n$ and $Y_i[j] = R_2(X_i[j])$.

Further, for $1 \le i \le q$, let $\boldsymbol{X}_i \stackrel{\text{def}}{=} \{X_i[j] \mid 1 \le j \le m_i\}$ and $\boldsymbol{Y}_i \stackrel{\text{def}}{=} \{Y_i[j] \mid 1 \le j \le m_i\}$.

We first show the following lemma.

Lemma A.1 *Let q, $m_1, \ldots, m_q$, σ, $N_1, \ldots, N_q$, $M_1, \ldots, M_q$, $C_1, \ldots, C_q$ be as described above. Then*

$$\Pr(R_1, R_2 \stackrel{R}{\leftarrow} \mathit{Rand}(n,n) : 1 \le {}^\forall i \le q, \mathsf{p8'\text{-}Encrypt}_{R_1,R_2}(N_i, M_i) = C_i) \ge \frac{1}{2^{\sigma n}} \cdot \left(1 - \frac{\sigma^2}{2^{n+1}}\right) \ . \quad (7)$$

Proof. As usual, instead of choosing R_1 and R_2 uniformly at random, we choose A_i and $Y_i[j]$ in an incremental manner. We first choose $A_1, \ldots, A_q$, and then we choose $Y_1[1], \ldots, Y_1[m_1]$, and we choose $Y_2[1], \ldots, Y_2[m_2]$, and so on.

We define the following q events $\mathrm{BAD}[t]$ $(1 \le t \le q)$: Suppose that $A_1, \ldots, A_{t-1}$ are fixed (thus $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{t-1}$ are fixed) and none of $\mathrm{BAD}[1], \ldots, \mathrm{BAD}[t-1]$ occurs. For randomly chosen A_t (this will fix $\boldsymbol{X}_t$), define the following $(t-1)$ conditions: Cond. A-s $(1 \le s \le t-1)$.

Cond. A-s $(1 \le s \le t-1)$: $\boldsymbol{X}_s \cap \boldsymbol{X}_t \ne \emptyset$.

We say that $\mathrm{BAD}[t]$ occurs if at least one of the above $(t-1)$ conditions occurs.

Intuitively, Cond. A-s $(1 \le s \le t)$ ensure that currently fixed $\boldsymbol{X}_t$ is different from all the previously fixed $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{t-1}$. Notice that, from the condition on C_i in (6), there is no collision among the elements in $\boldsymbol{X}_t$. For any A_t, $\boldsymbol{X}_t$ has m_t distinct elements.

We upper bound the probability of $\mathrm{BAD}[t]$ $(1 \le t \le q)$. Now we see that

$$\Pr_{A_t}(\text{Cond. A-}s) \le \frac{m_s \cdot m_t}{2^n} \ ,$$

since there are exactly m_s elements in $\boldsymbol{X}_s$ and exactly m_t elements in $\boldsymbol{X}_t$, and these elements collide with probability 2^{-n} because of the randomness of A_t. Therefore,

$$\Pr_{A_t}(\mathrm{BAD}[t]) \le \sum_{1 \le s \le t-1} \frac{m_s \cdot m_t}{2^n} = \frac{(m_1 + \cdots + m_{t-1}) \cdot m_t}{2^n} \ .$$

Now the left hand side of (7) is lower bounded by

$$\Pr_{A_1,\ldots,A_q}(\text{none of } \mathrm{BAD}[1], \ldots, \mathrm{BAD}[q] \text{ occurs}) \cdot \frac{1}{2^{\sigma n}} \quad (8)$$

since, if none of $\mathrm{BAD}[t]$ occurs, then $\boldsymbol{X}_1 \cup \cdots \cup \boldsymbol{X}_q$ has σ distinct elements, and thus R_2 has σ distinct inputs. Then, (8) is lower bounded by

$$\frac{1}{2^{\sigma n}} \cdot \left(1 - \sum_{1 \le t \le q} \Pr_{A_t}(\mathrm{BAD}[t])\right) \ge \frac{1}{2^{\sigma n}} \cdot \left(1 - \sum_{1 \le t \le q} \frac{(m_1 + \cdots + m_{t-1}) \cdot m_t}{2^n}\right) \ .$$

Finally, we have

$$\sum_{1 \le t \le q} \frac{(m_1 + \cdots + m_{t-1}) \cdot m_t}{2^n} = \frac{(m_1 + \cdots + m_q)^2}{2^{n+1}} - \frac{m_1^2 + \cdots + m_q^2}{2^{n+1}} \le \frac{\sigma^2}{2^{n+1}} \ ,$$

and the lemma follows. Q.E.D.

We now prove Lemma 4.1.

Proof (of Lemma 4.1). Let $\mathcal{O}(\cdot,\cdot)$ be either $\mathsf{p8'\text{-}Encrypt}_{R_1,R_2}(\cdot,\cdot)$ or $\$(\cdot,\cdot)$. The adversary $\mathcal{A}$ has oracle access to $\mathcal{O}(\cdot,\cdot)$. Since $\mathcal{A}$ is computationally unbounded, there is no loss of generality to assume that $\mathcal{A}$ is deterministic. Also, there is no loss of generality to assume that $\mathcal{A}$ makes q queries, and the length of each queries is a multiple of n bits.

For the i-th query $\mathcal{A}$ makes to $\mathcal{O}(\cdot,\cdot)$, define the query-answer pair (N_i, M_i, C_i), where $\mathcal{A}$'s query was (N_i, M_i) and the answer it got was C_i.

Suppose that we run $\mathcal{A}$ with the oracle $\mathcal{O}(\cdot,\cdot)$. For this run, we define view v of $\mathcal{A}$ as

$$v \stackrel{\text{def}}{=} \langle C_1, \ldots, C_q \rangle \ . \tag{9}$$

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 σn-bit string V and return the i-th m_i blocks as the answer for the i-th query $\mathcal{A}$ makes (instead of the oracle), then

- $\mathcal{A}$'s queries $(N_1, M_1), \ldots, (N_q, M_q)$ are uniquely determined,
- the unique parsing of V into the format defined in (9) is determined, and
- the final output of $\mathcal{A}$ (0 or 1) is uniquely determined.

We note that since $\mathcal{A}$ is nonce-respecting, the corresponding $N_1, \ldots, N_q$ are distinct.

Let $\boldsymbol{V}_{one}$ be a set of all σn-bit strings V such that $\mathcal{A}$ outputs 1, and let $N_{one} \stackrel{\text{def}}{=} \#\boldsymbol{V}_{one}$. Also, let $\boldsymbol{V}_{good}$ be a set of all σn-bit strings V such that the corresponding parsing satisfies (6), and let $N_{good} \stackrel{\text{def}}{=} \#\boldsymbol{V}_{good}$.

For notational simplicity, define

$$p_{rand} \stackrel{\text{def}}{=} \Pr(\mathcal{A}^{\$(\cdot,\cdot)} = 1) \ .$$

Then we have

$$p_{rand} = \sum_{V \in \boldsymbol{V}_{one}} \Pr(1 \le {}^{\forall} i \le q, \$(N_i, M_i) = C_i) = \sum_{V \in \boldsymbol{V}_{one}} \frac{1}{2^{\sigma n}} = \frac{N_{one}}{2^{\sigma n}} \ . \tag{10}$$

Next define

$$p_{real} \stackrel{\text{def}}{=} \Pr(R_1, R_2 \stackrel{R}{\leftarrow} \mathrm{Rand}(n,n) : \mathcal{A}^{\mathsf{p8'\text{-}Encrypt}_{R_1,R_2}(\cdot,\cdot)} = 1) \ .$$

Then from Lemma A.1, we have

$$\begin{aligned}
p_{real} &= \sum_{V \in \boldsymbol{V}_{one}} \Pr(R_1, R_2 \stackrel{R}{\leftarrow} \mathrm{Rand}(n,n) : 1 \le {}^{\forall} i \le q, \mathsf{p8'\text{-}Encrypt}_{R_1,R_2}(N_i, M_i) = C_i) \\
&\ge \sum_{V \in (\boldsymbol{V}_{one} \cap \boldsymbol{V}_{good})} \Pr(R_1, R_2 \stackrel{R}{\leftarrow} \mathrm{Rand}(n,n) : 1 \le {}^{\forall} i \le q, \mathsf{p8'\text{-}Encrypt}_{R_1,R_2}(N_i, M_i) = C_i) \\
&\ge \sum_{V \in (\boldsymbol{V}_{one} \cap \boldsymbol{V}_{good})} \frac{1}{2^{\sigma n}} \cdot \left(1 - \frac{\sigma^2}{2^{n+1}}\right) \ .
\end{aligned} \tag{11}$$

We next count N_{good}. Suppose that the message of $\mathcal{A}$'s first query (N_1, M_1) has m_1 blocks. Then the first n bits of V can take any value except for $M_1[1] \oplus [1]_n$, the second n bits of V can take any value except for $M_1[2] \oplus [2]_n$ and $M_1[1] \oplus [1]_n \oplus M_1[2] \oplus [2]_n$, the third n bits of V can take any value except for $M_1[3] \oplus [3]_n$, $M_1[2] \oplus [2]_n \oplus M_1[3] \oplus [3]_n$, $M_1[1] \oplus [1]_n \oplus M_1[3] \oplus [3]_n$, and so on. In particular, at most j values are not allowed for the j-th block ($1 \le j \le m_1 - 1$), and the m_1-th block can take any value. That is, the first m_1 blocks of V can take at least

$$(2^n - 1) \cdot (2^n - 2) \cdots (2^n - m_1 - 1) \cdot (2^n) \ge 2^{m_1 n} \left(1 - \frac{m_1^2}{2^{n+1}}\right)$$

values. When we choose one of the above $2^{m_1 n}\left(1 - \frac{m_1^2}{2^{n+1}}\right)$ values, then m_2 is determined, and we have at least $2^{m_2 n}\left(1 - \frac{m_2^2}{2^{n+1}}\right)$ values for the next m_2 blocks of V. By continuing the same analysis up to q-th m_q blocks, N_{good} is at least

$$2^{(m_1+\cdots+m_q)n} \cdot \left(1 - \frac{m_1^2}{2^{n+1}}\right) \cdots \left(1 - \frac{m_q^2}{2^{n+1}}\right) \cdot 2^{(\sigma-(m_1+\cdots+m_q))n} \ge 2^{\sigma n} \cdot \left(1 - \frac{\sigma^2}{2^{n+1}}\right) \ .$$

$2^{(\sigma-(m_1+\cdots+m_q))n}$ is multiplied since, in case of $\sigma > m_1+\cdots+m_q$, the remaining $(\sigma-(m_1+\cdots+m_q))$ bits can take any value. Then we have $\#\{V \mid V \in (\boldsymbol{V}_{one} \cap \boldsymbol{V}_{good})\} \ge N_{one} - 2^{\sigma n} \cdot \frac{\sigma^2}{2^{n+1}}$, and (11) is lower bounded by

$$\left(N_{one} - 2^{\sigma n} \cdot \frac{\sigma^2}{2^{n+1}}\right) \cdot \frac{1}{2^{\sigma n}} \cdot \left(1 - \frac{\sigma^2}{2^{n+1}}\right) = \left(\frac{N_{one}}{2^{\sigma n}} - \frac{\sigma^2}{2^{n+1}}\right) \cdot \left(1 - \frac{\sigma^2}{2^{n+1}}\right) \ .$$

From (10) we have

$$p_{real} \ge \left(p_{rand} - \frac{\sigma^2}{2^{n+1}}\right) \cdot \left(1 - \frac{\sigma^2}{2^{n+1}}\right) \ge p_{rand} - \frac{\sigma^2}{2^n} \ . \tag{12}$$

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

$$1 - p_{real} \ge 1 - p_{rand} - \frac{\sigma^2}{2^n} \ . \tag{13}$$

Finally, (12) and (13) give $|p_{real} - p_{rand}| \le \sigma^2/2^n$. Q.E.D.

A.1 Discussion of the Previous Work [18]

[18, p. 269, Lemma 7] might be seen to correspond to Lemma 4.1. However, there is a problem with the definition of their encryption scheme. Their encryption scheme, which we call $p8''[n]$, is described as follows: The key generation algorithm for $p8''[n]$ returns a randomly selected permutation P_1 from $\mathrm{Perm}(n)$. The encryption algorithm for $p8''[n]$ takes P_1 as a "key" and uses P_1 and P_2 instead of E_K and $E_{K\oplus\Delta}$, but it is not defined how P_2 is derive from P_1. We note that [12, p. 166, Lemma 2] has a similar problem, which is described in Appendix B.1.

We also adopt the strong notion of privacy, indistinguishability from random strings [22]. This security notion is strictly stronger than the left-or-right indistinguishability used in [18, p. 269, Lemma 7].

We present the full security proof for $p8'[n]$ in order to achieve this strong security notion and to establish self contained security proof.

B Proof of Lemma 5.1

To prove Lemma 5.1, we define $p9'$-E$[n]$, a variant of $p9'[n]$. The tagging algorithm for $p9'$-E$[n]$ takes only messages of length multiple of n, and it does not perform the final encryption. Specifically, the key generation algorithm for $p9'$-E$[n]$ returns a randomly selected function R_1 from $\mathrm{Rand}(n,n)$. The tagging algorithm for $p9'$-E$[n]$, $\mathsf{p9'\text{-}E\text{-}Tag}$, takes R_1 as a "key" and a message M such that $|M| = mn$ for some $m \geq 1$. In pseudocode:

```
Algorithm p9'-E-Tag_{R_1}(M)
    Break M into n-bit blocks M[1]||···||M[m]
    Y[0] ← 0^n
    For i = 1 to m do:
        X[i] ← M[i] ⊕ Y[i − 1]
        Y[i] ← R_1(X[i])
    Return Y[1] ⊕ ··· ⊕ Y[m]
```

The verification algorithm is defined in the natural way.

Notation. We fix some notation. For q and σ in Lemma 5.1, let $m_1, \ldots, m_q$ be integers such that $m_i \geq 1$ and $\sigma \geq m_1 + \cdots + m_q$. Let $M_1, \ldots, M_q$ be fixed and distinct bit strings such that $|M_i| = m_i n$. Also, let $m_{\mathtt{max}} = \max\{m_1, \ldots, m_q\}$. Further, let $M_i = M_i[1], \ldots, M_i[m_i]$, where $M_i[j] \in \{0,1\}^n$. Then for $M_1, \ldots, M_q$, we define the following sequences $S[1], \ldots, S[m_{\mathtt{max}}]$ and $S'[1], \ldots, S'[m_{\mathtt{max}}]$ of integers:

$$\begin{cases} S[j] \stackrel{\text{def}}{=} \#\{(M_i[1], \ldots, M_i[j]) \mid 1 \leq i \leq q \text{ and } j \leq m_i\} \ , \text{ and} \\ S'[j] \stackrel{\text{def}}{=} \#\{i \mid 1 \leq i \leq q \text{ and } j = m_i\} \ . \end{cases}$$

Note that $S[1] + \cdots + S[m_{\mathtt{max}}] \leq \sigma$ and $S'[1] + \cdots + S'[m_{\mathtt{max}}] = q$.

Let $Y_i[0]$ be 0^n. For a function R_1 and for $j \geq 1$, let $X_i[j] = M_i[j] \oplus Y_i[j-1]$ and $Y_i[j] = R_1(X_i[j])$. See Fig. 1. Note that $\mathsf{p9'\text{-}E\text{-}Tag}_{R_1}(M_i) = Y_i[1] \oplus \cdots \oplus Y_i[m_i]$.

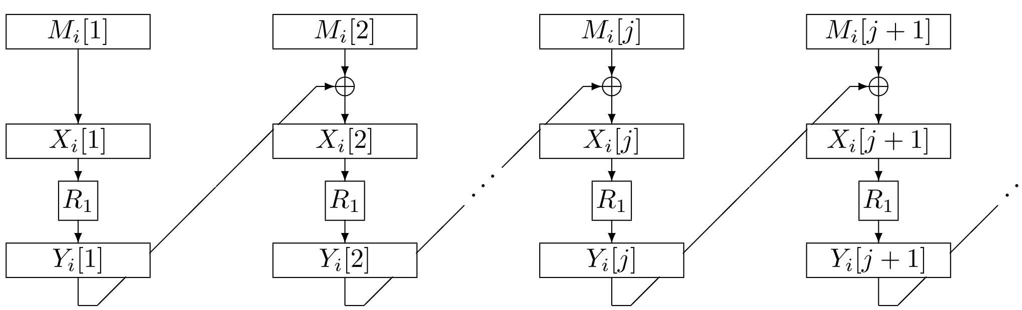

Fig. 1. The labeling convention for $\mathsf{p9'\text{-}E\text{-}Tag}_{R_1}(M_i)$.

Further, for $1 \leq j \leq m_{\mathtt{max}}$, let $\boldsymbol{X}[j] \stackrel{\text{def}}{=} \{X_i[j] \mid 1 \leq i \leq q \text{ and } j \leq m_i\}$, $\boldsymbol{Y}[j] \stackrel{\text{def}}{=} \{Y_i[j] \mid 1 \leq i \leq q \text{ and } j \leq m_i\}$, and $\boldsymbol{Z}[j] \stackrel{\text{def}}{=} \{Y_i[1] \oplus \cdots \oplus Y_i[j] \mid 1 \leq i \leq q \text{ and } j = m_i\}$.

We first show the following lemma.

Lemma B.1 *Let q, $m_1, \ldots, m_q$, σ, $M_1, \ldots, M_q$ be as described above. Then*

$$\Pr(R_1 \stackrel{R}{\leftarrow} Rand(n,n) : 1 \leq {}^{\exists}i < {}^{\exists}j \leq q, \mathsf{p9'\text{-}E\text{-}Tag}_{R_1}(M_i) = \mathsf{p9'\text{-}E\text{-}Tag}_{R_1}(M_j)) \leq \frac{\sigma^2 + q^2}{2^{n+1}} \ . \quad (14)$$

Proof. As usual, instead of choosing R_1 uniformly at random, we choose $Y_i[j]$ in an incremental manner. We first choose $Y_1[1], Y_2[1], \ldots, Y_q[1]$, and we choose $Y_1[2], Y_2[2], \ldots, Y_q[2]$, and so on.

We define the following $m_{\mathtt{max}}$ events $\mathrm{BAD}[t]$ ($1 \le t \le m_{\mathtt{max}}$): Suppose that $\boldsymbol{Y}[1], \ldots, \boldsymbol{Y}[t-1]$ are fixed (thus $\boldsymbol{X}[1], \ldots, \boldsymbol{X}[t]$ and $\boldsymbol{Z}[1], \ldots, \boldsymbol{Z}[t-1]$ are fixed), and none of $\mathrm{BAD}[1], \ldots, \mathrm{BAD}[t-1]$ occurs. For randomly chosen $\boldsymbol{Y}[t]$ (this will fix $\boldsymbol{X}[t+1]$ and $\boldsymbol{Z}[t]$), define the following $t+1+(t-1)+1 = 2t+1$ conditions: Cond. A-s ($1 \le s \le t$), Cond. B, Cond. C-s ($1 \le s \le t-1$), and Cond. D.

Cond. A-s ($1 \le s \le t$): $\boldsymbol{X}[s] \cap \boldsymbol{X}[t+1] \ne \emptyset$.

Cond. B: There exists (i, i') ($1 \le i < i' \le q$) such that

$$(M_i[1], \ldots, M_i[t+1]) \ne (M_{i'}[1], \ldots, M_{i'}[t+1])$$

and

$$Y_i[t] \oplus M_i[t+1] = Y_{i'}[t] \oplus M_{i'}[t+1] \enspace .$$

Cond. C-s ($1 \le s \le t-1$): $\boldsymbol{Z}[s] \cap \boldsymbol{Z}[t] \ne \emptyset$.

Cond. D: There exists (i, i') ($1 \le i < i' \le q$) such that

$$(M_i[1], \ldots, M_i[t]) \ne (M_{i'}[1], \ldots, M_{i'}[t])$$

and

$$Y_i[1] \oplus \cdots \oplus Y_i[t] = Y_{i'}[1] \oplus \cdots \oplus Y_{i'}[t] \enspace ,$$

where $m_i = m_{i'} = t$.

We say that $\mathrm{BAD}[t]$ occurs if at least one of the above $2t+1$ conditions occurs.

Intuitively, Cond. A-s ($1 \le s \le t$) ensure that we can randomly choose the next $\boldsymbol{Y}[t+1]$ independent of the previously fixed $\boldsymbol{Y}[1], \ldots, \boldsymbol{Y}[t]$, and Cond. B ensures that $\#\boldsymbol{X}[t+1] = S[t+1]$. Similarly, Cond. C-$s$ ($1 \le s \le t-1$) ensure that the currently fixed $Y_i[1] \oplus \cdots \oplus Y_i[t]$ (where $t = m_i$) is different from the previously fixed $Y_{i'}[1] \oplus \cdots \oplus Y_{i'}[m_{i'}]$, and Cond. D ensures that $\#\boldsymbol{Z}[t] = S'[t]$.

We note that, $\mathrm{BAD}[1]$ and $\mathrm{BAD}[m_{\mathtt{max}}]$ are slightly different from $\mathrm{BAD}[2], \ldots, \mathrm{BAD}[m_{\mathtt{max}}-1]$: we do not have to consider Cond. C-s in $\mathrm{BAD}[1]$, and we do not have to consider Cond. A-s and Cond. B in $\mathrm{BAD}[m_{\mathtt{max}}]$. Also, note that $\#\boldsymbol{Z}[t] = S[t]$ for all $1 \le t \le m_{\mathtt{max}}$ and $\boldsymbol{Z}[s] \cap \boldsymbol{Z}[t] = \emptyset$ for all $1 \le s < t \le m_{\mathtt{max}}$ imply $\mathsf{p9'\text{-}E\text{-}Tag}_{R_1}(M_i) \ne \mathsf{p9'\text{-}E\text{-}Tag}_{R_1}(M_j)$ for all $1 \le i < j \le q$.

We upper bound the probability of $\mathrm{BAD}[t]$ ($1 \le t \le m_{\mathtt{max}}$). Since none of $\mathrm{BAD}[1], \ldots, \mathrm{BAD}[t-1]$ occurs, we have $(2^n)^{S[t]}$ choice of $Y_1[t], \ldots, Y_q[t]$.

Now we see that

$$\Pr_{Y_1[t], \ldots, Y_q[t]}(\text{Cond. A-}s) \le \frac{S[s] \cdot S[t+1]}{2^n} \enspace ,$$

since there are exactly $S[s]$ elements in $\boldsymbol{X}[s]$ and at most $S[t+1]$ elements in $\boldsymbol{X}[t+1]$, and these elements collide with probability at most 2^{-n} because of the randomness of $Y_1[t], \ldots, Y_q[t]$. Next we have

$$\Pr_{Y_1[t], \ldots, Y_q[t]}(\text{Cond. B}) \le \frac{S[t+1] \cdot (S[t+1]-1)}{2^{n+1}} \le \frac{S[t+1]^2}{2^{n+1}} \enspace ,$$

since we have $\binom{S[t+1]}{2}$ choice of (i, i'). Next, we see that

$$\Pr_{Y_1[t], \ldots, Y_q[t]}(\text{Cond. C-}s) \le \frac{S'[s] \cdot S'[t]}{2^n} \enspace ,$$

since there are exactly $S'[s]$ elements in $\boldsymbol{Z}[s]$ and at most $S'[t]$ elements in $\boldsymbol{Z}[t]$. Then we have

$$\Pr_{Y_1[t],\ldots,Y_q[t]}(\text{Cond. D}) \le \frac{S'[t]\cdot(S'[t]-1)}{2^{n+1}} \le \frac{S'[t]^2}{2^{n+1}} \ ,$$

since we have $\binom{S[t]}{2}$ choice of (i,i'). Therefore,

$$\begin{aligned}\Pr_{Y_1[t],\ldots,Y_q[t]}(\text{BAD}[t]) \ \le\ & \frac{(S[1]+\cdots+S[t])\cdot S[t+1]}{2^n} + \frac{S[t+1]^2}{2^{n+1}} \\ & + \frac{(S'[1]+\cdots+S'[t-1])\cdot S'[t]}{2^n} + \frac{S'[t]^2}{2^{n+1}} \ .\end{aligned}$$

Now the left hand side of (14) is upper bounded by

$$\sum_{1\le t\le m_{\mathtt{max}}} \Pr_{Y_1[t],\ldots,Y_q[t]}(\text{BAD}[t]) \tag{15}$$

since, if none of BAD$[t]$ occurs, then we do not have a collision. Finally, we have

$$\sum_{1\le t\le m_{\mathtt{max}}} \frac{2\cdot(S[1]+\cdots+S[t])\cdot S[t+1]+S[t+1]^2}{2^{n+1}} \le \frac{(S[1]+\cdots+S[m_{\mathtt{max}}])^2}{2^{n+1}}$$

and

$$\sum_{1\le t\le m_{\mathtt{max}}} \frac{2\cdot(S'[1]+\cdots+S'[t-1])\cdot S'[t]+S'[t]^2}{2^{n+1}} = \frac{(S'[1]+\cdots+S'[m_{\mathtt{max}}])^2}{2^{n+1}} \ .$$

Therefore, (15) is upper bounded by $(\sigma^2+q^2)/2^{n+1}$. Q.E.D.

Next we have the following lemma.

Lemma B.2 *Let q, $m_1,\ldots,m_q$, σ, $M_1,\ldots,M_q$ be as in Lemma B.1. Also, let $T_1,\ldots,T_q$ be arbitrarily fixed n-bit strings. Then*

$$\Pr(R_1,R_2 \stackrel{R}{\leftarrow} \mathit{Rand}(n,n) : 1\le {}^\forall i\le q, \mathsf{p9'\text{-}Tag}_{R_1,R_2}(M_i)=T_i) \ge \frac{1}{2^{qn}}\left(1-\frac{\sigma^2+q^2}{2^{n+1}}\right) \ . \tag{16}$$

Proof. The left hand side of (16) is at least

$$\Pr(R_1 \stackrel{R}{\leftarrow} \mathrm{Rand}(n,n) : 1\le {}^\forall i < {}^\forall j\le q, \mathsf{p9'\text{-}E\text{-}Tag}_{R_1}(M_i)\ne \mathsf{p9'\text{-}E\text{-}Tag}_{R_1}(M_j))\cdot\frac{1}{2^{qn}} \ ,$$

since, if there is no collision among the outputs of $\mathsf{p9'\text{-}E\text{-}Tag}_{R_1}(\cdot)$, then R_2 has q distinct inputs. From Lemma B.1, the lemma follows. Q.E.D.

We now prove Lemma 5.1.

Proof (of Lemma 5.1). Let $\mathcal{O}(\cdot)$ be either $\mathsf{p9'\text{-}Tag}_{R_1,R_2}(\cdot)$ or $g(\cdot)$. The adversary $\mathcal{A}$ has oracle access to $\mathcal{O}(\cdot)$. Since $\mathcal{A}$ is computationally unbounded, there is no loss of generality to assume that $\mathcal{A}$ is deterministic. Also, there is no loss of generality to assume that $\mathcal{A}$ makes q queries.

For the i-th query $\mathcal{A}$ makes to $\mathcal{O}(\cdot)$, define the query-answer pair (M_i,T_i), where $\mathcal{A}$'s query was M_i and the answer it got was T_i.

Suppose that we run $\mathcal{A}$ with the oracle $\mathcal{O}(\cdot)$. For this run, we define view v of $\mathcal{A}$ as

$$v \stackrel{\text{def}}{=} \langle T_1, \ldots, T_q \rangle \ . \tag{17}$$

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

- $\mathcal{A}$'s queries $(M_1, \ldots, M_q)$ are uniquely determined, and
- the final output of $\mathcal{A}$ (0 or 1) is uniquely determined.

We note that since $\mathcal{A}$ never repeats a query, $M_1, \ldots, M_q$ are distinct.

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

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

Then we have

$$p_{rand} = \sum_{V \in \boldsymbol{V}_{one}} \Pr(g \stackrel{R}{\leftarrow} \mathrm{Rand}(*, n) : 1 \le {}^{\forall} i \le q, g(M_i) = T_i) = \sum_{V \in \boldsymbol{V}_{one}} \frac{1}{2^{qn}} = \frac{N_{one}}{2^{qn}} \ . \tag{18}$$

Next let

$$p_{real} \stackrel{\text{def}}{=} \Pr(R_1, R_2 \stackrel{R}{\leftarrow} \mathrm{Rand}(n, n) : \mathcal{A}^{\mathsf{p9'\text{-}Tag}_{R_1,R_2}(\cdot)} = 1) \ .$$

Then from Lemma B.2, we have

$$\begin{aligned}
p_{real} &= \sum_{V \in \boldsymbol{V}_{one}} \Pr(R_1, R_2 \stackrel{R}{\leftarrow} \mathrm{Rand}(n, n) : 1 \le {}^{\forall} i \le q, \mathsf{p9'\text{-}Tag}_{R_1,R_2}(M_i) = T_i) \\
&\ge \sum_{V \in \boldsymbol{V}_{one}} \left(1 - \frac{\sigma^2 + q^2}{2^{n+1}}\right) \cdot \frac{1}{2^{qn}} \\
&= \frac{N_{one}}{2^{qn}} \left(1 - \frac{\sigma^2 + q^2}{2^{n+1}}\right) \ .
\end{aligned}$$

From (18) we have

$$p_{real} \ge p_{rand}\left(1 - \frac{\sigma^2 + q^2}{2^{n+1}}\right) \ge p_{rand} - \frac{\sigma^2 + q^2}{2^{n+1}} \ . \tag{19}$$

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

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

Finally, (19) and (20) give $|p_{real} - p_{rand}| \le (\sigma^2 + q^2)/2^{n+1}$. Q.E.D.

B.1 Discussion of the Previous Work [12]

[12, p. 162, Lemma 1] corresponds to our Lemma 5.1. Then one might wonder if the relevant portion can be re-used. However, in the proof of [12, p. 162, Lemma 1], there is a flaw in the analysis of Game 5. We use our notation. Let $q = 2$ in Lemma B.1. Then [12, p. 166] says

$$\Pr(R_1 \stackrel{R}{\leftarrow} \mathrm{Rand}(n,n) : \mathsf{p9'\text{-}E\text{-}Tag}_{R_1}(M_1) = \mathsf{p9'\text{-}E\text{-}Tag}_{R_1}(M_2)) = \frac{1}{2^n} \ ,$$

since $Y_1[1]$ is a random string in $\{0,1\}^n$, where $Y_1[1] = R_1(M_1[1])$. However, if $M_1[1] = M_2[1]$, then we have $Y_1[1] = Y_2[1]$, where $Y_2[1] = R_1(M_2[1])$, and their randomness disappears. This part needs to be fixed, which is done in Lemma B.1.

Also, [12, p. 166, Lemma 2] doesn't hold. There is a problem with the definition of their MAC. Their MAC, which we call $p9''[n]$, is described as follows: the key generation algorithm for $p9''[n]$ returns a randomly selected permutation P_1 from $\mathrm{Perm}(n)$. The tagging algorithm for $p9''[n]$ takes P_1 as a "key" and uses P_1 and P_2 instead of E_K and $E_{K\oplus\Delta}$, and outputs a full n-bit tag, where $P_2 \in \mathrm{Perm}(n) \setminus \{P_1\}$ is determined from P_1 by some means. The verification algorithm is defined in the natural way. Then [12, p. 159] says the security of $p9''[n]$ does not depend on how P_2 is derived from P_1, which is not correct. For example if P_2 is chosen as $P_2 = P_1^{-1}$, then it is easy to make a forgery.

We present the full security proof for $p9'[n]$ in order to avoid presenting proof covered with patches, and to establish self contained security proof.