HCTR+: An Optimally Secure TBC-based Accordion Mode

Nilanjan Datta[1,2], Avijit Dutta[1,2], Shibam Ghosh[3,4], Eik List[5] and Hrithik Nandi[1,6]

Institute for Advancing Intelligence, TCG CREST, Kolkata, India
Academy of Scientific and Innovative Research, Ghaziabad, India
{nilanjan.datta,avijit.dutta}@tcgcrest.org
Inria, Paris, France
Department of Computer Science, University of Haifa, Haifa, Israel
shibam.ghosh@inria.fr
Nanyang Technological University, Singapore, Singapore
elist@posteo.net
Ramakrishna Mission Vivekananda Educational and Research Institute, Belur, India
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 wide-block cipher by Naor and Reingold. Since then, numerous constructions have been proposed, many of which are built on existing block ciphers and are secure up to the birthday bound for the total number of blocks queried. Although there has been a recent slowdown in the development of such ciphers, the latest NIST proposal for Accordion modes has reignited the 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 to the symmetric key research community. 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 wide 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-256.

Keywords: Tweakable wide block cipher · Tweakable block cipher · Accordion mode · Provable security · Optimal security · AES · HCTR

1 Introduction

A Tweakable Enciphering Scheme (TES) is a function $\widetilde{\mathcal{E}} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{C}$ that maps plaintexts M of variable lengths to ciphertexts C under the control of a secret key K and a public tweak T: $C = \widetilde{\mathcal{E}}_K^T(M)$ [HR03]. $\mathcal{E}$ must be length-preserving and efficiently invertible. The security requirement demanded from TES is that of a strong tweakable pseudorandom permutation, i.e., it is a function mapping (T, M) to $C = \widetilde{\mathcal{E}}_K^T(M)$ and

(T, C) to $M = \widetilde{\mathcal{D}}_K^T(C)$ that is indistinguishable (when the key K is random and secret) from a permutation chosen randomly from a $\mathcal{T}$-indexed family of permutations.

TES are important for a diverse set of applications. Besides mere privacy protection, they have shown to be of interest for realizing strong and robust authenticated encryption [HKR15, ST13], onion authenticated encryption [RZ18], (full-)disk and file-based encryption [Dwo10, IEE24, Rog11], bandwidth-efficient network-protocol packet encryption, and key wrap [Dwo12, RS06]. In several of these applications, a tweak input is crucial to separate the encryption of equal blocks at different locations efficiently.

TES have attracted vast amounts of research since their first proposal two decades ago (we give a brief summary in Appendix A). Yet, their versatility has seen a recent emphasis. Having organized many world-leading international cryptographic competitions in the past decades, the NIST and the cryptographic community have invested enormous effort into them. As one consequence, being aware that standardization of many schemes for various purposes is cumbersome, the NIST appreciated the manifold uses of wide encryption schemes. At a 2023 workshop, the NIST announced its plan for potential standardization of a state-of-the-art and AES-based wide encryption scheme under their term of *Accordion mode* [CDD+24]. According to the NIST document [CDD+24]

> *A well-designed accordion mode should provide significant advantages over most block-cipher modes that NIST had specified in the SP 800-38 series.*

Thus, the NIST wants a one-size-fits-all accordion mode for authenticated encryption (with associated data), tweakable encryption schemes, deterministic authenticated encryption schemes, and more. Accordion modes with additional security properties such as multi-user security, beyond birthday bound security, commitment security, key-dependent input security, and nonce-hiding security have been encouraged.

1.1 An Overview of Highly Secure Accordion Ciphers

According to the NIST requirements, the most important platform for the mode is likely to be high-end processors in desktop or laptop computers and those used in cloud environments, with hardware AES support where security is the primary concern.[1] Since we typically work with 128-bit blocks, achieving security beyond the birthday bound is highly desirable.[2] An accordion mode that supports variable-length tweaks and ensures strong security can be useful for building various cryptographic applications with higher security. For example, one can design an n-bit-secure misuse-resistant authenticated encryption (MRAE) scheme by applying the Encode-then-Encipher (EtE) [BR00] paradigm with an n-bit-secure accordion mode, where n denotes the block size of the underlying block cipher and the (nonce, associated data) pair serves as the tweak. Similarly, one can also construct an optimally secure deterministic authenticated encryption (DAE) scheme from an n-bit-secure accordion mode by the EtE paradigm and leaving the tweak constant. This also provides desirably high security when used for key wrapping. Besides, one can also design n-bit-secure nonce-hiding AEAD schemes [BNT19] by considering the associated data as the tweak and encrypting a padded message (message followed by a 10^* block) containing the nonce. Thus, an n-bit-secure (a.k.a. optimally secure) accordion mode allows to build AEAD constructions with several targeted applications in the defense-in-depth category. The discussion above motivates us to design an efficient accordion mode that should provide n-bit security.

Since the NIST's proposal, several designs of accordion modes have emerged. Lee [Lee24] presented a variant of the HCTR construction, called Double-block HCTR (DbHCTR),

[1] It has been mentioned in [CDD+24] that performance is not as critical as security.

[2] NIST has proposed that the accordion mode should also support 256-bit block ciphers, subject to the fact that NIST approves one such block cipher in the future, which may reduce the need for this property.

Table 1: Comparison of beyond-birthday-bound secure (Tweakable) Enciphering Schemes. (T)BC/(T)MFC/$\otimes_b$) = (Tweakable) block cipher/(tweakable) multi-fork cipher/field multiplication in an b-bit field. $\ell/\tau/n$ = message length/tweak length (in blocks)/block size of the primitive. ✓/– = feature is present/absent. † = security degrades gracefully with increasing number of tweak repetitions.

Construction	Primitive	Tweak	Security (bit)	#Op./block Prim.	$\otimes_n$	$\otimes_{2n}$	#Keys	Message length (bit)
TCT2 [ST13]	BC	✓	$2n/3$	4	–	–	$\ell+\tau+19$	$\geq 2n$
bbb-ddd-AES [DMMT24]	BC	✓	$2n/3^{\dagger}$	2	2	–	1	$\geq 2n$
Db-HCTR [Lee24]	BC	✓	$2n/3$	1	4	–	3	$\geq 2n$
AES-CTET^{+} [CEL^{+}21]	BC	✓	$2n/3$	2	2	–	6	wn ($w \geq 2$)
DaryaiNoor [BVA24]	BC	✓	n	$\simeq 1$	–	2	1	$\geq 4n$
LargeBlock [MI11]	TBC	–	n	1	4	–	$\ell+2$	$\geq 2n$
THCTR [DN18]	TBC	✓	$n^{\dagger}$	1	2	–	3	$\geq 2n$
ZCZ [BLN18]	TBC	–	n	1.5	–	–	1	$\geq 2n$
Kohinoor [BVA24]	TMFC	✓	n	1	–	2	1	$\geq 4n$
PHCTR+ [**Ours**]	TBC	✓	n	3	–	–	1	$\geq 2n$
ZHCTR+ [**Ours**]	TBC	✓	n	2	–	–	1	$\geq 2n$

which uses a $2n$-bit state in the HCTR construction, where CTET^{+} [CEL^{+}21] is combined with the masked counter mode, wrapped with two concatenated universal hash functions. It has been claimed that DbHCTR achieves $2n/3$-bit security bound. Duy et al. [DFUB24] have proposed ACCOR-S and ACCOR-L, of which the former one achieves a security bound in the order of $q^2a^2/2^{129}$, whereas the latter one achieves security in the order of $qa\ell_{\mathsf{ctr}}/2^n$, 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. Unlike DbHCTR, the security proofs of both constructions need the underlying block cipher to be secure under related keys, which is a stronger assumption than the standard one. Recently, Dobraunig et al. [DMMT24] proposed an AES based accordion mode, called bbb-ddd-AES, which instantiates the Docked Double-Decker construction by Gunsing et al. [GDM22] with AES for up to $2^{2n/3}$ queries. However, the security bound for both ACCOR-L and bbb-ddd-AES degrades gracefully to the birthday bound with an increasing number of tweak repetitions. Recently, Bhati et al. [BVA24] proposed a generic framework for designing an accordion mode, called GEM, and its two instantiations KohiNoor and DaryaiNoor. The first one is based on ButterKnife, a tweakable primitive called tweakable multi-forkcipher [ACL^{+}24], and the latter one is based on the AES block cipher. Although both the constructions are secure up to 2^n queries, but the security of the former primitive, i.e., ButterKnife, is not well understood yet.

1.2 Our Contribution

In this paper, we propose an efficient n-bit-secure accordion mode, called HCTR+ built on a tweakable block cipher (TBC). From a high-level view, our construction is a single-keyed variant of HCTR, where the underlying n-bit keyed almost-xor-universal hash functions have been replaced by a $2n$-bit TBC-based almost-xor-universal hash function, the block cipher (a.k.a. SPRP) in the left-hand side of the HCTR construction has been replaced by a three-round TBC-based Luby Rackoff construction TLR3 [CDMS10], and the counter mode encryption has been replaced by a tweakable counter mode encryption $\widetilde{\mathsf{CTRT}}$ with a TBC, wherein the counter is fed into as the tweak to the primitive. $\widetilde{\mathsf{CTRT}}$ is almost identical to the CTRT construction [PS16] and the IVCTRT construction [LN17] except that the tweak space for the tweakable block cipher used in both the constructions (CTRT and IVCTRT) contains an indicator $i \in \mathbb{N}$ used for domain separation.

We propose two TBC-based instantiations for the underlying $2n$-bit hash function of HCTR+, PHASH+ and ZHASH+, which can be seen as variants of the underlying hash functions in PMAC2x [LN17] and ZMAC [IMPS17], respectively. We have shown that both hash functions achieve $O(2^{-2n})$-almost-xor-universality. We propose PHCTR+ and ZHCTR+ by instantiating HCTR+ using the hash functions above, and Deoxys-BC-256 [JNPS16] as the underlying tweakable block cipher. We compare PHCTR+ and ZHCTR+ with the existing beyond-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 support for a tweak in Table 1. Finally, we implement ZHCTR+ and PHCTR+ in software when their underlying TBC is instantiated with Deoxys-BC-256, and show that they can run in ca. 1 and 1.2 cycles per byte on recent x64 architectures with AES-NI.

Justification for Choosing Deoxys-BC. Our construction does not fully fit the NIST's current call, which demands that a mode should be based on the AES block cipher. However, we argue that the major underlying reasons for requesting the AES in accordion modes are

- it is an international open and widespread accepted standard,
- it has stood the test of time by considerable cryptanalytical efforts that have fostered trust in its security,
- it supports key lengths for the widely accepted security level of at least 128 bit,
- there are available secure implementations for many widespread platforms,
- schemes based on it can benefit from available native instructions and hardware acceleration of AES operations in off-the-shelf processors in widespread architectures, from Intel, AMD, ARM, and others,
- it enables significant parallelism for large inputs, leveraging modern CPU's ability to execute multiple AES instructions simultaneously.

Thus, the NIST's decision to demand schemes to build upon a single AES-based primitive is absolutely reasonable. Still, we would argue that those also hold for Deoxys-BC, which renders it a well-suited choice of primitive.

By instantiating our design with Deoxys-BC [JNPS16] that is based on the AES round function and the TWEAKEY schedule [JNP14a], it can also be efficiently implemented using the AES-NI instruction set available on off-the-shelf processors in most contemporary widespread architectures. Moreover, it has been standardized by ISO/IEC [ISO22] for widespread use and has resisted considerable efforts of cryptanalysis since its proposal, with at most 11 out of 14 rounds attacked with 256-bit tweakey and only under strong related-tweakey assumptions [CFZ+25, SZY+22].

From an implementor's perspective, the key schedule of Rijndael is known to be slow in practice and not well-suited for on-the-fly computation, as repeatedly deriving round keys during runtime can create a performance bottleneck, which is usually avoided by precomputing and storing the round keys. In contrast, a permutation-based tweakey schedule as in Deoxys-BC, QARMA [Ava17], or QARMAv2 [ABD+23] allows deriving round tweakeys at lower overhead. While not without cost, the tweakey schedule of Deoxys-BC requires only a lightweight shuffle permutation per tweakey word.

Comparison with Existing n-bit-secure Accordion Modes. We would like to compare the properties of our proposal with those of the closest existing highly secure wide block-cipher schemes in the literature for aiding readers. While a fully fair comparison among modes that use different primitives may be difficult or even impossible, that still tiny portfolio of TBC-based wide block ciphers renders it useful to also include schemes with

other primitives, so that we consider LargeBlock1/2 [MI11], THCTR [DN18], ZCZ [BLN18], as well as those above and KohiNoor and DaryaiNoor [BVA24]. Among those, DaryaiNoor and bbb-ddd-AES use a classical block cipher, Kohinoor uses a tweakable multi-forkcipher from [ACL+24], ZCZ a TBC with $3n$-bit tweak and key, and only LargeBlock and THCTR use a TBC with a $2n$-bit tweak and key, as HCTR+. We note that the use of a TBC can alleviate proofs and reduce the number of necessary keys as the tweak can be used. Moreover, a block-cipher-based construction may be more widely supported than a TBC-based scheme. Nonetheless, from the standardization of Deoxys-BC [ISO22], we can imagine that TBCs, and an AES-round-based TBC in particular, see further adoption in the mid-term future.

The primary benefit of our construction is its optimal security bound that does not limit the number of tweak repetitions unlike THCTR and bbb-ddd-AES. Hence, our proposed constructions can be easily extended to construct optimally secure MRAE, DAE, and nonce-hiding AEAD constructions. In terms of security, our constructions are on par with LargeBlock, THCTR, ZCZ, KohiNoor, and DaryaiNoor. However, the security bound of THCTR gracefully degrades with the repetition of the tweak and LargeBlock is an inefficient construction as it requires $\ell + 2$ keys to process a message of ℓ blocks. Thus, we are left with ZCZ, KohiNoor, and DaryaiNoor to compare with our constructions. Implementation results in Table 2 show that ZCZ performs worse than both of our constructions while the performance of KohiNoor and DaryaiNoor is comparable to ours. Therefore, we focus on comparing our proposals with KohiNoor and DaryaiNoor to provide a clearer understanding of why our schemes would be preferable over the GEM instances.

Our modes use the ISO-standard Deoxys-BC that has been extensively analyzed and also been selected in the Deoxys CAESAR selection for high-security applications. On the contrary, ButterKnife, the primitive in KohiNoor has not stood the test of time yet; though, both use the AES round function and benefit from hardware support.

The overall structure of GEM would become a four-round Feistel network with AES if the variable part of the message would be empty. Thus, while similar, the domains of GEM and HCTR+ differ slightly, with the former requiring messages of at least $4n$ bits. Otherwise, it would have only birthday-bound security. On the other hand, HCTR+ supports also $2n$-bit messages out of the box, supporting the requirement of NIST's proposal [CDD+24].

On a side note, our instantiations leave options to the user for a more refined analysis in the context of e.g. quantum-aided analysis in future work. ZHCTR+ with ZHASH+ offers a little higher throughput compared to PHCTR+ with PHASH+ when messages become longer than 1 KB, as we will show for our implementations. On the other hand, ZHASH+ masks the inputs to the TBC, which is fine for classical security but may be susceptible to period-finding attacks for quantum-aided adversaries (see, e.g., the studies in [KLLN16]). In contrast, our PHASH+-based instantiation uses only tweak updates. Thus, it *has the potential* to offer also some security in quantum settings, similar to how Bhaumik et al. have illustrated it for QCB [BBC+21]. However, a precise analysis of PHCTR+ in quantum settings is an interesting future work.

Note: We would like to point out that we have updated the ZHASH+ construction as described in the ToSC proceeding version [DDG+25] as well as in the old ePrint version (20250822:123141). We thank Takuto Takami, Takahiro Kaneko and Tetsu Iwata for identifying an issue in the older version of the ZHASH+ construction. They demonstrated in a private communication that the older ZHASH+ construction does not achieve almost-xor-universal security, and that directly leads to an distinguishing attack on the ZHCTR+ construction. In this version of the paper, we propose a simple modification to the ZHASH+ construction (cf. Fig. 4) to mitigate this attack. In particular, we have removed the domain separation from line 9 of the older ZHASH+ function and use an independent n-bit hash key $K_{h,2}$ in line 9 of the function. This revision incurs a negligible change in the security bound of the ZHCTR+ construction from its older version. Nevertheless, the security

proof of the almoxt-xor-universality of the ZHASH+ construction remains unchanged. However, due to this change, there is a minimal updation in the implementation result of the construction (cf. Sect. 7).

1.3 Organization

Sect. 2 is comprised of the notations and security definitions that we will use in the remainder of this work. In Sect. 3, we provide a formal specification of our design HCTR+, justifying its design rationale and presenting the main security result that the construction achieves optimal security. We provide its proof 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 and prove their almost-XOR-universality in Sect. 6. Finally, in Sect. 7, 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 $\mathcal{X}$, $X \xleftarrow{\$} \mathcal{X}$ denotes that X is sampled uniformly at random from $\mathcal{X}$. We write $X \leftarrow Y$ to denote that Y is assigned to 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 $1 \le i \le \ell - 1$ and $1 \le |X_\ell| \le n$. We denote it as $(X_1, X_2, \dots, X_\ell) \xleftarrow{n} \mathsf{Parse}(X)$. For a fixed natural number n, we define a padding function $\mathsf{Pad}_n : \{0,1\}^* \to \{0,1\}^*$ as follows:

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

For any binary string $X \in \{0,1\}^*$ such that $|X| \ge n$, we write $(X_1, X_2) \xleftarrow{n} X$ 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^i_a 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 $1 \le b \le a$, $(a)_b$ denotes $a(a-1)\dots(a-b+1)$, where $(a)_0 = 1$ by convention. For any function $\Phi : \mathcal{X} \to \{0,1\}^{2n}$, 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 $\mathsf{Msb}_i(S)$ and $\mathsf{Lsb}_i(S)$ to denote the most significant and the least significant i bits of a binary string S respectively, such that $|S| \ge i$.

2.1 Tweakable Block Cipher

A *tweakable block cipher* (TBC) is a mapping $\widetilde{\mathsf{E}} : \mathcal{K} \times \mathcal{T} \times \mathcal{X} \to \mathcal{X}$, 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 $T \in \mathcal{T}$, $X \mapsto \widetilde{\mathsf{E}}(K, T, X)$ is a permutation over $\mathcal{X}$. We denote with $\mathsf{TBC}(\mathcal{K}, \mathcal{T}, \mathcal{X})$ 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 $\widetilde{\mathsf{P}} : \mathcal{T} \times \mathcal{X} \to \mathcal{X}$ such that for all tweaks $T \in \mathcal{T}$, $X \mapsto \widetilde{\mathsf{P}}(T, X)$ is a permutation over $\mathcal{X}$. We write $\mathsf{TP}(\mathcal{T}, \mathcal{X})$ 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 $\widetilde{\mathsf{P}}$ 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 $\mathsf{Rng}(\Pi[T, \cdot])$ as $\{Y : \exists X, \Pi[T, X] = Y\}$. Similarly, for each tweak $T \in \mathcal{T}$, we define the set $\mathsf{Dom}(\Pi[T, \cdot])$ 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 $\mathsf{TP}(\mathcal{T}, \mathcal{X})$. 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 $\widetilde{\mathsf{E}} : \mathcal{K} \times \mathcal{T} \times \mathcal{X} \to \mathcal{X}$ 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 $\widetilde{\mathsf{E}}$, $\mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{\mathsf{E}}}(\mathcal{A})$, is defined as

$$\left| \Pr\left[K \leftarrow\!\$\, \mathcal{K},\ \mathcal{A}^{\widetilde{\mathsf{E}}_K(\cdot,\cdot), \widetilde{\mathsf{E}}_K^{-1}(\cdot,\cdot)} = 1\right] - \Pr\left[\widetilde{\mathsf{P}} \leftarrow\!\$\, \mathsf{TP}(\mathcal{T}, \mathcal{X}),\ \mathcal{A}^{\widetilde{\mathsf{P}}(\cdot,\cdot), \widetilde{\mathsf{P}}^{-1}(\cdot,\cdot)} = 1\right] \right|.$$

We say that $\widetilde{\mathsf{E}}$ is $(q, \mathtt{t}, \epsilon)$-secure if the maximum strong tweakable pseudorandom permutation advantage of $\widetilde{\mathsf{E}}$ is ϵ where the maximum is taken over all distinguishers $\mathcal{A}$ that makes q queries to its oracle and runs time at most $\mathtt{t}$. For a tweakable block cipher, one typically considers $\mathcal{X}$ 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.[3]

Tweakable Enciphering Scheme: If the tweak size and the input size of $\widetilde{\mathsf{E}}$ is of variable length, then we call $\widetilde{\mathsf{E}}$ 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 "$(q, \ell_{\max}, \mathtt{t})$ *chosen-plaintext chosen-ciphertext adversary* $\mathcal{A}$ *against the stprp security of some tweakable enciphering scheme* C", we mean that $\mathcal{A}$ makes q queries such that the maximum message length of each query is $\ell_{\max}$ with maximum running time is at most $\mathtt{t}$.

2.2 Almost XOR Universal Hash Function

Let $\mathsf{H} : \mathcal{K}_h \times \mathcal{X} \to \{0,1\}^n$ 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

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

2.3 H-Coefficient Technique

The 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

[3] Sampling uniformly a tweakable random permutation from the set of all tweakable permutations is not mathematically well-defined when the tweak space or the input space is infinite. In that case, our notation means that a random tweakable permutation is simulated using lazy sampling.

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 $\mathcal{O}_{\mathrm{re}}$ or the ideal oracle $\mathcal{O}_{\mathrm{id}}$. Let $\Theta = \Theta_{\mathtt{good}} \sqcup \Theta_{\mathtt{bad}}$ be some partition of the set of attainable transcripts Θ. For every good attainable transcript $\tau \in \Theta_{\mathtt{good}}$, let*

$$\frac{\Pr[\mathsf{X}_{\mathrm{re}} = \tau]}{\Pr[\mathsf{X}_{\mathrm{id}} = \tau]} \geq 1 - \epsilon_{\mathrm{good}},$$

for some $\epsilon_{\mathrm{good}} \geq 0$, and there exists $\epsilon_{\mathrm{bad}} \geq 0$ such that $\Pr[\mathsf{X}_{\mathrm{id}} \in \Theta_{\mathtt{bad}}] \leq \epsilon_{\mathrm{bad}}$ holds. Then,

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

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 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 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: the

KDF algorithm first takes a master key K as input and derives five uniform random keys $(K_h, K_1, K_2, K_3, K_4)$. 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 a $2n$-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 three round TBC based Luby Rackoff construction, called TLR3, which was 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. Let the output of TLR3 be (V_1, V_2). We tap two intermediate n-bit values of the construction U_2 and V_2, as shown in Fig. 2, and take their xor to derive the value W which is used as an input in a TBC based counter mode of encryption. On the other hand, Z is used as a tweak in the underlying TBC of the counter mode encryption. 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 denote 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.

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. $(K_h, K_1, K_2, K_3, K_4) \leftarrow \mathsf{KDF}(K)$;	2. $(K_h, K_1, K_2, K_3, K_4) \leftarrow \mathsf{KDF}(K)$;
3. $U_1\|U_2 \leftarrow (M_1\|M_2) \oplus \mathsf{H}_{K_h}(M_R\|T)$;	3. $V_1\|V_2 \leftarrow (C_1\|C_2) \oplus \mathsf{H}_{K_h}(C_R\|T)$;
4. $Z \leftarrow \widetilde{\mathsf{E}}_{K_1}(U_2, U_1)$; $V_2 \leftarrow \widetilde{\mathsf{E}}_{K_2}(Z, U_2)$;	4. $Z \leftarrow \widetilde{\mathsf{E}}^{-1}_{K_3}(V_2, V_1)$; $U_2 \leftarrow \widetilde{\mathsf{E}}^{-1}_{K_2}(Z, V_2)$;
5. $V_1 \leftarrow \widetilde{\mathsf{E}}_{K_3}(V_2, Z)$; $W \leftarrow U_2 \oplus V_2$;	5. $U_1 \leftarrow \widetilde{\mathsf{E}}^{-1}_{K_1}(U_2, Z)$; $W \leftarrow U_2 \oplus V_2$;
6. **for** $i = 1$ to $\lceil \|M_R\|/n \rceil$	6. **for** $i = 1$ to $\lceil \|C_R\|/n \rceil$
7. $\quad S_i \leftarrow \widetilde{\mathsf{E}}_{K_4}(Z \oplus \langle i \rangle_n, W)$;	7. $\quad S_i \leftarrow \widetilde{\mathsf{E}}_{K_4}(Z \oplus \langle i \rangle_n, W)$;
8. $S \leftarrow S_1\| \cdots \|S_{\lceil \|M_R\|/n \rceil}$;	8. $S \leftarrow S_1\| \cdots \|S_{\lceil \|C_R\|/n \rceil}$;
9. $C_R \leftarrow \mathsf{Msb}_{\|M_R\|}(S) \oplus M_R$;	9. $M_R \leftarrow \mathsf{Msb}_{\|C_R\|}(S) \oplus C_R$;
10. $(C_1, C_2) \leftarrow (V_1, V_2) \oplus \mathsf{H}_{K_h}(C_R\|T)$;	10. $(M_1, M_2) \leftarrow (U_1, U_2) \oplus \mathsf{H}_{K_h}(M_R\|T)$;
11. **return** $(C_1\|C_2\|C_R)$;	11. **return** $(M_1\|M_2\|M_R)$;

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

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 $\widetilde{Z} = (Z \oplus \langle 1 \rangle_n, \ldots, Z \oplus \langle \ell - 2 \rangle_n)$,

where $\ell = \lceil |M|/n \rceil$. Given such a sequence $\widetilde{Z}$, a key K, and an n-bit input W, we define the tweakable block cipher based counter mode encryption as follows:

$$\widetilde{\mathsf{CTRT}}[\widetilde{\mathsf{E}}_K](\widetilde{Z}, W, \ell) \stackrel{\mathsf{def}}{=} \|_{i=1}^{\ell-2} \Big(\widetilde{\mathsf{E}}_K(Z \oplus \langle i \rangle_n, W) \Big).$$

3.2 Security Result of HCTR+

In this section, we state the security result of HCTR+. In specific, we state that if $\widetilde{\mathsf{E}}$ is an (n, n) tweakable block cipher, $\mathsf{H} = (\mathsf{H}[1], \mathsf{H}[2])$ is an ϵ-axu $2n$-bit keyed hash function such that each $\mathsf{H}[i]$ 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+.

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

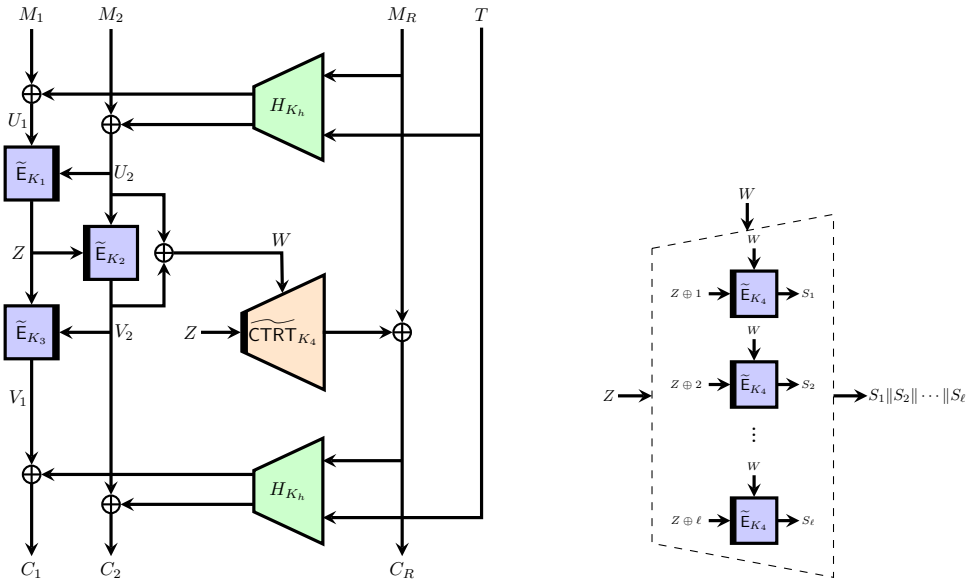
$$\mathbf{Adv}^{\mathrm{STPRP}}_{\mathsf{HCTR}+[\widetilde{\mathsf{E}},\mathsf{H}]}(\mathcal{A}) \leq 4\mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{\mathsf{E}}}(\mathcal{A}') + q^2\epsilon + \frac{q^2\epsilon_2}{2^n} + \frac{2\min\{4q^2\ell_{\max}, \sigma^2\}}{2^{2n}} + \frac{3q^2}{2^{2n}},$$

where $q \leq 2^{n-1}$, σ is the total number of message blocks queried, $\ell_{\max}$ is the maximum number of message blocks in any single query, and $\mathtt{t}' = O(\mathtt{t} + \sigma + qt_H)$, t_H denotes the time for computing the hash function 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 three round TBC based Luby Rackoff construction TLR3 on the left side of the construction and a TBC based counter mode of encryption $\widetilde{\mathsf{CTRT}}$ 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 tweakable SPRP. Although TLR3 ensures optimal tweakable strong pseudorandom permutation security, we need to generate $2n$ bit random values from the TLR3 construction. We choose the output of the first TBC call of the TLR3 construction as a tweak and the xor of the input and the output of the second TBC call of the TLR3 construction as an input to the $\widetilde{\mathsf{CTRT}}$ construction. Since the input size of TLR3 is $2n$ bits, we use a $2n$ 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 TLR3. Note that, we are not externally providing tweaks to the underlying TBC of the TLR3 construction. Instead, the intermediate n-bit state of the construction serves as a tweak. Moreover, we tap two such intermediate n-bit states Z and $W = (U_2 \oplus V_2)$ that is fed to the $\widetilde{\mathsf{CTRT}}$ construction, where $(Z \oplus \langle 1 \rangle_n, Z \oplus \langle 2 \rangle_n, \ldots, Z \oplus \langle \ell - 2 \rangle_n)$ are used as tweak for the underlying TBC of $\widetilde{\mathsf{CTRT}}$ and W is used as the input of the TBC. We would like to emphasize that in



(a) $\mathsf{HCTR}+$ (b) $\widetilde{\mathsf{CTRT}}_{K_4}$

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

$\widetilde{\mathsf{CTRT}}$ 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].

Remark 1. We note that $\widetilde{\mathsf{CTRT}}$ mode used in the right part of the construction requires $2n$-bit entropy, which should be provided from the left side of the construction. Now, other than our choice of Z and W, there are internal variables U_1, U_2, V_1, V_2. An adversary can enforce a collision between any of these variables with an appropriate choice of tweaks and messages (or tweaks and ciphertexts). This phenomenon leads us not to feed any of these variables directly as the tweak or input in $\widetilde{\mathsf{CTRT}}$.

4 Proof of Theorem 2

Let $\mathcal{A}$ be a $(q, \ell_{\max}, \mathtt{t})$ distinguisher against the strong tweakable pseudorandom permutation security of $\mathsf{HCTR}+[\widetilde{\mathsf{E}}, \mathsf{H}]$. Initially, we replace the four independently keyed tweakable block ciphers $\widetilde{\mathsf{E}}_{K_1}, \widetilde{\mathsf{E}}_{K_2}, \widetilde{\mathsf{E}}_{K_3}$, and $\widetilde{\mathsf{E}}_{K_4}$ with four independently sampled tweakable random permutations $\widetilde{\mathsf{P}} := (\widetilde{\mathsf{P}}_1, \widetilde{\mathsf{P}}_2, \widetilde{\mathsf{P}}_3, \widetilde{\mathsf{P}}_4)$ at the cost of the strong tweakable pseudorandom permutation advantage of the underlying TBC. Therefore, we have

$$\mathbf{Adv}^{\mathrm{STPRP}}_{\mathsf{HCTR}+[\widetilde{\mathsf{E}},\mathsf{H}]}(\mathcal{A}) \leq 4\mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{\mathsf{E}}}(\mathcal{A}) + \underbrace{\mathbf{Adv}^{\mathrm{STPRP}}_{\mathsf{HCTR}+^*[\widetilde{\mathsf{P}},\mathsf{H}]}(\mathcal{A})}_{\delta}$$

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

δ. Note that, we have

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

where the first probability is taken over the randomness of $\widetilde{\mathsf{P}}_i \leftarrow\!\$\ \mathsf{TP}(\{0,1\}^n, \{0,1\}^n)$ for $i \in [4]$ and the second probability is computed over the randomness of $\mathsf{PP} \leftarrow\!\$\ \mathsf{TP}(\{0,1\}^*, \{0,1\}^*)$. Moreover, the maximum is taken over non-trivial adversaries.[4] Hence, δ can not be larger than the advantage of the best non-trivial adversary between the two worlds $(\mathsf{HCTR}+^*, (\mathsf{HCTR}+^*)^{-1})$ and $(\mathsf{PP}, (\mathsf{PP})^{-1})$. This 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 the real world $(\mathsf{HCTR}+^*, (\mathsf{HCTR}+^*)^{-1})$ or with the ideal world $(\mathsf{PP}, (\mathsf{PP})^{-1})$. In the online phase of the interaction, $\mathcal{A}$ obtains either $C = \mathsf{HCTR}+^*(T, M)$ in the real world or $\mathsf{PP}(T, M)$ 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 $M = (\mathsf{HCTR}+^*)^{-1}(T, C)$ or $(\mathsf{PP})^{-1}(T, C)$ 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_h. In the real world, K_h is the actual hash key used in the construction, whereas in the ideal world, a dummy hash key K_h is sampled uniformly and independently from the hash key space. This enables the adversary to compute the pair of tuples (U_1^i, U_2^i) and (V_1^i, V_2^i), where

$$(U_1^i \| U_2^i) = \mathsf{H}_{K_h}(M_R^i \| T^i) \oplus (M_1^i \| M_2^i),\ (V_1^i \| V_2^i) = \mathsf{H}_{K_h}(C_R^i \| T^i) \oplus (C_1^i \| C_2^i).$$

The distinguisher is also provided with intermediate variables as additional information. In the real world, the distinguisher is provided with the pair of tuples $(Z^1, Z^2, \ldots, Z^q)$ and $(W^1, W^2, \ldots, W^q)$, which have been generated in the construction $\mathsf{HCTR}+^*$. However, in the ideal world, these additional informations are required to be generated. We sample the q tuple $(Z^1, Z^2, \ldots, Z^q)$ in the ideal world using a sampler S whose objective would be to sample the tuple in such a way that it becomes close to the distribution of the tuple $(Z^1, Z^2, \ldots, Z^q)$ generated in the real world. On the other hand, the tuple $(W^1, W^2, \ldots, W^q)$ is generated in the ideal world as $W^i = U_2^i \oplus V_2^i$ for all $i \in [q]$. To carry out the sampling, S will simulate the tweakable random permutations $\widetilde{\mathsf{P}}_1$ and $\widetilde{\mathsf{P}}_3^{-1}$ used in the real construction with the help of the following tables $\Pi_1[\cdot,\cdot]$ and $\Pi_3^{-1}[\cdot,\cdot]$. In other words, $\widetilde{\mathsf{P}}_1$ will be simulated using the table $\Pi_1[\cdot,\cdot]$ and $\widetilde{\mathsf{P}}_3^{-1}$ will be simulated using the table $\Pi_3^{-1}[\cdot,\cdot]$. The tables are initialized as empty. The sampler S runs as follows.

[4] A non-trivial adversary is one who does not repeat queries.

For i-th encryption query,

- 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:

$$Z^i \coloneqq \Pi_1[U_2^i, U_1^i] \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathsf{Rng}(\Pi_1[U_2^i, \cdot])$$

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

For i-th decryption query,

- Checks if $\Pi_3^{-1}[V_2^i, V_1^i]$ has already been set. If it has not been set, S samples Z^i as follows:

$$Z^i \coloneqq \Pi_3^{-1}[V_2^i, V_1^i] \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathsf{Rng}(\Pi_3^{-1}[V_2^i, \cdot])$$

  and set $Z^i \leftarrow \Pi_3^{-1}[V_2^i, V_1^i]$.

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 $\tau = (\tau', K_h)$, where

$$\tau' = \Big((T^1, M^1, C^1, Z^1, W^1), \ldots, (T^q, M^q, C^q, Z^q, W^q)\Big).$$

4.2 Defining and Bounding Bad Transcripts

Let $[q_{\mathsf{en}}]$ (resp. $[q_{\mathsf{de}}]$) be the set of encryption (resp. decryption) queries and q denotes the total number of queries, i.e., $q = q_{\mathsf{en}} + q_{\mathsf{de}}$. 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 $\tau = (\tau', K_h)$ is bad, if there exist two queries i and j (w.l.o.g $j < i$), such that either of the following holds:

1. $\mathsf{Bad1}$: For any $i \neq j \in [q]$, such that $(M_1^i \| M_2^i) \oplus \mathsf{H}_{K_h}(M_R^i \| T^i) = (M_1^j \| M_2^j) \oplus \mathsf{H}_{K_h}(M_R^j \| T^j)$.
2. $\mathsf{Bad2}$: For any $i \neq j \in [q]$, such that $(C_1^i \| C_2^i) \oplus \mathsf{H}_{K_h}(C_R^i \| T^i) = (C_1^j \| C_2^j) \oplus \mathsf{H}_{K_h}(C_R^j \| T^j)$.
3. $\mathsf{Bad3}$: For $i \in [q_{\mathsf{de}}]$, $j \in [q]$ and $j < i$ such that $Z^i = Z^j, M_2^i \oplus \mathsf{H}_{K_h}[2](M_R^i \| T^i) = M_2^j \oplus \mathsf{H}_{K_h}[2](M_R^j \| T^j)$.
4. $\mathsf{Bad4}$: For $i \in [q_{\mathsf{en}}]$, $j \in [q]$ and $j < i$ such that $Z^i = Z^j, C_2^i \oplus \mathsf{H}_{K_h}[2](C_R^i \| T^i) = C_2^j \oplus \mathsf{H}_{K_h}[2](C_R^j \| T^j)$.
5. $\mathsf{Bad5}$: For any $i > j \in [q]$ and $\alpha \in [\ell_i - 2], \beta \in [\ell_j - 2]$, such that $Z^i \oplus \langle \alpha \rangle_n = Z^j \oplus \langle \beta \rangle_n$ and $W^i = W^j$:

6. Bad6: For any $i \neq j \in [q]$ and $\alpha \in [\ell_i - 2], \beta \in [\ell_j - 2]$, such that $Z^i \oplus \langle \alpha \rangle_n = Z^j \oplus \langle \beta \rangle_n$ and $M^i_{\alpha+2} \oplus C^i_{\alpha+2} = M^j_{\beta+2} \oplus C^j_{\beta+2}$.

Let $\Theta_{\mathtt{bad}}$ denotes the set of all attainable transcripts τ such that it satisfies one of the above conditions and the event Bad denotes

$$\mathtt{Bad} \coloneqq \mathsf{Bad1} \vee \mathsf{Bad2} \vee (\mathsf{Bad3} \mid \neg\mathsf{Bad2}) \vee (\mathsf{Bad4} \mid \neg\mathsf{Bad1}) \vee \mathsf{Bad5} \vee \mathsf{Bad6}.$$

Following Lemma establishes 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*

$$\epsilon_{\mathtt{bad}} = \Pr[\mathsf{X}_{\mathrm{id}} \in \Theta_{\mathtt{bad}}] \leq q^2\epsilon + \frac{q^2\epsilon_2}{2^n} + \frac{2\min\{4q^2\ell_{\max}, \sigma^2\}}{2^{2n}} + \frac{2q^2}{2^{2n}}.$$

Proof. We individually upper bound the probability of the events and then apply the union bound to derive the bound based on the probability that a transcript is bad in the ideal world.

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:

$$\mathsf{H}_{K_h}(M^i_R \| T^i) \oplus \mathsf{H}_{K_h}(M^j_R \| T^j) = (M^i_1 \| M^i_2) \oplus (M^j_1 \| M^j_2).$$

If $(M^i_R \| T^i) = (M^j_R \| T^j)$, then the event Bad1 is boiled down to $(M^i_1 \| M^i_2) = (M^j_1 \| M^j_2)$. Since the adversary is non-trivial, the probability of the event Bad1 holds, when $(M^i_R \| T^i) = (M^j_R \| T^j)$, is zero. On the other hand, if $(M^i_R \| T^i) \neq (M^j_R \| T^j)$, then 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 H. Therefore, by summing over all possible choices of (i, j), we have

$$\Pr[\mathsf{Bad1}] \leq \frac{q^2\epsilon}{2}. \tag{1}$$

Bounding Bad2: We bound this event similar to bounding as Bad1. For a fixed choice of indices $i, j \in [q]$, the probability of the event Bad2 is boiled down to the following equation:

$$\mathsf{H}_{K_h}(C^i_R \| T^i) \oplus \mathsf{H}_{K_h}(C^j_R \| T^j) = (C^i_1 \| C^i_2) \oplus (C^j_1 \| C^j_2).$$

If $(C^i_R \| T^i) = (C^j_R \| T^j)$, then the event Bad2 is boiled down to $(C^i_1 \| C^i_2) = (C^j_1 \| C^j_2)$. Since the adversary is non-trivial, the probability of the event Bad2 holds, when $(C^i_R \| T^i) = (C^j_R \| T^j)$, is zero. On the other hand, if $(C^i_R \| T^i) \neq (C^j_R \| T^j)$, then 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 H. Therefore, by summing over all possible choices of (i, j), we have

$$\Pr[\mathsf{Bad2}] \leq \frac{q^2\epsilon}{2}. \tag{2}$$

Bounding Bad3|¬Bad2: For a fixed choice of distinct indices $i \in [q_{\mathsf{de}}]$ and $j \in [q]$ with $j < i$, we bound the probability of the event

$$Z^i = Z^j, M^i_2 \oplus \mathsf{H}_{K_h}[2](M^i_R \| T^i) = M^j_2 \oplus \mathsf{H}_{K_h}[2](M^j_R \| T^j)$$

in the following two cases: (a) if $(M^i_R \| T^i) = (M^j_R \| T^j)$, then the above event gets boiled down to the following:

$$Z^i = Z^j, M^i_2 = M^j_2.$$

Since, i-th query is a decryption query, $M_2^i = M_2^j$ holds with probability at most $1/2^{n-1}$. On the other hand, to bound the probability of the event $Z^i = Z^j$, we have the following two cases: (i) if $V_i^2 = V_j^2$, then the event $Z^i = Z^j$ holds with probability 0. On the other hand, (ii) if $V_i^2 \neq V_j^2$, then the event $Z^i = Z^j$ holds with probability 2^{-n} due to the sampling of Z^i as the i-th query is a decryption query.

(b) The other case is $(M_R^i \| T^i) \neq (M_R^j \| T^j)$. In that case, the event $M_2^i \oplus \mathsf{H}_{K_h}[2](M_R^i \| T^i) = M_2^j \oplus \mathsf{H}_{K_h}[2](M_R^j \| T^j)$ is upper bounded by ϵ_2, where ϵ_2 is the almost-xor-universal advantage of $\mathsf{H}[2]$. On the other hand, to bound the probability of the event $Z^i = Z^j$, we have the following two cases as before: (i) if $V_i^2 = V_j^2$, then the event $Z^i = Z^j$ holds with probability 0. On the other hand, (ii) if $V_i^2 \neq V_j^2$, then the event $Z^i = Z^j$ holds with probability 2^{-n} due to the sampling of Z^i as the i-th query is a decryption query. Therefore, by summing over all possible choices of (i, j), we have

$$\Pr[\mathsf{Bad3} | \neg \mathsf{Bad2}] \leq \frac{q^2}{2^{2n}} + \frac{q^2 \epsilon_2}{2^{n+1}}. \tag{3}$$

Bounding Bad4|¬Bad1: For a fixed choice of distinct indices $i \in [q_{\mathsf{en}}]$ and $j \in [q]$ with $j < i$, we bound the probability of the event

$$Z^i = Z^j, C_2^i \oplus \mathsf{H}_{K_h}[2](C_R^i \| T^i) = C_2^j \oplus \mathsf{H}_{K_h}[2](C_R^j \| T^j)$$

in the following two cases: (a) if $(C_R^i \| T^i) = (C_R^j \| T^j)$, then the above event gets boiled down to the following:

$$Z^i = Z^j, C_2^i = C_2^j.$$

Since, i-th query is an encryption query, $C_2^i = C_2^j$ holds with probability at most $1/2^{n-1}$. On the other hand, to bound the probability of the event $Z^i = Z^j$, we have the following two cases: (i) if $U_i^2 = U_j^2$, then the event $Z^i = Z^j$ holds with probability 0. On the other hand, (ii) if $U_i^2 \neq U_j^2$, then the event $Z^i = Z^j$ holds with probability 2^{-n} due to the sampling of Z^i as the i-th query is an encryption query.

(b) The other case is $(C_R^i \| T^i) \neq (C_R^j \| T^j)$. In that case, the event $C_2^i \oplus \mathsf{H}_{K_h}[2](C_R^i \| T^i) = C_2^j \oplus \mathsf{H}_{K_h}[2](C_R^j \| T^j)$ is upper bounded by ϵ_2, where ϵ_2 is the almost-xor-universal advantage of $\mathsf{H}[2]$. On the other hand, to bound the probability of the event $Z^i = Z^j$, we have the following two cases as before: (a) if $U_i^2 = U_j^2$, then the event $Z^i = Z^j$ holds with probability 0. On the other hand, (b) if $U_i^2 \neq U_j^2$, then the event $Z^i = Z^j$ holds with probability 2^{-n} due to the sampling of Z^i as the i-th query is an encryption query. Therefore, by summing over all possible choices of (i, j), we have

$$\Pr[\mathsf{Bad4} | \neg \mathsf{Bad1}] \leq \frac{q^2}{2^{2n}} + \frac{q^2 \epsilon_2}{2^{n+1}}. \tag{4}$$

Bounding Bad5: For a fixed choice of indices $i > j \in [q]$, $\alpha \in [\ell_i - 2]$, $\beta \in [\ell_j - 2]$, we bound the event the event

$$Z^i \oplus \langle \alpha \rangle_n = Z^j \oplus \langle \beta \rangle_n, W^i = W^j.$$

Note that, due to randomness of Z^i, we have $Z^i \oplus \langle \alpha \rangle_n = Z^j \oplus \langle \beta \rangle_n$ is upper bounded by $1/(2^n - q) \leq 2/2^n$, using the assumption $q \leq 2^{n-1}$. Now, the event $W^i = W^j$ is boiled down to the equation

$$\begin{aligned} &M_2^i \oplus \mathsf{H}_{K_h}[2](M_R^i \| T^i) \oplus C_2^i \oplus \mathsf{H}_{K_h}[2](C_R^i \| T^i) \\ &= M_2^j \oplus \mathsf{H}_{K_h}[2](M_R^j \| T^j) \oplus C_2^j \oplus \mathsf{H}_{K_h}[2](C_R^j \| T^j). \end{aligned}$$

The probability of the above event is upper bounded by $1/2^{n-1}$ due to the randomness of C_2^i (in case i-th query is an encryption query) or due to the randomness of M_2^i (in case i-th query

being a decryption query). Note that, for a fixed indices i and j, $\alpha \leq \ell_i \leq \ell, \beta \leq \ell_j \leq \ell$. Therefore, for a fixed indices i and j, the total number of possible choices for α, β is $2\ell_{\max}$. On the other hand we can also bound (i, α) and (j, β) pairs by $\binom{\sigma}{2}$ ways. Therefore, by summing over all possible choices of indices i, j, α, β we have

$$\Pr[\mathsf{Bad5}] \leq \frac{\min\{4q^2\ell_{\max}, \sigma^2\}}{2^{2n}}. \tag{5}$$

Bounding Bad6: Similar to bounding the event $\mathsf{Bad5}$, 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 $2/2^n$. The second event, i.e., $M^i_{\alpha+2} \oplus C^i_{\alpha+2} = M^j_{\beta+2} \oplus C^j_{\beta+2}$ is upper bounded by $1/2^{n-1}$, due to the randomness of $C^i_{\alpha+2}$ (in case i-th query being an encryption query) or due to the randomness of $M^i_{\alpha+2}$ (in case i-th query being a decryption query). Therefore, by summing over all possible choices of indices i, j, α, β we have

$$\Pr[\mathsf{Bad6}] \leq \frac{\min\{4q^2\ell_{\max}, \sigma^2\}}{2^{2n}}. \tag{6}$$

The result follows by applying the union bound on the probability of the above events as stated in Eqn. (1) through Eqn. (6). □

4.3 Analysis of Good Transcripts

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

Lemma 2. *Let $\tau = (\tau', K_h)$ be a good transcript, where*

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

Then

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

Proof. Let $\tau \in \Theta_{\mathsf{good}}$ 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 $\widetilde{\mathsf{P}}_i$, for $i = 1, \ldots, 4$.
- Let $\mathcal{T}_i^{\mathsf{en}}$ (resp. $\mathcal{T}_i^{\mathsf{de}}$) be the set of distinct tweaks queried to $\widetilde{\mathsf{P}}_i$ during encryption (resp. decryption), for $i = 1, \ldots, 3$.
- Let $\alpha_T, \beta_T, \gamma_T$ and δ_T be the number of calls to $\widetilde{\mathsf{P}}_i$ with tweak $T \in \mathcal{T}_i$ for $i = 1, \ldots, 4$ respectively.
- Let $\alpha_T^{\mathsf{en}}, \beta_T^{\mathsf{en}}$, and γ_T^{en} (resp. $\alpha_T^{\mathsf{de}}, \beta_T^{\mathsf{de}}$, and γ_T^{de}) be the number of calls to $\widetilde{\mathsf{P}}_i$ with tweak $T \in \mathcal{T}_i$ during encryption (resp. decryption), for $i = 1, \ldots, 3$ respectively.
- Let L be the number of distinct input lengths to the construction and we denote those input lengths with $\ell_1, \ell_2, \ldots, \ell_L$. Let $\ell_{\min} = \min\{\ell_1, \ell_2, \ldots, \ell_L\}$
- 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

$$\sum_{T \in \mathcal{T}_1} \alpha_T = \sum_{T \in \mathcal{T}_2} \beta_T = \sum_{T \in \mathcal{T}_3} \gamma_T = q \text{ and } \sum_{T \in \mathcal{T}_1^{\mathsf{de}}} \alpha_T^{\mathsf{de}} + \sum_{T \in \mathcal{T}_3^{\mathsf{en}}} \gamma_T^{\mathsf{en}} \leq q. \tag{7}$$

Without loss of generality, we assume that each ℓ_i is a multiple of n i.e. each input length ℓ_i consists of κ_i blocks and $\ell_i = n\kappa_i$. 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_h has been sampled uniformly at random from the hash key space $\mathcal{K}_h$. Since we consider τ to be 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

$$\Pr[\mathsf{X}_{\mathrm{re}} = \tau] = \frac{1}{|\mathcal{K}_h|} \times \prod_{T \in \mathcal{T}_1} \frac{1}{(2^n)_{\alpha_T}} \times \prod_{T \in \mathcal{T}_2} \frac{1}{(2^n)_{\beta_T}} \times \prod_{T \in \mathcal{T}_3} \frac{1}{(2^n)_{\gamma_T}} \times \prod_{T \in \mathcal{T}_4} \frac{1}{(2^n)_{\delta_T}}. \tag{8}$$

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 $(\mathsf{PP})^{-1}$ of a decryption query $C = C_1 \| C_2 \| C_R$. After the interaction is over, a dummy hash key K_h has been sampled uniformly at random from the hash key space $\mathcal{K}_h$. Finally, the intermediate random variable Z has been sampled according to the procedures described in Sect. 4.1.1. Thus, the ideal interpolation probability for the good transcript τ is

$$\Pr[\mathsf{X}_{\mathrm{id}} = \tau] = \frac{1}{|\mathcal{K}_h|} \times \prod_{i=1}^{L} \prod_{j=1}^{t_i} \frac{1}{(2^{\ell_i})_{q_{i,j}}} \times \prod_{T \in \mathcal{T}_1^{\mathsf{en}}} \frac{1}{(2^n)_{\alpha_T^{\mathsf{en}}}} \times \prod_{T \in \mathcal{T}_3^{\mathsf{de}}} \frac{1}{(2^n)_{\gamma_T^{\mathsf{de}}}}. \tag{9}$$

4.3.3 Ratio of Real to Ideal Interpolation Probability

By taking ratio of Eqn. (8) to Eqn. (9), we have the following

$$\begin{aligned}\frac{\Pr[\mathsf{X}_{\mathrm{re}}=\tau]}{\Pr[\mathsf{X}_{\mathrm{id}}=\tau]} &= \frac{\prod_{i=1}^{L}\prod_{j=1}^{t_i}(2^{\ell_i})_{q_{i,j}} \times \prod_{T\in\mathcal{T}_1^{\mathsf{en}}}(2^n)_{\alpha_T^{\mathsf{en}}} \times \prod_{T\in\mathcal{T}_3^{\mathsf{de}}}(2^n)_{\gamma_T^{\mathsf{de}}}}{\prod_{T\in\mathcal{T}_1}(2^n)_{\alpha_T} \times \prod_{T\in\mathcal{T}_2}(2^n)_{\beta_T} \times \prod_{T\in\mathcal{T}_3}(2^n)_{\gamma_T} \times \prod_{T\in\mathcal{T}_4}(2^n)_{\delta_T}} \\ &= \frac{\prod_{i=1}^{L}\prod_{j=1}^{t_i}(2^{\ell_i})_{q_{i,j}}}{\prod_{T\in\mathcal{T}_1^{\mathsf{de}}}(2^n)_{\alpha_T^{\mathsf{de}}} \times \prod_{T\in\mathcal{T}_3^{\mathsf{en}}}(2^n)_{\gamma_T^{\mathsf{en}}} \times \prod_{T\in\mathcal{T}_2}(2^n)_{\beta_T} \times \prod_{T\in\mathcal{T}_4}(2^n)_{\delta_T}} \\ &\geq \frac{\prod_{i=1}^{L}\prod_{j=1}^{t_i}(2^{\ell_i})_{q_{i,j}}}{2^{2nq} \times \prod_{T\in\mathcal{T}_4}(2^n)^{\delta_T}} \quad \text{[using the fact } (2^n)_x \leq (2^n)^x \text{ and Eqn. 7]}\end{aligned}$$

Applying the fact that $\sum_{T\in\mathcal{T}_4}\delta_T \leq (\sum_{i=1}^{L}\sum_{j=1}^{t_i}\kappa_i\cdot q_{i,j}) - 2q$, we get

$$\begin{aligned}\frac{\Pr[\mathsf{X}_{\mathrm{re}}=\tau]}{\Pr[\mathsf{X}_{\mathrm{id}}=\tau]} &\geq \frac{\prod_{i=1}^{L}\prod_{j=1}^{t_i}(2^{n\kappa_i})_{q_{i,j}}}{2^{2nq} \times (2^n)^{\sum_{i=1}^{L}\sum_{j=1}^{t_i}\kappa_i\cdot q_{i,j}} \times 2^{-2nq}} \geq \frac{\prod_{i=1}^{L}\prod_{j=1}^{t_i}(2^{n\kappa_i})_{q_{i,j}}}{(2^n)^{\sum_{i=1}^{L}\sum_{j=1}^{t_i}\kappa_i\cdot q_{i,j}}} \\ &\geq \prod_{i=1}^{L}\prod_{j=1}^{t_i}\prod_{r=1}^{q_{i,j}-1}\left(1-\frac{r}{2^{n\kappa_i}}\right) \geq 1-\sum_{i=1}^{L}\sum_{j=1}^{t_i}\sum_{r=1}^{q_{i,j}-1}\frac{r}{2^{n\kappa_i}} \\ &\geq 1-\sum_{i=1}^{L}\sum_{j=1}^{t_i}\frac{q_{i,j}^2}{2^{n\kappa_i}} \qquad (10)\end{aligned}$$

Since, $\kappa_{\min}$ is the number of blocks contained in $\ell_{\min}$, where $\ell_{\min} \geq 2n$ and $\sum_{i=1}^{L}\sum_{j=1}^{t_i} q_{i,j} \leq q$, following Eqn. (10), we have

$$\frac{\Pr[\mathsf{X}_{\mathrm{re}}=\tau]}{\Pr[\mathsf{X}_{\mathrm{id}}=\tau]} \geq 1-\frac{q^2}{2^{n\kappa_{\min}}} = 1-\frac{q^2}{2^{2n}}.$$

This concludes the proof of Lemma 2. □

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 SPRPs that achieve 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 and the KDF algorithm as follows: an n-bit tweakable block cipher $\widetilde{\mathsf{E}}$ with master secret key K derives the subkeys $K_h, K_1, \ldots, K_4$, where

$$K_i \leftarrow \widetilde{\mathsf{E}}_K(10^{n-1}, \langle i\rangle_n), 1 \leq i \leq 4, \quad K_h \leftarrow \widetilde{\mathsf{E}}_K(10^{n-1}, \langle 5\rangle_n).$$

On input a key K_h, a message D and a tweak T, we first apply an injective encoding function on $T\|D$ to derive ℓ many blocks $B[1], B[2], \ldots, B[\ell]$. Then, we process each block $B[i]$ by evaluating a TBC $\widetilde{\mathsf{E}}_{K_h}$, 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.,

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

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

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

whereas we generate $Y[\ell]$ as

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

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 $(010\|\mathsf{Lsb}_{n-3}(Y_\ell))$ and V is obtained by evaluating the tweakable block cipher on input $Y[\ell]$ with tweak $(011\|\mathsf{Lsb}_{n-3}(X_\ell))$.

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:

$$\textsc{Encode}(T, D) := \mathsf{Pad}_n(T) \parallel \mathsf{Pad}_n(D) \parallel (\langle |T|\rangle_{n/2} \| \langle |D|\rangle_{n/2}).$$

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

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

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

$$5 + 2(\ell + 2) + 3 + \left\lceil \frac{L_M - 2n}{n} \right\rceil \tag{11}$$

tweakable block cipher calls, where five tweakable block cipher calls are required to derive five keys, three 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. (11), the construction requires a total of at most

$$2\left(\frac{L_M + L_T}{n}\right) + \left\lceil \frac{L_M - 2n}{n} \right\rceil + 14$$

tweakable block cipher calls.

$\mathsf{PHASH}+(K_h, D, T)$

1. $(X[0], Y[0]) \leftarrow (0^n, 0^n)$;
2. $(B[1], B[2], \ldots, B[\ell]) \leftarrow \textsc{Encode}(T, D)$;
3. **for** $i = 1$ to ℓ
4. $\quad Z[i] \leftarrow \widetilde{\mathsf{E}}_{K_h}((000 \| \langle i \rangle_{n-3}), B[i])$;
5. $\quad X[i] \leftarrow X[i-1] \oplus Z[i]$;
6. $\quad Y[i] \leftarrow 2Y[i-1] \oplus Z[i]$;
7. $U \leftarrow \widetilde{\mathsf{E}}_{K_h}((010 \| \mathsf{Lsb}_{n-3}(Y[\ell]), X[\ell])$;
8. $V \leftarrow \widetilde{\mathsf{E}}_{K_h}((011 \| \mathsf{Lsb}_{n-3}(X[\ell]), Y[\ell])$;
9. **return** (U, V);

$\textsc{Encode}(T, D)$

1. $Len \leftarrow \langle |T| \rangle_{n/2} \| \langle |D| \rangle_{n/2}$;
2. $B \leftarrow \mathsf{Pad}_n(T) \| \mathsf{Pad}_n(D) \| Len$;
3. $(B[1], B[2], \ldots, B[\ell]) \xleftarrow{n} \mathsf{Parse}(B)$;
4. **return** $(B[1], B[2], \ldots, B[\ell])$;

$\mathrm{KDF}(K)$

1. **for** $i = 1$ to 4
2. $\quad K_i \leftarrow \widetilde{\mathsf{E}}_K(10^{n-1}, \langle i \rangle_n)$;
3. $K_h \leftarrow \widetilde{\mathsf{E}}_K(10^{n-1}, \langle 5 \rangle_n)$;
4. **return** $(K_h, K_1, K_2, K_3, K_4)$;

$\mathsf{PHCTR}+(K, M, T)$

1. $((M_1 \| M_2), M_R) \xleftarrow{2n} M$;
2. $(K_h, K_1, K_2, K_3, K_4) \leftarrow \mathsf{KDF}(K)$;
3. $H_1 \| H_2 \xleftarrow{n} \mathsf{PHASH}+(K_h, M_R, T)$;
4. $U_1 \leftarrow M_1 \oplus H_1$; $U_2 \leftarrow M_2 \oplus H_2$;
5. $Z \leftarrow \widetilde{\mathsf{E}}_{K_1}(U_2, U_1)$; $V_2 \leftarrow \widetilde{\mathsf{E}}_{K_2}(Z, U_2)$;
6. $V_1 \leftarrow \widetilde{\mathsf{E}}_{K_3}(V_2, Z)$; $W \leftarrow U_2 \oplus V_2$;
7. **for** $i = 1$ to $\lceil |M_R|/n \rceil$
8. $\quad S_i \leftarrow \widetilde{\mathsf{E}}_{K_4}(Z \oplus \langle i \rangle_n, W)$;
9. $S \stackrel{\mathsf{def}}{=} S_1 \| \ldots \| S_{\lceil |M_R|/n \rceil}$;
10. $C_R \leftarrow \mathsf{Msb}_{|M_R|}(S) \oplus M_R$;
11. $H'_1 \| H'_2 \xleftarrow{n} \mathsf{PHASH}+(K_h, C_R, T)$;
12. $C_1 \leftarrow V_1 \oplus H'_1$; $C_2 \leftarrow V_2 \oplus H'_2$;
13. **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.

Remark 2. We would like to mention that the original proposal of the PMAC2x [LN17] 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. In a nutshell, the attack exploited the non-injectivity of the padding function. Later on, the authors have revised their eprint version of the paper [LN16] and introduced 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 [LN17] construction. We would like to mention that we have applied an injective encoding function on the tweak and the message that encodes their length information, making the output of the function non-colliding. Hence, the attack of Minematsu and Iwata is invalid for PHASH+ construction.

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 and prove the following result which bounds the almost-xor-universal advantage of the PHASH+ construction as well as the almost-xor-universal advantage of each of its n-bit output.

Lemma 3. *Let $\mathcal{K}$ be a non-empty finite set and $\widetilde{\mathsf{E}} : \mathcal{K} \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ 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*

$$\Pr[K_h \leftarrow\!\$\, \mathcal{K} : \mathsf{PHASH}+[\widetilde{\mathsf{E}}_{K_h}](M) \oplus \mathsf{PHASH}+[\widetilde{\mathsf{E}}_{K_h}](M') = \Delta] \leq \frac{118}{(2^n-1)^2} + \frac{1}{2^{2n}}\,.$$

Lemma 4. *Let $\mathcal{K}$ be a non-empty finite set and $\widetilde{\mathsf{E}} : \mathcal{K} \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ 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*

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

Proofs of the above two lemmas are deferred to Section 6.1 and Section 6.2 respectively.

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

$$\mathbf{Adv}^{\mathrm{STPRP}}_{\mathsf{PHCTR}+[\widetilde{\mathsf{E}}]}(\mathcal{A}) \leq 6\mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{\mathsf{E}}}(\mathcal{A}') + \frac{183q^2}{(2^n-1)^2} + \frac{2\min\{4q^2\ell_{\max}, \sigma^2\}}{2^{2n}} + \frac{3q^2}{2^{2n}} + \frac{10}{2^n}\,,$$

where $q \leq 2^{n-1}$, σ is the total number of message and tweaks blocks queried, $\ell_{\max}$ is the maximum number of message blocks in any single query, $\mathtt{t}' = O(\mathtt{t} + \sigma + 2qt_H)$, and t_H is the time for computing PHASH+*.*

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

$$\mathbf{Adv}^{\mathrm{STPRP}}_{\mathsf{PHCTR}+}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{\mathsf{E}}}(\mathcal{A}') + \mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{\mathsf{PHCTR}}+}(\mathcal{A}'') + \frac{10}{2^n}. \tag{12}$$

Thus, it is enough to bound the stprp advantage of the $\widetilde{\mathsf{PHCTR}}+$ construction. To do this, we first replace the underlying tweakable block cipher $\widetilde{\mathsf{E}}_{K_h}$ with an n-bit tweakable random permutation at the cost of its tprp advantage and denote the resulting construction as PHCTR+*. Thus, we have

$$\mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{\mathsf{PHCTR}}+}(\mathcal{A}'') \leq \mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{\mathsf{E}}}(\mathcal{A}''') + \mathbf{Adv}^{\mathrm{STPRP}}_{\mathsf{PHCTR}+^*}(\mathcal{B}). \tag{13}$$

Therefore, it is enough to bound the stprp advantage of the PHCTR+* construction, which is followed from combining Theorem 2, Lemma 3, and Lemma 4. Finally, the result follows by combining Eqn. (12), Eqn. (13) and the stprp advantage of PHCTR+* construction. □

5.3 ZHCTR+: An Optimally Secure WBC

We propose another optimally secure single keyed variable-input-length tweakable SPRP, called ZHCTR+. ZHCTR+ is a specific instance of HCTR+ with $2n$-bit hash keys, where we instantiate the underlying $2n$-bit keyed hash function of HCTR+ with ZHASH+ and the KDF algorithm as follows: an n-bit tweakable block cipher $\widetilde{\mathsf{E}}$ with master secret key K derives the subkeys $K_h, K_1, \ldots, K_4$, where

$$K_i \leftarrow \widetilde{\mathsf{E}}_K(10^{n-1}, \langle i \rangle_n), 1 \leq i \leq 4, \quad K_h \leftarrow (\widetilde{\mathsf{E}}_K(10^{n-1}, \langle 5 \rangle_n) \parallel \widetilde{\mathsf{E}}_K(10^{n-1}, \langle 6 \rangle_n)).$$

ZHASH+ is structurally similar to the hash function in ZMAC [IMPS17] with some small differences, as explained in Sect. 5.3. On input a $2n$-bit hash key K_h, a message D and a tweak T, we first apply an injective encoding function on $T \| D$ to derive ℓ many $2n$-bit blocks $B[1], B[2], \ldots, B[\ell]$. We also derive two masking values L_l and L_r by evaluating the tweakable block cipher $\widetilde{\mathsf{E}}$ 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 $\widetilde{\mathsf{E}}$ with input $S_l[i]$ and the tweak is $S_r[i]$. 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 $010\|\mathsf{Lsb}_{n-3}(V)$ and Y_2 is obtained by evaluating the tweakable block cipher on input V with tweak $011\|\mathsf{Lsb}_{n-3}(U)$.

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:

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

We parse the resulting string B into ℓ $2n$-bit blocks. Since we have encoded the length information of T and D, 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+, let L_M denote the length of the input message M in number of bits to the construction and L_T denote the length of the tweak T in number of bits to the construction. In each layer of the hash function, we require $\ell/2 + 2$ tweakable block cipher calls, where

$$\ell \leq \left(\frac{(L_M - 2n + 2n) + (L_T + 2n) + n + n}{2n} \right) = \frac{L_M + L_T + 4n}{2n}.$$

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

$$6 + (\ell + 4) + 3 + \left\lceil \frac{L_M - 2n}{n} \right\rceil \tag{14}$$

tweakable block cipher calls, where six tweakable block cipher calls are required to derive six keys, three 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

$$\left(\frac{L_M + L_T}{2n} + \left\lceil \frac{L_M - 2n}{n} \right\rceil + 14 \right)$$

tweakble block cipher calls.

$\mathsf{ZHASH}+(K_h, D, T)$

1. $(K_{h,1}, K_{h,2}) \leftarrow K_h$;
2. $(U, V) \leftarrow (0^n, 0^n)$;
3. $L_l \leftarrow \widetilde{\mathsf{E}}_{K_{h,1}}(000\|0^{n-3}, 0^n)$;
4. $L_r \leftarrow \widetilde{\mathsf{E}}_{K_{h,1}}(000\|0^{n-4}\|1), 0^n)$;
5. $(B[1], B[2], \ldots, B[\ell]) \leftarrow \textsc{Encode}(T, D)$;
6. **for** $i = 1$ to ℓ
7. $\quad (X_l[i], X_r[i]) \xleftarrow{n} B[i]$;
8. $\quad S_l[i] \leftarrow L_l \oplus X_l[i],\ \ S_r[i] \leftarrow L_r \oplus X_r[i]$;
9. $\quad C_l[i] \leftarrow \widetilde{\mathsf{E}}_{K_{h,2}}(S_r[i]), S_l[i])$;
10. $\quad C_r[i] \leftarrow C_l[i] \oplus X_r[i]$;
11. $\quad U \leftarrow 2U \oplus C_l[i]$; $V \leftarrow V \oplus C_r[i]$;
12. $\quad (L_l, L_r) \leftarrow (2L_l, 2L_r)$;
13. $Y_1 \leftarrow \widetilde{\mathsf{E}}_{K_{h,1}}((010\|\mathsf{Lsb}_{n-3}(V)), U)$;
14. $Y_2 \leftarrow \widetilde{\mathsf{E}}_{K_{h,1}}((011\|\mathsf{Lsb}_{n-3}(U)), V)$;
15. **return** (Y_1, Y_2);

$\textsc{Encode}(T, D)$

1. $Len \leftarrow \langle |T| \rangle_{n/2} \| \langle |D| \rangle_{n/2}$;
2. $B \leftarrow \mathsf{Pad}_{2n}(T) \| \mathsf{Pad}_{2n}(D) \| Len \| 0^n$;
3. $(B[1], B[2], \ldots, B[\ell]) \xleftarrow{2n} \mathsf{Parse}(B)$;
4. **return** $(B[1], B[2], \ldots, B[\ell])$;

$\mathrm{KDF}(K)$

1. **for** $i = 1$ to 4
2. $\quad K_i \leftarrow \widetilde{\mathsf{E}}_K(10^{n-1}, \langle i \rangle_n)$;
3. $K_h \leftarrow (\widetilde{\mathsf{E}}_K(10^{n-1}, \langle 5 \rangle_n) \parallel \widetilde{\mathsf{E}}_K(10^{n-1}, \langle 6 \rangle_n))$;
4. **return** $(K_h, K_1, K_2, K_3, K_4)$;

$\mathsf{ZHCTR}+(K, M, T)$

1. $((M_1\|M_2), M_R) \xleftarrow{2n} M$;
2. $(K_h, K_1, K_2, K_3, K_4) \leftarrow \mathsf{KDF}(K)$;
3. $H_1\|H_2 \xleftarrow{n} \mathsf{ZHASH}+(K_h, M_R, T)$;
4. $U_1 \leftarrow M_1 \oplus H_1$;
5. $U_2 \leftarrow M_2 \oplus H_2$;
6. $Z \leftarrow \widetilde{\mathsf{E}}_{K_1}(U_2, U_1)$;
7. $V_2 \leftarrow \widetilde{\mathsf{E}}_{K_2}(Z, U_2)$;
8. $V_1 \leftarrow \widetilde{\mathsf{E}}_{K_3}(V_2, Z)$;
9. $W \leftarrow U_2 \oplus V_2$;
10. **for** $i = 1$ to $\lceil |M_R|/n \rceil$
11. $\quad S_i \leftarrow \widetilde{\mathsf{E}}_{K_4}(Z \oplus \langle i \rangle_n, W)$;
12. $S \stackrel{\mathsf{def}}{=} S_1\| \ldots \|S_{\lceil |M_R|/n \rceil}$;
13. $C_R \leftarrow \mathsf{Msb}_{|M_R|}(S) \oplus M_R$;
14. $H_1'\|H_2' \xleftarrow{n} \mathsf{ZHASH}+(K_h, C_R, T)$;
15. $C_1 \leftarrow V_1 \oplus H_1'$;
16. $C_2 \leftarrow V_2 \oplus H_2'$;
17. **return** $(C_1\|C_2\|C_R)$;

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

Differences Between ZMAC and ZHASH+. The primary differences between the two constructions are the following:

(i) In ZMAC, after processing ℓ many message blocks $(B[1], B[2], \ldots, B[\ell])$, where each

$B[i] \in \{0,1\}^{n+t}$, the running variables (U, V) can be expressed as follows:

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

where $B[i] = (X_l[i] \| X_r[i])$, $X_l[i] \in \{0,1\}^n, X_r[i] \in \{0,1\}^t$, and $C_r[i] = \mathsf{Msb}_t(C_l[i]) \oplus X_r[i]$.[5] On the other hand, in ZHASH+, the running variable (U, V) is expressed as follows:

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

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

$$Y_1 = \widetilde{\mathsf{E}}_{K_{h,1}}(0\|V,U) \oplus \widetilde{\mathsf{E}}_{K_{h,1}}(1\|V,U), Y_2 = \widetilde{\mathsf{E}}_{K_{h,1}}(2\|V,U) \oplus \widetilde{\mathsf{E}}_{K_{h,1}}(3\|V,U).$$

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

$$Y_1 = \widetilde{\mathsf{E}}_{K_{h,1}}(010\|\mathsf{Lsb}_{n-3}(V),U), Y_2 = \widetilde{\mathsf{E}}_{K_{h,1}}(011\|\mathsf{Lsb}_{n-3}(U),V).$$

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 $\mathsf{Lsb}_{n-3}(V)$ as the tweak in deriving both Y_1 and Y_2 values with appropriate domain separation. However, as we will show later that such a 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 and prove the following result which bounds the almost-xor-universal advantage of the ZHASH+ construction (in Lemma 5) as well as 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 $\widetilde{\mathsf{E}} : \mathcal{K} \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ 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*

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

Lemma 6. *Let $\mathcal{K}$ be a non-empty finite set and $\widetilde{\mathsf{E}} : \mathcal{K} \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ 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*

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

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

Proofs of the above two lemmas are deferred to Section 6.3 and Section 6.4 respectively.

Theorem 4. *Let $\mathcal{K}$ be a non-empty finite set. Let $\widetilde{\mathsf{E}} : \mathcal{K} \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be an (n,n) tweakable block cipher. Then, for any $(q, \ell_{\max}, \mathtt{t})$-chosen plaintext chosen ciphertext adaptive adversary $\mathcal{A}$ against the stprp security of $\mathsf{ZHCTR}+[\widetilde{\mathsf{E}}]$ construction such that each query is of length at least $2n$ bits, there exists a $(\sigma', \mathtt{t}')$-chosen plaintext chosen ciphertext adversary $\mathcal{A}'$ against the stprp security of $\widetilde{\mathsf{E}}$, such that*

$$\mathbf{Adv}^{\mathrm{STPRP}}_{\mathsf{ZHCTR}+[\widetilde{\mathsf{E}}]}(\mathcal{A}) \leq 7\mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{\mathsf{E}}}(\mathcal{A}) + \frac{183q^2}{(2^n-1)^2} + \frac{2\min\{4q^2\ell_{\max}, \sigma^2\}}{2^{2n}} + \frac{3q^2}{2^{2n}} + \frac{15}{2^n},$$

where $q \leq 2^{n-1}$, σ is the total number of message and tweaks blocks queried, $\ell_{\max}$ is the maximum number of message blocks in any single query, $\mathtt{t}' = O(\mathtt{t} + \sigma + 2qt_H)$, and t_H is the time for computing $\mathsf{ZHASH}+$.

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

$$\mathbf{Adv}^{\mathrm{STPRP}}_{\mathsf{ZHCTR}+}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{\mathsf{E}}}(\mathcal{A}') + \mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{\mathsf{ZHCTR}}+}(\mathcal{A}'') + \frac{15}{2^n}. \tag{15}$$

Thus, it is enough to bound the stprp advantage of the $\widetilde{\mathsf{ZHCTR}}+$ construction. To do this, we first replace the underlying tweakable block ciphers $\widetilde{\mathsf{E}}_{K_{h,1}}$ and $\widetilde{\mathsf{E}}_{K_{h,2}}$ with two independently sampled n-bit tweakable random permutations at the cost of its tprp advantage and denote the resulting construction as $\mathsf{ZHCTR}+^*$. Thus, we have

$$\mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{\mathsf{ZHCTR}}+}(\mathcal{A}'') \leq 2\mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{\mathsf{E}}}(\mathcal{A}''') + \mathbf{Adv}^{\mathrm{STPRP}}_{\mathsf{ZHCTR}+^*}(\mathcal{B}). \tag{16}$$

Therefore, it is enough to bound the stprp advantage of the $\mathsf{ZHCTR}+^*$ construction, which is followed from combining Theorem 2, Lemma 5, and Lemma 6. Finally, the result follows by combining Eqn. (15), Eqn. (16) and the stprp advantage of $\mathsf{ZHCTR}+^*$ construction. □

6 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 $\mathsf{PHASH}+$ and $\mathsf{ZHASH}+$ as well as the AXU advantages for each of their n-bit outputs.

6.1 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., $\Delta = \Delta_1 \| \Delta_2$, and $\Delta_1, \Delta_2 \in \{0,1\}^n$. Let E denote the event that the xor of the hash output of $\mathsf{PHASH}+$ on two distinct messages M and M' attains the value Δ. i.e.,

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

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

$$U \leftarrow \widetilde{\mathsf{E}}_{K_h}(010 \| \mathsf{Lsb}_{n-3}(Y[\ell]), X[\ell]),\ V \leftarrow \widetilde{\mathsf{E}}_{K_h}(011 \| \mathsf{Lsb}_{n-3}(X[\ell]), Y[\ell])$$

(see line 7 and line 8 of Fig. 3). Since the domain separation strings 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 $\mathsf{Lsb}_{n-3}(X[\ell])$ as $\widehat{X}[\ell]$ and $\mathsf{Lsb}_{n-3}(Y[\ell])$ as $\widehat{Y}[\ell]$. Now, to bound the probability of the event 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 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:

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

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 $\exists h \in \{1, \ldots, \ell\}$ 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 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 $\exists h \neq s \in \{1, \ldots, \ell\}$ such that $B[h] \neq B'[h]$ and $B[s] \neq B'[s]$. Therefore,

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

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

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

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

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

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

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

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

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

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

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

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:

$$\Pr[\mathsf{E}] = \Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \le \frac{1}{2^{2n}}. \tag{17}$$

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

- Case-A: $(X[\ell] = X'[\ell']$ and $Y[\ell] \neq Y'[\ell'])$. Bounding the probability of the event E holds under this case again gives rise to the following two subcases: (a) if $\mathsf{Lsb}_{n-3}(Y[\ell]) = \mathsf{Lsb}_{n-3}(Y'[\ell'])$, 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 $\mathsf{Lsb}_{n-3}(Y[\ell]) = \mathsf{Lsb}_{n-3}(Y'[\ell'])$ implies the following two equations:

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

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 $\mathsf{Lsb}_{n-3}(Y[\ell]) \neq \mathsf{Lsb}_{n-3}(Y'[\ell'])$, 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:

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

- 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

$$\Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \leq \frac{57}{2^n(2^n - 1)}. \tag{19}$$

- 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 $\mathsf{Lsb}_{n-3}(X[\ell]) = \mathsf{Lsb}_{n-3}(X'[\ell'])$ and $\mathsf{Lsb}_{n-3}(Y[\ell]) = \mathsf{Lsb}_{n-3}(Y'[\ell'])$, 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 $\Delta = \Delta_1 \| \Delta_2$ 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 $\mathsf{Lsb}_{n-3}(X[\ell]) = \mathsf{Lsb}_{n-3}(X'[\ell'])$ and $\mathsf{Lsb}_{n-3}(Y[\ell]) \neq \mathsf{Lsb}_{n-3}(Y'[\ell'])$, then V and V' are distinct and they are not independent random variables. However, since $\mathsf{Lsb}_{n-3}Y[\ell] \neq \mathsf{Lsb}_{n-3}(Y'[\ell'])$, it implies that U and U' are two independent random variables. Thus, for a fixed $2n$ bit string Δ such that $\Delta = \Delta_1 \| \Delta_2$ 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 $\mathsf{Lsb}_{n-3}(X[\ell]) \neq \mathsf{Lsb}_{n-3}(X'[\ell'])$ and $\mathsf{Lsb}_{n-3}(Y[\ell]) = \mathsf{Lsb}_{n-3}(Y'[\ell'])$, then U and U' are distinct and they are not independent random variables. However, since $\mathsf{Lsb}_{n-3}(X[\ell]) \neq \mathsf{Lsb}_{n-3}(X'[\ell'])$, it implies that V and V' are two independent random variables. Thus, for a fixed non-zero $2n$ bit string Δ such that $\Delta = \Delta_1 \| \Delta_2$ 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 $\mathsf{Lsb}_{n-3}(X[\ell]) \neq \mathsf{Lsb}_{n-3}(X'[\ell'])$ and $\mathsf{Lsb}_{n-3}(Y[\ell]) \neq \mathsf{Lsb}_{n-3}(Y'[\ell'])$, then V and V' are two independent random variables and U and U' are two independent random variables. Thus, for a fixed $\Delta_1 \| \Delta_2$, 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 E in Case C as follows:

$$\Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \leq \frac{4}{(2^n-1)^2}. \tag{20}$$

Now, by combining Eqn. (18), Eqn. (19), and Eqn. (20), we derive the upper bound on the probability of the event E in Case-II as follows:

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

Finally, by combining Eqn. (17) and Eqn. (21), we derive the upper bound on the probability of the event E as

$$\Pr[\mathsf{E}] = \Pr[U \oplus U' = \Delta_1, V \oplus V' = \Delta_2] \leq \frac{118}{(2^n-1)^2} + \frac{1}{2^{2n}}. \tag{22}$$

6.2 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 E_1 and E_2 denote the events $U \oplus U' = \delta$ and $V \oplus V' = \delta$ respectively. As before, the analyses of upper bounding the probability of the events E_1 and 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,

$$\Pr[U \oplus U' = \delta] = \begin{cases} \frac{1}{2^{2n}} \text{ if } \delta = 0^n \\ 0 \text{ otherwise} \end{cases} \quad \text{and } \Pr[V \oplus V' = \delta] = \begin{cases} \frac{1}{2^{2n}} \text{ if } \delta = 0^n \\ 0 \text{ otherwise} \end{cases} \tag{23}$$

Case-II: $\neg(X[\ell] = X'[\ell']$ 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 $\mathsf{Lsb}_{n-3}(Y[\ell]) = \mathsf{Lsb}_{n-3}(Y'[\ell'])$, 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 $\mathsf{Lsb}_{n-3}(Y[\ell]) = \mathsf{Lsb}_{n-3}(Y'[\ell'])$ holds. Note that, $X[\ell] = X'[\ell']$ and $\mathsf{Lsb}_{n-3}(Y[\ell]) = \mathsf{Lsb}_{n-3}(Y'[\ell'])$ implies the following system of equations hold:

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

  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

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

  (b) If $\mathsf{Lsb}_{n-3}(Y[\ell]) \neq \mathsf{Lsb}_{n-3}(Y'[\ell'])$, 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

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

  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:

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

- 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,

$$\Pr[U \oplus U' = \delta] \leq \frac{2}{2^n - 1} \text{ and } \Pr[V \oplus V' = \delta] \leq \frac{57}{2^n}. \tag{25}$$

- Case-C: ($X[\ell] \neq X'[\ell']$ and $Y[\ell] \neq Y'[\ell']$). We analyze this case into the following four subcases: (a) if $\mathsf{Lsb}_{n-3}(X[\ell]) = \mathsf{Lsb}_{n-3}(X'[\ell'])$ and $\mathsf{Lsb}_{n-3}(Y[\ell]) = \mathsf{Lsb}_{n-3}(Y'[\ell'])$, then it follows that U and U' 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 $\mathsf{Lsb}_{n-3}(X[\ell]) = \mathsf{Lsb}_{n-3}(X'[\ell'])$ and $\mathsf{Lsb}_{n-3}(Y[\ell]) \neq \mathsf{Lsb}_{n-3}(Y'[\ell'])$, then it follows that V and V' are distinct and they are not independent random variables. However, since $\mathsf{Lsb}_{n-3}(Y[\ell]) \neq \mathsf{Lsb}_{n-3}(Y'[\ell'])$, 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 $\mathsf{Lsb}_{n-3}(X[\ell]) \neq \mathsf{Lsb}_{n-3}(X'[\ell'])$ and $\mathsf{Lsb}_{n-3}(Y[\ell]) = \mathsf{Lsb}_{n-3}(Y'[\ell'])$, then it follows that U and U' are distinct and they are not independent random variables. However, since $\mathsf{Lsb}_{n-3}(X[\ell]) \neq \mathsf{Lsb}_{n-3}(X'[\ell'])$, it implies that V and V' are two independent random variables. Thus, for a fixed n-bit string δ, we have

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

(d) If both $\mathsf{Lsb}_{n-3}(X[\ell]) \neq \mathsf{Lsb}_{n-3}(X'[\ell'])$ and $\mathsf{Lsb}_{n-3}(Y[\ell]) \neq \mathsf{Lsb}_{n-3}(Y'[\ell'])$, 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:

$$\Pr[U \oplus U' = \delta] \leq \frac{4}{2^n - 1} \text{ and } \Pr[V \oplus V' = \delta] \leq \frac{4}{2^n - 1}. \tag{26}$$

By combining Eqn. (24), Eqn. (25), and Eqn. (26), 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:

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

Finally, by combining Eqn. (23) and Eqn. (27), we derive the upper bound on the probability of the event E_b for $b \in \{1, 2\}$ as follows:

$$\Pr[U \oplus U' = \delta] \leq \frac{63}{2^n - 1} + \frac{1}{2^{2n}} \text{ and } \Pr[V \oplus V' = \delta] \leq \frac{63}{2^n - 1} + \frac{1}{2^{2n}}. \tag{28}$$

6.3 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., $\Delta = \Delta_1 \| \Delta_2$, 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.,

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

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

$$Y_1 \leftarrow \widetilde{\mathsf{E}}_{K_{h,1}}(010 \| \mathsf{Lsb}_{n-3}(V), U),\ Y_2 \leftarrow \widetilde{\mathsf{E}}_{K_{h,1}}(011 \| \mathsf{Lsb}_{n-3}(U), V)$$

(see line 12 and line 13 of Fig. 4). Since the domain separation strings 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 $\mathsf{Lsb}_{n-3}(V)$ as $\widehat{V}$ and $\mathsf{Lsb}_{n-3}(U)$ as $\widehat{U}$. Now, to bound the probability of the event E holds, we have the following two cases:

Case-I: $(U = U' \text{ and } V = V')$. This event implies $\widehat{U} = \widehat{U}'$ and $\widehat{V} = \widehat{V}'$ 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 the analysis for the sake of completeness. Note that, the condition $U = U'$ and $V = V'$ implies the following two equations:

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

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 $\exists h \in \{1, \ldots, \ell\}$ such that $B[h] \neq B'[h]$ and $B[s] = B'[s]$ for all $s \in \{1, \ldots, \ell\} \setminus \{h\}$. Therefore,

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

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

$$Y_1 \leftarrow \widetilde{\mathsf{E}}_{K_{h,1}}(010 \| \mathsf{Lsb}_{n-3}(V), U),\ Y_2 \leftarrow \widetilde{\mathsf{E}}_{K_{h,1}}(011 \| \mathsf{Lsb}_{n-3}(V), U),$$

then we cannot guarantee the axu advantage of ZHASH+ up to 2^{-2n}. Note that, the bad event would be $U = U'$ and $\widehat{V} = \widehat{V}'$. 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 $\mathsf{Lsb}_{n-3}(X_r[h]) = \mathsf{Lsb}_{n-3}(X_r'[h])$ and for all $s \neq h$, $B[s] = B'[s]$. In this case, we have

$$\Delta\widehat{U} = 2^{\ell-h}(C_l[h] \oplus C_l'[h]),\ \Delta\widehat{V} = \mathsf{Lsb}_{n-3}(C_l[h] \oplus X_r[h]) \oplus \mathsf{Lsb}_{n-3}(C_l'[h] \oplus X_r'[h]).$$

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 $\Delta\widehat{V} = 0^n$. As a result, the event $U = U'$ implies $\widehat{V} = \widehat{V}'$ and hence the axu advantage of ZHASH+ is dropped to 2^{-n}.

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

$$\Delta U := U \oplus U' \quad \Rightarrow \quad 2^{\ell-h}(C_l[h] \oplus C'_l[h]) \oplus 2^{\ell-s}(C_l[s] \oplus C'_l[s]) \oplus \underbrace{\bigoplus_{i \neq h,s} 2^{\ell-i}(C_l[i] \oplus C'_l[i])}_{\Sigma},$$

$$\Delta V := V \oplus V' \quad \Rightarrow \quad (C_r[h] \oplus C'_r[h]) \oplus (C_r[s] \oplus C'_r[s]) \oplus \underbrace{\bigoplus_{i \neq h,s} (C_r[i] \oplus C'_r[i])}_{\Theta}.$$

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:

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

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

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

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:

$$\Pr[\mathsf{E}] = \Pr[Y_1 \oplus Y'_1 = \Delta_1, Y_2 \oplus Y'_2 = \Delta_2] \le \frac{1}{2^{2n}}. \tag{29}$$

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 $\mathsf{Lsb}_{n-3}(V) = \mathsf{Lsb}_{n-3}(V')$, 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 $\mathsf{Lsb}_{n-3}(V) = \mathsf{Lsb}_{n-3}(V')$ implies the following two equations:

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

  Since the rank of the above system of equations is at least 1, the probability that the above system of equations holds 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,

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

  (b) If $\mathsf{Lsb}_{n-3}(V) \neq \mathsf{Lsb}_{n-3}(V')$, 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.,

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

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

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

- 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

$$\Pr[Y_1 \oplus Y'_1 = \Delta_1, Y_2 \oplus Y'_2 = \Delta_2] \le \frac{57}{2^n(2^n-1)}. \tag{31}$$

- 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 $\mathsf{Lsb}_{n-3}(U) = \mathsf{Lsb}_{n-3}(U')$ and $\mathsf{Lsb}_{n-3}(V) = \mathsf{Lsb}_{n-3}(V')$, 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 $\Delta = \Delta_1 \| \Delta_2$ 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 $\mathsf{Lsb}_{n-3}(U) = \mathsf{Lsb}_{n-3}(U')$ and $\mathsf{Lsb}_{n-3}(V) \neq \mathsf{Lsb}_{n-3}(V')$, then Y_2 and Y_2' are distinct and they are not independent random variables. However, since $\mathsf{Lsb}_{n-3}(V) \neq \mathsf{Lsb}_{n-3}(V')$, it implies that Y_1 and Y_1' are two independent random variables. Thus, for a fixed $2n$ bit string Δ such that $\Delta = \Delta_1 \| \Delta_2$ 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 $\mathsf{Lsb}_{n-3}(U) \neq \mathsf{Lsb}_{n-3}(U')$ and $\mathsf{Lsb}_{n-3}(V) = \mathsf{Lsb}_{n-3}(V')$, then Y_1 and Y_1' are distinct and they are not independent random variables. However, since $\mathsf{Lsb}_{n-3}(U) \neq \mathsf{Lsb}_{n-3}(U')$, 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 $\Delta = \Delta_1 \| \Delta_2$ 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 $\mathsf{Lsb}_{n-3}(U) \neq \mathsf{Lsb}_{n-3}(U')$ and $\mathsf{Lsb}_{n-3}(V) \neq \mathsf{Lsb}_{n-3}(V')$, 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 $\Delta_1 \| \Delta_2$, 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:
$$\Pr[Y_1 \oplus Y_1' = \Delta_1, Y_2 \oplus Y_2' = \Delta_2] \leq \frac{4}{(2^n - 1)^2}. \tag{32}$$

Now, by combining Eqn. (30), Eqn. (31), and Eqn. (32), we derive the upper bound on the probability of the event E in Case-II as follows:
$$\Pr[Y_1 \oplus Y_1' = \Delta_1, Y_2 \oplus Y_2' = \Delta_2] \leq \frac{118}{(2^n - 1)^2}. \tag{33}$$

Finally, by combining Eqn. (29) and Eqn. (33), we derive the upper bound on the probability of the event E as
$$\Pr[\mathsf{E}] = \Pr[Y_1 \oplus Y_1' = \Delta_1, Y_2 \oplus Y_2' = \Delta_2] \leq \frac{118}{(2^n - 1)^2} + \frac{1}{2^{2n}}. \tag{34}$$

6.4 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: $(U = U' \text{ and } V = V')$. 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 $Y_1 \oplus Y'_1$ boils 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

$$\Pr[Y_b \oplus Y'_b = \delta] = \begin{cases} \frac{1}{2^{2n}} \text{ if } \delta = 0^n \\ 0 \text{ otherwise} \end{cases} \tag{35}$$

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

- Case-A: $(U = U' \text{ and } V \neq V')$. We split this subcase further into two subcases as follows: (a) if $\mathsf{Lsb}_{n-3}(V) = \mathsf{Lsb}_{n-3}(V')$, 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 $\mathsf{Lsb}_{n-3}(V) = \mathsf{Lsb}_{n-3}(V')$ holds. Note that, $U = U'$ and $\mathsf{Lsb}_{n-3}(V) = \mathsf{Lsb}_{n-3}(V')$ implies the following system of equations hold:

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

  Note that the rank of the above system of equations is at least 1. Hence, the probability of the above system of equations holds 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 ensure $V \neq V'$. 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 $\mathsf{Lsb}_{n-3}(V) \neq \mathsf{Lsb}_{n-3}(V')$, 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:

$$\Pr[Y_1 \oplus Y_1' = \delta] \leq \frac{57}{2^n}, \qquad \Pr[Y_2 \oplus Y_2' = \delta] \leq \frac{2}{2^n - 1}. \tag{36}$$

- 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,

$$\Pr[Y_1 \oplus Y_1' = \delta] \leq \frac{2}{2^n - 1}, \Pr[Y_2 \oplus Y_2' = \delta] \leq \frac{57}{2^n}. \tag{37}$$

- Case-C: ($U \neq U'$ and $V \neq V'$). We analyze this case into the following four subcases: (a) if $\mathsf{Lsb}_{n-3}(U) = \mathsf{Lsb}_{n-3}(U')$ and $\mathsf{Lsb}_{n-3}(V) = \mathsf{Lsb}_{n-3}(V')$, then it follows that Y_1 and Y_1' 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 $\mathsf{Lsb}_{n-3}(U) = \mathsf{Lsb}_{n-3}(U')$ and $\mathsf{Lsb}_{n-3}(V) \neq \mathsf{Lsb}_{n-3}(V')$, then it follows that Y_2 and Y_2' are distinct and they are not independent random variables. However, since $\mathsf{Lsb}_{n-3}(V) \neq \mathsf{Lsb}_{n-3}(V')$, 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 $\mathsf{Lsb}_{n-3}(U) \neq \mathsf{Lsb}_{n-3}(U')$ and $\mathsf{Lsb}_{n-3}(V) = \mathsf{Lsb}_{n-3}(V')$, then it follows that Y_1 and Y_1' are distinct and they are not independent random variables. However, since $\mathsf{Lsb}_{n-3}(U) \neq \mathsf{Lsb}_{n-3}(U')$, 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 $\mathsf{Lsb}_{n-3}(U) \neq \mathsf{Lsb}_{n-3}(U')$ and $\mathsf{Lsb}_{n-3}(V) \neq \mathsf{Lsb}_{n-3}(V')$, 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:

$$\Pr[Y_1 \oplus Y_1' = \delta] \leq \frac{4}{2^n - 1}, \qquad \Pr[Y_2 \oplus Y_2' = \delta] \leq \frac{4}{2^n - 1}. \tag{38}$$

By combining Eqn. (36), Eqn. (37), and Eqn. (38), 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:

$$\Pr[Y_1 \oplus Y_1' = \delta] \leq \frac{63}{2^n - 1}, \qquad \Pr[Y_2 \oplus Y_2' = \delta] \leq \frac{63}{2^n - 1}. \tag{39}$$

Finally, by combining Eqn. (35) and Eqn. (39), we derive the upper bound on the probability of the event E_b for $b \in \{1, 2\}$ as follows:

$$\Pr[Y_b \oplus Y_b' = \delta] \leq \frac{63}{2^n - 1} + \frac{1}{2^{2n}}. \tag{40}$$

7 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-256 [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-256 [JNPS16] with 128-bit key, tweak, and state size. Deoxys-BC-256 is a tweakable block cipher that adheres to the TWEAKEY framework [JNP14b], where the key and tweak are combined into a single entity called a tweakey. The round function of Deoxys-BC-256 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-256 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

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

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

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

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-256 is the use of AES-NI instruction set. Using AES-NI instructions, Deoxys-BC-256 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-256 as the underlying TBC. The performance is benchmarked in an 11th Gen 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 and the cache is warmed up before measurements start.

We evaluate the performance of PHCTR+ and ZHCTR+, comparing them with the existing n-bit-secure SPRP, ZCZ, which utilizes Deoxys-BC-384 [JNPS16] as the underlying TBC. Furthermore, we compare the performance of PHCTR+ and ZHCTR+, with the two concrete GEM instantiations: KohiNoor and DaryaiNoor.
The performance results in cycles per byte (c/b) are presented in Table 2. For our evaluation, the total number of CPU cycles required to encrypt a message with a given tweak was 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.[6]

Note that ZCZ is a variable-input-length SPRP, and is not a tweakable SPRP. Thus, we employ an empty tweak for PHCTR+ and ZHCTR+ to compare with ZCZ. To highlight

[6]The source codes of the implementation are available at https://github.com/ShibamCrS/HCTR_PLUS.git.

Table 2: Software performance (c/b) of PHCTR+ and ZHCTR+, and comparison with ZCZ, KohiNoor (KN) and DaryaiNoor (DN). The colors highlight the columns that correspond to the tweak lengths of ZCZ, KohiNoor, or DaryaiNoor.

	ZHCTR+			PHCTR+			ZCZ	KN	DN
Msg./Tweak	0 B	16 B	256 B	0 B	16 B	256 B	0 B	16 B	16 B
1 KB	1.68	1.73	1.45	1.58	1.60	1.35	6.77	1.27	1.47
2 KB	1.36	1.39	1.27	1.38	1.39	1.28	5.97	1.07	1.34
4 KB	1.19	1.21	1.16	1.28	1.29	1.23	5.42	0.98	1.28
8 KB	1.12	1.13	1.10	1.23	1.23	1.20	5.16	0.94	1.25
16 KB	1.08	1.09	1.07	1.20	1.21	1.19	4.84	0.91	1.23
32 KB	1.06	1.06	1.06	1.19	1.19	1.18	4.57	0.91	1.22
64 KB	1.05	1.05	1.05	1.20	1.20	1.19	4.44	0.90	1.22

the comparison with ZCZ, we marked the columns corresponding to zero tweaks with blue color in Table 2. From the results, PHCTR+ and ZHCTR+ are about four times better in terms of software performance than ZCZ despite the fact that our proposals require more 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-256 as the underlying primitive, whereas ZCZ used Deoxys-BC-384 [JNPS16], with a 128-bit key and 256-bit tweak. Notably, Deoxys-BC-256 has 14 rounds, compared to 16 rounds in Deoxys-BC-384, and involves fewer tweakey schedule operations per round, reducing one shuffle and one LFSR call per round, which enhances performance.

On the other hand, ZHCTR+ demonstrates approximately 20% higher performance compared to DaryaiNoor, which employs a variant of the CTR-based weak pseudorandom function SoCTR, utilizing AES-128 for encryption and GHASH-256 as the hash function. Additionally, PHCTR+ exhibits performance nearly equivalent to that of DaryaiNoor, showing no significant difference in efficiency. When comparing ZHCTR+ and PHCTR+ to KohiNoor, which is based on the GCTR-3 construction with ButterKnife-256, we observe a moderate difference in performance. ZHCTR+ runs with only a slight overhead of approximately 10%, while PHCTR+ exhibits around a 25% increase in computation time. However, this trade-off comes with a significant advantage—ZHCTR+ and PHCTR+ utilize Deoxys, a standardized and well-analyzed primitive, whereas the primitive of KohiNoor has not undergone the same level of scrutiny.

We would like to mention that performance wise ZHCTR+ is slightly better than PHCTR+. This difference appears due to the fact that tweak processing of ZHCTR+ is efficient compared to that of PHCTR+ as the tweak is encoded as a counter in PHASH+, which gets incremented at each TBC call. Nevertheless, PHCTR+ saves $2n$ bit state compared to ZHCTR+, which might be useful for hardware constrained devices. Moreover, as PHCTR+ uses only tweak updates and no linear hashing or doubling in masks, it has the potential to offer also some security in quantum settings.

8 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-256. Software implementations of the resulting two constructions, PHCTR+ and ZHCTR+ indicate a significant performance improvement compared to the ZCZ construction. We would like to note that there exist block cipher-based tweakable block ciphers (TBCs) that achieve security beyond the birthday bound, such as CLRW2 and CLRW1^4. Consequently, it is possible to instantiate the TBCs in our constructions using these alternatives. This would naturally transform our constructions into block cipher-based ones. However, a limitation of this approach is, on the one hand, reduced efficiency and, on the other hand, that the resulting constructions are only secure up to $2^{3n/4}$ queries, as both CLRW2 and CLRW1^4 offer security up to $2^{3n/4}$ queries in the standard model. Achieving (near) optimal security using block cipher-based TBCs either requires a large number of block cipher calls or relies on the ideal cipher assumption for the underlying block cipher. It remains an interesting open problem to analyze the multi-user security of HCTR+ and transform it into a context-committing AE scheme.

Acknowledgements. We are highly thankful to the anonymous reviewers for their fruitful comments and suggestions. This research has been partially supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) project no. 509754807, the Ministry of Education in Singapore under Grant RG102/24 and the ERC grant ReSCALE (number 101041545). A part of this project has been discussed during the Asian Symmetric Key workshop 2024 held at Kolkata. Once again we would like to thank Takuto Takami, Takahiro Kaneko and Tetsu Iwata for identifying a security flaw in the older version of the paper.

References

[ABD+23] Roberto Avanzi, Subhadeep Banik, Orr Dunkelman, Maria Eichlseder, 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.

[ACL+24] Elena Andreeva, Benoît Cogliati, Virginie Lallemand, Marine Minier, Antoon Purnal, and Arnab Roy. Masked Iterate-Fork-Iterate: A New Design Paradigm for Tweakable Expanding Pseudorandom Function. In Christina Pöpper and Lejla Batina, editors, *ACNS II*, volume 14584 of *Lecture Notes in Computer Science*, pages 433–459. Springer, 2024.

[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.

[BBC+21] Ritam Bhaumik, Xavier Bonnetain, André Chailloux, Gaëtan Leurent, María Naya-Plasencia, André Schrottenloher, and Yannick Seurin. QCB: efficient quantum-secure authenticated encryption. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 668–698. Springer, 2021.

[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.

[BNT19] Mihir Bellare, Ruth Ng, and Björn Tackmann. Nonces are noticed: AEAD revisited. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 235–265. Springer, 2019.

[BR00] Mihir Bellare and Phillip Rogaway. Encode-Then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography. In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*, pages 317–330. Springer, 2000.

[BVA24] Amit Singh Bhati, Michiel Verbauwhede, and Elena Andreeva. Breaking, Repairing and Enhancing XCBv2 into the Tweakable Enciphering Mode GEM. *IACR Cryptol. ePrint Arch.*, page 1554, 2024.

[CB18] Paul Crowley and Eric Biggers. Adiantum: length-preserving encryption for entry-level processors. *IACR Trans. Symmetric Cryptol.*, 2018(4):39–61, 2018.

[CDD+24] Yu Long Chen, Michael Davidson, Morris Dworkin, Jinkeon Kang, John Kelsey, Yu Sasaki, Meltem Sönmez Turan, Donghoon Chang, Nicky Mouha, and Alyssa Thompson. Proposal of Requirements for an Accordion Mode: Discussion Draft for the NIST Accordion Mode Workshop 2024. Technical report, National Institute of Standards and Technology, April 10 2024.

[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.

[CFZ+25] Yincen Chen, Qinggan Fu, Ning Zhao, Jiahao Zhao, Ling Song, and Qianqian Yang. A Holistic Framework for Impossible Boomerang Attacks. *IACR Cryptol. ePrint Arch.*, page 159, 2025.

[CGLS22] Debrup Chakraborty, Sebati Ghosh, Cuauhtemoc Mancillas López, and Palash Sarkar. FAST: 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.

[DDG+25] Nilanjan Datta, Avijit Dutta, Shibam Ghosh, Eik List, and Hrithik Nandi. Hctr+: An optimally secure tbc-based accordion mode. *IACR Transactions on Symmetric Cryptology*, 2025(3):183–229, Sep. 2025.

[DFUB24] Hieu Nguyen Duy, 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.

[Dwo10] Morris Dworkin. NIST Special Publication 800-38E: Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices. Technical report, US National Institute for Standards in Technology, January 2010.

[Dwo12] Morris Dworkin. NIST Special Publication 800-38F: Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping. Technical report, US National Institute for Standards in Technology, December 2012.

[GDM22] Aldo Gunsing, 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.

[HKR15] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust Authenticated-Encryption AEZ and the Problem That It Solves. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT I*, volume 9056 of *Lecture Notes in Computer Science*, pages 15–44. Springer, 2015.

[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.

[IEE24] IEEE. Ieee p1619/d12 draft standard p1619/for cryptographic protection of data on block-oriented storage devices. Technical report, IEEE, October 2024.

[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.

[ISO22] ISO/IEC. 18033-7:2022(en) Information security – Encryption algorithms – Part 7: Tweakable block ciphers. Technical report, ISO/IEC JTC 1/SC 27, 2022.

[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.

[JNP14a] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and keys for block ciphers: The TWEAKEY framework. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2014.

[JNP14b] 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.yp.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.

[KLLN16] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking Symmetric Cryptosystems Using Quantum Period Finding. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO II*, volume 9815 of *Lecture Notes in Computer Science*, pages 207–237. Springer, 2016.

[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.

[LN17] 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 sivx. *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/âĹnaor/.

[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.

[Rog11] Phillip Rogaway. Evaluation of Some Blockcipher Modes of Operation. Technical report, Cryptography Research and Evaluation Committees (CRYPTREC), February 10 2011.

[RS06] Phillip Rogaway and Thomas Shrimpton. A Provable-Security Treatment of the Key-Wrap Problem. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 373–390. Springer, 2006.

[RZ18] Phillip Rogaway and Yusi Zhang. Onion-AE: Foundations of Nested Encryption. *Proc. Priv. Enhancing Technol.*, 2018(2):85–104, 2018.

[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.

[SZY+22] Ling Song, Nana Zhang, Qianqian Yang, Danping Shi, Jiahao Zhao, Lei Hu, and Jian Weng. Optimizing Rectangle Attacks: A Unified and Generic Framework for Key Recovery. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT I*, volume 13791 of *Lecture Notes in Computer Science*, pages 410–440. Springer, 2022.

[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.

A Related Works on Tweakable Enciphering Schemes

Research of TES modes was informally started with Naor and Reingold's [NR97, NR99a, NR99b] wide-block cipher. Their design approach was based on a paradigm called *hash-encipher-hash* that involves applying an invertible blockwise-universal hash function on the input followed by enciphering the result in ECB mode and then applying yet another invertible blockwise-universal hash function. While they did not fully specify the mode of operation they came close to showing how to construct an invertible blockwise-universal hash function from an xor-universal hash function.
Following [NR97, NR99a, NR99b] the field of designing tweakable enciphering schemes 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.

A.1 Encrypt-Mix-Encrypt

In [HR03], Halevi and Rogaway introduced the design paradigm of tweakable enciphering scheme wherein a simple mixing layer is sandwiched between two invertible encryption layers. As an instantiation of this generic design framework, Halevi and Rogaway [HR03] proposed CMC (CBC-Mix-CBC) wherein CBC mode of encryption is used in both encryption layers with a simple linear mixing function in between. Due to the CBC structure, CMC is inherently a sequential construction. As a follow-up work of CMC, Halevi and Rogaway [HR04] 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 of at least one block. In [Jiv14], Jivsov presented a variant of the EME construction, called WCFB, and showed that it works extremely well for commonly occurring plaintext and repeated operations on the same wide-block. In [BN15], Bhaumik et al. proposed FMix, a single-keyed inverse-free tweakable enciphering scheme.

A.2 Hash-Encrypt-Hash

As the name suggests, this design paradigm of tweakable enciphering scheme invokes an encryption layer in between 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-length block cipher. Their 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] gives only high-level insights 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[7]. The advantage of using the counter mode encryption is to tackle 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. proposed HCTR [WFW05] with a single block-cipher 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] proposed an efficient single-keyed variant of the HCTR construction, dubbed HCTR2, that uses an almost-xor-universal hash function and allows more pre-computation for better performance. In [CGLS22], Chakraborty et al. proposed a PRF-based inverse-free tweakable enciphering scheme, called FAST, and showed that it achieves a security bound in the order of $\sigma^2/2^n$.
Among the above-mentioned constructions, only CMC and EME* are block-cipher based constructions with a lightweight masking layer in between two encryption layers, whereas the other two paradigms require field multiplications (as a part of the hash function evaluation) along with the block-cipher evaluations. 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. 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.

A.3 BBB Secure Tweakable Enciphering Schemes

In [MI11], Minematsu and Iwata 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 proposed the TCT2 construction, that achieves $2n/3$-bit security. Both constructions [MI11, ST13] require two primitives, a block cipher and a universal hash function. In [DN18], Dutta and Nandi proposed a tweakable variant of the HCTR construction, called THCTR, that uses a tweakable block cipher as a primitive

[7] An 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.

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 repetitions 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 degrades gracefully as the maximum number of repeated tweaks increases. In [BLN18], Bhaumik et al. came up with 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.