An Efficient SPRP-secure Construction based on Pseudo Random Involution

Mridul Nandi

Indian Statistical Institute, Kolkata
mridul.nandi@gmail.com

Abstract. Here we present a new security notion called as pseudo random involution or PRI which are associated with tweakable involution enciphering schemes or TIES (i.e., the encryption and decryption are same algorithm). This new security notion is important in two reasons. Firstly, it is the natural security notion for TIES which are having practical importance. Secondly, we show that there is a generic method to obtain a sprp-secure tweakable enciphering scheme (TES) from pri-secure construction. The generic method costs an extra xor with an extra key. In this paper, we also propose an efficient pri-secure construction Hash-Counter Involution or HCI and based on it we obtain a sprp-secure construction which is real improvement over XCB. We call the new construction as MXCB or Modified-XCB. HCH, XCB and HCTR are some of the popular counter based enciphering schemes, where HCTR is more efficient among them and HCH, XCB guarantee more security compare to HCTR. The new proposal MXCB has efficiency similar to HCTR and guarantees more security similar to HCH and XCB. We consider this new construction to be an important in light of the current activities of the IEEE working group on storage security which is working towards a standard for a wide block TES.

Keywords: Modes of operation, involution, tweakable enciphering scheme, strong pseudo random permutation, poly-hash, counter.

1 Introduction

A mode of operation is a method of constructing an encryption algorithm which can encrypt arbitrary length messages. It uses a cryptographic object called block cipher, as an underlying object and possibly some algebraic operations such as finite field multiplication. (Strong) Pseudo Random Permutation or (S)PRP [13], authenticity and privacy [10, 20, 21] are some of the desired security notions for symmetric key encryptions. Later, Liskov et al. [12] followed by Halevi-Rogaway [8] considered tweakable version of length-preserving SPRP, which allows us to process associated data or tweak as a part of the messages. Disk-encryption is one of the important application for the length-preserving tweakable SPRP as mentioned in [8]. Motivated by disc-encryption algorithms, there are several tweakable-SPRP proposals. We list some of these important constructions based on three categories.

1. Hash-Encrypt-Hash: First introduced by Naor-Reingold [17, 18], consists of Encryption layer between two layers of invertible hash. Similar approach is considered in TET [6] and HEH [22], where latter is an improvement over TET.
2. Encrypt-Mix-Encrypt: Halevi-Rogaway [8] introduced Encryption, mixing and Encryption approach. Some of the constructions of this type are CMC [8], EME [7] and EME* [5] (modification of EME which can encrypt arbitrary size messages).

3. Hash-Ctr-Hash: This approach is fist observed in the original proposal of XCB [16]. Later, HCTR [24], HCH [3] and a new version of XCB with security bound [15] are of this type. The first hash function layer is to generate counter. Based on the counter, we obtain ciphertext except one block which is computed by using the second hash layer. In this paper, we are mainly interested in this type of modes of operations.

1.1 Tweakable involution enciphering scheme and its pri security notion

In practice, we need to implement both encryption and decryption and sometimes we need both together as a single application. For example, in disk encryption, it is always better if the same device can read and write the data. Ideal choice for this kind of scenario is an involution which is inverse of itself. In other words, tweakable involution enciphering schemes or TIES are those enciphering schemes whose encryption and decryption algorithm are same. If there are some amount of differences in encryption and decryption, then efficiency of the combined implementation (both encryption and decryption together) is reduced to some extent. More precisely, one can observe this loss of efficiency in hardware implementation as we need to use several multiplexors to control encryption and decryption together. Obviously, we would not have to face this problem in TIES. Unfortunately, any TIES can not be strong pseudo random permutation. More particularly, the involution property itself can be used to distinguish it from an uniform random permutation. In this paper we first define a suitable security definition for TIES. We call it as pseudo random involution (similar to pseudo random function or permutation). After defining the new security notion, it is a natural question that how strong is this security notion with respect to the known security. We show that any pri-secure construction is wprf (weak pseudo random function) and moreover, it can be prf if we have some restrictions on the distinguisher. More importantly, we show a generic method to construct a sprp from a pri-secure construction at the cost of one xor and one extra key. Thus, we believe that the new security notion can have importance in both theoretical and practical point of view.

1.2 Hash-Counter Involution or HCI and MXCB or Modified-XCB

Now it is also more important to find out a secure candidate for the new class of enciphering scheme TIES. Without showing the existence, there is no practical value to have a new security notion. We propose Hash-Counter Involution or HCI which belongs to Hash-Ctr-Hash category. Moreover, structure wise it is similar and much simpler than XCB. We would like to note that XCB is not a TIES. It needs several changes to make it TIES. We have mainly done those changes in HCI which makes it more simpler, efficient and involution. We show that HCI is pri-secure. By using our generic method one can obtain a secure TES also. Note that in the generic construction, we need a completely different key than the key of secure TIES. But we propose a variant of the generic method as we are not willing to use any more extra key. We use a part of the key (poly-hash key) of HCI for the generic construction. We have similar quadratic security bound as we have in XCB, HCH etc. We term the construction obtained as described above as MXCB or Modified-XCB. This new construction is improved construction over XCB with respect to efficiency, design complexity, the secret key storage etc. We believe that MXCB is an important design in the light of the IEEE working group on storage security's current activity [9]. This group is working towards a standard for a wide block TES and this new construction no doubt is a strong candidate for this.

The paper is organized as follows. In section 2, we first provide a necessary background with all notations used in the paper. We define some of the security notions of symmetric key encryption which are relevant to this paper in the same section. In section 3, we introduce the new enciphering scheme TIES and a proper security notion for this class of enciphering scheme. This section also contains relationship between the new security notion with some of the known security notions and provides generic method to obtain sprp-secure construction from pri-secure. In section 4, we propose tweakable enciphering scheme HCI and we make pri-security analysis. In section 5, we propose an efficient sprp-secure MXCB and provides its security analysis. Finally we conclude with possible future research work.

2 Preliminaries

In this section, we provide some basic terminologies which are used in the paper. We also define several popular security notions used in symmetric key cryptography.

2.1 Notations and Definitions

In this paper, we fix an integer n which is the *block size* of the underlying block cipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$. For each block cipher key $K \in \{0,1\}^k$, $E_K(\cdot) := E(K,\cdot)$ is a permutation on $\{0,1\}^n$. The inverse permutation is denoted as $E_K^{-1}(\cdot)$ or $E_K^{-1}(\cdot)$.

Given any $X \in \{0,1\}^*$ we define the number of blocks of X as $||X|| = \lceil |X|/n \rceil$. For any non-negative integer s and a binary string $X = x_1 x_2 \cdots x_r \in \{0,1\}^*$, $x_i \in \{0,1\}$, we write

$$\begin{aligned} X[s] &= x_1 \cdots x_s \text{ if } s \leq r \\ &= X \parallel 0^{s-r} \text{ if } r < s \end{aligned}$$

Note that if $s = 0$, $X[s] = \lambda$, the empty string. XOR or $\oplus$ denotes the bitwise exclusive or for two strings of equal size. We can extend the definition for unequal strings as follows. Let $X \in \{0,1\}^s, Y \in \{0,1\}^r$ then we define $X \hat{\oplus} Y := X \oplus Y[s]$ and $X \check{\oplus} Y := X[r] \oplus Y$. If $|X| \leq n$ then we denote $\overline{X} := X[n]$.

Unless we mention, we use the notation $\rho \xleftarrow{\$} S$ to mean that ρ is chosen uniformly from the set S and it is independent from all previously described random variables. Note that the set S has to be finite[1]. For example, if we write $\boldsymbol{\rho}, \boldsymbol{\rho}' \xleftarrow{\$} S$ then it means that $\boldsymbol{\rho}$ and $\boldsymbol{\rho}'$ are independently and uniformly distributed over S. When we say $\boldsymbol{\rho}_\ell \xleftarrow{\$} S_\ell$, for $\ell \geq 2n$ we mean that $\boldsymbol{\rho}_\ell$'s are independently and uniformly distributed over the set S_ℓ. Now we define some useful sets used in the paper.

Message space: $\mathcal{M} = \{0,1\}^{\geq 2n} = \cup_{i \geq 2n} \{0,1\}^i$ called as *message space* or *ciphertext space*. The messages and ciphertexts of this paper are from this set.

Tweak space: $\mathcal{T} = \{0,1\}^{\ell_{\text{tw}}}$ called as *tweak space* for some $\ell_{\text{tw}} \geq 0$. We write $t := \lceil \ell_{\text{tw}}/n \rceil$ and called as the number of blocks of a tweak. Tweak actually corresponds to the associative data of a message or a ciphertext. For example, in disk encryption it can be the sector address of the data stored in a sector.

[1] It can be a continuous real interval, but here we are not interested in continuous real intervals.

Function space: $\mathrm{Func}(A, B)$ denotes the set of all functions from A to B. We also write the set of all tweakable functions from A to B as $\mathrm{TFunc}(A, B) := \mathrm{Func}(\mathcal{T} \times A, B)$. Given $f : \mathcal{T} \times A \to B$ and $T \in \mathcal{T}$, we denote $f^T(\cdot) := f(T, \cdot) : A \to B$. If $A = \{0,1\}^i, B = \{0,1\}^j$, we simply write $\mathrm{Func}(i, j)$ or $\mathrm{TFunc}(i, j)$ instead of $\mathrm{Func}(A, B)$ and $\mathrm{TFunc}(A, B)$ respectively.

Permutation space: $\mathrm{Perm}(i)$ is the set of all permutations on $\{0,1\}^i$, and the set of all tweakable permutations on $\{0,1\}^i$ is defined as $\mathrm{TPerm}(i) = \{\pi \in \mathrm{Func}^{\mathcal{T}}(i, i) : \pi^T \in \mathrm{Perm}(i)\ \forall T \in \mathcal{T}\}$. We denote the inverse permutation of π^T as π^{-T} or $\pi^{-1}(T, \cdot)$.

Involution space: A function π on $\mathcal{M}$ is called **involution** if for all $X \in \mathcal{M}$, $\pi(\pi(X)) = X$. It is easy to see that involution should be a permutation and $\pi = \pi^{-1}$. Let $\mathrm{Inv}(i) = \{\pi \in \mathrm{Perm}(i) : \pi = \pi^{-1}\}$ denote the set of all involutions. Similarly, we denote the set of all tweakable involutions by $\mathrm{TInv}(i) = \{\pi \in \mathrm{TFunc}(i, i) : \pi^T \in \mathrm{Inv}(i)\ \forall T \in \mathcal{T}\}$.

2.2 Security notions

A function $f : \mathcal{M} \to \mathcal{M}$ is called *length-preserving* if for all $x \in \mathcal{M}$, $|x| = |f(x)|$. We denote f_i for the restricted function f with domain $\{0,1\}^i$, for all $i \geq 2n$. Hence $f_i \in \mathrm{Func}(i, i)$. If π is a length-preserving permutation then $\pi_i \in \mathrm{Perm}(i)$. A tweakable length preserving or TLP function (or permutation) on $\mathcal{M}$ is $f : \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ where $f(T, \cdot)$ is length-preserving function (or permutation) on $\mathcal{M}$.

A keyed family $\{\mathbf{E}_K : K \in \mathcal{K}\}$ is called as a *tweakable enciphering scheme* or **TES** on $\mathcal{M}$ if for each key $K \in \mathcal{K}$, $\mathbf{E}_K : \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ is a tweakable length-preserving permutation. We denote the inverse of the above permutation as $\mathbf{E}_K^{-1}(T, \cdot)$ or $\mathbf{E}_K^{-T}(\cdot)$. We also denote the TES as $\mathbf{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ where $\mathbf{E}(K, T, M) = \mathbf{E}_K(T, M)$ and its inverse as $\mathbf{E}^{-1}$. $\mathbf{E}$ is also called as the encryption algorithm of TES and $\mathbf{E}^{-1}$ is called as the decryption algorithm. Now we define some ideal candidates of tweakable length-preserving functions or permutations.

TLP-URP: $\boldsymbol{\pi} = (\boldsymbol{\pi}_{2n}, \boldsymbol{\pi}_{2n+1}, \ldots) \xleftarrow{\$} \mathrm{TPerm}(\mathcal{M})$ is called as *tweakable length preserving uniform random permutation* or TLP-URP. More precisely, for each $\ell \geq 2n$, we have $\boldsymbol{\pi}_\ell \xleftarrow{\$} \mathrm{TPerm}(\ell)$. We call $\boldsymbol{\pi}$ as the *tweakable length preserving uniform random permutation*. $\boldsymbol{\pi}^{-1}$ corresponds to the collections of all inverses of $\boldsymbol{\pi}_\ell$, i.e., $\boldsymbol{\pi}^{-1} = (\boldsymbol{\pi}_{n+1}^{-1}, \boldsymbol{\pi}_{n+1}^{-1}, \ldots)$.

- Given a query (T, M) to $\boldsymbol{\pi}$ (or $\boldsymbol{\pi}^{-1}$) it returns $\boldsymbol{\pi}_\ell(T, M)$ (or $\boldsymbol{\pi}_\ell^{-1}(T, M)$ respectively) where $|M| = \ell$. If (T, M) is a new[2] query to $\boldsymbol{\pi}$ then the response $C \xleftarrow{\$} \{0,1\}^\ell \setminus R^T$ where R^T denotes the set of all previous responses of $\boldsymbol{\pi}$ when the tweak is T and $M \in \{0,1\}^\ell$. A similar probability distribution of responses for the inverse query holds. We have $M \xleftarrow{\$} \{0,1\}^\ell \setminus D^T$ where D^T denotes the set of all previous responses of $\boldsymbol{\pi}^{-1}$.

TLP-URF: $\boldsymbol{\rho} = (\boldsymbol{\rho}_{2n}, \boldsymbol{\rho}_{2n+1}, \ldots) \xleftarrow{\$} \mathrm{TFunc}(\mathcal{M})$ is called as *tweakable length preserving uniform random function* or TLP-URF. More precisely, for each $\ell \geq 2n$, we have $\boldsymbol{\rho}_\ell \xleftarrow{\$} \mathrm{TFunc}(\ell)$. Since it is a random function there is no inverse of it.

- Given a query (T, M) to $\boldsymbol{\rho}$ it returns $\boldsymbol{\rho}_\ell(T, M)$ where $|M| = \ell$. If (T, M) is a new query to $\boldsymbol{\rho}$ then the response $C \xleftarrow{\$} \{0,1\}^\ell$ where $M \in \{0,1\}^\ell$.

[2] it is not queried before

Let $\mathcal{A}$ be an oracle algorithm having access of two oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ where each oracles takes inputs from the set $\mathcal{T} \times \mathcal{M}$ and gives outputs from $\mathcal{M}$. The $\mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2} \Rightarrow 1$ means that $\mathcal{A}$ outputs 1 after interacting with $\mathcal{O}_1$ and $\mathcal{O}_2$. Similarly we define $\mathcal{A}^{\mathcal{O}_1} \Rightarrow 1$ when $\mathcal{A}$ has access of one oracle. We say that $\mathcal{A}$ is (q, σ, m)-*distinguisher* if it makes at most q queries having at most σ many blocks and the the number of blocks of the longest query is at most m. We say that $\mathcal{A}$ is (q, σ, m, t)-distinguisher if $\mathcal{A}$ is (q, σ, m)-distinguisher and moreover it runs in at most time t.

Now we define prf (or pseudo random function), weak-prf, prp (pseudo random permutation) and sprp (strong prp) security notions for a tweakable length-preserving enciphering scheme. Let $\mathbf{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ be a TES.

(1) prf :

$$\mathbf{Adv}^{\widetilde{\text{prf}}}_{\mathbf{E}}(\mathcal{A}) = \left|\Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathbf{E}_K} \Rightarrow 1\right] - \Pr\left[\boldsymbol{\rho} \xleftarrow{\$} \text{Func}^{\mathcal{T}}(\mathcal{M}) : \mathcal{A}^{\boldsymbol{\rho}} \Rightarrow 1\right]\right|. \tag{1}$$

$$\mathbf{Adv}^{\pm\widetilde{\text{prf}}}_{\mathbf{E}}(\mathcal{A}) = \left|\Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathbf{E}_K,\mathbf{E}_K^{-1}} \Rightarrow 1\right] - \Pr\left[\boldsymbol{\rho},\boldsymbol{\rho}' \xleftarrow{\$} \text{Func}^{\mathcal{T}}(\mathcal{M}) : \mathcal{A}^{\boldsymbol{\rho},\boldsymbol{\rho}'} \Rightarrow 1\right]\right|. \tag{2}$$

We define $\mathbf{Adv}^{\widetilde{\text{prf}}}_{\mathbf{E}}(q, \sigma, m)$ by $\max_{\mathcal{A}} \mathbf{Adv}^{\widetilde{\text{prf}}}_{\mathbf{E}}(\mathcal{A})$ where maximum is taken over all (q, σ, m)-distinguishers $\mathcal{A}$. Similarly we can define $\mathbf{Adv}^{\widetilde{\text{prf}}}_{\mathbf{E}}(q, \sigma, m, t)$, $\mathbf{Adv}^{\pm\widetilde{\text{prf}}}_{\mathbf{E}}(q, \sigma, m)$, $\mathbf{Adv}^{\pm\widetilde{\text{prf}}}_{\mathbf{E}}(q, \sigma, m, t)$. We define the *weak-prf advantage* of $\mathbf{E}$ as

$$\mathbf{Adv}^{\widetilde{\text{wprf}}}_{\mathbf{E}}(q, \sigma, m) = \max_{\mathcal{A}} \mathbf{Adv}^{\widetilde{\text{prf}}}_{\mathbf{E}}(\mathcal{A})$$

where maximum is taken over all (q, σ, m)-distinguishers $\mathcal{A}$ whose all queries are uniformly and independently distributed over $\{0,1\}^{\ell_i}$ where ℓ_i denotes the query-length for i^{th} query[3].

(2) (s)prp : We define the (s)prp-advantage of an adversary $\mathcal{A}$ for $\mathbf{E}$

$$\mathbf{Adv}^{\widetilde{\text{prp}}}_{\mathbf{E}}(\mathcal{A}) = \left|\Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathbf{E}_K} \Rightarrow 1\right] - \Pr\left[\boldsymbol{\pi} \xleftarrow{\$} \text{Perm}^{\mathcal{T}}(\mathcal{M}) : \mathcal{A}^{\boldsymbol{\pi}} \Rightarrow 1\right]\right| \tag{3}$$

$$\mathbf{Adv}^{\pm\widetilde{\text{prp}}}_{\mathbf{E}}(\mathcal{A}) = \left|\Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathbf{E}_K,\mathbf{E}_K^{-1}} \Rightarrow 1\right] - \Pr\left[\boldsymbol{\pi} \xleftarrow{\$} \text{Perm}^{\mathcal{T}}(\mathcal{M}) : \mathcal{A}^{\boldsymbol{\pi},\boldsymbol{\pi}^{-1}} \Rightarrow 1\right]\right| \tag{4}$$

We define $\mathbf{Adv}^{\widetilde{\text{prp}}}_{\mathbf{E}}(q, \sigma, m)$ by $\max_{\mathcal{A}} \mathbf{Adv}^{\widetilde{\text{prp}}}_{\mathbf{E}}(\mathcal{A})$ where maximum is taken over all (q, σ, m)-distinguishers $\mathcal{A}$. Similarly we can define $\mathbf{Adv}^{\widetilde{\text{prp}}}_{\mathbf{E}}(q, \sigma, m, t)$, $\mathbf{Adv}^{\pm\widetilde{\text{prp}}}_{\mathbf{E}}(q, \sigma, m)$, $\mathbf{Adv}^{\pm\widetilde{\text{prp}}}_{\mathbf{E}}(q, \sigma, m, t)$. When $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ is a block cipher, we define prp-advantage of $\mathcal{A}$ by

$$\mathbf{Adv}^{\text{prp}}_{E}(\mathcal{A}) = \left|\Pr\left[K \xleftarrow{\$} \{0,1\}^k : A^{E_K} \Rightarrow 1\right] - \Pr\left[\boldsymbol{\pi}_n \xleftarrow{\$} \text{Perm}(\{0,1\}^n) : A^{\boldsymbol{\pi}_n} \Rightarrow 1\right]\right|.$$

We similarly define $\mathbf{Adv}^{\text{prp}}_{E}(q, m)$ by $\max_{\mathcal{A}} \mathbf{Adv}^{\text{prp}}_{\mathbf{E}}(\mathcal{A})$ where maximum is taken over all (q, q, m)-distinguishers $\mathcal{A}$ (note that the number of blocks has to be equal with the number of queries). We have used $\pm$ to denote the strong security (secure even if the distinguisher has access of inverse query). The presence of tilde means it is in tweakable mode.

[3] We can not have an uniform distribution over $\mathcal{M}$ and hence we restrict the distribution on a query-length where the attacker can choose the query-length as well as the tweak.

Assumption Without loss of generality, we assume that the distinguisher $\mathcal{A}$ does not make any *pointless* queries. In other words, it does not repeat any encryption or decryption query and if it obtains Y as the response of encryption (or decryption) query with input (X, T) then it does not ask decryption (or encryption respectively) query with input (Y, T) where $T \in T$ is the tweak part of the query. These queries are called *pointless* as the adversary knows what it would get as responses for such queries.

It is well known that TRP-URP and TRP-URF are very close to each other. Let $\mathcal{A}$ be any (q, σ, m)-distinguisher which does not make any pointless queries then we have the following result. A proof of the similar statement can be found in [8].

Theorem 1. $\mathbf{Adv}_{\boldsymbol{\rho}}^{\pm\widetilde{\text{prp}}}(\mathcal{A}) = \mathbf{Adv}_{\boldsymbol{\pi}}^{\pm\widetilde{\text{prf}}}(\mathcal{A}) \leq \frac{q(q-1)}{2^{n+1}}$.

3 Involution Enciphering Scheme and PRI-Security Notion

In most of the applications, we need to implement encryption and decryption together. If we have a tweakable enciphering scheme $\mathbf{E}$ such that for each key K, and each tweak $T \in \mathcal{T}$, $\mathbf{E}_K(T, \cdot)$ is a length-preserving involution then $\mathbf{E}$ is also the decryption algorithm of itself. In this case, we call it as a tweakable involution enciphering scheme or TIES. A single implementation of $\mathbf{E}$ is sufficient for both encryption and decryption. Thus, TIES can have practical interest. Unfortunately, a TIES can not be PRP (pseudo random permutation). Since a distinguisher first makes an encryption query $\mathbf{E}_K(T, X) = Y$ and then asks another encryption query $\mathbf{E}_K(T, Y) = X'$. In case of a TIES $\mathbf{E}$, $X = X'$ with probability one but for a tweakable length-preserving uniform random permutation $\boldsymbol{\pi}$, $X = X'$ holds with negligible probability. So we need a new security notion for TIES. In this section, we first define a new security notion called as *pri*-security or *pseudo random involution security*. After defining the new notion we show relationship between this new notion and prf-security or wprf-security. Finally we propose a genric method to obtain a sprp-secure construction given a pri-secure construction.

By the notation $\boldsymbol{\tau} = (\boldsymbol{\tau}_{2n}, \boldsymbol{\tau}_{2n+1}, \ldots) \xleftarrow{\$} \text{TInv}(\mathcal{M})$, we mean that for each $\ell \geq 2n$, $\boldsymbol{\tau}_\ell \xleftarrow{\$} \text{TInv}(\ell)$. We call $\boldsymbol{\tau}$ as the TLP-URI or *tweakable length-preserving uniform random involution*. Thus we have so far defined three ideal candidates a TLP-URP $\boldsymbol{\pi}$, TLP-URF $\boldsymbol{\rho}$ and TLP-URI $\boldsymbol{\tau}$. $\boldsymbol{\tau}$ can be considered as an ideal candidate as we consider all possible TLP involution and then we choose one uniformly from the set. One can check the following probability distribution of the response of $\boldsymbol{\tau}$.

- Given a query (T, M) to $\boldsymbol{\tau}$ (or $\boldsymbol{\tau}^{-1}$) it returns $\boldsymbol{\tau}_\ell(T, M)$ (or $\boldsymbol{\tau}_\ell^{-1}(T, M)$ respectively) where $|M| = \ell$. If (T, M) is different from all previous queries and responses (including the tweak) then the response $C \xleftarrow{\$} \{0,1\}^\ell \setminus (R^T \cup D^T)$ where $M \in \{0,1\}^\ell$, R^T denotes the set of all previous responses of $\boldsymbol{\tau}$ with tweak T and D^T denotes the set of all previous queries with tweak T.

Definition 1. pseudo random involution security
Let $\{\mathbf{E}_K : K \in \mathcal{K}\}$ *be a keyed family of tweakable length-preserving involutions defined on* $\mathcal{M}$ *and* $\boldsymbol{\tau} \xleftarrow{\$} \text{TInv}(\mathcal{M})$. *We define the* PRI-advantage *or* pseudo random involution *of a distinguisher* $\mathcal{A}$ *for a* TIES $\mathbf{E}$ *as*

$$\mathbf{Adv}_{\mathbf{E}}^{\widetilde{\text{pri}}}(\mathcal{A}) = \left|\Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathbf{E}_K(.,.)} \Rightarrow 1\right] - \Pr\left[\boldsymbol{\tau} \xleftarrow{\$} \text{TInv}(\mathcal{M}) : \mathcal{A}^{\boldsymbol{\tau}(.,.)} \Rightarrow 1\right]\right| \tag{5}$$

We define $\mathbf{Adv}^{\widetilde{\text{pri}}}_{\mathbf{E}}(q,\sigma,m)$ *by* $\max_{\mathcal{A}}\mathbf{Adv}^{\widetilde{\text{pri}}}_{\mathbf{E}}(\mathcal{A})$ *where maximum is taken over all* (q,σ,m)*-distinguishers. For a computational advantage we define* $\mathbf{Adv}^{\widetilde{\text{pri}}}_{\mathbf{E}}(q,\sigma,m,t)$ *by* $\max_{\mathcal{A}}\mathbf{Adv}^{\widetilde{\text{pri}}}_{\mathbf{E}}(\mathcal{A})$ *where maximum is taken over all* (q,σ,m,t)*-distinguishers.*

Like other security notions, we compute the advantage at distinguishing a TIES $\mathbf{E}$ from the ideal candidate of TIES $\boldsymbol{\tau}$. A similar kind of treatment can be found in prf or prf security notions. Thus, it is the natural way to define a security notion for a TIES.

Now we prove some relationship between pri-security with prf-security and wprf-security. Then we provide a generic construction to make a sprp-secure from a given pri-secure.

3.1 pri-security ⇔ prf-security (for involution-allowed adversary)

For a similar reason of not being a prp, a TIES is also not a prf or pseudo random function. But if we restrict the adversary not to ask any query which is response of a previous query, then we can prove that prf-security is equivalent to prf-security. We call those adversary as involution-allowed adversary (or distinguisher).

Definition 2. *Let* $\mathcal{O}$ *be a tweakable length-preserving oracle defined on* $\mathcal{M}$ *with the tweak space* $\mathcal{T}$*. We say that an oracle algorithm* $\mathcal{A}^{\mathcal{O}}$ *is an* involution-allowed distinguisher *if* $(T^i,X^i)\neq(T^{i'},Y^{i'})$*, for all* $1\leq i'<i$*, where* (X^i,T^i) *is the* i^{th} *query and* Y^i *is the response.*

Proposition 1. *1. Let* $\boldsymbol{\rho}\xleftarrow{\$}\text{Func}^{\mathcal{T}}(\mathcal{M})$ *i.e.,* $\boldsymbol{\rho}_\ell\xleftarrow{\$}\text{TFunc}(\ell,\ell)$ *for all* $\ell\geq 2n$*. Then for any* (q,σ,m)*-involution-allowed distinguisher* $\mathcal{A}$*,* $\mathbf{Adv}^{\widetilde{\text{pri}}}_{\boldsymbol{\rho}}(\mathcal{A})=\mathbf{Adv}^{\widetilde{\text{prf}}}_{\boldsymbol{\tau}}(\mathcal{A})\leq q(q-1)/2^n$*.*

2. For any TIES $\mathbf{E}$ *and any* (q,σ,m)*-involution-allowed distinguisher* $\mathcal{A}$ *we have*

$$|\mathbf{Adv}^{\widetilde{\text{pri}}}_{\mathbf{E}}(\mathcal{A})-\mathbf{Adv}^{\widetilde{\text{prf}}}_{\mathbf{E}}(\mathcal{A})|\leq q(q-1)/2^n.$$

Proof. We only prove the first part since the second part is an immediate corollary of the first part by using triangle inequality. Note that for any distinguisher $\mathbf{Adv}^{\widetilde{\text{pri}}}_{\boldsymbol{\rho}}(\mathcal{A})=\mathbf{Adv}^{\widetilde{\text{prf}}}_{\boldsymbol{\tau}}(\mathcal{A})$ and equal to

$$\left|\Pr[\boldsymbol{\rho}\xleftarrow{\$}\text{TFunc}(\mathcal{M}):\mathcal{A}^{\boldsymbol{\rho}(\cdot,\cdot)}\Rightarrow 1]-\Pr[\boldsymbol{\tau}\xleftarrow{\$}\text{TInv}(\mathcal{M}):\mathcal{A}^{\boldsymbol{\tau}(\cdot,\cdot)}\Rightarrow 1]\right|$$

Now we prove that the abouve term is upper bounded by $q(q-1)/2^n$. The proof is very similar to prf-prp switching lemma [1]. Here we have pri instead of prp. We can equivalently define $\boldsymbol{\tau}$ as follows.

Initialization We initialize a tweakable partial function τ as empty. We denote the domain and range of τ^T as $Domain^T$ and $Range^T$ for all $T\in\mathcal{T}$. Initially these sets are empty sets. We have a flag *bad* initially it set as false.

Response Now on each query $(T,X)\in\{0,1\}^\ell$ we choose $Y\xleftarrow{\$}\{0,1\}^\ell$ for some $\ell\geq 2n$. If $Y\in Domain^T\cup Range^T$ then we set *bad* as true and we choose $Y\in\{0,1\}^\ell\setminus(Domain^T\cup Range^T)$. Y is the response of the query. Modify the domain and range by adding the relation $\tau^T(X)=Y$.

Note that if *bad* is false then the above game is nothing but equivalent to $\boldsymbol{\rho}$ otherwise it is equivalent to $\boldsymbol{\tau}$. Here we use the fact that $\mathcal{A}$ is an involution-allowed distinguisher. Thus, by using a game technique rule from [1]

$$\left|\Pr[\boldsymbol{\rho} \xleftarrow{\$} \mathrm{TFunc}(\mathcal{M}) : \mathcal{A}^{\boldsymbol{\rho}(.,.)} \Rightarrow 1] - \Pr[\boldsymbol{\tau} \xleftarrow{\$} \mathrm{TInv}(\mathcal{M}) : \mathcal{A}^{\boldsymbol{\tau}(.,.)} \Rightarrow 1]\right| \leq \Pr[bad = true].$$

Note that, $\Pr[bad = true]$ can be calculated in any of the two games. We calculate it when $\mathcal{A}$ is interacting with $\boldsymbol{\rho}$. After i^{th} query the size of $Domain^T \cup Range^T$ is at most $2i$ and hence $\Pr[bad = true] \leq \sum_{i=1}^{1}(2i-2)/2^n = q(q-1)/2^n$. Hence proved. ■

3.2 pri-security ⇒ wprf-security

Now we prove that any pri-secure TIES is also a wprf-secure. Intuitively, any adversary for weak-prf security is involution-allowed adversary with high probability since the queries are chosen uniformly and independently.

Proposition 2. *For any* TIES $\mathbf{E}$, $\mathbf{Adv}^{\widetilde{\text{wprf}}}_{\mathbf{E}}(q,\sigma,m) \leq \mathbf{Adv}^{\widetilde{\text{pri}}}_{\mathbf{E}}(q,\sigma,m) + 2q(q-1)/2^n$.

Proof. Since all inputs are chosen uniformly and independently, all queries are distinct and different from the responses is with probability at least $1 - q(q-1)/2^n$. In this case $\mathcal{A}$ is behaving like an involution-allowed adversary and hence by using the second part of the above proposition 1 we obtain our result. ■

3.3 A Generic Construction of SPRP-secure from PRI-secure

Let $\mathbf{E} : \{\mathbf{E}_K : K \in \mathcal{K}\}$ be a length-preserving tweakable involution. Now we choose $(K,h) \xleftarrow{\$} \mathcal{K} \times \{0,1\}^n$. Define a tweakable length-preserving enciphering scheme as

$$\overline{\mathbf{E}}_{K,h}(T,M) = \mathbf{E}_K(T,M) \leftarrow\!\!\oplus\ h, \quad \overline{\mathbf{E}}^{-1}_{K,h}(T,C) = \mathbf{E}^{-1}_K(T, C \leftarrow\!\!\oplus\ h).$$

Thus, $\overline{\mathbf{E}}$ is a simple modification over TIES $\mathbf{E}$. Now we prove that $\overline{\mathbf{E}}$ is SPRP-secure whenever $\mathbf{E}$ is PRI-secure.

Theorem 2. *For any* TIES $\mathbf{E}$ *and any* (q,σ,m)*-distinguisher* $\mathcal{A}$*, we have*

$$\mathbf{Adv}^{\pm\widetilde{\text{prp}}}_{\overline{\mathbf{E}}}(\mathcal{A}) \leq \mathbf{Adv}^{\widetilde{\text{pri}}}_{\mathbf{E}}(q,\sigma,m) + q(q-1)/2^n.$$

Proof. To prove it, we first consider $\boldsymbol{\tau} \xleftarrow{\$} \mathrm{TInv}(\mathcal{M})$ and $\overline{\boldsymbol{\tau}}_\ell(T,M) = \boldsymbol{\tau}_\ell(T, M \leftarrow\!\!\oplus\ h)$, for all $T \in \mathcal{T}, M \in \{0,1\}^\ell$, $\ell \geq 2n$. We first prove that $\overline{\boldsymbol{\tau}}$ is SPRP-secure. Then by using replacement argument we can prove that $\overline{\mathbf{E}}$ is SPRP-secure whenever $\mathbf{E}$ is PRI-secure.

We assume that there is no pointless queries by $\mathcal{A}$. Now suppose $((T^i,X^i),\mathsf{ty}^i) \in \{0,1\}^{\ell_i} \times \{\mathsf{enc},\mathsf{dec}\}$ is the i^{th} query and Y^i is the response where ty^i corresponds to the type of query encryption or decryption. We define bad as true if there is a collision among Z^i, W^i's where $Z^i = X^i$ and $W^i = Y^i \leftarrow\!\!\oplus\ h$ if $\mathsf{ty}^i = \mathsf{enc}$. If $\mathsf{ty}^i = \mathsf{dec}$ then $Z^i = Y^i$ and $W^i = X^i \leftarrow\!\!\oplus\ h$. Otherwise we set bad as false. Note that when $\mathcal{A}$ interacts with tweakable length-preserving uniform random permutation the bad is true with probability at most $q(q-1)/2^n$. If bad does not set true then the

two games $\mathcal{A}^{\overline{\boldsymbol{\tau}},\overline{\boldsymbol{\tau}}^{-1}}$ and $\mathcal{A}^{\boldsymbol{\pi},\boldsymbol{\pi}^{-1}}$ are equivalent. One can write it more formally by using game-playing technique [1]. Hence $\mathbf{Adv}_{\overline{\boldsymbol{\tau}}}^{\pm\widetilde{\mathrm{prp}}}(\mathcal{A}) \leq \frac{q(q-1)}{2^n}$. Now by using replacement argument we can prove that for any TIES $\mathbf{E}$, and for any distinguisher $\mathcal{A}$ making at most q queries

$$\mathbf{Adv}_{\overline{\mathbf{E}}}^{\pm\widetilde{\mathrm{prp}}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathbf{E}}^{\mathrm{pri}}(q) + \frac{q(q-1)}{2^n}. \tag{6}$$

■

Remark 1. We have shown a method to obtain a sprp-secure TES from a given pri-secure TIES. Moreover, the construction $\overline{\mathbf{E}}$ is obtained by simply making one xor with a key. Thus, in terms of efficiency point of view, $\overline{\mathbf{E}}$ has exactly same performance as in $\mathbf{E}$. Since $\mathbf{E}$ is an involution, a good performance can be found whenever both encryption and decryption are concerned together.

4 HCI or Hash-Counter Involution : A Construction of PRI-secure TIES

We first define a TIES called as HCI or hash-counter involution. We denote it as $\mathbf{E}_{\mathrm{HCI}}$ or simply $\mathbf{E}$. It has $\mathcal{M}$ as a message space, $\mathcal{T}$ as a tweak space and $\{0,1\}^k \times \{0,1\}^n$ as a key space where k is the key size of the underlying block cipher E. Now we define two functions, poly-hash and counter function. Note that the poly hash function defined here is different from the poly hash used in the definition of HCTR, HCH or XCB. The main advantage of this new definition is that the information of length of the message is not needed. In this paper we fix an irreducible polynomial of degree n and hence we have a field multiplication $*$ defined over $\{0,1\}^n$.

(1) poly-hash : For each $h \in \{0,1\}^n$, $\mathcal{H}_h^{\mathrm{poly}} : \{0,1\}^* \to \{0,1\}^n$ and it is defined as

$$\mathcal{H}_h^{\mathrm{poly}}(X_1, \cdots, X_m) = \overline{X}_m * h \oplus X_{m-1} * h^2 \oplus \cdots \oplus X_1 * h^m \oplus h^{m+1}$$

where $|X_1| = \cdots = |X_{m-1}| = n, |X_m| \leq n$. h is known as the poly-hash key. By using fundamental theorem of algebra, we can say that for any fixed $X \neq X' \in \{0,1\}^*$ and $a \in \{0,1\}^n$,

$$\Pr[h \xleftarrow{\$} \{0,1\}^n : \mathcal{H}_h^{\mathrm{poly}}(X) \oplus \mathcal{H}_h^{\mathrm{poly}}(X') = a] \leq \frac{d}{2^n}$$

where $d = \max\{||X||, ||X'||\} + 1$. Moreover, if X, X' and a are random variables and h is chosen independent with X, X' and a, then we also have the same bound. One can prove it by conditioning on X, X' and a. By using Horner's rule, we need only m sequential multiplications for a m-block input X.

(2) counter : Let $X_1 \parallel \cdots \parallel X_m \in \{0,1\}^{(m-1)n+s}$, $|X_1| = \cdots = |X_{m-1}| = n, 1 \leq |X_m| := s \leq n$ and $S \in \{0,1\}^n$. Now for any permutation $\pi : \{0,1\}^n \to \{0,1\}^n$ we define

$$\mathrm{ctr}_\pi^S(X) = Y_1 \parallel \cdots \parallel Y_m$$

where $Y_i = \pi(S+i-1) \oplus X_i, 1 \leq i \leq m$ and $+$ denote 2^n modulo addition. These computations are parallel in nature and moreover, if the computation of π has several rounds (e.g., AES has 10 rounds) then one can have pipeline implementation in hardware. A detail on hardware implementation can be found in [19].

In Figure 1 a pseudo code of HCI is given. It is easy to see that for any tweak T, $\mathbf{E}_{K,h}^T$ is an involution on $\mathcal{M}$. Moreover, the structure is similar and much more simpler than that of XCB. By using the generic extension stated in Section 3.3, we can get a Tweakable Enciphering Scheme $\overline{\mathrm{HCI}}$ based on HCI. Now we give a useful discussion about this new hash-counter based construction HCI.

Key-Generation : $(K, h) \xleftarrow{\$} \{0,1\}^k \times \{0,1\}^n$.

Involution $\mathbf{E}_{K,h}$: Input $(T, X) \in \mathcal{T} \times \mathcal{M}$

step-1 $U \leftarrow E_K(X_0)$;
step-2 $S \leftarrow U \oplus \mathcal{H}_h^{\text{poly}}(T \parallel X_1 \parallel \cdots \parallel X_m)$;
step-3 $(Y_1, \cdots Y_m) \leftarrow \text{ctr}_{E_K}^S(X_1 \parallel \cdots \parallel X_m)$;
step-4 $V \leftarrow S \oplus \mathcal{H}_h^{\text{poly}}(T \parallel Y_1 \parallel \cdots \parallel Y_m)$;
step-5 $Y_0 \leftarrow E_K^{-1}(V)$;
step-6 return $Y := Y_0 \parallel \cdots \parallel Y_m$;

Fig. 1. Here, we write $X = X_0 \parallel X_1 \parallel \cdots \parallel X_m$ where $|X_i| = n$, $0 \le i \le m-1$, $0 < |X_m| < n$ and $m \ge 2$.

A Useful Discussion on HCI :

Efficiency Comparison In hardware implementation, we can make step-1, step-2 in parallel and step-3, step-4 in parallel. A simple calculation shows that we need almost $(m+1)+(m+10)+11 = 2m+22$ clock cycles for a complete invocation of HCI with $m+1$ block messages and one block tweak. In [19], it was shown that HCTR needs 87 clock cycles for 32 block messages and one block tweak. Thus, HCI is performing similar to HCTR, one of the efficient constructions among TES [19]. Moreover, it has quadratic security (see Theorem 3) unlike HCTR which has cubic bound so far. Clearly, this is better construction than XCB in all respects.

1. It needs only two secret keys of size $n+k$ and we do not need any pre-computation. XCB either need to store four keys or need to compute these keys online which costs four extra block cipher invocations. It also costs four key-scheduling algorithms.
2. The two invocations of hash functions are different and it needs length of the message as a part of the input. In case of MXCB the hash functions are same and it takes only messages and tweak as input.
3. Moreover, it is an involution whereas XCB is not (this is mainly because of different choices of block cipher keys). Thus a single implementation would suffice to have both encryption and decryption. Thus, our new proposal is much more simpler variant of XCB in terms of key-storage, efficiency and design complexity.

sprp-security One can object the statement that HCI is an involution where XCB is not as HCI is not sprp-secure. But in the last section we propose a generic method to obtain sprp-secure $\overline{\text{HCI}}$ which costs only one extra XOR with a new key. Thus, $\overline{\text{HCI}}$ is sprp-secure and it is one xor away from involution. In the next section we also show that there is no harm to choose the new key as the poly-hash key of HCI. As a result, there is no need to choose extra key and we have a sprp-secure construction based on only two keys. We call the construction as MXCB or Modified-XCB.

message space Like XCB, we need to assume that the message size is at least $2n$ otherwise it is easy to find a distinguishing attack (for MXCB and XCB both). In [15], it is mentioned that the messages of length between n and $2n$ can be taken care by using a not-repeating tweak (like nonce).

But we can use much simpler construction such as OCB [**?**] and the message size can be less than n in the presence of nonce. Thus, the new construction is more meaningful when the message size is at least $2n$.

4.1 PRI-security analysis of HCI

In this section we provide a pri-security bound for HCI. Suppose after making q encryption queries, the involution-allowed (q, σ, m)-distinguisher $\mathcal{A}$ obtains the following messages, ciphertexts and tweaks for $1 \le i \le q$.

- i^{th} message : $M^i = M_0^i \parallel M_1^i \parallel \cdots \parallel M_{m_i}^i$ where $|M_0^i| = \cdots = |M_{m_i-1}^i| = n$, $1 \le |M_{m_i}^i| = s_i \le n$. We write $N^i = M_1^i \parallel \cdots \parallel M_{m_i}^i$.
- i^{th} ciphertext : $C^i = C_0^i \parallel C_1^i \parallel \cdots \parallel C_{m_i}^i$ where $|C_0^i| = \cdots = |C_{m_i-1}^i| = n$, $|C_{m_i}^i| = s_i$. We write $D^i = C_1^i \parallel \cdots \parallel C_{m_i}^i$.
- i^{th} tweak : $T^i \in \mathcal{T} = \{0,1\}^{\ell_{tw}}$ and we write $t = \lceil \ell_{tw}/n \rceil$.

Now, $\sum_s m_s \le \sigma$, and $\max_s m_s \le m$. It is easy to see that $\sum_{1 \le s < s' \le q} (m^s + m^{s'}) \le (q-1)\sigma$ and $\sum_{1 \le s < s' \le q} (m^s + m^{s'})^2 \le 2m^2q^2/2^n$. Let $\pi \xleftarrow{\$} \text{Perm}(n)$ and $\mathbf{E}_{\pi,h}$ denotes the HCI where block cipher E_K is replaced by the uniform random permutation π. Since $\mathcal{A}$ is involution-allowed distinguisher, $(T^i, M^i) \ne (T^j, C^j)$ for any $1 \le j < i \le q$. We denote the intermediate variables are U^i (U in step-1), S^i (S in step-2) and V^i (V in step-4) for $1 \le i \le q$. Now we list all inputs of the underlying block cipher E_K as $M_0^i,\ C_0^i,\ S^i + j,\ \ 0 \le j \le m_i - 1, 1 \le i \le q$ and the corresponding outputs are $U^i,\ V^i,\ M_j^i \oplus C_j^i,\ \ 1 \le j \le m_i, 1 \le i \le q$. We equivalently describe the response of $\mathbf{E}_{\pi,h}$ in Figure 1.

Intuitively, we first choose a random string R^i for a possible response. Then we make intermediate calculation and if bad event does not occur then we return the random string R^i. Otherwise we modify the string R^i such that it exactly simulates $\mathbf{E}_{\pi,h}$. Thus, if bad is false then for any distinguisher involution-allowed $\mathcal{A}$ the response distribution of $\mathbf{E}_{\pi,h}$ and $\boldsymbol{\rho}$ are identical. Thus, $\widetilde{\mathbf{Adv}}_{\mathbf{E}}^{\text{prf}}(\mathcal{A}) \le \Pr[\mathsf{bad} = T]$ where the probability is computed when $\mathcal{A}$ is interacting with $\boldsymbol{\rho}$.

Now it is easy to see that $\mathsf{bad} = T$ if and only if either there is a collision among the set of inputs or among the set of outputs except the collision of the form $M_0^i = M_0^{i'}$ or $M_0^i = C_0^{i'}$ with $1 \le i' < i$ and collision of its corresponding output. Since we compute the probability when $\mathcal{A}$ is interacting with $\boldsymbol{\rho}$ the probability distributions of C^i's are uniform and independent to the probability distribution of all previous queries including M^i. Moreover h is independent to all query-responses since it is not used in the computation of queries. The inputs and outputs of π are computed from M^i, C^i and h (even if these are not actual inputs in $\boldsymbol{\rho}$-game but it would be the actual inputs and outputs in $\mathbf{E}_{\pi,h}$).

- We first compute the probability that $S^i + j = M_0^{i'}$ for some $1 \le i, i' \le q$, $0 \le j \le m_i - 1$. In other words, $\mathcal{H}_h^{\text{poly}}(N^i) = U^i \oplus (M_0^{i'} - j) = a$ (say). If we fix the block cipher key, the value of U^i is also fixed and hence a is fixed. If M^i's are independently distributed with h then by using fundamental theorem of algebra, the above collision probability is upper bounded by $(m_i + t)/2^n$ (note that t is the number of blocks of tweaks). Thus,

$$\Pr[S^i + j = M_0^{i'},\ \ 1 \le i, i' \le q, 0 \le j \le m_i - 1] \le \frac{q(\sigma + qt)}{2^n}. \tag{7}$$

Initialization : $h \xleftarrow{\$} \{0,1\}^n$ and initialize $\mathsf{bad} = F$, a partial function f with domain Dom and range Ran. Initialize $Dom = Dom' = Ran = \emptyset$.

Response of $\mathbf{E}_{\pi,h}$: On query $(T^i, M^i) \in \mathcal{T} \times \mathcal{M}$

001 $R = (R^i_0, \cdots, R^i_{m_i}) \xleftarrow{\$} \{0,1\}^{n(m_i+1)}$;
002 **If** $M^i_0 \in Dom'$ then $U^i \leftarrow f(M^i_0)$, else if $M^i_0 \in Dom$ then $U^i \leftarrow f(M^i_0)$ and $\mathsf{bad} = T$;
003 **Else**
004 $U^i \xleftarrow{\$} \{0,1\}^n$, if $U^i \in Ran$ then $\mathsf{bad} = T$ and $U^i \xleftarrow{\$} \{0,1\}^n \setminus Ran$,
$f(M^i_0) \leftarrow U^i$, $Dom' = Dom' \cup \{M^i_0\}$;
005 **Endif**
006 $S^i = U^i \oplus \mathcal{H}^{\mathrm{poly}}_h(T \parallel M^i)$;
007 **For** $j = 0$ to $m_i - 1$
008 **If** $R^i_{j+1} \oplus M^i_{j+1} \in Ran$ then $\mathsf{bad} = T$ and $R^i_{j+1} \xleftarrow{\$} \{0,1\}^n \setminus Ran$;
009 **If** $S^i + j \in Dom$ then $\mathsf{bad} = T$ and $R^i_{j+1} = f(S^i + j) \oplus M^i_j$;
Else $f(S^i + j) \leftarrow R^i_{j+1} \oplus M^i_{j+1}$
010 **EndFor**
011 $V^i \leftarrow S^i \oplus \mathcal{H}^{\mathrm{poly}}_h(T \parallel R^i_1 \parallel \cdots \parallel R^i_{m_i-1} \parallel R^i_{m_i}[|M^i|])$;
012 **If** $V^i \in Ran$ then $\mathsf{bad} = T$ and $R^i_0 \leftarrow f^{-1}(V^i)$;
013 **Else If** $R^i_0 \in Dom$ then $\mathsf{bad} = T$ and $R^i_0 \xleftarrow{\$} \{0,1\}^n \setminus Dom$;
014 **Endif**
$f(R^0_i) \leftarrow V^i$ and $Dom' = Dom' \cup \{R^i_0\}$;
015 return $C^i = R^i[|M^i|]$;

Fig. 2. The statements within the boxes are executed when we consider the response of $\mathbf{E}_{\pi,h}$. The game without the box statement is equivalent to random game $\boldsymbol{\rho}$. The function f is defined online like a uniform random permutation. The set Dom is the set of all inputs of π (and hence f) so far and Dom' corresponds to mainly inputs of the form M^i_0 and C^i_0.

A similar argument shows that

$$\Pr[S^i + j = C^{i'}_0, \quad 1 \le i, i' \le q, 0 \le j \le m_i - 1] \le \frac{q(\sigma + qt)}{2^n}. \tag{8}$$

– Now we compute the probability of collision of the form $S^i + j = S^{i'} + j'$ with $(i, j) \ne (i', j')$. Obviously, $i \ne i'$ and we can write all these collision as $S^i - S^{i'} \in [-m_i + 1, m_{i'} + 1], 1 \le i < i' \le q$. For each $i < i'$, the probability that $S^i - S^{i'} \in [-m_i + 1, m_{i'} - 1]$ is less than $(m_i + m_{i'} + t)(m_i + m_{i'} - 1)/2^n$ since the degree of the polynomial is at most $m_i + m_{i'} + t$. Summing over all distinct pairs $(i.i')$ we have

$$\Pr[S^i + j = S^{i'} + j', \quad 1 \le i, i' \le q, 0 \le j \le m_i - 1] \le \frac{t(q-1)\sigma + 2m^2q^2}{2^n}. \tag{9}$$

– Since C^i_0 is chosen uniformly and independently with $M^{i'}_0$ for $i' \le i$, we have $\Pr[C^i_0 = M^{i'}_0] = 1/2^n$. Similarly, $\Pr[C^i_0 = C^{i'}_0] = 1/2^n$.

$$\Pr[C^i_0 = M^{i'}_0 \text{ or } C^i_0 = C^{i'}_0 \quad 1 \le i' \le i \le q] \le \frac{q^2}{2^n}. \tag{10}$$

– Till now we have computed probabilities for collision among inputs or π. Now we compute the collision probability for the ranges of π. We first compute the probability of collision among $M^i_j \oplus R^i_j$ and U^i's, $1 \le j \le m_i, 1 \le i \le q$. Since R^i_j's and U^i's (for new M^i_0, i.e., if it does not appear before as $M^{i'}_0$ or $C^{i'}_0$ for some $i' < i$) are uniformly and independently distributed. Thus,

$$\Pr[U^i = U^{i'} \text{ or } U^i = V^{i'} \wedge \mathsf{bad} = T] \le \frac{\sigma^2}{2^n}. \tag{11}$$

$$\Pr[M^i_j \oplus R^i_j = M^{i'}_{j'} \oplus R^{i'}_{j'}, \quad 1 \le i \le i' \le q, 1 \le j \le m_i - 1, 1 \le j' \le m_{i'} - 1] \le \frac{\sigma^2}{2^{n+1}}. \tag{12}$$

$$\Pr[M^i_j \oplus R^i_j = U^{i'} \text{ or } V^{i'}, \quad 1 \le i, i' \le q, 1 \le j \le m_i - 1] \le \frac{2q\sigma}{2^n}. \tag{13}$$

The eq 13 follows from the fact U^i is independent with R^i_j and $V^{i'} = U^{i'} \oplus \mathcal{H}^{\mathrm{poly}}_h(N^{i'})$.

– Now, for $i \neq i'$, we compute $\Pr[V^i = V^{i'}]$. $V^i = V^{i'}$ holds if and only if $\mathcal{H}^{\mathrm{poly}}_h(N^i) \oplus \mathcal{H}^{\mathrm{poly}}_h(N^{i'}) \oplus \mathcal{H}^{\mathrm{poly}}_h(D^i) \oplus \mathcal{H}^{\mathrm{poly}}_h(D^{i'}) = U^i \oplus U^{i'}$. By linearity we have $h^d \oplus \mathcal{H}^{\mathrm{poly}}_h(N^i \oplus N^{i'} \oplus D^i \oplus D^{i'}) = U^i \oplus U^{i'}$ where d the largest degree between $\mathcal{H}^{\mathrm{poly}}_h(N^i)$ and $\mathcal{H}^{\mathrm{poly}}_h(N^i)$ (recall that $\mathcal{H}^{\mathrm{poly}}_h$ is a monic polynomial). Now we assume that $N^i_1 \oplus D^i_1 = (R^i_1 \oplus M^i_1)$'s are distinct. Condition on the event $N^i \oplus N^{i'} \oplus D^i \oplus D^{i'}$ is non-zero string and hence the above conditional probability $\Pr[V^i = V^{i'}\ i \neq i' | (R^i_1 \oplus M^i_1)$'s are distinct] is bounded by $\frac{(m+t)q^2}{2^n}$. Note the probability that $(R^i_1 \oplus M^i_1)$'s are not distinct is taken take care in Eq 12.

Combining all these probability bound we obtain that $\Pr[\mathsf{bad} = T] \le \frac{10q^2m(m+t)}{2^n}$. We can summarize the above discussion into the following main theorem of the section.

Theorem 3.

1. $\mathbf{Adv}^{\pm\widetilde{\mathrm{prf}}}_{\mathbf{E}_{\pi,h}}(\mathcal{A}) \le \frac{10q^2m(m+t)}{2^n}$ *where* $\mathcal{A}$ *is any* (q, σ, m)*-involution allowed distinguisher.*
2. $\mathbf{Adv}^{\mathrm{pri}}_{\mathbf{E}_{\pi,h}}(\mathcal{A}) \le \frac{11q^2m(m+t)}{2^n}$ *where* $\mathcal{A}$ *is any* (q, σ, m)*-distinguisher.*
3. $\mathbf{Adv}^{\mathrm{pri}}_{\mathrm{HCI}}(q, \sigma, m, t) \le \mathbf{Adv}^{\pm\mathrm{prp}}_{E}(\sigma+q, t') + \frac{11q^2m(m+t)}{2^n}$ *where* $t' = t + \mathrm{O}(\sigma)$ *and* E *is the underlying block cipher.*

5 Modified-XCB or MXCB, an efficient strong pseudo random permutation

In this section we propose a new hash-counter-hash construction called as MXCB or modified-XCB. This is a variant of $\overline{\mathrm{HCI}}$. More precisely, we use poly hash key to xor firsi block output of HCI. Now we define more formally modified-XCB for each tweak $T \in \mathcal{T}$, The secret key for MXCB is chosen as $(K, h) \xleftarrow{\$} \{0,1\}^k \times \{0,1\}^n$. Let $X = X_0 \parallel X_1 \parallel \cdots \parallel X_m \in \mathcal{M}$ and $\mathsf{ty} \in \{\mathsf{enc}, \mathsf{dec}\}$. We denote $\mathbf{E}_{\mathrm{HCI}}$ for HCI enciphering scheme with poly hash key h and block cipher E_K.

encryption
if $(\mathsf{ty} = \mathsf{enc})$ then
$Y_0 \parallel Y_1 \parallel \cdots \parallel Y_m \leftarrow \mathbf{E}_{\mathrm{HCI}}(T, X)$;
return $(Y_0 \oplus h) \parallel Y_1 \parallel \cdots \parallel Y_m$;

decryption
if (ty = dec) then
$Y_0 \parallel Y_1 \parallel \cdots \parallel Y_m \leftarrow \mathbf{E}_{\mathrm{HCI}}(T, X_0 \oplus h, X_1, \cdots, X_m)$;
return $Y_0 \parallel Y_1 \parallel \cdots \parallel Y_m$;

Now we have the following result.

Theorem 4. $\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\mathrm{MXCB}_{\pi,h}}(\mathcal{A}) \leq m^2q^2/2^n$ *where* $\mathcal{A}$ *is any* (q, σ, m)*-distinguisher and hence*

$$\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\mathrm{MXCB}_{\pi,h}}(q, \sigma, m) \leq \frac{12q^2m(m+t)}{2^n}$$

$$\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\mathrm{MXCB}_{K,h}}(q, \sigma, m, t) \leq \mathbf{Adv}^{\pm\mathrm{prp}}_{E}(\sigma + q, t') + \frac{12q^2m(m+t)}{2^n}$$

where $t' = t + \mathrm{O}(\sigma)$.

Proof. The above result can be proved similar to HCI. We define bad event is true if it is true in HCI game or $h \oplus R^i_0 \in Dom$ for encryption query or $h \oplus M^i_0$ for decryption query. Thus, we have to the probability $q^2/2^n$ with the bad event probability for HCI game. Since we are interested in SPRP-security we need to add $q^2/2^{2n}$ probability (see Theorem 1) which is the distinguishing advantage between uniform random function and uniform random permutation game. Thus we have proved the theorem. ■

6 Conclusion

We introduce a new security notion PRI for TES whose encryption and decryption algorithm are same. This security notion is important in both practical and theoretical point of view. Since encryption and decryption are same, it is sufficient to implement one algorithm. Moreover, there are some relationships between this new security notion with known security notions. Since PRI can not be SPRP, we provide a generic method to obtain SPRP-secure. We also provide two new candidate HCI for PRI-security and MXCB for SPRP-security. We believe that these construction are important as the performance is similar to the one of the best performer.

References

1. Mihir Bellare and Phillip Rogaway. Code-Based Game-Playing Proofs and the Security of Triple Encryption. Cryptology ePrint Archive, Report 2004/331.
2. D. Chakraborty and P. Sarkar. A new mode of encryption providing a strong tweakable pseudo-random permutation. Fast Software Encryption FSE 2006, LNCS vol. 4047, Springer, pp. 293309, 2006.
3. D. Chakraborty and P. Sarkar. HCH: A new tweakable enciphering scheme using the Hash- Encrypt-Hash approach. Advances in Cryptology INDOCRYPT 2006, LNCS vol. 4329, Springer, pp. 287302, 2006.
4. Joan Daemen and Vincent Rijmen The Design of Rijndael: AES The Advanced Encryption Standard. Springer 2002. http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael-ammended.pdf
5. S. Halevi. EME*: Extending EME to handle arbitrary-length messages with associated data. Advances in Cryptology INDOCRYPT 2004, LNCS vol. 3348, Springer, pp. 315327, 2004.
6. S. Halevi. TET: A wide-block tweakable mode based on Naor-Reingold. Advances in Cryptology 2007. Cryptology ePrint archive, report 20007/14, 2007.

7. S. Halevi and P. Rogaway. A parallelizable enciphering mode. Topics in Cryptology CT-RSA 2004, LNCS vol. 2964, Springer, pp. 292304, 2004.
8. S. Halevi and P. Rogaway. A tweakable enciphering mode. Advances in Cryptology CRYPTO 2003, LNCS vol. 2729, Springer, pp. 482499, 2003.
9. IEEE Security in Storage Working Group (SISWG). PRP Modes Comparison IEEE P1619.2, March, 2007, IEEE Computer Society, Available at: http://siswg.org/
10. C. Jutla. Encryption modes with almost free message integrity. Advances in Cryptology EUROCRYPT 2001. Lecture Notes in Computer Science, vol. 2045, Springer-Verlag, 2001.
11. A. Karatsuba and Y. Ofman. Multiplication of multidigit numbers by automata. Soviet Physics-Doklady, 7:595596, 1963.
12. M. Liskov, R. Rivest, and D. Wagner. Tweakable block ciphers. Advances in Cryptology CRYPTO 2002, LNCS vol. 2442, Springer, pp. 3146, 2002.
13. M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. SIAM Journal of Computing, vol. 17, no. 2, pp. 373386, 1988.
14. S. Lucks. Faster Luby-Rackoff ciphers. Fast Software Encryption 1996, LNCS vol. 1039, Springer, pp. 189203, 1996.
15. D. McGrew and S. Fluhrer. The extended codebook (XCB) mode of operation. Cryptology ePrint archive, report 2007/298. To appear in Proceedings in Selected Areas in Cryptography.
16. McGrew and S. Fluhrer, The Extended Codebook (XCB) Mode of Operation, Cryptology ePrint Archive: Report 2004/278, October 25, 2004. http://eprint.iacr.org/2004/278
17. Moni Naor and Omer Reingold. A pseudo-random encryption mode. Manuscript available from www.wisdom.weizmann.ac.il/naor.
18. M. Naor and O. Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revisited. Journal of Cryptology, vol. 12, no. 1, pp. 2966, 1999.
19. Cuauhtemoc Mancillas-López and Debrup Chakraborty and Francisco Rodríguez-Henríquez. Efficient Implementations of Some Tweakable Enciphering Schemes in Reconfigurable Hardware. INDOCRYPT-2007, Lecture Notes in Computer Science, vol 4859, 2007.
20. P. Rogaway. Authenticated-encryption with associated-data. Ninth ACM Conference on Computer and Communications Security (CCS-9). ACM Press, 2002.
21. P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In ACM Conference on Computer and Communications Security (CCS-8). ACM Press, 196205.
22. Palash Sarkar. Improving Upon the TET Mode of Operation. Cryptology ePrint archive, report 2007/317.
23. Victor Shoup. On fast and provably secure message authentication based on universal hashing. CRYPTO-1996, volume 1109, Lecture Notes in Computer Science, pages 313328. Springer, 1996.
24. P. Wang, D. Feng, and W. Wu. HCTR: a variable-input-length enciphering mode. Information Security and Cryptography, CISC 2005, LNCS vol. 3822, Springer, pp. 175188, 2005.