How to Enhance the Security of the 3GPP Confidentiality and Integrity Algorithms

Tetsu Iwata and Kaoru Kurosawa

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

Abstract. We consider the 3GPP confidentiality and integrity schemes that were 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 version of $f9$, it was shown that these proofs are incorrect; it is *impossible* to prove $f8$ and $f9'$ secure under the standard PRP assumption on the underlying block cipher. Following the results, it was shown 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 $f8$. Needless to say, the assumptions here are stronger than the standard PRP assumptions, and it is natural to seek a *practical* way to modify $f8'$ and $f9'$ to establish security proofs under the standard PRP assumption. In this paper, we propose $f8^+$ and $f9^+$, slightly modified versions of $f8'$ and $f9'$, but they allow proofs of security under the standard PRP assumption. Our results are practical in the sense that we insist on the minimal modifications; $f8^+$ is obtained from $f8'$ by setting the key modifier to all-zero, and $f9^+$ is obtained from $f9'$ by setting the key modifier to all-zero, and using the encryptions of two constants in the CBC MAC computation.

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 [25]. For example, we have CTR mode [3] and CBC encryption mode [3] for symmetric encryption schemes, PMAC [9], XCBC [8] and OMAC [16] for message authentication codes, and IAPM [20], OCB mode [26], CCM mode [27, 19], EAX mode [6], CWC mode [23] and GCM mode [24] 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 [21] and that $f9'$ is a secure MAC [13], where $f9'$ is a generalized version of $f9$. However, these claims were disproven [17]. 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 [17] 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.

Generalized Versions of $f8$ and $f9$: $f8'$ and $f9'$. Given the results in [17], it is logical to ask if there are assumptions under which $f8$ and $f9$ are actually secure and, if so, what those assumptions are. 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].

The paper [14] proved that the above hypotheses are in fact correct and, in doing so, [14] clarifies what assumptions are actually necessary in order for the $f8$ and $f9$ modes to be secure. In more detail, [14] first considers a generalized version of $f8$, which is called $f8'$. $f8'$ is a nonce-based symmetric encryption scheme, and is the natural nonce-based extension of the original $f8$. Then it is shown 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.

Then [14] next considers a generalized version of $f9$, which is called $f9'$. $f9'$ 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. Then it is shown 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.

Our Contribution. Because the assumptions made for $f8'$ and $f9'$ are stronger than the standard PRP assumptions (as proven necessary in [17]), in this paper, we consider the following question; What is the minimal modification on $f8'$ and $f9'$ to achieve the provable security results with the standard PRP assumptions on the underlying block cipher? We view the answer to this question gives us an important practical result. Namely, $f8$ and $f9$ can be easily replaced with minimal cost, especially to be prepared for the worst case that KASUMI is known to be vulnerable to related-key attacks.

In this paper, we propose $f8^+$ and $f9^+$, refinements of $f8'$ and $f9'$. Unlike $f8'$ and $f9'$, our $f8^+$ and $f9^+$ are provably secure with the standard PRP assumptions. Furthermore, they require very small modifications to $f8'$ and $f9'$. In particular,

- $f8^+$ is obtained from $f8'$ by setting the key modifier to all-zero, and
- $f9^+$ is obtained from $f9'$ by setting the key modifier to all-zero, using the encryption of all-zero as the initial value of the CBC chain, and xoring the encryption of all-one before the final encryption.

These small modifications *increase* the security, allowing us to prove the security of $f8^+$ and $f9^+$ under the standard PRP assumption. Intuitively, this implies that the security of $f8^+$ and $f9^+$ is irrelevant to the resistance of KASUMI against related key attacks. $f8^+$ and $f9^+$ are provably secure if KASUMI is merely secure in the sense of a PRP.

Our results are practical in the sense that we insist on the "minimal modification," and therefore we are able to switch the modes easily. Although $f8^+$ and $f9^+$ are not competitive to CTR mode and OMAC in terms of efficiency, we find that switching to CTR mode and OMAC are costly and expensive, and it is quite unreasonable to switch to them just because to reduce the security assumption on the block cipher.

We suggest the following use of our results. (1) If there is a chance to replace $f8$ and $f9$, especially to be prepared for the worst case that KASUMI is known to be vulnerable to related-key attacks, $f8^+$ and $f9^+$ are reasonable replacements since they only require small modifications and the costs for switching should not be too expensive. These modifications may be handled by "patching" $f8$ and $f9$. (2) When the whole system is updated, the future system should support more conventional modes such as CTR mode and OMAC. In this case, the cost for replacement should not be a problem.

We prove that $f8^+$ is secure in the sense of indistinguishability from random strings, and $f9^+$ is a secure pseudorandom function, which provably implies a secure MAC, if the underlying block cipher is secure in the sense of a PRP. We note that because of the "key reuse" nature in $f8^+$ and $f9^+$, their security proofs require much elaborate treatment compared to the cases for $f8'$ and $f9'$.

Related Works. Initial security evaluation of KASUMI, $f8$ and $f9$ can be found in [12]. Knudsen and Mitchell analyzed the security of $f9'$ against forgery and key recovery attacks [22]. Blunden and Escott showed related key attacks on reduced round KASUMI [10].

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.

PRPs. The PRP notion was introduced in [25] and later made concrete in [4].

Let $\mathrm{Perm}(n)$ denote the set of all permutations on $\{0,1\}^n$, and let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a family of permutations, i.e., a block cipher. Let $\mathcal{A}$ be an adversary with access to an oracle and returns a bit. Then

$$\mathbf{Adv}_E^{\mathrm{prp}}(\mathcal{A}) \stackrel{\mathrm{def}}{=} \left| \Pr(K \stackrel{R}{\leftarrow} \{0,1\}^k : \mathcal{A}^{E_K(\cdot)} = 1) - \Pr(G \stackrel{R}{\leftarrow} \mathrm{Perm}(n) : \mathcal{A}^{G(\cdot)} = 1) \right|$$

is defined as the *PRP-advantage* of $\mathcal{A}$ on E. Intuitively, we say that E is a *secure PRP* if the PRP-advantage of all adversaries using reasonable resources is small.

We briefly remark that modern block ciphers, e.g., AES [11], are designed to be secure PRP.

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

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 $\mathrm{KM} = (01)^{64}$ (or `0x55...55` in hexadecimal) called the key modifier. In more detail, the encryption algorithm is defined in Fig. 1. In Fig. 1, $[i-1]_{64}$ 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.

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

Fig. 1. Algorithm $\mathsf{f8\text{-}Encrypt}_K(\text{COUNT}, \text{BEARER}, \text{DIRECTION}, M)$.

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.

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 $\text{KM} = (10)^{64}$ (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 $\mathsf{pad}_{64}(\text{COUNT}, \text{FRESH}, \text{DIRECTION}, M)$ 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,

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

Then the tagging algorithm is defined in Fig. 2. In Fig. 2, "$M[1] \| \cdots \| M[m] \leftarrow M$" is a shorthand for "break M into 64-bit blocks $M[1] \| \cdots \| M[m]$." The $f9$ verification algorithm is defined in the natural way by tag recomputation.

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.

A Generalized Version of $f8$*:* $f8'$ *[17, 14].* $f8'$ is a nonce-based deterministic symmetric encryption scheme, which is a generalized (and weakened) version

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

Fig. 2. Algorithm $\mathsf{f9\text{-}Tag}_K(\mathrm{COUNT}, \mathrm{FRESH}, \mathrm{DIRECTION}, M)$.

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 $f8'[E, \Delta]$ 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 is in Fig. 3. In Fig. 3, $[i-1]_n$ 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, we allow the adversary to choose these values. Consequently, $f8'$ can be considered as a weakened version of $f8$ since it gives the 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 $f8$.

A Generalized Version of $f9$: $f9'$ [13, 22, 17, 14]. 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 $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ 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 $1 \le l \le n$. 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 $M[i]$ $(1 \le i \le m-1)$ is n bits. The last block $M[m]$ may have fewer than n bits. In $f9'$, we use 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,

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

Thus, we simply ignore COUNT, FRESH and DIRECTION. Equivalently, we consider them as a part of the message. The rest of the tagging algorithm is the same as $f9$. The pseudocode is given in Fig. 5. In Fig. 5, "$M[1] \| \cdots \| M[m] \leftarrow M$" is a shorthand for "break M into n-bit blocks $M[1] \| \cdots \| M[m]$."

Note that the 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 Proposed Schemes: Specifications of $f8^+$ and $f9^+$

4.1 Proposed Refinement of $f8'$: $f8^+$

$f8^+$ is a nonce-based deterministic symmetric encryption scheme, which is a refinement of $f8'$. Definening $f8^+$ is simple: we set $\Delta \leftarrow 0^n$ in $f8'$.

For full specification, $f8^+$ uses a block cipher $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ as the underlying primitive. Let $f8^+[E]$ be $f8^+$, where E is used as the underlying primitive. The $f8^+[E]$ key generation algorithm returns a random k-bit key K. The $f8^+[E]$ encryption algorithm, which we call $\mathsf{f8^+\text{-}Encrypt}$, takes an n-bit nonce N. 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 is given in Fig. 4.

```
Algorithm f8'-Encrypt_K(N, M)
    m ← ⌈|M|/n⌉
    Y[0] ← 0^n
    A ← N
    A ← E_{K⊕Δ}(A)
    For i ← 1 to m do:
        X[i] ← A ⊕ [i − 1]_n ⊕ Y[i − 1]
        Y[i] ← E_K(X[i])
    C ← M ⊕ (the leftmost |M| bits
                  of Y[1]‖ ··· ‖Y[m])
    Return C
```

Fig. 3. Algorithm $\mathsf{f8'\text{-}Encrypt}_K(N, M)$.

```
Algorithm f8+-Encrypt_K(N, M)
    m ← ⌈|M|/n⌉
    Y[0] ← 0^n
    A ← N
    A ← E_K(A)
    For i ← 1 to m do:
        X[i] ← A ⊕ [i − 1]_n ⊕ Y[i − 1]
        Y[i] ← E_K(X[i])
    C ← M ⊕ (the leftmost |M| bits
                  of Y[1]‖ ··· ‖Y[m])
    Return C
```

Fig. 4. Algorithm $\mathsf{f8^+\text{-}Encrypt}_K(N, M)$.

Decryption is done in an obvious way.

Note that the only difference between $\mathsf{f8^+\text{-}Encrypt}_K(\cdot,\cdot)$ and $\mathsf{f8'\text{-}Encrypt}_K(\cdot,\cdot)$ is in the 4-th line. With this small modification, as we will show shortly, $f8^+$ has much higher security assurance than that of $f8'$.

4.2 Proposed Refinement of $f9'$: $f9^+$

$f9^+$ is a message authentication code, which is a refinement of $f9'$. As with $f8^+$, definening $f9^+$ is simple:

– We set $\Delta \leftarrow 0^n$ in $f9'$,
– We set the initial value of the CBC chain to be $E_K(0^n)$ instead of 0^n, and
– We xor $E_K(1^n)$ before the last encryption.

For full specification, $f9^+$ uses a block cipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ as the underlying primitive. Let $f9^+[E, l]$ be $f9^+$, where E is used as the underlying block cipher, and the tag length is l, where $1 \le l \le n$. The key generation algorithm returns a random k-bit key K. The tagging algorithm, which we call $\mathsf{f9}^+\text{-}\mathsf{Tag}$, takes a k-bit key K and a message $M \in \{0,1\}^*$ as input and returns an l-bit tag T. $f9^+$ uses $\mathsf{pad}'_n(\cdot)$ defined in (1). The pseudocode is given in Fig. 6. In Fig. 6, "$M[1]\|\cdots\|M[m] \leftarrow M$" is a shorthand for "break M into n-bit blocks $M[1]\|\cdots\|M[m]$."

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

Fig. 5. Algorithm $\mathsf{f9}'\text{-}\mathsf{Tag}_K(M)$.

Algorithm $\mathsf{f9}^+\text{-}\mathsf{Tag}_K(M)$
- $M \leftarrow \mathsf{pad}'_n(M)$
- $M[1]\|\cdots\|M[m] \leftarrow M$
- $Y[0] \leftarrow E_K(0^n)$
- For $i \leftarrow 1$ to m do:
  - $X[i] \leftarrow M[i] \oplus Y[i-1]$
  - $Y[i] \leftarrow E_K(X[i])$
- $T \leftarrow E_K(Y[1] \oplus \cdots \oplus Y[m] \oplus E_K(1^n))$
- $T \leftarrow$ the leftmost l bits of T
- Return T

Fig. 6. Algorithm $\mathsf{f9}^+\text{-}\mathsf{Tag}_K(M)$.

The verification algorithm is defined in the natural way.

Notice that the only difference between $\mathsf{f9}^+\text{-}\mathsf{Tag}_K(\cdot)$ and $\mathsf{f9}'\text{-}\mathsf{Tag}_K(\cdot)$ is in the 3-rd and 7-th lines. With these small modifications, as we will show shortly, $f9^+$ has much higher security assurance than that of $f9'$.

4.3 Design Rational on $f9^+$

One might try to set $\Delta \leftarrow 0^n$ in $f9'$ (which eliminates the needs of related keys), and preserve the rest of $f9'$. Unlike the case for $f8^+$, this does not work. In fact, the above mentioned MAC is easily forgeable. The attack proceeds as follows.

1. The adversary $\mathcal{A}$ first queries a message M_1 such that $1 \le |M_1| < n$, to obtain the tag $T_1 = E_K(E_K(\mathsf{pad}'_n(M_1)))$.
2. Then $\mathcal{A}$ queries a message $M_2 \leftarrow \mathsf{pad}'_n(M_1)\|0^n\|M'_2$, where M'_2 is a string such that $|M'_2| < n$ and $\mathsf{pad}'_n(M'_2) = T_1 \oplus \mathsf{pad}'_n(M_1)$, to obtain the tag T_2. By a simple calculation, one can verify that $T_2 = E_K(T_1)$.
3. Next $\mathcal{A}$ queries a message $M_3 \leftarrow T_1\|M'_3$, where M'_3 is a string such that $|M'_3| < n$ and $\mathsf{pad}'_n(M'_3) = T_1 \oplus T_2$, to obtain the tag T_3. By a simple calculation, one can verify that $T_3 = E_K(0^n)$.

4. Finally, $\mathcal{A}$ outputs a forgery attempt (M^*, T^*), where $M^* \leftarrow 0^n \| T_3'$, T_3' is a string such that $|T_3'| < n$ and $\mathsf{pad}_n'(T_3') = T_3$, and $T^* \leftarrow T_3$.

There are several cases where this attack fails. These cases are $T_1 \oplus \mathsf{pad}_n'(M_1) = 0^n$ (since M_2' does not exist), $T_1 \oplus T_2 = 0^n$ (since M_3' does not exist), and $M^* = M_3$ (since M^* should be a new massage). Assuming that the underlying block cipher is a random permutation, these cases occur with only negligible probabilities. Therefore, the above attack succeeds in forgery with overwhelming probability even if the underlying block cipher is a random permutation [2].

Therefore, merely setting $\Delta \leftarrow 0^n$ does not work. This motivates us to "mask" the input to the first block cipher invocation, as well as the final invocation. We used $Y[0] \leftarrow E_K(0^n)$ and $E_K(1^n)$ as masks, however, any other constant is fine.

5 Security of $f8^+$

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

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, a 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 [26]. 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.

[2] We note that [13, p. 157, Section 2.2] shows similar attack. But the attack in [13] cannot be applied here since padding is not considered.

Provable Security Results. Let $p8^+[n]$ be a variant of $f8^+$ that uses a random function on n bits instead of E_K. Specifically, the key generation algorithm for $p8^+[n]$ returns a randomly selected function R from $\mathrm{Rand}(n, n)$. The encryption algorithm for $p8^+[n]$, $\mathsf{p8^+\text{-}Encrypt}$, takes R as a "key" and uses it instead of E_K. 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 $\sigma = \sum_{1 \le i \le q} \lceil |M_i|/n \rceil$.

Lemma 5.1. *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*

$$\mathbf{Adv}^{\mathrm{priv}}_{p8^+[n]}(\mathcal{A}) \le \frac{2\sigma^2}{2^n} \quad . \tag{2}$$

A proof sketch is given in Appendix A, and a proof is given in the full version of this paper [18].

We now present our main result for $f8^+$ (Theorem 5.1 below). At a high level, our theorem shows that if a block cipher E is a secure PRP, then the construction $f8^+[E]$ 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 $f8^+[E]$ and making at most q oracle queries totaling at most σ blocks, we can construct a PRP adversary $\mathcal{B}$ attacking E such that $\mathcal{B}$ uses similar resources as $\mathcal{A}$ and $\mathcal{B}$ has advantage $\mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) \ge \mathbf{Adv}^{\mathrm{priv}}_{f8^+[E]}(\mathcal{A}) - 4\sigma^2/2^n$. If we assume that E is a secure PRP and that $\mathcal{A}$ (and therefore $\mathcal{B}$) uses reasonable resources, then $\mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B})$ must be small by definition, and thus $\mathbf{Adv}^{\mathrm{priv}}_{f8^+[E]}(\mathcal{A})$ must also be small. This means that under the assumptions on E being a secure PRP, $f8^+[E]$ is provably secure.

Since many block ciphers, including AES and KASUMI, are believed to be a secure PRP, this theorem means that $f8^+$ constructions built from these block ciphers will be provably secure.

Our main theorem statement for $f8^+$ is given below.

Theorem 5.1 (Main Theorem for $f8^+$). *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. Let $f8^+[E]$ be as described in Section 4.1. 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 PRP adversary $\mathcal{B}$ against E such that*

$$\mathbf{Adv}^{\mathrm{priv}}_{f8^+[E]}(\mathcal{A}) \le \frac{4\sigma^2}{2^n} + \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) \quad . \tag{3}$$

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

A proof is done by applying a well known PRF/PRP switching lemma (see [4, Proposition 2.5]), and we add $(q+\sigma)^2/2^{n+1} \le 2\sigma^2/2^n$ to the bound in Lemma 5.1, and the rest of the proof of Theorem 5.1 is completely standard.

Notice the difference between Theorem 5.1 and the result for $f8'$ in [14, p. 435, Theorem 4.1]. $\mathbf{Adv}^{\mathrm{priv}}_{f8'[E,\Delta]}(\mathcal{A})$ is upper bounded by $(3\sigma^2+q^2)/2^{n+1}+\mathbf{Adv}^{\mathrm{prp\text{-}rka}}_{\Phi,E}(\mathcal{B})$. Intuitively, $f8'$ preserves privacy under chosen-plaintext attacks if the pair $(E_K(\cdot), E_{K\oplus\Delta}(\cdot))$ and a pair of two independent random permutations are indistinguishable, while $f8^+$ achieves the same security goal if $E_K(\cdot)$ is a secure PRP.

6 Security of $f9^+$

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

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 a random function on n bits instead of E_K. Specifically, the key generation algorithm for $p9^+[n]$ returns a randomly selected functions R from $\mathrm{Rand}(n,n)$. The tagging algorithm for $p9^+[n]$, $\mathsf{p9^+\text{-}Tag}$, takes R as a "key" and uses it instead of E_K. 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 $\sigma = \sum_{1\le i\le q} |\mathsf{pad}'_n(M_i)|/n$.

Lemma 6.1. *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*

$$\mathbf{Adv}^{\mathrm{prf}}_{p9^+[n]}(\mathcal{A}) \le \frac{5\sigma^2}{2^n} \ . \tag{4}$$

A proof sketch is given in Appendix B, and a proof is given in the full version of this paper [18].

We now present our main result for $f9^+$ (Theorem 6.1), which we interpret as follows: our theorem shows that if a block cipher E is a secure PRP, then the construction $f9^+[E,l]$ based on E will be a provably secure message authentication code. In more detail, we show that given any adversary $\mathcal{A}$ attacking $f9^+[E,l]$ and making at most q oracle queries totaling at most σ blocks, we can construct a PRP adversary $\mathcal{B}$ against E such that $\mathcal{B}$ uses similar resources as $\mathcal{A}$ and $\mathcal{B}$ has advantage $\mathbf{Adv}_E^{\mathrm{prp}}(\mathcal{B}) \geq \mathbf{Adv}_{f9^+[E,l]}^{\mathrm{prf}}(\mathcal{A}) - 10\sigma^2/2^n$. If we assume that E is a secure PRP and that $\mathcal{A}$ (and therefore $\mathcal{B}$) uses reasonable resources, then $\mathbf{Adv}_E^{\mathrm{prp}}(\mathcal{B})$ must be small by definition. Therefore $\mathbf{Adv}_{f9^+[E,l]}^{\mathrm{prf}}(\mathcal{A})$ must be small as well, proving that under the assumption on E being a secure PRP, $f9^+[E,l]$ is provably secure.

Since many block ciphers, including AES and KASUMI, are believed to be a secure PRP, this theorem means that $f9^+$ constructions built from these block ciphers will be provably secure.

The precise theorem statement is as follows:

Theorem 6.1 (Main Theorem for $f9^+$). *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher, and let l, $1 \leq l \leq n$, be a constant. Let $f9^+[E,l]$ be as described in Section 4.2. If $\mathcal{A}$ is a PRF adversary which asks at most q queries totaling at most σ blocks, then we can construct a PRP adversary $\mathcal{B}$ against E such that*

$$\mathbf{Adv}_{f9^+[E,l]}^{\mathrm{prf}}(\mathcal{A}) \leq \frac{10\sigma^2}{2^n} + \mathbf{Adv}_E^{\mathrm{prp}}(\mathcal{B}) \ . \tag{5}$$

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

As is the case in Theorem 5.1, a proof is done by applying the PRF/PRP switching lemma, and $(q+\sigma+2)^2/2^{n+1} \leq 9\sigma^2/2^{n+1}$ is added to the bound in Lemma 6.1, and the rest of the proof is standard.

Notice the difference between Theorem 6.1 and the result for $f9'$ in [14, p. 438, Theorem 5.1]. $\mathbf{Adv}_{f9'[E,\Delta,l]}^{\mathrm{prf}}(\mathcal{A})$ is upper bounded by $(3q^2 + 2\sigma^2 + 2\sigma q)/2^{n+1} + \mathbf{Adv}_{\Phi,E}^{\mathrm{prp\text{-}rka}}(\mathcal{B})$. Intuitively, $f9'$ is pseudorandom if the pair $(E_K(\cdot), E_{K\oplus\Delta}(\cdot))$ and a pair of two independent random permutations are indistinguishable, while $f9^+$ is pseudorandom if $E_K(\cdot)$ is a secure PRP.

7 Conclusion

In this paper, we proposed $f8^+$ and $f9^+$, which are refinements of the original $f8'$ and $f9'$. $f8^+$ and $f9^+$ are designed with two goals; (1) minimal modifications to $f8'$ and $f9'$, and (2) provable security results with the standard PRP assumption on the underlying block cipher. Since we make only "small" modifications, these modes can be practical candidates for future replacement of $f8$ and $f9$. Especially, we believe that $f8^+$ is simple enough to be replaced easily.

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,* pages 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, pages 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 B. Roy and W. Meier, editors, *Fast Software Encryption, FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 389–407. Springer-Verlag, Berlin Germany, 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. CBC MACs for arbitrary-length messages: The three key constructions. In M. Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 197–215. Springer-Verlag, Berlin Germany, 2000.
9. 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.
10. 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.
11. J. Daemen and V. Rijmen. *The Design of Rijndael*. Springer-Verlag, Berlin Germany, 2002.
12. 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`.
13. 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.
14. T. Iwata and T. Kohno. New security proofs for the 3GPP confidentiality and integrity algorithms. In B. Roy and W. Meier, editors, *Fast Software Encryption, FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 427–445. Springer-Verlag, Berlin Germany, 2004.

15. T. Iwata and T. Kohno. New security proofs for the 3GPP confidentiality and integrity algorithms. Full version of [14], available at IACR Cryptology ePrint Archive, Report 2004/019, `http://eprint.iacr.org/`, 2004.
16. 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.
17. 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.
18. T. Iwata and K. Kurosawa. How to enhance the security of the 3GPP confidentiality and integrity algorithms. Full version of this paper, available from the authors, 2005.
19. 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.
20. 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.
21. 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.
22. L.R. Knudsen and C.J. Mitchell. Analysis of 3gpp-MAC and two-key 3gpp-MAC. *Discrete Applied Mathematics*, vol. 128, no. 1, pages 181–191, 2003.
23. T. Kohno, J. Viega, and D. Whiting. CWC: A high-performance conventional authenticated encryption mode. In B. Roy and W. Meier, editors, *Fast Software Encryption, FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 408–426. Springer-Verlag, Berlin Germany, 2004.
24. D.A. McGrew, and J. Viega. The security and performance of the Galois/Counter Mode of operation. In IACR Cryptology ePrint Archive, Report 2004/193, `http://eprint.iacr.org/`, 2004.
25. M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.,* vol. 17, no. 2, pages 373–386, April 1988.
26. 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.
27. 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 Sketch of Lemma 5.1

We sketch the proof of Lemma 5.1 here, leaving the details to [18]. The adversary has an oracle which is either $\mathsf{p8}^+\text{-}\mathsf{Encrypt}_R(\cdot,\cdot)$ or $\$(\cdot,\cdot)$. 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 $N_1,\ldots,N_q$ 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:

For any i $(1 \le i \le q)$, the multiset

$$\{0^n, M_i[1] \oplus C_i[1] \oplus [1]_n, \ldots, M_i[m_i - 1] \oplus C_i[m_i - 1] \oplus [m_i - 1]_n\} \quad (6)$$

has m_i distinct points

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

For (N_i, M_i) and the function R, let $A_i = R(N_i)$, 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 $Y_i[j] = R(X_i[j])$. 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 $\boldsymbol{N} \stackrel{\text{def}}{=} \{N_i \mid 1 \le i \le q\}$.

Then for randomly chosen A_t (this will fix $\boldsymbol{X}_t$), define the following $(t-1)+1 = t$ conditions: Cond. A-s $(1 \le s \le t-1)$ and Cond. B.

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

We say that BAD$[t]$ occurs if at least one of the above t events occurs.

Intuitively, we show that if all the query-answer pairs satisfy (6) and BAD$[t]$ does not occur, then the adversary cannot distinguish between $\mathsf{p8}^+\text{-}\mathsf{Encrypt}_R(\cdot,\cdot)$ and $\$(\cdot,\cdot)$. The proof is completed by upper bounding the probability that some query-answer pair fails to satisfy (6), or some BAD$[t]$ occurs.

B Proof Sketch of Lemma 6.1

To prove Lemma 6.1, 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. That is, we consider that messages have already padded. Also, it does not perform the final encryption and it does not mask with $R(1^n)$. Specifically, the key generation algorithm for $p9^+$-E$[n]$ returns a randomly selected function R from Rand(n, n). The tagging algorithm for $p9^+$-E$[n]$, $\mathsf{p9}^+\text{-}\mathsf{E}\text{-}\mathsf{Tag}$, takes R as a "key" and a message M such that $|M| = mn$ for some $m \ge 1$. The pseudocode is given in Fig. 7. "$M[1] \| \cdots \| M[m] \leftarrow M$" is a shorthand for "break M into n-bit blocks $M[1] \| \cdots \| M[m]$." The verification algorithm is defined in the natural way.

We next fix some notation. For q and σ in Lemma 6.1, let $m_1, \ldots, m_q$ be integers such that $m_i \ge 1$ and $\sigma \ge m_1 + \cdots + m_q$. Let $M_1, \ldots, M_q$ be fixed and distinct bit strings such that $|M_i| = m_i n$.

Then we have the following lemma.

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

- $1 \le {}^\exists i < {}^\exists j \le q, \mathsf{p9}^+\text{-}\mathsf{E}\text{-}\mathsf{Tag}_R(M_i) = \mathsf{p9}^+\text{-}\mathsf{E}\text{-}\mathsf{Tag}_R(M_j)$*, or*
- $1 \le {}^\exists i \le q$*, $R(1^n)$ is used in the computation of* $\mathsf{p9}^+\text{-}\mathsf{E}\text{-}\mathsf{Tag}_R(M_i)$

Algorithm $\mathsf{p9}^{+}\text{-}\mathsf{E}\text{-}\mathsf{Tag}_R(M)$

$$
\begin{array}{l}
M[1]\|\cdots\|M[m] \leftarrow M \\
Y[0] \leftarrow R(0^n) \\
\text{For } i \leftarrow 1 \text{ to } m \text{ do:} \\
\quad X[i] \leftarrow M[i] \oplus Y[i-1] \\
\quad Y[i] \leftarrow R(X[i]) \\
\text{Return } Y[1] \oplus \cdots \oplus Y[m]
\end{array}
$$

Fig. 7. Algorithm $\mathsf{p9}^{+}\text{-}\mathsf{E}\text{-}\mathsf{Tag}_R(M)$.

is at most $3\sigma^2/2^n$ *where the probability is taken over the random choice of* $R \stackrel{R}{\leftarrow} Rand(n,n)$.

Given the above Lemma B.1, it is easy to prove the following lemma.

Lemma B.2. *Let* q *and* σ *be as in Lemma 6.1. Also, let* $M_1, \ldots, M_q$ *be arbitrarily fixed and distinct bit strings, and let* $T_1, \ldots, T_q$ *be arbitrarily fixed* n*-bit strings. Then*

$$\Pr(R \stackrel{R}{\leftarrow} Rand(n,n) : 1 \le {}^{\forall} i \le q, \mathsf{p9}^{+}\text{-}\mathsf{Tag}_R(M_i) = T_i) \ge \frac{1}{2^{qn}}\left(1 - \frac{5\sigma^2}{2^n}\right) .$$

Given the above lemma, the proof of Lemma 6.1 is standard.