Electronic Preprint for Journal of Information Processing Vol.20 No.4

Regular Paper

# Length-preserving CBC Enciphering Scheme and Its Security Analysis

Hidenori Kuwakado[1,a)]

Received: November 28, 2011, Accepted: June 1, 2012

**Abstract:** We propose a length-preserving enciphering scheme that achieves PRP security and streamable decryption. No enciphering scheme satisfying these properties is known. Our enciphering scheme is suitable for secure communication on narrowband channels and memory-constrained devices. Although length-preserving enciphering schemes satisfying the SPRP security, which is stronger than the PRP security, are known, it is impossible to support the SPRP security and the streamability at the same time. Namely, the memory to store an entire plaintext/ciphertext is required. When the decryption is performed with memory-constrained devices, the PRP security is the strongest concept of achievable security.

**Keywords:** blockcipher, mode of operation, length-preserving, pseudorandom permutation

## 1. Introduction

### 1.1 Background

Due to development of sensor devices and small wireless devices, technologies for achieving secure communication on resource-constrained networks are much in demand these days. Lightweight cryptography is an important primitive for achieving secure communication on resource-constrained networks. In particular, lightweight blockciphers/hash functions have been studied actively [1], [4], [5], [8], [9], [13], [18]. Since a blockcipher is a permutation on small domain, a mode of operation is required to encrypt large data. Use of the mode of operation often causes length-expansion of data. For example, the CBC mode requires an initialization vector and the CTR mode requires a counter. In general, the length-expansion is not desirable for narrowband channels. There are cases where length-preservation is a requirement due to technical or economic constrains.

The strongest notion of security for a length-preserving encryption scheme is strong pseudorandom permutation (SPRP) and tweakable SPRP. Motivated by the application to disk encryption, constructions of tweakable SPRP have been proposed. We took notice of the fact that the SPRP constructions are not streamable. Namely, no part of a ciphertext is obtained before reading through a plaintext, and no part of a plaintext is obtained before reading through a ciphertext. This is undesirable for memory-constrained devices because memory to store entire data is required. This also implies that the SPRP constructions are not suitable for real-time communication. In such cases, one may want to use an encryption scheme that achieves a stronger notion of security (i.e., PRP) than the strongest notion of security (i.e., SPRP). If the PRP security is sufficient for applications, then it may be possible to decrypt a ciphertext streamably. To the best of our knowledge, there is no mode of operation satisfying (1) length-preserving, (2) streamable decryption, and (3) PRP security.

### 1.2 Our Contribution

This paper proposes a mode of operation satisfying the three properties above (called LPCBC). LPCBC uses a blockcipher and a pseudorandom function as underlying primitives. Roughly speaking, LPCBC is a reversing CBC mode such that an initialization vector is replaced with the output of the pseudorandom function (**Fig. 1**). LPCBC does not require that the length of a plaintext be a multiple of the block size. In order to achieve streamable decryption, the pseudorandom function is required to be streamable. For example, HMAC is a streamable pseudorandom function. Under the assumption that a blockcipher is a PRP for independent two keys, we analyze the PRP security of LPCBC.

### 1.3 Related Works

The ECB mode is a length-preserving and streamable enciphering scheme. It is well-known that the ECB mode has drawbacks on security. Other modes of operation (e.g., the CBC mode, the CTR mode) are not length-preserving.

Length-preserving enciphering scheme has been studied for disk encryption. Disk encryption may be somewhat different from the encryption for secure communication. Disk encryption handles only data such that the size is a multiple of the block size of an underlying blockcipher. Disk encryption is required to be tweakable, but is not required to be streamable. Hence, disk encryption is usually a tweakable SPRP. Constructions of tweakable SPRP can be classified into three types. The first type is the ECB mode between two invertible universal hash functions (or $\epsilon$-almost XOR universal functions). The Naor-Reingold mode [15], TET [11], and HEH [16] are of this type. The second type is the

[1] Kobe University, Kobe, Hyogo 657–8501, Japan
[a)] kuwakado@kobe-u.ac.jp

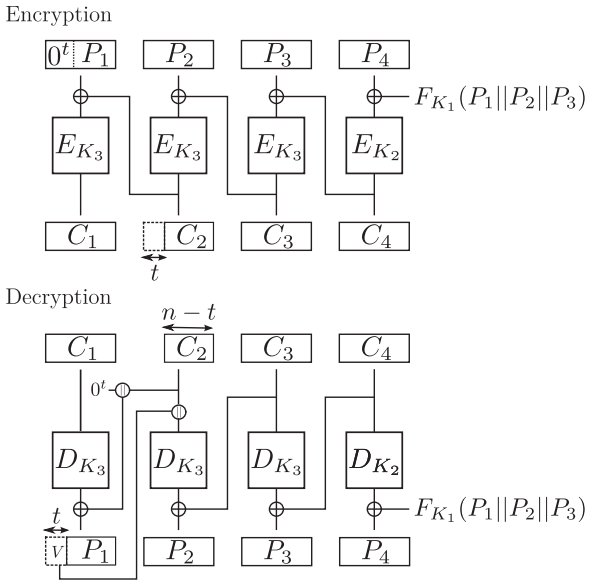


**Fig. 1** Length-preserving CBC mode ($m = 4$).

CTR mode between universal hash functions. HCTR [19] and HCH [6] are of this types. The third type consists of two layers of encryption. CMC [12] and EME* [10] are of this types. Unlike other two types, the third type does not require a universal hash function. Universal hash functions (or $\epsilon$-almost XOR universal functions) are often used for constructing a SPRP, and their algorithms are not complicated. However, we only know a few applications in which the universal hash function is implemented. On the other hand, cryptographic hash functions such as SHA are widely implemented in applications. This situation encourages us to use cryptographic hash functions instead of universal hash functions.

Bellare and Rogaway [3] have proposed a length-preserving encryption based on the CBC mode and the CBC-MAC. Actually, our scheme is somewhat analogous to their scheme. Their scheme and our scheme differ in the following respects.

(1) Their scheme computes the CBC-MAC of an entire plaintext. Our scheme computes the MAC of a part of a plaintext.
(2) To the best of our knowledge, the security proof of their scheme is not given. Precisely speaking, their scheme is not a SPRP, but it is unknown whether their scheme is a PRP or not. The security proof of our scheme is given in this paper.
(3) Their scheme only handles a plaintext such that the length is a multiple of the block size. Our scheme does not have such a limitation.

Minematsu and Tsunoo [14] have proposed a hybrid symmetric encryption. In their article, they showed a hybrid large block PRP, which consists of a PRP, a weak PRF, and an $\epsilon$-almost XOR universal function. The hybrid large block PRP has a structure similar to the Feistel structure. The hybrid large block PRP is not streamable since the first block of a plaintext cannot be obtained only from the first block of a ciphertext.

## 2. Definition

Let $\mathcal{E} : \mathcal{K}_\mathcal{E} \times \mathcal{D}_\mathcal{E} \to \mathcal{D}_\mathcal{E}$ be a keyed permutation. We define the advantage of $A$ in distinguishing $\mathcal{E}$ from a *random permutation* $\pi$ on $\mathcal{D}_\mathcal{E}$ as follows:

$$\mathbf{Adv}^{\mathrm{prp}}_{\mathcal{E}}(A) = \Pr\left[K \xleftarrow{\$} \mathcal{K}_\mathcal{E} : A^{\mathcal{E}_K} \Rightarrow 1\right] - \Pr\left[\pi \xleftarrow{\$} \mathrm{Perm}(\mathcal{D}_\mathcal{E}) : A^{\pi} \Rightarrow 1\right], \quad (1)$$

where $\mathrm{Perm}(\mathcal{D}_\mathcal{E})$ is the set of all permutations on $\mathcal{D}_\mathcal{E}$. If $\mathbf{Adv}^{\mathrm{prp}}_{\mathcal{E}}(A)$ is small for any reasonable $A$, then $\mathcal{E}$ is called a *pseudorandom permutation*. We also define the advantage of $A$ in distinguishing $\mathcal{E}$ with two keys from two random permutations on $\mathcal{D}_\mathcal{E}$ as follows:

$$\mathbf{Adv}^{\mathrm{tkprp}}_{\mathcal{E}}(A) = \Pr\left[K_1 \xleftarrow{\$} \mathcal{K}_\mathcal{E}, K_2 \xleftarrow{\$} \mathcal{K}_\mathcal{E} : A^{\mathcal{E}_{K_1},\mathcal{E}_{K_2}} \Rightarrow 1\right] - \Pr\left[\pi_1 \xleftarrow{\$} \mathrm{Perm}(\mathcal{D}_\mathcal{E}), \pi_2 \xleftarrow{\$} \mathrm{Perm}(\mathcal{D}_\mathcal{E}) : A^{\pi_1,\pi_2} \Rightarrow 1\right].$$

Let $\mathcal{F} : \mathcal{K}_\mathcal{F} \times \mathcal{D}_\mathcal{F} \to \mathcal{R}_\mathcal{F}$ be a keyed function. We define the advantage of $A$ in distinguishing $\mathcal{F}$ from a *random function* $\rho$ as follows:

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathcal{F}}(A) = \Pr\left[K \xleftarrow{\$} \mathcal{K}_\mathcal{F} : A^{\mathcal{F}_K} \Rightarrow 1\right] - \Pr\left[\rho \xleftarrow{\$} \mathrm{Func}(\mathcal{D}_\mathcal{F}, \mathcal{R}_\mathcal{F}) : A^{\rho} \Rightarrow 1\right]$$

where $\mathrm{Func}(\mathcal{D}_\mathcal{F}, \mathcal{R}_\mathcal{F})$ is the set of all functions from $\mathcal{D}_\mathcal{F}$ to $\mathcal{R}_\mathcal{F}$. If $\mathbf{Adv}^{\mathrm{prf}}_{\mathcal{F}}(A)$ is small for any reasonable $A$, then $\mathcal{F}$ is called a *pseudorandom function*.

Suppose that a permutation $\mathcal{E}$ uses a function $\lambda$ and two permutations $\mu, \nu$ as subroutines, denoted by $\mathcal{E}^{\lambda,\mu,\nu}$. Theorem 1 described below implies that it is sufficient to analyze the security of $\mathcal{E}^{\rho,\phi,\omega}$ using a random function $\rho$ and two random permutations $\phi, \omega$ instead of the security of $\mathcal{E}^{F_{K_1},E_{K_2},E_{K_3}}$ using a pseudorandom function $F$ and a blockcipher $E$.

**Theorem 1** Suppose that $\lambda$ is a function from $\mathcal{R}_\lambda$ to $\mathcal{D}_\lambda$ and $\mu, \nu$ are permutations on $\mathcal{D}$. Let $\rho$ be a random function chosen from $\mathrm{Func}(\mathcal{R}_\lambda, \mathcal{D}_\lambda)$ and let $\phi, \omega$ be independently random permutations chosen from $\mathrm{Perm}(\mathcal{D})$. Let $F$ be a pseudorandom function $\mathcal{K}_F \times \mathcal{R}_\lambda \to \mathcal{D}_\lambda$, and let $E$ a blockcipher $\mathcal{K}_E \times \mathcal{D} \to \mathcal{D}$. Suppose that $K_1$ is chosen from $\mathcal{K}_F$ at random and $K_2, K_3$ are independently chosen from $\mathcal{K}_E$ at random. Let $A$ be an adversary to $\mathcal{E}^{\lambda,\mu,\nu}$. Then, there exist adversaries $B, C$ satisfying

$$\mathbf{Adv}^{\mathrm{prp}}_{\mathcal{E}^{F_{K_1},E_{K_2},E_{K_3}}}(A) \le \mathbf{Adv}^{\mathrm{prp}}_{\mathcal{E}^{\rho,\phi,\omega}}(A) + 2\mathbf{Adv}^{\mathrm{prf}}_{F}(B) + 2\mathbf{Adv}^{\mathrm{tkprp}}_{E}(C), \quad (2)$$

where the extra computational resource of $B$ (or $C$) is bounded by some small constant multiplied by the number of oracle queries made by $B$ (or $C$).

*Proof.* We first construct an adversary $B$ attacking the security of $F$ as follows. Let $O_B$ be $B$'s oracle that is either $F_{K_1}$ or $\rho$. After choosing random permutations $\phi, \omega$, $B$ simulates $\mathcal{E}^{O_B,\phi,\omega}$ and runs $A$ as a subroutine. $B$ finally outputs the value that $A$ outputs. The number of queries made by $B$ is equal to that made by $A$. The extra work which $B$ does over $A$ is to perform the algorithm of $\mathcal{E}$. The probability that $B$ outputs 1 is given by

$$\Pr\left[B^{O_B} \Rightarrow 1 | O_B = F_{K_1}\right] = \Pr\left[A^{\mathcal{E}^{F_{K_1},\phi,\omega}} \Rightarrow 1\right],$$
$$\Pr\left[B^{O_B} \Rightarrow 1 | O_B = \rho\right] = \Pr\left[A^{\mathcal{E}^{\rho,\phi,\omega}} \Rightarrow 1\right].$$

Using the equations above, we write the advantage of $B$ as

$$\begin{aligned}\mathbf{Adv}^{\mathrm{prf}}_{F}(B) &= \Pr\left[B^{F_{K_1}} \Rightarrow 1\right] - \Pr\left[B^{\rho} \Rightarrow 1\right] \\ &= \Pr\left[B^{O_B} \Rightarrow 1 | O_B = F_{K_1}\right]\Pr\left[O_B = F_{K_1}\right] \\ &\quad -\Pr\left[B^{O_B} \Rightarrow 1 | O_B = \rho\right]\Pr\left[O_B = \rho\right] \\ &= \frac{1}{2}\left(\Pr\left[B^{O_B} \Rightarrow 1 | O_B = F_{K_1}\right] - \Pr\left[B^{O_B} \Rightarrow 1 | O_B = \rho\right]\right) \\ &= \frac{1}{2}\left(\Pr\left[A^{\mathcal{E}^{F_{K_1},\phi,\omega}} \Rightarrow 1\right] - \Pr\left[A^{\mathcal{E}^{\rho,\phi,\omega}} \Rightarrow 1\right]\right). \qquad (3)\end{aligned}$$

We next construct an adversary $C$ attacking the security of $E$ as follows. Let $O_C$ be $C$'s oracle which is either $(E_{K_2}, E_{K_3})$ or $(\phi, \omega)$. After choosing a key $K_1$ from $\mathcal{K}_F$ at random, $C$ simulates $\mathcal{E}^{F_{K_1},O_C}$ and runs $A$ as a subroutine. $C$ finally outputs the value that $A$ outputs. The number of queries made by $C$ is equal to that made by $A$. The extra work which $C$ does over $A$ is to perform $\mathcal{E}$ and $F$. The probability that $C$ outputs 1 is given by

$$\Pr\left[C^{O_C} \Rightarrow 1 | O_C = (E_{K_2}, E_{K_3})\right] = \Pr\left[A^{\mathcal{E}^{F_{K_1},E_{K_2},E_{K_3}}} \Rightarrow 1\right],$$
$$\Pr\left[C^{O_C} \Rightarrow 1 | O_C = (\phi, \omega)\right] = \Pr\left[A^{\mathcal{E}^{F_{K_1},\phi,\omega}} \Rightarrow 1\right].$$

Using the equations above, we write the advantage of $C$ as

$$\begin{aligned}\mathbf{Adv}^{\mathrm{tkprp}}_{E}(C) &= \Pr\left[C^{E_{K_2},E_{K_3}} \Rightarrow 1\right] - \Pr\left[C^{\phi,\omega} \Rightarrow 1\right] \\ &= \Pr\left[C^{O_C} \Rightarrow 1 | O_C = (E_{K_2}, E_{K_3})\right]\Pr\left[O_C = (E_{K_2}, E_{K_3})\right] \\ &\quad -\Pr\left[C^{O_C} \Rightarrow 1 | O_C = (\phi, \omega)\right]\Pr\left[O_C = (\phi, \omega)\right] \\ &= \frac{1}{2}\left(\Pr\left[C^{O_C} \Rightarrow 1 | O_C = (E_{K_2}, E_{K_3})\right]\right. \\ &\quad \left.-\Pr\left[C^{O_C} \Rightarrow 1 | O_C = (\phi, \omega)\right]\right) \\ &= \frac{1}{2}\left(\Pr\left[A^{\mathcal{E}^{F_{K_1},E_{K_2},E_{K_3}}} \Rightarrow 1\right] - \Pr\left[A^{\mathcal{E}^{F_{K_1},\phi,\omega}} \Rightarrow 1\right]\right). \qquad (4)\end{aligned}$$

Adding Eq. (3) to Eq. (4) yields

$$\begin{aligned}&\mathbf{Adv}^{\mathrm{prf}}_{F}(B) + \mathbf{Adv}^{\mathrm{tkprp}}_{E}(C) \\ &= \frac{1}{2}\left(\Pr\left[A^{\mathcal{E}^{F_{K_1},E_{K_2},E_{K_3}}} \Rightarrow 1\right] - \Pr\left[A^{\mathcal{E}^{\rho,\phi,\omega}} \Rightarrow 1\right]\right) \\ &= \frac{1}{2}\left(\Pr\left[A^{\mathcal{E}^{F_{K_1},E_{K_2},E_{K_3}}} \Rightarrow 1\right] - \Pr\left[A^{\pi} \Rightarrow 1\right]\right. \\ &\quad \left.+\Pr\left[A^{\pi} \Rightarrow 1\right] - \Pr\left[A^{\mathcal{E}^{\rho,\phi,\omega}} \Rightarrow 1\right]\right) \\ &= \frac{1}{2}\left(\mathbf{Adv}^{\mathrm{prp}}_{\mathcal{E}^{F_{K_1},E_{K_2},E_{K_3}}}(A) - \mathbf{Adv}^{\mathrm{prp}}_{\mathcal{E}^{\rho,\phi,\omega}}(A)\right),\end{aligned}$$

where $\pi$ is a random permutation on the domain of $\mathcal{E}$. Hence, we obtain

$$\mathbf{Adv}^{\mathrm{prp}}_{\mathcal{E}^{F_{K_1},E_{K_2},E_{K_3}}}(A) = \mathbf{Adv}^{\mathrm{prp}}_{\mathcal{E}^{\rho,\phi,\omega}}(A) + 2\mathbf{Adv}^{\mathrm{prf}}_{F}(B) + 2\mathbf{Adv}^{\mathrm{tkprp}}_{E}(C).$$

There may be adversaries $B'$, $C'$ that are better than $B, C$ and have the same computational resource as $B, C$. ∎

## 3. Length-preserving CBC Mode

This section describes a new length-preserving enciphering scheme (LPCBC) that is constructed from pseudorandom function $F : \mathcal{K}_F \times \{0,1\}^{\ell-n} \to \{0,1\}^n$ and blockcipher $E : \mathcal{K}_E \times \{0,1\}^n \to \{0,1\}^n$. The enciphering scheme has key space $\mathcal{K}_F \times \mathcal{K}_E \times \mathcal{K}_E$. The plaintext space and the ciphertext space are $\{0,1\}^\ell$ where $\ell \geq n$. We assume that $\ell$ is fixed. Let $t = n - (\ell \bmod n) \bmod n$.

We describe the encryption $\mathcal{E}$ and the decryption $\mathcal{D}$ of LPCBC below and illustrate an example of LPCBC in Fig. 1. Roughly speaking, LPCBC is a reversing CBC mode such that an initialization vector is replaced with the output of pseudorandom function $F$.

**Encryption** $\mathcal{E}_{K_1,K_2,K_3}(P)$ where $(K_1, K_2, K_3) \in \mathcal{K}_F \times \mathcal{K}_E \times \mathcal{K}_E$ and $P \in \{0,1\}^\ell$.

1. Divide $P$ into $m$ blocks $P_1, P_2, \ldots, P_m$ such that $P_1$ is an $(n-t)$-bit block and all other blocks are $n$-bit blocks
2. $C_{m+1} \leftarrow F_{K_1}(P_1 \,\|\, P_2 \,\|\, \ldots \,\|\, P_{m-1})$.
3. For $i = m$ to 1 do
   if $i = m$, then $C_m \leftarrow E_{K_2}(P_m \oplus C_{m+1})$,
   if $m > i > 2$, then $C_i \leftarrow E_{K_3}(P_i \oplus C_{i+1})$,
   if $i = 2$, then $C'_2 \leftarrow E_{K_3}(P_2 \oplus C_3)$, $C_2 \leftarrow$ (the right $n-t$ bits of $C'_2$),
   if $i = 1$, then $C_i \leftarrow E_{K_3}((0^t \,\|\, P_1) \oplus C'_2)$.
4. Return $C_1 \,\|\, C_2 \,\|\, C_3 \,\|\, \ldots \,\|\, C_m$ as a ciphertext $C$.

**Decryption** $\mathcal{D}_{K_1,K_2,K_3}(C)$ where $(K_1, K_2, K_3) \in \mathcal{K}_F \times \mathcal{K}_E \times \mathcal{K}_E$ and $C \in \{0,1\}^\ell$.

1. Divide $C$ into $m$ blocks $C_1, C_2, \ldots, C_m$ such that $C_2$ is an $(n-t)$-bit block and all other blocks are $n$-bit blocks.
2. $U \leftarrow D_{K_2}(C_1) \oplus (0^t \,\|\, C_2)$, $P_1 \leftarrow$(the right $(n-t)$ bits of $U$), and $V \leftarrow$(the left $t$ bits of $U$).
3. For $i = 2$ to $m$ do
   if $i = 2$, then $P_2 \leftarrow D_{K_2}(V \,\|\, C_2) \oplus C_3$,
   if $2 < i < m$, then $P_i \leftarrow D_{K_2}(C_i) \oplus C_{i+1}$,
   if $i = m$, then $P_m \leftarrow D_{K_2}(C_m) \oplus F_{K_1}(P_1 \,\|\, \ldots \,\|\, P_{m-1})$.
4. Return $P_1 \,\|\, P_2 \,\|\, \ldots \,\|\, P_m$ as a plaintext $P$.

In order to handle any length of a plaintext, LPCBC uses ciphertext stealing [7], [17] for the first two blocks. In the decryption, any plaintext block $P_i$ except for the last plaintext block $P_m$ can be computed only from two ciphertext blocks $C_i, C_{i+1}$. Notice that it is unnecessary to keep plaintext blocks $P_1, \ldots, P_{m-1}$ if $F$ is a streamable pseudorandom function such as HMAC.

## 4. Security Analysis

### 4.1 Main Theorem

Let $\mathcal{E}$ be the encryption of LPCBC for an $\ell$-bit plaintext $P_1 \,\|\, P_2 \,\|\, \ldots \,\|\, P_m$ where $m \geq 2$. **Figure 2** describes the pseudo code of $\widehat{\mathcal{E}}$. In Fig. 2, a function $\lambda$ corresponds to the pseudorandom function $F_{K_1}$, a function $\mu$ corresponds to the blockcipher $E_{K_2}$, and a function $\nu$ corresponds to the blockcipher $E_{K_3}$. We analyze the security of $\mathcal{E}$ such that $\lambda$ is a random function $\rho$, $\mu$ and $\nu$ are independently random permutations $\phi, \omega$ because of Theorem 1. We will prove the following theorem in Section 4.2.

**Theorem 2** Consider any adversary $A$ that makes at most $q$ queries to $\mathcal{E}$ or $\pi$. Here, $\mathcal{E}$'s oracles are $\rho \xleftarrow{\$} \mathrm{Func}(\{0,1\}^{n(m-1)}, \{0,1\}^n)$, $\phi \xleftarrow{\$} \mathrm{Perm}(\{0,1\}^n)$, and $\omega \xleftarrow{\$} \mathrm{Perm}(\{0,1\}^n)$, and $\pi$ is chosen from $\mathrm{Perm}(\{0,1\}^{nm})$ at random. The advantage of $A$ is given by

$$\mathbf{Adv}^{\mathrm{prp}}_{\mathcal{E}^{\rho,\phi,\omega}}(A) = \Pr\left[A^{\mathcal{E}^{\rho,\phi,\omega}} \Rightarrow 1\right] - \Pr\left[A^{\pi} \Rightarrow 1\right].$$

Suppose that $1 \leq q \leq 2^{(n+1)/2}$ and $m, n \geq 2$. Then, we have

$$\frac{0.14(q-2)(q-3)}{2^{n+1}} \leq \mathbf{Adv}^{\mathrm{prp}}_{\mathcal{E}^{\rho,\phi,\omega}}(A) \leq \frac{6q(q-1) + q'(q'-1)}{2^{n+1}} \qquad (5)$$

where $q' = q(m-1)$. ∎

```
1: function E(P_1, ..., P_m)
2:     Q ← P_1 || ... || P_{m-1}
3:     C_{m+1} ← λ(Q)
4:     Z_m ← P_m ⊕ C_{m+1}
5:     C_m ← μ(Z_m)
6:     for i = m − 1 to 1 do
7:         if i ≠ 1 then
8:             Z_i ← P_i ⊕ C_{i+1}
9:         else
10:            Z_1 ← (0^t || P_1) ⊕ C'_2
11:        end if
12:        C_i ← ν(Z_i)
13:        if i = 2 then
14:            C'_2 ← C_2
15:            C_2 ← the right (n − t) bits of C'_2
16:        end if
17:    end for
18:    return C_1, ..., C_m
19: end function
```

**Fig. 2** The pseudo code of $\mathcal{E}$.

```
1: function λ(Q)
2:     L ←$ {0,1}^n
3:     return L
4: end function
```

```
1: function μ(Z_m)
2:     U ←$ {0,1}^n
3:     return U
4: end function
```

```
1: function ν(Z_i)
2:     V ←$ {0,1}^n
3:     return V
4: end function
```

**Fig. 3** Game 1.

```
1: function λ(Q)
2:     L ←$ {0,1}^n
3:     return L
4: end function
```

```
1: function μ(Z_m)
2:     U ←$ {0,1}^n
3:     return U
4: end function
```

```
1: function ν(Z_i)
2:     if ν[Z_i] =⊥ then
3:         V ←$ {0,1}^n
4:         ν[Z_i] ← V
5:     else
6:         V ← ν[Z_i]
7:         bad_1 ← true
8:     end if
9:     return V
10: end function
```

**Fig. 4** Game 2.

LPCBC is not a SPRP because $P_1$ is determined only by $C_1$ and $C_2$. Namely, if $C_1$ and $C_2$ are fixed in the decryption, then $P_1$ is fixed. If an adversary is allowed to have access to the decryption oracle, then the adversary can easily distinguish LPCBC from a random permutation. In order to achieve streamable decryption, we have to give up constructing a SPRP.

### 4.2 Proof of Theorem 2

The pseudo code of $\mathcal{E}$ in Fig. 2 is common in all six games. The games shown in **Fig. 3** – **Fig. 8** differ in the definition of functions $\lambda, \mu, \nu$. The last game (i.e., Game 6) is the pseudo code of $\mathcal{E}^{\rho,\phi,\omega}$.

Game 1: We define Game 1 as Fig. 3. In Game 1, each $C_i$ is always chosen from $\{0,1\}^n$ uniformly even if the same input is given to $\nu$. Noting that $A$ does not make the same query, we have

```
1: function λ(Q)
2:     L ←$ {0,1}^n
3:     return L
4: end function
```

```
1: function μ(Z_m)
2:     U ←$ {0,1}^n
3:     return U
4: end function
```

```
1: function ν(Z_i)
2:     if ν[Z_i] =⊥ then
3:         V ←$ {0,1}^n
4:         if V ∈ V then
5:             V ←$ {0,1}^n \ V
6:             V ← V ∪ {V}
7:             bad_2 ← true
8:         end if
9:         ν[Z_i] ← V
10:    else
11:        V ← ν[Z_i]
12:        bad_1 ← true
13:    end if
14:    return V
15: end function
```

**Fig. 5** Game 3.

```
1: function λ(Q)
2:     if λ[Q] =⊥ then
3:         L ←$ {0,1}^n
4:         λ[Q] ← L
5:     else
6:         L ← λ[Q]
7:     end if
8:     return L
9: end function
```

```
1: function μ(Z_m)
2:     U ←$ {0,1}^n
3:     return U
4: end function
```

```
1: function ν(Z_i)
2:     if ν[Z_i] =⊥ then
3:         V ←$ {0,1}^n
4:         if V ∈ V then
5:             V ←$ {0,1}^n \ V
6:             V ← V ∪ {V}
7:             bad_2 ← true
8:         end if
9:         ν[Z_i] ← V
10:    else
11:        V ← ν[Z_i]
12:        bad_1 ← true
13:    end if
14:    return V
15: end function
```

**Fig. 6** Game 4.

$$\Pr\left[A^{\text{Game1}} \Rightarrow 1\right] - \Pr\left[\varrho \xleftarrow{\$} \text{Func}(\{0,1\}^{nm}, \{0,1\}^{nm}) : A^{\varrho} \Rightarrow 1\right] = 0. \quad (6)$$

Game 2: We define Game 2 as Fig. 4. In Fig. 4, a table $\mu[Z_i]$ is initialized with $\perp$ and a flag $bad_1$ is initialized with `false`. The flag is set to true if and only if Game 2 behaves differently from Game 1, that is, the same input $Z_i$ is given to $\nu$.

$$\Pr\left[A^{\text{Game2}} \Rightarrow 1\right] - \Pr\left[A^{\text{Game1}} \Rightarrow 1\right] \leq \Pr\left[A \text{ sets } bad_1\right] \quad (7)$$

Let $P_i^{(j)}$ be the $i$-th plaintext block of the $j$-th query made by $A$. Corresponding variables are denoted by superscript notation $(j)$. Let $r$ be the number of invocations of $\nu$. Given $r$, $(i, j)$ is uniquely determined by $r = (j-1)(m-1) + (m-i)$ because the adversary is allowed to make queries only to $\mathcal{E}$ and is not allowed to make queries to subroutines $\lambda, \mu, \nu$. In order to describe the correspondence, we define functions rtoi($r$), rtoj($r$) as

```
function λ(Q)
    if λ[Q] =⊥ then
        L ←$ {0,1}^n
        λ[Q] ← L
    else
        L ← λ[Q]
    end if
    return L
end function
```

```
function μ(Z_m)
    if μ[Z_m] =⊥ then
        U ←$ {0,1}^n
        μ[Z_m] ← U
    else
        U ← μ[Z_m]
        bad_3 ← true
    end if
    return U
end function
```

```
function ν(Z_i)
    if ν[Z_i] =⊥ then
        V ←$ {0,1}^n
        if V ∈ V then
            V ←$ {0,1}^n \ V
            V ← V ∪ {V}
            bad_2 ← true
        end if
        ν[Z_i] ← V
    else
        V ← ν[Z_i]
        bad_1 ← true
    end if
    return V
end function
```

**Fig. 7** Game 5.

```
function λ(Q)
    if λ[Q] =⊥ then
        L ←$ {0,1}^n
        λ[Q] ← L
    else
        L ← λ[Q]
    end if
    return L
end function
```

```
function μ(Z_m)
    if μ[Z_m] =⊥ then
        U ←$ {0,1}^n
        if U ∈ U then
            U ←$ {0,1}^n \ U
            U ← U ∪ {U}
            bad_4 ← true
        end if
        μ[Z_m] ← U
    else
        U ← μ[Z_m]
        bad_3 ← true
    end if
    return U
end function
```

```
function ν(Z_i)
    if ν[Z_i] =⊥ then
        V ←$ {0,1}^n
        if V ∈ V then
            V ←$ {0,1}^n \ V
            V ← V ∪ {V}
            bad_2 ← true
        end if
        ν[Z_i] ← V
    else
        V ← ν[Z_i]
        bad_1 ← true
    end if
    return V
end function
```

**Fig. 8** Game 6.

$$i = \mathrm{rtoi}(r) = m - (r \bmod (m-1)), \quad j = \mathrm{rtoj}(r) = \left\lfloor \frac{r}{m-1} \right\rfloor.$$

By using the functions, superscript $[r]$ is often used instead of $(i, j)$. For example, $P_i^{(j)}$ is identical to $P^{[r]}$. Let $\mathtt{B1}^{[r]}$ be the event that $C^{[a]} = C^{[b]}$ for $1 \le \exists\, a < \exists\, b < r$. Supposing that $\mathtt{B1}^{[r-1]}$ does not occur, we evaluate the probability that $\mathtt{B1}^{[r]}$ occurs, which is the probability that $Z^{[r]}$ collides with $Z^{[a]}$ for $1 \le \exists\, a \le r-1$. We have no idea how $P^{[r]}$ was chosen because it depends on $A$. However, one of the following cases holds.

( 1 ) $P^{[r]}$ is the $(m-1)$-th block (i.e., $\mathrm{rtoi}(r) = m-1$):
Let $j = \mathrm{rtoj}(r)$. $C_m^{(j)}$ is chosen from $\{0,1\}^n$ at random due to function $\mu$. Hence, $Z_{m-1}^{(j)}$ is uniformly distributed on $\{0,1\}^n$ regardless of $P_i^{(j)}$.

( 2 ) $P^{[r]}$ is a subsequent block (i.e., $1 \le \mathrm{rtoi}(r) \le m-2$):
Let $i = \mathrm{rtoi}(r)$ and $j = \mathrm{rtoj}(r)$. Since the value of $Z^{[r-1]}$ is fresh by our assumption of $\overline{\mathtt{B1}^{[r-1]}}$, $C^{[r-1]}$ is uniformly distributed on $\{0,1\}^n$. It follows that $Z^{[r]}$ is also distributed on $\{0,1\}^n$ uniformly because $C^{[r-1]}$ was chosen after $A$ chose $P^{[r]}$.

In both of cases, we obtain

$$\Pr\left[\mathtt{B1}^{[r]} | \overline{\mathtt{B1}^{[r-1]}}\right] \le \frac{r-1}{2^n}.$$

When $A$ makes $q$ queries to $\mathcal{E}$, function $\nu$ is invoked $q(m-1)$ times. Hence, the probability that $bad_1$ is set to `true` is given by

$$\Pr[A \text{ sets } bad_1] \le \sum_{r=1}^{q(m-1)} \Pr\left[\mathtt{B1}^{[r]} | \overline{\mathtt{B1}^{[r-1]}}\right] \le \frac{q(m-1)(q(m-1)-1)}{2^{n+1}}.$$

Substituting the inequality above into Eq. (7) yields

$$\Pr\left[A^{\mathrm{Game2}} \Rightarrow 1\right] - \Pr\left[A^{\mathrm{Game1}} \Rightarrow 1\right] \le \frac{q(m-1)(q(m-1)-1)}{2^{n+1}}. \tag{8}$$

Game 3: We define Game 3 as Fig. 5. In Fig. 5, a set $\mathcal{V}$ is initialized with the empty set, and a flag $bad_2$ is initialized with `false`. The flag is set to true if and only if Game 3 behaves differently from Game 2.

$$\Pr\left[A^{\mathrm{Game3}} \Rightarrow 1\right] - \Pr\left[A^{\mathrm{Game2}} \Rightarrow 1\right] \le \Pr[A \text{ sets } bad_2] \tag{9}$$

Let $\mathtt{B2}^{[r]}$ be the event that $bad_2$ is set to `true` in $r$ invocations of $\nu$. Suppose that $\mathtt{B2}^{[r-1]}$ does not occur. Then, the probability that $\mathtt{B1}^{[r]}$ occurs is

$$\Pr\left[\mathtt{B2}^{[r]} | \overline{\mathtt{B2}^{[r-1]}}\right] \le \frac{r-1}{2^n}.$$

When $A$ makes $q$ queries to $\mathcal{E}$, function $\nu$ is invoked $q(m-1)$ times. Hence, the probability that $bad_2$ is set to `true` is given by

$$\Pr[A \text{ sets } bad_2] \le \sum_{r=1}^{q(m-1)} \Pr\left[\mathtt{B2}^{[r]} | \overline{\mathtt{B2}^{[r-1]}}\right] \le \frac{q(m-1)(q(m-1)-1)}{2^{n+1}}.$$

Substituting the inequality above into Eq. (9) yields

$$\Pr\left[A^{\mathrm{Game3}} \Rightarrow 1\right] - \Pr\left[A^{\mathrm{Game2}} \Rightarrow 1\right] \le \frac{q(m-1)(q(m-1)-1)}{2^{n+1}}. \tag{10}$$

Game 4: We define Game 4 as Fig. 6. In Fig. 6, a table $\lambda[Q]$ is initialized with $\perp$. From the viewpoint of $A$, Game 4 is identical to Game 3 because ciphertext $(C_1, \ldots, C_m)$ is independent of $\lambda(Q)$.

$$\Pr\left[A^{\mathrm{Game4}} \Rightarrow 1\right] - \Pr\left[A^{\mathrm{Game3}} \Rightarrow 1\right] = 0 \tag{11}$$

Game 5: We define Game 5 as Fig. 7. In Fig. 7, a table $\mu[Z_m]$ is initialized with $\perp$ and a flag $bad_3$ is initialized with `false`. The flag is set to `true` if and only if Game 5 behaves differently from Game 4.

$$\Pr\left[A^{\text{Game5}} \Rightarrow 1\right] - \Pr\left[A^{\text{Game4}} \Rightarrow 1\right] \le \Pr\left[A \text{ sets } bad_3\right] \quad (12)$$

Let $\mathtt{B3}^{[q]}$ be the event that $A$ sets $bad_3$ in $q$ queries to $\mathcal{E}$. Supposing that $\mathtt{B3}^{[q-1]}$ does not occur, we evaluate the probability that $\mathtt{B3}^{[q]}$ occurs, denoted by $\Pr\left[\mathtt{B3}^{[q]} | \overline{\mathtt{B3}^{[q-1]}}\right]$. The assumption implies that $Z_m^{(1)}, Z_m^{(2)}, \ldots, Z_m^{(q-1)}$ are different each other. In order to make $\mathtt{B3}^{[q]}$ occur, $A$ must choose $P_m^{(q)}$ satisfying

$$P_m^{(q)} = P_m^{(k)} \oplus \lambda(Q^{(k)}) \oplus \lambda(Q^{(q)}). \quad (13)$$

for $1 \le \exists\, k \le q-1$. That is, $A$ has to guess the value of $\lambda(Q^{(k)}) \oplus \lambda(Q^{(q)})$. Notice that $\lambda(Q^{(k)})$ and $\lambda(Q^{(q)})$ are unknown to $A$ and we have no idea how $P_m^{(k)}$ was chosen. However, one of the following three cases holds when $k$ is fixed.

(1) $Q^{(q)} = Q^{(k)}$:
It follows that $P_m^{(q)} = P_m^{(k)}$ because $\lambda(Q^{(q)}) = \lambda(Q^{(k)})$. This case does not occur since $A$ does not repeat the same query.

(2) $Q^{(q)} \ne Q^{(k)} \wedge Q^{(q)} = Q^{(t)}$ for $1 \le \exists\, t \le q-1, t \ne k$:
The condition means that $Q^{(q)}$ is an already-answered query. Since $\lambda(Q^{(t)})$ and $\lambda(Q^{(k)})$ are independently chosen from $\{0,1\}^n$, $\lambda(Q^{(k)}) \oplus \lambda(Q^{(q)})$ is uniformly distributed on $\{0,1\}^n$. The probability that $A$ succeeds in guessing $\lambda(Q^{(k)}) \oplus \lambda(Q^{(q)})$ is at most $2^{-n}$.

(3) $Q^{(q)} \ne Q^{(t)}$ for $1 \le \forall\, t \le q-1$:
The condition means that $Q^{(q)}$ is fresh. Since $\lambda(Q^{(q)})$ is chosen from $\{0,1\}^n$ independently from $\lambda(Q^{(k)})$, $\lambda(Q^{(k)}) \oplus \lambda(Q^{(q)})$ is uniformly distributed on $\{0,1\}^n$. The probability that Eq. (13) holds is at most $2^{-n}$.

We hence obtain

$$\Pr\left[\mathtt{B3}^{[q]} | \overline{\mathtt{B3}^{[q-1]}}\right] \le \frac{q-1}{2^{n-1}}.$$

The probability that $A$ sets $bad_3$ in $q$ queries is given by

$$\Pr\left[A \text{ sets } bad_3\right] \le \sum_{i=1}^{q} \Pr\left[\mathtt{B3}^{[i]} | \overline{\mathtt{B3}^{[i-1]}}\right] \le \frac{q(q-1)}{2^n}.$$

Substituting the inequality above into Eq. (12) yields

$$\Pr\left[A^{\text{Game5}} \Rightarrow 1\right] - \Pr\left[A^{\text{Game4}} \Rightarrow 1\right] \le \frac{q(q-1)}{2^n}. \quad (14)$$

Game 6: We define Game 6 as Fig. 8. In Fig. 8, a set $\mathcal{U}$ is initialized with the empty set, and a flag $bad_4$ is initialized with `false`. The flag is set to `true` if and only if Game 6 behaves differently from Game 5.

$$\Pr\left[A^{\text{Game6}} \Rightarrow 1\right] - \Pr\left[A^{\text{Game5}} \Rightarrow 1\right] \le \Pr\left[A \text{ sets } bad_4\right] \quad (15)$$

When $A$ makes $q$ queries to $\mathcal{E}$, the probability that $A$ sets $bad_4$ is

$$\Pr\left[A \text{ sets } bad_4\right] \le \frac{q(q-1)}{2^{n+1}}.$$

Substituting the inequality above into Eq. (15) yields

$$\Pr\left[A^{\text{Game6}} \Rightarrow 1\right] - \Pr\left[A^{\text{Game5}} \Rightarrow 1\right] \le \frac{q(q-1)}{2^{n+1}}. \quad (16)$$

Finally, recalling that

$$\Pr\left[\varrho \xleftarrow{\$} \text{Func}(\{0,1\}^{nm}, \{0,1\}^{nm}) : A^{\varrho} \Rightarrow 1\right] - \Pr\left[\pi \xleftarrow{\$} \text{Perm}(\{0,1\}^{nm}) : A^{\pi} \Rightarrow 1\right] \le \frac{q(q-1)}{2^{n+1}},$$

we obtain the upper bound on the advantage of $A$ from the differences between two games (i.e., Eqs. (6), (8), (10), (11), (14), (16)) as follows:

$$\mathbf{Adv}^{\text{prp}}_{\mathcal{E}}(A) = \Pr\left[A^{\text{Game6}} \Rightarrow 1\right] - \Pr\left[\pi \xleftarrow{\$} \text{Perm}(\{0,1\}^{nm}) : A^{\pi} \Rightarrow 1\right] \le \frac{6q(q-1) + q'(q'-1)}{2^{n+1}},$$

where $q$ is the number of queries to $\mathcal{E}$ and $q' = q(m-1)$. The inequality above is the right-hand inequality of Eq. (5).

We next evaluate the lower bound of the advantage of $A$. Consider the following algorithm such that $A$ makes $q$ queries to an oracle $\mathcal{O} \in \{\mathcal{E}^{\rho,\phi,\omega}, \pi\}$.

(1) Let $P_m^{(j)} = 0^n$ for $1 \le j \le q-2$. Choose $P_1^{(j)}, \ldots, P_{m-1}^{(j)}$ at random for $j = 1, 2, \ldots, q-2$. Make queries $(P_1^{(j)}, \ldots, P_m^{(j)})$ to $\mathcal{O}$ for $j = 1, 2, \ldots, q-2$. Let $C_m^{(j)}$ be the $m$-th ciphertext block for the $j$-th query.

(2) Find $j, j'$ such that $C_m^{(j)} = C_m^{(j')}$ for $1 \le j < j' \le q-2$.

(3) If such $j, j'$ are found, then make the $(q-1)$-th query $(P_1^{(j)}, \ldots, P_{m-1}^{(j)}, 10^{n-1})$ and the $q$-th query $(P_1^{(j')}, \ldots, P_{m-1}^{(j')}, 10^{n-1})$ to $\mathcal{O}$. If $C_m^{(q-1)} = C_m^{(q)}$, then return 1, otherwise return 0.

(4) If such $j, j'$ are not found, then return 0;

Suppose that $\mathcal{O} = \mathcal{E}^{\rho,\phi,\omega}$. According to Fact 14 in the article written by Bellare et al. [2], the probability that $A$ finds such $j, j'$ is

$$\Pr\left[A \text{ finds such } j, j'\right] \ge 0.3 \cdot \frac{(q-2)(q-3)}{2^n}.$$

The inequality above requires the assumption that $1 \le q \le 2^{(n+1)/2}$. When $A$ finds such $j, j'$, $A$ always outputs 1. Hence, we have

$$\Pr\left[A^{\mathcal{O}} \Rightarrow 1 \mid \mathcal{O} = \mathcal{E}\right] \ge \frac{0.3(q-2)(q-3)}{2^n}. \quad (17)$$

Suppose that $\mathcal{O} = \pi$. The probability that $A$ finds such $j, j'$ is

$$\Pr\left[A \text{ finds such } j, j'\right] \le \frac{(q-2)(q-3)}{2^{n+1}}$$

When $A$ finds such $j, j'$, the probability that $A$ outputs 1 is

$$\Pr\left[A^{\mathcal{O}} \Rightarrow 1 \mid A \text{ finds such } j, j'\right] \le \frac{2^{n(m-1)} - 1}{2^{nm} - (q-1)}.$$

Hence, we have

$$\Pr\left[A^{\mathcal{O}} \Rightarrow 1 \mid \mathcal{O} = \pi\right] \le \frac{(q-2)(q-3)}{2^{n+1}} \cdot \frac{2^{n(m-1)} - 1}{2^{nm} - (q-1)}.$$

Since $\Pr\left[\mathcal{O} = \mathcal{E}^{\rho,\phi,\omega}\right] = \Pr\left[\mathcal{O} = \pi\right] = 1/2$, the advantage of $A$ is given by

$$\begin{aligned}
&\Pr\left[A^{\mathcal{E}} \Rightarrow 1\right] - \Pr\left[A^{\pi} \Rightarrow 1\right] \\
&\ge \frac{1}{2} \cdot \frac{0.3(q-2)(q-3)}{2^n} - \frac{1}{2} \frac{(q-2)(q-3)}{2^{n+1}} \cdot \frac{2^{n(m-1)} - 1}{2^{nm} - (q-1)} \\
&\ge \frac{(q-2)(q-3)}{2^{n+1}} \left(0.3 - \frac{1}{2} \cdot \frac{2^{n(m-1)} - 1}{2^{nm} - (q-1)}\right) \\
&\ge \frac{0.14(q-2)(q-3)}{2^{n+1}},
\end{aligned}$$

where the last inequality holds for $n \ge 2$. The inequality above is the left-hand inequality of Eq. (5). This completes the proof of Theorem 2.

## 5. Concluding Remarks

This paper proposes a length-preserving enciphering scheme that achieves the PRP security and the streamable decryption. Our enciphering scheme is suitable for secure communication on narrowband channels and memory-constrained devices. Our enciphering scheme requires a streamable pseudorandom function and a blockcipher as primitives. For example, the streamable pseudorandom function is instantiated with HMAC. Our enciphering scheme requires three keys, which might be a problem in some application. Reducing the number of keys might therefore be significant in improving usability.

### References


[1] Aumasson, J.-P., Henzen, L., Meier, W. and Naya-Plasencia, M.: QUARK: A lightweight hash, *Cryptographic Hardware and Embedded Systems – CHES 2010, Lecture Notes in Computer Science*, Vol.6225, pp.1–15 (2010).

[2] Bellare, M., Desai, A., Jokipii, E. and Rogaway, P.: A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation, pp.1–31 (2000), available from ⟨http://www-cse.ucsd.edu/users/mihir/papers/sym-enc.html⟩.

[3] Bellare, M. and Rogaway, P.W.: Block cipher mode of operation for secure, length-preserving encryption, United States Patent 5673319 (1997).

[4] Bogdanov, A., Knudsen, L., Leander, G., Paar, C., Poschmann, A., Robshaw, M., Seurin, Y. and Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher, *Cryptographic Hardware and Embedded Systems – CHES 2007, Lecture Notes in Computer Science*, Vol.4727, pp.450–466 (2007).

[5] Cannière, C., Dunkelman, O. and Knežević, M.: KATAN and KTANTAN – A Family of Small and Efficient Hardware-Oriented Block Ciphers, *Cryptographic Hardware and Embedded Systems – CHES 2009, 11th International Workshop, Lecture Notes in Computer Science*, Vol.5747, pp.272–288 (2009).

[6] Chakraborty, D. and Sarkar, P.: HCH: A New Tweakable Enciphering Scheme Using the Hash-Counter-Hash Approach, Cryptology ePrint Archive, Report 2007/028 (2007), available from ⟨http://eprint.iacr.org/⟩.

[7] Daemen, J.: Cipher and Hash Function Design Strategies Based on Linear and Differential Cryptanalysis, PhD Thesis, Katholieke Universiteit Leuven (1995).

[8] Guo, J., Peyrin, T. and Poschmann, A.: The PHOTON Family of Lightweight Hash Functions, *Advances in Cryptology – CRYPTO 2011, Lecture Notes in Computer Science*, Vol.6841, pp.222–239 (2011).

[9] Guo, J., Peyrin, T., Poschmann, A. and Robshaw, M.: The LED Block Cipher, *Cryptographic Hardware and Embedded Systems – CHES 2011, Lecture Notes in Computer Science*, Vol.6917, pp.326–341 (2011).

[10] Halevi, S.: EME*: Extending EME to Handle Arbitrary-Length Messages with Associated Data, *Progress in Cryptology – INDOCRYPT 2004, Lecture Notes in Computer Science*, Vol.3348, pp.315–327 (2004).

[11] Halevi, S.: Invertible Universal Hashing and the TET Encryption Mode, *Advances in Cryptology – CRYPTO 2007, Lecture Notes in Computer Science*, Vol.4622, pp.412–429 (2007).

[12] Halevi, S. and Rogaway, P.: A Tweakable Enciphering Mode, *Advances in Cryptology – CRYPTO 2003, Lecture Notes in Computer Science*, Vol.2729, pp.482–499 (2003).

[13] Knudsen, L., Leander, G., Poschmann, A. and Robshaw, M.J.B.: PRINTcipher: A Block Cipher for IC-Printing, *Cryptographic Hardware and Embedded Systems – CHES 2010, Lecture Notes in Computer Science*, Vol.6225, pp.16–31 (2010).

[14] Minematsu, K. and Tsunoo, Y.: Hybrid Symmetric Encryption Using Known-Plaintext Attack-Secure Components, *Information Security and Cryptology – ICISC 2005, Lecture Notes in Computer Science*, Vol.3935, pp.242–260 (2006).

[15] Naor, M. and Reingold, O.: A Pseudo-Random Encryption Mode (2001), available from ⟨http://www.wisdom.weizmann.ac.il/˜naor/PAPERS/nr-mode.ps⟩

[16] Sarkar, P.: Improving Upon the TET Mode of Operation, *Information Security and Cryptology – ICISC 2007, Lecture Notes in Computer Science*, Vol.4817, pp.180–192 (2007).

[17] Schneier, B.: *APPLIED CRYPTOGRAPHY* (*Second Edition*), John Wiley & Sons, Inc. (1996).

[18] Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T. and Shirai, T.: Piccolo: An Ultra-Lightweight Blockcipher, *Cryptographic Hardware and Embedded Systems – CHES 2011, Lecture Notes in Computer Science*, Vol.6917, pp.342–357 (2011).

[19] Wang, P., Feng, D. and Wu, W.: HCTR: A Variable-Input-Length Enciphering Mode, *Information Security and Cryptology, Lecture Notes in Computer Science*, Vol.3822, pp.175–188 (2005).


**Hidenori Kuwakado** received his B.E., M.E. and D.E. degrees from Kobe University in 1990, 1992, and 1999 respectively. He worked for Nippon Telegraph and Telephone Corporation from 1992 to 1996. From 1996 to 2002 he was a Research Associate in the Faculty of Engineering, Kobe University. From 2002 to 2007, he was an Associate Professor in the Faculty of Engineering, Kobe University. Since 2007, he has been an Associate Professor in Graduate School of Engineering, Kobe University. His research interests are in cryptography and information security.