Designing Tweakable Enciphering Schemes Using Public Permutations

Debrup Chakraborty[1], Avijit Dutta[2], and Samir Kundu[1]

[1] Indian Statistical Institute
203 B.T. Road, Kolkata-700108
E-mail: debrup@isical.ac.in, samirkundu3@gmail.com
[2] Institute for Advancing Intelligence, TCG-CREST
Salt Lake, Kolkata-700091
E-mail: avirocks.dutta13@gmail.com

Abstract. A tweakable enciphering scheme (TES) is a length preserving (tweakable) encryption scheme that provides (tweakable) strong pseudorandom permutation security on arbitrarily long messages. TES is traditionally built using block ciphers and the security of the mode depends on the strong pseudorandom permutation security of the underlying block cipher. In this paper, we construct TESs using public random permutations. Public random permutations are being considered as a replacement of block cipher in several cryptographic schemes including AEs, MACs, etc. However, to our knowledge, a systematic study of constructing TES using public random permutations is missing. In this paper, we give a generic construction of a TES which uses a public random permutation, a length expanding public permutation based PRF and a hash function which is both almost xor universal and almost regular. Further, we propose a concrete length expanding public permutation based PRF construction. We also propose a single keyed TES using a public random permutation and an AXU and almost regular hash function.

1 Introduction

Permutation Based Cryptography. A cryptographic permutation is a key-less public permutation that is designed to behave as a random permutation. In recent years cryptographic permutations have started to evolve as a useful primitive in parallel to the block ciphers. The primary feature of a cryptographic permutation is that it does not use any key and hence separate processing of the key and the data input is not required as in a block cipher. This makes cryptographic permutations a more efficient primitive compared to block ciphers in certain scenarios. The use of cryptographic permutation gained popularity during the SHA-3 competition [1], as several submitted candidates in the competition were based on this type of primitive. Furthermore, the selection of the permutation-based Keccak sponge function as the SHA-3 standard has generated ample confidence within the community for using this primitive [49]. In 2007, Bertoni et al. defined the cryptographic permutation based sponge function [7], which was initially aimed for hashing. Soon after, several efficient modes for encryption, authentication and authenticated encryption were developed [45, 5, 6]. Today, permutation based sponge-based constructions have become a successful and a full-fledged alternative to the block cipher-based modes. In fact, in the first round of the ongoing NIST lightweight competition [47], 24 out of 57 submitted constructions are based on cryptographic permutations, and out of 24, 16 permutation based proposals have qualified for round 2. These statistics, beyond any doubt, clearly depict the wide adoption of permutation based schemes [3, 4, 9, 15, 26, 32] in parallel to the block cipher based designs. Apart from the modes, several cryptographic permutations have also been designed which are claimed to be efficient than standard block ciphers [8, 13, 4].

Besides the permutation based designs of encryption/authentication schemes, a long line of research has been carried out in the study of designing block cipher and tweakable block cipher out of public random permutations. Even Mansour (EM) [36] and Iterated Even Mansour (IEM) ciphers are notable approaches in this direction. EM cipher is defined as $\mathrm{EM}(x) \stackrel{\Delta}{=} \pi(x \oplus k_1) \oplus k_2$, where π is a public random permutation and k_1, k_2 are two independent keys. Iterating EM cipher for $r \geq 2$ times with r independent permutations and $r+1$ independent round keys defines the r-round IEM cipher, i.e. $\mathrm{EM}^r(x) \stackrel{\Delta}{=} k_{r+1} \oplus \pi_r(k_r \oplus \pi_{r-1}(\ldots(\pi_2(k_2 \oplus \pi_1(k_1 \oplus x))\ldots))$. A long line of research has studied the security of r-round IEM [14, 25, 31, 27]. Recently, Chen et al. have designed two public permutation based PRFs [24] which have been proven to be secure beyond the birthday bound.

Tweakable Enciphering Schemes. A *Tweakable Enciphering Scheme* or in short TES is a deterministic length preserving encryption scheme which provides security against adaptive chosen plaintext and ciphertext attacks, i.e., no efficient adversary should be able to distinguish ciphertexts from random strings and should not be able to tamper a ciphertext so that it gets decrypted to something meaningful. The security requirement of a TES is very similar to that of a deterministic authenticated encryption (DAE) scheme [2]. However, DAE schemes are not length preserving; the ciphertext resulting from the DAE is always expanded by the expansion factor defined by a specific DAE scheme. It is thus the length preserving property that makes TES a separate cryptographic primitive from DAE. The length preserving feature of TES makes it a suitable candidate for low level disk encryption [20, 16]. One can see a tweakable enciphering scheme as a tweakable block cipher [43] with arbitrary block lengths and are thus sometimes called wide block modes.

Over the years, there have been several proposals of TES constructions and most of them are build on top of block ciphers. Constructions like CMC [38], EME [39], EME* [37], FMix [11], AEZ [40] are build only using block ciphers whereas XCB [44, 17], HCTR [51], HCH [20] uses both block ciphers and universal hash functions. There are few constructions of TES using stream ciphers [18, 50].

Most block cipher based schemes have been proven to be secure assuming the block cipher to be a strong pseudorandom permutation, as these constructions require the decryption functionality of the block cipher for deciphering the ciphertext. However, there are some constructions such as FMix [11], AEZ [40] and FAST [16], which do not require the decryption functionality of the block cipher and hence their security can be proved under the assumption that the underlying block cipher is a pseudorandom function. Such schemes are called *inverse free* TESs. Moreover, the security of all these constructions caps at birthday bound [1]. Dutta and Nandi [34] proposed a tweakable block cipher based TES and proved its security beyond the birthday bound [2] assuming the underlying block cipher to be a tweakable strong pseudorandom permutation.

Our Contributions. Although several modes for authentication, hash function, and authenticated encryption, have been developed using public permutations till date, to our knowledge, the only work which describes a TES built using a public random permutation is [5]. The construction in [5] uses four round Luby Rackoff construction using two pseudorandom functions and the pseudorandom functions are constructed using public permutations. Concrete security bounds and formal security proofs for the TES scheme are not provided in [5] and to the best of our knowledge, there is no provably secure public permutation based TES scheme. We initiate a study of such a construction in this paper. Our concrete contributions are the following.

1. First, we propose a generic construction of a public permutation based TES, called ppTES. Our proposal closely resembles the HCTR construction. ppTES is designed using a public permutation π, a length expanding public permutation based pseudorandom function[3] $\mathsf{F}_k^{\pi'}$, where π and π' are two independent public random permutations over the same space. Additionally, ppTES uses a keyed hash function H_{k_h}, which is required to be both almost xor universal (AXU) and almost regular (AR) (we further call such functions as AXUAR functions). We prove that if $\mathsf{F}_k^{\pi'}$ is a secure length expanding public permutation based PRF and the hash function is a secure AXUAR function, then ppTES is secure against adaptive chosen plaintext and ciphertext adversaries.
2. As our second contribution, we construct a length expanding public permutation based PRF which we call ppCTR. ppCTR essentially is a counter mode of encryption

[1] A cryptographic construction is said to be birthday bound secure if its security retains as long as the number of queries is upto $2^{n/2}$, where n is the block size of the underlying primitive. In literaure, there are plenty of constructions which are birthday bound secure [19, 21, 16, 22].

[2] A cryptographic construction is said to be beyond birthday bound secure if its security retains even if the number of queries exceeds $2^{n/2}$, where n is the block size of the underlying primitive. Examples of beyond birthday bound secure construction includes [28, 46, 29, 30, 35, 33].

[3] Informally, a length expanding PRF takes an input x and the number of blocks b and outputs b many blocks, where block refers to an element of $\{0, 1\}^n$, for some fixed n.

where the block ciphers are replaced by the single round Even Mansour [36] construction. We show that ppCTR offers a tight $n/2$ bit security. We use ppCTR and the PolyHash [52] function in ppTES construction to realize a concrete TES which we call ppHCTR. ppHCTR requires two keys and two independent public permutations.

3. Finally, we propose ppHCTR+, a public permutation based TES which uses a single key and a single public permutation. Along with the permutation, ppHCTR+ also requires an AXUAR hash function and the only key required in ppHCTR+ is the hash key of the AXUAR hash function. We prove that ppHCTR+ is a birthday bound secure public permutation based TES.

We would like to mention that any block-cipher based TES can be converted to a public permutation based scheme by replacing the block ciphers with a single round EM construction. But such direct replacement of block cipher by the EM scheme will require multiple keys, for example a direct replacement of the block cipher with the single round EM construction in HCTR mode results in a three keyed (along with the hash key) construction with two independent permutations. Whereas our proposed construction ppHCTR+ requires only the hash key and a single random permutation. Additionally, ppHCTR+ saves a few XOR counts compared to the direct replacement of the block cipher with single round EM construction. Also, ppHCTR+ provides comparable security to the existing block cipher based TES schemes.

2 Preliminaries

Basic Notations. For a finite set $\mathcal{X}$, $X \leftarrow_\$ \mathcal{X}$ denotes that X is sampled uniformly at random from $\mathcal{X}$. For a sequence of r random variables $(X_1, \ldots, X_r)$, $X_1, \ldots, X_r \leftarrow_\$ \mathcal{X}$ denotes that X_i's are independently and uniformly sampled from $\mathcal{X}$. For $q \in \mathbb{N}$, we write $[q]$ to refer to the set $\{1, \ldots, q\}$. For $n \in \mathbb{N}$, $\{0,1\}^n$ denotes the set of all binary strings of length n and $\{0,1\}^{\geq n}$ denotes the set of all binary strings of length at least n. Therefore, $\{0,1\}^{\geq 0}$ is the set of all binary strings of arbitrary length (including the empty string ε) and denoted by $\{0,1\}^*$. An element of $\{0,1\}^n$ is called a *block*. For $x \in \{0,1\}^*$, $|x|$ denotes the length of x in bits. For $s \in \mathbb{N}$, $\mathsf{first}(s, x)$ denotes the first s bits of a binary string x whose length is at least s. For $x, y \in \{0,1\}^*$, $x\|y$ denotes the concatenation of x followed by y. For $x, y \in \{0,1\}^n$, we write $x \oplus y$ to denote their bitwise xor. For any $x \in \{0,1\}^*$, $\mathsf{parse}_n(x)$ parses x as $x_1\|x_2\|\ldots\|x_\ell$ where each x_i, for $i \in [\ell-1]$, is a block and $0 \leq |x_\ell| \leq n$. For a sequence of elements $x^1, x^2, \ldots, x^s \in \{0,1\}^*$, we write x^i_a to denote the a-th block of the i-th element x^i. $\langle j \rangle$ denotes the n-bit binary representation of a non negative integer $j < 2^n$. For integers $1 \leq b \leq a$, we write $\mathbf{P}(a, b)$ to denote $a(a-1)\ldots(a-b+1)$, where $\mathbf{P}(a, 0) = 1$ by convention.

The set of all functions from $\mathcal{X}$ to $\mathcal{Y}$ is denoted by $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$. When $\mathcal{Y} = \{0,1\}^n$, then we denote $\mathsf{Func}(\mathcal{X}, \{0,1\}^n)$ simply as $\mathsf{Func}_{\mathcal{X}}(n)$ and sometimes we write $\mathsf{Func}(n)$ by omitting $\mathcal{X}$ when the domain of the function is understood from the context. We denote the set of all n bit permutations by $\mathsf{Perm}(n)$.

2.1 Security Definitions

In this paper, we adapt the definitions of PRF and TES in the random permutation model.

PRF BASED ON PUBLIC RANDOM PERMUTATION. Let $\mathsf{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a keyed function from $\mathcal{X}$ to $\mathcal{Y}$ constructed using d many n-bit permutations $\boldsymbol{\pi} \stackrel{\Delta}{=} (\pi_1, \ldots, \pi_d)$, where $\mathcal{K}$ is called the key space, $\mathcal{X}$ is called the input space and $\mathcal{Y}$ is called the output space. We consider the Pseudo Random Function (PRF) security of F under public permutation model where we assume that $\pi_1, \ldots, \pi_d \leftarrow\!\!\$\, \mathsf{Perm}(n)$ and the distinguisher D is given access to either $(\mathsf{F}_K^{\boldsymbol{\pi}}; \pi_1^{\pm}, \ldots, \pi_d^{\pm})$ for a random key $K \leftarrow\!\!\$\, \mathcal{K}$ or $(\mathsf{RF}; \pi_1^{\pm}, \ldots, \pi_d^{\pm})$ for $\mathsf{RF} \leftarrow\!\!\$\, \mathsf{Func}(\mathcal{X}, \mathcal{Y})$. The superscript $\pm$ for the π_i's denotes that the distinguisher can query π_i in both the forward and reverse directions. Query of the distinguisher to π_i is called the *primitive query* and query to $\mathsf{F}_K^{\boldsymbol{\pi}}$ or RF is called the *construction query*. We define the PRF advantage of F in public permutation model with respect to the distinguisher D that makes q construction queries and total q_p primitive queries as

$$\mathbf{Adv}_{\mathsf{F}}^{\mathrm{PRF}}(\mathsf{D}) \stackrel{\Delta}{=} \left| \; \Pr[\mathsf{D}^{\mathsf{F}_K^{\boldsymbol{\pi}}; \pi_1^{\pm}, \ldots, \pi_d^{\pm}} \to 1] - \Pr[\mathsf{D}^{\mathsf{RF}; \pi_1^{\pm}, \ldots, \pi_d^{\pm}} \to 1] \; \right|,$$

where $K \leftarrow\!\!\$\, \mathcal{K}, \pi_1, \ldots, \pi_d \leftarrow\!\!\$\, \mathsf{Perm}(n)$ and $\mathsf{RF} \leftarrow\!\!\$\, \mathsf{Func}(\mathcal{X}, \mathcal{Y})$. F is said to be a (q, q_p, t)-secure PRF if $\mathbf{Adv}_{\mathsf{F}}^{\mathrm{PRF}}(q, q_p, t) \stackrel{\Delta}{=} \max_{\mathsf{D}} \mathbf{Adv}_{\mathsf{F}}^{\mathrm{PRF}}(\mathsf{D}) \le \epsilon$, where the maximum is taken over all distinguishers D that makes q construction queries, total q_p primitive queries and runs for time at most t.

TES BASED ON PUBLIC RANDOM PERMUTATION. Let $\mathcal{K}, \mathcal{T}$ and $\mathcal{M}$ be three non-empty finite sets. A *tweakable enciphering scheme* (TES) $\mathfrak{T}$ is defined by a pair of efficient algorithms $\mathfrak{T} = (\mathsf{Enc}, \mathsf{Dec})$, where $\mathsf{Enc} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ and $\mathsf{Dec} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$. Let Enc and Dec be constructed by d many n-bit permutations $\boldsymbol{\pi} \stackrel{\Delta}{=} (\pi_1, \ldots, \pi_d)$, then we call them by $\mathsf{Enc}^{\boldsymbol{\pi}}$ and $\mathsf{Dec}^{\boldsymbol{\pi}}$. For all $k \in \mathcal{K}$ and all $T \in \mathcal{T}$, $\mathsf{Enc}_k^{\boldsymbol{\pi}}(T, \cdot)$ is a length preserving permutation over $\mathcal{M}$, i.e., $|\mathsf{Enc}_k^{\boldsymbol{\pi}}(T, M)| = |M|$ for all $M \in \mathcal{M}$. For the correctness, one requires that for all $k \in \mathcal{K}$, for all $T \in \mathcal{T}$, and for all $M \in \mathcal{M}$, $\mathsf{Dec}_k^{\boldsymbol{\pi}}(T, \mathsf{Enc}_k^{\boldsymbol{\pi}}(T, M)) = M$. A *tweakable permutation* with tweak space $\mathcal{T}$ and domain $\mathcal{M}$ is a mapping $\widetilde{\Pi} : \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ such that for all tweak $T \in \mathcal{T}$, $M \mapsto \widetilde{\Pi}(T, M)$ is a permutation of $\mathcal{M}$. We often write $\widetilde{\Pi}^T(M)$ for $\widetilde{\Pi}(T, M)$. $\mathsf{TP}(\mathcal{T}, \mathcal{M})$ denotes the set of all such tweakable permutations.

We consider the tweakable Strong Pseudo Random Permutation (tSPRP) security of $\mathfrak{T}$ in public permutation model where we assume that $\pi_1, \ldots, \pi_d \leftarrow\!\!\$\, \mathsf{Perm}(n)$ and the distinguisher D is given access to either the oracles $(\mathfrak{T}.\mathsf{Enc}_K^{\boldsymbol{\pi}}; \mathfrak{T}.\mathsf{Dec}_K^{\boldsymbol{\pi}};\, \pi_1^{\pm}, \ldots, \pi_d^{\pm})$ for a random key $K \leftarrow\!\!\$\, \mathcal{K}$ or the oracles $(\widetilde{\Pi}; \widetilde{\Pi}^{-1}; \pi_1^{\pm}, \ldots, \pi_d^{\pm})$ for $\widetilde{\Pi} \leftarrow\!\!\$\, \mathsf{TP}(\mathcal{T}, \mathcal{M})$. We call such a distinguisher as Chosen Ciphertext Attack (CCA) distinguisher. We define the tSPRP advantage of $\mathfrak{T}$ in public permutation model with respect to the CCA distinguisher D that makes q_e encryption queries to the first oracle, q_d decryption queries to the second oracle and altogether q_p primitive queries as

$$\mathbf{Adv}_{\mathfrak{T}}^{\mathrm{tSPRP}}(\mathsf{D}) \triangleq \left|\ \Pr[\mathsf{D}^{\mathfrak{T}.\mathsf{Enc}_K^{\boldsymbol{\pi}}; \mathfrak{T}.\mathsf{Dec}_K^{\boldsymbol{\pi}}; \pi_1^{\pm}, \ldots, \pi_d^{\pm}} \rightarrow 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}; \widetilde{\Pi}^{-1}; \pi_1^{\pm}, \ldots, \pi_d^{\pm}} \rightarrow 1]\ \right|,$$

where $K \leftarrow\!\!\$\, \mathcal{K}, \pi_1, \ldots, \pi_d \leftarrow\!\!\$\, \mathsf{Perm}(n)$ and $\widetilde{\Pi} \leftarrow\!\!\$\, \mathsf{TP}(\mathcal{T}, \mathcal{M})$. $\mathfrak{T}$ is said to be a $(q_e, q_d, q_p, \ell, \sigma, t)$-secure tSPRP if

$$\mathbf{Adv}_{\mathfrak{T}}^{\mathrm{tSPRP}}(q_e, q_d, q_p, \ell, \sigma, t) \triangleq \max_{\mathsf{D}} \mathbf{Adv}_{\mathfrak{T}}^{\mathrm{tSPRP}}(\mathsf{D}) \leq \epsilon,$$

where the maximum is taken over all CCA distinguishers D that run at most time t and make q_e encryption, q_d decryption and altogether q_p primitive queries with a maximum of ℓ data blocks present in a single encryption or decryption queried message and total σ many data blocks queried throughout all the encryption and decryption queries.

In all of the above definitions of security advantage, we omit the time parameter t for information-theoretic distinguisher [4]. In the rest of the paper, we assume information-theoretic *non-trivial* distinguishers, i.e., they do not ask duplicate queries or queries to which they already can compute the answers by themselves from the earlier query-response. Since, we assume the distinguishers are computationally unbounded, without loss of generality, we limit the distinguishers to be deterministic.

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 $\mathsf{H} : \mathcal{K}_h \times \mathcal{X} \rightarrow \{0,1\}^n$. 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 $\delta \in \{0,1\}^n$,

$$\Pr[K_h \leftarrow\!\!\$\, \mathcal{K}_h : \mathsf{H}_{K_h}(X) \oplus \mathsf{H}_{K_h}(X') = \delta] \leq \epsilon. \tag{1}$$

Moreover, H is said to be an ϵ-Almost Regular (AR) hash function if for any $X \in \mathcal{X}$ and for any $\delta \in \{0,1\}^n$,

$$\Pr[K_h \leftarrow\!\!\$\, \mathcal{K}_h : \mathsf{H}_{K_h}(X) = \delta] \leq \epsilon. \tag{2}$$

A keyed hash function is said to be an $(\epsilon_{\mathrm{axu}}, \epsilon_{\mathrm{reg}})$-AXUAR hash function if it is ϵ_{axu}-AXU and ϵ_{reg}-AR hash function.

PolyHash Function. PolyHash [52] is one of the popular examples of algebraic hash function, defined as follows: for a fixed key $k_h \in \{0,1\}^n$ and for a message $M \in \{0,1\}^*$, we first apply a padding rule 0^* i.e., pad the minimum number of zeros to the end of M,

[4] An information-theoretic distinguisher is the one who is computationally unbounded but can make a limited number of queries to its available oracles.

so that the total number of bits in the padded message becomes a multiple of n. Let the padded message be $M^* = M_1 \| M_2 \| \ldots \| M_l$ where $l = \lceil |M|/n \rceil$ and for each i, $|M_i| = n$. Then,

$$\mathsf{Poly}_{k_h}(M) = M_1 \cdot {k_h}^{l+1} \oplus M_2 \cdot {k_h}^{l} \oplus \ldots \oplus M_l \cdot k_h^2 \oplus \langle |M| \rangle \cdot k_h, \tag{3}$$

where l is the number of blocks of M^* and the multiplications in Eqn. (3) are in the field $\mathrm{GF}(2^n)$. If $M = \varepsilon$, the empty string, we define $\mathsf{Poly}_{k_h}(\varepsilon) = k_h^2 \oplus k_h$. Note that the use of the non-injective padding rule (i.e., appending 0^* at the end of the message) does not make the hash function insecure as the definition includes the message length information which is the safeguard against the xor universal attack. The following result says that the PolyHash defined in Eqn. (3) with an n-bit key, is an $(\ell/2^n, \ell/2^n)$-AXUAR hash function, where ℓ is the maximum number of message blocks. Proof of the lemma is straightforward and hence omitted.

Lemma 1. PolyHash *as defined in Eqn.* (3) *is* $(\ell/2^n, \ell/2^n)$-AXUAR *hash function.*

2.2 An Useful Result

Let $\mathfrak{T}$ be a public permutation based tweakable enciphering scheme over the message space $\mathcal{M}$ and the tweak space $\mathcal{T}$. Let us assume that $\mathfrak{T}$ is based on d many permutations $\pi_1, \ldots, \pi_d$. Let $\$_0$ and $\$_1$ are two functions sampled uniformly and independently from $\mathsf{Func}(\mathcal{M}, \mathcal{M})$ and $\pi_1, \ldots, \pi_d$ are d many n-bit random permutations sampled uniformly and independently from $\mathsf{Perm}(n)$. Then, the following result says that a uniform length-preserving random permutation is very close to a uniform length-preserving random function. More formally,

Theorem 1. *Let $\mathfrak{T}$ be a public permutation based* TES *over a message space $\mathcal{M} \subseteq \{0,1\}^*$ which is based on d many n-bit independent random permutations $\pi_1, \ldots, \pi_d$. Let $\$_0$ and $\$_1$ are two functions sampled uniformly and independently from $\mathsf{Func}(\mathcal{M}, \mathcal{M})$ and $\pi_1, \ldots, \pi_d$ are d many n-bit random permutations sampled independently to $\$_0$ and $\$_1$. Then, for any information theoretic non-trivial* CCA *distinguisher* D, *making altogether q encryption and decryption queries and total q_p primitive queries, we have,*

$$\begin{aligned}\mathbf{Adv}_{\mathfrak{T}}^{\mathrm{tSPRP}}(\mathsf{D}) \leq & \underbrace{\left| \Pr[\mathsf{D}^{\mathfrak{T}.\mathsf{Enc}_K^{\boldsymbol{\pi}}; \mathfrak{T}.\mathsf{Dec}_K^{\boldsymbol{\pi}}; \pi_1^{\pm}, \ldots, \pi_d^{\pm}} \to 1] - \Pr[\mathsf{D}^{\$_0; \$_1; \pi_1^{\pm}, \ldots, \pi_d^{\pm}} \to 1] \right|}_{\mathbf{Adv}_{\mathfrak{T}}^{\pm\mathrm{rnd}}(\mathsf{D})} \\ & + \frac{q(q-1)}{2^{m+1}}, \quad \textit{where } m = \min\{\ell : \mathcal{M} \cap \{0,1\}^{\ell} \neq \phi\}.\end{aligned} \tag{4}$$

The above result has been already been used in the standard model in several places including in [12, 42]. The proof of Theorem 1 is very similar to the proof given in [42] and hence we omit it here.

2.3 H-Coefficient Technqiue

In this section, we briefly discuss the H-Coefficient Technique, which was introduced by Patarin [48] and regained attention since the work of Chen and Steinberger [23] to analyze the security of iterated Even-Mansour [36] cipher. Since then, it has been successfully used as a tool to upper bound the statistical distance between the responses of two interactive systems and is typically used to prove the pseudo randomness of several constructions against information theoretic distinguishers. We consider a information theoretic deterministic distinguisher D with access to either the real oracle, i.e., the construction of our interest, or the ideal oracle which is usually considered to be a uniform random function or permutation. The collection of all the queries made by D to the oracle and the responses received by D from the oracle, is called the *attack transcript* of D, denoted as τ. Sometimes, we allow the oracle to release more internal information to D only after it completes all its queries, but before it outputs the decision bit. In this case, the transcript of D includes the additional information about the oracle and clearly the maximum distinguishing advantage of D in this setting can not be less than that of without additional information. The transcript τ is a random variable and the randomness of the distribution of τ comes only from the randomness of the oracle with which D interacts.

Let $\mathsf{T}_{\rm re}$ and $\mathsf{T}_{\rm id}$ denote the random variable that takes the transcript τ resulting from the interaction between D and the real world or between D and the ideal world respectively. The probability of realizing a transcript τ in the real (resp. ideal) world is called the *real (resp. ideal) interpolation probability*. A transcript τ is said to be *attainable* with respect to D if its ideal interpolation probability is non-zero (i.e., $\Pr[\mathsf{T}_{\rm id} = \tau] > 0$). We denote the set of all attainable transcripts by $\mathcal{V}$. Following these notations, we state the main theorem of H-Coefficient Technique [48, 23] as follows:

Theorem 2 (H-Coefficient Technique). *Let D be a fixed deterministic distinguisher that has access to either the real oracle $\mathcal{O}_{\rm re}$ or the ideal oracle $\mathcal{O}_{\rm id}$. Let $\mathcal{V} = \mathcal{V}_{\rm g} \cup \mathcal{V}_{\rm b}$, $\mathcal{V}_{\rm g} \cap \mathcal{V}_{\rm b} = \emptyset$, be some partition of the set of all attainable transcripts of D. Suppose there exists $\epsilon_{\rm ratio} \geq 0$ such that for any $\tau \in \mathcal{V}_{\rm g}$,*

$$\frac{\Pr[\mathsf{T}_{\rm re} = \tau]}{\Pr[\mathsf{T}_{\rm id} = \tau]} \geq 1 - \epsilon_{\rm ratio},$$

and there exists $\epsilon_{\rm bad} \geq 0$ such that $\Pr[\mathsf{T}_{\rm id} \in \mathcal{V}_{\rm b}] \leq \epsilon_{\rm bad}$. Then,

$$\mathbf{Adv}^{\mathcal{O}_{\rm id}}_{\mathcal{O}_{\rm re}}(\mathsf{D}) \stackrel{\Delta}{=} |\Pr[\mathsf{D}^{\mathcal{O}_{\rm re}} \rightarrow 1] - \Pr[\mathsf{D}^{\mathcal{O}_{\rm id}} \rightarrow 1]| \leq \epsilon_{\rm ratio} + \epsilon_{\rm bad}. \tag{5}$$

3 HCTR Construction

HCTR is one of the popular tweakable enciphering modes, proposed by Wang et al. [51], that turns an n-bit strong pseudorandom permutation into a variable length tweakable

$\mathsf{HCTR.Enc}_{k,k_h}(T, M)$	$\mathsf{HCTR.Dec}_{k,k_h}(T, C)$
1. $M_1\Vert\ldots\Vert M_l \leftarrow \mathsf{parse}_n(M)$;	1. $C_1\Vert\ldots\Vert C_l \leftarrow \mathsf{parse}_n(C)$;
2. $\mathbf{M_L} \leftarrow M_1; \mathbf{M_R} \leftarrow (M_2\Vert\ldots\Vert M_l)$;	2. $\mathbf{C_L} \leftarrow C_1; \mathbf{C_R} \leftarrow (C_2\Vert\ldots\Vert C_l)$;
3. $U \leftarrow \mathbf{M_L} \oplus \mathsf{Poly}_{k_h}(\mathbf{M_R}\Vert T)$;	3. $V \leftarrow \mathbf{C_L} \oplus \mathsf{Poly}_{k_h}(\mathbf{C_R}\Vert T)$;
4. $V \leftarrow \mathsf{E}_k(U)$; $Z \leftarrow U \oplus V$;	4. $U \leftarrow \mathsf{E}_k^{-1}(V)$; $Z \leftarrow U \oplus V$;
5. **for** $i = 1$ to l	5. **for** $i = 1$ to l
6. $\quad S_i \leftarrow \mathsf{E}_k(Z \oplus i)$;	6. $\quad S_i \leftarrow \mathsf{E}_k(Z \oplus i)$;
7. $\mathbf{S} \triangleq S_1\Vert\ldots\Vert S_l$;	7. $\mathbf{S} \triangleq S_1\Vert\ldots\Vert S_l$;
8. $\mathbf{C_R} \leftarrow \mathsf{first}(\vert\mathbf{M_R}\vert, \mathbf{S}) \oplus \mathbf{M_R}$;	8. $\mathbf{M_R} \leftarrow \mathsf{first}(\vert\mathbf{C_R}\vert, \mathbf{S}) \oplus \mathbf{C_R}$;
9. $\mathbf{C_L} \leftarrow V \oplus \mathsf{Poly}_{k_h}(\mathbf{C_R}\Vert T)$;	9. $\mathbf{M_L} \leftarrow U \oplus \mathsf{Poly}_{k_h}(\mathbf{M_R}\Vert T)$;
10. **return** $(\mathbf{C_L}\Vert\mathbf{C_R})$;	10. **return** $(\mathbf{M_L}\Vert\mathbf{M_R})$;

Fig. 3.1. HCTR construction based on an n-bit block cipher E_k and an n-bit Polyhash function. Left part of the algorithm is the encryption function and right part is the decryption function.

strong pseudorandom permutation. The encryption and decryption algorithm of HCTR is shown in Fig. 3.1 and its pictorial representation is shown in Fig. 3.2.

We explain the encryption algorithm of HCTR using an example. The decryption algorithm can be understood in a similar way. Suppose the input message $M = (M_1\Vert M_2)$ and for the sake of simplicity, we assume that $|M_1| = |M_2| = n$, i.e., M consists of two full blocks. Therefore, in step (2) of the algorithm, the variable $\mathbf{M_L}$ is assigned to M_1 and $\mathbf{M_R}$ is assigned to M_2. In step (3) of the algorithm, we evaluate the poly hash Poly_{k_h} on $(M_2\Vert T)$ which results to $M_2 \cdot k_h^3 \oplus T \cdot k_h^2 \oplus \langle |M_2| + |T| \rangle \cdot k_h$ which is xored with the n-bit value M_1 to produce U. In step (4), we take the xor of U and its encryption $V = \mathsf{E}_k(U)$ to produce Z. In step (6), we compute the key stream $\mathbf{S} = S_1\Vert S_2$ where each $|S_1| = |S_2| = n$. Since, $|\mathbf{M_R}| = n$, $\mathbf{C_R}$ will be $M_2 \oplus S_1$, which becomes the input along with tweak T to the poly hash function Poly_{k_h}. Evaluation of the poly hash on input $\mathbf{C_R}\Vert T$ results to $\mathbf{C_R} \cdot k_h^3 \oplus T \cdot k_h^2 \oplus \langle |\mathbf{C_R}| + |T| \rangle \cdot k_h$. Then the result is xored with V to produce $\mathbf{C_L}$, which is returned along with $\mathbf{C_R}$ as the encryption of $M = M_1\Vert M_2$.

Wang et al. [51] have shown that HCTR is a secure TES against all adaptive chosen plaintext and chosen ciphertext adversaries that make roughly $2^{n/3}$ encryption and decryption queries. Later Chakraborty and Nandi [19] improved its security bound to $O(\sigma^2/2^n)$, where σ is the total number of message blocks among all q queries. Recently, Dutta and Nandi [34] proposed a tweakable block cipher based HCTR, called *tweakable* HCTR, and showed its security beyond the birthday bound.

Remark 1. In [51], authors defined the output of the PolyHash to be the hash key k_h for ε. But that definition of the PolyHash function leads to an attack on the construction

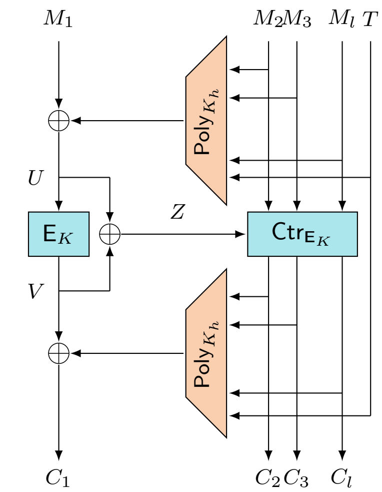

Fig. 3.2. HCTR construction with tweak T and message $M_1 \| M_2 \| \ldots \| M_l$ and the corresponding ciphertext $C_1 \| C_2 \| \ldots \| C_l$. Poly_{K_h} is the polynomial hash function with hash key K_h. $\mathsf{Ctr}_{\mathsf{E}_K}$ is the block cipher based counter mode of encryption.

as reported in [41]. This attack does not work if the message space contains messages of length at least $n + 1$. We redefine the output of the $\mathsf{PolyHash}$ for an empty input string to be $k_h^2 \oplus k_h$, which eliminates the message length restriction.

Motivated by HCTR, we first replace the block cipher based counter mode part of HCTR with a public permutation based length expanding PRF, and the block cipher E_K (see Fig. 3.2) with a public permutation π. We show that such combination yields a secure public permutation based TES, which we call ppTES as described in section 4. In section 6, we construct a public permutation based length expanding PRF, which we call ppCTR. Using ppCTR along with the the PolyHash function, we instantiate ppTES to realize a public permutation based TES, which we call ppHCTR. However, ppHCTR requires two independent public permutations, a key for the ppCTR and another independent hash key for the PolyHash function. Next, we go one step further to reduce the number of keys and permutations used in ppHCTR and come up with a single keyed (for the PolyHash function) and single permutation based TES construction, $\mathsf{ppHCTR+}$. We describe $\mathsf{ppHCTR+}$ in section 7.

4 ppTES : A Generic Public Permutation Based TES

ppTES is based on three cryptographic components: (i) an n-bit public random permutation π_1, (ii) an AXUAR hash function H_{k_h} which maps $\{0,1\}^*$ to $\{0,1\}^n$, and (iii) a public permutation based length expanding PRF $\mathsf{F}_k^{\pi_2}$, where π_2 is a n-bit independent public random permutation independent of π_1. The message space of ppTES is $\{0,1\}^{\geq n}$ and the tweak space is $\{0,1\}^{\mathtt{tw}}$. The working principle of ppTES is exactly same as HCTR where the block cipher is replaced by a public permutation π_1 and the counter mode encryption is replaced by a public permutation based length expanding PRF $\mathsf{F}_k^{\pi_2}$. The algorithmic description of encryption and decryption function of ppTES is shown in Fig. 4.1. The description in Fig. 4.1 mentions $\mathsf{F}_k^{\pi_2}$, which is a length expanding PRF. We describe this primitive next.

$\mathsf{ppTES.Enc}_{k,k_h}^{\pi_1,\pi_2}(T, M)$	$\mathsf{ppTES.Dec}_{k,k_h}^{\pi_1,\pi_2}(T, C)$
1. $M_1\\|\ldots\\|M_l \leftarrow \mathsf{parse}_n(M)$;	1. $C_1\\|\ldots\\|C_l \leftarrow \mathsf{parse}_n(C)$;
2. $\mathbf{M_L} \leftarrow M_1; \mathbf{M_R} \leftarrow (M_2\\|\ldots\\|M_l)$;	2. $\mathbf{C_L} \leftarrow C_1; \mathbf{C_R} \leftarrow (C_2\\|\ldots\\|C_l)$;
3. $U \leftarrow \mathbf{M_L} \oplus \mathsf{H}_{k_h}(\mathbf{M_R}\\|T)$;	3. $V \leftarrow \mathbf{C_L} \oplus \mathsf{H}_{k_h}(\mathbf{C_R}\\|T)$;
4. $V \leftarrow \pi_1(U)$; $Z \leftarrow U \oplus V$;	4. $U \leftarrow \pi_1^{-1}(V)$; $Z \leftarrow U \oplus V$;
5. $\mathbf{S} \triangleq S_1\\|\ldots\\|S_{\ell-1} \leftarrow \mathsf{F}_k^{\pi_2}(Z, l)$;	5. $\mathbf{S} \triangleq S_1\\|\ldots\\|S_{\ell-1} \leftarrow \mathsf{F}_k^{\pi_2}(Z, l)$;
6. $\mathbf{C_R} \leftarrow \mathsf{first}(\lvert\mathbf{M_R}\rvert, \mathbf{S}) \oplus \mathbf{M_R}$;	6. $\mathbf{M_R} \leftarrow \mathsf{first}(\lvert\mathbf{C_R}\rvert, \mathbf{S}) \oplus \mathbf{C_R}$;
7. $\mathbf{C_L} \leftarrow V \oplus \mathsf{H}_{k_h}(\mathbf{C_R}\\|T)$;	7. $\mathbf{M_L} \leftarrow U \oplus \mathsf{H}_{k_h}(\mathbf{M_R}\\|T)$;
8. **return** $(\mathbf{C_L}\\|\mathbf{C_R})$;	8. **return** $(\mathbf{M_L}\\|\mathbf{M_R})$;

Fig. 4.1. ppTES based on an n-bit public random permutations π_1, an AXUAR hash function H_{k_h} and a public permutation based length expanding PRF $\mathsf{F}_k^{\pi_2}$. $M \in \{0,1\}^{\geq n}$ is the input message and $T \in \{0,1\}^{\mathtt{tw}}$ is the tweak. Left part of the algorithm is the encryption function and right part is the decryption function.

As in case of HCTR to explain the encryption algorithm we use a two block message $M = (M_1\|M_2)$, where $|M_1| = |M_2| = n$. On input M, in step (2) of the algorithm, the variable $\mathbf{M_L}$ is assigned to M_1 and $\mathbf{M_R}$ is assigned to M_2. In step (3) of the algorithm, we evaluate the hash value H_{k_h} on $(M_2\|T)$ which is xored with the n-bit value M_1 to produce U. In step (4), we take the xor of U and its permuted value $V = \pi_1(U)$ to produce Z. In step (5), we compute the key stream $\mathbf{S} = S_1$ using length expanding PRF $\mathsf{F}_k^{\pi_2}$ where $|S_1| = n$. Since, $|\mathbf{M_R}| = n$, $\mathbf{C_R}$ will be $M_2 \oplus S_1$, which becomes the input along with tweak T to the hash function H_{k_h}. Then the resulting hash value is xored with V to produce $\mathbf{C_L}$, which is returned along with $\mathbf{C_R}$ as the encryption of $M = M_1\|M_2$.

4.1 Length Expanding Pseudorandom Function

For an arbitrary large positive integer L, Let $\mathcal{F} \subseteq \mathsf{Func}(\{0,1\}^n \times \mathbb{N}, \cup_{0<i\leq L}\{0,1\}^{ni})$, such that $F \in \mathcal{F}$ if and only if the following two conditions are satisfied:

1. For every $x \in \{0,1\}^n$ and every $b \in [L]$, $|F(x,b)| = nb$.
2. For every $x \in \{0,1\}^n$ and every $b, b' \in [L]$, $b \geq b'$, $\mathsf{first}(nb', F(x,b)) = F(x,b')$.

We call a uniform random element of $\mathcal{F}$ a *length expanding random function*.

In Fig. 4.2 we give an algorithmic description of a length expanding random function ρ. The algorithm depicts ρ as a lazy sampler which gives as output $\rho(x,b)$ upon receiving a query (x,b). For any input (x,b), it first checks whether x is a fresh element or not. If it is fresh, then it samples b many blocks uniformly at random from $\{0,1\}^{nb}$. If it is not fresh, then it first checks whether the number of requested blocks b' in the earlier query for input x is less than the number of requested blocks in the current query for the same input. In that case, it first fetches b' many blocks which are already stored at $\mathbb{T}[x]$, and then samples the remaining blocks, i.e., $b - b'$ blocks independently and uniformly at random from $\{0,1\}^{n(b-b')}$ which is appended with the first b' many fetched blocks and finally updates the entry $\mathbb{T}[x]$ with the output of the current query. The final case is if the number of requested blocks in the current query for input x is less than the number of requested blocks in the earlier query with the same input. Then it fetches the first b many blocks out of b' many blocks which are already stored at $\mathbb{T}[x]$ and returns it.

Informally, length expanding pseudorandom function is a function which is indistinguishable from a length expanding random function by any efficient distinguisher. For the sake of our construction, we require a public permutation based length expanding PRF which we formally define next.

Definition 1. Public Permutation Based Length Expanding PRF . *Let L be an arbitrary large positive integer and let $\mathsf{F} : \mathcal{K} \times \{0,1\}^n \times [L] \rightarrow \cup_{1 \leq i \leq L} \{0,1\}^{ni}$ be a keyed function based on d many n-bit permutations $\boldsymbol{\pi} \triangleq (\pi_1, \ldots, \pi_d)$ such that for any input $(x,b) \in \{0,1\}^n \times [L]$, $\mathsf{F}_k^{\boldsymbol{\pi}}(x,b)$ returns $(y_1, \ldots, y_b)$ where each $y_i \in \{0,1\}^n$. We consider the length expanding* PRF *security of* F *under public permutation model where we assume that $\pi_1, \ldots, \pi_d \leftarrow\$ \mathsf{Perm}(n)$ and the distinguisher* D *is given access to either of the world $(\mathsf{F}_K^{\boldsymbol{\pi}}, \pi_1^{\pm}, \ldots, \pi_d^{\pm})$ for a random key $K \leftarrow\$ \mathcal{K}$ or $(\rho, \pi_1^{\pm}, \ldots, \pi_d^{\pm})$, where ρ works as shown in Fig 4.2. We define the* LENPRF *advantage of* F *in public permutation model with respect to the distinguisher* D *that makes q construction queries and total q_p primitive queries as*

$$\mathbf{Adv}_{\mathsf{F}}^{\mathrm{LENPRF}}(\mathsf{D}) \triangleq \left|\ \Pr[\mathsf{D}^{\mathsf{F}_K^{\boldsymbol{\pi}}, \pi_1^{\pm}, \ldots, \pi_d^{\pm}} \rightarrow 1] - \Pr[\mathsf{D}^{\rho, \pi_1^{\pm}, \ldots, \pi_d^{\pm}} \rightarrow 1]\ \right|,$$

where $K \leftarrow\$ \mathcal{K}, \pi_1, \ldots, \pi_d \leftarrow\$ \mathsf{Perm}(n)$. F is said to be a (q, q_p, σ, t)-secure LENPRF if $\mathbf{Adv}_{\mathsf{F}}^{\mathrm{LENPRF}}(q, q_p, \sigma, t) \triangleq \max_{\mathsf{D}} \mathbf{Adv}_{\mathsf{F}}^{\mathrm{LENPRF}}(\mathsf{D}) \leq \epsilon$, where the maximum is taken over all distinguishers D that makes q construction queries with total $\sigma = (b_1 + \ldots + b_q)$ blocks, where b_i is the number of blocks requested at i-th construction query. It also makes total q_p primitive queries and runs for time at most t. As before, for information theoretic distinguisher, we omit the time parameter t and in the rest of the paper, we assume the distinguisher is information theoretic.

Algorithm for ρ

`initialize:`
  `for all` $x \in \{0,1\}^n$
    $\mathbb{T}[x] \leftarrow \perp; \mathbb{L}[x] \leftarrow \perp;$
  `end for`;
`on input` $(x, b) \neq (x', b')$;
  `if` $x = x'$
    `if` $b > b'$, `then`
      $Y \triangleq (y_{b'+1}, y_{b'+2}, \ldots, y_b) \leftarrow_{\$} \{0,1\}^{n(b-b')}$;
      $\mathbb{T}[x] \leftarrow \mathbb{T}[x] \| Y$; $\mathbb{L}[x] \leftarrow b$; **return** $\mathbb{T}[x]$;
    `else` **return** $\mathbb{T}[x']_{1,\ldots,b}$;
    `end if`;
  `else`
    $Y \triangleq (y_1, \ldots, y_b) \leftarrow_{\$} \{0,1\}^{nb}$;
    $\mathbb{T}[x] \leftarrow Y$; $\mathbb{L}[x] \leftarrow b$;
    return $\mathbb{T}[x]$;
  `end if`;

Fig. 4.2. Algorithm corresponding to a length expanding random function. $\mathbb{T}[x]_{1,\ldots,b}$ denotes the first b many blocks stored at the x-th entry of table $\mathbb{T}$.

Remark 2. The length expanding PRF is a weaker notion than the notion of variable output length PRF [10]. For a length expanding PRF, if two queries have the same input with different number of requesting blocks, then one output is a prefix of other. In case of variable output length PRF, outputs for two queries are completely random even if they have the same input with different number of requesting blocks.

4.2 Security of ppTES

In this section, we show that if $\pi_1, \pi_2 \leftarrow_{\$} \mathsf{Perm}(n)$ are two independently sampled n-bit public random permutations, $K \leftarrow_{\$} \{0,1\}^n$ be a uniformly sampled n-bit key, H is an $(\epsilon_{\mathrm{axu}}, \epsilon_{\mathrm{reg}})$-AXUAR n-bit keyed hash function and $\mathsf{F}_K^{\pi_2}$ is a secure public permutation based length expanding PRF, then ppTES is a public permutation based secure TES against all $(q_e, q_d, q_{p_1} + q_{p_2}, \ell, \sigma)$ information theoretic adaptive CCA distinguishers that make q_e many encryption, q_d many decryption queries with total σ many blocks queried among all $q \triangleq q_e + q_d$ queries and ℓ is the maximum number of message blocks present in a single encryption or decryption query. Moreover, it also makes q_{p_1} primitive queries to π_1 and q_{p_2} primitive queries to π_2. Formally, the following result bounds the tSPRP advantage of ppTES in public permutation model.

Theorem 3. *Let $\mathcal{K}_h$ be a finite and non-empty set, $\pi_1, \pi_2 \leftarrow\!\!\$\ \mathsf{Perm}(n)$ be two independently sampled n-bit public random permutations and $K \leftarrow\!\!\$\ \{0,1\}^n$ be an n-bit random key. Let $\mathsf{H} : \mathcal{K}_h \times \{0,1\}^* \rightarrow \{0,1\}^n$ be an $(\epsilon_{\text{axu}}, \epsilon_{\text{reg}})$-AXUAR n-bit keyed hash function. Let $\mathsf{F}_K^{\pi_2}$ be a secure LENPRF. Then, for any $(q_e, q_d, q_{p_1} + q_{p_2}, \ell, \sigma)$ information theoretic adaptive CCA distinguisher D against the tSPRP security of $\mathsf{ppTES}[\pi_1, \pi_2, K, \mathsf{H}]$ in the public permutation model, there exists a LENPRF adversary B against the length expanding PRF security of $\mathsf{F}_K^{\pi_2}$ in the public permutation model, where σ is the total number of message blocks queried, such that*

$$\mathbf{Adv}_{\mathsf{ppTES}}^{\text{tSPRP}}(\mathsf{D}) \leq \mathbf{Adv}_{\mathsf{F}}^{\text{LENPRF}}(\mathsf{B}) + q^2\epsilon_{\text{axu}} + 2qq_{p_1}\epsilon_{\text{reg}} + \frac{q^2}{2^{n+1}} + \frac{q(q-1)}{2^{n+1}}.$$

The proof of this result is given in section 5.

5 Proof of Theorem 3

As a matter of convenience, we refer to the construction $\mathsf{ppTES}[\pi_1, \pi_2, K, \mathsf{H}]$ as simply ppTES when the underlying primitives are assumed to be understood.

5.1 Initial Set Up

By Theorem 1, we have

$$\mathbf{Adv}_{\mathsf{ppTES}}^{\text{tSPRP}}(\mathsf{D}) \leq \mathbf{Adv}_{\mathsf{ppTES}}^{\pm\text{rnd}}(\mathsf{D}) + \frac{q(q-1)}{2^{n+1}}, \tag{6}$$

where recall that n is the minimum message length allowed for ppTES. Therefore, we bound the $\pm$rnd advantage of ppTES. Let D be any information theoretic non-trivial adaptive deterministic CCA distinguisher with access to the oracles in either of the following two worlds: in the real world it interacts with $\mathcal{O}_{\text{re}} = (\mathsf{ppTES.Enc}_{K,K_h}^{\pi_1,\pi_2}, \mathsf{ppTES.Dec}_{K,K_h}^{\pi_1,\pi_2}, \pi_1^{\pm}, \pi_2^{\pm})$ for an n-bit random key K, a random hash key K_h and two independent n-bit random permutations π_1 and π_2 or in the ideal world it interacts with $\mathcal{O}_{\text{id}} = (\$_0, \$_1, \pi_1^{\pm}, \pi_2^{\pm})$, where $\$_0$ and $\$_1$ are two independent random functions that output uniform random strings for every distinct input. Now, our goal is to upper bound the maximum advantage in distinguishing the real world from the ideal one.

For doing this, as the first step of the proof, we replace $\mathsf{F}_K^{\pi_1,\pi_2}$ with the function ρ as described in Fig. 4.2. We call the resulting construction as ppTES^*. This replacement comes at the cost of the length expanding PRF security of $\mathsf{F}_K^{\pi'}$ in the random permutation model, where the PRF adversary B simulates D as follows: it first samples a hash key $K_h \leftarrow\!\!\$\ \mathcal{K}_h$ and an n-bit random permutation $\pi \leftarrow\!\!\$\ \mathsf{Perm}(n)$. Then, for any input (M, T), it computes

$$Z \leftarrow \pi_1(\mathsf{H}_{K_h}(\mathbf{M_R} \| T) \oplus \mathbf{M_L}) \oplus \mathsf{H}_{K_h}(\mathbf{M_R} \| T) \oplus \mathbf{M_L}.$$

Then it calls its own oracle with $(Z, \lceil \frac{|M|}{n} \rceil)$ as input and receives the $n\lceil \frac{|M|}{n} \rceil$ bit output $\mathbf{S}$. Then it masks the first $|\mathbf{M_R}|$ bits of $\mathbf{S}$ with $\mathbf{M_R}$ and produces the ciphertext blocks $\mathbf{C_R}$ which is hashed along with T and the hash output is masked with $\pi_1(\mathsf{H}_{K_h}((\mathbf{M_R}\|T)\oplus\mathbf{M_L})$ to generate the first ciphertext block $\mathbf{C_L}$. For any primitive query x made by D to π_1, B accordingly returns the value $\pi_1(x)$. Similarly, it returns the response for backard query to π_1. For any primitive query x made by D to π_2, B forwards the query to its own oracle and returns the received response. Similarly, it returns the response for backward query to π_2. Finally B outputs the same bit as returned by D. Therefore, we have

$$\mathbf{Adv}^{\pm\text{rnd}}_{\mathsf{ppTES}}(\mathsf{D}) \leq \mathbf{Adv}^{\text{LENPRF}}_{\mathsf{F}}(\mathsf{B}) + \underbrace{\mathbf{Adv}^{\pm\text{rnd}}_{\mathsf{ppTES}^*}(\mathsf{D})}_{\delta^*}. \tag{7}$$

5.2 Attack Transcript

Our main goal is to bound δ^*, i.e., we need to distinguish the two worlds: the real world $\mathcal{O}_{\text{re}} = (\mathsf{ppTES}^*.\mathsf{Enc}^{\pi_1,\pi_2}_{K,K_h}, \mathsf{ppTES}^*.\mathsf{Dec}^{\pi_1,\pi_2}_{K,K_h}, \pi_1^{\pm}, \pi_2^{\pm})$ from the ideal world $\mathcal{O}_{\text{id}} = (\$_0, \$_1, \pi_1^{\pm}, \pi_2^{\pm})$, where K is an n-bit random key, K_h is a random hash key and π_1, π_2 are two independent n-bit random permutations. Since, we consider the maximum distinguishing advantage, let us assume that D be the information theoretic non-trivial adaptive CCA distinguisher for which the distinguishing advantage is maximum. Let D makes q_e (resp. q_d) encryption (resp. decryption) queries and q_{p_1} primitive queries to π_1 and q_{p_2} primitive queries to π_2. Since, our proof is in random permutation model, D can query the primitive in forward and reverse direction. After the interaction is over, the real world returns the hash key K_h and the ideal world samples a dummy hash key $K_h \leftarrow_{\$} \mathcal{K}_h$ and returns it to D. Finally, D outputs a single bit. Let $\tau \triangleq \{(T^1, M^1, C^1), (T^2, M^2, C^2), \ldots, (T^q, M^q, C^q)\}$ be the list of construction queries and responses (i.e., including encryption and decryption queries), $\tau_{p_1} \triangleq \{(x_1, y_1), (x_2, y_2), \ldots, (x_{q_{p_1}}, y_{q_{p_1}})\}$ and $\tau_{p_2} \triangleq \{(u_1, v_1), (u_2, v_2), \ldots, (u_{q_{p_2}}, v_{q_{p_2}})\}$ be the two list of primitive queries and responses to π_1 and π_2 respectively made by D. The triplet $\tau' = (\tau, \tau_{p_1}, \tau_{p_2}, K_h)$ constitutes the query transcript of the attack.

5.3 Definition and Probabilty of Bad Transcripts

In this section, we define bad transcripts and bound their probability in the ideal world. From transcript τ', we derive the following notation: for $i \in q$, $U_i = M_1^i \oplus \mathsf{H}_{k_h}(M_2^i\| \ldots \|M_{l_i}^i\|T^i)$, $V_i = C_1^i \oplus \mathsf{H}_{k_h}(C_2^i\| \ldots \|C_{l_i}^i\|T^i)$ and $Z_i = U_i \oplus V_i$. Having set up the notation, we identify an event to be bad if for any two construction queries there is a collision in the Z_i values or there is a non-trivial input or output collision of the permutation π_1.

Definition 2 (Bad Transcript for ppTES^*). *An attainable transcript $\tau' = (\tau, \tau_p, \tau'_p, K_h)$ is called* **bad** *for ppTES^* if any of the following conditions hold:*

- $\mathsf{B.1}$ *:* $\exists\ i \neq j \in [q]$ *such that,* $U^i = U^j$*.*
- $\mathsf{B.2}$ *:* $\exists\ i \neq j \in [q]$ *such that* $V^i = V^j$*.*
- $\mathsf{B.3}$ *:* $\exists\ i \in [q]$ *and* $j \in [q_p]$ *such that* $U^i = x_j$*.*
- $\mathsf{B.4}$ *:* $\exists\ i \in [q]$ *and* $j \in [q_p]$ *such that* $V^i = y_j$*.*
- $\mathsf{B.5}$ *:* $\exists\ i, j \in [q]$ *such that* $Z^i = Z^j$*.*

Lemma 2. *Let* T_{id} *be the random variable that takes the transcript resulting from the interaction between the distinguisher and the ideal world and* $\mathcal{V}_{\mathrm{b}}$ *be the set of all attainable* **bad** *transcripts for* ppTES^**. Then we have,*

$$\Pr[\mathsf{T}_{\mathrm{id}} \in \mathcal{V}_{\mathrm{b}}] \leq \epsilon_{\mathrm{bad}} = q^2\epsilon_{\mathrm{axu}} + 2qq_p\epsilon_{\mathrm{reg}} + \frac{q^2}{2^{n+1}}.$$

Proof. By the union bound,

$$\Pr[\mathsf{T}_{\mathrm{id}} \in \mathcal{V}_{\mathrm{b}}] \leq \sum_{i=1}^{4} \Pr[\mathsf{B.i}] + \Pr[\mathsf{B.5} \mid \overline{\mathsf{B.1}} \wedge \overline{\mathsf{B.2}} \wedge \overline{\mathsf{B.3}} \wedge \overline{\mathsf{B.4}}]. \tag{8}$$

In the following, we bound the probability of all the bad events individually. The lemma will follow by adding the individual bounds.

Bounding $\mathsf{B.1}$. For two fixed values of i and j, we compute the probability of the event $U^i = U^j$. Note that $U^i = U^j$ implies the hash equation: $\mathsf{H}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{H}_{K_h}(\mathbf{M}^j_{\mathbf{R}} \| T^j) = M^i_1 \oplus M^j_1$. By fixing the value of all other random variables in the hash equation, the probability of this event is bounded by the AXU advantage of the hash function. Therefore, by summing over all possible choices of i and j, we have

$$\Pr[\mathsf{B.1}] \leq \binom{q}{2}\epsilon_{\mathrm{axu}}. \tag{9}$$

Bounding $\mathsf{B.2}$. This event is similar to that of $\mathsf{B.1}$ where we consider the output collision of π. Note that, $V^i = V^j$ implies the hash equation: $\mathsf{H}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{H}_{K_h}(\mathbf{C}^j_{\mathbf{R}} \| T^j) = C^i_1 \oplus C^j_1$. Similar to $\mathsf{B.1}$, we bound the event using the AXU advantage of the the hash function and thus we have

$$\Pr[\mathsf{B.2}] \leq \binom{q}{2}\epsilon_{\mathrm{axu}}. \tag{10}$$

Bounding $\mathsf{B.3}$. For two fixed values of i and j, we compute the probability of the event $U^i = x_j$. Note that $U^i = x_j$ implies the hash equation: $\mathsf{H}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) = M^i_1 \oplus x_j$. By fixing the value of all other random variables in the hash equation, the probability of this event is bounded by the AR advantage of the hash function. Therefore, by summing over all possible choices of i and j, we have

$$\Pr[\mathsf{B.3}] \leq qq_{p_1}\epsilon_{\mathrm{reg}}. \tag{11}$$

Bounding $\mathsf{B.4}$. For two fixed values of i and j, we compute the probability of the event $V^i = y_j$. Note that $V^i = y_j$ implies the hash equation: $\mathsf{H}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) = C^i_1 \oplus y_j$. Similar to $\mathsf{B.3}$, we bound the event using the AR advantage of the hash function and thus we have

$$\Pr[\mathsf{B.4}] \leq q q_{p_1} \epsilon_{\text{reg}}. \tag{12}$$

Bounding $\mathsf{B.5} \mid \overline{\mathsf{B.1}} \wedge \overline{\mathsf{B.2}} \wedge \overline{\mathsf{B.3}} \wedge \overline{\mathsf{B.4}}$. To bound this event, we first fix the value of i and j. Note that $Z^i = Z^j$ implies $U^i \oplus V^i = U^j \oplus V^j$. Now, due to the condition, we have $U^i \neq U^j$ and $V^i \neq V^j$. Therefore, we obtain the following hash equation:

$$\mathsf{H}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{H}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{H}_{K_h}(\mathbf{M}^j_{\mathbf{R}} \| T^j) \oplus \mathsf{H}_{K_h}(\mathbf{C}^j_{\mathbf{R}} \| T^j) = W, \tag{13}$$

where $W = M^i_1 \oplus M^j_1 \oplus C^i_1 \oplus C^j_1$. W.l.o.g we assume that $i < j$. If the j-th query is an encryption query, then C^j_1 is uniformly distributed in the ideal world and if the j-th query is a decryption query, then M^j_1 is uniformly distributed in the ideal world. Combining the above two arguments and by varying over all possible choices of indices, we have

$$\Pr[\mathsf{B.5}] \leq \frac{\binom{q}{2}}{2^n}. \tag{14}$$

The proof follows from Eqn. (8)-Eqn. (12) and Eqn. (14). $\square$

5.4 Analysis of Good Transcript

In this section, we show that for a good transcript $\tau' = (\tau, \tau_{p_1}, \tau_{p_2}, k_h)$, realizing τ' is almost as likely in the real world as in the ideal world.

Lemma 3. *Let $\tau' = (\tau, \tau_{p_1}, \tau_{p_2}, k_h)$ be a good transcript. Then*

$$\frac{\Pr[\mathsf{T}_{\text{re}} = \tau']}{\Pr[\mathsf{T}_{\text{id}} = \tau']} \geq 1.$$

Proof. Since, in the ideal world, the encryption and the decryption oracle behaves perfectly random, we have

$$\Pr[\mathsf{T}_{\text{id}} = \tau'] = \frac{1}{|\mathcal{K}_h|} \frac{1}{\mathbf{P}(2^n, q_{p_1})} \cdot \frac{1}{\mathbf{P}(2^n, q_{p_2})} \cdot \frac{1}{2^{n\sigma}}, \tag{15}$$

where σ is the total number of blocks queried among all q construction queries that includes encryption and decryption queries.

Real Interpolation Probability. Since, τ' is a good transcript, all the inputs and outputs of π_1 are fresh. Moreover, all Z_i values are distinct. Therefore, the outputs of ρ are all uniformly random. Since, there are total $q_{p_1} + q$ many invocations of π_1, we have

$$\Pr[\mathsf{T}_{\text{re}} = \tau'] = \frac{1}{|\mathcal{K}_h|} \frac{1}{\mathbf{P}(2^n, q_{p_1} + q)} \cdot \frac{1}{\mathbf{P}(2^n, q_{p_2})} \cdot \frac{1}{(2^n)^{\sigma - q}}. \tag{16}$$

By doing a simple algebraic calculation, it is easy to see that the ratio of Eqn. (16) to Eqn. (15) is at least 1 and hence proves the result. □

By combining Lemma 2, Lemma 3, Theorem 2, Eqn. (6) and Eqn. (7), the result follows. □

6 ppCTR: Public Permutation Based Length Expanding PRF

In this section, we propose ppCTR, a public permutation based length expanding PRF. Our proposed construction is a public permutation variant of the block cipher based standard counter mode encryption where the block cipher is replaced by a single round EM [36] cipher. The working principle of ppCTR is as follows: it takes an n-bit public random permutation π and an n-bit random key k. Then for any n-bit input value z and an integer b, it outputs b many blocks where the j-th block S_j is defined as follows:

$$S_j \stackrel{\Delta}{=} \pi(z \oplus \gamma^j k) \oplus \gamma^j k,\ j \in [b],$$

where γ is the root of a primitive polynomial of $\mathrm{GF}(2^n)$. In the following section, we

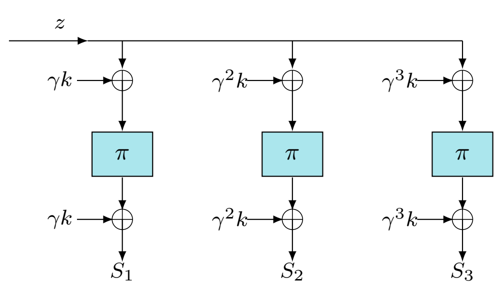

Fig. 6.1. ppCTR construction with an n-bit input z and an integer $b = 3$ and corresponding output $S_1 \| S_2 \| S_3$. π is the public random permutation, k is the key and γ is the root of a primitive polynomial of $\mathrm{GF}(2^n)$.

state and prove that ppCTR is a public permutation based secure LENPRF against all adversaries that makes roughly $2^{n/2}$ construction and primitive queries. It is needless to say that the above bound is tight as EM cipher is known to have a tight birthday bound security [36].

6.1 Security Analysis of ppCTR

In this section, we show that ppCTR is a public permutation based length expanding PRF.

Theorem 4. *Let* $\pi \leftarrow_\$ \mathsf{Perm}(n)$ *be an* n*-bit public random permutation and let* $K \leftarrow_\$ \{0,1\}^n$ *be an* n*-bit random key. Then, for any* (q, q_p, σ) *adversary* D *against the* LENPRF *security of* $\mathsf{ppCTR}[\pi, K]$*, we have*

$$\mathbf{Adv}^{\mathrm{LENPRF}}_{\mathsf{ppCTR}}(\mathsf{D}) \leq \frac{\sigma^2}{2^n} + \frac{2\sigma q_p}{2^n},$$

where σ *is the total number of blocks queried across all* q *queries.*

Proof. Let $\mathsf{D}_{\max}$ be the distinguisher with maximum distinguishing advantage in distinguishing the following two worlds: (a) in the real world it interacts with $\mathcal{O}_{\mathrm{re}} = (\mathsf{ppCTR}[\pi, K], \pi^{\pm})$ for a random n-bit key K and a random n-bit permutation π and (b) in the ideal world it has access to $\mathcal{O}_{\mathrm{id}} = (\rho, \pi^{\pm})$, where ρ works in the similar way as shown in Fig. 4.2. It makes q construction queries and q_p primitive queries. After the interaction is over, the real world returns K to $\mathsf{D}_{\max}$ and the ideal world randomly samples a dummy key $K \leftarrow_\$ \{0,1\}^n$ and returns to $\mathsf{D}_{\max}$. Finally, $\mathsf{D}_{\max}$ outputs a bit. Let $\tau \overset{\Delta}{=} \{(z_1, b_1, \mathbf{S}^1), (z_2, b_2, \mathbf{S}^2), \ldots, (z_q, b_q, \mathbf{S}^q)\}$ be the list of construction queries and responses, where $\mathbf{S}^i = (S^i_1, \ldots, S^i_{b_i})$ and $\tau_p \overset{\Delta}{=} \{(x_1, y_1), (x_2, y_2), \ldots, (x_{q_p}, y_{q_p})\}$ be the list of primitive queries and responses to π made by $\mathsf{D}_{\max}$. Let $\sigma = (b_1 + \ldots + b_q)$ denotes the total number of blocks queried across all q queries. The triplet $\tau' = (\tau, \tau_p, K)$ constitutes the query transcript of the attack. We define a relation $\sim$ over τ such that $(z_i, b_i, \mathbf{S}_i) \sim (z_j, b_j, \mathbf{S}_j)$ if and only if $z_i = z_j$. Thus, $\sim$ induces a partition on τ and let us assume we have r many such partitions. Each partition contains c_i many elements and therefore, $c_1 + \ldots + c_r = q$. Note that, there exists a total ordering among b_i values in each component. This allows us to sort the elements of each component in the ascending order of their b values. After rearrangement, we have the following:

$$\begin{cases} \{(z_1, b^1_1, \mathbf{S}^1_1), \ldots, (z_1, b^1_{c_1}, \mathbf{S}^1_{c_1})\} \\ \{(z_2, b^2_1, \mathbf{S}^2_1), \ldots, (z_2, b^2_{c_1}, \mathbf{S}^2_{c_2})\} \\ \quad\vdots \qquad\quad \vdots \qquad\quad \vdots \qquad\quad \vdots \\ \{(z_r, b^r_1, \mathbf{S}^r_1), \ldots, (z_r, b^r_{c_1}, \mathbf{S}^r_{c_1})\} \end{cases}$$

Note that, for each $i \in [r]$, $b^i_{c_i} \geq b^i_{c_i - 1} \geq \ldots \geq b^i_1$ and $\mathbf{S}^i_j$ is a prefix of $\mathbf{S}^i_{j+1}$ for all $j \in [c_i]$.

6.2 Definition and Probability of Bad Transcripts

In this section, we define bad transcripts and bound their probability in the ideal world. Informally, we define an event to be bad if it introduces any non-trivial input or output collision of the permutation π.

Definition 3. (**Bad Transcript for ppCTR**) : *An attainable transcript* $\tau' = (\tau, \tau_p, K)$ *is called a* **bad** *transcript for* ppCTR *if any of the following conditions hold:*

- B.1 : $\exists\ i \neq j \in [r]$, $\alpha \in [\ell_{c_i}]$ and $\beta \in [\ell_{c_j}]$ such that $z_i \oplus \gamma^\alpha K = z_j \oplus \gamma^\beta K$.
- B.2 : $\exists\ i \in [r]$, $j \in [q_p]$ and $\alpha \in [\ell_{c_i}]$ such that $z_i \oplus \gamma^\alpha K = x_j$.
- B.3 : $\exists\ i \neq j \in [r]$, $\alpha \in [\ell_{c_i}]$ and $\beta \in [\ell_{c_j}]$ such that $S^i_\alpha \oplus \gamma^\alpha K = S^j_\beta \oplus \gamma^\beta K$.
- B.4 : $\exists\ i \in [r]$, $j \in [q_p]$ and $\alpha \in [\ell_{c_i}]$ such that $S^i_\alpha \oplus \gamma^\alpha K = y_j$.

Lemma 4. *Let T_{id} be the random variable that takes the transcript resulting from the interaction between the distinguisher and the ideal world and $\mathcal{V}_{\mathrm{b}}$ be the set of all attainable* **bad** *transcripts for* ppCTR. *Then we have,*

$$\Pr[\mathsf{T}_{\mathrm{id}} \in \mathcal{V}_{\mathrm{b}}] \leq \epsilon_{\mathrm{bad}} = \frac{\sigma^2}{2^n} + \frac{2\sigma q_p}{2^n}.$$

Proof. By the union bound,

$$\Pr[\mathsf{T}_{\mathrm{id}} \in \mathcal{V}_{\mathrm{b}}] \leq \sum_{i=1}^{4} \Pr[\mathsf{B.i}]. \tag{17}$$

In the following, we bound the probability of all the bad events individually. The lemma will follow by adding the individual bounds.

Bounding B.1. To bound this event, we first fix a value of the indices $i \neq j \in [r]$ and $\alpha \in [\ell_{c_i}], \beta \in [\ell_{c_j}]$. For such a fixed choice of indices, we bound the probability of the event $z_i \oplus \gamma^\alpha K = z_j \oplus \gamma^\beta K$. Now, if $\alpha = \beta$, then the probability of the event is zero as $z_i \neq z_j$. Therefore, we assume that $\alpha \neq \beta$. For this choice of indices, we write the event as

$$K = (\gamma^\alpha \oplus \gamma^\beta)^{-1}(z^i \oplus z^j). \tag{18}$$

The probability of Eqn. (18) is 2^{-n}, due to the randomness of the key K. Therefore, by varying over all possible choices of i, j, α and β, we have

$$\Pr[\mathsf{B.1}] \leq \frac{\sigma^2}{2^{n+1}}. \tag{19}$$

Bounding B.2. For a fixed choice of $i \in [r], j \in [q_p]$ and $\alpha \in [\ell_{c_i}]$, the probability of the event $K = \gamma^{-\alpha}(z^i \oplus x_j)$ is bounded by 2^{-n} due to the randomness of K. Therefore, by varying over all possible choices of i, j and α, we have

$$\Pr[\mathsf{B.2}] \leq \frac{q_p}{2^n}(b_{c_1} + \cdots + b_{c_r}) \leq \frac{\sigma q_p}{2^n}. \tag{20}$$

Bounding B.3. Bounding this event is similar to that of B.1. To bound this event, we first fix the value of the indices $i \neq j \in [r]$ and $\alpha \in [\ell_{c_i}], \beta \in [\ell_{c_j}]$. For such a fixed choice of indices, we bound the probability of the event $S^i_\alpha \oplus \gamma^\alpha K = S^j_\beta \oplus \gamma^\beta K$. Now we have the following two cases:

- **Case A.** Let us consider that $\alpha = \beta$. As $i \neq j$, without loss of generality, we assume that $i < j$. Therefore, the event boils down to $S^i_\alpha = S^j_\alpha$, which is bounded by 2^{-n} due to the randomness of S^j_α. Therefore, by varying over all possible choices of i, j and α, we have
$$\Pr[\mathsf{B.3}] \leq \frac{\sigma^2}{2^{n+1}}$$
- **Case B.** if $\alpha \neq \beta$, then the event can be equivalently written as
$$K = (\gamma^\alpha \oplus \gamma^\beta)^{-1}(S^i_\alpha \oplus S^i_\beta). \tag{21}$$
Since, $\alpha \neq \beta$, we have $\gamma^\alpha \oplus \gamma^\beta \neq 0$ and therefore, the probability of Eqn. (21) is 2^{-n} due to the randomness of the key K. Therefore, by varying over all possible choices of i, j, α and β, we have
$$\Pr[\mathsf{B.3}] \leq \frac{\sigma^2}{2^{n+1}}.$$

By taking the maximum of the above two, we have
$$\Pr[\mathsf{B.3}] \leq \frac{\sigma^2}{2^{n+1}}. \tag{22}$$

Bounding B.4. Bounding this event is exactly identical to that of B.2, where we use the randomness of K to bound the event. Therefore, we have
$$\Pr[\mathsf{B.4}] \leq \frac{q_p}{2^n}(b_{c_1} + \cdots + b_{c_r}) \leq \frac{\sigma q_p}{2^n}. \tag{23}$$

The proof follows from Eqn. (17) and Eqn. (19)-Eqn. (23). □

6.3 Analysis of Good Transcript

In this section, we show that for a good transcript $\tau' = (\tau, \tau_p, k)$, realizing τ' is almost as likely in the real world as in the ideal world.

Lemma 5. *Let $\tau' = (\tau, \tau_p, k)$ be a good transcript. Then*
$$\frac{\Pr[\mathsf{T}_{\mathrm{re}} = \tau']}{\Pr[\mathsf{T}_{\mathrm{id}} = \tau']} \geq 1.$$

Proof. Consider a good transcript $\tau' = (\tau, \tau_p, k)$. In the ideal world, ρ randomly samples nb_{c_i} bit output for i-th class and the key k is sampled uniformly from $\{0,1\}^n$ and independent to all other sampled random variables. Thus, we have
$$\Pr[\mathsf{T}_{\mathrm{id}} = \tau'] = \frac{1}{2^n} \cdot \frac{1}{\mathbf{P}(2^n, q_p)} \cdot \prod_{i=1}^{r} \frac{1}{2^{nqb_{c_i}}}. \tag{24}$$

For computing the real interpolation probability, as τ' is good, all the inputs and outputs of π are distinct. The total number of π invocations including the primitive queries is $(b_{c_1} + \ldots + b_{c_r} + q_p)$. Therefore,

$$\Pr[\mathsf{T}_{\text{re}} = \tau'] = \frac{1}{2^n} \cdot \frac{1}{\mathbf{P}(2^n, b_{c_1} + \ldots + b_{c_r} + q_p)}. \tag{25}$$

It is trivial to see that the ratio of Eqn. (25) to Eqn. (24) is at least 1. Hence the result of Lemma 5 follows. Finally, by combining Lemma 4, Lemma 5 and Theorem 2, the result of Theorem 4 follows. □

6.4 ppHCTR : An Instantiation of ppTES with ppCTR and PolyHash

We instantiate the public permutation based length expanding PRF $\mathsf{F}_k^{\pi_2}$ of $\mathsf{ppTES}[\pi_1, \pi_2, k, \mathsf{H}]$ with $\mathsf{ppCTR}[\pi_2, k]$ and its underlying AXUAR hash function H_{k_h} with the PolyHash function Poly_{k_h}, as described in Eqn. (3), to realize a practical candidate of a public permutation based TES, referred to as $\mathsf{ppHCTR}[\pi_1, \pi_2, k, \mathsf{Poly}_{k_h}]$. We assume that the tweak is μ blocks long, i.e., $\mathtt{tw} = n\mu$ and thus, for any $i \in [q]$, the maximum degree of $\mathsf{Poly}_{k_h}(M_2^i \| \ldots \| M_{l_i}^i \| T^i)$ is $\hat{l}_i + \mu$, where $\hat{l}_i = \lceil \frac{|\mathbf{M}_{\mathbf{R}}^i|}{n} \rceil$. Since, $\hat{l}_i \leq \ell$ for all $i \in [q]$, where ℓ denotes the maximum number of message blocks among all q queries, therefore the AXU and the AR advantage of the PolyHash function is $(\ell + \mu)/2^n$. Note that, ppHCTR requires two independent n-bit random permutations π_1 and π_2, an n-bit random key K and an independent n-bit random hash key K_h for the PolyHash function. Security result of ppHCTR follows trivially from Theorem 3 and Theorem 4 which can be summarized as follows:

Theorem 5. *Let $\pi_1, \pi_2 \leftarrow_{\$} \mathsf{Perm}(n)$ be two independent n-bit public random permutations and let $K \leftarrow_{\$} \{0,1\}^n$ be an n-bit random key. Let $K_h \leftarrow_{\$} \{0,1\}^n$ be an n-bit random hash key of* PolyHash *function as described in Eqn. (3). Then, for any $(q_e, q_d, q_{p_1} + q_{p_2}, \ell, \sigma)$ information theoretic non-trivial adaptive* CCA *distinguisher* D *against the* tSPRP *security of* $\mathsf{ppHCTR}[\pi_1, \pi_2, K, \mathsf{Poly}_{K_h}]$, *we have*

$$\mathbf{Adv}_{\mathsf{ppHCTR}}^{\text{tSPRP}}(\mathsf{D}) \leq \frac{\sigma^2}{2^n} + \frac{2\sigma q_{p_2}}{2^n} + \frac{q^2\ell}{2^n} + \frac{2qq_{p_1}\ell}{2^n} + \frac{\mu q^2}{2^n} + \frac{2\mu q q_p}{2^n} + \frac{q^2}{2^{n+1}} + \frac{q(q-1)}{2^{n+1}},$$

where $q = q_e + q_d$, ℓ is the maximum number of message blocks and μ is the number of tweak blocks.

7 ppHCTR+ : A Single-Keyed Variant of ppHCTR

In the last section, we have seen that ppHCTR, a public permutation based TES, requires two independent n-bit public random permutations and two independent n-bit keys. In this section, we propose a single permutation and single keyed variant of ppHCTR, referred to as ppHCTR+. The construction is based on an n-bit public random permutation π and an n-bit random hash key of the PolyHash function as described in Eqn. (3). We consider that the tweak size is μ blocks long. The encryption and decryption algorithm of ppHCTR+ is shown in Fig. 7.1.

$\mathsf{ppHCTR{+}.Enc}^{\pi}_{k_h}(T, M)$	$\mathsf{ppHCTR{+}.Dec}^{\pi}_{k_h}(T, C)$
1. $(M_1\|\ldots\|M_l) \leftarrow \mathsf{parse}_n(M)$;	1. $(C_1\|\ldots\|C_l) \leftarrow \mathsf{parse}_n(C)$;
2. $\mathbf{M_L} \leftarrow M_1; \mathbf{M_R} \leftarrow (M_2\|\ldots\|M_l)$;	2. $\mathbf{C_L} \leftarrow C_1; \mathbf{C_R} \leftarrow (C_2\|\ldots\|C_l)$;
3. $U \leftarrow M_L \oplus \mathsf{Poly}_{k_h}(\mathbf{M_R}\|T)$;	3. $V \leftarrow C_1 \oplus \mathsf{Poly}_{k_h}(\mathbf{C_R}\|T)$;
4. $V \leftarrow \pi(U)$; $Z \leftarrow U \oplus V$;	4. $U \leftarrow \pi^{-1}(V)$; $Z \leftarrow U \oplus V$;
5. **for** $j = 1$ to $l-1$	5. **for** $j = 1$ to $l-1$
6. $\quad Z_j \leftarrow Z \oplus j$;	6. $\quad Z_j \leftarrow Z \oplus j$;
7. $\quad S_j \leftarrow \pi(Z_j) \oplus Z_j$;	7. $\quad S_j \leftarrow \pi(Z_j) \oplus Z_j$;
8. $\mathbf{S} \triangleq (S_1\|\ldots\|S_{l-1})$;	8. $\mathbf{S} \triangleq (S_1\|\ldots\|S_{l-1})$;
9. $\mathbf{C_R} \leftarrow \mathbf{M_R} \oplus \mathsf{first}(\|\mathbf{M_R}\|, \mathbf{S})$;	9. $\mathbf{M_R} \leftarrow \mathbf{C_R} \oplus \mathsf{first}(\|\mathbf{C_R}\|, \mathbf{S})$;
10. $C_L \leftarrow V \oplus \mathsf{Poly}_{k_h}(\mathbf{C_R}\|T)$;	10. $M_L \leftarrow V \oplus \mathsf{Poly}_{k_h}(\mathbf{M_R}\|T)$;
11. **return** $(\mathbf{C_L}\|\mathbf{C_R})$;	11. **return** $(\mathbf{M_L}\|\mathbf{M_R})$;

Fig. 7.1. ppHCTR+ based on an n-bit public random permutation π and an n-bit random hash key k_h. Left part is the encryption algorithm and right part is its decryption algorithm.

To see the dataflow of the encryption algorithm we consider an input message $M = (M_1\|M_2)$, where $|M_1| = |M_2| = n$, i.e., M consists of two full blocks. Therefore, in step (2) of the algorithm, the variable $\mathbf{M_L}$ is assigned to M_1 and $\mathbf{M_R}$ is assigned to M_2. In step (3) of the algorithm, we evaluate the poly hash Poly_{k_h} on $(M_2\|T)$ which results to $M_2 \cdot k_h^3 \oplus T \cdot k_h^2 \oplus \langle |M_2| + |T| \rangle \cdot k_h$ which is xored with the n-bit value M_1 to produce U. In step (4), we take the xor of U and $V = \pi(U)$ to produce Z. In step (6) and (7), we compute the key stream $\mathbf{S} = S_1$ where each $|S_1| = n$ by $S_1 = \pi(Z \oplus 1) \oplus (Z \oplus 1)$. Since, $|\mathbf{M_R}| = n$, $\mathbf{C_R}$ will be $M_2 \oplus S_1$, which becomes the input along with tweak T to the poly hash function Poly_{k_h}. Evaluation of the poly hash on input $\mathbf{C_R}\|T$ results to $\mathbf{C_R} \cdot k_h^3 \oplus T \cdot k_h^2 \oplus \langle |\mathbf{C_R}| + |T| \rangle \cdot k_h$. Then the result is xored with V to produce $\mathbf{C_L}$, which is returned along with $\mathbf{C_R}$ as the encryption of $M = M_1\|M_2$. The decryption works in a similar way.

7.1 Security Result of ppHCTR+

The security result of ppHCTR+ is as follows:

Theorem 6. *Let $\pi \leftarrow_{\$} \mathsf{Perm}(n)$ be an n-bit public random permutation and let $K_h \leftarrow_{\$} \{0,1\}^n$ be an n-bit random hash key of* PolyHash *function as described in Eqn.* (3)*. Then, for any $(q_e, q_d, q_p, \ell, \sigma)$ information theoretic non-trivial adaptive* CCA *distinguisher* D *against the* tSPRP *security of* $\mathsf{ppHCTR+}[\pi, \mathsf{Poly}_{K_h}]$*, we have*

$$\mathbf{Adv}^{\mathrm{tSPRP}}_{\mathsf{ppHCTR+}}(\mathsf{D}) \leq \frac{9\sigma^2}{2^n} + \frac{6\mu\sigma^2}{2^n} + \frac{4q_p\sigma(\mu+1)}{2^n} + \frac{q(q-1)}{2^{n+1}},$$

where σ is the total number of message blocks for all $q \stackrel{\Delta}{=} q_e + q_d$ queries and μ is the number of tweak blocks.

8 Proof of Theorem 6

In section 6.4, we propose ppHCTR, which uses two independent random permutations and two independent random keys, which allowed us to use the generic security result of ppTES in order to derive the security result of ppHCTR. However, for the single keyed variant of it, we cannot use the generic result of ppTES due to the input / output dependency and that demands an independent security proof for $\mathsf{ppHCTR+}$.

For the sake of simplicity, we refer $\mathsf{ppHCTR+}[\pi, \mathsf{Poly}_{K_h}]$ as $\mathsf{ppHCTR+}$ when the underlying primitives are assumed to be understood. By Theorem 1, we have

$$\mathbf{Adv}^{\mathrm{tSPRP}}_{\mathsf{ppHCTR+}}(\mathsf{D}) \leq \mathbf{Adv}^{\pm\mathrm{rnd}}_{\mathsf{ppHCTR+}}(\mathsf{D}) + \frac{q(q-1)}{2^{n+1}}, \tag{26}$$

where recall that n is the minimum message length allowed for $\mathsf{ppHCTR+}$. Therefore, we bound the $\pm$rnd advantage of $\mathsf{ppHCTR+}$. Let D be any information theoretic non-trivial adaptive deterministic CCA distinguisher with access to the oracles in either of the following two worlds: in the real world it interacts with $\mathcal{O}_{\mathrm{re}} = (\mathsf{ppHCTR+.Enc}^{\pi}_{K_h}, \mathsf{ppHCTR+.Dec}^{\pi}_{K_h}, \pi^{\pm})$ for an n-bit random hash key K_h and a random n-bit permutation π or in the ideal world it interacts with $\mathcal{O}_{\mathrm{id}} = (\$_0, \$_1, \pi^{\pm})$, where $\$_0$ and $\$_1$ are two independent random functions such that for any input, it responds with uniform values. Now, our goal is to upper bound the maximum advantage in distinguishing the real world from the ideal one.

Let D be the maximum distinguishing advantage achieving distinguisher that makes q_e (resp. q_d) encryption (resp. decryption) queries and q_p primitive queries. After the interaction is over, the underlying hash key is revealed to D and finally, D outputs a bit. Let $\tau \stackrel{\Delta}{=} \{(T^1, M^1, C^1), (T^2, M^2, C^2), \ldots, (T^q, M^q, C^q)\}$ be the list of construction queries and responses and $\tau_p \stackrel{\Delta}{=} \{(x_1, y_1), (x_2, y_2), \ldots, (x_{q_p}, y_{q_p})\}$ be the list of primitive queries and responses where each T^i is exactly μ blocks long. The triplet $\tau' = (\tau, \tau_p, K_h)$ constitutes the query transcript of the attack. Now, we characterize the set of bad transcripts and good transcripts.

8.1 Definition and Probability of Bad Transcripts

In this section, we define bad transcripts and bound their probabilities in the ideal world. The defining criterion of the bad event is any non-trivial collision in the input or output of the permutation. As defined in Fig. 7.1, $\mathbf{M^i_R}$ denotes $M^i_2\|\ldots\|M^i_{l_i}$ and $\mathbf{C^i_R}$ denotes $C^i_2\|\ldots\|C^i_{l_i}$. Moreover, for a transcript τ', we denote $U^i = \mathsf{Poly}_{K_h}(\mathbf{M^i_R}\|T^i) \oplus M^i_1, V^i = \mathsf{Poly}_{K_h}(\mathbf{C^i_R}\|T^i) \oplus C^i_1$ and $Z^i_\alpha = U^i \oplus V^i \oplus \langle\alpha\rangle$.

Definition 4. (**Bad Transcript for $\mathsf{ppHCTR+}$**) : *An attainable transcript $\tau' = (\tau, \tau_p, K_h)$ is called a* **bad** *transcript for* $\mathsf{ppHCTR+}$ *if any of the following conditions hold:*

- $\mathsf{B.1}$ *: $\exists\ i \neq j \in [q]$ such that, $U^i = U^j$.*
- $\mathsf{B.2}$ *: $\exists\ i, j \in [q]$ and $\alpha \in [l_j - 1]$ such that, $U^i = Z^j_\alpha$.*
- $\mathsf{B.3}$ *: $\exists\ i, j \in [q]$, $\alpha \in [l_i - 1]$ and $\beta \in [l_j - 1]$ with $(i, \alpha) \neq (j, \beta)$ such that $Z^i_\alpha = Z^j_\beta$, where $(i, \alpha) \neq (j, \beta)$.*
- $\mathsf{B.4}$ *: $\exists\ i \neq j \in [q]$ such that $V^i = V^j$.*
- $\mathsf{B.5}$ *: $\exists\ i, j \in [q]$ and $\alpha \in [l_j - 1]$ such that $V^i = Z^j_\alpha \oplus M^j_{\alpha+1} \oplus C^j_{\alpha+1}$.*
- $\mathsf{B.6}$ *: $\exists\ i, j \in [q]$, $\alpha \in [l_i - 1]$ and $\beta \in [l_j - 1]$ with $(i, \alpha) \neq (j, \beta)$ such that $Z^i_\alpha \oplus M^i_{\alpha+1} \oplus C^i_{\alpha+1} = Z^j_\beta \oplus M^j_{\beta+1} \oplus C^j_{\beta+1}$.*
- $\mathsf{B.7}$ *: $\exists\ i \in [q]$ and $j \in [q_p]$ such that $U^i = x_j$.*
- $\mathsf{B.8}$ *: $\exists\ i \in [q]$, $j \in [q_p]$ and $\alpha \in [l_i - 1]$ such that $Z^i_\alpha = x_j$.*
- $\mathsf{B.9}$ *: $\exists\ i \in [q]$ and $j \in [q_p]$ such that $V^i = y_j$.*
- $\mathsf{B.10}$ *: $\exists\ i \in [q]$, $j \in [q_p]$ and $\alpha \in [l_i - 1]$ such that $Z^i_\alpha \oplus M^i_{\alpha+1} \oplus C^i_{\alpha+1} = y_j$.*

Lemma 6. *Let T_{id} be the random variable that takes the transcript resulting from the interaction between the distinguisher and the ideal world and $\mathcal{V}_{\mathrm{b}}$ be the set of all attainable bad transcripts for* $\mathsf{ppHCTR+}$*. Then, by assuming $q \leq \sigma$, we have*

$$\Pr[\mathsf{T}_{\mathrm{id}} \in \mathcal{V}_{\mathrm{b}}] \leq \epsilon_{\mathrm{bad}} = \frac{9\sigma^2}{2^n} + \frac{6\mu\sigma^2}{2^n} + \frac{4q_p\sigma(\mu+1)}{2^n}.$$

Proof. By the union bound,

$$\Pr[\mathsf{T}_{\mathrm{id}} \in \mathcal{V}_{\mathrm{b}}] \leq \sum_{i=1}^{10} \Pr[\mathsf{B.i}]. \tag{27}$$

In the following, we bound the probability of all the bad events individually. The lemma will follow by adding the individual bounds.

Notation. We consider that the tweak is μ blocks long, i.e., $\mathtt{tw} = n\mu$. Therefore, for any $i \in [q]$, the maximum degree of $\mathsf{Poly}_{k_h}(\mathbf{M^i_R}\|T^i)$ is $\hat{l}_i + \mu$, where $\hat{l}_i \stackrel{\Delta}{=} \lceil \frac{|\mathbf{M^i_R}|}{n} \rceil$. Let $\hat{\ell}_{i,j}$ denotes $\max\{\hat{l}_i, \hat{l}_j\} + \mu$ and $\hat{\sigma} = q\mu + (\hat{l}_1 + \ldots + \hat{l}_q)$ denotes the total number of message blocks of $\mathbf{M}_{\mathbf{R}}^i$ (including the tweak blocks) across all q queries. Therefore, $\sigma = (\hat{\sigma} - q\mu + q)$ which implies that $\sigma - q = \hat{l}_1 + \ldots + \hat{l}_q$. Since, $\hat{\ell}_{i,j} \leq \hat{l}_i + \hat{l}_j + \mu$, we have

$$\sum_{1 \leq i < j \leq q} \hat{\ell}_{i,j} \leq \binom{q}{2}\mu + \sum_{1 \leq i < j \leq q} (\hat{l}_i + \hat{l}_j) \leq (q-1)\hat{\sigma} \leq q\sigma + \mu q^2. \tag{28}$$

Bounding $\mathsf{B.1}$. Bounding this event is equivalent to bounding

$$\mathsf{Poly}_{K_h}(\mathbf{M}_{\mathbf{R}}^i \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{M}_{\mathbf{R}}^j \| T^j) = M_1^i \oplus M_1^j.$$

If $\mathbf{M}_{\mathbf{R}}^i \| T^i = \mathbf{M}_{\mathbf{R}}^j \| T^j$ then the probability of this event is zero, otherwise it is bounded by the AXU advantage of the $\mathsf{PolyHash}$ and hence from Eqn. (28) and by assuming $q \leq \sigma$, we have

$$\Pr[\mathsf{B.1}] \leq \sum_{1 \leq i < j \leq q} \frac{\hat{\ell}_{i,j}}{2^n} \leq \frac{q\sigma + \mu q^2}{2^n} \leq \frac{\sigma^2(\mu+1)}{2^n}. \tag{29}$$

Bounding $\mathsf{B.2}$. To bound the probability of $\mathsf{B.2}$, we first fix the value of i, j and α. Note that $Z_\alpha^j = Z^j \oplus \langle \alpha \rangle$. Therefore, $U^i = Z_\alpha^j$ implies $U^i \oplus U^j \oplus V^j = \langle \alpha \rangle$. Now, this essentially implies the following hash equation:

$$\mathsf{Poly}_{K_h}(\mathbf{M}_{\mathbf{R}}^i \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{M}_{\mathbf{R}}^j \| T^j) \oplus \mathsf{Poly}_{K_h}(\mathbf{C}_{\mathbf{R}}^j \| T^j) = M_1^i \oplus M_1^j \oplus C_1^j \oplus \langle \alpha \rangle. \tag{30}$$

Based on the values of i and j, we have the following two subcases:

- **Case A:** If $i \neq j$, then we first assume that $i < j$. Then, if the j-th query is an encryption query, then C_1^j is random and therefore by conditioning on the hash key and using the randomness of C_1^j, probability of Eqn. (30) can be bounded by 2^{-n} as C_1^j is uniformly distributed in the ideal world. Similarly, if the j-th query is a decryption query, then M_1^j is random and therefore by conditioning on the hash key and using the randomness of M_1^j, probability of Eqn. (30) can be bounded by 2^{-n} as M_1^j is uniformly distributed in the ideal world. Therefore, by varying over possible choices of i and (j, α), we have

$$\Pr[\mathsf{B.2}] \leq \frac{q\sigma}{2^n}.$$

  On the other hand if $i > j$, then by conditioning all other random variables, we bound the probability of the event using the AXU advantage of the $\mathsf{PolyHash}$ function. Therefore, we have

$$\Pr[\mathsf{B.2}] \leq \sum_{1 \leq i < j \leq q} \frac{\hat{\ell}_{i,j}}{2^n} \leq \frac{q\sigma + \mu q^2}{2^n}.$$

  By considering the maximum of the above two, we have

$$\Pr[\mathsf{B.2}] \leq \frac{q\sigma + \mu q^2}{2^n}. \tag{31}$$

- **Case B:** If $i = j$, then, Eqn. (30) boils down to the following hash equation:

$$\mathsf{Poly}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) = C^i_1 \oplus \langle \alpha \rangle. \tag{32}$$

Note that for a fixed choice of i and α, Eqn. (32) can be bounded by the AR advantage of the $\mathsf{PolyHash}$ function. Therefore,

$$\Pr[\mathsf{B.2}] = \sum_{i=1}^{q} \sum_{\alpha=1}^{\hat{l}_i} \frac{\hat{l}_i + \mu}{2^n} = \frac{1}{2^n} \sum_{i=1}^{q} \hat{l}_i^2 + \frac{1}{2^n} \sum_{i=1}^{q} \hat{l}_i \mu \leq \frac{\sigma^2 + q^2}{2^n} + \frac{\mu\sigma}{2^n}. \tag{33}$$

By considering both the cases and by assuming $q \leq \sigma$, we have

$$\Pr[\mathsf{B.2}] \leq \frac{\sigma^2 + q^2 + \mu\sigma}{2^n} + \frac{q\sigma + \mu q^2}{2^n} \leq \frac{3\sigma^2(\mu + 1)}{2^n}. \tag{34}$$

Bounding $\mathsf{B.3}$. To bound the probability of $\mathsf{B.3}$, we first fix the value of i, j, α and β such that $(i, \alpha) \neq (j, \beta)$. Note that $Z^i_\alpha = Z^j_\beta$ implies the following hash equation:

$$\mathsf{Poly}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{M}^j_{\mathbf{R}} \| T^j) \oplus \mathsf{Poly}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{C}^j_{\mathbf{R}} \| T^j) = W,$$

where $W = M^i_1 \oplus M^j_1 \oplus C^i_1 \oplus C^j_1 \oplus \langle \alpha \rangle \oplus \langle \beta \rangle$. Note that for $i = j$, the probability of this event is zero. For $i \neq j$, without loss of generality we assume that $i < j$, if the j-th query is an encryption query, then C^j_1 is uniformly distributed in the ideal world which is used to bound the probability of the event by conditioning the hash key and all other random variables. Similarly, if the j-th query is a decryption query, then M^j_1 is uniformly distributed in the ideal world which is used to bound the probability of the event by conditioning the hash key and all other random variables. Combining the above two arguments with the assumption $q \leq \sigma$ and by varying over all possible choices of indices, we have

$$\Pr[\mathsf{B.3}] = \frac{\binom{\sigma - q}{2}}{2^n} \leq \frac{\sigma^2 + q^2}{2^{n+1}} \leq \frac{\sigma^2}{2^n}. \tag{35}$$

Bounding $\mathsf{B.4}$. Bounding this event is equivalent to bounding

$$\mathsf{Poly}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{C}^j_{\mathbf{R}} \| T^j) = C^i_1 \oplus C^j_1.$$

If $\mathbf{C}^i_{\mathbf{R}} \| T^i = \mathbf{C}^j_{\mathbf{R}} \| T^j$ then the probability of this event is zero, otherwise it is bounded by the AXU advantage of the $\mathsf{PolyHash}$ and hence from Eqn. (28) and by the assumpion $q \leq \sigma$, we have

$$\Pr[\mathsf{B.4}] \leq \sum_{1 \leq i < j \leq q} \frac{\hat{\ell}_{i,j}}{2^n} \leq \frac{q\sigma + \mu q^2}{2^n} \leq \frac{\sigma^2(\mu + 1)}{2^n}. \tag{36}$$

Bounding $\mathsf{B.5}$. We first fix the values of i, j and α and compute the probability of $V^i = M^j_{\alpha+1} \oplus C^j_{\alpha+1} \oplus Z^j_\alpha$. This event boils down to computing the probability of the following event: $\mathsf{Poly}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{M}^j_{\mathbf{R}} \| T^j) \oplus \mathsf{Poly}_{K_h}(\mathbf{C}^j_{\mathbf{R}} \| T^j) = W$, where $W = C^i_1 \oplus M^j_{\alpha+1} \oplus C^j_{\alpha+1} \oplus M^j_1 \oplus C^j_1 \oplus \langle \alpha \rangle$. Now, we have two subcases as follows:

- **Case A:** if $i = j$, then we have $\mathsf{Poly}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) = C^i_1 \oplus M^i_{\alpha+1} \oplus C^i_{\alpha+1} \oplus M^i_1 \oplus C^i_1 \oplus \langle \alpha \rangle$, which can be bounded using the AR advantage of the $\mathsf{PolyHash}$ function after conditioning all other random variables. Therefore, by assuming $q \leq \sigma$, we have

$$\Pr[\mathsf{B.5}] = \sum_{i=1}^{q} \sum_{\alpha=1}^{\hat{l}_i} \frac{\hat{l}_i + \mu}{2^n} = \frac{1}{2^n} \sum_{i=1}^{q} \hat{l}_i^2 + \frac{1}{2^n} \sum_{i=1}^{q} \hat{l}_i \mu \leq \frac{2\sigma^2}{2^n} + \frac{\mu\sigma}{2^n}. \tag{37}$$

- **Case B:** Now we consider the case when $i \neq j$ and without loss of generality we assume that $i < j$. Then by fixing the hash key K_h, the probability of the above event is the probability over the random draw of C^j_1 (if j-th query is an encryption query) or M^j_1 (if j-th query is a decryption query), which is at most 2^{-n}. Therefore, varying over all the possible choice of i, j and α and $q \leq \sigma$, we have

$$\Pr[\mathsf{B.5}] \leq \frac{q\sigma}{2^n} \leq \frac{\sigma^2}{2^n}. \tag{38}$$

Taking the maximum of Eqn. (37) and (38), we have

$$\Pr[\mathsf{B.5}] \leq \frac{2\sigma^2}{2^n} + \frac{\mu\sigma}{2^n}. \tag{39}$$

Bounding B.6. To bound this event we first fix i, j and α, β and then we compute the probability of $M^i_{\alpha+1} \oplus C^i_{\alpha+1} \oplus Z^i_\alpha = M^j_{\beta+1} \oplus C^j_{\beta+1} \oplus Z^j_\beta$. Now, we have the following subcases based on the values of i and j.

- **Case A:** If $i = j$, then the above event boils down to the following event $M^i_{\alpha+1} \oplus C^i_{\alpha+1} \oplus M^i_{\beta+1} \oplus C^i_{\beta+1} = \langle \alpha \rangle \oplus \langle \beta \rangle$. Since $\alpha \neq \beta$, without loss of generality we assume that $\alpha < \beta$. Therefore, using the randomness of C^i_β (if i-th query is encryption) or using the randomness of M^i_β (if i-th query is decryption), the probability of the event is bounded by 2^{-n}. By summing over all possible values of i, α and β, we have

$$\Pr[\mathsf{B.6}] \leq \sum_{i=1}^{q} \frac{\binom{\hat{l}_i}{2}}{2^n} \leq \frac{1}{2^{n+1}} (\sum_{i=1}^{q} \hat{l}_i)^2 = \frac{(\sigma - q)^2}{2^{n+1}} \leq \frac{\sigma^2 + q^2}{2^{n+1}}. \tag{40}$$

- **Case B:** If $i \neq j$, then we bound the probability of the event similar to that of $\mathsf{B.3}$, that is $1/2^n$ and therefore, by summing over all possible values of i, j, α and β, we have

$$\Pr[\mathsf{B.6}] \leq \frac{\sigma^2 + q^2}{2^{n+1}}. \tag{41}$$

By taking the maximum of Eqn. (40) and (41) and by assuming $q \leq \sigma$, we have

$$\Pr[\mathsf{B.6}] \leq \frac{\sigma^2 + q^2}{2^{n+1}} \leq \frac{\sigma^2}{2^n}. \tag{42}$$

Bounding $\mathsf{B.7}$**.** Bounding this event is equivalent to bounding $\mathsf{Poly}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) = M^i_1 \oplus x_j$. This event is bounded by the AR advantage of the $\mathsf{PolyHash}$ and hence from Eqn. (28) and by assuming $q \leq \sigma$, we have

$$\Pr[\mathsf{B.7}] \leq \sum_{i=1}^{q} \sum_{j=1}^{q_p} \frac{\hat{l}_i + \mu}{2^n} \leq \frac{(\sigma - q)q_p}{2^n} + \frac{\mu q q_p}{2^n} \leq \frac{q_p \sigma(\mu + 1)}{2^n}. \tag{43}$$

Bounding $\mathsf{B.8}$**.** To bound the probability of $\mathsf{B.8}$, we first fix the value of i, j and α. Note that $Z^i_\alpha = x_j$ implies the following hash equation: $\mathsf{Poly}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) = M^i_1 \oplus C^i_1 \oplus \langle \alpha \rangle \oplus x_j$. If the construction query comes after the primitive query then we can bound the probability of the event using the randomness of C^i_1 (if the construction query is an encryption query) or using the randomness of M^i_1 (if the construction query is a decryption query). Therefore, by conditioning the hash key and all other random variables, the bound will be 2^{-n}. Therefore, we have

$$\Pr[\mathsf{B.8}] = \frac{(\sigma - q)q_p}{2^n} \leq \frac{\sigma q_p}{2^n}.$$

On the other hand, if the primitive query comes after the construction query, then we condition every other random variables and bound the probability of this event by using the AR advantage of the $\mathsf{PolyHash}$ function. Therefore, we have

$$\mathsf{Pr}[\mathsf{B.8}] \leq \sum_{i=1}^{q} \sum_{j=1}^{q_p} \frac{\hat{l}_i + \mu}{2^n} \leq \frac{(\sigma - q)q_p}{2^n} + \frac{\mu q q_p}{2^n} \leq \frac{q_p(\sigma + q\mu)}{2^n}.$$

Therefore, by taking the maximum of the above two and by assuming $q \leq \sigma$, we have

$$\Pr[\mathsf{B.8}] \leq \frac{q_p \sigma(\mu + 1)}{2^n}. \tag{44}$$

Bounding $\mathsf{B.9}$**.** Bounding this event is equivalent to bounding $\mathsf{Poly}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) = C^i_1 \oplus y_j$. This event is bounded by the AR advantage of the $\mathsf{PolyHash}$ and hence from Eqn. (28) and by assuming $q \leq \sigma$, we have

$$\Pr[\mathsf{B.9}] \leq \sum_{i=1}^{q} \sum_{j=1}^{q_p} \frac{\hat{l}_i + \mu}{2^n} \leq \frac{(\sigma - q)q_p}{2^n} + \frac{\mu q q_p}{2^n} \leq \frac{q_p \sigma(\mu + 1)}{2^n}. \tag{45}$$

Bounding $\mathsf{B.10}$**.** To bound the probability of $\mathsf{B.10}$, we first fix the value of i, j and α. Note that $M^i_{\alpha+1} \oplus C^i_{\alpha+1} \oplus Z^i_\alpha = y_j$ implies the hash equation: $\mathsf{Poly}_{K_h}(\mathbf{M}^i_{\mathbf{R}} \| T^i) \oplus \mathsf{Poly}_{K_h}(\mathbf{C}^i_{\mathbf{R}} \| T^i) = W$, where $W = M^i_{\alpha+1} \oplus C^i_{\alpha+1} \oplus M^i_1 \oplus C^i_1 \oplus \langle \alpha \rangle \oplus y_j$. Similar to $\mathsf{B.8}$, we bound the event as

$$\Pr[\mathsf{B.10}] \leq \frac{q_p \sigma(\mu + 1)}{2^n}. \tag{46}$$

The proof follows from Eqn. (27), Eqn. (29)-Eqn. (46) and $q \leq \sigma$. $\square$

8.2 Analysis of Good Transcript

In this section, we show that for a good transcript $\tau' = (\tau, \tau_p, k_h)$, realizing τ' is almost as likely in the real world as in the ideal world.

Lemma 7. *Let $\tau' = (\tau, \tau_p, k_h)$ be a good transcript. Then*

$$\frac{\Pr[\mathsf{T}_{\mathrm{re}} = \tau']}{\Pr[\mathsf{T}_{\mathrm{id}} = \tau']} \geq 1.$$

Proof. Since, in the ideal world, the encryption and the decryption oracle behaves perfectly random, we have

$$\Pr[\mathsf{T}_{\mathrm{id}} = \tau'] = \frac{1}{|\mathcal{K}_h|} \frac{1}{\mathbf{P}(2^n, q_p)} \frac{1}{2^{n\sigma}}, \tag{47}$$

where σ is the total number of message blocks queried among all q queries.

Real Interpolation Probability. Since τ' is a good transcript, all the inputs and outputs of π are fresh as we have eliminated all the internal input and output collisions of π, including the primitive queries while defining the bad events. Since there are total $\sigma + q_p$ invocation of π, including the primitive queries, therefore, the required probability is,

$$\Pr[\mathsf{T}_{\mathrm{re}} = \tau'] = \frac{1}{|\mathcal{K}_h|} \frac{1}{\mathbf{P}(2^n, q_p)} \frac{1}{\mathbf{P}(2^n - q_p, \sigma)}. \tag{48}$$

By doing a simple algebraic calculation, it is easy to show that the ratio of Eqn. (48) to Eqn. (47) is at least 1. This proves Lemma 7. □

By combining Lemma 6, Lemma 7, Theorem 2 and Eqn. (26), the result of Theorem 6 follows. □

Discussion. We would like to note here that a simple birthday bound attack reveals the hash key of the Polyhash function for ppHCTR and ppHCTR+. This would allow an adversary to generate the ciphertext for any plaintext. The same attack also works for HCTR construction. A simple remedy of this problem is to introduce additional permutation calls after the hash evaluation in upper and bottom layers. This would resolve the problem of revealing the hash difference to any adversary, which in turn makes the recovery of the hash key difficult. A formal security analysis of this modified construction is beyond the scope of this paper.

9 Conclusion

Permutation based cryptography is a promising new addition in the cryptographic literature. There has been a continued effort in building cryptographic schemes using public permutations as the base primitive. Permutation based designs are generally lightweight. The overwhelming number of candidates using permutation based designs in the ongoing NIST competition of lightweight ciphers bears a proof of the fact that permutation based designs are preferred for computationally constrained scenarios.

There are permutation based designs available for various cryptographic schemes like authenticated encryption, authenticated encryption with associated data, message authentication codes, collision resistant hash etc., but to our knowledge there are no existing permutation based construction of tweakable enciphering schemes. Tweakable enciphering schemes are a class of encryption schemes which are length preserving and have thus found its use in low level disk encryption or encryption of any storage media which is organized as sectors. All the exisiting tweakable enciphering schemes are either build on top of block-ciphers, pesudorandom functions, or tweakable block ciphers [20, 38, 39, 51, 11, 34, 16]. In this paper, we study the security of tweakable enciphering schemes built on a low level primitive like public random permutation. We initiate the study with a generic construction of a public permutation based TES, called ppTES. Then we construct ppCTR, a public permutation based length expanding PRF and finally, we propose a single keyed and single permutation based TES which we call ppHCTR+. To the best of our knowledge, this is the first provably secure public permutation based TES.

Our constructions, both ppTES and ppHCTR+ requires both the forward and inverse calls of the permutation. Most existing public random permutations are more efficient in their forward calls compared to the inverse calls, thus a inverse free construction like [16, 11] is worth studying. Another direction of future research would be to construct a permutation based TES which is beyond birthday bound secure.

References

1. https://csrc.nist.gov/projects/hash-functions/sha-3-project.
2. Mohammad Abbadi, M Bellare, R Canetti, H Krawczyk, M Bellare, P Rogaway, D Wagner, J Black, P Rogaway, D Boneh, et al. Deterministic authenticated-encryption: A provable-security treatment of the keywrap problem. *Journal of Applied Sciences*, 8(21):pp–1, 1996.
3. Zhenzhen Bao, Avik Chakraborti, Nilanjan Datta, Jian Guo, Mridul Nandi, Thomas Peyrin, and Kan Yasuda. Photon-beetle authenticated encryption and hash family. *NIST LWC*, 2019.
4. Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. Gimli : A cross-platform permutation. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 299–320. Springer, 2017.
5. Guido Bertoni, Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Farfalle: parallel permutation-based cryptography. *IACR Trans. Symmetric Cryptol.*, 2017(4):1–38, 2017.
6. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge-based pseudo-random number generators. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA,*

August 17-20, 2010. Proceedings, volume 6225 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2010.

7. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions. In *ECRYPT hash workshop*, volume 2007. Citeseer, 2007.
8. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 313–314. Springer, 2013.
9. Tim Beyne, Yu Long Chen, Christoph Dobraunig, and Bart Mennink. Elephant. *NIST LWC*, 2019.
10. Srimanta Bhattacharya and Mridul Nandi. Revisiting variable output length XOR pseudorandom function. *IACR Cryptol. ePrint Arch.*, 2019:249, 2019.
11. Ritam Bhaumik and Mridul Nandi. An inverse-free single-keyed tweakable enciphering scheme. In Tetsu Iwata and Jung Hee Cheon, editors, *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*, volume 9453 of *Lecture Notes in Computer Science*, pages 159–180. Springer, 2015.
12. Ritam Bhaumik and Mridul Nandi. An inverse-free single-keyed tweakable enciphering scheme. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 159–180. Springer, 2015.
13. Andrey Bogdanov, Miroslav Knezevic, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede. spongent: A lightweight hash function. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*, pages 312–325. Springer, 2011.
14. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Francois-Xavier Standaert, John Steinberger, and Elmar Tischhauser. Key-alternating ciphers in a provable setting: Encryption using a small number of public permutations. In *Advances in Cryptology – EUROCRYPT 2012*, pages 45–62. Springer, 2012.
15. Bishwajit Chakraborty and Mridul Nandi. Orange. *NIST LWC*, 2019.
16. Debrup Chakraborty, Sebati Ghosh, Cuauhtemoc Mancillas-López, and Palash Sarkar. FAST: disk encryption and beyond. *IACR Cryptology ePrint Archive*, 2017:849, 2017.
17. Debrup Chakraborty, Vicente Hernandez-Jimenez, and Palash Sarkar. Another look at XCB. *Cryptography and Communications*, 7(4):439–468, 2015.
18. Debrup Chakraborty, Cuauhtemoc Mancillas-López, and Palash Sarkar. STES: A stream cipher based low cost scheme for securing stored data. *IACR Cryptology ePrint Archive*, 2013:347, 2013.
19. 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.
20. Debrup Chakraborty and Palash Sarkar. HCH: A New Tweakable Enciphering Scheme Using the Hash-Counter-Hash Approach. *IEEE Transactions on Information Theory*, 54(4):1683–1699, 2008.
21. Donghoon Chang, Nilanjan Datta, Avijit Dutta, Bart Mennink, Mridul Nandi, Somitra Sanadhya, and Ferdinand Sibleyras. Release of unverified plaintext: Tight unified model and application to ANYDAE. *IACR Trans. Symmetric Cryptol.*, 2019(4):119–146, 2019.
22. Donghoon Chang and Mridul Nandi. A short proof of the PRP/PRF switching lemma. *IACR Cryptol. ePrint Arch.*, 2008:78, 2008.
23. Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In *EUROCRYPT 2014. Proceedings*, pages 327–350, 2014.
24. Yu Long Chen, Eran Lambooij, and Bart Mennink. How to build pseudorandom functions from public random permutations. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, pages 266–293, 2019.

25. Benoit Cogliati and Yannick Seurin. On the provable security of the iterated even-mansour cipher against related-key and chosen-key attacks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 584–613. Springer, 2015.
26. Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Xoodyak, a lightweight cryptographic scheme. *NIST LWC*, 2019.
27. Yuanxi Dai, Yannick Seurin, John Steinberger, and Aishwarya Thiruvengadam. Indifferentiability of iterated even-mansour ciphers with non-idealized key-schedules: Five rounds are necessary and sufficient. In *Annual International Cryptology Conference*, pages 524–555. Springer, 2017.
28. Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Goutam Paul. Double-block hash-then-sum: a paradigm for constructing bbb secure prf. *IACR Transactions on Symmetric Cryptology*, pages 36–92, 2018.
29. Nilanjan Datta, Avijit Dutta, Mridul Nandi, Goutam Paul, and Liting Zhang. Single key variant of pmac_plus. *IACR Transactions on Symmetric Cryptology*, pages 268–305, 2017.
30. Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Kan Yasuda. Encrypt or decrypt? to make a single-key beyond birthday secure nonce-based mac. In *Annual International Cryptology Conference*, pages 631–661. Springer, 2018.
31. Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. Key recovery attacks on iterated even–mansour encryption schemes. *Journal of Cryptology*, 29(4):697–728, 2016.
32. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2. *NIST LWC*, 2019.
33. Avijit Dutta. Minimizing the two-round tweakable even-mansour cipher. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 601–629. Springer, 2020.
34. Avijit Dutta and Mridul Nandi. Tweakable HCTR: A BBB secure tweakable enciphering scheme. In *Progress in Cryptology - INDOCRYPT 2018 - 19th International Conference on Cryptology in India, New Delhi, India, December 9-12, 2018, Proceedings*, pages 47–69, 2018.
35. Avijit Dutta, Mridul Nandi, and Suprita Talnikar. Beyond birthday bound secure mac in faulty nonce model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 437–466. Springer, 2019.
36. Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *J. Cryptology*, 10(3):151–162, 1997.
37. Shai Halevi. EME*: Extending EME to handle arbitrary-length messages with associated data.
38. 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.
39. 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.
40. Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 15–44. Springer, 2015.
41. Manish Kumar. Security of XCB and HCTR. In *M.Tech.(Computer Science) Thesis*. Indian Statistical Institute, Kolkata, 2018.
42. Moses Liskov, Ronald L Rivest, and David Wagner. Tweakable block ciphers. In *Annual International Cryptology Conference*, pages 31–46. Springer, 2002.
43. Moses Liskov, Ronald L. Rivest, and David A. Wagner. Tweakable block ciphers. *J. Cryptology*, 24(3):588–613, 2011.

44. David A. McGrew and Scott R. Fluhrer. The Security of the Extended Codebook (XCB) Mode of Operation. In Carlisle M. Adams, Ali Miri, and Michael J. Wiener, editors, *Selected Areas in Cryptography*, volume 4876 of *Lecture Notes in Computer Science*, pages 311–327. Springer, 2007.
45. Bart Mennink, Reza Reyhanitabar, and Damian Vizár. Security of full-state keyed sponge and duplex: Applications to authenticated encryption. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 465–489. Springer, 2015.
46. Kazuhiko Minematsu. Beyond-birthday-bound security based on tweakable block cipher. In *International Workshop on Fast Software Encryption*, pages 308–326. Springer, 2009.
47. NIST. Online: https://csrc.nist.gov/projects/lightweight-cryptography.
48. Jacques Patarin. The "Coefficients H" Technique. In *Selected Areas in Cryptography, SAC*, pages 328–345, 2008.
49. Phillip Rogaway, Mihir Bellare, and John Black. Sha-3 standard. *ACM Transactions on Information and System Security (TISSEC)*, 6(3):365–403, 2003.
50. Palash Sarkar. Tweakable enciphering schemes from stream ciphers with IV. *IACR Cryptol. ePrint Arch.*, 2009:321, 2009.
51. 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.
52. Mark N Wegman and J Lawrence Carter. New classes and applications of hash functions. In *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*, pages 175–182. IEEE, 1979.