An Improved Security Bound for HCTR

Debrup Chakraborty and Mridul Nandi

Department of Computer Science
CINVESTAV-IPN
Mexico City, Mexico
email: debrup@cs.cinvestav.mx, mridul.nandi@gmail.com

Abstract. HCTR was proposed by Wang, Feng and Wu in 2005. It is a mode of operation which provides a tweakable strong pseudorandom permutation. Though HCTR is quite an efficient mode, the authors showed a cubic security bound for HCTR which makes it unsuitable for applications where tweakable strong pseudorandom permutations are required. In this paper we show that HCTR has a better security bound than what the authors showed. We prove that the distinguishing advantage of an adversary in distinguishing HCTR and its inverse from a random permutation and its inverse is bounded above by $4.5\sigma^2/2^n$, where n is the block-length of the block-cipher and σ is the number of n-block queries made by the adversary (including the tweak).

1 Introduction

A block-cipher mode of operation is a specific way to use a block-cipher to encrypt messages longer than the block-length of the block-cipher. In the literature there are different modes of operations which provide different kinds of security services like confidentiality, authentication etc. A tweakable enciphering scheme (TES) is a specific kind of mode of operation. They are based on the notion of tweakable block ciphers introduced in [9]. TES are length preserving encryption schemes which can encrypt variable length messages. The security that these modes provide is that of a strong pseudorandom permutation (SPRP), i.e., a TES is considered secure if it is infeasible for any computationally bounded chosen plaintext chosen ciphertext adversary to distinguish between the TES and a random permutation. A TES takes as input a quantity called a tweak other than the message and the key. The tweak is supposed to be a public quantity which enriches the variability of the cipher-text produced.

The first construction of a wide block SPRP was provided by Naor and Reingold [14], but their construction was not a TES as the concept of tweaks came after their construction. A fully defined TES for arbitrary length messages using a block cipher was first presented in [6]. In [6] it was also stated that a possible application area for such encryption schemes could be low level disc encryption, where the encryption/decryption algorithm resides on the disc controller which has access to the disc sectors but has no access to the high level partitions of the disc like directories files, etc. The disc controller encrypts a message before writing it to a specific sector and decrypts the message after reading it from the sector. Additionally it was suggested in [6] that sector addresses can be used as tweaks. Because of the specific nature of this application, a length preserving enciphering scheme is required and under this scenario, a strong pseudorandom permutation can provide the highest possible security.

In the last few years there have been numerous proposals for TES. These proposals fall in three basic categories: Encrypt-Mask-Encrypt type, Hash-ECB-Hash type and Hash-Counter-Hash type. CMC [6], EME [7] and EME* [4] fall under the Encrypt-Mask-Encrypt group. PEP [3], TET [5] and HEH[15] fall under the Hash-ECB-Hash type and XCB [11], HCTR [17], HCH, HCHfp [2], ABL [12] fall under the Hash-Counter-Hash type.

The Encrypt-Mask-Encrypt type constructions require two layers of encryption with a light weight masking layer in between. The other two paradigms require a single layer of encryption sandwiched between two universal hash layers. Thus, the only significant cost for Encrypt-Mask-Encrypt type constructions are the block-cipher calls, whereas for the other two paradigms both block-cipher calls and finite field multiplications are required. More specifically, the Encrypt-Mask-Encrypt paradigm uses about $2m$ block cipher calls for encrypting a m block message and a the other two paradigms require m block-cipher calls and $2m$ field multiplications. Recently, in [16] many different constructions for the Hash-Encrypt-Hash paradigm were proposed using blockwise universal hash functions. One of these proposals called HEH[BRW] uses m field multiplications unlike other constructions of the same category. A detailed comparison of different TES can be found in [2, 5, 15].

In a recent study [10], some performance data regarding various tweakable enciphering schemes in reconfigurable hardware was reported. This study and the comparisons presented in [2, 5, 15] indicate that HCTR is one of the most efficient candidates among all proposed TES. But, the security guarantee that the designers of HCTR claimed is insufficient in many practical scenarios. This makes HCTR an uninteresting candidate.

In this paper we show that HCTR provides better security than that claimed by the authors. In fact HCTR provides the same security as other other proposed TES. We consider this result to be 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 [8].

The crux of this paper is a security proof for HCTR. The proof technique that we use is a sequence of games as used in [2, 5, 15]. The previously reported game based proofs for TES performs the final collision analysis on a non-interactive game which runs on a fixed transcript and thus does not depend on the distribution of the queries provided by the adversary. In our proof we do not require the non-interactive game, as we can show that the final collision probabilities are independent of the distribution of the adversarial queries. This observation makes our proof different from the proof in [17] and helps to obtain a better bound.

2 The Construction

In the discussion which follows we shall denote the concatenation of two strings X and Y by $X||Y$. By $|X|$ we shall mean the length of X in bits. $\mathsf{bin}_n(\ell)$ will denote the n bit binary representation of ℓ. For $X, Y \in GF(2^n)$, $X \oplus Y$ and XY will denote addition and multiplication in the field respectively.

HCTR uses two basic building blocks. A universal polynomial hash function and a counter mode of operation. The hash used in case of HCTR is defined as:

$$H_h(X) = X_1 h^{m+1} \oplus X_2 h^m \oplus \ldots \oplus \mathsf{pad}_r(X_m) h^2 \oplus \mathsf{bin}_n(|X|) h \tag{1}$$

Where h is an n-bit hash key and $X = X_1||X_2||\ldots||X_m$, such that $|X_i| = n$ bits $(i = 1, 2, \ldots m-1)$, $0 < |X_m| \leq n$. The pad function is defined as $\mathsf{pad}_r(X_m) := X_m||0^r$ where $r = n - |X_m|$. Thus, $|\mathsf{pad}_r(X_m)| = n$. If $X = \lambda$, the empty string, we define $H_h(\lambda) = h$. In addition to the hash function, HCTR requires a counter mode of operation. Given an n-bit string S, a sequence $S_1, \ldots, S_m$ is defined, where each S_i depends on S. Given such a sequence and a key K the counter mode is defined as follows.

$$\mathsf{Ctr}_{K,S}(A_1, \ldots, A_m) = (A_1 \oplus E_K(S_1), \ldots, A_m \oplus E_K(S_m)). \tag{2}$$

Where $S_i = S \oplus \mathsf{bin}_n(i)$. In case the last block A_m is incomplete then $A_m \oplus E_K(S_m)$ in Eq. 2 is replaced by $A_m \oplus \mathsf{drop}_r(E_K(S_m))$, where $r = n - |A_m|$ and $\mathsf{drop}_r(E_K(S_m))$ is the first $(n-r)$ bits of $E_K(S_m)$. The encryption and decryption operations using HCTR are described in Fig. 1, and a high-level description is provided in Fig. 2. If $m = 1$ (when we have one block message), we ignore line 4 in both encryption and decryption algorithm.

Fig. 1. Encryption using HCTR. K is the block-cipher key, h the hash key and T the tweak.

Algorithm $\mathbf{E}_{K,h}^T(P_1, \ldots, P_m)$	**Algorithm** $\mathbf{D}_{K,h}^T(C_1, \ldots, C_m)$
1. $MM \leftarrow P_1 \oplus H_h(P_2\|\|\ldots\|\|P_m\|\|T)$;	1. $CC \leftarrow C_1 \oplus H_h(C_2\|\|C_3\|\|\ldots\|\|C_m\|\|T)$;
2. $CC \leftarrow E_K(MM)$;	2. $MM \leftarrow E_K^{-1}(CC)$;
3. $S \leftarrow MM \oplus CC$;	3. $S \leftarrow MM \oplus CC$;
4. $(C_2, \ldots, C_{m-1}, C_m)$ $\leftarrow \mathsf{Ctr}_{K,S}(P_2, \ldots, P_m)$;	4. $(P_2, \ldots, P_{m-1}, P_m)$ $\leftarrow \mathsf{Ctr}_{K,S}(C_2, \ldots, C_m)$;
5. $C_1 \leftarrow CC \oplus H_h(C_2\|\|C_3\|\|\ldots\|\|C_m\|\|T)$;	5. $P_1 \leftarrow MM \oplus H_h(P_2\|\|\ldots\|\|P_m\|\|T)$;
6. return $(C_1, \ldots, C_m)$;	6. return $(P_1, \ldots, P_m)$;

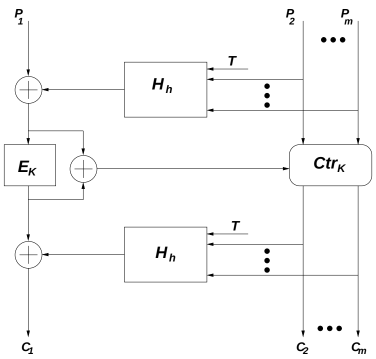

Fig. 2. Encryption using HCTR. Here K is the key for the block cipher $E_K()$ and h is the key for the universal hash function $H_h()$.

HCTR requires m block-cipher calls and $2m+2t+2$ finite field multiplications to encrypt a m block message with a t block tweak. It can be used on any fixed length tweaks. The authors of HCTR prove that the maximum advantage of a chosen plain text and chosen ciphertext adversary in distinguishing HCTR from a random permutation is $\frac{0.5q^2+((2+t)\sigma^2+\sigma^3)}{2^n}$. Where t denotes the length of the tweak and σ denotes the number of blocks of queries made by the adversary. This cubic bound makes HCTR less attractive than other tweakable enciphering schemes all of which are known to have a security bound of the order of $\frac{\sigma^2}{2^n}$.

In a recent work [13] a general construction of tweakable SPRP was reported by using universal hash functions, tweakable block-ciphers and a weak pseudorandom function. The paper [13] also reports a variant of HCTR which comes as an instantiation of their general construction. They claim that this variant of HCTR has a quadratic security bound. But, this variant is quite different and also inefficient from the original specification of HCTR. The variant reported in [13] needs one more block-cipher call than the original HCTR.

3 Improved Bound for HCTR

3.1 Definitions and Notation

The discussion in this section is based on [6]. An n-bit block cipher is a function $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$, where $\mathcal{K} \neq \emptyset$ is the key space and for any $K \in \mathcal{K}$, $E(K,.)$ is a permutation. We write $E_K(\,)$ instead of $E(K,.)$.

An adversary A is a probabilistic algorithm which has access to some oracles and which outputs either 0 or 1. Oracles are written as superscripts. The notation $A^{\mathcal{O}_1,\mathcal{O}_2} \Rightarrow 1$ denotes the event that the adversary A, interacts with the oracles $\mathcal{O}_1, \mathcal{O}_2$, and finally outputs the bit 1. In what follows, by the notation $X \stackrel{\$}{\leftarrow} \mathcal{S}$, we will denote the event of choosing X uniformly at random from the finite set $\mathcal{S}$.

Let $\mathrm{Perm}(n)$ denote the set of all permutations on $\{0,1\}^n$. We require $E(,)$ to be a strong pseudorandom permutation. The advantage of an adversary A in breaking the strong pseudorandomness of $E(,)$ is defined in the following manner.

$$\mathbf{Adv}_E^{\pm\mathrm{prp}}(A) = \left|\Pr\left[K \xleftarrow{\$} \mathcal{K} : A^{E_K(\,),E_K^{-1}(\,)} \Rightarrow 1\right] - \right.$$
$$\left.\Pr\left[\pi \xleftarrow{\$} \mathrm{Perm}(n) : A^{\pi(\,),\pi^{-1}(\,)} \Rightarrow 1\right]\right|.$$

A tweakable enciphering scheme is a function $\mathbf{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$, where $\mathcal{K} \neq \emptyset$ and $\mathcal{T} \neq \emptyset$ are the key space and the tweak space respectively. The message and the cipher spaces are $\mathcal{M}$. For HCTR we have $\mathcal{M} = \cup_{i>n}\{0,1\}^i$. We shall write $\mathbf{E}_K^T(.)$ instead of $\mathbf{E}(K,T,.)$. The inverse of an enciphering scheme is $\mathbf{D} = \mathbf{E}^{-1}$ where $X = \mathbf{D}_K^T(Y)$ if and only if $\mathbf{E}_K^T(X) = Y$.

Let $\mathrm{Perm}^{\mathcal{T}}(\mathcal{M})$ denote the set of all functions $\boldsymbol{\pi} : \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ where $\boldsymbol{\pi}(\mathcal{T},.)$ is a length preserving permutation. Such a $\boldsymbol{\pi} \in \mathrm{Perm}^{\mathcal{T}}(\mathcal{M})$ is called a tweak indexed permutation. For a tweakable enciphering scheme $\mathbf{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$, we define the advantage an adversary A has in distinguishing $\mathbf{E}$ and its inverse from a random tweak indexed permutation and its inverse in the following manner.

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

Here, $\boldsymbol{\pi} \xleftarrow{\$} \mathrm{Perm}^{\mathcal{T}}(\mathcal{M})$ means that for each ℓ such that $\{0,1\}^\ell \subseteq \mathcal{M}$ and $T \in \mathcal{T}$ we choose a tweakable random permutation π^T from $\mathrm{Perm}(\ell)$ independently. We define $\mathbf{Adv}_{\mathbf{E}}^{\pm\widetilde{\mathrm{prp}}}(q,\sigma)$ by $\max_A \mathbf{Adv}_{\mathbf{E}}^{\pm\widetilde{\mathrm{prp}}}(A)$ where maximum is taken over all adversaries which makes at most q queries having at most σ many blocks. For a computational advantage we define $\mathbf{Adv}_{\mathbf{E}}^{\pm\widetilde{\mathrm{prp}}}(q,\sigma,t)$ by $\max_A \mathbf{Adv}_{\mathbf{E}}^{\pm\widetilde{\mathrm{prp}}}(A)$. In addition to the previous restrictions on A, he can run in time at most t.

Pointless queries: Let T, P and C represent tweak, plaintext and ciphertext respectively. We assume that an adversary never repeats a query, i.e., it does not ask the encryption oracle with a particular value of (T,P) more than once and neither does it ask the decryption oracle with a particular value of (T,C) more than once. Furthermore, an adversary never queries its deciphering oracle with (T,C) if it got C in response to an encipher query (T,P) for some P. Similarly, the adversary never queries its enciphering oracle with (T,P) if it got P as a response to a decipher query of (T,C) for some C. These queries are called *pointless* as the adversary knows what it would get as responses for such queries.

The notation HCTR[E] denotes a tweakable enciphering scheme, where the n-bit block cipher E is used in the manner specified by HCTR. We will use the notation $\mathbf{E}_\pi$ as a shorthand for HCTR[Perm(n)] and $\mathbf{D}_\pi$ will denote the inverse of $\mathbf{E}_\pi$. Thus, the notation $A^{\mathbf{E}_\pi,\mathbf{D}_\pi}$ will denote an adversary interacting with the oracles $\mathbf{E}_\pi$ and $\mathbf{D}_\pi$.

3.2 Statement of Results

The following theorem specifies the security of HCTR.

Theorem 1. *Fix n, σ to be positive integers and an n-bit block cipher $E : \mathcal{K} \times \{0,1\}^n \rightarrow \{0,1\}^n$. Then*

$$\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\mathrm{HCTR}[\mathrm{Perm}(n)]}(\sigma) \leq \frac{4.5\sigma^2}{2^n}. \tag{4}$$

$$\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\mathrm{HCTR}[E]}(\sigma, t) \leq \frac{4.5\sigma^2}{2^n} + \mathbf{Adv}^{\pm\mathrm{prp}}_{E}(\sigma, t') \tag{5}$$

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

The above result and its proof is similar to previous work (see for example [6, 7, 3]). As mentioned in [6], Equation (5) embodies a standard way to pass from the information theoretic setting to the complexity theoretic setting.

For proving (4), we need to consider an adversary's advantage in distinguishing a tweakable enciphering scheme $\mathbf{E}$ from an oracle which simply returns random bit strings. This advantage is defined in the following manner.

$$\begin{aligned}\mathbf{Adv}^{\pm\mathrm{rnd}}_{\mathrm{HCTR}[\mathrm{Perm}(n)]}(A) = \Big|&\Pr\left[\pi \xleftarrow{\$} \mathrm{Perm}(n) : A^{\mathbf{E}_\pi,\mathbf{D}_\pi} \Rightarrow 1\right] \\ &- \Pr\left[A^{\$(.,.),\$(.,.)} \Rightarrow 1\right]\Big|\end{aligned} \tag{6}$$

where $\$(., M)$ or $\$(., C)$ returns independently distributed random bits of length $|M|$ or $|C|$ respectively. The basic idea of proving (4) is as follows.

$$\begin{aligned}\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\mathrm{HCH}[\mathrm{Perm}(n)]}(A) &= \Big(\Pr\left[\pi \xleftarrow{\$} \mathrm{Perm}(n) : A^{\mathbf{E}_\pi,\mathbf{D}_\pi} \Rightarrow 1\right] \\ &\quad - \Pr\left[\boldsymbol{\pi} \xleftarrow{\$} \mathrm{Perm}^{\mathcal{T}}(\mathcal{M}) : A^{\boldsymbol{\pi}(.,.),\boldsymbol{\pi}^{-1}(.,.)} \Rightarrow 1\right]\Big) \\ &= \Big(\Pr\left[\pi \xleftarrow{\$} \mathrm{Perm}(n) : A^{\mathbf{E}_\pi,\mathbf{D}_\pi} \Rightarrow 1\right] \\ &\quad - \Pr\left[A^{\$(.,.),\$(.,.)} \Rightarrow 1\right]\Big) \\ &\quad + \Big(\Pr\left[A^{\$(.,.),\$(.,.)} \Rightarrow 1\right] \\ &\quad - \Pr\left[\boldsymbol{\pi} \xleftarrow{\$} \mathrm{Perm}^{\mathcal{T}}(\mathcal{M}) : A^{\boldsymbol{\pi}(.,.),\boldsymbol{\pi}^{-1}(.,.)} \Rightarrow 1\right]\Big) \\ &\leq \mathbf{Adv}^{\pm\mathrm{rnd}}_{\mathrm{HCH}[\mathrm{Perm}(n)]}(A) + \binom{q}{2}\frac{1}{2^n}\end{aligned} \tag{7}$$

where q is the number of queries made by the adversary. For a proof of the last inequality see [6]. Thus, the main task of the proof now reduces to obtaining an upper bound on $\mathbf{Adv}^{\pm\mathrm{rnd}}_{\mathrm{HCTR}[\mathrm{Perm}(n)]}(\sigma)$. In section 4 we prove that

$$\mathbf{Adv}^{\pm\mathrm{rnd}}_{\mathrm{HCTR}[\mathrm{Perm}(n)]}(\sigma) \leq \frac{4\sigma^2}{2^n}. \tag{8}$$

Using equation (8) and (7) we obtain equation (4).

4 The Game Sequence

We shall model the interaction of the adversary with HCTR by a sequence of games. We shall start with the game HCTR1 which describes the mode HCTR, and with small changes we shall reach the game RAND2 which will represent an oracle which returns just random strings and we shall bound the advantage of an adversary in distinguishing between the games HCTR1 and RAND1. Where G represents a game by $\Pr[A^G \Rightarrow 1]$ we shall mean the probability that A outputs 1 by interacting with the game G. Next we describe the games.

Game HCTR1: In HCTR1, the adversary interacts with $\mathbf{E}_\pi$ and $\mathbf{D}_\pi$ where π is a randomly chosen permutation from $\mathrm{Perm}(n)$. Instead of initially choosing π, we build up π in the following manner.

Initially π is assumed to be undefined everywhere. When $\pi(X)$ is needed, but the value of π is not yet defined at X, then a random value is chosen among the available range values. Similarly when $\pi^{-1}(Y)$ is required and there is no X yet defined for which $\pi(X) = Y$, we choose a random value for $\pi^{-1}(Y)$ from the available domain values.

The domain and range of π are maintained in two sets *Domain* and *Range*, and $\overline{Domain}$ and $\overline{Range}$ are the complements of *Domain* and *Range* relative to $\{0,1\}^n$. The game HCTR1 is shown in Figure 3. The figure shows the subroutines Ch-π, Ch-π^{-1}, the initialization steps and how the game responds to a encipher/decipher query of the adversary. The i^{th} query of the adversary depends on its previous queries, the responses to those queries and on some coins of the adversary. When $l^s = n$, we ignore the line 103 to line 109.

The game HCTR1 accurately represents the attack scenario, and by our choice of notation, we can write

$$\Pr[A^{\mathbf{E}_\pi,\mathbf{D}_\pi} \Rightarrow 1] = \Pr[A^{\mathrm{HCTR1}} \Rightarrow 1]. \tag{9}$$

Game RAND1: We modify HCTR1 by deleting the boxed entries in HCTR1 and call the modified game as RAND1. By deleting the boxed entries it cannot be guaranteed that π is a permutation as though we do the consistency checks but we do not reset the values of Y (in Ch-π) and X (in Ch-π^{-1}). Thus, the games HCTR1 and RAND1 are identical apart from what happens when the bad flag is set. By using the result from [1], we obtain

$$|\Pr[A^{\mathrm{HCTR1}} \Rightarrow 1] - \Pr[A^{\mathrm{RAND1}} \Rightarrow 1]| \leq \Pr[A^{\mathrm{RAND1}} \text{ sets } \mathsf{bad}] \tag{10}$$

Fig. 3. Games HCTR1 and RAND1

Subroutine Ch-$\pi(X)$

$Y \xleftarrow{\$} \{0,1\}^n$; **if** $Y \in \mathit{Range}$ **then** $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{Y \xleftarrow{\$} \overline{\mathit{Range}}}$; **endif**;
if $X \in \mathit{Domain}$ **then** $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{Y \leftarrow \pi(X)}$; **endif**
$\pi(X) \leftarrow Y$; $\mathit{Domain} \leftarrow \mathit{Domain} \cup \{X\}$; $\mathit{Range} \leftarrow \mathit{Range} \cup \{Y\}$; **return**$(Y)$;

Subroutine Ch-$\pi^{-1}(Y)$

$X \xleftarrow{\$} \{0,1\}^n$; **if** $X \in \mathit{Domain}$, $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{X \xleftarrow{\$} \overline{\mathit{Domain}}}$; **endif**;
if $Y \in \mathit{Range}$ **then** $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{X \leftarrow \pi^{-1}(Y)}$; **endif**;
$\pi(X) \leftarrow Y$; $\mathit{Domain} \leftarrow \mathit{Domain} \cup \{X\}$; $\mathit{Range} \leftarrow \mathit{Range} \cup \{Y\}$; **return**$(X)$;

<u>Initialization</u>:
for all $X \in \{0,1\}^n$ $\pi(X) = \mathsf{undef}$ **endfor**
$\mathsf{bad} = \mathsf{false}$

Respond to the s^{th} query as follows: (Assume $l^s = n(m^s - 1) + r^s$, with $0 \leq r^s < n$.)

<u>Encipher query:</u> $\mathsf{Enc}(T^s; P_1^s, P_2^s, \ldots P_{m^s}^s)$	<u>Decipher query:</u> $\mathsf{Dec}(C_1^s, C_2^s, \ldots, C_{m^s}^s, T^s)$
$MM^s \leftarrow P_1^s \oplus H_h(P_2^s \| \ldots \| P_m^s \| T^s)$;	$CC^s \leftarrow C_1^s \oplus H_h(C_2^s \| \ldots \| C_m^s \| T^s)$;
$CC^s \leftarrow$ Ch-$\pi(MM^s)$;	$MM^s \leftarrow$ Ch-$\pi^{-1}(CC^s)$
$S^s \leftarrow MM^s \oplus CC^s$;	$S^s \leftarrow MM^s \oplus CC^s$;
for $i = 1$ to $m^s - 2$,	**for** $i = 1$ to $m^s - 2$,
$Z_i^s \leftarrow$ Ch-$\pi(S^s \oplus \mathsf{bin}_n(i))$;	$Z_i^s \leftarrow$ Ch-$\pi(S^s \oplus \mathsf{bin}_n(i))$;
$C_{i+1}^s \leftarrow P_{i+1}^s \oplus Z_i^s$;	$P_{i+1}^s \leftarrow C_{i+1}^s \oplus Z_i^s$;
end for	**end for**
$Z_{m^s}^s \leftarrow$ Ch-$\pi(S^s \oplus \mathsf{bin}_n(m^s - 1))$;	$Z_{m^s}^s \leftarrow$ Ch-$\pi(S^s \oplus \mathsf{bin}_n(m^s - 1))$;
$C_{m^s}^s \leftarrow P_{m^s}^s \oplus \mathsf{drop}_{n-r^s}(Z_{m^s}^s)$;	$P_{m^s}^s \leftarrow C_{m^s}^s \oplus \mathsf{drop}_{n-r^s}(Z_{m^s}^s)$;
	$P_1^s \leftarrow MM^s \oplus H_h(P_2^s \| \ldots \| P_m^s \| T^s)$;
$C_1^s \leftarrow CC^s \oplus H_h(C_2^s \| \ldots \| C_{m^s}^s \| T^s)$;	
return $C_1^s \| C_2^s \| \ldots \| C_{m^s}^s$	**return** $P_2^s \| \ldots \| P_{m^s}^s$

Another important thing to note is that in RAND1 in line 103, for a encryption query CC^s (and MM^s for a decryption query) gets set to a random n bit string. Similarly 105 and 108 Z_i^s gets set to random values. Thus the the adversary gets random strings in response to both his encryption and decryption queries. Hence,

$$\Pr[A^{\mathrm{RAND1}} \Rightarrow 1] = \Pr[A^{\$(.,.),\$(.,.)} \Rightarrow 1] \tag{11}$$

So using Equations (6), (10) and (11) we get

$$\begin{aligned}\mathbf{Adv}^{\pm\mathrm{rnd}}_{\mathrm{HCTR}[\mathrm{Perm}(n)]}(A) &= |\Pr[A^{\mathbf{E}_\pi,\mathbf{D}_\pi} \Rightarrow 1] - \Pr[A^{\$(.,.),\$(.,.)} \Rightarrow 1]| \quad (12)\\ &= |\Pr[A^{\mathrm{HCTR1}} \Rightarrow 1] - \Pr[A^{\mathrm{RAND1}} \Rightarrow 1]| \\ &\le \Pr[A^{\mathrm{RAND1}} \text{ sets } \mathsf{bad}] \quad (13)\end{aligned}$$

Game RAND2: Now we make some subtle changes in the game RAND1 to get a new game RAND2 which is described in Figure 4. In game RAND1 the permutation was not maintained and a call to the permutation was responded by returning random strings, so in Game RAND2 we no more use the subroutines Ch-π and Ch-π^{-1}. Here we immediately return random strings to the adversary in response to his encryption or decryption queries. Later in the finalization step we adjust variables and maintain multi sets $\mathcal{D}$ and $\mathcal{R}$ where we list the elements that were supposed to be inputs and outputs of the permutation. In the second phase of the finalization step, we check for collisions in the sets $\mathcal{D}$ and $\mathcal{R}$, and in the event of a collision we set the bad flag to true.

Game RAND1 and Game RAND2 are indistinguishable to the adversary, as in both cases he gets random strings in response to his queries. Also, the probability with which RAND1 sets bad is same as the probability with which RAND2 sets bad. Thus we get:

$$\Pr[A^{\mathrm{RAND1}} \text{ sets } \mathsf{bad}] = \Pr[A^{\mathrm{RAND2}} \text{ sets } \mathsf{bad}] \tag{14}$$

Thus from Equations (13) and (14) we obtain

$$\mathbf{Adv}^{\pm\mathrm{rnd}}_{\mathrm{HCTR}[\mathrm{Perm}(n)]}(A) \le \Pr[A^{\mathrm{RAND2}} \text{ sets } \mathsf{bad}] \tag{15}$$

Now our goal would be to bound $\Pr[A^{\mathrm{RAND2}} \text{ sets } \mathsf{bad}]$. We notice that in Game RAND2 the bad flag is set when there is a collision in either of the sets $\mathcal{D}$ or $\mathcal{R}$. So if COLLD and COLLR denote the events of a collision in $\mathcal{D}$ and $\mathcal{R}$ respectively then we have

$$\Pr[A^{\mathrm{RAND2}} \text{ sets } \mathsf{bad}] \le \Pr[\mathsf{COLLR}] + \Pr[\mathsf{COLLD}]$$

In many previously reported game based proofs for strong pseudorandom permutations including the proof given in [17], the final collision analysis is done on a non-interactive game. The non-interactive game is generally obtained by eliminating the randomness present in the distribution of the queries presented

Fig. 4. Game RAND2

Respond to the s^{th} adversary query as follows:

ENCIPHER QUERY $\mathsf{Enc}(T^s; P_1^s, P_2^s, \ldots, P_{m^s}^s)$

$ty^s = \mathsf{Enc}$; $C_1^s||C_2^s||\ldots||C_{m^s-1}^s||D_{m^s}^s \xleftarrow{\$} \{0,1\}^{nm^s}$;
$C_{m^s}^s \leftarrow \mathsf{drop}_{n-r^s}(D_{m^s})$ **return** $C_1^s||C_2^s||\ldots||C_{m^s}^s$;

DECIPHER QUERY $\mathsf{Dec}(T^s; C_1^s, C_2^s, \ldots, C_{m^s}^s)$

$ty^s = \mathsf{Dec}$; $P_1^s||P_2^s||\ldots||P_{m_s-1}^s||V_{m^s}^s \xleftarrow{\$} \{0,1\}^{nm^s}$;
$P_{m^s}^s \leftarrow \mathsf{drop}_{n-r^s}(V_{m^s})$ **return** $P_1^s||P_2^s||\ldots||P_{m^s}^s$;

Finalization:

Case $ty^s = \mathsf{Enc}$:

$MM^s \leftarrow P_1^s \oplus H_h(P_2^s||\ldots||P_m^s||T^s)$;
$CC^s \leftarrow C_1^s \oplus H_h(C_2^s||\ldots||C_m^s||T^s)$;
$S^s \leftarrow MM^s \oplus CC^s$;
$\mathcal{D} \leftarrow \mathcal{D} \cup \{MM^s\}$;
$\mathcal{R} \leftarrow \mathcal{R} \cup \{CC^s\}$;
for $i = 2$ to $m^s - 1$,
  $Y_i^s \leftarrow C_i^s \oplus P_i^s$;
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{S^s \oplus \mathsf{bin}_n(i-1)\}$;
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{Y_i^s\}$;
end for
$Y_{m^s}^s \leftarrow D_{m^s}^s \oplus P_{m^s}^s$
$\mathcal{D} \leftarrow \mathcal{D} \cup \{S^s \oplus \mathsf{bin}_n(m^s-1)\}$;
$\mathcal{R} \leftarrow \mathcal{R} \cup \{Y_{m^s}^s\}$;

Case $ty^s = \mathsf{Dec}$:

$MM^s \leftarrow P_1^s \oplus H_h(P_2^s||\ldots||P_m^s||T^s)$;
$CC^s \leftarrow C_1^s \oplus H_h(C_2^s||\ldots||C_m^s||T^s)$;
$S^s \leftarrow MM^s \oplus CC^s$;
$\mathcal{D} \leftarrow \mathcal{D} \cup \{MM^s\}$;
$\mathcal{R} \leftarrow \mathcal{R} \cup \{CC^s\}$;
for $i = 2$ to $m^s - 1$,
  $Y_i^s \leftarrow C_i^s \oplus P_i^s$;
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{S^s \oplus \mathsf{bin}_n(i-1)\}$;
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{Y_i^s\}$;
end for
$Y_{m^s}^s \leftarrow V_{m^s}^s \oplus C_{m^s}^s$
$\mathcal{D} \leftarrow \mathcal{D} \cup \{S^s \oplus \mathsf{bin}_n(m^s-1)\}$;
$\mathcal{R} \leftarrow \mathcal{R} \cup \{Y_{m^s}^s\}$;

SECOND PHASE

$\mathsf{bad} = \mathsf{false}$;
if (some value occurs more than once in $\mathcal{D}$) **then** $\mathsf{bad} = \mathsf{true}$ **endif**;
if (some value occurs more than once in $\mathcal{R}$) **then** $\mathsf{bad} = \mathsf{true}$ **endif**.

by the adversary. To achieve this the final non-interactive game runs on a fixed transcript which maximizes the probability of bad being set to true. In this case as we will soon see, such a de-randomization is not required. Because of the specific structure of the game RAND2 the probability COLLR and COLLD would be independent of the distribution of the queries supplied by the adversary, hence a final collision analysis can be done on the game RAND2 itself.

4.1 Bounding collision probability in $\mathcal{D}$ and $\mathcal{R}$

In the analysis we consider the sets $\mathcal{D}$ and $\mathcal{R}$ to consist of the formal variables instead of their values. For example, whenever we set $\mathcal{D} \leftarrow \mathcal{D} \cup \{X\}$ for some variable X we think of it as setting $\mathcal{D} \leftarrow \mathcal{D} \cup \{\text{“}X\text{”}\}$ where “X” is the name of that formal variable. This is the same technique as used in [6]. Our goal is to bound the probability that two formal variables in the sets $\mathcal{D}$ and $\mathcal{R}$ take the same value. After q queries of the adversary where the s^{th} query has m^s blocks of plaintext or ciphertext and t block of tweak, then the sets $\mathcal{D}$ and $\mathcal{R}$ can be written as follows:

$$\begin{array}{ll}
\text{Elements in } \mathcal{D}: & MM^s = P_1^s \oplus Q^s, \\
& S_j^s = S^s \oplus \mathsf{bin}_n(j) = (P_1^s \oplus C_1^s) \oplus (Q^s \oplus B^s \oplus \mathsf{bin}_n(j)), \\
& \text{where } Q^s = H_h(P_2^s \parallel \cdots \parallel P_{m^s}^s \parallel T^s) \text{ and} \\
& B^s = H_h(C_2^s \parallel \cdots \parallel C_{m^s}^s \parallel T^s), \\
& 1 \le s \le q,\ 1 \le i \le m^s - 1, \\
& \\
\text{Elements in } \mathcal{R}: & CC^s = C_1^s \oplus B^s, \\
& Y_i^s = C_i^s \oplus P_i^s, \\
& 2 \le i \le m^s,\ 1 \le s \le q.
\end{array}$$

Before we present the collision analysis let us identify the random variables based on which the probability of collision would be computed. In game RAND2 the hash key h is selected uniformly from the set $\{0,1\}^n$. The outputs that the adversary receives are also uniformly distributed, and are independent of the previous queries supplied by the adversary and the outputs obtained by the adversary. The i^{th} query supplied by the adversary may depend on the previous outputs obtained by the adversary, but as the output of GAME2 is not dependent in any way on the hash key h thus the queries supplied by the adversary are independent of h.

We consider T^s as t n-bit blocks. Thus, for any s, $H_h(P_2^s||\cdots||P_{m^s}^s||T^s)$ or $H_h(C_2^s||\cdots||C_{m^s}^s||T^s)$ has degree at most $m^s + t$. We denote $\sigma = qt + \sum_s m^s$. We denote $\ell^{s,s'} = \max\{m^s, m^{s'}\} + t$. Since $\ell^{s,s'} \le m^s + m^{s'} + t$, we have the following inequality

$$\begin{aligned}\sum_{1\le s<s'\le q} \ell^{s,s'} &\le \binom{q}{2}t + \sum_{1\le s<s'\le q}(m^s + m^{s'})\\ &\le \binom{q}{2}t + (q-1)(\sigma - qt)\\ &\le (q-1)\sigma + \frac{qt(q-1)}{2} - qt(q-1)\\ &\le (q-1)\sigma.\end{aligned}$$

We also note that the response of encryption or decryption query are completely independent of h (the poly hash key). Thus, inputs of $H_h(\cdot)$ for each query are independent with h. So we can use the fundamental theorem of algebra to claim that the probability that h is a root of a d degree polynomial is at most $d/2^n$ where h is chosen uniformly and independently from the coefficient of the polynomial (which is true in case of H_h in RAND2 game).

First we consider the collisions in $\mathcal{R}$.

– We first consider collision among CC^s. Let $s' \neq s$. Now, $\Pr[CC^s = CC^{s'}] \le \ell^{s,s'}/2^n$ where the probability is computed under the uniform choice of $h \in \{0,1\}^n$. We know that $CC^s \oplus CC^{s'}$ is a non-zero polynomial of h with degree at most $\ell^{s,s'}$. By using fundamental theorem of algebra we have the above bound for the collision probability. Thus,

$$\begin{aligned}\Pr[CC^s = CC^{s'} : \text{for some } 1 \le s < s' \le q] &\le \sum_{1\le s<s'\le q} \frac{\ell^{s,s'}}{2^n}\\ &\le \frac{(q-1)\sigma}{2^n}.\end{aligned} \tag{16}$$

Similarly we can compute collision probability between Y_i^s and $CC^{s'}$. For each s', there are $(\sigma - qt - q)$ many Y_i^s's. For each such choice, $\Pr[CC^{s'} = Y_i^s] \le (m^{s'} + t)/2^n$. Thus,

$$\begin{aligned}&\Pr[CC^{s'} = Y_i^s : \text{ for some } 1 \le s \neq s' \le q, 2 \le i \le m^s]\\ &\qquad \le \sum_{1\le s'\le q} \frac{(\sigma - qt - q)(m^{s'} + t)}{2^n}\\ &\qquad \le \sigma^2/2^n.\end{aligned} \tag{17}$$

– Now we consider collision among Y_i^s, $2 \le i \le m^s$, $1 \le s \le q$. For the pairs $(Y_i^s, Y_{i'}^{s'})$ with $s' \le s$ and $(s,i) \neq (s',i')$, the collision probability is $1/2^n$, since either P^s or C^s is chosen uniformly and independently from the rest of the variables. There are $\binom{\sigma-qt-q}{2}$ pairs of this form. Thus,

$$\begin{aligned}&\Pr[Y_i^s = Y_{i'}^{s'} : \text{ for some } 1 \le s \le s' \le q, 1 \le i, i' \le q, (s,i) \neq (s',i')]\\ &\qquad \le \binom{\sigma - qt - q}{2}/2^n.\end{aligned} \tag{18}$$

Combining equation (16), (17) and (18) we obtain

$$\Pr[\mathrm{COLLR}] \le \frac{4\sigma^2}{2^{n+1}}. \tag{19}$$

Now we consider collision in domain $\mathcal{D}$.

- Similar to equations (16) and (17), we have

$$\begin{aligned}\Pr[MM^s = MM^{s'} : \text{ for some } 1 \le s < s' \le q] &\le \sum_{1\le s<s'\le q} \frac{\ell^{s,s'}}{2^n} \\ &\le (q-1)\sigma/2^n. \end{aligned} \tag{20}$$

$$\Pr[MM^{s'} = S_i^s : \text{ for some } 1 \le s \ne s' \le q, 2 \le i \le m^s] \le \sigma^2/2^n. \tag{21}$$

- Now we consider collision among $S_i^s = S^s \oplus \mathsf{bin}_n(i)$, $2 \le i \le m^s$, $1 \le s \le q$. Note that, $S_i^s = S_{i'}^{s'}$ implies that $(P_1^s \oplus C_1^s) \oplus (Q^s \oplus B^s \oplus \mathsf{bin}_n(i)) = (P_1^{s'} \oplus C_1^{s'}) \oplus (Q^{s'} \oplus B^{s'} \oplus \mathsf{bin}_n(i'))$. Let $s' \le s$ and $(s,i) \ne (s',i')$. Thus, either C_1^s (in case s^{th} query is encryption) or P_1^s (in case s^{th} query is decryption) is uniformly and independently distributed with all other variables stated in the above equality. Thus, the collision probability is $1/2^n$. Since there are $\binom{\sigma-qt-q}{2}$ pairs of this form, we have

$$\begin{aligned}\Pr[S_i^s = S_{i'}^{s'} : \text{ for some } 1 \le s \le s' \le q, 1 \le i, i' \le q, (s,i) \ne (s',i')] \\ \le \binom{\sigma - qt - q}{2}/2^n. \end{aligned} \tag{22}$$

The equations (20), (21) and (22) imply the following similar bound for domain collision probability.

$$\Pr[\mathrm{COLLD}] \le \frac{4\sigma^2}{2^{n+1}}. \tag{23}$$

Combining the domain and range collision probabilities, we obtain the probability of bad being set to true in RAND2 to be at most $8\sigma^2/2^{n+1}$. Thus, by using equations (19) and (23), we have

$$\mathbf{Adv}^{\pm\mathrm{rnd}}_{\mathrm{HCTR}[\mathrm{Perm}(n)]}(A) \le \frac{4\sigma^2}{2^n}. \tag{24}$$

5 Discussions

Why our bound is different from [17]: The analysis that we perform is very similar to that presented in [17]. As stated earlier, the authors in [17] presents their collision analysis on a non-interactive game where the plain texts and ciphertexts are fixed. Thus they obtain a different bound for the probability of collisions between S_i^s and $S_{i'}^{s'}$. As they consider the plaintext and ciphertexts to be fixed thus they conclude that the probability of collision between each pair is less than $\ell/2^n$, where ℓ is the maximum length of a query supplied by the adversary. Thus according to their analysis they obtain

$$\Pr[S_i^s = S_{i'}^{s'} : \text{ for some } 1 \leq s \leq s' \leq q, 1 \leq i, i' \leq q, (s,i) \neq (s', i')] \leq \ell \binom{\sigma - qt - q}{2} / 2^n. \tag{25}$$

This term contributes to the cubic security bound reported in [17].

The bound claimed in [13] : In [13] a improved bound provided of a variant of HCTR. Firstly, the variant uses one more block-cipher call than HCTR making it less efficient than the original construction. Secondly, they claim that the security bound of modified HCTR is $O(\frac{q^2\ell^2}{2^n})$, where ℓ is the maximum query length. This bound is uniformly larger than our bound.

6 Conclusion

We provided a improved security analysis of the HCTR mode of operation. This work thus establish that HCTR provides same security guarantee as provided by CMC, EME, EME*, XCB, PEP, HCH, TET, and HEH (to our knowledge these are the only TES with a security proof).

References

1. Mihir Bellare and Phillip Rogaway. Code-based game-playing proofs and the security of triple encryption. Cryptology ePrint Archive, Report 2004/331, 2004. http://eprint.iacr.org/.
2. Debrup Chakraborty and Palash Sarkar. HCH: A new tweakable enciphering scheme using the hash-encrypt-hash approach. In Rana Barua and Tanja Lange, editors, *INDOCRYPT*, volume 4329 of *Lecture Notes in Computer Science*, pages 287–302. Springer, 2006. Extended version in http://eprint.iacr.org/2007/028.
3. Debrup Chakraborty and Palash Sarkar. A new mode of encryption providing a tweakable strong pseudo-random permutation. In Matthew J. B. Robshaw, editor, *FSE*, volume 4047 of *Lecture Notes in Computer Science*, pages 293–309. Springer, 2006.
4. Shai Halevi. EME*: Extending EME to handle arbitrary-length messages with associated data. In Anne Canteaut and Kapalee Viswanathan, editors, *INDOCRYPT*, volume 3348 of *Lecture Notes in Computer Science*, pages 315–327. Springer, 2004.
5. Shai Halevi. Invertible universal hashing and the tet encryption mode. In *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 412–429. Springer, 2007.
6. Shai Halevi and Phillip Rogaway. A tweakable enciphering mode. In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 482–499. Springer, 2003.

7. Shai Halevi and Phillip Rogaway. A parallelizable enciphering mode. In Tatsuaki Okamoto, editor, *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 292–304. Springer, 2004.
8. IEEE Security in Storage Working Group (SISWG). PRP modes comparison IEEE p1619.2. IEEE Computer Society, March 2007. Available at:http://siswg.org/.
9. Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2002.
10. Cuauhtemoc Mancillas-López, Debrup Chakraborty, and Francisco Rodríguez-Henríquez. Efficient implementations of some tweakable enciphering schemes in reconfigurable hardware. In *INDOCRYPT*, volume 4859 of *Lecture Notes in Computer Science*, pages 414–424. Springer, 2007.
11. David A. McGrew and Scott R. Fluhrer. The extended codebook (XCB) mode of operation. Cryptology ePrint Archive, Report 2004/278, 2004. http://eprint.iacr.org/.
12. David A. McGrew and John Viega. Arbitrary block length mode, 2004. http://grouper.ieee.org/groups/1619/email/pdf00005.pdf.
13. Kazuhiko Minematsu and Toshiyasu Matsushima. Tweakable enciphering schemes from hash-sum-expansion. In *INDOCRYPT*, volume 4859 of *Lecture Notes in Computer Science*, pages 252–267. Springer, 2007.
14. Moni Naor and Omer Reingold. A pseudo-random encryption mode. Manuscript available from www.wisdom.weizmann.ac.il/˜naor.
15. Palash Sarkar. Improving upon the TET mode of operation. In *INDOCRYPT*, volume 4817 of *Lecture Notes in Computer Science*, pages 180–192. Springer, 2007.
16. Palash Sarkar. Efficient tweakable enciphering schemes from (block-wise) universal hash functions. Cryptology ePrint Archive, Report 2008/004, 2008. http://eprint.iacr.org/.
17. Peng Wang, Dengguo Feng, and Wenling Wu. HCTR: A variable-input-length enciphering mode. In Dengguo Feng, Dongdai Lin, and Moti Yung, editors, *CISC*, volume 3822 of *Lecture Notes in Computer Science*, pages 175–188. Springer, 2005.