




Optimally Secure TBC Based Accordion Mode

Nilanjan Datta¹, Avijit Dutta^{1,2}, Shibam Ghosh³ and Hrithik Nandi^{1,4}

¹ Institute for Advancing Intelligence (IAI) TCG CREST, Kolkata, India

² Academy of Scientific and Innovative Research (AcSIR), Ghaziabad, India

³ Department of Computer Science, University of Haifa, Haifa, Israel

⁴ Ramakrishna Mission Vivekananda Educational and Research Institute, India

nilanjan.datta@tcgcrest.org, avirocks.dutta13@gmail.com, sghosh03@campus.haifa.ac.il, hrithik.nandi.85@tcgcrest.org

Abstract. The design of tweakable wide block ciphers has advanced significantly over the past two decades. This evolution began with the approach of designing a wide block cipher by Naor and Reingold. Since then, numerous tweakable wide block ciphers have been proposed, many of which build on existing block ciphers and are secure up to the birthday bound for the total number of blocks queried. Although there has been a slowdown in the development of tweakable wide block cipher modes in last couple of years, the latest NIST proposal for accordion modes has reignited interest and momentum in the design and analysis of these ciphers. Although new designs have emerged, their security often falls short of optimal (i.e., n -bit) security, where n is the output size of the primitive. In this direction, designing an efficient tweakable wide block cipher with n -bit security seems to be an interesting research problem. An optimally secure tweakable wide block cipher mode can easily be turned into a misuse-resistant RUP secure authenticated encryption scheme with optimal security. This paper proposes HCTR+, which turns an n -bit tweakable block cipher (TBC) with n -bit tweak into a variable input length tweakable block cipher. Unlike tweakable HCTR, HCTR+ ensures n -bit security regardless of tweak repetitions. We also propose two TBC-based almost-xor-universal hash functions, named PHASH+ and ZHASH+, and use them as the underlying hash functions in the HCTR+ construction to create two TBC-based n -bit secure tweakable wide block cipher modes, PHCTR+ and ZHCTR+. Experimental results show that both PHCTR+ and ZHCTR+ exhibit excellent software performance when their underlying TBC is instantiated with Deoxys-BC-128-128.

Keywords: Tweakable Wide Block Enciphering, Tweakable Block Cipher, TSPRP, Optimal Security, HCTR

1 Introduction

A Tweakable Enciphering Scheme (TES) is a function $\tilde{\mathcal{E}} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \rightarrow \mathcal{C}$ that maps a plaintext M into a ciphertext C under the control of a key K and a tweak T , denoted as $C = \tilde{\mathcal{E}}_K^T(M)$. The ciphertext must have the same length as the plaintext and there must be an inverse function $\tilde{\mathcal{D}}_K^T$ to $\tilde{\mathcal{E}}_K^T$. The security guarantee we require is that the function should behave like a strong tweakable pseudorandom permutation, which intuitively means that an oracle that maps (T, M) into $C = \tilde{\mathcal{E}}_K^T(M)$ and (T, C) into $M = \tilde{\mathcal{D}}_K^T(C)$ must be indistinguishable (when the key K is random and secret) from an oracle that realizes T -indexed family of random permutations and their inverses. A secure tweakable enciphering scheme (a.k.a tweakable wide block cipher) is a useful tool to solve the disk-sector encryption problem, where one stores at disk sector location T , the encryption $C = \tilde{\mathcal{E}}_K^T(M)$ of a message M .

Design of TES was informally started with the approach of Naor and Reingold [NR99b, NR99a, NR97] in designing wide block cipher. Their design approach was based on a paradigm called *hash-encipher-hash* that involves applying an invertible blockwise-universal hash function ¹ on the input followed by enciphering the result in ECB mode and then applying yet another invertible blockwise-universal hash-function. Although, they did not fully specify the mode of operation, but in [NR99b], they came closer to show how to make the invertible blockwise-universal hash function out of an xor-universal hash function.

1.1 Revisiting Tweakable Enciphering Schemes

Following [NR99b, NR99a, NR97], the field of designing tweakable enciphering schemes has gained a significant momentum. Over the last two decades, a number of designs on tweakable enciphering schemes have been proposed. The design landscape of tweakable enciphering schemes can be broadly categorized into two distinct classes: (i) Encrypt-Mix-Encrypt, and (ii) Hash-Encrypt-Hash.

1.1.1 Encrypt-Mix-Encrypt

In [HR03], Halevi and Rogaway have introduced the design paradigm of tweakable enciphering scheme, in which a simple mixing layer is sandwiched between two invertible encryption layers. As an instantiation of this generic design framework, Halevi and Rogaway [HR03] have proposed CMC (CBC-Mix-CBC) construction, where CBC mode of encryption is used in both the encryption layers with a simple linear mixing function in between of them. Due to the CBC structure, CMC is inherently a sequential construction. As a follow up work of CMC, Halevi and Rogaway [HR04] have subsequently proposed a parallel construction, called EME (ECB-Mix-ECB) that uses ECB mode of encryption in both the layers. The drawback of EME is that it works only for full-block messages. In a later work, Halevi [Hal04] extended this construction and proposed EME* that is capable of handling arbitrary length messages. In [Jiv14], Jivsov presented a variant of the EME construction, called WCFB and have shown that it works extremely well for commonly occurring plaintext and repeated operations on the same wide block. In [BN15], Bhaumik et al. have proposed FMix, a single-keyed inverse free tweakable enciphering scheme.

1.1.2 Hash-Encrypt-Hash

As the name suggests, this design paradigm of tweakable enciphering scheme invokes an encryption layer in between of two universal hash functions. This design framework was introduced by Naor and Reingold in [NR97, NR99a] to propose a wide block cipher from a fixed input block cipher. Their proposed construction uses an invertible ECB mode of encryption layer which is sandwiched between two invertible pairwise independent blockwise-universal hash functions. However, the description of the construction given in [NR97, NR99a] is at a top level and also the latter work [NR99b] does not fully specify a mode of operation. In [MF04], McGrew et al. have proposed the Hash-Counter-Hash type construction, called XCB, that instantiate the encryption layer of Hash-Encrypt-Hash paradigm with the counter mode encryption which is sandwiched between two almost-xor universal hash functions ². The advantage of using the counter mode encryption is to tackle the variable length messages easily. XCB requires five independent keys and two block cipher invocations (excluding the block cipher calls in counter mode encryption). Later, Wang et al. have proposed HCTR [WFW05] construction with a single block cipher

¹A hash function is said to be an ϵ -blockwise-universal hash function, if the collision probability of any two output blocks of the hash function applied on two distinct messages, is at most ϵ .

²An n -bit keyed hash function is said to be an ϵ -almost-xor-universal hash function, if for any two distinct messages M, M' and for any n -bit string Δ , the probability that xor of the hash value of two messages attains Δ , is at most ϵ .

call (excluding the block cipher calls in counter mode encryption) and two keys (one hash key and one block cipher key). It was shown that HCTR achieves a security bound of $\sigma^3/2^n$, where σ denotes the total number of message blocks queried to the construction. In [CS06], Chakraborty and Sarkar have proposed PEP, an instantiation of the Hash-Encrypt-Hash paradigm by sandwiching a ECB mode of operation in the encryption layer in between of two layers of polynomial hash functions. In the next year, Halevi proposed TET [Hal07], an efficient variant of the PEP construction. Later, Chakraborty et al. came up with HEH [CS08] that improves upon TET. Meanwhile, Chakraborty and Nandi [CN08a] have shown that HCTR achieves birthday bound security in the order of $\sigma^2/2^n$, assuming the underlying hash function to be almost-xor-universal. In [Kum18], Kumar observed that the underlying hash function of HCTR construction does not satisfy the almost-xor-universal property. In [CB18], Crowley and Biggers have proposed a tweakable enciphering scheme, called Adiantum, that closely follows the Hash-Counter-Hash design paradigm, where it uses an ϵ -almost-xor-universal n -bit keyed hash function, an n -bit block cipher and the variable input length pseudorandom function of HCTR is replaced by a IV-based stream cipher. In a recent work, Crowley et al. [CHB21] have proposed a single-keyed efficient variant of the HCTR construction, dubbed HCTR2, that uses an efficient almost-xor-universal hash function and allows more pre-computation for better performance. In [CGLS22], Chakraborty et al. have proposed a PRF based inverse free tweakable enciphering scheme, called FAST, and showed it achieves a security bound in the order of $\sigma^2/2^n$.

Amongst the above mentioned constructions, only CMC and EME* are block cipher based constructions with a light weight masking layer in between of two encryption layers, whereas the other two paradigms require the field multiplication (as a part of the hash function evaluation) along with the block cipher evaluation. Thus, the only significant cost for Encrypt-Mix-Encrypt type constructions are the block cipher calls, whereas for the other two paradigms the cost involved in both evaluating the block cipher calls and the finite field multiplications. A detailed comparison of the performance and efficiency of different tweakable enciphering schemes can be found in [Hal07, CS08, Sar07]. This comparison study along with [MCR07] suggests that HCTR is one of the most efficient candidates amongst all the above tweakable enciphering schemes.

1.1.3 Deck Based Construction

In [GDM22], Gunging et al. have proposed two tweakable wide block cipher modes from doubly-extendable cryptographic keyed (deck) functions and a keyed hash function: double-decker and docked-double-decker. Double-decker is a direct generalization of Farfalle wideblock cipher [BDH⁺17], and is a four-round Feistel network on two arbitrarily large branches, where the middle two rounds call deck functions and the first and last rounds call the blinded keyed hash function. On the other hand, Docked-double-decker is a variant of the double-decker construction, where the bulk of the input to the deck functions is moved to the keyed hash functions. The security advantage of both of the constructions are reduced to the pseudorandom function advantage of the underlying round function and the blinded keyed hashing distinguishing advantage. Dobraunig et al. [DMMT24] have instantiated the Docked-Double-Decker construction with AES and finite field multiplication to realize three concrete constructions, called ddd-AES, ddd-AES⁺, and bbb-ddd-AES. It has been shown that first two constructions are secured up to $\sigma^2/2^n$, where σ denotes the total number of blocks queried to the construction. On the other hand, bbb-ddd-AES is secured up to $2^{2n/3}$ queries provided the same tweak is not used too often.

It is to be noted that the security guarantee by most of the above mentioned constructions becomes vacuous after about $2^{n/2}$ blocks have been enciphered. Only a few constructions are there that achieve security beyond the birthday bound.

1.2 BBB Secure Tweakable Enciphering Schemes

In [MI11], Minematsu and Iwata have first proposed two beyond birthday bound secure constructions that achieve n -bit security. However, their proposed construction turns a fixed and small length block cipher to a large length block cipher, called wide block cipher mode. In [ST13], Shrimpton and Terashima have proposed TCT2 construction, that achieves $2n/3$ bit security. We would like to note that both the constructions [MI11,ST13] require two primitives, a block cipher and a universal hash function. In [DN18], Dutta and Nandi have proposed a tweakable variant of the HCTR construction, called THCTR, that uses tweakable block cipher as a primitive instead of a block cipher in the HCTR construction. It has been shown that THCTR achieves security in the order of $\mu q/2^n$, where μ represents the maximum number of tweak repetition in encryption and decryption queries. The security bound of THCTR indicates that the construction achieves n -bit tweakable sprp security, if tweaks are distinct in each query, and it degrades gracefully as the maximum number of repeated tweaks increases. In [BLN18], Bhaumik et al. have proposed a tweakable block cipher based wide block cipher mode, called ZCZ and showed that it achieves n -bit security. ZCZ follows the ZHash-Counter-ZHash structure, where ZHash is a $2n$ -bit tweakable block cipher based hash function used in ZMAC [IMPS17] construction. Moreover, ZCZ uses tweakable block ciphers, where tweak-size should be more than the block-size, and offers optimal properties in terms of both performance and security. It requires only $3\ell/2$ calls (which is optimal) to the primitive for processing ℓ -block messages and provides n -bit security for a primitive with an n -bit state and tweak size. However, this construction does not support arbitrary-length tweaks as input. Recently, Dobraunig et al. [DMMT24] have proposed a beyond birthday bound secure block cipher based tweakable enciphering scheme, called bbb-ddd-AES, which is a specific instantiation of Docked-Double-Decker construction with AES as its underlying block cipher. In particular, authors have shown that bbb-ddd-AES is secured up to $2^{2n/3}$ queries, provided the same tweak is not used too often.

1.3 Designing Accordion Modes

It is evident from the last discussion that during the last two decades, the symmetric-key community have proposed a considerable corpus of tweakable enciphering schemes. To standardize construction, recently NIST has initiated a call for standardizing length preserving tweakable variable-input-length strong pseudorandom permutation (equivalently tweakable enciphering scheme), that they call as *Accordion* mode. The term accordion signifies that the mode would act as a cipher, not only on a single block but on a range of input sizes and should support arbitrary length tweaks. It is desirable that a well-designed accordion mode could potentially provide significant advantages over most of the block cipher modes that NIST currently approves and specified in the SP 800-38 series. As reported in the call that an accordion mode should easily be extended to design authenticated encryption (with associated data) schemes, tweakable encryption schemes, deterministic authenticated encryption schemes etc. Moreover, an accordion mode should also provide multi-user security, beyond the birthday bound security, key and context commitment, key-dependent input security, and nonce-hiding security.

To cater the need of the NIST requirement on the design of accordion mode, researchers have started to submit various design proposals on accordion modes. Amongst the new designs, Lee [Lee24] presented a beyond birthday bound secure variant of the HCTR construction, called Double-block HCTR (DbHCTR). It uses a $2n$ -bit state in the HCTR construction, where a fixed-length ($2n$ -bit) beyond birthday bound secure SPRP construction CTET+ [CEL+21] is combined with the masked counter mode. The construction employs a $2n$ -bit hash function, which is essentially the concatenation of two independent n -bit polynomial hashes. It requires one block cipher call and four field multiplications

per block and supports arbitrary length tweaks. Duy et al. [DFUB24] have proposed two accordion modes based on Hash-Encrypt-Hash paradigm, called ACCOR-S and ACCOR-L. The authors have proved that ACCOR-S achieves a security bound in the order of $q^2 a^2 / 2^{129}$, whereas ACCOR-L achieves a security bound in the order of $q a \ell_{\text{ctr}} / 2^n$, where a denotes the total number of blocks (including message and tweak), and ℓ_{ctr} denotes the number of message blocks processed in the counter mode of encryption.³ Both of these two constructions require ℓ block cipher calls along with two universal hash function evaluations, where ℓ denotes the total number of message blocks. Naito et al. [NSS24] have proposed a wide block encryption mode, called FFF construction that achieves context committing security.

1.4 Accordion Mode with Optimal Security: Motivation

As mentioned in the NIST requirements, the accordion modes are supposed to work in high-end applications where security is the primary concern. As we typically work on 128-bit blocks, security beyond the birthday bound is highly desired. Till now, none of the tweakable enciphering schemes has achieved n -bit security, when n is the output size of the primitive⁴. An accordion mode that accepts variable-length tweaks and provides optimal security can be used to build cryptographic tools that can be used in different applications demanding stronger security. For example, one can design an optimally secure misuse-resistant authenticated encryption (MRAE) scheme using an accordion mode, where one considers the (nonce, associated data) pair as the tweak and encrypts the padded message (message padded with a 10^* block) using an accordion mode. If the underlying accordion mode provides optimal security (even with repeated tweaks), the resulting construction provides optimal MRAE security. We emphasize that having an accordion mode with optimal security for repeated tweaks is crucial because, in the verified decryption algorithm of an AE scheme, the nonce and associated data (i.e., the tweak in the underlying accordion mode) can be repeated any number of times. Moreover such an MRAE scheme, by definition, will provide RUP security. Thus, an optimally secure accordion mode is extremely useful in building AEAD constructions targeting applications in the depth-in-defence category. In addition, one can also trivially design a deterministic authenticated encryption (DAE) by setting the tweak as empty and encrypting a message padded with one 10^* block using the accordion mode. Such a DAE scheme can be used as a key-wrapping algorithm. Another possible application is to design optimally secure non-hiding AEAD schemes by considering the associated data as the tweak and encrypting a padded message (message followed by a 10^* block) containing the nonce. In addition, an AEAD from a carefully constructed accordion mode should achieve additional security such as key or context-committing security, multi-user security, and key-dependent input security. The above discussion motivates us to design an efficient accordion mode with n -bit security that should provide n -bit security even with tweak repetitions.

Therefore, to summarize, all the existing beyond the birthday bound secure tweakable enciphering schemes either fall off from full n -bit security or achieve security that degrades gracefully with tweak repetition.

The above summary immediately raises the question whether we can have a tweakable enciphering scheme that provides **full n -bit security even in the presence of arbitrary tweak repetition.**

³We would like to mention that authors proved the security bound of the construction assuming the underlying block cipher is related-key secure, which is a stronger proof model than the standard one.

⁴THCTR provides n -bit security under the constraint that all the tweaks must be distinct.

1.5 Our Contribution

This paper introduces HCTR+, a single-keyed, n -bit secure accordion mode derived from an n -bit tweakable block cipher with n -bit tweak. At a high level, the construction is a variant of the HCTR construction, where the underlying n -bit keyed almost-xor-universal hash functions are replaced by $2n$ -bit keyed almost-xor-universal hash function. The block cipher (a.k.a sprp) in the left-hand side of the HCTR construction is replaced by a four-round TBC-based Luby Rackoff construction, dubbed as TLR4, and the counter mode encryption is replaced by a tweakable counter mode encryption $\widetilde{\text{CTR}}$, where the underlying block cipher of the counter mode is replaced by tweakable block cipher, and the counter is fed in as a tweak in the TBC. $\widetilde{\text{CTR}}$ is almost identical to the CTRT construction [PS16] and the IVCTR construction [LN17a] except that the tweak space for the tweakable block cipher used in both the constructions (CTRT and IVCTR) contain an indicator $i \in \mathbb{N}$ used for domain separation. Note that, our proposed construction belongs to the Hash-Counter-Hash paradigm. The core idea to ensure n -bit security from Hash-Counter-Hash type design is to use an n -bit secure strong pseudorandom permutation on the left side of the construction, and an n -bit secure variable input length pseudorandom function on the right. The sprp guarantee in HCTR+ has been realized through the TLR4 construction and the n -bit variable input length pseudorandom function has been realized through $\widetilde{\text{CTR}}$ construction. To instantiate our mode, we propose two hash functions PHASH+ and ZHASH+ which can be seen as a simple variant of the PMAC2x [LN17b] and ZMAC [IMPS17] constructions respectively. We have shown that both the hash function achieves $O(2^{-2n})$ -almost-xor-universal security. Finally, we instantiate our construction HCTR+ using the above two hash functions, and Deoxys-BC-128-128 [JNPS16] as the underlying tweakable block cipher. The resulting two constructions, dubbed as PHCTR+ and ZHCTR+, demonstrate excellent software performance, as shown in Table 2. The results indicate that our constructions achieve roughly 2.5 times improvement in cycles per byte (CPB) over the ZCZ construction, the only existing n -bit secure sprp construction with no tweak. We compare our proposed constructions PHCTR+ and ZHCTR+ with the existing beyond the birthday bound secure tweakable enciphering schemes in terms of the number of primitive calls, security, number of operations per block, number of keys required and the presence of tweak in Table 1.

Table 1: Comparative Study of Beyond the Birthday Bound Secure Tweakable Enciphering Scheme. TBC denotes Tweakable Block Cipher, BC denotes Block Cipher, and FM denotes Field Multiplication. We use ℓ and τ to denote the message and tweak length respectively (in blocks). The \dagger symbol indicates that the security degrades gracefully with maximum repetition of tweaks. Security is mentioned in terms of the number of bits.

Construction	Primitive	Security	# Ops per Block	# of Keys	Tweak
LargeBlock [MI11]	TBC	n	1 TBC + 4 FM	$\ell + 2$	\times
TCT2 [ST13]	BC	$2n/3$	4 TBC	$\ell + \tau + 19$	\checkmark
THCTR [DN18]	TBC	$n \dagger$	1 TBC + 2 FM	3	\checkmark
bbb-ddd-AES [DMMT24]	BC	$2n/3 \dagger$	2 BC + 2 FM	3	\checkmark
Db-HCTR [Lee24]	BC	$2n/3$	1 BC + 4 FM	3	\checkmark
ZCZ [BLN18]	TBC	n	1.5 TBC	1	\times
ZHCTR+ [This Paper]	TBC	n	3 TBC	1	\checkmark
PHCTR+ [This Paper]	TBC	n	3 TBC	1	\checkmark

Effect of Arbitrary Tweak Repetition: As we mentioned earlier a tweakable enciphering scheme can be easily turned into an (n)AE scheme by incorporating the nonce and associated data into the tweak. Now, if a tweakable enciphering scheme achieves full n -bit security even in the arbitrary tweak repetition scenario, then it implies that the corresponding AE scheme achieves full n -bit misuse-resistant security. On the other hand, a few tweakable enciphering schemes achieve n -bit security in graceful setting [DN18, DMMT24], where the security linearly degrades with the tweak repetition. Such a scheme cannot be easily adapted to an AE scheme, because the security model of the tweakable enciphering scheme restricts the adversary to make a limited number of decryption queries with the same tweak. As a result, the adversary for the corresponding AE scheme also cannot repeat nonces in the decryption queries - a scenario which is not practical at all.

Discussion: NIST mandates that Accordion mode should build upon AES block ciphers, it has several reasons for that: (i) it will primarily be used in powerful processors and cloud environments, where AES hardware acceleration is advantageous; and (ii) the mode must provide significant parallelism for large input sizes, a feature supported by modern CPUs, which can execute multiple AES instructions in parallel. Although our proposals are based on tweakable block cipher (instead of a block cipher), we have instantiated the tweakable block cipher with Deoxys-128-128 in our two constructions ZHCTR+ and PHCTR+, which leverages the AES round functions. This ensures users benefit from the key advantages of AES, namely the ability to exploit AES instruction sets available in modern hardware and achieve substantial parallelism for large inputs, allowing modern CPUs to handle multiple AES operations simultaneously.

Organization: Sect. 2 is comprised of all notations, security definitions, and some useful results. In Sect. 3, we provide a formal specification of our design HCTR+, justifying its design rationale and presenting the main security result demonstrating that the construction achieves optimal security. We provide the proof of our security result in Sect. 4. We have instantiated our construction with PHASH+ and ZHASH+ and provide concrete security results for the resulting constructions PHCTR+ and ZHCTR+ in Sect. 5. Finally, in Sect. 6, we discuss the implementation of the PHCTR+ and ZHCTR+ and provide detailed software performance results that justify the efficiency of our proposed mode.

2 Preliminaries

NOTATION. For a set \mathcal{X} , $X \leftarrow \mathcal{X}$ denotes that X is sampled uniformly at random from \mathcal{X} . We write $X \leftarrow Y$ to denote that Y is assigned in variable X . We denote an empty set as \emptyset . We say two sets \mathcal{X} and \mathcal{Y} are disjoint if $\mathcal{X} \cap \mathcal{Y} = \emptyset$ and we denote their union as $\mathcal{X} \sqcup \mathcal{Y}$ (which we refer to as *disjoint union*). For any natural number n , $\{0, 1\}^n$ denotes the set of all bit strings of length n and $\{0, 1\}^*$ denotes the set of all binary strings of arbitrary finite length. Sometimes, we call an element of $\{0, 1\}^n$ a *block*.

For $X, Y \in \{0, 1\}^n$, we write $X \oplus Y$ to denote the bitwise xor of X and Y . For any element $X \in \{0, 1\}^*$, we write $|X|$ to denote the number of bits of X and for any two $X, Y \in \{0, 1\}^*$, we write $X\|Y$ to denote the concatenation of X followed by Y . We also represent the concatenation of two arbitrary strings X, Y as (X, Y) . For any $X \in \{0, 1\}^*$, we parse X as $X = X_1\|X_2\|\dots\|X_\ell$, where each $|X_i| = n$ for $1 \leq i \leq \ell - 1$ and $1 \leq |X_\ell| \leq n$. We denote it as $(X_1, X_2, \dots, X_\ell) \stackrel{n}{\leftarrow} \text{Parse}(X)$. For a fixed natural number n , we define an injective function $\text{Pad}_n : \{0, 1\}^* \rightarrow \{0, 1\}^*$ as follows:

$$\text{Pad}_n(X) = \begin{cases} X; & \text{if } |X| \text{ is a multiple of } n \\ X\|0^{n-(|X| \bmod n)}; & \text{otherwise.} \end{cases}$$

For any binary string $X \in \{0, 1\}^*$ such that $|X| \geq n$, we write $(X_1, X_2) \stackrel{n}{\leftarrow} X$ to denote that X_1 is the most significant n -bit string of X and X_2 is the remaining string of X . For a sequence of elements $X^1, X^2, \dots, X^s \in \{0, 1\}^*$, we write X_a^i to denote the a -th block of the i -th element X^i .

For any natural number q , $[q]$ denotes the set $\{1, \dots, q\}$. For integers $1 \leq b \leq a$, $(a)_b$ denotes $a(a-1) \dots (a-b+1)$, where $(a)_0 = 1$ by convention. For any function $\Phi : \mathcal{X} \rightarrow \{0, 1\}^{2n}$, we define Φ^1 and Φ^2 be two functions from \mathcal{X} to $\{0, 1\}^n$ such that for all $x \in \mathcal{X}$, $\Phi^1(x)$ denotes the leftmost n bits of the $2n$ bit output of $\Phi(x)$ and $\Phi^2(x)$ denotes the rightmost n bits of the $2n$ bit output of $\Phi(x)$. For a positive non-zero integer i , $\langle i \rangle_n$ denotes the n -bit binary representation of integer i . For any natural number i , we use the notation $\text{Msb}_i(S)$ and $\text{Lsb}_i(S)$ to denote the most significant and the least significant i bits of a binary string S respectively, such that $|S| \geq i$.

2.1 Tweakable Block Cipher

A *tweakable block cipher* (TBC) is a mapping $\tilde{\text{E}} : \mathcal{K} \times \mathcal{T} \times \mathcal{X} \rightarrow \mathcal{X}$, where \mathcal{K} is called the key space, \mathcal{T} is called the tweak space and \mathcal{X} is the input space, such that for all keys $K \in \mathcal{K}$ and for all tweaks $T \in \mathcal{T}$, $X \mapsto \tilde{\text{E}}(K, T, X)$ is a permutation over \mathcal{X} . We denote with $\text{TBC}(\mathcal{K}, \mathcal{T}, \mathcal{X})$ the set of all tweakable block ciphers with key space \mathcal{K} , tweak space \mathcal{T} , and message space \mathcal{X} .

A *tweakable permutation* with tweak space \mathcal{T} and message space \mathcal{X} , is a mapping $\tilde{\text{P}} : \mathcal{T} \times \mathcal{X} \rightarrow \mathcal{X}$ such that for all tweaks $T \in \mathcal{T}$, $X \mapsto \tilde{\text{P}}(T, X)$ is a permutation over \mathcal{X} . We write $\text{TP}(\mathcal{T}, \mathcal{X})$ to denote the set of all tweakable permutations with tweak space \mathcal{T} and message space \mathcal{X} . We often simulate a tweakable random permutation $\tilde{\text{P}}$ with the help of a table $\Pi[\cdot, \cdot]$ which is initialized as empty. For each tweak $T \in \mathcal{T}$, we define the set $\text{Rng}(\Pi[T, \cdot])$ as $\{Y : \exists X, \Pi[T, X] = Y\}$. Similarly, for each tweak $T \in \mathcal{T}$, we define the set $\text{Dom}(\Pi[T, \cdot])$ as $\{X : \exists Y, \Pi[T, X] = Y\}$.

A tweakable block cipher is required to satisfy that for a uniformly sampled key K , the tweakable block cipher be indistinguishable from a tweakable permutation chosen at random from $\text{TP}(\mathcal{T}, \mathcal{X})$. Sometimes we consider the tweak space \mathcal{T} as $\mathcal{T} = \mathcal{D} \times \mathcal{T}'$ where \mathcal{D} is a finite set of natural numbers, called *domain-separators* which are encoded in d bit string.

Definition 1 (STPRP security). Let $\tilde{\text{E}} : \mathcal{K} \times \mathcal{T} \times \mathcal{X} \rightarrow \mathcal{X}$ be a tweakable block cipher and let \mathcal{A} be a deterministic, adaptive adversary. The advantage of \mathcal{A} in breaking the *strong tweakable pseudorandom permutation* security of $\tilde{\text{E}}$ is defined as

$$\text{Adv}_{\tilde{\text{E}}}^{\text{STPRP}}(\mathcal{A}) \stackrel{\text{def}}{=} \left| \Pr[K \leftarrow_{\$} \mathcal{K}, \mathcal{A}^{\tilde{\text{E}}_K(\cdot, \cdot), \tilde{\text{E}}_K^{-1}(\cdot, \cdot)} = 1] - \Pr[\tilde{\text{P}} \leftarrow_{\$} \text{TP}(\mathcal{T}, \mathcal{X}), \mathcal{A}^{\tilde{\text{P}}(\cdot, \cdot), \tilde{\text{P}}^{-1}(\cdot, \cdot)} = 1] \right|.$$

We say that $\tilde{\text{E}}$ is $(q, \mathbf{t}, \epsilon)$ -secure if the maximum strong tweakable pseudorandom permutation advantage of $\tilde{\text{E}}$ is ϵ where the maximum is taken over all distinguishers \mathcal{A} that makes q queries to its oracle and runs time at most \mathbf{t} . For a tweakable block cipher, one typically consider \mathcal{X} to be $\{0, 1\}^n$, the set of all n -bit binary strings. This paper considers the tweak space \mathcal{T} and the input space \mathcal{X} to be the set of all n -bit binary strings.

TWEAKABLE ENCRYPTING SCHEME: If the tweak size and the input size of $\tilde{\text{E}}$ is of variable length, then we call $\tilde{\text{E}}$ to be a length preserving tweakable enciphering scheme. In that case, the resources of the adversary \mathcal{A} are not only limited to the number of queries but also depend on the maximum message length queried to the cipher. By an “ $(q, \ell_{\max}, \mathbf{t})$ chosen-plaintext chosen-ciphertext adversary \mathcal{A} against the stprp security of some tweakable enciphering scheme \mathcal{C} ”, we mean that \mathcal{A} makes q queries such that the maximum message length of each query is ℓ_{\max} with maximum running time is at most \mathbf{t} .

2.2 Almost XOR Universal Hash Function

Let $H : \mathcal{K}_h \times \mathcal{X} \rightarrow \{0, 1\}^n$ be an n -bit keyed hash function. We call H to be an ϵ -almost-xor universal keyed hash function, if for every $X \neq X'$, and for every $Y \in \{0, 1\}^n$, we have

$$\Pr[K_h \leftarrow \mathcal{K}_h : H_{K_h}(X) \oplus H_{K_h}(X') = Y] \leq \epsilon.$$

2.3 H-Coefficients Technique

H-Coefficient Technique, introduced by Patarin [Pat08], provides a “systematic” way to upper bound the statistical distance between the answers of two interactive systems and is typically used to prove the information-theoretic pseudo randomness of constructions. Let \mathcal{A} be a computationally unbounded deterministic distinguisher that interacts with 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 and responses that \mathcal{A} made and received to and from the oracle, is called the *transcript* of \mathcal{A} , denoted as τ . Sometimes, we allow the oracle to release more internal information to \mathcal{A} only after \mathcal{A} completes all its queries and responses, but before it outputs its decision bit. In this case, the transcript of \mathcal{A} includes the additional information about the oracle and clearly the maximum distinguishing advantage of \mathcal{A} in this setting can not be less than that of without additional information. Observe that the transcript τ is a random variable and the randomness of the distribution of τ only comes from the randomness of the oracle with which \mathcal{A} interacts.

Let X_{re} denote the random variable that takes a transcript τ realized in the real world. Similarly, X_{id} denotes the random variable that takes a transcript τ realized in the ideal world. The probability of realizing a transcript τ in the ideal (resp. real) world is called *ideal (resp. real) interpolation probability*. A transcript τ is said to be attainable with respect to a distinguisher \mathcal{D} , if its ideal interpolation probability is non-zero. Let Θ denote the set of all attainable transcripts. Following these notations, we state the main theorem of the H-Coefficient technique as follows:

Theorem 1 (H-Coefficient). *Let \mathcal{A} be a fixed deterministic distinguisher that has access to either the real oracle \mathcal{O}_{re} or the ideal oracle \mathcal{O}_{id} . Let $\Theta = \Theta_{\text{good}} \sqcup \Theta_{\text{bad}}$ be some partition of the set of attainable transcripts Θ . For any good attainable transcript $\tau \in \Theta_{\text{good}}$, let*

$$\frac{\Pr[X_{\text{re}} = \tau]}{\Pr[X_{\text{id}} = \tau]} \geq 1 - \epsilon_{\text{good}},$$

for some $\epsilon_{\text{good}} \geq 0$, and there exists $\epsilon_{\text{bad}} \geq 0$ such that $\Pr[X_{\text{id}} \in \Theta_{\text{bad}}] \leq \epsilon_{\text{bad}}$ holds. Then,

$$\text{Adv}_{\mathcal{O}_{\text{re}}}^{\mathcal{O}_{\text{id}}}(\mathcal{A}) := |\Pr[\mathcal{A}^{\mathcal{O}_{\text{re}}} = 1] - \Pr[\mathcal{A}^{\mathcal{O}_{\text{id}}} = 1]| \leq \epsilon_{\text{good}} + \epsilon_{\text{bad}}.$$

3 Specification and Security Result of HCTR+

In [WFW05], Wang et al. have proposed HCTR, which is a mode of operation that turns an n -bit strong pseudorandom permutation into a tweakable strong pseudorandom permutation, supporting arbitrary and variable length input and tweak which is no less than n bits. For any message $M \in \{0, 1\}^*$ and a tweak $T \in \{0, 1\}^*$ such that $|M| \geq n$, HCTR works as follows: it first parses the message M into two parts M_1 and M_R , where M_1 is the first n bits of M and M_R is the remaining bit string of M . Then, it applies an n -bit keyed hash function H on the string $M_R || T$ and xor its n -bit output value with the first message block M_1 to produce an intermediate value X which is encrypted through an n -bit block cipher E . The output of the block cipher Y is xor-ed with X to produce an initialization value IV . This initialization value which acts as a counter in the block cipher

based counter mode encryption to produce $n\ell$ bits keystream, where ℓ is the number of message blocks of M_R . Then the first $|M_R|$ bits of $n\ell$ bits keystream is blinded with the corresponding message M_R to produce a ciphertext C_R of length $|M_R|$. Finally, the same keyed hash function H is applied on $C_R\|T$ and xor its output with Y to produce the first ciphertext block C_1 . Wang et al. [WFW05] have shown that HCTR achieves $\sigma^3/2^n$ security bound, where σ denotes the total number of message blocks queried to the construction across all encryption and decryption queries. Later in [CN08a], Chakraborty and Nandi have improved its security bound from $\sigma^3/2^n$ to $\sigma^2/2^n$.

3.1 Specification of HCTR+

HCTR+ is a single-keyed tweakable and double-block variant of the HCTR construction. For any message $M \in \{0,1\}^*$ and a tweak $T \in \{0,1\}^*$ such that $|M| \geq 2n$, HCTR+ works as follows: it first derives six tweakable block cipher keys K_1, \dots, K_6 as shown below:

$$K_i \leftarrow \tilde{E}_K(10^{n-1}, \langle i \rangle_n), 1 \leq i \leq 6.$$

Then, it parses the message M into two parts $M_1\|M_2$ and M_R , where $M_1\|M_2$ is the first $2n$ bits of M and M_R is the remaining bit string of M . Then, it applies an $2n$ -bit keyed hash function H on the string $M_R\|T$ and xor its $2n$ -bit output value with the first two message blocks $M_1\|M_2$ to produce a $2n$ -bit intermediate value (U_1, U_2) . Then, we feed this pair of n -bit strings (U_1, U_2) to the four round TBC based Luby Rackoff construction, called TLR4⁵. Let the output of TLR4 be (V_1, V_2) . We tap two intermediate n -bit values of the construction Z and W , as shown in Fig. 2, and use them in a TBC based counter mode of encryption. In particular, we use Z as a tweak in the underlying TBC of the counter mode and W as an input value to the counter mode. We would like to mention that every time a message is processed, its tweaks are incremented at every call of the TBC, however, the input value of the TBC W remains fixed throughout the message processing part. Let ℓ denotes the number of message blocks of the message M . Now the output of the TBC based counter mode encryption is an $n(\ell - 2)$ -bit string. The most significant $|M_R|$ bits of $n(\ell - 2)$ bits keystream is masked with M_R to produce the ciphertext C_R of length $|M_R|$. Finally, the same $2n$ -bit keyed hash function is applied on $C_R\|T$ and xor its output with (V_1, V_2) to produce $2n$ -bit ciphertext $C_1\|C_2$. A schematic diagram of the construction is shown in Fig. 2.

As can be seen from the algorithm in Fig. 1, there are primarily three building blocks used in the construction; a $2n$ -bit keyed almost-xor-universal hash function H , an n -bit tweakable block cipher with n -bit tweak and a tweakable block cipher based counter mode encryption. Given an n -bit binary string Z , we define a sequence $\tilde{Z} = (Z \oplus \langle 1 \rangle_n, \dots, Z \oplus \langle \ell - 2 \rangle_n)$, where $\ell = \lceil |M|/n \rceil$. Given such a sequence \tilde{Z} , a key K , and an n -bit input W , we define the tweakable block cipher based counter mode encryption as follows:

$$\widetilde{\text{CTR}}[\tilde{E}_K](\tilde{Z}, W, l) \stackrel{\text{def}}{=} \parallel_{i=1}^{\ell-2} \left(\tilde{E}_K(Z \oplus \langle i \rangle_n, W) \right).$$

3.2 Security Result of HCTR+

In this section, we state the security result of HCTR+. In specific, we state that if \tilde{E} is an (n, n) tweakable block cipher, $H = (H[1], H[2])$ is an ϵ -axu $2n$ -bit keyed hash function such that each $H[i]$ is an ϵ_i -axu n -bit keyed hash function, then HCTR+ is a secure tweakable enciphering scheme against all chosen plaintext and chosen ciphertext adaptive adversaries that make roughly 2^n many encryption and decryption queries. Formally, the following result bounds the tweakable sprp advantage of HCTR+.

⁵Three round TBC based Luby Rackoff construction has been proposed by Coron et al. [CDMS10] and shown its SPRP security up to 2^n queries, where n denotes the output length of the TBC.

HCTR+.Enc(K, T, M)	HCTR+.Dec(K, T, C)
1. $((M_1 \ M_2), M_R) \xleftarrow{2n} M$;	1. $((C_1 \ C_2), C_R) \xleftarrow{2n} C$;
2. for $i = 1$ to 6	2. for $i = 1$ to 6
3. $K_i \leftarrow \tilde{\mathbf{E}}_K(10^{n-1}, \langle i \rangle_n)$;	3. $K_i \leftarrow \tilde{\mathbf{E}}_K(10^{n-1}, \langle i \rangle_n)$;
4. $U_1 \ U_2 \leftarrow (M_1 \ M_2) \oplus \mathbf{H}_{K_6}(M_R \ T)$;	4. $V_1 \ V_2 \leftarrow (C_1 \ C_2) \oplus \mathbf{H}_{K_6}(C_R \ T)$;
5. $Z \leftarrow \tilde{\mathbf{E}}_{K_1}(U_2, U_1)$; $W \leftarrow \tilde{\mathbf{E}}_{K_2}(Z, U_2)$;	5. $W \leftarrow \tilde{\mathbf{E}}_{K_4}^{-1}(V_1, V_2)$; $Z \leftarrow \tilde{\mathbf{E}}_{K_3}^{-1}(W, V_1)$;
6. $V_1 \leftarrow \tilde{\mathbf{E}}_{K_3}(W, Z)$; $V_2 \leftarrow \tilde{\mathbf{E}}_{K_4}(V_1, W)$;	6. $U_2 \leftarrow \tilde{\mathbf{E}}_{K_2}^{-1}(Z, W)$; $U_1 \leftarrow \tilde{\mathbf{E}}_{K_1}^{-1}(U_2, Z)$;
7. for $i = 1$ to $\lceil M_R /n \rceil$	7. for $i = 1$ to $\lceil C_R /n \rceil$
8. $S_i \leftarrow \tilde{\mathbf{E}}_{K_5}(Z \oplus \langle i \rangle_n, W)$;	8. $S_i \leftarrow \tilde{\mathbf{E}}_{K_5}(Z \oplus \langle i \rangle_n, W)$;
9. $S \leftarrow S_1 \ \dots \ S_{\lceil M_R /n \rceil}$;	9. $S \leftarrow S_1 \ \dots \ S_{\lceil C_R /n \rceil}$;
10. $C_R \leftarrow \text{Msb}_{ M_R }(S) \oplus M_R$;	10. $M_R \leftarrow \text{Msb}_{ C_R }(S) \oplus C_R$;
11. $(C_1, C_2) \leftarrow (V_1, V_2) \oplus \mathbf{H}_{K_6}(C_R \ T)$;	11. $(M_1, M_2) \leftarrow (U_1, U_2) \oplus \mathbf{H}_{K_6}(M_R \ T)$;
12. return $(C_1 \ C_2 \ C_R)$;	12. return $(M_1 \ M_2 \ M_R)$;

Figure 1: HCTR+ construction based on an n -bit tweakable block cipher $\tilde{\mathbf{E}}$ with n -bit tweak and an $2n$ -bit keyed hash function \mathbf{H} . (Left): Encryption algorithm of HCTR+ and (Right): Decryption algorithm of HCTR+

Theorem 2. *Let \mathcal{K} be a non-empty finite set. Let $\tilde{\mathbf{E}} : \mathcal{K} \times \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be an (n, n) tweakable block cipher. Let $\mathbf{H} : \{0, 1\}^n \times \{0, 1\}^* \rightarrow \{0, 1\}^{2n}$ be a $2n$ -bit keyed hash function such that \mathbf{H} is an ϵ -almost-xor-universal $2n$ -bit keyed hash function and each $\mathbf{H}[i]$, where $\mathbf{H} = (\mathbf{H}[1], \mathbf{H}[2])$, is an ϵ_i -almost-xor universal n -bit keyed hash function. Then, for any $(q, \ell_{\max}, \mathfrak{t})$ -chosen plaintext chosen ciphertext adaptive adversary \mathcal{A} against the stprp security of $\text{HCTR}^+[\tilde{\mathbf{E}}, \mathbf{H}]$ such that each query is of length at least $2n$ bits, there exists a (σ, \mathfrak{t}') -chosen plaintext chosen ciphertext adversary \mathcal{A}' against the stprp security of $\tilde{\mathbf{E}}$, such that*

$$\text{Adv}_{\text{HCTR}^+[\tilde{\mathbf{E}}, \mathbf{H}]}^{\text{STPRP}}(\mathcal{A}) \leq 6\text{Adv}_{\tilde{\mathbf{E}}}^{\text{STPRP}}(\mathcal{A}') + \frac{q^2 \epsilon_1}{2^{n+1}} + \frac{q^2 \epsilon_2}{2^{n+1}} + q^2 \epsilon + \frac{2q^2 \ell_{\max}}{2^{2n}} + \frac{2q^2}{2^{2n}} + \frac{15}{2^n},$$

where σ is the total number of message blocks queried, and $\mathfrak{t}' = O(\mathfrak{t} + \sigma + qt_H)$, t_H denotes the time for computing the hash function \mathbf{H} .

3.3 Design Rationale of HCTR+

The motivation of our construction is to ensure that we should achieve n -bit tweakable sprp security and unlike [DN18], the security bound should not depend on repetition of the tweak. In order to do this, we need an n -bit secure strong pseudorandom permutation on the left side of the construction and an n -bit secure variable output length pseudorandom function on the right side of the construction. We use four round TBC based Luby Rackoff construction TLR4 on the left side of the construction and a TBC based counter mode of encryption CTR on the right side of the construction.

It has been shown [CDMS10] that three round TBC based Luby Rackoff construction TLR3 is n -bit secure sprp. Although, TLR3 ensures optimal security, we have observed that it does not suffice in our context as we need to generate $2n$ bit entropy from the left side of the construction, which will be used in the $\widetilde{\text{CTRT}}$ construction. Otherwise, we would have ended up with security that depends on the tweak repetition. Hence, it lead us to increase one more round of TLR3. Since, the input size of TLR4 is $2n$ bits, we use $2n$ bit keyed hash function that hashes the rest of the message M_R along with the tweak T . The $2n$ bit output of the hash is blinded with the first two message blocks (M_1, M_2) to generate the input of TLR4. Note that, we are not externally providing tweaks to the underlying TBC of the TLR4 construction. Instead, the intermediate n -bit states of the construction are served as tweak. Moreover, we tap two such intermediate n -bit states Z and W that is fed to the $\widetilde{\text{CTRT}}$ construction, where $(Z \oplus \langle 1 \rangle_n, Z \oplus \langle 2 \rangle_n, \dots, Z \oplus \langle \ell - 2 \rangle_n)$ are used as tweak for the underlying TBC of $\widetilde{\text{CTRT}}$ and W is used as the input of the TBC. We would like to emphasize that in $\widetilde{\text{CTRT}}$ mode, the counter has been used in the tweak part of the underlying TBC instead of incrementing the input value W at every call of the TBC. This role swap is necessary due to the single query distinguishing attack on the THCTR construction demonstrated in two independent works by Andreeva et al. [ABPV21] and by Khairallah [Kha24].

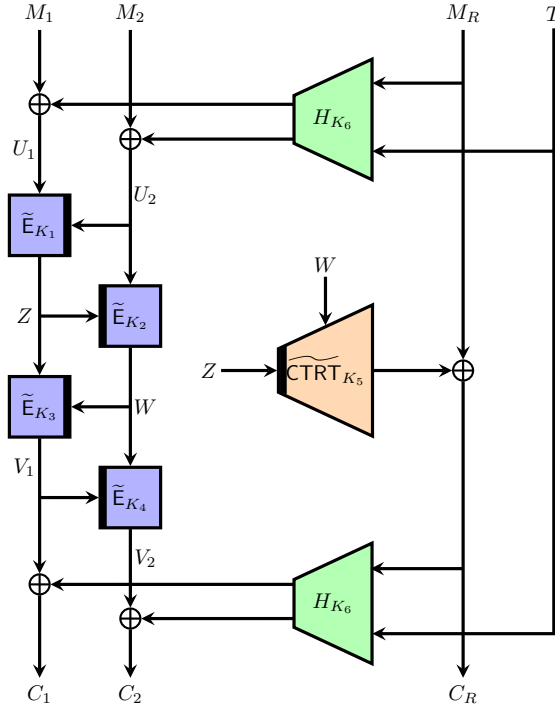


Figure 2: Schematic representation of HCTR+ construction. H_{K_6} is a $2n$ -bit keyed hash function, \widetilde{E} is an n -bit tweakable block cipher with n -bit tweak and $\widetilde{\text{CTRT}}$ is a tweakable block cipher based counter mode encryption, where the tweak of the underlying TBC is incremented in every call of the primitive.

Why to choose the internal values Z and W : By looking at Fig. 2, it is evident that other than Z and W , there are four internal variables U_1, U_2, V_1, V_2 . An adversary can enforce a collision between U_1^i and U_1^j by choosing $T^i = T^j$, $(M_R^i, M_1^i) = (M_R^j, M_1^j)$ and

($M_2^i \neq M_2^j$). Similarly, an adversary can enforce a collision between U_2^i and U_2^j . In addition, with appropriate choice of messages, an adversary can also control the difference of (U_1^i and U_1^j) or (U_2^i and U_2^j). This phenomenon leads us not to feed U_1 (or U_2) directly as the tweak or input in the $\widetilde{\text{CTR}}\text{T}$ as we cannot bound Bad6 or Bad7 , as defined in Defn. 2, (if U_1 or U_2 is tweak) up to n -bit because the adversary can control the tweak difference. Similarly, we cannot bound Bad3 , as defined in Defn. 2, (if U_1 or U_2 is input) up to n -bit because the adversary can enforce a collision. The same argument holds true for V_1 or V_2 as well. In a similar retrospect, we do not think that TLR4 can be replaced with TLR3 as for TLR3 , the intermediate values would be U_1, U_2, Z, V_1 , and V_2 . As we argued that we cannot use U_1, U_2, V_1 or V_2 either as a tweak or an input to the $\widetilde{\text{CTR}}\text{T}$ mode. Therefore, we are only left with the intermediate value Z , which should be used both as a tweak and an input to the $\widetilde{\text{CTR}}\text{T}$ mode. Note that, this option does not give optimal security because it only ensures n -bit entropy, whereas we require $2n$ -bit entropy for the $\widetilde{\text{CTR}}\text{T}$ mode.

4 Proof of Theorem 2

Let \mathcal{A} be a $(q, \ell_{\max}, \mathbf{t})$ distinguisher against the strong tweakable pseudorandom permutation security of $\text{HCTR}^+[\widetilde{\text{E}}, \text{H}]$. We consider another construction R-HCTR^+ that is identical to the HCTR^+ construction, the only difference is that the key derivation step of HCTR^+ construction is replaced by uniform random sampling of six n -bit keys. Thus, by following the PRP-PRF switching lemma [CN08b], we have

$$\text{Adv}_{\text{HCTR}^+[\widetilde{\text{E}}, \text{H}]}^{\text{STPRP}}(\mathcal{A}) \leq \text{Adv}_{\widetilde{\text{E}}}^{\text{TPRP}}(\mathcal{A}) + \text{Adv}_{\text{R-HCTR}^+[\widetilde{\text{E}}, \text{H}]}^{\text{STPRP}}(\mathcal{A}) + \frac{15}{2^n}.$$

Thus, it is enough to bound the stprp advantage of the R-HCTR^+ construction. Initially, we replace the five independently keyed tweakable block ciphers $\widetilde{\text{E}}_{K_1}, \widetilde{\text{E}}_{K_2}, \widetilde{\text{E}}_{K_3}, \widetilde{\text{E}}_{K_4}$, and $\widetilde{\text{E}}_{K_5}$ with five independently sampled tweakable random permutations $\widetilde{\text{P}} := (\widetilde{\text{P}}_1, \widetilde{\text{P}}_2, \widetilde{\text{P}}_3, \widetilde{\text{P}}_4, \widetilde{\text{P}}_5)$ at the cost of the strong tweakable pseudorandom permutation advantage of the underlying TBC. Therefore, we have

$$\text{Adv}_{\text{R-HCTR}^+[\widetilde{\text{E}}, \text{H}]}^{\text{STPRP}}(\mathcal{A}) \leq 5\text{Adv}_{\widetilde{\text{E}}}^{\text{STPRP}}(\mathcal{A}) + \underbrace{\text{Adv}_{\text{R-HCTR}^+[\widetilde{\text{P}}, \text{H}]}^{\text{STPRP}}(\mathcal{A})}_{\delta}$$

From now onwards, we omit $\widetilde{\text{P}}$ and H from the notation $\text{R-HCTR}^+[\widetilde{\text{P}}, \text{H}]$ and simply write R-HCTR^+ whenever they are understood from the context. Now, our goal is to upper bound δ . Note that, we have

$$\delta \leq \max_{\mathcal{A}} \left| \Pr[\mathcal{A}^{\text{R-HCTR}^+, (\text{R-HCTR}^+)^{-1}} = 1] - \Pr[\mathcal{A}^{\text{PP}, \text{PP}^{-1}} = 1] \right|,$$

where the first probability is taken over the randomness of $\widetilde{\text{P}}_i \leftarrow_{\$} \text{TP}(\{0, 1\}^n, \{0, 1\}^n)$ for $i \in [6]$ and the second probability is computed over the randomness of $\text{PP} \leftarrow_{\$} \text{TP}(\{0, 1\}^*, \{0, 1\}^*)$. Moreover, the maximum is taken over non-trivial adversaries⁶. Hence, δ can not be larger than the advantage of the best non-trivial adversary between the two world $(\text{R-HCTR}^+, (\text{R-HCTR}^+)^{-1})$ and $(\text{PP}, (\text{PP})^{-1})$. This formulation allows us to apply the H-Coefficient Technique [Pat08].

4.1 Extended Query Transcript

We fix a non-trivial distinguisher \mathcal{A} and assume that \mathcal{A} is computationally unbounded and hence without loss of generality a deterministic distinguisher. \mathcal{A} interacts either with

⁶A non-trivial adversary is one who does not repeat queries.

the real world $(\text{R-HCTR}+^*, (\text{R-HCTR}+^*)^{-1})$ or with the ideal world $(\text{PP}, (\text{PP})^{-1})$. In the online phase of the interaction, \mathcal{A} obtains either $C = \text{R-HCTR}+^*(T, M)$ in the real world or $\text{PP}(T, M)$ corresponding to the encryption query (T, M) , where $M = M_1 \| M_2 \| M_R$ and $C = C_1 \| C_2 \| C_R$. Similarly, it will get $M = (\text{R-HCTR}+^*)^{-1}(T, C)$ or $(\text{PP})^{-1}(T, C)$ corresponding to the decryption query (T, C) .

4.1.1 Releasing Additional Informations

After the interaction is over, but before outputting the decision bit, the distinguisher is provided the hash key K_6 . In the real world, K_6 is the actual hash key used in the construction, whereas in the ideal world a dummy hash key K_6 is sampled uniformly and independently from the hash key space $\{0, 1\}^n$. The distinguisher is also provided with intermediate variables as an additional information. In the real world, the distinguisher is provided with the pair of tuples (Z^1, Z^2, \dots, Z^q) and (W^1, W^2, \dots, W^q) , which have been generated in the construction $\text{R-HCTR}+^*$. However, in the ideal world, these additional information are required to be generated. We sample these pair of q tuples (Z^1, Z^2, \dots, Z^q) and (W^1, W^2, \dots, W^q) in the ideal world using a sampler S whose objective would be to sample these pair of tuples in such a way so that it becomes close to the distribution of the the tuple (Z^1, Z^2, \dots, Z^q) and (W^1, W^2, \dots, W^q) respectively, generated in the real world. To do this, S will simulate the tweakable random permutations \tilde{P}_1 and \tilde{P}_2 used in the real construction with the help of the following two tables $\Pi_1[\cdot, \cdot]$ and $\Pi_2[\cdot, \cdot]$. In other words, \tilde{P}_1 will be simulated using the table $\Pi_1[\cdot, \cdot]$ and \tilde{P}_2 will be simulated using the table $\Pi_2[\cdot, \cdot]$. Each of the tables are initialized as empty. The sampler S runs as follows.

For i -th encryption or decryption query,

1. Compute $U_1^i := M_1^i \oplus \text{H}_{K_6}(T^i, M_R^i)$
2. Compute $U_2^i := M_2^i \oplus \text{H}_{K_6}(T^i, M_R^i)$
3. Checks if $\Pi_1[U_2^i, U_1^i]$ has already been set. If it has not been set, S samples Z^i as follows:

$$Z^i := \Pi_1[U_2^i, U_1^i] \leftarrow_{\$} \{0, 1\}^n \setminus \text{Rng}(\Pi_1[U_2^i, \cdot])$$

and set $Z^i \leftarrow \Pi_1[U_2^i, U_1^i]$.

4. Checks if $\Pi_2[Z^i, U_2^i]$ has already been set. If it has not been set, S samples W^i as follows:

$$W^i := \Pi_2[Z^i, U_2^i] \leftarrow_{\$} \{0, 1\}^n \setminus \text{Rng}(\Pi_2[Z^i, \cdot])$$

and set $W^i \leftarrow \Pi_2[Z^i, U_2^i]$.

The adversary is provided these intermediate variables Z^i, W^i for all $i \in [q]$. Note that, these additional informations do not degrade the advantage of the adversary as it is always possible to discard them. Therefore, after releasing the additional informations, the overall attack transcript is $\tau = (\tau', K_6)$, where

$$\tau' = \left((T^1, M^1, C^1, Z^1, W^1), \dots, (T^q, M^q, C^q, Z^q, W^q) \right).$$

4.2 Defining and Bounding Bad Transcripts

Let Θ be the set of all transcripts τ such that the probability of realizing it in the ideal world is non-zero. We begin with defining the bad transcripts and bound their probability in the ideal world. We would like to note that the underlying principle for identifying the bad events is

non-trivial collisions in the tweak-input or tweak-output pairs for any call to the tweakable permutations of the construction.

Definition 2. An attainable transcript $\tau = (\tau', K_6)$ is **bad**, if there exist two queries i and j (w.l.o.g $j < i$), such that either of the following holds:

1. **Bad1:** For any $i, j \in [q]$, such that $i \neq j$:

$$\mathsf{H}_{K_6}(T^i, M_R^i) \oplus \mathsf{H}_{K_6}(T^j, M_R^j) = (M_1^i \| M_2^i) \oplus (M_1^j \| M_2^j)$$

2. **Bad2:** For any $i, j \in [q]$, such that $i \neq j$:

$$Z^i = Z^j \quad \text{and} \quad M_2^i \oplus \mathsf{H}_{K_6}[2](T^i, M_R^i) = M_2^j \oplus \mathsf{H}_{K_6}[2](T^j, M_R^j)$$

3. **Bad3:** For any $i, j \in [q]$, such that $i \neq j$:

$$Z^i = Z^j \quad \text{and} \quad W^i = W^j$$

4. **Bad4:** For any $i, j \in [q]$, such that $i \neq j$:

$$W^i = W^j \quad \text{and} \quad C_1^i \oplus \mathsf{H}_{K_6}[1](T^i, C_R^i) = C_1^j \oplus \mathsf{H}_{K_6}[1](T^j, C_R^j)$$

5. **Bad5:** For any $i, j \in [q]$, such that $i \neq j$:

$$\mathsf{H}_{K_6}(T^i, C_R^i) \oplus \mathsf{H}_{K_6}(T^j, C_R^j) = (C_1^i \| C_2^i) \oplus (C_1^j \| C_2^j)$$

6. **Bad6:** For any $i, j \in [q]$ and $\alpha \in [\ell_i - 2], \beta \in [\ell_j - 2]$, such that $i \neq j$:

$$Z^i \oplus \langle \alpha \rangle_n = Z^j \oplus \langle \beta \rangle_n \quad \text{and} \quad W^i = W^j$$

7. **Bad7:** For any $i, j \in [q]$ and $\alpha \in [\ell_i - 2], \beta \in [\ell_j - 2]$, such that $i \neq j$:

$$Z^i \oplus \langle \alpha \rangle_n = Z^j \oplus \langle \beta \rangle_n \quad \text{and} \quad M_{\alpha+2}^i \oplus C_{\alpha+2}^i = M_{\beta+2}^j \oplus C_{\beta+2}^j$$

Let Θ_{bad} denotes the set of all attainable transcripts τ such that it satisfies one of the above conditions and the event **Bad** denotes

$$\text{Bad} := \bigvee_{i=1}^7 \text{Bad}i.$$

Following Lemma establish an upper bound on the probability of the event **Bad** holds under the ideal world distribution.

Lemma 1. *Let X_{id} and the event **Bad** be defined as above. Then, for any integer q such that $q \leq 2^{n-1}$, one has*

$$\epsilon_{\text{bad}} = \Pr[X_{\text{id}} \in \Theta_{\text{bad}}] \leq \frac{q^2 \epsilon_1}{2^{n+1}} + \frac{q^2 \epsilon_2}{2^{n+1}} + q^2 \epsilon + \frac{2q^2 \ell_{\max}}{2^{2n}} + \frac{q^2}{2^{2n+1}}.$$

Proof. Recall that $\text{Bad} := \text{Bad1} \vee \text{Bad2} \vee \text{Bad3} \vee \text{Bad4} \vee \text{Bad5} \vee \text{Bad6} \vee \text{Bad7}$. We bound the probability of the individual events $\text{Bad}i$, and then by virtue of the union bound, we sum up the individual bounds to obtain the overall bound of the probability of the event Bad .

Bounding Bad1: For a fixed choice of distinct indices $i, j \in [q]$, the probability of the event Bad1 is boiled down to the following equation:

$$\mathbf{H}_{K_6}(T^i, M_R^i) \oplus \mathbf{H}_{K_6}(T^j, M_R^j) = (M_1^i \oplus M_1^j) \parallel (M_2^i \oplus M_2^j).$$

Note that, by definition of the almost-xor-universal of a $2n$ -bit keyed hash function, the above event is bounded by ϵ , where ϵ is the almost-xor-universal advantage of the $2n$ -bit keyed hash function \mathbf{H} . Therefore, by summing over all possible choices of (i, j) , we have

$$\Pr[\text{Bad1}] \leq \frac{q^2 \epsilon}{2}. \quad (1)$$

Bounding Bad2: For a fixed choice of distinct indices $i > j \in [q]$, the probability of the event $Z^i = Z^j$ is upper bounded by $1/2^n$ using the randomness of Z^i and the event $M_2^i \oplus \mathbf{H}_{K_6}[2](T^i, M_R^i) = M_2^j \oplus \mathbf{H}_{K_6}[2](T^j, M_R^j)$ is upper bounded by ϵ_2 , where ϵ_2 is the almost-xor-universal advantage of $\mathbf{H}[2]$. Therefore, by summing over all possible choices of (i, j) , we have

$$\Pr[\text{Bad2}] \leq \frac{q^2 \epsilon_2}{2^{n+1}}. \quad (2)$$

Bounding Bad3: For a fixed choice of distinct indices $i > j \in [q]$, the probability of each event $Z^i = Z^j$ and $W^i = W^j$ is upper bounded by $1/2^n$, due to the randomness and the independence of the sampling of Z^i and W^i in the ideal world. Hence, by summing over all possible choices of indices, we have

$$\Pr[\text{Bad3}] \leq \frac{q^2}{2^{2n+1}}. \quad (3)$$

Bounding Bad4: For a fixed choice of distinct indices $i > j \in [q]$, the probability of the first event $W^i = W^j$ is upper bounded by $1/2^n$ using randomness of W^i and the second event $C_1^i \oplus \mathbf{H}_{K_6}[1](T^i, C_R^i) = C_1^j \oplus \mathbf{H}_{K_6}[1](T^j, C_R^j)$ is upper bounded by ϵ_1 , where ϵ_1 is the almost-xor-universal advantage of $\mathbf{H}[1]$. Therefore, by summing over all possible choices of (i, j) , we have

$$\Pr[\text{Bad4}] \leq \frac{q^2 \epsilon_1}{2^{n+1}}. \quad (4)$$

Bounding Bad5: We bound this event similar to bounding as Bad1 . For a fixed choice of indices $i, j \in [q]$, the probability of the event Bad5 is boiled down to the following equation:

$$\mathbf{H}_{K_6}(T^i, C_R^i) \oplus \mathbf{H}_{K_6}(T^j, C_R^j) = (C_1^i \oplus C_1^j) \parallel (C_2^i \oplus C_2^j).$$

Note that, by definition of the almost-xor-universal of a $2n$ -bit keyed hash function, the above event is bounded by ϵ , where ϵ is the almost-xor-universal advantage of the $2n$ -bit keyed hash function \mathbf{H} . Therefore, by summing over all possible choices of (i, j) , we have

$$\Pr[\text{Bad5}] \leq \frac{q^2 \epsilon}{2}. \quad (5)$$

Bounding Bad6: For a fixed choices of indices $i > j \in [q]$, $\alpha \in [\ell_i - 2]$, $\beta \in [\ell_j - 2]$, the event $Z^i \oplus \langle \alpha \rangle_n = Z^j \oplus \langle \beta \rangle_n$ is upper bounded by $1/2^n$ due to randomness of Z^i . Similarly, using the randomness of W^i the event $W^i = W^j$ is upper bounded by $1/2^n$. Therefore, by summing over all possible choices of indices i, j, α, β we have

$$\Pr[\text{Bad6}] \leq \frac{q^2 \ell_{\max}}{2^{2n}}. \quad (6)$$

Bounding Bad7: Similar to bounding the event Bad6, for a fixed choice of indices $i > j \in [q]$, $\alpha \in [\ell_i - 2]$, $\beta \in [\ell_j - 2]$, the event $Z^i \oplus \langle \alpha \rangle_n = Z^j \oplus \langle \beta \rangle_n$ is upper bounded by $1/2^n$. The second event, i.e., $M_{\alpha+2}^i \oplus C_{\alpha+2}^i = M_{\beta+2}^j \oplus C_{\beta+2}^j$ is upper bounded by $1/2^n$, due to the randomness of $C_{\alpha+2}^i$ (in case i -th query being an encryption query) or due to the randomness of $M_{\alpha+2}^i$ (in case i -th query being a decryption query). Therefore, by summing over all possible choices of indices i, j, α, β we have

$$\Pr[\text{Bad7}] \leq \frac{q^2 \ell_{\max}}{2^{2n}}. \quad (7)$$

The result follows by applying the union bound on the probability of the above events as stated in Eqn. (1)-Eqn. (7). \square

4.3 Analysis of Good Transcripts

In this section, we fix a good transcript $\tau = (\tau', K_6)$ and we have to lower bound the ratio of real to ideal interpolation probability. Formally, we have the following lemma.

Lemma 2. *Let $\tau = (\tau', K_6)$, where $\tau' = \left((T^1, M^1, C^1, Z^1, W^1), \dots, (T^q, M^q, C^q, Z^q, W^q) \right)$ be a good transcript. Then*

$$\frac{\Pr[X_{\text{re}} = \tau]}{\Pr[X_{\text{id}} = \tau]} \geq 1 - \underbrace{\frac{q^2}{2^{2n}}}_{\epsilon_{\text{good}}}.$$

Proof. Let $\tau = (\tau', K_6) \in \Theta_{\text{good}}$ and we have to lower bound the ratio of real and ideal interpolation probability. To do so, we begin by introducing a few notations required to calculate the ratio.

- Let \mathcal{T}_i be the set of tweaks queried to \tilde{P}_i , for $i = 1, \dots, 5$.
- Let $\alpha_T, \beta_T, \gamma_T, \delta_T$, and ξ_T be the number of calls to \tilde{P}_i with tweak $T \in \mathcal{T}_i$ for $i = 1, \dots, 5$ respectively.
- Let L be the number of distinct input lengths to the construction and we denote those input lengths with $\ell_1, \ell_2, \dots, \ell_L$. Let $\ell_{\min} = \min\{\ell_1, \ell_2, \dots, \ell_L\}$
- Let q_i be the number of queries of length ℓ_i .
- Let t_i be the number of distinct tweaks queried to the construction for inputs of length ℓ_i and $q_{i,j}$ be the number of calls of j -th tweak for inputs of length ℓ_i .

Therefore, it holds that

$$\sum_{T \in \mathcal{T}_1} \alpha_T = \sum_{T \in \mathcal{T}_2} \beta_T = \sum_{T \in \mathcal{T}_3} \gamma_T = \sum_{T \in \mathcal{T}_4} \delta_T = q. \quad (8)$$

Without loss of generality, we assume that each ℓ_i is a multiple of n . Thus, we assume that each input length ℓ_i consists of κ_i blocks, i.e., $\ell_i = n\kappa_i$. Moreover, $\ell_{\min} \geq 2n$.

4.3.1 Real Interpolation Probability

To compute the real interpolation probability, we would like to note that the hash key K_6 has been sampled uniformly at random from $\{0, 1\}^n$. Since we consider τ is a good transcript, there will be no collision in the input or the output to the pair of independent tweakable random permutations for some fixed tweak queried to it. Thus, to compute the real interpolation probability for the good transcript τ , we count the number of times each tweakable random permutation has been invoked for each distinct tweak. Therefore, we have

$$\begin{aligned} \Pr[X_{\text{re}} = \tau] &= \frac{1}{2^n} \times \prod_{T \in \mathcal{T}_1} \frac{1}{(2^n)_{\alpha_T}} \times \prod_{T \in \mathcal{T}_2} \frac{1}{(2^n)_{\beta_T}} \times \prod_{T \in \mathcal{T}_3} \frac{1}{(2^n)_{\gamma_T}} \\ &\quad \times \prod_{T \in \mathcal{T}_4} \frac{1}{(2^n)_{\delta_T}} \times \prod_{T \in \mathcal{T}_5} \frac{1}{(2^n)_{\xi_T}} \end{aligned} \quad (9)$$

4.3.2 Ideal Interpolation Probability

In the ideal world, the ciphertext $C = C_1 \| C_2 \| C_R$ is the response of the tweakable random permutation PP of an encryption query $M = M_1 \| M_2 \| M_R$. Similarly, $M = M_1 \| M_2 \| M_R$ is the response of the tweakable random permutation $(\text{PP})^{-1}$ of a decryption query $C = C_1 \| C_2 \| C_R$. After the interaction is over, a dummy hash key K_6 has been sampled uniformly at random from $\{0, 1\}^n$. Finally, the intermediate random variables Z, W have been sampled according to the procedures described in Sect. 4.1. Thus, the ideal interpolation probability for the good transcript τ is

$$\Pr[X_{\text{id}} = \tau] = \frac{1}{2^n} \times \prod_{i=1}^L \prod_{j=1}^{t_i} \frac{1}{(2^{\ell_i})_{q_{i,j}}} \times \prod_{T \in \mathcal{T}_1} \frac{1}{(2^n)_{\alpha_T}} \times \prod_{T \in \mathcal{T}_2} \frac{1}{(2^n)_{\beta_T}}. \quad (10)$$

4.3.3 Ratio of Real to Ideal Interpolation Probability

By taking ratio of Eqn. (9) to Eqn. (10), we have the following

$$\frac{\Pr[X_{\text{re}} = \tau]}{\Pr[X_{\text{id}} = \tau]} = \frac{\prod_{i=1}^L \prod_{j=1}^{t_i} (2^{\ell_i})_{q_{i,j}}}{\prod_{T \in \mathcal{T}_3} (2^n)_{\gamma_T} \times \prod_{T \in \mathcal{T}_4} (2^n)_{\delta_T} \times \prod_{T \in \mathcal{T}_5} (2^n)_{\xi_T}}$$

Applying the fact that $(2^n)_{\gamma_T} \leq (2^n)^{\gamma_T}$ for $T \in \mathcal{T}_3$, $(2^n)_{\delta_T} \leq (2^n)^{\delta_T}$ for $T \in \mathcal{T}_4$ and $(2^n)_{\xi_T} \leq (2^n)^{\xi_T}$ for $T \in \mathcal{T}_5$ we have

$$\begin{aligned} \frac{\Pr[X_{\text{re}} = \tau]}{\Pr[X_{\text{id}} = \tau]} &\geq \frac{\prod_{i=1}^L \prod_{j=1}^{t_i} (2^{\ell_i})_{q_{i,j}}}{\prod_{T \in \mathcal{T}_3} (2^n)^{\gamma_T} \times \prod_{T \in \mathcal{T}_4} (2^n)^{\delta_T} \times \prod_{T \in \mathcal{T}_5} (2^n)^{\xi_T}} \\ &\geq \frac{\prod_{i=1}^L \prod_{j=1}^{t_i} (2^{\ell_i})_{q_{i,j}}}{(2^n)^{\sum_{T \in \mathcal{T}_3} \gamma_T} \times (2^n)^{\sum_{T \in \mathcal{T}_4} \delta_T} \times (2^n)^{\sum_{T \in \mathcal{T}_5} \xi_T}} \end{aligned}$$

Following Eqn. (8) and $\sum_{T \in \mathcal{T}_5} \xi_T \leq (\sum_{i=1}^L \sum_{j=1}^{t_i} \kappa_i \cdot q_{i,j}) - 2q$, we get

$$\begin{aligned} \frac{\Pr[X_{\text{re}} = \tau]}{\Pr[X_{\text{id}} = \tau]} &\geq \frac{\prod_{i=1}^L \prod_{j=1}^{t_i} (2^{n\kappa_i})_{q_{i,j}}}{2^{2nq} \times (2^n)^{\sum_{i=1}^L \sum_{j=1}^{t_i} \kappa_i \cdot q_{i,j}} \times 2^{-2nq}} \geq \frac{\prod_{i=1}^L \prod_{j=1}^{t_i} (2^{n\kappa_i})_{q_{i,j}}}{(2^n)^{\sum_{i=1}^L \sum_{j=1}^{t_i} \kappa_i \cdot q_{i,j}}} \\ &\geq \prod_{i=1}^L \prod_{j=1}^{t_i} \prod_{r=1}^{q_{i,j}-1} \left(1 - \frac{r}{2^{n\kappa_i}}\right) \geq 1 - \sum_{i=1}^L \sum_{j=1}^{t_i} \sum_{r=1}^{q_{i,j}-1} \frac{r}{2^{n\kappa_i}} \\ &\geq 1 - \sum_{i=1}^L \sum_{j=1}^{t_i} \frac{q_{i,j}^2}{2^{n\kappa_i}} \end{aligned} \quad (11)$$

Since, κ_{\min} is the number of blocks contained in ℓ_{\min} , where $\ell_{\min} \geq 2n$ and $\sum_{i=1}^L \sum_{j=1}^{t_i} q_{i,j} \leq q$, following Eqn. (11), we have

$$\frac{\Pr[X_{\text{re}} = \tau]}{\Pr[X_{\text{id}} = \tau]} \geq 1 - \frac{q^2}{2^{n\kappa_{\min}}} = 1 - \frac{q^2}{2^{2n}}.$$

This concludes the proof of Lemma 2. \square

Finally, by applying H-Coefficient technique, Lemma 1 and Lemma 2, the result of Theorem 2 follows.

5 Instantiation of HCTR+

In this section, we propose two single-keyed variable input length tweakable sprp that achieves optimal security. Our constructions can be seen as an instantiation of our generic framework HCTR+ by replacing its $2n$ -bit generic keyed hash function with some concrete proposals. In the following two sections, we provide a complete description of the two constructions along with their corresponding security bound.

5.1 PHCTR+: An Optimally Secure WBC

In this section, we propose an optimally secure single keyed variable input length tweakable sprp, dubbed as PHCTR+. Our construction is a specific instance of the HCTR+ construction, where we instantiate the underlying $2n$ -bit keyed hash function of HCTR+ by PHASH+ function. PHASH+ is structurally almost similar to the PMAC2x [LN16] construction with a slight difference, explained below.

On input a key K_h , a message D and a tweak T , we first apply an injective encoding function on $T \parallel D$ to derive ℓ many blocks $B[1], B[2], \dots, B[\ell]$. Then, we process each block $B[i]$ by evaluating a TBC \tilde{E}_{K_h} , as shown in line-4 of Fig. 3, which yields a $2n$ -bit state (X, Y) . The leftmost n -bit part of the state is obtained by taking the xor of the output of the tweakable block cipher evaluated on individual blocks $B[i]$ and the rightmost n -bit state is obtained by taking a linear combination of the output of the tweakable block cipher evaluated on individual blocks $B[i]$, i.e.,

$$X[\ell] = Z[1] \oplus Z[2] \oplus \dots \oplus Z[\ell], \quad Y[\ell] = 2^{\ell-1}Z[1] \oplus 2^{\ell-2}Z[2] \oplus \dots \oplus Z[\ell],$$

where $Z[i]$ denotes the output of the tweakable block cipher evaluated on the block $B[i]$ with tweak $(000 \parallel \langle i \rangle_{n-3})$. In this regard, we would like to mention that PMAC2x construction derives $Y[\ell]$ as follows:

$$Y[\ell] = 2^\ell Z[1] \oplus 2^{\ell-1} Z[2] \oplus \dots \oplus 2Z[\ell],$$

whereas we generate $Y[\ell]$ as

$$Y[\ell] = 2^{\ell-1} Z[1] \oplus 2^{\ell-2} Z[2] \oplus \dots \oplus Z[\ell].$$

Thus, our construction potentially saves one doubling operation compared to the PMAC2x construction. Finally, we compute the $2n$ bit hash output (U, V) , where U is obtained by evaluating the tweakable block cipher on input $X[\ell]$ with tweak $(010 \parallel \text{Lsb}_{n-3}(Y_\ell))$ and V is obtained by evaluating the tweakable block cipher on input $Y[\ell]$ with tweak $(011 \parallel \text{Lsb}_{n-3}(X_\ell))$.

ENCODING FUNCTION. The encoding function takes a variable length tweak T and a variable length message D and outputs a string $B \in (\{0, 1\}^n)^+$ as follows:

$$\text{ENCODE}(T, D) := \text{Pad}_n(T) \parallel \text{Pad}_n(D) \parallel (\langle T \rangle_{n/2} \parallel \langle D \rangle_{n/2}).$$

<p>PHASH+(K_h, D, T)</p> <ol style="list-style-type: none"> 1. $(X[0], Y[0]) \leftarrow (0^n, 0^n)$; 2. $(B[1], B[2], \dots, B[\ell]) \leftarrow \text{ENCODE}(T, D)$; 3. for $i = 1$ to ℓ; 4. $Z[i] \leftarrow \tilde{\text{E}}_{K_h}((000\ \langle i \rangle_{n-3}), B[i])$; 5. $X[i] \leftarrow X[i-1] \oplus Z[i]$; 6. $Y[i] \leftarrow 2Y[i-1] \oplus Z[i]$; 7. $U \leftarrow \tilde{\text{E}}_{K_h}((010\ \text{Lsb}_{n-3}(Y[\ell]), X[\ell])$; 8. $V \leftarrow \tilde{\text{E}}_{K_h}((011\ \text{Lsb}_{n-3}(X[\ell]), Y[\ell])$; 9. return (U, V); <hr style="border: 0.5px solid black;"/> <p>ENCODE(T, D)</p> <ol style="list-style-type: none"> 1. $Len \leftarrow \langle T \rangle_{n/2} \ \langle D \rangle_{n/2}$ 2. $B \leftarrow \text{Pad}_n(T) \ \text{Pad}_n(D) \ \text{Len}$; 3. $(B[1], B[2], \dots, B[\ell]) \stackrel{n}{\leftarrow} \text{Parse}(B)$; 4. return $(B[1], B[2], \dots, B[\ell])$; 	<p>PHCTR+(K, M, T)</p> <ol style="list-style-type: none"> 1. $((M_1\ M_2), M_R) \stackrel{2n}{\leftarrow} M$; 2. for $i = 1$ to 6 3. $K_i \leftarrow \tilde{\text{E}}_K(10^{n-1}, \langle i \rangle_n)$; 4. $H_1\ H_2 \stackrel{n}{\leftarrow} \text{PHASH+}(K_6, M_R, T)$; 5. $U_1 \leftarrow M_1 \oplus H_1$; $U_2 \leftarrow M_2 \oplus H_2$; 6. $Z \leftarrow \tilde{\text{E}}_{K_1}(U_2, U_1)$; $W \leftarrow \tilde{\text{E}}_{K_2}(Z, U_2)$; 7. $V_1 \leftarrow \tilde{\text{E}}_{K_3}(W, Z)$; $V_2 \leftarrow \tilde{\text{E}}_{K_4}(V_1, W)$; 8. for $i = 1$ to $\lceil M_R /n \rceil$ 9. $S_i \leftarrow \tilde{\text{E}}_{K_5}(Z \oplus \langle i \rangle_n, W)$; 10. $S \stackrel{\text{def}}{=} S_1 \ \dots\ S_{\lceil M_R /n \rceil}$; 11. $C_R \leftarrow \text{Msb}_{ M_R }(S) \oplus M_R$; 12. $H'_1\ H'_2 \stackrel{n}{\leftarrow} \text{PHASH+}(K_6, C_R, T)$; 13. $C_1 \leftarrow V_1 \oplus H'_1$; $C_2 \leftarrow V_2 \oplus H'_2$; 14. return $(C_1\ C_2\ C_R)$;
--	---

Figure 3: We describe the hash function PHASH+ in the left hand side of the algorithm and PHCTR+ construction is described in the right hand side of the algorithm. ENCODING function is described in the lower left of the algorithm.

Since we have encoded the length information of T and D , it is immediate to see that the encoding function is injective. An algorithmic description of the encoding function is given in Fig. 3.

To count the number of tweakable block cipher calls of PHCTR+ construction, let L_M denote the length of the input message M in number of bits to the construction and L_T denotes the length of the tweak T in number of bits to the construction. In each of the two layers of the hash function, we require $\ell + 2$ many tweakable block cipher calls, where

$$\ell = \left(\left\lfloor \frac{L_M - 2n}{n} \right\rfloor + \left\lfloor \frac{L_T}{n} \right\rfloor + 3 \right).$$

Therefore, in the PHCTR+ construction, we require a total of

$$6 + 2(\ell + 2) + 4 + \left\lceil \frac{L_M - 2n}{n} \right\rceil \quad (12)$$

tweakable block cipher calls, where six tweakable block cipher calls are required to derive six keys, four tweakable block cipher calls are required in the left hand side of the construction and $\lceil (L_M - 2n)/n \rceil$ many tweakable block cipher calls are required in counter mode encryption. Therefore, by plugging-in the value of ℓ in Eqn. (12), the construction requires a total of

$$2 \left(\left\lfloor \frac{L_M - 2n}{n} \right\rfloor + \left\lfloor \frac{L_T}{n} \right\rfloor \right) + \left\lceil \frac{L_M}{n} \right\rceil + 20$$

tweakable block cipher calls.

Remark 1. We would like to mention that the original proposal of the PMAC2x [LN17b] construction was shown to be insecure by Minematsu and Iwata [MI17] by exploiting the fact that different tweaks were used in processing the last message block based on whether the message is full or not. Later on, the authors have revised their eprint version of the paper [LN16] and introduces the always padding restriction on the input message, i.e., the input message is always padded with 10^* irrespective of whether the last message block is full or partial. This fix removed the flaw of the original design of PMAC2x [LN17b] construction. We would like to mention that we have adopted the revised construction of PMAC2x [LN16], i.e., we always pad the input message with 10^* , to design PHASH+.

5.2 Security Result of PHCTR+

In this section we state the security result of PHCTR+ construction, which is described as above. Before that, we state the following result on the almost-xor-universal advantage of the PHASH+ construction (in Lemma 3) and the almost-xor-universal advantage of each of its n -bit output (in Lemma 4).

Lemma 3. *Let \mathcal{K} be a non-empty finite set and $\tilde{E} : \mathcal{K} \times \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a tweakable block cipher. Let PHASH+ be defined as above in Fig. 3. Then, for any two distinct messages M and M' such that the number of message blocks in M and M' be ℓ and ℓ' respectively, and for any $2n$ -bit string Δ , we have*

$$\Pr[K_h \leftarrow \mathcal{K} : \text{PHASH+}[\tilde{E}_{K_h}](M) \oplus \text{ZHASH+}[\tilde{E}_{K_h}](M') = \Delta] \leq \frac{118}{(2^n - 1)^2} + \frac{1}{2^{2n}}$$

Lemma 4. *Let \mathcal{K} be a non-empty finite set and $\tilde{E} : \mathcal{K} \times \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a tweakable block cipher. Let U denote the leftmost n -bit output of PHASH+ and V denote the rightmost n -bit output of PHASH+. Then, for any n -bit string δ , we have*

$$\begin{aligned} \Pr[K_h \leftarrow \mathcal{K} : U \oplus U' = \delta] &\leq \frac{63}{2^n - 1} + \frac{1}{2^{2n}}, \\ \Pr[K_h \leftarrow \mathcal{K} : V \oplus V' = \delta] &\leq \frac{63}{2^n - 1} + \frac{1}{2^{2n}}. \end{aligned}$$

Proofs of the above two lemmas are deferred to Appendix A.1 and A.2. By combining Theorem 2, Lemma 3, and Lemma 4, we derive the security bound of PHCTR+ construction as follows:

Theorem 3. *Let \mathcal{K} be a non-empty finite set. Let $\tilde{E} : \mathcal{K} \times \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be an (n, n) tweakable block cipher. Then, for any $(q, \ell_{\max}, \mathfrak{t})$ -chosen plaintext chosen ciphertext adaptive adversary \mathcal{A} against the stprp security of PHCTR+ $[\tilde{E}]$ construction such that each query is of length at least $2n$ bits, there exists a (σ, \mathfrak{t}') -chosen plaintext chosen ciphertext adversary \mathcal{A}' against the stprp security of \tilde{E} , such that*

$$\text{Adv}_{\text{PHCTR+}[\tilde{E}]}^{\text{STPRP}}(\mathcal{A}) \leq 6\text{Adv}_{\tilde{E}}^{\text{STPRP}}(\mathcal{A}') + \frac{183q^2}{(2^n - 1)^2} + \frac{2q^2\ell_{\max}}{2^{2n}} + \frac{2q^2}{2^{2n}} + \frac{15}{2^n}, \quad (13)$$

where σ is the total number of message and tweaks blocks queried, $\mathfrak{t}' = O(\mathfrak{t} + \sigma + 2qt_H)$, and t_H is the time for computing PHASH+.

5.3 ZHCTR+: An Optimally Secure WBC

In this section, we propose another optimally secure single keyed variable input length tweakable sprp, dubbed as ZHCTR+. As before, ZHCTR+ is a specific instance of the

HCTR+ construction, where we instantiate the underlying $2n$ -bit keyed hash function of HCTR+ by ZHASH+ function. ZHASH+ is structurally similar to the ZMAC [IMPS17] construction with some subtle differences between the two designs, explained in Sect. 5.3. On input a key K_h , a message D and a tweak T , we first apply an injective encoding

ZHASH+(K_h, D, T)	ZHCTR+(K, M, T)
<ol style="list-style-type: none"> 1. $(U, V) \leftarrow (0^n, 0^n)$; 2. $L_l \leftarrow \tilde{\mathbf{E}}_{K_h}(000\ 0^{n-3}, 0^n)$; 3. $L_r \leftarrow \tilde{\mathbf{E}}_{K_h}(000\ 0^{n-4}\ 1, 0^n)$; 4. $(B[1], B[2], \dots, B[\ell]) \leftarrow \text{ENCODE}(T, D)$; 5. for $i = 1$ to ℓ; 6. $(X_l[i], X_r[i]) \xleftarrow{n} B[i]$; 7. $S_l[i] \leftarrow L_l \oplus X_l[i], S_r[i] \leftarrow L_r \oplus X_r[i]$; 8. $C_l[i] \leftarrow \tilde{\mathbf{E}}_{K_h}((001\ \text{Lsb}_{n-3}(S_r[i])), S_l[i])$; 9. $C_r[i] \leftarrow C_l[i] \oplus X_r[i]$; 10. $U \leftarrow 2U \oplus C_l[i]; V \leftarrow V \oplus C_r[i]$; 11. $(L_l, L_r) \leftarrow (2L_l, 2L_r)$; 12. $Y_1 \leftarrow \tilde{\mathbf{E}}_{K_h}((010\ \text{Lsb}_{n-3}(V)), U)$; 13. $Y_2 \leftarrow \tilde{\mathbf{E}}_{K_h}((011\ \text{Lsb}_{n-3}(U)), V)$; 14. return (Y_1, Y_2); 	<ol style="list-style-type: none"> 1. $((M_1\ M_2), M_R) \xleftarrow{2n} M$; 2. for $i = 1$ to 6 3. $K_i \leftarrow \tilde{\mathbf{E}}_K(10^{n-1}, \langle i \rangle_n)$; 4. $H_1\ H_2 \xleftarrow{n} \text{ZHASH}+(K_6, M_R, T)$; 5. $U_1 \leftarrow M_1 \oplus H_1$; 6. $U_2 \leftarrow M_2 \oplus H_2$; 7. $Z \leftarrow \tilde{\mathbf{E}}_{K_1}(U_2, U_1)$; 8. $W \leftarrow \tilde{\mathbf{E}}_{K_2}(Z, U_2)$; 9. $V_1 \leftarrow \tilde{\mathbf{E}}_{K_3}(W, Z)$; 10. $V_2 \leftarrow \tilde{\mathbf{E}}_{K_4}(V_1, W)$; 11. for $i = 1$ to $\lceil M_R /n \rceil$ 12. $S_i \leftarrow \tilde{\mathbf{E}}_{K_5}(Z \oplus \langle i \rangle_n, W)$; 13. $S \stackrel{\text{def}}{=} S_1\ \dots\ S_{\lceil M_R /n \rceil}$; 14. $C_R \leftarrow \text{Msb}_{ M_R }(S) \oplus M_R$; 15. $H'_1\ H'_2 \xleftarrow{n} \text{ZHASH}+(K_6, C_R, T)$; 16. $C_1 \leftarrow V_1 \oplus H'_1$; 17. $C_2 \leftarrow V_2 \oplus H'_2$; 18. return $(C_1\ C_2\ C_R)$;
<p>ENCODE(T, D)</p> <ol style="list-style-type: none"> 1. $Len \leftarrow \langle T \rangle_{n/2} \ \langle D \rangle_{n/2}$; 2. $B \leftarrow \text{Pad}_{2n}(T)\ \text{Pad}_{2n}(D)\ Len\ 0^n$; 3. $(B[1], B[2], \dots, B[\ell]) \xleftarrow{2n} \text{Parse}(B)$; 4. return $(B[1], B[2], \dots, B[\ell])$; 	

Figure 4: We describe the hash function ZHASH+ in the left hand side of the algorithm and ZHCTR+ construction is described in the right hand side of the algorithm. ENCODING function is described in the lower left of the algorithm.

function on $T\|D$ to derive ℓ many $2n$ -bit blocks $B[1], B[2], \dots, B[\ell]$. We also derive two masking values L_l and L_r by evaluating the tweakable block cipher $\tilde{\mathbf{E}}$ on input 0^n and tweak $000\|0^{n-3}$ (for generating L_l) and $000\|0^{n-4}\|1$ (for generating L_r). Then, we process each $2n$ -bit block $B[i]$ as follows: we split $B[i]$ into two n -bit blocks $X_l[i]$ and $X_r[i]$ and they are masked with L_l and L_r respectively to generate two n -bit values $S_l[i]$ and $S_r[i]$. Then, we evaluate a tweakable block cipher $\tilde{\mathbf{E}}$ with input $S_l[i]$ and the tweak is $001\|\text{Lsb}_{n-3}(S_r[i])$. Let the output be $C_l[i]$, which is masked with $X_r[i]$ to generate $C_r[i]$. Then we update two running variables U as $2U \oplus C_l[i]$ and V as $V \oplus C_r[i]$. We also update

L_l and L_r by doubling its old value. Finally, we compute the $2n$ bit hash output (Y_1, Y_2) , where Y_1 is obtained by evaluating the tweakable block cipher on input U with tweak $010\|\text{Lsb}_{n-3}(V)$ and Y_2 is obtained by evaluating the tweakable block cipher on input V with tweak $011\|\text{Lsb}_{n-3}(U)$.

ENCODING FUNCTION. The encoding function takes a variable length tweak T and a variable length message D and outputs a string $B \in (\{0, 1\}^n)^+$ as follows:

$$\text{ENCODE}(T, D) := \text{Pad}_{2n}(T) \parallel \text{Pad}_{2n}(D) \parallel (\langle |T| \rangle_{n/2} \parallel \langle |D| \rangle_{n/2}) \parallel 0^n.$$

We parse the resulting string B into ℓ many $2n$ -bit blocks. Since, we have encoded the length information of T and D , it is immediate to see that the encoding function is injective. An algorithmic description of the encoding function is given in Fig. 4.

To count the number of tweakable block cipher calls of ZHCTR+ construction, let L_M denote the length of the input message M in number of bits to the construction and L_T denotes the length of the tweak T in number of bits to the construction. In each of the two layers of the hash function, we require $\ell + 4$ many tweakable block cipher calls, where

$$\ell \leq \left(\frac{(L_M - 2n + n) + (L_T + n) + n + n}{2n} \right) = \frac{L_M + L_T + 2n}{2n}.$$

Therefore, in the ZHCTR+ construction, we require a total of

$$6 + 2(\ell + 4) + 4 + \left\lceil \frac{L_M - 2n}{n} \right\rceil \quad (14)$$

tweakable block cipher calls, where six tweakable block cipher calls are required to derive six keys, four tweakable block cipher calls are required in the left hand side of the construction and $\lceil \frac{L_M - 2n}{n} \rceil$ many tweakable block cipher calls are required in counter mode encryption. Therefore, by plugging-in the value of ℓ in Eqn. (14), the construction requires at most

$$\left(\frac{L_M + L_T}{n} + \left\lceil \frac{L_M - 2n}{n} \right\rceil + 20 \right)$$

tweakable block cipher calls.

Differences Between ZMAC and ZHASH+: The primary differences between the two constructions are the following: (i) in ZMAC, after processing ℓ many message blocks $(B[1], B[2], \dots, B[\ell])$, where each $B[i] \in \{0, 1\}^{n+t}$, the running variables (U, V) can be expressed as follows:

$$\begin{aligned} U &= 2^\ell C_l[1] \oplus 2^{\ell-1} C_l[2] \oplus \dots \oplus 2 C_l[\ell] \\ V &= C_r[1] \oplus C_r[2] \oplus \dots \oplus C_r[\ell] \end{aligned}$$

where $B[i] = (X_l[i] \parallel X_r[i])$, $X_l[i] \in \{0, 1\}^n$, $X_r[i] \in \{0, 1\}^t$, and $C_r[i] = \text{Msb}_t(C_l[i]) \oplus X_r[i]$ ⁷. On the other hand, in ZHASH+, the running variable (U, V) is expressed as follows:

$$\begin{aligned} U &= 2^{\ell-1} C_l[1] \oplus 2^{\ell-2} C_l[2] \oplus \dots \oplus C_l[\ell] \\ V &= C_r[1] \oplus C_r[2] \oplus \dots \oplus C_r[\ell] \end{aligned}$$

where $B[i] \in \{0, 1\}^{2n}$, $X_l[i], X_r[i] \in \{0, 1\}^n$, and $C_r[i] = C_l[i] \oplus X_r[i]$. Thus, we are saving one doubling operation in our construction compared to the ZMAC construction. (ii) In the finalization function of ZMAC, Y_1 and Y_2 are derived as follows:

$$Y_1 = \tilde{\text{E}}_K(0 \parallel V, U) \oplus \tilde{\text{E}}_K(1 \parallel V, U), Y_2 = \tilde{\text{E}}_K(2 \parallel V, U) \oplus \tilde{\text{E}}_K(3 \parallel V, U).$$

⁷We are assuming $t \leq n$. If $t > n$, then $C_l[i]$ is appropriately padded with 10^*

On the other hand, in ZHASH+, we derive Y_1 and Y_2 as follows:

$$Y_1 = \tilde{E}_K(010 \parallel \text{Lsb}_{n-3}(V), U), Y_2 = \tilde{E}_K(011 \parallel \text{Lsb}_{n-3}(U), V).$$

Thus, it saves two TBC calls in the ZHASH+ design. We would like to mention here that instead of considering (Y_1, Y_2) as the $2n$ -bit output of the hash function, we could have considered (U, V) as the output of the hash function ZHASH+. But then, as mentioned in [IMPS17], ZHASH+ cannot be proven to be an almost-xor-universal function. We would also like to make the following remark here: we could have used U as input and $\text{Lsb}_{n-3}(V)$ as the tweak in deriving both Y_1 and Y_2 values with appropriate domain separation. However, as we will show later that such choice would result in 2^{-n} almost-xor-universal advantage of the ZHASH+ function (please see Remark 3). In the following section, we bound the almost-xor-universal advantage of the ZHASH+ function.

5.4 Security Result of ZHCTR+

In this section, we state the security result of ZHCTR+ construction. Before that, we state the following result on the almost-xor-universal advantage of the ZHASH+ construction (in Lemma 5) and the almost-xor-universal advantage of each of its n -bit output (in Lemma 6).

Lemma 5. *Let \mathcal{K} be a non-empty finite set and $\tilde{E} : \mathcal{K} \times \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a tweakable block cipher. Let ZHASH+ be defined as above in Fig. 4. Then, for any two distinct messages M and M' such that the number of message blocks in M and M' be ℓ and ℓ' respectively, and for any $2n$ -bit string Δ , we have*

$$\Pr[K_h \leftarrow \mathcal{K} : \text{ZHASH+}[\tilde{E}_{K_h}](M) \oplus \text{ZHASH+}[\tilde{E}_{K_h}](M') = \Delta] \leq \frac{118}{(2^n - 1)^2} + \frac{1}{2^{2n}}$$

Lemma 6. *Let \mathcal{K} be a non-empty finite set and $\tilde{E} : \mathcal{K} \times \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a tweakable block cipher. Let Y_1 denote the leftmost n -bit output of ZHASH+ and Y_2 denote the rightmost n -bit output of ZHASH+. Then, for any n -bit string δ , and for $b \in \{1, 2\}$, we have*

$$\Pr[K_h \leftarrow \mathcal{K} : Y_b \oplus Y'_b = \delta] \leq \frac{63}{2^n - 1} + \frac{1}{2^{2n}}.$$

Proofs of the above two lemmas are deferred to Appendix A.3 and A.4. By combining Theorem 2, Lemma 5, and Lemma 6, we derive the security bound of ZHCTR+ construction as follows:

Theorem 4. *Let \mathcal{K} be a non-empty finite set. Let $\tilde{E} : \mathcal{K} \times \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be an (n, n) tweakable block cipher. Then, for any (q, ℓ_{\max}, τ) -chosen plaintext chosen ciphertext adaptive adversary \mathcal{A} against the stprp security of ZHCTR+ $[\tilde{E}]$ construction such that each query is of length at least $2n$ bits, there exists a (σ', τ') -chosen plaintext chosen ciphertext adversary \mathcal{A}' against the stprp security of \tilde{E} , such that*

$$\text{Adv}_{\text{ZHCTR+}[\tilde{E}]}^{\text{STPRP}}(\mathcal{A}) \leq 6\text{Adv}_{\tilde{E}}^{\text{STPRP}}(\mathcal{A}') + \frac{183q^2}{(2^n - 1)^2} + \frac{2q^2\ell_{\max}}{2^{2n}} + \frac{2q^2}{2^{2n}} + \frac{15}{2^n}, \quad (15)$$

where σ is the total number of message and tweak blocks queried, $\tau' = O(\tau + \sigma + 2qt_H)$, and t_H is the time for computing ZHASH+.

6 Implementation

HCTR+ employs a tweakable block cipher as its underlying primitive, with the tweak size matching the state size. Additionally, HCTR+ requires a $2n$ -bit hash function. As

discussed in Section 5, we have instantiated HCTR+ using PHASH+ and ZHASH+, which we refer to as PHCTR+ and ZHCTR+, respectively. These hash functions also rely on a tweakable block cipher as their underlying primitive, where the tweak size equals the block size. Various tweakable block ciphers can be used to instantiate our constructions, such as SKINNY-128-128 [BJK⁺16], QARMA-128-128 [Ava17], QARMAv2-128-128 [ABD⁺23] and Deoxys-BC-128-128 [JNPS16]. These tweakable block ciphers feature a 128-bit key, tweak, and state size.

Choice of TBC: In the concrete instantiations of PHCTR+ and ZHCTR+, we propose using Deoxys-BC-128-128 [JNPS16] with 128-bit key, tweak, and state size. Deoxys-BC-128-128 is a tweakable block cipher that adheres to the TWEAKEY framework [JNP14], where the key and tweak are combined into a single entity called a tweakey. The round function of Deoxys-BC-128-128 closely resembles that of AES, consisting of 14 rounds. Each round involves the application of AddRoundTweakey, SubBytes, ShiftRow, and MixColumn. Following the final round, an additional AddRoundTweakey operation is performed. Unlike AES, however, the final round in Deoxys-BC-128-128 also includes the MixColumn operation.

Each AddRoundTweakey operation requires a subtweakey, denoted as STK_i for the i -th round, generated by a specific tweakey schedule algorithm. The initial tweakey $TK = TK_0^0 || TK_0^1$ is formed by initializing the 128-bit key K and the 128-bit tweak T as $TK_0^0 = T$ and $TK_0^1 = K$. The tweakey schedule algorithm is defined as

$$TK_{i+1}^0 = H(TK_i^0) \text{ and } TK_{i+1}^1 = H(L(TK_i^1)), \text{ for } i = 0, \dots, 13.$$

Here H denotes the byte permutation defined as

$$\begin{bmatrix} 0 & 4 & 8 & 12 \\ 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 5 & 9 & 13 \\ 6 & 10 & 14 & 2 \\ 11 & 15 & 3 & 7 \\ 12 & 0 & 4 & 8 \end{bmatrix}$$

and L denotes the application of an LFSR to each of the 16 bytes of a 128-bit where the LFSR is defined as

$$(x_7 || x_6 || x_5 || x_4 || x_3 || x_2 || x_1 || x_0) \rightarrow (x_6 || x_5 || x_4 || x_3 || x_2 || x_1 || x_0 || (x_7 \oplus x_5)).$$

Finally, the subtweakey for the i -th round is computed as

$$STK_i = TK_i^0 \oplus TK_i^1 \oplus RC_i \text{ for } i = 0, \dots, 14,$$

where RC_i is the i -th round constant with the following form

$$RC_i = \begin{bmatrix} 1 & rc_i & 0 & 0 \\ 2 & rc_i & 0 & 0 \\ 4 & rc_i & 0 & 0 \\ 8 & rc_i & 0 & 0 \end{bmatrix}$$

and the value rc_i denotes the i -th key schedule constants of the AES.

One notable property of HCTR+ construction is that the key used in the tweakable block cipher is derived once and the same derived keys are used throughout the computation of the cipher. Meanwhile, the tweaks change in each call to the block cipher. Thus, it is evident that for better performance, we need to choose a tweakable block cipher where the tweak and key schedule perform separately, which is the case in Deoxys. Furthermore, the tweakey schedule in Deoxys is lightweight. Another reason to use Deoxys-BC-128-128 is the use of AES-NI instruction set. Using AES-NI instructions, Deoxys-BC-128-128 achieves good performance in the software implementation.

Software Performance: Here we discuss the software performance of the C implementation of PHCTR+ and ZHCTR+ using Deoxys-BC-128-128 as the underlying TBC. The performance is benchmarked in an Intel(R) Core(TM) i7-1165G7 machine with 2.80GHz CPU, 32 GB DDR5 RAM, and supporting the AES-NI instructions. All the experiments are done with Intel TurboBoost and HyperThreading disabled. The compilation is done using GCC 12.2.0 with `-O3` optimization and `-march` set. The machine runs on Debian 12.6 with Linux kernel 6.1.0. To capture the average cost of encryption, each experiment is repeated 128 times. To reduce the influence of memory access, the cache is warmed before the time begins.

We evaluate the performance of PHCTR+ and ZHCTR+, comparing them with the only existing n -bit secure sprp, ZCZ, which utilizes Deoxys-BC-128-256 [JNPS16] as the underlying TBC. The performance results, measured in Cycles Per Byte (CPB), are presented in Table 2. To calculate CPB, the total number of CPU cycles required to encrypt a message with a given tweak is divided by the message’s length and the tweak’s length. These measurements are performed for various message and tweak lengths, as detailed in Table 2, where the cost for key setup is excluded. The source codes of the implementation are available at <https://github.com/ShibamCrS/HCTR-.git>.

Table 2: Software performance (CPB) of PHCTR+ and ZHCTR+, and comparison with ZCZ. Note that ZCZ does not accept any tweak, so the comparison of ZCZ should be with zero length tweak of PHCTR+ and ZHCTR+ (blue colored columns).

Name	PHCTR+						ZHCTR+						ZCZ	
	Tweak Length (Bytes)													
Message Length (Bytes)	0B	16B	32B	64B	128B	256B	0B	16B	32B	64B	128B	256B	-	
	128B	5.99	5.52	5.09	4.67	3.87	2.93	5.25	5.04	4.43	3.82	3.03	2.23	14.10
	256B	4.71	4.56	4.34	4.29	3.87	3.26	4.06	4.00	3.73	3.47	3.05	2.45	11.14
	512B	3.82	3.79	3.73	3.76	3.56	3.29	3.18	3.14	3.05	2.95	2.75	2.44	8.31
	1KB	3.42	3.40	3.37	3.40	3.30	3.18	2.68	2.68	2.64	2.60	2.50	2.35	6.77
	2KB	3.20	3.20	3.19	3.20	3.17	3.10	2.44	2.44	2.42	2.41	2.36	2.28	5.97
	4KB	3.11	3.10	3.10	3.11	3.10	3.06	2.31	2.32	2.31	2.31	2.28	2.24	5.42
	8KB	3.06	3.06	3.05	3.06	3.05	3.04	2.26	2.26	2.26	2.25	2.25	2.23	5.16
	16KB	3.04	3.03	3.03	3.03	3.03	3.03	2.24	2.24	2.23	2.23	2.23	2.21	4.84
	32KB	3.02	3.02	3.02	3.02	3.02	3.01	2.22	2.22	2.22	2.22	2.22	2.21	4.57
64KB	2.90	2.89	2.89	2.89	2.89	2.89	2.21	2.21	2.21	2.21	2.21	2.21	4.44	

Note that, ZCZ is not a tweakable cipher. Thus, the comparison is fair when the tweak length of PHCTR+ and ZHCTR+ is zero. To highlight the comparison with ZCZ, we marked the columns corresponding to zero tweaks with blue color in Table 2. This is evident from the results in Table 2, PHCTR+ and ZHCTR+ are about 3 times better in terms of software performance than ZCZ despite that ZCZ requires 1.5 TBC calls per message, whereas our proposal requires 3 TBC calls per message. The key to improving our scheme’s performance in practice lies in the tweak-size of the TBC. While ZCZ requires a larger tweak-size than the state size to support an additional domain and block-counter, our construction uses a TBC where the tweak size matches the state size. Specifically, we instantiate Deoxys-BC-128-128 as the underlying primitive, whereas ZCZ used Deoxys-BC-384 (equivalent to Deoxys-BC-128-256 in the original Deoxys paper [JNPS16], with a 128-bit key and 256-bit tweak). Notably, Deoxys-BC-128-128 has 14-rounds, compared to 16-rounds in Deoxys-BC-128-256, and involves fewer tweakey schedule operations per round, reducing one shuffle and one LFSR call per round, which enhances performance.

Remark 2. We would like to point out here that VAESENC instruction significantly enhances the performance of our constructions. We observed nearly a twofold improvement in the performance of PHCTR+. However, we chose not to leverage these performance gains to keep the implementation requirements minimal. Additionally, since ZCZ does not utilize VAESENC and our comparisons are based on the author’s implementation of ZCZ, it would not be a fair comparison if we used VAESENC for our benchmarks.

7 Conclusion

In this paper, we have proposed HCTR+, a single-keyed, optimally secure accordion mode based on an n -bit tweakable block cipher with an n -bit tweak and a $2n$ -bit keyed hash function. The strength of our construction lies in its ability to ensure optimal security even in the presence of arbitrary tweak repetitions. This robustness against tweak misuse provides a significant advancement in the field of tweakable enciphering schemes. We have instantiated the underlying hash function of our mode with PHASH+ and ZHASH+ and the underlying TBC with Deoxys-BC-128-128. Software implementations of the resulting two constructions, PHCTR+ and ZHCTR+ indicate a significant performance improvement compared to the ZCZ construction, the only existing n -bit secure sprp construction with no tweak. It remains open to analyze the multi user security of HCTR+ and transform it to a context committing AE scheme.

References

- [ABD⁺23] Roberto Avanzi, Subhadeep Banik, Orr Dunkelman, Maria Eichseder, Shibam Ghosh, Marcel Nageler, and Francesco Regazzoni. The qarmav2 family of tweakable block ciphers. *IACR Trans. Symmetric Cryptol.*, 2023(3):25–73, 2023.
- [ABPV21] Elena Andreeva, Amit Singh Bhati, Bart Preneel, and Damian Vizár. 1, 2, 3, fork: Counter mode variants based on a generalized forkcipher. *IACR Trans. Symmetric Cryptol.*, 2021(3):1–35, 2021.
- [Ava17] Roberto Avanzi. The QARMA block cipher family. almost MDS matrices over rings with zero divisors, nearly symmetric even-mansour constructions with non-involutory central rounds, and search heuristics for low-latency s-boxes. *IACR Trans. Symmetric Cryptol.*, 2017(1):4–44, 2017.
- [BDH⁺17] 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.
- [BJK⁺16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In *CRYPTO (2)*, volume 9815 of *Lecture Notes in Computer Science*, pages 123–153. Springer, 2016.
- [BLN18] Ritam Bhaumik, Eik List, and Mridul Nandi. ZCZ - achieving n -bit SPRP security with a minimal number of tweakable-block-cipher calls. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*, volume 11272 of *Lecture Notes in Computer Science*, pages 336–366. Springer, 2018.
- [BN15] 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.

- [CB18] Paul Crowley and Eric Biggers. Adiantum: length-preserving encryption for entry-level processors. *IACR Trans. Symmetric Cryptol.*, 2018(4):39–61, 2018.
- [CDMS10] Jean-Sébastien Coron, Yevgeniy Dodis, Avradip Mandal, and Yannick Seurin. A domain extender for the ideal cipher. In Daniele Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, volume 5978 of *Lecture Notes in Computer Science*, pages 273–289. Springer, 2010.
- [CEL⁺21] Benoît Cogliati, Jordan Ethan, Virginie Lallemand, ByeongHak Lee, Jooyoung Lee, and Marine Minier. CTET+: A beyond-birthday-bound secure tweakable enciphering scheme using a single pseudorandom permutation. *IACR Trans. Symmetric Cryptol.*, 2021(4):1–35, 2021.
- [CGLS22] Debrup Chakraborty, Sebati Ghosh, Cuauhtemoc Mancillas López, and Palash Sarkar. \mathcal{F} : Disk encryption and beyond. *Adv. Math. Commun.*, 16(1):185–230, 2022.
- [CHB21] Paul Crowley, Nathan Huckleberry, and Eric Biggers. Length-preserving encryption with HCTR2. *IACR Cryptol. ePrint Arch.*, page 1441, 2021.
- [CN08a] Debrup Chakraborty and Mridul Nandi. An improved security bound for HCTR. In Kaisa Nyberg, editor, *Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers*, volume 5086 of *Lecture Notes in Computer Science*, pages 289–302. Springer, 2008.
- [CN08b] Donghoon Chang and Mridul Nandi. A short proof of the PRP/PRF switching lemma. *IACR Cryptol. ePrint Arch.*, page 78, 2008.
- [CS06] Debrup Chakraborty and Palash Sarkar. A new mode of encryption providing a tweakable strong pseudo-random permutation. In Matthew J. B. Robshaw, editor, *Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers*, volume 4047 of *Lecture Notes in Computer Science*, pages 293–309. Springer, 2006.
- [CS08] Debrup Chakraborty and Palash Sarkar. HCH: A new tweakable enciphering scheme using the hash-counter-hash approach. *IEEE Trans. Inf. Theory*, 54(4):1683–1699, 2008.
- [DFUB24] Hieu Nguyen Duy2, Pablo Garcia Fernandez, Aleksei Udovenko, and Alex Biryukov. Accordion mode based on hash-encrypt-hash, 2024. Online: <https://csrc.nist.gov/csrc/media/Presentations/2024/accordion-mode-based-on-hash-encrypt-hash/images-media/sess-7-fernandez-acm-workshop-2024.pdf>.
- [DMMT24] Christoph Dobraunig, Krystian Matusiewicz, Bart Mennink, and Alexander Tereschenko. Efficient instances of docked double decker with AES. *IACR Cryptol. ePrint Arch.*, page 84, 2024.
- [DN18] Avijit Dutta and Mridul Nandi. Tweakable HCTR: A BBB secure tweakable enciphering scheme. In Debrup Chakraborty and Tetsu Iwata, editors, *Progress in Cryptology - INDOCRYPT 2018 - 19th International Conference on Cryptology in India, New Delhi, India, December 9-12, 2018, Proceedings*, volume 11356 of *Lecture Notes in Computer Science*, pages 47–69. Springer, 2018.

- [GDM22] Aldo Gunesing, Joan Daemen, and Bart Mennink. Deck-based wide block cipher modes and an exposition of the blinded keyed hashing model. *IACR Cryptol. ePrint Arch.*, page 247, 2022.
- [Hal04] Shai Halevi. Eme^{*}: Extending EME to handle arbitrary-length messages with associated data. In Anne Canteaut and Kapalee Viswanathan, editors, *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*, volume 3348 of *Lecture Notes in Computer Science*, pages 315–327. Springer, 2004.
- [Hal07] Shai Halevi. Invertible universal hashing and the TET encryption mode. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, volume 4622 of *Lecture Notes in Computer Science*, pages 412–429. Springer, 2007.
- [HR03] Shai Halevi and Phillip Rogaway. A tweakable enciphering mode. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 482–499. Springer, 2003.
- [HR04] Shai Halevi and Phillip Rogaway. A parallelizable enciphering mode. In Tatsuaki Okamoto, editor, *Topics in Cryptology - CT-RSA 2004, The Cryptographers’ Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, volume 2964 of *Lecture Notes in Computer Science*, pages 292–304. Springer, 2004.
- [IMPS17] Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A fast tweakable block cipher mode for highly secure message authentication. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 34–65. Springer, 2017.
- [Jiv14] Andrey Jivsov. WCFB: A wide block encryption for large data sets. In David Garcia Rosado, Carlos Blanco, Daniel Mellado, Jan Jürjens, and Luis Enrique Sánchez, editors, *WOSIS 2014 - Proceedings of the 11th International Workshop on Security in Information Systems, Lisbon, Portugal, 27 April, 2014*, pages 75–82. SciTePress, 2014.
- [JNP14] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and keys for block ciphers: The TWEAKEY framework. In *ASIACRYPT (2)*, volume 8874 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2014.
- [JNPS16] Jérémy Jean, Ivica Nikolic, Thomas Peyrin, and Yannick Seurin. Deoxys v1. 41, 2016. Online: <https://competitions.cr.yyp.to/round3/deoxysv141.pdf>.
- [Kha24] Mustafa Khairallah. A note on -tweakable HCTR: A BBB secure tweakable enciphering scheme-. *IACR Cryptol. ePrint Arch.*, page 600, 2024.
- [Kum18] Manish Kumar. Security of xcb and hctr, 2018. Online: <http://library.isical.ac.in:8080/jspui/handle/10263/6953>.

- [Lee24] Byeonghak Lee. A bbb secure accordion mode from hctr, 2024. Online: <https://csrc.nist.gov/Presentations/2024/a-bbb-secure-accordion-mode-from-hctr>.
- [LN16] Eik List and Mridul Nandi. Revisiting full-PRF-secure PMAC and using it for beyond-birthday authenticated encryption. *Cryptology ePrint Archive*, Paper 2016/1174, 2016.
- [LN17a] Eik List and Mridul Nandi. Revisiting full-prf-secure PMAC and using it for beyond-birthday authenticated encryption. In Helena Handschuh, editor, *Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14-17, 2017, Proceedings*, volume 10159 of *Lecture Notes in Computer Science*, pages 258–274. Springer, 2017.
- [LN17b] Eik List and Mridul Nandi. Revisiting full-prf-secure PMAC and using it for beyond-birthday authenticated encryption. In Helena Handschuh, editor, *Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14-17, 2017, Proceedings*, volume 10159 of *Lecture Notes in Computer Science*, pages 258–274. Springer, 2017.
- [MCR07] Cuauhtemoc Mancillas-López, Debrup Chakraborty, and Francisco Rodríguez-Henríquez. Efficient implementations of some tweakable enciphering schemes in reconfigurable hardware. In K. Srinathan, C. Pandu Rangan, and Moti Yung, editors, *Progress in Cryptology - INDOCRYPT 2007, 8th International Conference on Cryptology in India, Chennai, India, December 9-13, 2007, Proceedings*, volume 4859 of *Lecture Notes in Computer Science*, pages 414–424. Springer, 2007.
- [MF04] David A. McGrew and Scott R. Fluhrer. The extended codebook (XCB) mode of operation. *IACR Cryptol. ePrint Arch.*, page 278, 2004.
- [MI11] Kazuhiko Minematsu and Tetsu Iwata. Building blockcipher from tweakable blockcipher: Extending FSE 2009 proposal. In Liqun Chen, editor, *Cryptography and Coding - 13th IMA International Conference, IMACC 2011, Oxford, UK, December 12-15, 2011. Proceedings*, volume 7089 of *Lecture Notes in Computer Science*, pages 391–412. Springer, 2011.
- [MI17] Kazuhiko Minematsu and Tetsu Iwata. Cryptanalysis of pmacx, pmac2x, and sixv. *IACR Trans. Symmetric Cryptol.*, 2017(2):162–176, 2017.
- [NR97] Moni Naor and Omer Reingold. On the construction of pseudo-random permutations: Luby-rackoff revisited (extended abstract). In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 189–199. ACM, 1997.
- [NR99a] Moni Naor and Omer Reingold. On the construction of pseudorandom permutations: Luby-rackoff revisited. *J. Cryptol.*, 12(1):29–66, 1999.
- [NR99b] Moni Naor and Omer Reingold. A pseudo-random encryption mode, 1999. Online: <https://www.wisdom.weizmann.ac.il/~lnaor/>.
- [NSS24] Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Committing wide encryption mode with minimum ciphertext expansion, 2024. Online: <https://csrc.nist.gov/csrc/media/>

[Presentations/2024/committing-wide-encryption-mode/images-media/sess-8-naito-acm-workshop-2024.pdf](#).

- [Pat08] Jacques Patarin. The "coefficients h" technique. In *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, pages 328–345, 2008.
- [PS16] Thomas Peyrin and Yannick Seurin. Counter-in-tweak: Authenticated encryption modes for tweakable block ciphers. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 33–63. Springer, 2016.
- [Sar07] Palash Sarkar. Improving upon the TET mode of operation. In Kil-Hyun Nam and Gwangsoo Rhee, editors, *Information Security and Cryptology - ICISC 2007, 10th International Conference, Seoul, Korea, November 29-30, 2007, Proceedings*, volume 4817 of *Lecture Notes in Computer Science*, pages 180–192. Springer, 2007.
- [ST13] Thomas Shrimpton and R. Seth Terashima. A modular framework for building variable-input-length tweakable ciphers. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 405–423. Springer, 2013.
- [WFW05] Peng Wang, Dengguo Feng, and Wenling Wu. HCTR: A variable-input-length enciphering mode. In Dengguo Feng, Dongdai Lin, and Moti Yung, editors, *Information Security and Cryptology, First SKLOIS Conference, CISC 2005, Beijing, China, December 15-17, 2005, Proceedings*, volume 3822 of *Lecture Notes in Computer Science*, pages 175–188. Springer, 2005.

Appendix

A Proofs of AXU advantages for PHASH+ and ZHASH+

In this section, we will prove Lemma 3, Lemma 4, Lemma 5 and Lemma 6, which establish bounds on the AXU advantages of PHASH+ and ZHASH+ as well as the AXU advantages for each of their n -bit outputs.

A.1 Proof of Lemma 3

Let Δ be a fixed $2n$ bit string which is parsed as two n -bit strings Δ_1 and Δ_2 , i.e., $\Delta = \Delta_1 \parallel \Delta_2$, and $\Delta_1, \Delta_2 \in \{0, 1\}^n$. Let E denotes the event that the xor of hash output of PHASH+ on two distinct messages M and M' attains the value Δ . i.e.,

$$\Pr[E] = \Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2],$$

where the randomness of the above probability is defined over U, U', V , and V' . From the description of PHASH+ construction,

$$U \leftarrow \tilde{E}_{K_n}(010 \parallel \text{Lsb}_{n-3}(Y[\ell]), X[\ell]), \quad V \leftarrow \tilde{E}_{K_n}(011 \parallel \text{Lsb}_{n-3}(X[\ell]), Y[\ell])$$

(see line 7 and line 8 of Fig. 3). Since the domain separation string used in the tweak part of deriving U and V are different, it follows that U and V are two independently sampled n -bit strings. Let us denote $\text{Lsb}_{n-3}(X[\ell])$ as $\widehat{X}[\ell]$ and $\text{Lsb}_{n-3}(Y[\ell])$ as $\widehat{Y}[\ell]$. Now, to bound the probability of the event \mathbf{E} holds, we have the following two cases:

Case-I: ($X[\ell] = X'[\ell']$ and $Y[\ell] = Y'[\ell']$). This event implies $\widehat{X}_\ell = \widehat{X}_{\ell'}$ and $\widehat{Y}_\ell = \widehat{Y}_{\ell'}$ and hence it implies $U = U'$ and $V = V'$. The analysis mostly follows the analysis of [LN16, Theorem 1, Case 1]. We revisit to the analysis for the sake of completeness. Note that, the condition $X[\ell] = X'[\ell']$ and $Y[\ell] = Y'[\ell']$ implies the following two equations:

$$\begin{cases} (Z[1] \oplus Z[2] \oplus \dots \oplus Z[\ell]) \oplus (Z'[1] \oplus Z'[2] \oplus \dots \oplus Z'[\ell']) = 0^n \\ (2^{\ell-1}Z[1] \oplus 2^{\ell-2}Z[2] \oplus \dots \oplus Z[\ell]) \oplus (2^{\ell'-1}Z'[1] \oplus 2^{\ell'-2}Z'[2] \oplus \dots \oplus Z'[\ell']) = 0^n. \end{cases}$$

Now, if either of the Δ_1 or Δ_2 is a non-zero n -bit string, then the probability of the event is zero. Thus, if $\Delta = 0^{2n}$, then we bound the probability of the above system of equations hold in several cases as follows:

Case-A: Let $\ell = \ell'$ and $\exists h \in \{1, \dots, \ell\}$ such that $B[h] \neq B'[h]$ and $B[s] = B'[s]$ for all $s \neq h$. Therefore,

$$Y[\ell] \oplus Y'[\ell] \Rightarrow 2^{\ell-h}(Z[h] \oplus Z'[h]), X[\ell] \oplus X'[\ell] \Rightarrow Z[h] \oplus Z'[h].$$

To imply that $Y[\ell] = Y'[\ell]$ and $X[\ell] = X'[\ell]$, it must have to be the case that $Z[h] = Z'[h]$. But then it means that $B[h] = B'[h]$, which contradicts to the assumption that $B[h] \neq B'[h]$. Therefore, in this case, the probability of the event $Y[\ell] = Y'[\ell]$ and $X[\ell] = X'[\ell]$ is zero.

Case-B: Let $\ell = \ell'$ and $\exists h \neq s \in \{1, \dots, \ell\}$ such that $B[h] \neq B'[h]$ and $B[s] \neq B'[s]$. Therefore,

$$\begin{aligned} Y[\ell] \oplus Y'[\ell] &\Rightarrow 2^{\ell-h}(Z[h] \oplus Z'[h]) \oplus 2^{\ell-s}(Z[s] \oplus Z'[s]) \oplus \underbrace{\bigoplus_{i \neq h, s} 2^{\ell-i}(Z[i] \oplus Z'[i])}_{\Sigma}, \\ X[\ell] \oplus X'[\ell] &\Rightarrow (Z[h] \oplus Z'[h]) \oplus (Z[s] \oplus Z'[s]) \oplus \underbrace{\bigoplus_{i \neq h, s} (Z[i] \oplus Z'[i])}_{\Theta}. \end{aligned}$$

Note that each term of Σ and Θ are independent. Let $\lambda_h = 2^{\ell-h}$ and $\lambda_s = 2^{\ell-s}$. Then, $Y[\ell] = Y'[\ell], X[\ell] = X'[\ell]$ implies the following system of equations hold:

$$\begin{cases} \lambda_h \Delta Z[h] \oplus \lambda_s \Delta Z[s] = \Sigma \\ \Delta Z[h] \oplus \Delta Z[s] = \Theta \end{cases}$$

Note that the above system of equations has rank 2, and hence, it has a unique solution, namely

$$\begin{cases} \Delta Z[h] = (\lambda_s \Theta \oplus \Sigma) / (\lambda_h \oplus \lambda_s) \\ \Delta Z[s] = (\lambda_h \Theta \oplus \Sigma) / (\lambda_h \oplus \lambda_s) \end{cases}$$

Note that, since $\Delta Z[h]$ and $\Delta Z[s]$ for $h \neq s$ are independently distributed as different tweaks are employed and they are uniformly distributed over $\{0, 1\}^n$, it implies that

$$\Pr[\Delta Y[\ell] = 0^n, \Delta X[\ell] = 0^n] \leq \frac{1}{2^{2n}}.$$

Case-C: Let $\ell' = \ell + 1$. Then, we have

$$\begin{aligned}\Delta Y[\ell] &= \bigoplus_{1 \leq i \leq \ell} 2^{\ell-i} Z[i] \oplus \bigoplus_{1 \leq i \leq \ell'} 2^{\ell'-i} Z'[i] \\ &= Z[\ell] \oplus 2Z'[\ell] \oplus Z'[\ell+1] \oplus \bigoplus_{i \neq \ell, \ell'} 2^{\ell-i} \Delta Z[i] \\ \Delta X[\ell] &= Z[\ell] \oplus Z'[\ell] \oplus Z'[\ell+1] \oplus \bigoplus_{i \neq \ell, \ell'} (\Delta Z[i])\end{aligned}$$

Then $(\Delta X[\ell], \Delta Y[\ell]) = (0^n, 0^n)$ implies the following system of equations hold:

$$\begin{cases} Z[\ell] \oplus 2Z'[\ell] \oplus Z'[\ell+1] = \bigoplus_{i \neq \ell, \ell'} 2^{\ell-i} \Delta Z[i] \\ Z[\ell] \oplus Z'[\ell] \oplus Z'[\ell+1] = \bigoplus_{i \neq \ell, \ell'} (\Delta Z[i]) \end{cases}$$

Note that the above system of equations has rank 2, and hence, it has a unique solution. Moreover, as $(Z[\ell] \oplus Z'[\ell+1])$ and $Z'[\ell]$ are uniformly distributed over $\{0, 1\}^n$ and they are independent, it implies that

$$\Pr[\Delta X[\ell] = 0^n, \Delta Y[\ell] = 0^n] \leq \frac{1}{2^{2n}}.$$

Case-D: Let $\ell' \geq \ell + 2$. Then, we have

$$\begin{aligned}\Delta Y[\ell] &= \bigoplus_{1 \leq i \leq \ell} 2^{\ell-i} Z[i] \oplus \bigoplus_{1 \leq i \leq \ell'} 2^{\ell'-i} Z'[i] \\ &= 2Z'[\ell'-1] \oplus Z'[\ell'] \oplus \bigoplus_{1 \leq i \leq \ell} 2^{\ell-i} Z[i] \oplus \bigoplus_{1 \leq i \leq \ell'-2} 2^{\ell'-i} Z'[i] \\ \Delta X[\ell] &= Z'[\ell'-1] \oplus Z'[\ell'] \oplus \bigoplus_{1 \leq i \leq \ell} Z[i] \oplus \bigoplus_{1 \leq i \leq \ell'-2} Z'[i]\end{aligned}$$

Then $(\Delta X[\ell], \Delta Y[\ell]) = (0^n, 0^n)$ implies the following system of equations hold:

$$\begin{cases} 2Z'[\ell'-1] \oplus Z'[\ell'] = \bigoplus_{1 \leq i \leq \ell} 2^{\ell-i} Z[i] \oplus \bigoplus_{1 \leq i \leq \ell'-2} 2^{\ell'-i} Z'[i] \\ Z'[\ell'-1] \oplus Z'[\ell'] = \bigoplus_{1 \leq i \leq \ell} Z[i] \oplus \bigoplus_{1 \leq i \leq \ell'-2} Z'[i] \end{cases}$$

Note that the above system of equations has rank 2, and hence, it has a unique solution. Moreover, as $(Z'[\ell'] \oplus Z'[\ell'-1])$ are independent and uniformly distributed over $\{0, 1\}^n$, it implies that

$$\Pr[\Delta X[\ell] = 0^n, \Delta Y[\ell] = 0^n] \leq \frac{1}{2^{2n}}.$$

Therefore, by combining the above four cases, we derive the upper bound on the probability of the event E holds in Case-I as follows:

$$\Pr[\mathbf{E}] = \Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \leq \frac{1}{2^{2n}}. \quad (16)$$

Case-II: $\neg(X[\ell] = X'[\ell'] \text{ and } Y[\ell] = Y'[\ell'])$. In this case the followings can be possible:

- **Case-A:** $(X[\ell] = X'[\ell'] \text{ and } Y[\ell] \neq Y'[\ell'])$. Bounding the probability of the event E holds under this case again gives rise to the following two subcases: (a) if $\text{Lsb}_{n-3}(Y[\ell]) = \text{Lsb}_{n-3}(Y'[\ell'])$, then $U = U'$. However, since $Y[\ell] \neq Y'[\ell']$, it implies that $V \neq V'$. Therefore, for a fixed non-zero n -bit string Δ_2 , the event

$V \oplus V' = \Delta_2$ holds with probability at most $1/(2^n - 1)$. Moreover, the event $X[\ell] = X'[\ell]$ and $\text{Lsb}_{n-3}(Y[\ell]) = \text{Lsb}_{n-3}(Y'[\ell])$ implies the following two equations:

$$\begin{cases} (Z[1] \oplus Z[2] \oplus \dots \oplus Z[\ell]) \oplus (Z'[1] \oplus Z'[2] \oplus \dots \oplus Z'[\ell]) = 0^n \\ \text{Lsb}_{n-3}(2^{\ell-1}Z[1] \oplus 2^{\ell-2}Z[2] \oplus \dots \oplus Z[\ell]) \\ \oplus \text{Lsb}_{n-3}(2^{\ell-1}Z'[1] \oplus 2^{\ell-2}Z'[2] \oplus \dots \oplus Z'[\ell]) = 0^n. \end{cases}$$

Since the rank of the above system of equations is at least 1, the probability that the above system of equations holds is at most $2^3(2^3 - 1)/2^n$ as each assignment in the most significant three bits of $Y[\ell]$, there are exactly 7 assignments in the most significant three bits of $Y'[\ell]$ that eventually ensures $Y[\ell] \neq Y'[\ell]$. Therefore, we have,

$$\Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \leq \begin{cases} \frac{56}{2^n(2^n-1)} & \text{if } \Delta_1 = 0^n \text{ and } \Delta_2 \neq 0^n \\ 0 & \text{if } \Delta_1 \neq 0^n \text{ or } \Delta_2 = 0^n. \end{cases}$$

(b) If $\text{Lsb}_{n-3}(Y[\ell]) \neq \text{Lsb}_{n-3}(Y'[\ell])$, then U and U' are independently distributed. However, since $Y[\ell] \neq Y'[\ell]$, it implies that $V \neq V'$ and they are not independently distributed. Therefore, for a fixed non-zero n -bit string Δ_2 and for any n -bit string Δ_1 , the event $U \oplus U' = \Delta_1, V \oplus V' = \Delta_2$ holds with probability at most $1/2^n(2^n - 1)$. On the other hand, if $\Delta_2 = 0^n$, then the probability of the event becomes zero, i.e.,

$$\Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \leq \begin{cases} \frac{1}{2^n(2^n-1)} & \text{if } \Delta_2 \neq 0^n \\ 0 & \text{if } \Delta_2 = 0^n. \end{cases}$$

Therefore, by combining the two subcases, we derive the upper bound on the probability of the event **E** in Case-A as follows:

$$\Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \leq \frac{57}{2^n(2^n - 1)}. \quad (17)$$

- **Case-B:** ($Y[\ell] = Y'[\ell]$ and $X[\ell] \neq X'[\ell]$). This case is symmetrical to Case-A (by just swapping the role between U and V , and that of between U' and V'). Thus, we omit the details of the analysis and conclude that

$$\Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \leq \frac{57}{2^n(2^n - 1)}. \quad (18)$$

- **Case-C:** ($X[\ell] \neq X'[\ell]$ and $Y[\ell] \neq Y'[\ell]$). Bounding the probability of the event **E** holds under this case again gives rise to the following four subcases: (a) if $\text{Lsb}_{n-3}(X[\ell]) = \text{Lsb}_{n-3}(X'[\ell])$ and $\text{Lsb}_{n-3}(Y[\ell]) = \text{Lsb}_{n-3}(Y'[\ell])$, then it follows that U and U' are not two independent random variables. Similarly, V and V' are not two independent random variables. However, since $X[\ell] \neq X'[\ell]$ and $Y[\ell] \neq Y'[\ell]$, it implies that $U \neq U'$ and $V \neq V'$. Therefore, for a fixed $2n$ -bit string $\Delta = \Delta_1 \parallel \Delta_2$ such that $\Delta_1 \neq 0^n$ and $\Delta_2 \neq 0^n$, the probability of the event

$$U \oplus U' = \Delta_1, V \oplus V' = \Delta_2$$

is bounded by $1/(2^n - 1)^2$ due to the randomness U and V . Otherwise, the probability of the event would have been zero, i.e.,

$$\Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \leq \begin{cases} \frac{1}{(2^n-1)^2} & \text{if } \Delta_1 \neq 0^n \text{ and } \Delta_2 \neq 0^n \\ 0 & \text{otherwise.} \end{cases}$$

(b) If $\text{Lsb}_{n-3}(X[\ell]) = \text{Lsb}_{n-3}(X'[\ell'])$ and $\text{Lsb}_{n-3}(Y[\ell]) \neq \text{Lsb}_{n-3}(Y'[\ell'])$, then V and V' are distinct and they are not independent random variables. However, since $\text{Lsb}_{n-3}Y[\ell] \neq \text{Lsb}_{n-3}(Y'[\ell'])$, it implies that U and U' are two independent random variables. Thus, for a fixed $2n$ bit string Δ such that $\Delta = \Delta_1 \parallel \Delta_2$ with $\Delta_2 \neq 0^n$, we have

$$\Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \leq \frac{1}{2^n(2^n - 1)}.$$

(c) If $\text{Lsb}_{n-3}(X[\ell]) \neq \text{Lsb}_{n-3}(X'[\ell'])$ and $\text{Lsb}_{n-3}(Y[\ell]) = \text{Lsb}_{n-3}(Y'[\ell'])$, then U and U' are distinct and they are not independent random variables. However, since $\text{Lsb}_{n-3}(X[\ell]) \neq \text{Lsb}_{n-3}(X'[\ell'])$, it implies that V and V' are two independent random variables. Thus, for a fixed non-zero $2n$ bit string Δ such that $\Delta = \Delta_1 \parallel \Delta_2$ with $\Delta_1 \neq 0^n$, we have

$$\Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \leq \frac{1}{2^n(2^n - 1)}.$$

(d) If both $\text{Lsb}_{n-3}(X[\ell]) \neq \text{Lsb}_{n-3}(X'[\ell'])$ and $\text{Lsb}_{n-3}(Y[\ell]) \neq \text{Lsb}_{n-3}(Y'[\ell'])$, then V and V' are two independent random variables and U and U' are two independent random variables. Thus, for a fixed $\Delta_1 \parallel \Delta_2$, we have

$$\Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \leq \frac{1}{2^{2n}}.$$

Therefore, by combining the above four subcases, we derive the upper bound on the probability of the event \mathbf{E} in Case C as follows:

$$\Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \leq \frac{4}{(2^n - 1)^2}. \quad (19)$$

Now, by combining Eqn. (17), Eqn. (18), and Eqn. (19), we derive the upper bound on the probability of the event \mathbf{E} in Case-II as follows:

$$\Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \leq \frac{118}{(2^n - 1)^2}. \quad (20)$$

Finally, by combining Eqn. (16) and Eqn. (20), we derive the upper bound on the probability of the event \mathbf{E} as

$$\Pr[\mathbf{E}] = \Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \leq \frac{118}{(2^n - 1)^2} + \frac{1}{2^{2n}}. \quad (21)$$

A.2 Proof of Lemma 4

We bound the almost-xor-universal advantage of each of the n -bit output of PHASH+ in a similar way as we did in the proof of Lemma 3. Let \mathbf{E}_1 and \mathbf{E}_2 denote the events $U \oplus U' = \delta$ and $V \oplus V' = \delta$ respectively. As before, the analysis of upper bounding the probability of the events \mathbf{E}_1 and \mathbf{E}_2 are based on the following two cases:

Case-I: ($X[\ell] = X'[\ell']$ and $Y[\ell] = Y'[\ell']$). In this case $U = U'$. Therefore, for a fixed non-zero n -bit string δ , the probability of the event $U \oplus U' = \delta$ is zero. Similarly, for a fixed non-zero n -bit string δ , the probability of the event $V \oplus V' = \delta$ is zero. On the other hand, if $\delta = 0^n$, the probability of the event $U \oplus U' = \delta$ is boiled down to the probability of the event $X[\ell] = X'[\ell']$ and $Y[\ell] = Y'[\ell']$. Similarly, if $\delta = 0^n$, the probability of the event $V \oplus V'$ is boiled down to the probability of the event $X[\ell] = X'[\ell']$ and $Y[\ell] = Y'[\ell']$. Now, from the analysis of Case-I in the proof of Lemma 3, we have

$$\Pr[X[\ell] = X'[\ell'], Y[\ell] = Y'[\ell']] \leq \frac{1}{2^{2n}}.$$

Therefore,

$$\Pr[U \oplus U' = \delta] = \begin{cases} \frac{1}{2^{2n}} & \text{if } \delta = 0^n \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \Pr[V \oplus V' = \delta] = \begin{cases} \frac{1}{2^{2n}} & \text{if } \delta = 0^n \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

Case-II: $\neg(X[\ell] = X'[\ell'] \text{ and } Y[\ell] = Y'[\ell'])$. As before, we split this case into three subcases as follows:

- **Case-A:** ($X[\ell] = X'[\ell']$ and $Y[\ell] \neq Y'[\ell']$). We split this subcase further into two subcases as follows: (a) if $\text{Lsb}_{n-3}(Y[\ell]) = \text{Lsb}_{n-3}(Y'[\ell'])$, then it implies that $U = U'$. Therefore, for a fixed non-zero δ , the probability that $U \oplus U' = \delta$ holds is zero. On the other hand, if $\delta = 0^n$, then the probability is boiled down to evaluate the probability of the event $X[\ell] = X'[\ell']$ and $\text{Lsb}_{n-3}(Y[\ell]) = \text{Lsb}_{n-3}(Y'[\ell'])$ holds. Note that, $X[\ell] = X'[\ell']$ and $\text{Lsb}_{n-3}(Y[\ell]) = \text{Lsb}_{n-3}(Y'[\ell'])$ implies the following system of equations hold:

$$\begin{cases} (Z[1] \oplus Z[2] \oplus \dots \oplus Z[\ell]) \oplus (Z'[1] \oplus Z'[2] \oplus \dots \oplus Z'[\ell']) = 0^n \\ \text{Lsb}_{n-3}(2^{\ell-1}Z[1] \oplus 2^{\ell-2}Z[2] \oplus \dots \oplus Z[\ell]) \\ \oplus \text{Lsb}_{n-3}(2^{\ell'-1}Z'[1] \oplus 2^{\ell'-2}Z'[2] \oplus \dots \oplus Z'[\ell']) = 0^n. \end{cases}$$

Note that the rank of the above system of equations is at least 1. Hence, the probability of the above system of equations hold is at most $2^3(2^3 - 1)/2^n$ as for each assignment in the most significant three bits of $Y[\ell]$, there are exactly 7 assignments in the most significant three bits of $Y'[\ell']$ that eventually ensures $Y[\ell] \neq Y'[\ell']$. On the other hand, under the subcase (a) V and V' are distinct and they are not independent random variables. Thus, for any non-zero n -bit string δ , the probability of the event $V \oplus V' = \delta$ is at most $1/(2^n - 1)$. Therefore, by summarizing, we have

$$\Pr[U \oplus U' = \delta] \leq \begin{cases} \frac{56}{2^n} & \text{if } \delta = 0^n \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \Pr[V \oplus V' = \delta] \leq \begin{cases} \frac{1}{2^n - 1} & \text{if } \delta \neq 0^n \\ 0 & \text{otherwise} \end{cases}$$

(b) If $\text{Lsb}_{n-3}(Y[\ell]) \neq \text{Lsb}_{n-3}(Y'[\ell'])$, then it implies that U and U' are two independent random variables. Therefore, for a fixed n -bit string δ , the probability that $U \oplus U' = \delta$ holds is at most $1/2^n$. On the other hand, under the subcase (b) V and V' are distinct and they are not independent random variables. Thus, for any non-zero n -bit string δ , the probability of the event $V \oplus V' = \delta$ is at most $1/(2^n - 1)$. Therefore, by summarizing, we have

$$\Pr[U \oplus U' = \delta] \leq \frac{1}{2^n} \quad \text{and} \quad \Pr[V \oplus V' = \delta] \leq \begin{cases} \frac{1}{2^n - 1} & \text{if } \delta \neq 0^n \\ 0 & \text{otherwise} \end{cases}$$

Therefore, by combining the above two subcases, we derive the upper bound on the probability of the event E_1 and the probability of the event E_2 in Case-A as follows:

$$\Pr[U \oplus U' = \delta] \leq \frac{57}{2^n}, \quad \text{and} \quad \Pr[V \oplus V' = \delta] \leq \frac{2}{2^n - 1}. \quad (23)$$

- **Case-B:** ($Y[\ell] = Y'[\ell']$ and $X[\ell] \neq X'[\ell']$). This case is symmetrical to Case-A by just swapping the role between U and V and that of U' and V' . Thus, we omit the details of the analysis and conclude that for any n -bit string δ ,

$$\Pr[U \oplus U' = \delta] \leq \frac{2}{2^n - 1} \quad \text{and} \quad \Pr[V \oplus V' = \delta] \leq \frac{57}{2^n}. \quad (24)$$

- **Case-C:** ($X[\ell] \neq X'[\ell']$ and $Y[\ell] \neq Y'[\ell']$). We analyze this case into the following four subcases: (a) if $\text{Lsb}_{n-3}(X[\ell]) = \text{Lsb}_{n-3}(X'[\ell'])$ and $\text{Lsb}_{n-3}(Y[\ell]) = \text{Lsb}_{n-3}(Y'[\ell'])$, then it follows that U and U' are distinct and they are not independent random variables. Similarly, V and V' are distinct and they are not independent random variables. Therefore, for any non-zero n -bit string δ , the probability of the event $U \oplus U' = \delta$ is bounded by $1/2^n - 1$. Similarly, for any non-zero n -bit string δ , the probability of the event $V \oplus V' = \delta$ is bounded by $1/2^n - 1$. So, we have

$$\Pr[U \oplus U' = \delta] \leq \frac{1}{2^n - 1} \text{ and } \Pr[V \oplus V' = \delta] \leq \frac{1}{2^n - 1}$$

- (b) If $\text{Lsb}_{n-3}(X[\ell]) = \text{Lsb}_{n-3}(X'[\ell'])$ and $\text{Lsb}_{n-3}(Y[\ell]) \neq \text{Lsb}_{n-3}(Y'[\ell'])$, then it follows that V and V' are distinct and they are not independent random variables. However, since $\text{Lsb}_{n-3}(Y[\ell]) \neq \text{Lsb}_{n-3}(Y'[\ell'])$, it implies that U and U' are two independent random variables. Thus, for a fixed string δ , we have

$$\Pr[U \oplus U' = \delta] \leq \frac{1}{2^n} \text{ and } \Pr[V \oplus V' = \delta] \leq \begin{cases} \frac{1}{2^n - 1} & \text{if } \delta \neq 0^n \\ 0 & \text{otherwise} \end{cases}$$

- (c) If $\text{Lsb}_{n-3}(X[\ell]) \neq \text{Lsb}_{n-3}(X'[\ell'])$ and $\text{Lsb}_{n-3}(Y[\ell]) = \text{Lsb}_{n-3}(Y'[\ell'])$, then it follows that U and U' are distinct and they are not independent random variables. However, since $\text{Lsb}_{n-3}(X[\ell]) \neq \text{Lsb}_{n-3}(X'[\ell'])$, it implies that V and V' are two independent random variables. Thus, for a fixed n -bit string δ , we have

$$\Pr[U \oplus U' = \delta] \leq \begin{cases} \frac{1}{2^n - 1} & \text{if } \delta \neq 0^n \\ 0 & \text{otherwise} \end{cases} \text{ and } \Pr[V \oplus V' = \delta] \leq \frac{1}{2^n}.$$

- (d) If both $\text{Lsb}_{n-3}(X[\ell]) \neq \text{Lsb}_{n-3}(X'[\ell'])$ and $\text{Lsb}_{n-3}(Y[\ell]) \neq \text{Lsb}_{n-3}(Y'[\ell'])$, then it follows that V and V' are two independent random variables and U and U' are two independent random variables. Thus, for a fixed n -bit string δ we have

$$\Pr[U \oplus U' = \delta] \leq \frac{1}{2^n} \text{ and } \Pr[V \oplus V' = \delta] \leq \frac{1}{2^n}.$$

By combining the above four subcases, we derive the upper bound on the probability of the event E_1 and the probability of the event E_2 in **Case-C** as follows:

$$\Pr[U \oplus U' = \delta] \leq \frac{4}{2^n - 1} \text{ and } \Pr[V \oplus V' = \delta] \leq \frac{4}{2^n - 1}. \quad (25)$$

By combining Eqn. (23), Eqn. (24), and Eqn. (25), we derive the upper bound on the probability of the event E_1 and the probability of the event E_2 in **Case-II** as follows:

$$\Pr[U \oplus U' = \delta] \leq \frac{63}{2^n - 1} \text{ and } \Pr[V \oplus V' = \delta] \leq \frac{63}{2^n - 1}. \quad (26)$$

Finally, by combining Eqn. (22) and Eqn. (26), we derive the upper bound on the probability of the event E_b for $b \in \{1, 2\}$ as follows:

$$\Pr[U \oplus U' = \delta] \leq \frac{63}{2^n - 1} + \frac{1}{2^{2n}} \text{ and } \Pr[V \oplus V' = \delta] \leq \frac{63}{2^n - 1} + \frac{1}{2^{2n}}. \quad (27)$$

A.3 Proof of Lemma 5

Let Δ be a fixed $2n$ -bit string which is parsed as two n -bit strings Δ_1 and Δ_2 , i.e., $\Delta = \Delta_1 \parallel \Delta_2$, and $\Delta_1, \Delta_2 \in \{0, 1\}^n$. Let E denotes the event that the xor of the output of ZHASH+ on two distinct messages M and M' attains the value Δ , i.e.,

$$\Pr[E] = \Pr[Y_1 \oplus Y_1' = \Delta_1, Y_2 \oplus Y_2' = \Delta_2],$$

where the randomness of the above probability is defined over Y_1, Y_1', Y_2 , and Y_2' . Thus, we bound the probability of the event E holds. From the description of the ZHASH+ construction, we have

$$Y_1 \leftarrow \tilde{E}_{K_h}(010 \parallel \text{Lsb}_{n-3}(V), U), \quad Y_2 \leftarrow \tilde{E}_{K_h}(011 \parallel \text{Lsb}_{n-3}(U), V)$$

(see line 12 and line 13 of Fig. 4). Since the domain separation string used in the tweak part of deriving Y_1 and Y_2 are different, it follows that Y_1 and Y_2 are two independently sampled n -bit strings. Let us denote $\text{Lsb}_{n-3}(V)$ as \hat{V} and $\text{Lsb}_{n-3}(U)$ as \hat{U} . Now, to bound the probability of the event E holds, we have the following two cases:

Case-I: ($U = U'$ and $V = V'$). This event implies $\hat{U} = \hat{U}'$ and $\hat{V} = \hat{V}'$ and hence it implies $Y_1 = Y_1'$ and $Y_2 = Y_2'$. The analysis mostly follows the analysis of [IMPS17, Lemma 4]. We revisit to the analysis for the sake of completeness. Note that, the condition $U = U'$ and $V = V'$ implies the following two equations:

$$\begin{cases} (2^{\ell-1}C_l[1] \oplus 2^{\ell-2}C_l[2] \oplus \dots \oplus C_l[\ell]) \oplus (2^{\ell'-1}C'_l[1] \oplus 2^{\ell'-2}C'_l[2] \oplus \dots \oplus C'_l[\ell']) = 0^n \\ (C_r[1] \oplus C_r[2] \oplus \dots \oplus C_r[\ell]) \oplus (C'_r[1] \oplus C'_r[2] \oplus \dots \oplus C'_r[\ell']) = 0^n. \end{cases}$$

Note that, if either one of the Δ_1 or Δ_2 is a non-zero n -bit string, then the probability of the event is zero. Thus, if $\Delta = 0^{2n}$, then we bound the probability of the above system of equations hold in several cases as follows:

Case-A: Let $\ell = \ell'$ and $\exists h \in \{1, \dots, \ell\}$ such that $B[h] \neq B'[h]$ and $B[s] = B'[s]$ for all $s \in \{1, \dots, \ell\} \setminus \{h\}$. Therefore,

$$U \oplus U' \Rightarrow 2^{\ell-h}(C_l[h] \oplus C'_l[h]), \quad V \oplus V' \Rightarrow C_r[h] \oplus C'_r[h].$$

To imply that $U = U'$ and $V = V'$, it must have to be the case that $C_l[h] = C'_l[h]$ and $C_r[h] = C'_r[h]$. But then it means that $X_r[h] = X'_r[h]$ and $X_l[h] = X'_l[h]$, which contradicts to the assumption that $B[h] \neq B'[h]$. Therefore, in this case, the probability of the event $U = U'$ and $V = V'$ is zero.

Remark 3. We would like to remark here that if we had used

$$Y_1 \leftarrow \tilde{E}_{K_h}(010 \parallel \text{Lsb}_{n-3}(V), U), \quad Y_2 \leftarrow \tilde{E}_{K_h}(011 \parallel \text{Lsb}_{n-3}(V), U),$$

then we cannot guarantee the axu advantage of ZHASH+ up to 2^{-2n} . Note that, the bad event would be $U = U'$ and $\hat{V} = \hat{V}'$. This renders $Y_1 = Y_1'$ and $Y_2 = Y_2'$. Let us consider an adversary \mathcal{A} that chooses two messages M, M' of equal length in such way that ensures $X_l[h] = X'_l[h]$ and $\text{Lsb}_{n-3}(X_r[h]) = \text{Lsb}_{n-3}(X'_r[h])$ and for all $s \neq h$, $B[s] = B'[s]$. In this case, we have

$$\Delta \hat{U} = 2^{\ell-h}(C_l[h] \oplus C'_l[h]), \quad \Delta \hat{V} = \text{Lsb}_{n-3}(C_l[h] \oplus X_r[h]) \oplus \text{Lsb}_{n-3}(C'_l[h] \oplus X'_r[h]).$$

Note that $C_l[h] = C'_l[h]$ holds with probability 2^{-n} as they are independently distributed (the tweak involved in generating $C_l[h]$ is different from the tweak used in generating $C'_l[h]$). If this collision holds, then that implies $\Delta \hat{V} = 0^n$. As a result, the event $U = U'$ implies $\hat{V} = \hat{V}'$ and hence the axu advantage of ZHASH+ is dropped to 2^{-n} .

Case-B: Let $\ell = \ell'$ and $\exists h \neq s \in \{1, \dots, \ell\}$ such that $B[h] \neq B'[h]$ and $B[s] \neq B'[s]$. Therefore,

$$\begin{aligned} \Delta U := U \oplus U' &\Rightarrow 2^{\ell-h}(C_l[h] \oplus C'_l[h]) \oplus 2^{\ell-s}(C_l[s] \oplus C'_l[s]) \oplus \underbrace{\bigoplus_{i \neq h,s} 2^{\ell-i}(C_l[i] \oplus C'_l[i])}_{\Sigma}, \\ \Delta V := V \oplus V' &\Rightarrow (C_r[h] \oplus C'_r[h]) \oplus (C_r[s] \oplus C'_r[s]) \oplus \underbrace{\bigoplus_{i \neq h,s} (C_r[i] \oplus C'_r[i])}_{\Theta}. \end{aligned}$$

Note that each term of Σ and Θ are independent. Let $\lambda_h = 2^{\ell-h}$ and $\lambda_s = 2^{\ell-s}$. Note that, $C_r[h] = C_l[h] \oplus X_r[h]$ and $C_r[s] = C_l[s] \oplus X_r[s]$. Then, $(\Delta U, \Delta V) = (0^n, 0^n)$ implies the following system of equations hold:

$$\begin{cases} \lambda_h \Delta C_l[h] \oplus \lambda_s \Delta C_l[s] = \Sigma \\ \Delta C_l[h] \oplus \Delta C_l[s] = \Theta \oplus \Delta X_r[h] \oplus \Delta X_r[s] \end{cases}$$

Note that the above system of equations has rank 2, and hence, it has a unique solution, namely

$$\begin{cases} \Delta C_l[h] = (\lambda_s \delta \oplus \Sigma) / (\lambda_h \oplus \lambda_s) \\ \Delta C_l[s] = \delta \oplus (\lambda_s \delta \oplus \Sigma) / (\lambda_h \oplus \lambda_s) \end{cases}$$

Note that, since $\Delta C_l[h]$ and $\Delta C_l[s]$ for $h \neq s$ are independently distributed as different tweaks are employed and they are uniformly distributed over $\{0, 1\}^n$, it implies that

$$\Pr[\Delta U = 0^n, \Delta V = 0^n] \leq \frac{1}{2^{2n}}.$$

Case-C: Let $\ell' = \ell + 1$. Then, we have

$$\begin{aligned} \Delta U &= \bigoplus_{1 \leq i \leq \ell} 2^{\ell-i} C_l[i] \oplus \bigoplus_{1 \leq i \leq \ell'} 2^{\ell'-i} C'_l[i] \\ &= C_l[\ell] \oplus 2C'_l[\ell] \oplus C'_l[\ell+1] \oplus \bigoplus_{i \neq \ell, \ell'} 2^{\ell-i} \Delta C_l[i] \\ \Delta V &= C_l[\ell] \oplus C'_l[\ell] \oplus C'_l[\ell+1] \oplus \Delta X_r[\ell] \oplus X'_r[\ell+1] \oplus \bigoplus_{i \neq \ell, \ell'} (\Delta C_l[i] \oplus \Delta X_r[i]) \end{aligned}$$

Then $(\Delta U, \Delta V) = (0^n, 0^n)$ implies the following system of equations hold:

$$\begin{cases} C_l[\ell] \oplus 2C'_l[\ell] \oplus C'_l[\ell+1] = \bigoplus_{i \neq \ell, \ell'} 2^{\ell-i} \Delta C_l[i] \\ C_l[\ell] \oplus C'_l[\ell] \oplus C'_l[\ell+1] = \Delta X_r[\ell] \oplus X'_r[\ell+1] \oplus \bigoplus_{i \neq \ell, \ell'} (\Delta C_l[i] \oplus \Delta X_r[i]) \end{cases}$$

Note that the above system of equations has rank 2, and hence, it has a unique solution. Moreover, as $(C_l[\ell] \oplus C'_l[\ell+1])$ and $C'_l[\ell]$ are uniformly distributed over $\{0, 1\}^n$ and they are independent, it implies that

$$\Pr[\Delta U = 0^n, \Delta V = 0^n] \leq \frac{1}{2^{2n}}.$$

Case-D: Let $\ell' \geq \ell + 2$. Then, we have

$$\begin{aligned} \Delta U &= \bigoplus_{1 \leq i \leq \ell} 2^{\ell-i} C_l[i] \oplus \bigoplus_{1 \leq i \leq \ell'} 2^{\ell'-i} C'_l[i] \\ &= 2C'_l[\ell'-1] \oplus C'_l[\ell'] \oplus \bigoplus_{1 \leq i \leq \ell} 2^{\ell-i} C_l[i] \oplus \bigoplus_{1 \leq i \leq \ell'-2} 2^{\ell'-i} C'_l[i] \\ \Delta V &= C'_l[\ell'-1] \oplus C'_l[\ell'] \oplus \bigoplus_{1 \leq i \leq \ell} (C_l[i] \oplus X_r[i]) \oplus \bigoplus_{1 \leq i \leq \ell'-2} (C'_l[i] \oplus X'_r[i]) \end{aligned}$$

Then $(\Delta U, \Delta V) = (0^n, 0^n)$ implies the following system of equations hold:

$$\begin{cases} 2C'_l[\ell' - 1] \oplus C'_l[\ell'] = \bigoplus_{1 \leq i \leq \ell} 2^{\ell-i} C_l[i] \oplus \bigoplus_{1 \leq i \leq \ell'-2} 2^{\ell'-i} C'_l[i] \\ C'_l[\ell' - 1] \oplus C'_l[\ell'] = \bigoplus_{1 \leq i \leq \ell} (C_l[i] \oplus X_r[i]) \oplus \bigoplus_{1 \leq i \leq \ell'-2} (C'_l[i] \oplus X'_r[i]) \end{cases}$$

Note that the above system of equations has rank 2, and hence, it has a unique solution. Moreover, as $(C'_l[\ell'] \oplus C'_l[\ell' - 1])$ are independent and uniformly distributed over $\{0, 1\}^n$, it implies that

$$\Pr[\Delta U = 0^n, \Delta V = 0^n] \leq \frac{1}{2^{2n}}.$$

Therefore, by combining the above four cases, we derive the upper bound on the probability of the event **E** holds in Case-I as follows:

$$\Pr[\mathbf{E}] = \Pr[Y_1 \oplus Y'_1 = \Delta_1, Y_2 \oplus Y'_2 = \Delta_2] \leq \frac{1}{2^{2n}}. \quad (28)$$

Case-II: $\neg(U = U' \text{ and } V = V')$. Bounding the probability of the event **E** holds under the assumption that $\neg(U = U' \text{ and } V = V')$ holds give rise to the following three subcases.

- **Case-A:** $(U = U' \text{ and } V \neq V')$. Bounding the probability of the event **E** holds under this case again gives rise to the following two subcases: (a) if $\text{Lsb}_{n-3}(V) = \text{Lsb}_{n-3}(V')$, then $Y_1 = Y'_1$. However, since $V \neq V'$, it implies that $Y_2 \neq Y'_2$. Therefore, for a fixed non-zero n -bit string Δ_2 , the event $Y_2 \oplus Y'_2 = \Delta_2$ holds with probability at most $1/(2^n - 1)$. Moreover, the event $U = U'$ and $\text{Lsb}_{n-3}(V) = \text{Lsb}_{n-3}(V')$ implies the following two equations:

$$\begin{cases} (2^{\ell-1} C_l[1] \oplus 2^{\ell-2} C_l[2] \oplus \dots \oplus C_l[\ell]) \oplus (2^{\ell'-1} C'_l[1] \oplus 2^{\ell'-2} C'_l[2] \oplus \dots \oplus C'_l[\ell']) = 0^n \\ \text{Lsb}_{n-3}(C_r[1] \oplus C_r[2] \oplus \dots \oplus C_r[\ell]) \oplus \text{Lsb}_{n-3}(C'_r[1] \oplus C'_r[2] \oplus \dots \oplus C'_r[\ell']) = 0^n. \end{cases}$$

Since the rank of the above system of equations is at least 1, the probability that the above system of equations hold is at most $2^3(2^3 - 1)/2^n$ as each assignment in the most significant three bits of V , there are exactly 7 assignments in the most significant three bits of V' that eventually ensures $V \neq V'$. Therefore, we have,

$$\Pr[Y_1 \oplus Y'_1 = \Delta_1, Y_2 \oplus Y'_2 = \Delta_2] \leq \begin{cases} \frac{56}{2^n(2^n-1)} & \text{if } \Delta_1 = 0^n \text{ and } \Delta_2 \neq 0^n \\ 0 & \text{if } \Delta_1 \neq 0^n \text{ or } \Delta_2 = 0^n. \end{cases}$$

- (b) If $\text{Lsb}_{n-3}(V) \neq \text{Lsb}_{n-3}(V')$, then Y_1 and Y'_1 are independently distributed. However, since $V \neq V'$, it implies that $Y_2 \neq Y'_2$ and they are not independently distributed. Therefore, for a fixed non-zero n -bit string Δ_2 and for any n -bit string Δ_1 , the event $Y_1 \oplus Y'_1 = \Delta_1, Y_2 \oplus Y'_2 = \Delta_2$ holds with probability at most $1/2^n(2^n - 1)$. On the other hand, if $\Delta_2 = 0^n$, then the probability of the event becomes zero, i.e.,

$$\Pr[Y_1 \oplus Y'_1 = \Delta_1, Y_2 \oplus Y'_2 = \Delta_2] \leq \begin{cases} \frac{1}{2^n(2^n-1)} & \text{if } \Delta_2 \neq 0^n \\ 0 & \text{if } \Delta_2 = 0^n. \end{cases}$$

Therefore, by combining the two subcases, we derive the upper bound on the probability of the event **E** in Case-A as follows:

$$\Pr[Y_1 \oplus Y'_1 = \Delta_1, Y_2 \oplus Y'_2 = \Delta_2] \leq \frac{57}{2^n(2^n - 1)}. \quad (29)$$

- **Case-B:** $(V = V' \text{ and } U \neq U')$. This case is symmetrical to Case-A (by just swapping the role between Y_1 and Y_2 , and that of between Y'_1 and Y'_2). Thus, we omit the details of the analysis and conclude that

$$\Pr[Y_1 \oplus Y'_1 = \Delta_1, Y_2 \oplus Y'_2 = \Delta_2] \leq \frac{57}{2^n(2^n - 1)}. \quad (30)$$

- **Case-C:** ($U \neq U'$ and $V \neq V'$). Bounding the probability of the event **E** holds under this case again gives rise to the following four subcases: (a) if $\text{Lsb}_{n-3}(U) = \text{Lsb}_{n-3}(U')$ and $\text{Lsb}_{n-3}(V) = \text{Lsb}_{n-3}(V')$, then it follows that Y_1 and Y_1' are not two independent random variables. Similarly, Y_2 and Y_2' are not two independent random variables. However, since $U \neq U'$ and $V \neq V'$, it implies that $Y_1 \neq Y_1'$ and $Y_2 \neq Y_2'$. Therefore, for a fixed $2n$ -bit string $\Delta = \Delta_1 \parallel \Delta_2$ such that $\Delta_1 \neq 0^n$ and $\Delta_2 \neq 0^n$, the probability of the event

$$Y_1 \oplus Y_1' = \Delta_1, Y_2 \oplus Y_2' = \Delta_2$$

is bounded by $1/(2^n - 1)^2$ due to the randomness Y_1 and Y_2 . Otherwise, the probability of the event would have been zero, i.e.,

$$\Pr[Y_1 \oplus Y_1' = \Delta_1, Y_2 \oplus Y_2' = \Delta_2] \leq \begin{cases} \frac{1}{(2^n - 1)^2} & \text{if } \Delta_1 \neq 0^n \text{ and } \Delta_2 \neq 0^n \\ 0 & \text{otherwise .} \end{cases}$$

- (b) If $\text{Lsb}_{n-3}(U) = \text{Lsb}_{n-3}(U')$ and $\text{Lsb}_{n-3}(V) \neq \text{Lsb}_{n-3}(V')$, then Y_2 and Y_2' are distinct and they are not independent random variables. However, since $\text{Lsb}_{n-3}(V) \neq \text{Lsb}_{n-3}(V')$, it implies that Y_1 and Y_1' are two independent random variables. Thus, for a fixed $2n$ bit string Δ such that $\Delta = \Delta_1 \parallel \Delta_2$ with $\Delta_2 \neq 0^n$, we have

$$\Pr[Y_1 \oplus Y_1' = \Delta_1, Y_2 \oplus Y_2' = \Delta_2] \leq \frac{1}{2^n(2^n - 1)}.$$

- (c) If $\text{Lsb}_{n-3}(U) \neq \text{Lsb}_{n-3}(U')$ and $\text{Lsb}_{n-3}(V) = \text{Lsb}_{n-3}(V')$, then Y_1 and Y_1' are distinct and they are not independent random variables. However, since $\text{Lsb}_{n-3}(U) \neq \text{Lsb}_{n-3}(U')$, it implies that Y_2 and Y_2' are two independent random variables. Thus, for a fixed non-zero $2n$ bit string Δ such that $\Delta = \Delta_1 \parallel \Delta_2$ with $\Delta_1 \neq 0^n$, we have

$$\Pr[Y_1 \oplus Y_1' = \Delta_1, Y_2 \oplus Y_2' = \Delta_2] \leq \frac{1}{2^n(2^n - 1)}.$$

- (d) Finally, if both $\text{Lsb}_{n-3}(U) \neq \text{Lsb}_{n-3}(U')$ and $\text{Lsb}_{n-3}(V) \neq \text{Lsb}_{n-3}(V')$, then Y_2 and Y_2' are two independent random variables and Y_1 and Y_1' are two independent random variables. Thus, for a fixed $\Delta_1 \parallel \Delta_2$, we have

$$\Pr[Y_1 \oplus Y_1' = \Delta_1, Y_2 \oplus Y_2' = \Delta_2] \leq \frac{1}{2^{2n}}.$$

Therefore, by combining the above four subcases, we derive the upper bound on the probability of the event **E** in Case C as follows:

$$\Pr[Y_1 \oplus Y_1' = \Delta_1, Y_2 \oplus Y_2' = \Delta_2] \leq \frac{4}{(2^n - 1)^2}. \quad (31)$$

Now, by combining Eqn. (29), Eqn. (30), and Eqn. (31), we derive the upper bound on the probability of the event **E** in Case-II as follows:

$$\Pr[Y_1 \oplus Y_1' = \Delta_1, Y_2 \oplus Y_2' = \Delta_2] \leq \frac{118}{(2^n - 1)^2}. \quad (32)$$

Finally, by combining Eqn. (28) and Eqn. (32), we derive the upper bound on the probability of the event **E** as

$$\Pr[\mathbf{E}] = \Pr[Y_1 \oplus Y_1' = \Delta_1, Y_2 \oplus Y_2' = \Delta_2] \leq \frac{118}{(2^n - 1)^2} + \frac{1}{2^{2n}}. \quad (33)$$

A.4 Proof of Lemma 6

We bound the almost-xor-universal advantage of each of the n -bit output of ZHASH+ in a similar way as we did in the proof of Lemma 5. For $b \in \{1, 2\}$, let E_b denotes the event $Y_b \oplus Y'_b = \delta$. As before, the analysis of upper bounding the probability of the event E_b is based on the following two cases:

Case-I: ($U = U'$ and $V = V'$). In this case $Y_1 = Y'_1$. Therefore, for a fixed non-zero n -bit string δ , the probability of the event $Y_1 \oplus Y'_1 = \delta$ is zero. Similarly, for a fixed non-zero n -bit string δ , the probability of the event $Y_2 \oplus Y'_2 = \delta$ is zero. On the other hand, if $\delta = 0^n$, the probability of the event $Y_1 \oplus Y'_1$ is boiled down to the probability of the event $U = U'$ and $V = V'$. Similarly, if $\delta = 0^n$, the probability of the event $Y_2 \oplus Y'_2$ is boiled down to the probability of the event $U = U'$ and $V = V'$. Now, from the analysis of Case-I in the proof of Lemma 5, we have

$$\Pr[U = U', V = V'] \leq \frac{1}{2^{2n}}.$$

Therefore, for $b \in \{1, 2\}$, we have

$$\Pr[Y_b \oplus Y'_b = \delta] = \begin{cases} \frac{1}{2^{2n}} & \text{if } \delta = 0^n \\ 0 & \text{otherwise} \end{cases} \quad (34)$$

Case-II: $\neg(U = U' \text{ and } V = V')$. As before, we split this case into three subcases as follows:

- **Case-A:** ($U = U'$ and $V \neq V'$). We split this subcase further into two subcases as follows: (a) if $\text{Lsb}_{n-3}(V) = \text{Lsb}_{n-3}(V')$, then it implies that $Y_1 = Y'_1$. Therefore, for a fixed non-zero δ , the probability that $Y_1 \oplus Y'_1 = \delta$ holds is zero. On the other hand, if $\delta = 0^n$, then the probability is boiled down to evaluate the probability of the event $U = U'$ and $\text{Lsb}_{n-3}(V) = \text{Lsb}_{n-3}(V')$ holds. Note that, $U = U'$ and $\text{Lsb}_{n-3}(V) = \text{Lsb}_{n-3}(V')$ implies the following system of equations hold:

$$\begin{aligned} (2^{\ell-1}C_l[1] \oplus 2^{\ell-2}C_l[2] \oplus \dots \oplus C_l[\ell]) \oplus (2^{\ell'-1}C'_l[1] \oplus 2^{\ell'-2}C'_l[2] \oplus \dots \oplus C'_l[\ell']) &= 0^n \\ \text{Lsb}_{n-3}(C_r[1] \oplus C_r[2] \oplus \dots \oplus C_r[\ell]) \oplus \text{Lsb}_{n-3}(C'_r[1] \oplus C'_r[2] \oplus \dots \oplus C'_r[\ell']) &= 0^n. \end{aligned}$$

Note that the rank of the above system of equations is at least 1. Hence, the probability of the above system of equations hold is at most $2^3(2^3 - 1)/2^n$ as for each assignment in the most significant three bits of V , there are exactly 7 assignments in the most significant three bits of V' that eventually ensures $V \neq V'$. On the other hand, under the subcase (a) Y_2 and Y'_2 are distinct and they are not independent random variables. Thus, for any non-zero n -bit string δ , the probability of the event $Y_2 \oplus Y'_2 = \delta$ is at most $1/2^n$. Therefore, by summarizing, we have

$$\Pr[Y_1 \oplus Y'_1 = \delta] \leq \begin{cases} \frac{56}{2^n} & \text{if } \delta = 0^n \\ 0 & \text{otherwise} \end{cases}$$

On the other hand

$$\Pr[Y_2 \oplus Y'_2 = \delta] \leq \begin{cases} \frac{1}{2^{n-1}} & \text{if } \delta \neq 0^n \\ 0 & \text{otherwise} \end{cases}$$

- (b) If $\text{Lsb}_{n-3}(V) \neq \text{Lsb}_{n-3}(V')$, then it implies that Y_1 and Y'_1 are two independent random variables. Therefore, for a fixed n -bit string δ , the probability that $Y_1 \oplus Y'_1 = \delta$ holds is at most $1/2^n$. On the other hand, under the subcase (b) Y_2 and Y'_2 are distinct and they are not independent random variables. Thus, for any non-zero

n -bit string δ , the probability of the event $Y_2 \oplus Y'_2 = \delta$ is at most $1/2^n - 1$. Therefore, by summarizing, we have

$$\Pr[Y_1 \oplus Y'_1 = \delta] \leq \frac{1}{2^n}.$$

On the other hand

$$\Pr[Y_2 \oplus Y'_2 = \delta] \leq \begin{cases} \frac{1}{2^n - 1} & \text{if } \delta \neq 0^n \\ 0 & \text{otherwise} \end{cases}$$

Therefore, by combining the above two subcases, we derive the upper bound on the probability of the event E_1 and the probability of the event E_2 in Case-A as follows:

$$\Pr[Y_1 \oplus Y'_1 = \delta] \leq \frac{57}{2^n}, \quad \Pr[Y_2 \oplus Y'_2 = \delta] \leq \frac{2}{2^n - 1}. \quad (35)$$

- **Case-B:** ($V = V'$ and $U \neq U'$). This case is symmetrical to Case-A by just swapping the role between Y_1 and Y_2 and that of Y'_1 and Y'_2 . Thus, we omit the details of the analysis and conclude that for any n -bit string δ ,

$$\Pr[Y_1 \oplus Y'_1 = \delta] \leq \frac{2}{2^n - 1}, \Pr[Y_2 \oplus Y'_2 = \delta] \leq \frac{57}{2^n}. \quad (36)$$

- **Case-C:** ($U \neq U'$ and $V \neq V'$). We analyze this case into the following four subcases: (a) if $\text{Lsb}_{n-3}(U) = \text{Lsb}_{n-3}(U')$ and $\text{Lsb}_{n-3}(V) = \text{Lsb}_{n-3}(V')$, then it follows that Y_1 and Y'_1 are distinct and they are not independent random variables. Similarly, Y_2 and Y'_2 are distinct and they are not independent random variables. Therefore, for any non-zero n -bit string δ , the probability of the event $Y_1 \oplus Y'_1 = \delta$ is bounded by $1/2^n - 1$. Similarly, for any non-zero n -bit string δ , the probability of the event $Y_2 \oplus Y'_2 = \delta$ is bounded by $1/2^n - 1$, i.e., for $b \in \{1, 2\}$, we have

$$\Pr[Y_b \oplus Y'_b = \delta] \leq \frac{1}{2^n - 1}.$$

- (b) If $\text{Lsb}_{n-3}(U) = \text{Lsb}_{n-3}(U')$ and $\text{Lsb}_{n-3}(V) \neq \text{Lsb}_{n-3}(V')$, then it follows that Y_2 and Y'_2 are distinct and they are not independent random variables. However, since $\text{Lsb}_{n-3}(U) = \text{Lsb}_{n-3}(U')$, it implies that Y_1 and Y'_1 are two independent random variables. Thus, for a fixed string δ , we have

$$\Pr[Y_1 \oplus Y'_1 = \delta] \leq \frac{1}{2^n}$$

On the other hand,

$$\Pr[Y_2 \oplus Y'_2 = \delta] \leq \begin{cases} \frac{1}{2^n - 1} & \text{if } \delta \neq 0^n \\ 0 & \text{otherwise} \end{cases}$$

- (c) If $\text{Lsb}_{n-3}(U) \neq \text{Lsb}_{n-3}(U')$ and $\text{Lsb}_{n-3}(V) = \text{Lsb}_{n-3}(V')$, then it follows that Y_1 and Y'_1 are distinct and they are not independent random variables. However, since $\text{Lsb}_{n-3}(U) \neq \text{Lsb}_{n-3}(U')$, it implies that Y_2 and Y'_2 are two independent random variables. Thus, for a fixed n -bit string δ , we have

$$\Pr[Y_1 \oplus Y'_1 = \delta] \leq \begin{cases} \frac{1}{2^n - 1} & \text{if } \delta \neq 0^n \\ 0 & \text{otherwise} \end{cases}$$

On the other hand,

$$\Pr[Y_2 \oplus Y'_2 = \delta] \leq \frac{1}{2^n}.$$

(d) Finally, if both $\text{Lsb}_{n-3}(U) \neq \text{Lsb}_{n-3}(U')$ and $\text{Lsb}_{n-3}(V) \neq \text{Lsb}_{n-3}(V')$, then it follows that Y_2 and Y'_2 are two independent random variables and Y_1 and Y'_1 are two independent random variables. Thus, for a fixed n -bit string δ and for $b \in \{1, 2\}$ we have

$$\Pr[Y_b \oplus Y'_b = \delta] \leq \frac{1}{2^n}.$$

By combining the above four subcases, we derive the upper bound on the probability of the event E_1 and the probability of the event E_2 in **Case-C** as follows:

$$\Pr[Y_1 \oplus Y'_1 = \delta] \leq \frac{4}{2^n - 1}, \quad \Pr[Y_2 \oplus Y'_2 = \delta] \leq \frac{4}{2^n - 1}. \quad (37)$$

By combining Eqn. (35), Eqn. (36), and Eqn. (37), we derive the upper bound on the probability of the event E_1 and the probability of the event E_2 in **Case-II** as follows:

$$\Pr[Y_1 \oplus Y'_1 = \delta] \leq \frac{63}{2^n - 1}, \quad \Pr[Y_2 \oplus Y'_2 = \delta] \leq \frac{63}{2^n - 1}. \quad (38)$$

Finally, by combining Eqn. (34) and Eqn. (38), we derive the upper bound on the probability of the event E_b for $b \in \{1, 2\}$ as follows:

$$\Pr[Y_b \oplus Y'_b = \delta] \leq \frac{63}{2^n - 1} + \frac{1}{2^{2n}}. \quad (39)$$