Tweakable HCTR: A BBB Secure Tweakable Enciphering Scheme

Avijit Dutta and Mridul Nandi

Indian Statistical Institute, Kolkata.
avirocks.dutta13@gmail.com, mridul.nandi@gmail.com

Abstract. HCTR, proposed by Wang et al., is one of the most efficient candidates of tweakable enciphering schemes that turns an n-bit block cipher into a variable input length tweakable block cipher. Wang et al. have shown that HCTR offers a cubic security bound against all adaptive chosen plaintext and chosen ciphertext adversaries. Later in FSE 2008, Chakraborty and Nandi have improved its bound to $O(\sigma^2/2^n)$, where σ is the total number of blocks queried and n is the block size of the block cipher. In this paper, we propose **tweakable HCTR** that turns an n-bit tweakable block cipher to a variable input length tweakable block cipher by replacing all the block cipher calls of HCTR with tweakable block cipher. We show that when there is no repetition of the tweak, tweakable HCTR enjoys the optimal security against all adaptive chosen plaintext and chosen ciphertext adversaries. However, if the repetition of the tweak is limited, then the security of the construction remains close to the security bound in no repetition of the tweak case. Hence, it gives a graceful security degradation with the maximum number of repetition of tweaks.

Keywords: Tweakable Enciphering Scheme, HCTR, TSPRP, H-Coefficient.

1 Introduction

Tweakable Enciphering Scheme. A block cipher is a fundamental primitive in symmetric key cryptography that processes only fixed length messages. Examples of such block ciphers are DES [29], AES [10] etc. The general security notion for a block cipher is pseudorandom permutation (PRP) which says that any computationally bounded adversary should be unable to distinguish between a random permutation and a permutation picked at random from a keyed family of permutations over the input set. A stronger security notion for block cipher called strong pseudorandom permutation (SPRP) requires computationally bounded adversary should be unable to distinguish between a random permutation and its inverse from a permutation and its inverse, picked at random from the keyed family of permutations. A *mode of operation* of a block cipher specifies a particular way the block cipher should be used to process arbitrary and variable length messages; hence extending the domain of applicability from fixed length messages to long and variable length messages. As its security requirement, we require that it should be secure if the underlying block cipher

is a secure PRP, then the extended domain mode of operation also satisfies an appropriate notion of security.

The two major goals of a mode of operation that it wants to achieve are confidentiality and integrity. For example, CBC [36] mode provides only confidentiality whereas CBC-MAC [1] is a mode of operation that guarrantees only integrity. OCB [33] is a mode of operation which provides both confidentiality and integrity. A mode of operation that can encrypt arbitrary length messages and provides SPRP security is called a *length preserving transformation* for which no tag is produced. In that case, a change in the ciphertext remains undetected but the decryption of a tampered ciphertext results in a plaintext which is indistinguishable from a random string. The detection of tampering is possible by allowing additional redundancy in the message by higher level applications as discussed by Bellare and Rogaway [2].

A *Tweakable Enciphering Scheme* (TES) is a keyed family of length preserving transformations $\mathcal{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ where $\mathcal{K}$ and $\mathcal{T}$ are the finite and non-empty set of keys and tweaks respectively and $\mathcal{M}$ is a message space such that for all $K \in \mathcal{K}$ and all $T \in \mathcal{T}$, $\mathcal{E}_K(T, \cdot)$ is a length preserving permutation [1] over $\mathcal{M}$ and there must be an inverse $\mathcal{D}_K(T, \cdot)$ to $\mathcal{E}_K(T, \cdot)$. Unlike the key K, tweak T is public whose sole purpose is to introduce the variability of the ciphertext, similar to that of the role of IV in the mode of encryption.

The general security notion of a TES is tweakable strong pseudorandom permutation (TSPRP) which is to say that it is computationally infeasible for an adversary to distinguish the oracle that maps (T, M) into $\mathcal{E}_K(T, M)$ and maps (T, C) into $\mathcal{D}_K(T, C)$ when the key K is random and secret from an oracle that realizes a T-indexed family of random permutations and their inverses. A TSPRP secure TES is a desirable tool for solving the disk encryption problem as pointed out in [14] where the sector address of the disk plays the role of the tweak in TES.

1.1 Different Paradigm of Designing TES

In the past few years there have been various proposals of designing TES. If we categorize all these proposals, then we see that all the proposals falls under one of the following three categories:

Hash-Encrypt-Hash. Naor and Reingold [28] designed a wide block SPRP using a invertible ECB mode of encryption sandwiched between two invertible pairwise independent hash functions. This paradigm of construction is known as Hash-Encrypt-Hash. However, as discussed in [14] that the description given in [28] is at a top level and also the latter work [27] does not fully specify a mode of operation. Moreover, the construction was not a tweakable SPRP. Later in 2006, Chakraborty and Sarkar [8] first instantiated Hash-Encrypt-Hash mode with PEP by sandwiching a ECB type encryption layer in between of two layers of

[1] A length preserving permutation over $\mathcal{M}$ is a permutation π such that for all $M \in \mathcal{M}$, $|\pi(M)| = |M|$.

polynomial hashing. TET, a more efficient version of PEP, was later proposed by Halevi [13]. HEH, an improvement upon TET, is also reported in [34].

Encrypt-Mix-Encrypt. CBC-Mix-CBC (CMC), proposed by Halevi and Rogaway [14], is the first TES construction in which a mixing layer is sandwiched between two CBC layers; hence the design is inherently sequential. Later, Halevi and Rogaway proposed a parallel construction, called EME [15] in which the encryption layers are of ECB type. Later EME was extended to EME* [12] for handling arbitrary length messages. All of these constructions follow the same design principle where a simple mixing layer is sandwiched between two invertible encryption layers. Recently, Bhaumik et al. [3] proposed FMix, a variant of CMC, that uses a single block cipher key (instead of two block cipher keys used in CMC) and lifted up the requirement of the block cipher invertibilty.

Hash-Counter-Hash. This paradigm is similar to the Hash-Encrypt-Hash, but instead of a ECB layer, a counter mode encryption layer is sandwiched between two almost-xor universal hash function [2]. The advantage of using the counter mode encryption is to tackle the variable length messages easily. XCB [20] is the first Hash-Counter-Hash type construction that requires 5 block cipher keys and two block cipher calls (apart from block cipher calls in counter mode encryption). Later, Wang et al. proposed HCTR [35] with a single block cipher key and removed one extra invocation of block cipher call. FAST, a pseudorandom function (PRF) based TES construction following the Hash-Counter-Hash paradigm has recently been proposed by Chakraborty et al. [5].

Amongst the above mentioned constructions, only CMC and EME* are block cipher based constructions with a light weight masking layer in between of two encryption layers, whereas the other two paradigms require the field multiplication (as a part of the hash function evaluation) along with the block cipher evaluation. Thus, the only significant cost for Encrypt-Mix-Encrypt type constructions are the block cipher calls, whereas for the other two paradigms the cost involved in both evaluating the block cipher calls and the finite field multiplications. A detailed comparison of the performance and efficiency of different TES can be found in [7, 13, 34]. This comparison study along with [19] suggests that HCTR is one of the most efficient candidates amongst all proposed TES.

However, unlike other TES proposals which have the usual "**birthday bound**" type security, HCTR was initially shown to have the **cubic** security bound [35]. Later, the bound was improved to the birthday bound by Chakraborty and Nandi [6]. Chakraborty and Sarkar [7] proposed HCH, a simple variant of HCTR, in which they introduce one more block cipher call before initializing the counter and shown to have the birthday bound security.

[2] An almost-xor universal hash function is a keyed hash function such that for any two distinct messages, the probability, over the random draw of a hash key, the hash differential being equal to a specific output is small.

1.2 Our Contribution

In this paper, we propose **tweakable HCTR**, a variant of the HCTR construction, that yields a variable input length *tweakable block cipher* (TBC) [3] from a fixed input length tweakable block cipher, in which all the block ciphers of HCTR are replaced with TBC. In HCTR, the tweak is one of the inputs of the upper and lower layer hash function (i.e., H_{K_h} in Fig. 3.1), but in our construction, we process the tweak through another independent keyed $(n+m)$-bit hash function H'_L where the m-bit hash value becomes the tweak of the underlying tweakable block cipher and the remaining n-bit hash output is used to mask the input and the output of the leftmost TBC (see Fig. 1.1). We process tweak through an indepedent keyed hash function for allowing large sized tweaks.

We have shown that if there is no repetition of tweaks, or in other words, all the queried tweaks are distinct, then tweakable HCTR is secure upto 2^n many message blocks against any computationally unbounded chosen plaintext chosen ciphertext adaptive adversaries. Moreover, when the repetition of the tweak is limited, then the security we obtain is close to the optimal one. This is in contrast to the security of other nonce based constructions (e.g., Wegman-Carter MAC [4], AES-GCM [21] etc.) where a single time repetition of the nonce completely breaks the scheme. This property is called the **graceful degradation of security** when tweak repeats. Gracefully degrading secured construction based on tweakable block ciphers has been studied in [32] and the notion of tweak repetition has been studied in [22] by Mennink for proving $3n/4$-bit security of CLRW2. In [22], Mennink stated that:

"The condition on the occurrence of the tweak seems restrictive, but many modes of operation based on a tweakable block cipher query their primitives for tweaks that are constituted of a nonce or random number concatenated with a counter value: in a nonce-respecting setting, every nonce appears at most $1 + q_f$ *times, where* q_f *is the amount of forgery attempts."*

In practical settings like disk-encryption problem where the sector address plays the role of the tweak, tweak is not repeated arbitrarily and therefore the security of any tweakable scheme where the tweak repeats in a limited way, is worth to study.

1.3 Comparison with Minematsu-Iwata Proposal [24]

Hash-Sum-Expansion or (HSE) due to Minematsu and Matsushima [26] is a generic structure that underlies the construction of HCTR and HCH. HSE is instantiated with a TBC and a weak pseudorandom function (wPRF) [26] and its security proof shows that the expansion function of HCTR and HCH, which is achieved through the counter mode encryption, can be instantiated with any secure wPRF. However, HSE is shown to have the birthday bound security. Later,

[3] A tweakable block cipher is basically a simple block cipher with an additional parameter called tweak.

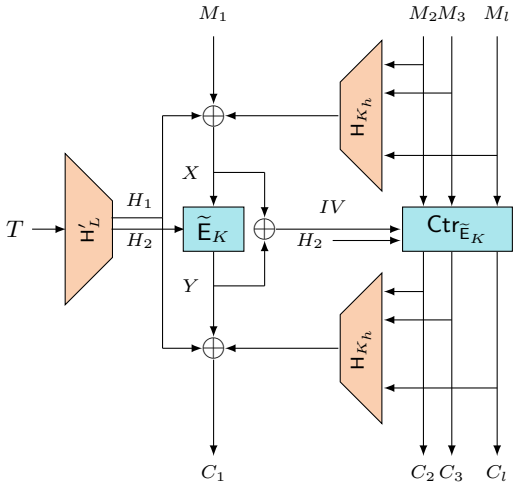

Fig. 1.1. Tweakable HCTR construction with tweak T and message $M_1\|M_2\|\ldots\|M_l$ and the corresponding ciphertext $C_1\|C_2\|\ldots\|C_l$. H_{K_h} is an n-bit almost-xor universal and almost regular hash function with hash key K_h. H'_L is an $(n+m)$-bit partial almost xor universal hash function with hash key L and $\mathsf{H}'_L(T) = (H_1, H_2)$, where H_1 is of size n bits and H_2 is of size m bits. $\widetilde{\mathsf{E}}_K$ is the tweakable block cipher and $\mathsf{Ctr}_{\widetilde{\mathsf{E}}_K}$ is the tweakable block cipher based counter mode encryption.

Minematsu and Iwata [24] designed a block cipher for processing arbitrary length messages. For processing messages of shorter length than $2n$ bits, they proposed **Small-Block Cipher**, which is instantiated with two independent keyed TBCs with tweak size (m) < block size (n) and an n-bit PolyHash function Poly_{K_h} which eventually provides sprp security upto $(n+m)/2$ bits [4]. The construction is identical to a scheme of [23]. To process messages larger than $2n$ bits, they proposed **Large Block-Cipher, Method 1** and **Large Block-Cipher, Method-2**. The former one is structurally similar to HCTR and hence is of interest to us. LBC-1 (abbreviation for Large Block-Cipher, Method 1) uses (a) a $2n$-bit block cipher E_{2n}, (b) a $2n$-bit keyed hash function H_K in upper and lower layer and (c) a wPRF F. It has been shown [24] that LBC-1 provides the optimal (i.e., 2^n) sprp security, where block size and tweak size is of n bits.

Now, to instantiate each of the primitives, (a) E_{2n} is instantiated through **Small-Block Cipher** method and hence it requires two independent keyed TBCs with tweak size and block size n and an n-bit PolyHash function. (b) $2n$-bit keyed hash function H_K is instantiated through the concatenation of two independent keyed n-bit PolyHash functions and (c) the wPRF F is instantiated through a counter mode of encryption based on two independent invocations of TBCs with tweak size and block size n. Therefore, LBC-1 requires altogether

[4] This security bound is beyond birthday in terms of the block size n, but with respect to the input size of TBC (i.e., $n+m$ bits), it is the birthday bound.

two independent keyed TBCs with n-bits tweak and block along with three independent keyed n-bit PolyHash functions. In contrast to this, our proposal requires an n-bit almost xor universal hash function (e.g., polyhash) in upper and lower layers, an $(n+m)$-bit partial-almost xor universal hash function [5] [25, 16] and a single instance of a TBC with tweak size m. Note that, in our case the tweak is provided as an additional input to the construction unlike to LBC-1 where the part of the input message is served as a tweak to the underlying TBC.

2 Preliminaries

Basic Notations. For a set $\mathcal{X}$, $X \leftarrow_{\$} \mathcal{X}$ denotes that X is sampled uniformly at random from $\mathcal{X}$ and independent of all other random variables defined so far. For two sets $\mathcal{X}$ and $\mathcal{Y}$, $\mathcal{X} \sqcup \mathcal{Y}$ denotes the disjoint union, i.e, when there is no common elements in $\mathcal{X}$ and $\mathcal{Y}$. $\{0,1\}^n$ denotes the set of all binary strings of length n and $\{0,1\}^*$ denotes the set of all binary strings of arbitrary length. 0^i denotes the string of length i with all bits zero. For any element $X \in \{0,1\}^*$, $|X|$ denotes the number of bits of X. For any two elements $X, Y \in \{0,1\}^*$, $X\|Y$ denotes the concatenation of X followed by Y. For $X, Y \in \{0,1\}^n$, we write $X \oplus Y$ to denote the xor of X and Y. For any $X \in \{0,1\}^*$, we parse X as $X = X_1\|X_2\|\dots\|X_l$ where for each $i = 1, \dots, l-1$, X_i is an element of $\{0,1\}^n$ and $1 \le |X_l| \le n$. We call each X_i a *block*. When there is a sequence of elements $X_1, X_2, \dots, X_s \in \{0,1\}^*$, we write X_a^i to denote the a-th block of the i-th element X_i. For any integer j, $\langle j \rangle$ denotes the n-bit binary representation of integer j. For integers $1 \le b \le a$, we write $(a)_b$ to denote $a(a-1)\dots(a-b+1)$, where $(a)_0 = 1$ by convention. We write $[q]$ to refer to the set $\{1, \dots, q\}$.

For a function $\Phi : \mathcal{X} \to \mathcal{Y}_1 \times \mathcal{Y}_2$, we write $\Phi(x) = (\phi_1, \phi_2)$ for all $x \in \mathcal{X}$. $\Phi[1]$ is the function from $\mathcal{X}$ to $\mathcal{Y}_1$ such that for all $x \in \mathcal{X}$, $\Phi[1](x) = \phi_1$. Similarly, $\Phi[2]$ is a function from $\mathcal{X}$ to $\mathcal{Y}_2$ such that $\Phi[2](x) = \phi_2$ for all $x \in \mathcal{X}$.

Block Ciphers. A *block cipher* (BC) with key space $\mathcal{K}$ and domain $\mathcal{X}$ is a mapping $\mathsf{E} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$ such that for all key $K \in \mathcal{K}$, $X \mapsto \mathsf{E}(K, X)$ is a permutation of $\mathcal{X}$. We denote $\mathsf{BC}(\mathcal{K}, \mathcal{X})$ the set of all block ciphers with key space $\mathcal{K}$ and domain $\mathcal{X}$. A *permutation* Π with domain $\mathcal{X}$ is a bijective mapping of $\mathcal{X}$ and $\mathsf{Perm}(\mathcal{X})$ denotes the set of all permutations over $\mathcal{X}$. $\mathsf{E} \in \mathsf{BC}(\mathcal{K}, \mathcal{X})$ is said to be a strong pseudorandom permutation or equivalently a strong block cipher if the sprp advantage of E against any chosen plaintext chosen ciphertext adaptive adversary A with oracle access to a permutation and its inverse with domain $\mathcal{X}$, defined as follows

$$\mathbf{Adv}^{\mathrm{SPRP}}_{\mathsf{E}}(\mathsf{A}) := |\Pr[K \leftarrow_{\$} \mathcal{K} : \mathsf{A}^{\mathsf{E}_K, \mathsf{E}_K^{-1}} = 1] - \Pr[\Pi \leftarrow_{\$} \mathsf{Perm}(\mathcal{X}) : \mathsf{A}^{\Pi, \Pi^{-1}} = 1]| \tag{1}$$

[5] Informally, a keyed hash function is said to be a partial-almost xor universal hash function, if for any two distinct inputs, the probability over the random draw of the hash key, that the first n-bit part of the sum of their hash ouput takes any value and the remaining m-bit part of the hash value collides, is very small.

that makes at most q queries with maximum running time t, is very small. When the adversary is given access only to the permutation and not its inverse, then we say the PRP advantage of A against E.

Tweakable Block Ciphers. A *tweakable block cipher* (TBC) with key space $\mathcal{K}$, tweak space $\mathcal{T}$ and domain $\mathcal{X}$ is a mapping $\widetilde{\mathsf{E}} : \mathcal{K} \times \mathcal{T} \times \mathcal{X} \to \mathcal{X}$ such that for all key $k \in \mathcal{K}$ and all tweak $t \in \mathcal{T}$, $x \mapsto \widetilde{\mathsf{E}}(k, t, x)$ is a permutation of $\mathcal{X}$. We often write $\widetilde{\mathsf{E}}_k(t, x)$ or $\widetilde{\mathsf{E}}_k^t(x)$ for $\widetilde{\mathsf{E}}(k, t, x)$. We call a tweakable block cipher as (m, n) tweakable block cipher if $\mathcal{T} = \{0,1\}^m$ and $\mathcal{X} = \{0,1\}^n$. We denote $\mathsf{TBC}(\mathcal{K}, \mathcal{T}, \mathcal{X})$ the set of all such (m, n) tweakable block ciphers with key space $\mathcal{K}$, tweak space $\mathcal{T}$ and domain $\mathcal{X}$. A *tweakable permutation* with tweak space $\mathcal{T}$ and domain $\mathcal{X}$ is a mapping $\widetilde{\Pi} : \mathcal{T} \times \mathcal{X} \to \mathcal{X}$ such that for all tweak $T \in \mathcal{T}$, $X \mapsto \widetilde{\Pi}(T, X)$ is a permutation of $\mathcal{X}$. We often write $\widetilde{\Pi}^T(X)$ for $\widetilde{\Pi}(T, X)$. $\mathsf{TP}(\mathcal{T}, \mathcal{X})$ denotes the set of all (m, n) tweakable permutations with tweak space $\mathcal{T}(= \{0,1\}^m)$ and domain $\mathcal{X}(= \{0,1\}^n)$.

Adversarial Model for TBC. An adversary A for TBC has access to either of the pair of oracles $(\widetilde{\mathsf{E}}_K(\cdot,\cdot), \widetilde{\mathsf{E}}_K^{-1}(\cdot,\cdot))$ for some fixed key $K \in \mathcal{K}$ or access to the pair of oracles $(\widetilde{\Pi}(\cdot,\cdot), \widetilde{\Pi}^{-1}(\cdot,\cdot))$ oracles for some $\widetilde{\Pi} \in \mathsf{TP}(\mathcal{T}, \mathcal{X})$. Adversary A queries to the pair of oracles in an interleaved and adaptive way and after the interaction is over, it outputs a single bit b. We assume that A can query any tweak for at most μ times in all its encryption and decryption queries, which is called the *maximum tweak multiplicity*, i.e., if $\mu = 1$ then each queried tweak is distinct. Moreover, we assume that A does not repeat any query to the encryption or the decryption oracle. We also assume that A does not query the decryption oracle (resp. the encryption oracle) with the value that it obtained as a result of a previous encryption query (resp. decryption query). We call such an adversary A, a *non-trivial* (μ, q, t) chosen plaintext chosen ciphertext adaptive adversary, where A makes total q many encryption and decryption queries with running time at most t and maximum tweak multiplicity μ. Sometimes we write $(\mu, q, \ell, \sigma, t)$ chosen plaintext chosen ciphertext adaptive adversary A to emphasize that the maximum number of message blocks in a queried message of A is ℓ and the total number of message blocks that A can query is σ. When the parameters $\ell = \sigma = 0$, then we simply write (μ, q, t).

Definition 1 (TSPRP Security). *Let $\widetilde{\mathsf{E}} \in \mathsf{TBC}(\mathcal{K}, \mathcal{T}, \mathcal{X})$ be a tweakable block cipher and A be a non-trivial (μ, q, t) chosen plaintext chosen ciphertext adaptive adversary with oracle access to a tweakable permutation and its inverse with tweak space $\mathcal{T}$ and domain $\mathcal{X}$. The advantage of A in breaking the* TSPRP *security of $\widetilde{\mathsf{E}}$ is defined as*

$$\mathbf{Adv}^{\mathrm{TSPRP}}_{\widetilde{\mathsf{E}}}(\mathsf{A}) := |\Pr[K \leftarrow_{\$} \mathcal{K} : \mathsf{A}^{\widetilde{\mathsf{E}}_K, \widetilde{\mathsf{E}}_K^{-1}} = 1] - \Pr[\widetilde{\Pi} \leftarrow_{\$} \mathsf{TP}(\mathcal{T}, \mathcal{X}) : \mathsf{A}^{\widetilde{\Pi}, \widetilde{\Pi}^{-1}} = 1]|, \tag{2}$$

where the adversary queries with tweak $T \in \mathcal{T}$ and input $X \in \mathcal{X}$. When the adversary is given access only to the tweakable permutation and not its inverse, then we say the tweakable pseudorandom permutation (TPRP) *advantage of* A

against $\widetilde{\mathsf{E}}$. *Informally,* $\widetilde{\mathsf{E}}$ *is said to be a tweakable strong pseudorandom permutation or equivalently a tweakable strong block cipher when the* TSPRP *advantage of* $\widetilde{\mathsf{E}}$ *against any adversary* A *that makes at most* q *queries with maximum running time* t*, as defined in Eqn.* (2)*, is very small.*

Almost (XOR) Universal and Almost Regular Hash Function. Let $\mathcal{K}_h, \mathcal{X}$ be two non-empty finite sets and H be an n-bit keyed function $\mathsf{H} : \mathcal{K}_h \times \mathcal{X} \to \{0,1\}^n$. Then,

- H is said to be an ϵ-almost xor universal (AXU) hash function if for any distinct $X, X' \in \mathcal{X}$ and for any $Y \in \{0,1\}^n$,

$$\Pr[K_h \leftarrow\!\!\$\, \mathcal{K}_h : \mathsf{H}_{K_h}(X) \oplus \mathsf{H}_{K_h}(X') = Y] \leq \epsilon. \tag{3}$$

  As a special case, when $Y = 0^n$, then H is said to be an ϵ-almost universal (AU) hash function.
- H is said to be an ϵ-almost regular hash function if for any $X \in \mathcal{X}$ and for any $Y \in \{0,1\}^n$,

$$\Pr[K_h \leftarrow\!\!\$\, \mathcal{K}_h : \mathsf{H}_{K_h}(X) = Y] \leq \epsilon. \tag{4}$$

It is easy to see that PolyHash with an n-bit key, as defined in [24, 11], is an $\ell/2^n$-AXU and $\ell/2^n$-almsot regular hash function, where ℓ is the maximum number of message blocks. Proof of this result can be found in [11].

Partial Almost (XOR) Universal Hash Function. Let $\mathcal{K}_h, \mathcal{X}$ be two non-empty finite sets and H be an $(n+m)$-bit keyed function $\mathsf{H} : \mathcal{K}_h \times \mathcal{X} \to \{0,1\}^n \times \{0,1\}^m$. Then, H is said to be an (n, m, ϵ)-partial almost xor universal (pAXU) hash function if for any distinct $X, X' \in \mathcal{X}$ and for any $Y \in \{0,1\}^n$,

$$\Pr[K_h \leftarrow\!\!\$\, \mathcal{K}_h : \mathsf{H}_{K_h}(X) \oplus \mathsf{H}_{K_h}(X') = (Y, 0^m)] \leq \epsilon. \tag{5}$$

Note that, an ϵ-AXU $(n+m)$-bit keyed hash function is an (n, m, ϵ)-pAXU. We write $\mathsf{H}_{K_h}(X) = (H_1, H_2)$, where $H_1 \in \{0,1\}^n$ and $H_2 \in \{0,1\}^m$.

3 Specification and Security Result of Tweakable HCTR

HCTR, as proposed by Wang et al. [35], is a mode of operation which turns an n-bit strong prp into a tweakable strong prp that supports arbitrary and variable length input and tweak which is no less than n bits. For any message $M \in \{0,1\}^*$ and a tweak T, HCTR works as follows: it first parses the message M into l many blocks such that its first $l-1$ message blocks are of length n-bits and the length of the last block is at most n. Then, it applies an n-bit PolyHash function on the string $M_2 \| \dots M_l \| T$ and xor its n-bit output value with the first message block M_1 to produce X. This X is then feeded into an n-bit block cipher E whose output Y is xor-ed with X to produce an IV value which acts a counter in the counter mode encryption to produce the ciphertext blocks $C_2 \| \dots \| C_l$. Finally, the first ciphertext block C_1 is generated by applying the same PolyHash on

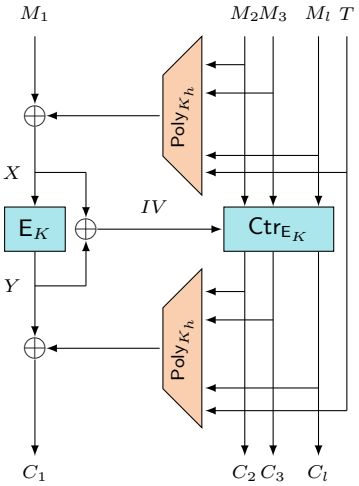

Fig. 3.1. HCTR construction with tweak T and message $M_1\|M_2\|\ldots\|M_l$ and the corresponding ciphertext $C_1\|C_2\|\ldots\|C_l$. Poly_{K_h} is the polynomial hash function with hash key K_h. $\mathsf{Ctr}_{\mathsf{E}_K}$ is the block cipher based counter mode of encryption.

$C_2\|\ldots\|C_l\|T$ and xor its output with Y. Schematic diagram of HCTR is shown in Fig. 3.1.

Wang et al. [35] have shown HCTR to be a secure TES against all adaptive chosen plaintext and chosen ciphertext adversaries that make roughly $2^{n/3}$ encryption and decryption queries. Later in FSE 2008, Chakraborty and Nandi [6] have improved its security bound to $O(\sigma^2/2^n)$.

3.1 Specification of Tweakable HCTR

Our proposal Tweakable HCTR, which we denote as $\widetilde{\mathsf{HCTR}}$, closely resembles to the original HCTR with the exception that (i) the strong block cipher of HCTR is replaced by a (m, n) tweakable strong block cipher, where m is the size of the block cipher tweak and n is the block size of the TBC and (ii) the tweak used for the construction, which is processed through the upper and lower hash function in HCTR, is now processed through an independent keyed $(n + m)$-bit partial AXU hash function whose n-bit output is masked with the input and the output of the leftmost tweakable block cipher and the remaining m-bit output plays the role of the tweak of the underlying TBC. Moreover, all the block cipher calls of the counter mode encryption used in HCTR are replaced by TBCs where the same m-bit hash value of the tweak becomes the tweak of the underlying tweakable block cipher used in the tweakable counter mode of encryption. Schematic diagram of the construction is shown in Fig. 1.1 and its algorithmic description is shown in Fig. 3.2.

$\mathsf{Enc}.\widetilde{\mathsf{HCTR}}[\widetilde{\mathsf{E}}_K, \mathsf{H}_{K_h}, \mathsf{H}'_L](T, M)$	$\mathsf{Dec}.\widetilde{\mathsf{HCTR}}[\widetilde{\mathsf{E}}_K, \mathsf{H}_{K_h}, \mathsf{H}'_L](T, C)$
1. $(H_1, H_2) \leftarrow \mathsf{H}'_L(T)$	1. $(H_1, H_2) \leftarrow \mathsf{H}'_L(T)$
2. $X \leftarrow H_1 \oplus M_1 \oplus \mathsf{H}_{K_h}(M_2 \| M_3 \| \ldots \| M_l)$	2. $Y \leftarrow C_1 \oplus \mathsf{H}_{K_h}(C_2 \| C_3 \| \ldots \| C_l) \oplus H_1$
3. $Y \leftarrow \widetilde{\mathsf{E}}_K(H_2, X)$	3. $X \leftarrow \widetilde{\mathsf{E}}_K^{-1}(H_2, Y)$
4. $IV \leftarrow X \oplus Y$	4. $IV \leftarrow X \oplus Y$
5. **for** $j = 2$ to l:	5. **for** $j = 2$ to l:
6. $\quad C_j \leftarrow M_j \oplus \widetilde{\mathsf{E}}_K(H_2, IV \oplus \langle j \rangle)$	6. $\quad M_j \leftarrow C_j \oplus \widetilde{\mathsf{E}}_K(H_2, IV \oplus \langle j \rangle)$
7. $C_1 \leftarrow Y \oplus \mathsf{H}_{K_h}(C_2 \| C_3 \| \ldots \| C_l) \oplus H_1$	7. $M_1 \leftarrow X \oplus \mathsf{H}_{K_h}(M_2 \| M_3 \| \ldots \| M_l) \oplus H_1$
8. **return** $(C_1 \| C_2 \| \ldots \| C_l)$	8. **return** $(M_1 \| M_2 \| \ldots \| M_l)$

Fig. 3.2. Tweakable HCTR Construction. Left part is the encryption algorithm of tweakable HCTR and right part is its decryption algorithm. $\langle j \rangle$ denotes the n-bit binary representation of integer j. H is an n-bit almost xor universal hash function and H' is an $(n+m)$-bit partial almost xor universal hash function.

As can be seen from the algorithm there are three basic building blocks used in the construction of $\widetilde{\mathsf{HCTR}}$; an n-bit keyed AXU hash function H, an $(n+m)$-bit keyed pAXU hash function H' and a tweakable counter mode of encryption.

Given an n-bit string IV, we define a sequence $(IV_1, \ldots, IV_l)$, where each IV_i is some function of IV. Given such a sequence $(IV_1, \ldots, IV_l)$, a key K, a message $M = M_1 \| M_2 \| \ldots \| M_l$ (for simplicity we assume that $|M|$ is a multiple of n) and the hash value of an $(n+m)$-bit keyed pAXU hash function of the tweak T (i.e., $\mathsf{H}'_L(T)$), the tweakable counter mode is defined as follows:

$$\mathsf{Ctr}_{\widetilde{\mathsf{E}}_K^{H_2}, IV}(M_1, \ldots, M_l) = \Big(M_1 \oplus \widetilde{\mathsf{E}}_K(H_2, IV_1), \ldots, M_l \oplus \widetilde{\mathsf{E}}_K(H_2, IV_l) \Big),$$

where $IV_i = IV \oplus \langle i \rangle$ and $\mathsf{H}'_L(T) = (H_1, H_2)$. In case the last block M_l is incomplete then $M_l \oplus \widetilde{\mathsf{E}}_K(H_2, IV_l)$ is replaced by $M_l \oplus \mathsf{drop}_r(\widetilde{\mathsf{E}}_K(H_2, IV_l))$, where $r = n - |M_l|$ and $\mathsf{drop}_r(\widetilde{\mathsf{E}}_K(H_2, IV_l))$ is the first $(n-r)$ bits of $\widetilde{\mathsf{E}}_K(H_2, IV_l)$. If $l = 1$ (when we have one block message), we ignore line 4 and 5 of both the encryption and the decryption algorithm of $\widetilde{\mathsf{HCTR}}$ construction.

3.2 Security Result of Tweakable HCTR

In this section, we state the security result of $\widetilde{\mathsf{HCTR}}$. In specific, we state that if $\widetilde{\mathsf{E}}$ is a (m, n) tweakable strong block cipher, H is an ϵ-axu n-bit keyed hash function, H' is a δ-partial AXU $(n+m)$-bit keyed hash function, and $\mathsf{H}'[2]$ is a δ_{au} almost universal m-bit keyed hash function, then $\widetilde{\mathsf{HCTR}}$ is a secure TES against all $(\mu, q, \ell, \sigma, t)$ chosen plaintext and chosen ciphertext adaptive adversaries that make roughly $2^n/\mu\ell$ many encryption and decryption queries, where ℓ is the maximum number of message blocks among all q queries and σ is the total

number of message blocks queried. Formally, the following result bounds the tsprp advantage of $\widetilde{\mathsf{HCTR}}$.

Theorem 1. *Let $\mathcal{M}, \mathcal{T}, \mathcal{K}$, $\mathcal{K}_h$ and $\mathcal{L}$ be finite and non-empty sets. Let $\widetilde{\mathsf{E}} : \mathcal{K} \times \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ be a (m,n) tweakable strong block cipher, $\mathsf{H} : \mathcal{K}_h \times \mathcal{M} \to \{0,1\}^n$ be an ϵ-AXU and ϵ_1-almost regular n-bit keyed hash function and $\mathsf{H}' : \mathcal{L} \times \mathcal{T} \to \{0,1\}^n \times \{0,1\}^m$ be an (n,m,δ)-partial AXU $(n+m)$-bit keyed hash function and $\mathsf{H}'[2]$ is a δ_{au}-almost univeral m-bit keyed hash function. Then, for any $(\mu, q, \ell, \sigma, t)$ chosen plaintext chosen ciphertext adaptive adversary A against the tsprp security of $\widetilde{\mathsf{HCTR}}[\widetilde{\mathsf{E}}, \mathsf{H}, \mathsf{H}']$, there exists a (μ, σ, t') chosen plaintext chosen ciphertext adaptive adversary A' against the tsprp security of $\widetilde{\mathsf{E}}$, where $t' = O(t + \sigma + q(2t_{\mathsf{H}} + t_{\mathsf{H}'}))$, σ is the total number of message blocks queried, t_{H} be the time for computing the hash function H, $t_{\mathsf{H}'}$ be the time for computing the hash function H' and $\mu \leq \min\{|\mathcal{T}|, q\}$, such that*

$$\mathbf{Adv}^{\mathrm{TSPRP}}_{\widetilde{\mathsf{HCTR}}[\widetilde{\mathsf{E}},\mathsf{H},\mathsf{H}']}(\mathsf{A}) \leq \mathbf{Adv}^{\mathrm{TSPRP}}_{\widetilde{\mathsf{E}}}(\mathsf{A}') + 2(\mu-1)(q\epsilon + \sigma/2^n) + 2q\sigma\delta_{\mathsf{au}}/2^n + q^2\delta + 2\max\{q\ell(\mu-1)/2^n + q\sigma\delta_{\mathsf{au}}/2^n, \sigma\epsilon_1\}.$$

By assuming $\epsilon, \epsilon_1 \approx 2^{-n}$, $\delta_{\mathsf{au}} \approx 2^{-m}, \delta \approx 2^{-(n+m)}$ and $m > n$, $\widetilde{\mathsf{HCTR}}$ is secured roughly upto $2^n/\mu\ell$ queries. Moreover, when all the tweaks are distnct, i.e., $\mu = 1$, then the tsprp security of $\widetilde{\mathsf{HCTR}}$ becomes

$$\mathbf{Adv}^{\mathrm{TSPRP}}_{\widetilde{\mathsf{HCTR}}[\widetilde{\mathsf{E}},\mathsf{H},\mathsf{H}']}(\mathsf{A}) \leq \mathbf{Adv}^{\mathrm{TSPRP}}_{\widetilde{\mathsf{E}}}(\mathsf{A}') + 2(\sigma\epsilon_1 + q\sigma\delta_{\mathsf{au}}/2^n) + q^2\delta.$$

Therefore, when all the tweaks in the encryption and decryption queries are distinct, then by assuming $\epsilon, \epsilon_1 \approx 2^{-n}$, $\delta_{\mathsf{au}} \approx 2^{-m}, \delta \approx 2^{(n+m)}$ and $m > n$, $\widetilde{\mathsf{HCTR}}$ is secured roughly upto 2^n many message blocks.

4 Proof of Theorem 1

In this section, we prove Theorem 1. We would like to note that we will often refer to the construction $\widetilde{\mathsf{HCTR}}[\widetilde{\mathsf{E}}, \mathsf{H}, \mathsf{H}']$ as simply $\widetilde{\mathsf{HCTR}}$ when the underlying primitives are assumed to be understood.

As the first step of the proof, we replace $\widetilde{\mathsf{E}}_K$ with an (m,n)-bit tweakable uniform random permutation $\widetilde{\Pi}$ and denote the resulting construction as $\widetilde{\mathsf{HCTR}}^*[\widetilde{\Pi}, \mathsf{H}, \mathsf{H}']$. It is easy to show that there exists an adversary against the tsprp security of $\widetilde{\mathsf{E}}$, making at most σ oracle queries and running in time at most $O(t+\sigma+q(2t_{\mathsf{H}}+t_{\mathsf{H}'}))$ with maximum tweak multiplicity μ, such that

$$\mathbf{Adv}^{\mathrm{TSPRP}}_{\widetilde{\mathsf{HCTR}}[\widetilde{\mathsf{E}},\mathsf{H},\mathsf{H}']}(\mathsf{A}) \leq \mathbf{Adv}^{\mathrm{TSPRP}}_{\widetilde{\mathsf{E}}}(\mathsf{A}') + \underbrace{\mathbf{Adv}^{\mathrm{TSPRP}}_{\widetilde{\mathsf{HCTR}}^*[\widetilde{\Pi},\mathsf{H},\mathsf{H}']}(\mathsf{A})}_{\delta^*}. \tag{6}$$

Now, our goal is to upper bound δ^*. For doing this, we first describe how the ideal oracle works. Let us assume that $n\ell$ be the maximum size of any message

M among all q many queries. Let $\mathcal{S}_i$ denotes the set of all binary strings of length i. Therefore, $\{0,1\}^{\leq n\ell}$, which denotes the set of all binary strings of length at most $n\ell$, can be written as $\mathcal{S}_1 \sqcup \mathcal{S}_2 \sqcup \ldots \sqcup \mathcal{S}_{n\ell}$. Now, for the i-th encryption or decryption query, the ideal oracle works as shown in Fig. 4.1.

Ideal oracle ($\$$) for Encryption On i^{th} input (T_i, M_i)	Ideal oracle ($\$^{-1}$) for Decryption On i^{th} input (T_i, C_i)
1. **if** $T_i = T_a$ `for some` $a \in [c]$	1. **if** $T_i = T_a$ `for some` $a \in [c]$
2. `if` $M_i \in \mathcal{D}_a$, `let` $M_i = M_j$ **for some** $j < i$	2. `if` $C_i \in \mathcal{R}_a$, `let` $C_i = C_j$ **for some** $j < i$
3. `then` $C_i \leftarrow C_j$	3. `then` $M_i \leftarrow M_j$
4. `else` $C_i \leftarrow_{\$} \mathcal{S}_{l_i} \setminus \mathcal{R}_a$	4. `else` $M_i \leftarrow_{\$} \mathcal{S}_{l_i} \setminus \mathcal{D}_a$
5. $\mathcal{D}_a = \mathcal{D}_a \cup \{M_i\}; \mathcal{R}_a = \mathcal{R}_a \cup \{C_i\}$	5. $\mathcal{D}_a = \mathcal{D}_a \cup \{M_i\}; \mathcal{R}_a = \mathcal{R}_a \cup \{C_i\}$
6. **else**	6. **else**
7. $c \leftarrow c+1; T_c \leftarrow T_i$	7. $c \leftarrow c+1; T_c \leftarrow T_i$
8. $C_i \leftarrow_{\$} \mathcal{S}_{l_i}$	8. $M_i \leftarrow_{\$} \mathcal{S}_{l_i}$
9. $\mathcal{D}_c = \mathcal{D}_c \cup \{M_i\}; \mathcal{R}_c = \mathcal{R}_c \cup \{C_i\}$	9. $\mathcal{D}_c = \mathcal{D}_c \cup \{M_i\}; \mathcal{R}_c = \mathcal{R}_c \cup \{C_i\}$
10. **return** C_i	10. **return** M_i

Fig. 4.1. Left part is the encryption algorithm of the ideal oracle and the right part is the decryption algorithm of the ideal oracle. c is the number of equivalent classes over the queried tweak space until the i-th query. $\mathcal{D}_a$ denotes the set of all already sampled output (for decryption) and queried input (for encryption) for a-th equivalent class and $\mathcal{R}_a$ denotes the set of all already sampled output (for encryption) and queried input (for decryption) for a-th equivalent class. l_i denotes the length of the i-th plaintext M_i, for encryption or the i-th ciphertext C_i for decryption.

In words, for the ith encryption query (T_i, M_i), the ideal oracle $\$$ first checks if the tag T_i matches with some previous existing tags. If so, then it samples the ciphertext C_i without replacement from the set of all binary strings of length $|M_i|$; otherwise, it samples the C_i uniformly at random from $\mathcal{S}_{|M_i|}$. Decryption oracle also works in the similar way, except that the oracle samples the plaintext instead of ciphertext. Since, we have assumed the distinguisher is non-trivial, line 2-3 of both the algorithm will not be executed. Therefore, we write

$$\delta^* \leq \max_{\mathsf{D}} \Pr[\mathsf{D}^{\mathsf{Enc}.\widetilde{\mathsf{HCTR}}^*, \mathsf{Dec}.\widetilde{\mathsf{HCTR}}^*} = 1] - \Pr[\mathsf{D}^{\$,\$^{-1}} = 1],$$

where the maximum is taken over all non-trivial distinguishers D that make total q many encryption and decryption queries with at most σ many blocks such that the maximum number of message blocks among all the queried messages is ℓ and the maximum tweak multiplicity μ. This formulation allows us to apply the H-

Coefficient Technique [31, 30], as we explain in more detail below, to prove

$$\delta^* \leq 2(\mu-1)(q\epsilon+\sigma/2^n)+2q\sigma\delta_{\mathsf{au}}/2^n+q^2\delta+2\max\{q\ell(\mu-1)/2^n+q\sigma\delta_{\mathsf{au}}/2^n, \sigma\epsilon_1\}. \tag{7}$$

H-Coefficient Technique. From now on, we fix a non-trivial distinguisher D that interacts with either (1) the real oracle $(\mathsf{Enc}.\widetilde{\mathsf{HCTR}}^*, \mathsf{Dec}.\widetilde{\mathsf{HCTR}}^*)$ for a (m,n)-bit tweakable random permutation $\widetilde{\Pi}$ and a pair of random hashing keys (K_h, L) or (2) the ideal oracle $(\$, \$^{-1})$, making q queries to its encryption and decryption oracle altogether with at most σ many blocks such that the maximum number of message blocks among all the queried messages is ℓ and the maximum tweak multiplicity is μ. When all the interactions between the oracle and D gets over, it outputs a single bit. We let,

$$\mathbf{Adv}(\mathsf{D}) = \Pr[\mathsf{D}^{\mathsf{Enc}.\widetilde{\mathsf{HCTR}}^*, \mathsf{Dec}.\widetilde{\mathsf{HCTR}}^*} = 1] - \Pr[\mathsf{D}^{\$,\$^{-1}} = 1].$$

We assume that D is computationally unbounded and hence without loss of generality deterministic. Let

$$\tau := \{(T_1, M_1, C_1), (T_2, M_2, C_2), \ldots, (T_q, M_q, C_q)\}$$

be the list of all queries of D and its corresponding responses such that for all $i = 1, 2, \ldots, q$, $|C_i| = |M_i|$. Note that, as D is assumed to be non-trivial, there cannot be any repetition of triplet in τ. τ is called the *query transcript* of the attack. For convenience, we slightly modify the experiment where we reveal to the distinguisher (after it made all its queries and obtains the corresponding responses but before it output its decision) the hashing keys (K_h, L), if we are in the real world, or a pair of uniformly random dummy keys (K_h, L) if we are in the ideal world. All in all, the transcript of the attack is $\tau' = (\tau, K_h, L)$.

A transcript τ' is said to be an *attainable* (with respect to D) transcript if the probability to realize this transcript in the ideal world is non-zero. We denote $\mathcal{V}$ to be the set of all attainable transcripts and X_{re} and X_{id} denotes the probability distribution of transcript τ' induced by the real world and the ideal world respectively. We state in the following the main lemma of the H-Coefficient technique (see [9] for the proof of the lemma).

Lemma 1. *Let D be a fixed deterministic distinguisher and $\mathcal{V} = \mathsf{GoodT} \sqcup \mathsf{BadT}$ (disjoint union) be some partition of the set of all attainable transcripts. Suppose there exists $\epsilon_{\mathrm{ratio}} \geq 0$ such that for any $\tau' \in \mathsf{GoodT}$,*

$$\frac{\Pr[X_{\mathrm{re}} = \tau']}{\Pr[X_{\mathrm{id}} = \tau']} \geq 1 - \epsilon_{\mathrm{ratio}},$$

and there exists $\epsilon_{\mathrm{bad}} \geq 0$ such that $\Pr[X_{\mathrm{id}} \in \mathsf{BadT}] \leq \epsilon_{\mathrm{bad}}$. Then, $\mathbf{Adv}(\mathsf{D}) \leq \epsilon_{\mathrm{ratio}} + \epsilon_{\mathrm{bad}}$.

The remaining of the proof of Theorem 1 is structured as follows: in Section. 4.1 we define bad transcripts and upper bound their probability in the ideal world; in Section 4.2, we analyze good transcripts and prove that they are almost as likely in the real and the ideal world. Theorem 1 then follows easily by combining Lemma 1, Eqn. (6) and (7) above, and Lemmas 2 and 3 proven below.

4.1 Definition and Probability of Bad Transcripts

We begin with defining the bad transcripts and bound their probability in the ideal world. We denote $\widehat{M}_i$ as $M_2^i \| \dots \| M_{l_i}^i$ and $\widehat{C}_i$ as $C_2^i \| \dots \| C_{l_i}^i$. We recall that for a transcript $\tau' = (\tau, K_h, L)$, we denote $X_i = H_{K_h}(\widehat{M}_i) \oplus M_1^i \oplus H_{1,i}, Y_i = H_{K_h}(\widehat{C}_i) \oplus C_1^i \oplus H_{1,i}$ and $IV_a^i = X_i \oplus Y_i \oplus \langle a \rangle$, where $\mathsf{H}'_L(T_i) = (H_{1,i}, H_{2,i})$.

Definition 2. *An attainable transcript $\tau' = (\tau, K_h, L)$ is said to be a* bad *transcript if one of the following conditions are met*

(B.1) *if there exists two queries $(T_i, M_i, C_i), (T_j, M_j, C_j)$ such that (a) $H_{2,i} = H_{2,j}$ and $X_i = X_j$ or (b) $H_{2,i} = H_{2,j}$ and $Y_i = Y_j$*

(B.2) *if there exists two queries (T_i, M_i, C_i) and (T_j, M_j, C_j) such that $H_{2,i} = H_{2,j}$ and $IV_a^i = IV_b^j$ for $a \in [l_i]$ and $b \in [l_j]$.*

(B.3) *if there exists distinct two queries (T_i, M_i, C_i) and (T_j, M_j, C_j) such that $H_{2,i} = H_{2,j}$ and $M_a^i \oplus C_a^i = M_b^j \oplus C_b^j$ for $a \in [l_i]$ and $b \in [l_j]$.*

(B.4) *if there exists two queries (T_i, M_i, C_i) and (T_j, M_j, C_j) such that $H_{2,i} = H_{2,j}$ and $X_i = IV_a^j$ for $a \in [l_j]$.*

(B.5) *if there exists two queries (T_i, M_i, C_i) and (T_j, M_j, C_j) such that $H_{2,i} = H_{2,j}$ and $Y_i = M_a^j \oplus C_a^j$ for $a \in [l_j]$.*

Note that in the ideal world, X_i and Y_i's are determined through the sampled random dummy hash key (K_h, L).

The underlying principle for identifying the bad events is that

if hash of two tweak value happens to collide in two different invocations of the cipher, then the block cipher input and output must not collide.

Let BadT denotes the set of all attainable transcripts τ' such that it satisfies either of the above conditions and the event B denotes $\mathsf{B} := \mathsf{B.1} \vee \mathsf{B.2} \vee \mathsf{B.3} \vee \mathsf{B.4} \vee \mathsf{B.5}$. We bound the probability of the event B in the ideal world as follows:

Lemma 2. *Let X_{id} and BadT be defined as above. Then we have,*

$$\begin{aligned}\Pr[X_{\mathrm{id}} \in \mathsf{BadT}] \le \epsilon_{\mathrm{bad}} &= 2(\mu-1)(q\epsilon + \sigma/2^n) + q^2\delta + 2q\sigma\delta_{\mathsf{au}}/2^n \\ &\quad + 2\max\{q\ell(\mu-1)/2^n + q\sigma\delta_{\mathsf{au}}/2^n, \sigma\epsilon_1\}.\end{aligned}$$

Proof. We let Θ_i denote the set of attainable transcripts satisfying only (B.i) condition. Recall that, in the ideal world, the pair of hash keys (K_h, L) is drawn uniformly and independently from the query transcript. Moreover, K_h is drawn independent of L. We are going to consider every conditions in turn.

Condition B.1. We first fix two distinct queries (T_i, M_i, C_i) and (T_j, M_j, C_j). Now, we compute the following probability over the random draw of the hash keys L and K_h.

$$\Pr[H_{2,i} = H_{2,j}, X_i = X_j]. \tag{8}$$

We can write Eqn. (8) as the joint probability of the following two events:

$$H_{2,i} = H_{2,j}, \quad \mathsf{H}_{K_h}(\widehat{M}_i) \oplus M_1^i \oplus H_{1,i} = \mathsf{H}_{K_h}(\widehat{M}_j) \oplus M_1^j \oplus H_{1,j}.$$

- **Case (a):** if $T_i = T_j$, then $(H_{1,i}, H_{2,i}) = (H_{1,j}, H_{2,j})$. Therefore, the above probability is bounded by ϵ, the AXU probability of H, as we assume the adversary is non-trivial. The number of choices of i is q and j is $\mu - 1$ and thus the overall probability becomes $q(\mu - 1)\epsilon$.
- **Case (b):** if $T_i \neq T_j$, then by conditioning the hash key K_h, the above probability is bounded by δ, the partial almost xor universal probability of the hash function H'. In this case, number of choices of (i, j) is $\binom{q}{2}$ and thus the overall probability becomes $\binom{q}{2}\delta$.

As a result, we have the following

$$\Pr[H_{2,i} = H_{2,j}, X_i = X_j] \leq q(\mu - 1)\epsilon + \binom{q}{2}\delta. \tag{9}$$

By doing the exact similar analysis, the probability over the random draw of the pair of hash keys (K_h, L),

$$\Pr[H_{2,i} = H_{2,j}, Y_i = Y_j] \leq q(\mu - 1)\epsilon + \binom{q}{2}\delta. \tag{10}$$

By summing Eqn. (9) and Eqn. (10), the overall probability becomes

$$\Pr[X_{\mathrm{id}} \in \Theta_1] \leq 2(\mu - 1)q\epsilon + q^2\delta. \tag{11}$$

Condition B.2. We fix two distinct queries (T_i, M_i, C_i) and (T_j, M_j, C_j) and consider the joint probability of $H_{2,i} = H_{2,j}$ and $IV_a^i = IV_b^j$. Note that,

$$IV_a^i = \mathsf{H}_{K_h}(\widehat{M}_i) \oplus \mathsf{H}_{K_h}(\widehat{C}_i) \oplus M_1^i \oplus C_1^i \oplus \langle a \rangle. \tag{12}$$
$$IV_j^b = \mathsf{H}_{K_h}(\widehat{M}_j) \oplus \mathsf{H}_{K_h}(\widehat{C}_j) \oplus M_1^j \oplus C_1^j \oplus \langle b \rangle. \tag{13}$$

Without loss of generality we assume that $i < j$. Now, for a fixed choice of $a \in [l_i]$ and $b \in [l_j]$ and by fixing the hash key K_h, the probability over the random draw of C_1^j (if j-th query is an encryption query) or the random draw of M_i^j (if j-th query is a decryption query) that (12) = (13) is at most 2^{-n}. We have the following two cases:

- **Case (a):** if $T_i = T_j$, then the probability that $H_{2,i} = H_{2,j}$ is one. In this case, number of choices of (i, a) is at most σ and the number of choices of j is at most $\mu - 1$. Note that, the choices of b is only 1 as for fixed values of IV_a^i, IV_b^j and a that satisfies $IV_a^i \oplus IV_b^j = \langle a \rangle \oplus \langle b \rangle$, value of b is uniquly determined. Summing over every possible choices of (i, a, j, b), we get

$$\Pr[X_{\mathrm{id}} \in \Theta_2] \leq \sigma(\mu - 1)/2^n. \tag{14}$$

- **Case (b):** if $T_i \neq T_j$, then the probability that $H_{2,i} = H_{2,j}$ is at most δ_{au}, which follows from the almost universal property of $\mathsf{H}'[2]$. As before, the

number of choices of (i, a) is at most σ and the number of choices of j is at most q. Moreover, as argued before, there is a unique choice of b for a fixed values of IV_a^i, IV_b^j and a that satisfies $IV_a^i \oplus IV_b^j = \langle a \rangle \oplus \langle b \rangle$. Summing over every possible choices of (i, a, j, b), we get

$$\Pr[X_{\text{id}} \in \Theta_2] \leq q\sigma\delta_{\mathsf{au}}/2^n. \tag{15}$$

By summing Eqn. (14) and Eqn. (15), we have the following:

$$\Pr[X_{\text{id}} \in \Theta_2] \leq \sigma(\mu - 1)/2^n + q\sigma\delta_{\mathsf{au}}/2^n. \tag{16}$$

Note that, when $i = j$, then we cannot have $IV_a^i = IV_b^j$ for $a \neq b$ and hence in that case the probability will become 0.

Condition B.3. Analysis of this condition is exactly similar to that of condition B.2 and therefore, we have

$$\Pr[X_{\text{id}} \in \Theta_3] \leq \sigma(\mu - 1)/2^n + q\sigma\delta_{\mathsf{au}}/2^n. \tag{17}$$

Condition B.4. We first fix two distinct queries (T_i, M_i, C_i), (T_j, M_j, C_j) and compute the following:

$$\Pr[H_{2,i} = H_{2,j}, X_i = IV_a^j].$$

For a fixed index $a \in [l_j]$, we compute the probability of $X_i = IV_a^j$. Recall that, $X_i = \mathsf{H}_{K_h}(\widehat{M_i}) \oplus M_1^i \oplus H_{1,i}$. Therefore, the probability of $X_i = IV_a^j$ is nothing but to calculate the probability of the event that

$$\mathsf{H}_{K_h}(\widehat{M_i}) \oplus \mathsf{H}_{K_h}(\widehat{M_j}) \oplus \mathsf{H}_{K_h}(\widehat{C_j}) = M_1^i \oplus M_1^j \oplus C_1^j \oplus H_{1,i} \oplus \langle a \rangle. \tag{18}$$

Without loss of generality we assume that $i < j$. If the j-th query is an encryption query, then C_1^j is random and hence over the random draw of C_1^j, the probability of Eqn. (18) is 2^{-n}. Similarly, if the j-th query is a decryption query, then M_1^j is random and hence over the random draw of M_1^j, the probability of Eqn. (18) is 2^{-n}. We have the following two cases:

- **Case (a):** if $T_i = T_j$, then the probability that $H_{2,i} = H_{2,j}$ is one. In this case, the number of choices of i is q and (j, a) is at most $(\mu - 1)\ell$. Therefore, by summing over every possible choices of (i, j, a), we get

$$\Pr[X_{\text{id}} \in \Theta_4] \leq q\ell(\mu - 1)/2^n. \tag{19}$$

- **Case (b):** if $T_i \neq T_j$, then the probability that $H_{2,i} = H_{2,j}$ is at most δ_{au}, which follows from the almost universal property of $\mathsf{H}'[2]$. Here, the number of choices of (j, a) is at most σ and the number of choices of i is at most q. Summing over every possible choices of (i, j, a), we get

$$\Pr[X_{\text{id}} \in \Theta_4] \leq q\sigma\delta_{\mathsf{au}}/2^n. \tag{20}$$

By summing Eqn. (19) and Eqn. (20), we obtain

$$\Pr[X_{\mathrm{id}} \in \Theta_4] \leq q\ell(\mu-1)/2^n + q\sigma\delta_{\mathsf{au}}/2^n. \tag{21}$$

When $i = j$, then calculating the joint probability of $H_{2,i} = H_{2,j}, X_i = IV_a^j$ is nothing but to calculate the probability of the event that

$$\mathsf{H}_{K_h}(\widehat{C}_i) = C_1^i \oplus H_{1,i} \oplus \langle a \rangle. \tag{22}$$

Note that, when $i = j$, then the probability of $H_{2,i} = H_{2,j}$ is one. Now, for a fixed $i \in [q]$ and $a \in [l_i]$, over the random draw the hash key K_h, the probability of the above event is bounded by ϵ_1 due to the almost regular property of the hash function. Now, summing over all possible choices of (i, a) we get

$$\Pr[X_{\mathrm{id}} \in \Theta_4] \leq \sigma\epsilon_1. \tag{23}$$

Therefore, from Eqn. (21) and Eqn. (23) we have

$$\Pr[X_{\mathrm{id}} \in \Theta_4] \leq \max\{q\ell(\mu-1)/2^n + q\sigma\delta_{\mathsf{au}}/2^n, \sigma\epsilon_1\}. \tag{24}$$

Condition B.5. Analysis of this condition is exactly similar to that of condition B.4. Therefore, we have

$$\Pr[X_{\mathrm{id}} \in \Theta_5] \leq \max\{q\ell(\mu-1)/2^n + q\sigma\delta_{\mathsf{au}}/2^n, \sigma\epsilon_1\}. \tag{25}$$

The result follows by the union bound of these conditions in Eqn. (11), Eqn. (16), Eqn. (17), Eqn. (24) and Eqn. (25).

4.2 Analysis of Good Transcripts.

In this section, we show that for a good transcript τ', realizing τ' is almost as likely in the real and the ideal world. Formally, we prove the following lemma.

Lemma 3. *Let $\tau' = (\tau, K_h, L)$ be a good transcript. Then*

$$\frac{\mathsf{p}_{\mathrm{re}}(\tau')}{\mathsf{p}_{\mathrm{id}}(\tau')} := \frac{\Pr[X_{\mathrm{re}} = \tau']}{\Pr[X_{\mathrm{id}} = \tau']} \geq 1.$$

Proof. Let $\tau' = (\tau, K_h, L) \in \mathsf{GoodT}$ and let $\tau = ((T_1, M_1, C_1), \ldots, (T_q, M_q, C_q))$. Now, we define an equivalence relation $\sim_\tau$ over τ such that two elements of τ are related through $\sim_\tau$, i.e., $(T_i, M_i, C_i) \sim_\tau (T_j, M_j, C_j)$, if and only if $\mathsf{H}'_L(T_i)[2] = \mathsf{H}'_L(T_j)[2]$. This equivalence relation induces a partition over τ and let $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_r$ be r many partitions of τ where $|\mathcal{P}_i| = \mathsf{q}_i$, called the multiplicity of the hash value of the tweak T_i. Therefore, we have $\mathsf{q}_1 + \mathsf{q}_2 + \ldots + \mathsf{q}_r = q$. Now, we consider any i-th partition $\mathcal{P}_i$ for $i = 1, \ldots, r$. Note that, $\mathcal{P}_i$ is of the form:

$$\mathcal{P}_i = ((T_{x_1}, M_{x_1}, C_{x_1}), \ldots, (T_{x_{\mathsf{q}_i}}, M_{x_{\mathsf{q}_i}}, C_{x_{\mathsf{q}_i}})),$$

where $\mathsf{H}'_L(T_{x_1})[2] = \mathsf{H}'_L(T_{x_2})[2] = \ldots = \mathsf{H}'_L(T_{x_{\mathsf{q}_i}})[2]$. We say two elements (T_x, M_x, C_x) and (T_y, M_y, C_y) of $\mathcal{P}_i$ are related through an equivalence relation

$\sim_\ell$ if and only if $|M_x| = |M_y|$ and hence $|C_x| = |C_y|$. Therefore, $\sim_\ell$ indcuces another v_i many *inner* partitions $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{\mathsf{v}_i}$ of $\mathcal{P}_i$ such that

$$c_1 + c_2 + \ldots + c_{\mathsf{v}_i} = \mathsf{q}_i,$$

where $c_j = |\mathcal{C}_j|$ denotes the number of elements in the j-th partition $\mathcal{C}_j$. Moreover, for the simplicity of the analysis, we assume that the length of each queried message is a multiple of n.

Ideal Interpolation Probability. To compute the ideal interpolation probability for the fixed trascript $\tau' = (\tau, K_h, L)$, we first consider any partition $\mathcal{P}_i$ in which q_i many hash values of the tweaks attain the same value. Now, let us consider the j-th inner partition $\mathcal{C}_j$ of $\mathcal{P}_i$ for which we have c_j many (M, C) pairs having the same length nl_j. Therefore, for $\mathcal{C}_j$, the probability becomes $1/(2^{nl_j})_{c_j}$. Similarly, for other inner partition $\mathcal{C}_{j'}$ of $\mathcal{P}_i$ in which $c_{j'}$ many (M, C) pairs having the same length $nl_{j'}$, the probability becomes $1/(2^{nl_{j'}})_{c_{j'}}$. Thus, for a fixed partition $\mathcal{P}_i$, the probability becomes

$$\prod_{j=1}^{\mathsf{v}_i} \frac{1}{(2^{nl_j})_{c_j}}.$$

Since, we have r many such partitions, the overall probability becomes

$$\prod_{i=1}^{r} \prod_{j=1}^{\mathsf{v}_i} \frac{1}{(2^{nl_j})_{c_j}}.$$

By summarizing the above, we have

$$\Pr[X_{\mathrm{id}} = \tau'] = \frac{1}{|\mathcal{K}_h|} \frac{1}{|\mathcal{L}|} \cdot \prod_{i=1}^{r} \prod_{j=1}^{\mathsf{v}_i} \frac{1}{(2^{nl_j})_{c_j}}, \tag{26}$$

where nl_j is the length of every message in partition $\mathcal{C}_j$.

Real Interpolation Probability. To compute the real interpolation probability for the fixed good transcript $\tau' = (\tau, K_h, L)$, we first consider several lists created from τ:

$$\mathcal{L}_A = ((H_{2,1}, X_1, Y_1), (H_{2,2}, X_2, Y_2), \ldots, (H_{2,q}, X_q, Y_q)),$$

where $X_i = M_1^i \oplus H_{K_h}(\hat{M}_i) \oplus H_{1,i}$ and $Y_i = C_1^i \oplus H_{K_h}(\hat{C}_i) \oplus H_{1,i}$. Moreover, we also create q many different lists from τ as follows:

$$\begin{aligned}
\mathcal{L}_1 &= ((H_{2,1}, IV_1^1, Z_1^1), (H_{2,1}, IV_2^1, Z_2^1), \ldots, (H_{2,1}, IV_{l_1-1}^1, Z_{l_1-1}^1)) \\
\mathcal{L}_2 &= ((H_{2,2}, IV_1^2, Z_1^2), (H_{2,2}, IV_2^2, Z_2^2), \ldots, (H_{2,2}, IV_{l_2-1}^2, Z_{l_2-1}^2)) \\
&\vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad\qquad \vdots \\
\mathcal{L}_q &= ((H_{2,q}, IV_1^q, Z_1^q), (H_{2,q}, IV_2^q, Z_2^q), \ldots, (H_{2,q}, IV_{l_q-1}^q, Z_{l_q-1}^q)),
\end{aligned}$$

where $IV_k^i = X_i \oplus Y_i \oplus \langle k \rangle$ and $Z_k^i = M_{k+1}^i \oplus C_{k+1}^i$. Now, we consider any partition $\mathcal{P}_i$, in which q_i many hash values of the tweaks (i.e., $H_{2,i}$) attain the same value. This implies that q_i many elements from the list $\mathcal{L}_A$ i.e.

$$((H_{2,k_1}, X_{k_1}, Y_{k_1}), \ldots, (H_{2,k_{\mathsf{q}_i}}, X_{k_{\mathsf{q}_i}}, Y_{k_{\mathsf{q}_i}}))$$

will have the same tweak value, but all the $X_{k_1}, X_{k_2}, \ldots, X_{K_{\mathsf{q}_i}}$ values are distinct. Similarly, all the $Y_{k_1}, Y_{k_2}, \ldots, Y_{K_{\mathsf{q}_i}}$ values are distinct, otherwise condition B.1 would have been satisfied. Moreover, q_i many lists from $\mathcal{L}_1, \ldots, \mathcal{L}_q$ will also have the same tweak value i.e., $H_{2,k_1} = H_{2,k_2} = \ldots = H_{2,k_{\mathsf{q}_i}}$ in

$$\begin{aligned}
\mathcal{L}_{k_1} &= ((H_{2,k_1}, IV_1^{k_1}, Z_1^{k_1}), (H_{2,k_1}, IV_2^{k_1}, Z_2^{k_1}), \ldots, (H_{2,k_1}, IV_{l_{k_1}-1}^{k_1}, Z_{l_{k_1}-1}^{k_1})) \\
\mathcal{L}_{k_2} &= ((H_{2,k_2}, IV_1^{k_2}, Z_1^{k_2}), (H_{2,k_2}, IV_2^{k_2}, Z_2^{k_2}), \ldots, (H_{2,k_2}, IV_{l_{k_2}-1}^{k_2}, Z_{l_{k_2}-1}^{k_2})) \\
&\vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad\qquad \vdots \\
\mathcal{L}_{k_{\mathsf{q}_i}} &= ((H_{2,k_{\mathsf{q}_i}}, IV_1^{k_{\mathsf{q}_i}}, Z_1^{k_{\mathsf{q}_i}}), (H_{2,k_{\mathsf{q}_i}}, IV_2^{k_{\mathsf{q}_i}}, Z_2^{k_{\mathsf{q}_i}}), \ldots, (H_{2,k_{\mathsf{q}_i}}, IV_{l_{k_{\mathsf{q}_i}}-1}^{k_{\mathsf{q}_i}}, Z_{l_{k_{\mathsf{q}_i}}-1}^{k_{\mathsf{q}_i}}))
\end{aligned}$$

As τ' is a good transcript, it is evident that $IV_\beta^\alpha \neq IV_{\beta'}^{\alpha'}$ where $\alpha, \alpha' \in \{k_1, \ldots, k_{\mathsf{q}_i}\}$ and $\beta \in [l_\alpha - 1], \beta' \in [l_{\alpha'} - 1]$ othwerise condition B.2 would have been satisfied. Similarly, as τ' is a good transcript, we have $Z_\beta^\alpha \neq Z_{\beta'}^{\alpha'}$ otherwise condition B.3 would have been satisfied. Moreover, due to condition B.4 and B.5, we also have $IV_\beta^\alpha \neq X_{\alpha'}$ and $Z_\beta^\alpha \neq Y_{\alpha'}$. This immediately gives us the probabilty for any such fixed partition $\mathcal{P}_i$ is

$$\frac{1}{(2^n)_{\mathsf{q}_i + (l_{k_1}-1)+(l_{k_2}-1)+\ldots+(l_{k_{\mathsf{q}_i}}-1)}} = \frac{1}{(2^n)_{l_{k_1}+l_{k_2}+\ldots+l_{k_{\mathsf{q}_i}}}}.$$

Now, let us consider the j-th inner partition $\mathcal{C}_j$ of $\mathcal{P}_i$ for which we have c_j many (M, C) pairs having the same message length nl_j. Therefore, for the fixed partition $\mathcal{P}_i$, the eventual probablity will be $1/(2^n)_{q_i+\theta}$, where $\theta = c_1(l_1 - 1) + c_2(l_2 - 1) + \ldots + c_{\mathsf{v}_i}(l_{\mathsf{v}_i} - 1)$. Summarizing above, we have

$$\Pr[X_{\mathrm{re}} = \tau'] = \frac{1}{|\mathcal{K}_h|} \cdot \frac{1}{|\mathcal{L}|} \cdot \prod_{i=1}^{r} \frac{1}{(2^n)_{\mathsf{q}_i+\theta}} = \frac{1}{|\mathcal{K}_h|} \cdot \frac{1}{|\mathcal{L}|} \cdot \prod_{i=1}^{r} \frac{1}{(2^n)_{c_1 l_1 + c_2 l_2 + \ldots + c_{\mathsf{v}_i} l_i}} \tag{27}$$

Compute the ratio. Finally, by taking the ratio of Eqn. (27) to Eqn. (26), we have

$$\frac{\Pr[X_{\mathrm{re}} = \tau']}{\Pr[X_{\mathrm{id}} = \tau']} = \prod_{i=1}^{r} \frac{\prod_{j=1}^{\mathsf{v}_i} (2^{nl_j})_{c_j}}{(2^n)_{c_1 l_1 + c_2 l_2 + \ldots + c_{\mathsf{v}_i} l_i}} = \prod_{i=1}^{r} \underbrace{\frac{(2^{nl_1})_{c_1} \cdot (2^{nl_2})_{c_2} \cdots (2^{nl_{\mathsf{v}_i}})_{c_{\mathsf{v}_i}}}{(2^n)_{c_1 l_1 + c_2 l_2 + \ldots + c_{\mathsf{v}_i} l_i}}}_{(\mathsf{R})}$$

The following proposition shows that for any $i = 1, \ldots, r$, $\mathsf{R} \geq 1$ and hence the result follows. □

Proposition 1. *For positive integers $c_1, \ldots, c_t$ and $l_1, \ldots, l_t$ such that $\sum_{i=1}^{t} c_i l_i \leq 2^n$, we have,*

$$(2^n)_{c_1 l_1 + c_2 l_2 + \ldots + c_t l_t} \leq \prod_{j=1}^{t} (2^{nl_j})_{c_j}.$$

Proof of the result is trivial and hence omitted.

Corollary of Theorem 1. When the input tweak size of the construction matches with the tweak size of the tweakable block cipher, then we can evade the hash function evaluation for processing tweaks. As a result, we directly feed the tweak of the construction to the tweakable block cipher and the security bound of the resulting construction is obtained as a simple corollary of Theorem 1. For an m-bit tweak T, we define the hash function $\mathsf{H}'_L(T)$ as $\mathsf{H}'_L(T) = (0^n, T)$. Note that, for this partial almost xor universal hash function, $\delta = 0$ and $\delta_{\mathsf{au}} = 0$. Therefore, following Theorem 1, the information theoretic security bound of tweakable HCTR^* for m-bit tweak becomes

$$\mathbf{Adv}^{\mathrm{TSPRP}}_{\widetilde{\mathsf{HCTR}}^*[\tilde{\Pi},\mathsf{H},\mathsf{H}']}(\mathsf{A}) \leq 2(\mu-1)(q\epsilon + \sigma/2^n) + 2\max\{q\ell(\mu-1)/2^n, \sigma\epsilon_1\}.$$

When all the tweaks in the encryption and decryption queries are distinct (i.e., $\mu = 1$), then by assuming $\epsilon, \epsilon_1 \approx 2^{-n}$, $\widetilde{\mathsf{HCTR}}^*$ is secured roughly upto 2^n many message blocks.

5 Conclusion

HCTR is one of the most efficient TES candidates which turns an n-bit block cipher into a variable length TBC. In this paper, we have proposed tweakable HCTR, that turns an (m, n)-bit TBC into a variable length TBC, allowing to process arbitrary large tweaks, and proven its optimal security (in terms of the block size) for the case of distinct tweak. Moreover, we have shown that the constuction gives a graceful security degradation with the maximum number of repetitions of tweak. It is evident that one can make the HCTR mode BBB secure by just doubling the size of all its primitives. Nevertheless, desigining a double block sprp is not trivial. For example, 5 round Feistel construction [18] provides 2^n security against all adaptive chosen plaintext and chosen ciphertext adversaries. Thus, designing an efficient TES based on an n-bit block cipher with beyond the birthday bound security still remains an interesting open problem. However, following [17], analysis of muti-key security of HCTR will be similar to the analysis of ours.

Acknowledgements

Authors are supported by the WISEKEY project of R.C.Bose Centre for Cryptology and Security. The authors would like to thank all the anonymous reviewers of Indocrypt 2018 for their invaluable comments and suggestions that help to improve the overall quality of the paper.

References

1. Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of cipher block chaining. In *CRYPTO '94*, pages 341–358, 1994.
2. Mihir Bellare and Phillip Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, pages 317–330, 2000.
3. Ritam Bhaumik and Mridul Nandi. An inverse-free single-keyed tweakable enciphering scheme. In *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, pages 159–180, 2015.
4. Larry Carter and Mark N. Wegman. Universal classes of hash functions. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.
5. Debrup Chakraborty, Sebati Ghosh, and Palash Sarkar. A fast single-key two-level universal hash function. *IACR Trans. Symmetric Cryptol.*, 2017(1):106–128, 2017.
6. Debrup Chakraborty and Mridul Nandi. An improved security bound for HCTR. In *Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers*, pages 289–302, 2008.
7. Debrup Chakraborty and Palash Sarkar. HCH: A new tweakable enciphering scheme using the hash-encrypt-hash approach. In *Progress in Cryptology - INDOCRYPT 2006, 7th International Conference on Cryptology in India, Kolkata, India, December 11-13, 2006, Proceedings*, pages 287–302, 2006.
8. Debrup Chakraborty and Palash Sarkar. A new mode of encryption providing a tweakable strong pseudo-random permutation. In *Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers*, pages 293–309, 2006.
9. Shan Chen, Rodolphe Lampe, Jooyoung Lee, Yannick Seurin, and John P. Steinberger. Minimizing the two-round even-mansour cipher. In *CRYPTO 2014. Proceedings, Part I*, pages 39–56, 2014.
10. Joan Daemen and Vincent Rijmen. Rijndael for AES. In *AES Candidate Conference*, pages 343–348, 2000.
11. Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Kan Yasuda. Encrypt or decrypt? to make a single-key beyond birthday secure nonce-based MAC. *IACR Cryptology ePrint Archive*, 2018:500, 2018.
12. Shai Halevi. Eme*: Extending EME to handle arbitrary-length messages with associated data. In *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*, pages 315–327, 2004.
13. Shai Halevi. Invertible universal hashing and the TET encryption mode. In *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, pages 412–429, 2007.
14. Shai Halevi and Phillip Rogaway. A tweakable enciphering mode. In *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, pages 482–499, 2003.

15. Shai Halevi and Phillip Rogaway. A parallelizable enciphering mode. In *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, pages 292–304, 2004.
16. Ashwin Jha, Eik List, Kazuhiko Minematsu, Sweta Mishra, and Mridul Nandi. XHX - A framework for optimally secure tweakable block ciphers from classical block ciphers and universal hashing. *IACR Cryptology ePrint Archive*, 2017:1075, 2017.
17. Jooyoung Lee, Atul Luykx, Bart Mennink, and Kazuhiko Minematsu. Connecting tweakable and multi-key blockcipher security. *Des. Codes Cryptography*, 86(3):623–640, 2018.
18. Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, 1988.
19. Cuauhtemoc Mancillas-López, Debrup Chakraborty, and Francisco Rodríguez-Henríquez. Efficient implementations of some tweakable enciphering schemes in reconfigurable hardware. In *Progress in Cryptology - INDOCRYPT 2007, 8th International Conference on Cryptology in India, Chennai, India, December 9-13, 2007, Proceedings*, pages 414–424, 2007.
20. David A. McGrew and Scott R. Fluhrer. The extended codebook (XCB) mode of operation. *IACR Cryptology ePrint Archive*, 2004:278, 2004.
21. David A. McGrew and John Viega. The security and performance of the galois/counter mode (GCM) of operation. In *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*, pages 343–355, 2004.
22. Bart Mennink. Towards tight security of cascaded LRW2. *IACR Cryptology ePrint Archive*, 2018:434, 2018.
23. Kazuhiko Minematsu. Beyond-birthday-bound security based on tweakable block cipher. In *Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 22-25, 2009, Revised Selected Papers*, pages 308–326, 2009.
24. Kazuhiko Minematsu and Tetsu Iwata. Building blockcipher from tweakable blockcipher: Extending FSE 2009 proposal. In *Cryptography and Coding - 13th IMA International Conference, IMACC 2011, Oxford, UK, December 12-15, 2011. Proceedings*, pages 391–412, 2011.
25. Kazuhiko Minematsu and Tetsu Iwata. Tweak-length extension for tweakable blockciphers. In *Cryptography and Coding - 15th IMA International Conference, IMACC 2015, Oxford, UK, December 15-17, 2015. Proceedings*, pages 77–93, 2015.
26. Kazuhiko Minematsu and Toshiyasu Matsushima. Tweakable enciphering schemes from hash-sum-expansion. In *Progress in Cryptology - INDOCRYPT 2007, 8th International Conference on Cryptology in India, Chennai, India, December 9-13, 2007, Proceedings*, pages 252–267, 2007.
27. Moni Naor and Omer Reingold. A pseudo-random encryption mode. *Manuscript available from www.wisdom.weizmann.ac.il/naor*.
28. Moni Naor and Omer Reingold. On the construction of pseudorandom permutations: Luby-rackoff revisited. *J. Cryptology*, 12(1):29–66, 1999.
29. National Bureau of Standards. Data encryption standard. *Federal Information Processing Standard*, 1977.
30. Jacques Patarin. A proof of security in o(2n) for the xor of two random permutations. In *ICITS 2008, Proceedings*, pages 232–248, 2008.
31. Jacques Patarin. The "Coefficients H" Technique. In *Selected Areas in Cryptography, SAC*, pages 328–345, 2008.

32. Thomas Peyrin and Yannick Seurin. Counter-in-tweak: Authenticated encryption modes for tweakable block ciphers. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, pages 33–63, 2016.
33. Phillip Rogaway, Mihir Bellare, John Black, and Ted Krovetz. OCB: a block-cipher mode of operation for efficient authenticated encryption. In *CCS 2001, Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, Pennsylvania, USA, November 6-8, 2001.*, pages 196–205, 2001.
34. Palash Sarkar. Improving upon the TET mode of operation. In *Information Security and Cryptology - ICISC 2007, 10th International Conference, Seoul, Korea, November 29-30, 2007, Proceedings*, pages 180–192, 2007.
35. Peng Wang, Dengguo Feng, and Wenling Wu. HCTR: A variable-input-length enciphering mode. In *Information Security and Cryptology, First SKLOIS Conference, CISC 2005, Beijing, China, December 15-17, 2005, Proceedings*, pages 175–188, 2005.
36. John L. Smith William F. Ehrsam, Carl H. W. Meyer and Walter L. Tuchman. Message verification and transmission error detection by block chaining. *US Patent 4074066*, 1976.