HCTR^{++} : A Beyond Birthday Bound Secure HCTR2 Variant

Gülnihal Öztürk[1], Onur Koçak[2], and Oğuz Yayla[1]

[1]Cryptography, IAM METU, Ankara, Türkiye ,
gulnihal.ozturk_01@metu.edu.tr, oguz@metu.edu.tr
[2]TÜBİTAK, Ankara, Türkiye , onur.kocak@tubitak.gov.tr

February 2026

Abstract

Current industry-standard block cipher modes of operation, such as CBC and GCM, are fundamentally limited by the birthday bound $O(2^{n/2})$, a constraint that has evolved from a theoretical concern into a practical security bottleneck in contemporary high-throughput, high-data-volume environments. To address this, the cryptographic community and NIST are prioritizing Beyond Birthday Bound (BBB) security to extend the operational security margin toward the full block size $O(2^n)$. Achieving BBB security requires a departure from traditional constructions, primarily utilizing three methodologies: XOR of Permutations (XORP), Tweakable Block Ciphers (TBCs), and Fresh Re-keying. While none of these innovative BBB modes have been formally standardized, NIST has initiated the Accordion Mode project, defining a new primitive class: the Tweakable Variable-Input-Length Strong Pseudorandom Permutation (VIL-SPRP). This primitive treats the entire message as a single, indivisible block and expects the submission of BBB-secure variants. To contribute to this standardization effort, we propose a simple BBB-secure variant of the HCTR2 algorithm based on Fresh Re-keying and a $2n$-bit ε-uniform-AXU keyed hash family. Under the stated hash assumptions and the ideal-cipher model, the construction achieves a beyond-birthday-bound security profile for bounded message lengths and sufficiently restricted tweak repetition. We first explain the core BBB methodologies, then discuss the operational mechanism of HCTR2, and finally present our proposed BBB-secure construction.

Keywords: Accordion mode, Beyond Birthday Bound Security, HCTR2.

1 Introduction

Block ciphers form the backbone of modern cryptographic systems, but they rely on specific modes of operation to secure data streams. While industry standards such as CBC (Cipher Block Chaining) [16] and GCM (Galois/Counter Mode) [17] have been heavily utilized over decades, they are constrained by a theoretical limit known as the birthday bound. In an era where data transmission speeds reach terabit-per-second levels and the volume of data encrypted under a single key grows exponentially, this limit has evolved from a theoretical concern into a practical security bottleneck. Consequently, NIST (National Institute of Standards and Technology) and the cryptographic community are pivoting towards next-generation modes that offer Beyond Birthday Bound (BBB) security, with innovative designs like Accordion [8] emerging as critical solutions.

Most traditional block cipher modes preserve confidentiality only up to $O(2^{n/2})$ processing complexity for an n-bit block size. For a standard 128-bit block cipher like AES, this limit is 2^{64} blocks. While this margin is generally acceptable for 128-bit ciphers under moderate loads, it becomes a critical vulnerability for 64-bit lightweight ciphers or high-throughput environments where data volumes approach the collision threshold. Based on the Birthday Paradox, this threshold represents the point where the probability of a collision between ciphertext blocks becomes non-negligible (occurring well before the full exhaustive search space is exhausted). In high-performance environments such as terabit-speed optical links, massive storage area networks (SANs), or long-lived encrypted tunnels, the accumulation of data can approach thresholds where distinguishing attacks or plaintext recovery become theoretically feasible. When this limit is approached or exceeded, an adversary can distinguish the output of the encryption scheme from a truly random permutation or in certain scenarios (such as nonce reuse in GCM [21] or Sweet32 class attacks [6]), partial plaintext information may be compromised.

To overcome these constraints without abandoning the efficiency of 128-bit primitives, BBB modes are designed to push the security margin from the standard $O(2^{n/2})$ birthday barrier to higher bounds, ideally, to the full block size of $O(2^n)$. Therefore, the cryptographic research community has aggressively pursued BBB security. A BBB mode of operation is designed to maintain its security properties, confidentiality, authenticity, or both, for data volumes significantly exceeding the $2^{n/2}$ limit, with the security margin extending beyond the conventional birthday barrier and, for suitable constructions, approaching the full block-size bound. Achieving BBB security

typically requires a departure from traditional single-key, single-permutation constructions. Theoretical research has focused on three primary methodologies for overcoming the birthday barrier: the XOR of Permutations (XORP), Tweakable Block Ciphers (TBC), and Fresh Re-keying. These methodologies have been effective in the design of numerous BBB secure modes of operation.

The first methodology, XORP , is grounded in the Sum of Permutations distinguishability proofs. The idea firstly suggested by Bellare et al [4] and Lucks [25]. As established by them and later refined by Patarin [28], the sum of two or more independent pseudorandom permutations behaves like a pseudorandom function with security bounds significantly higher than the birthday limit. This method mitigates collision risks by ensuring that the output is not the result of a single permutation evaluation, effectively masking the internal collisions that would otherwise leak information. This methodology has found practical application in several BBB-secure designs, including CENC [19] and its variant CHM which offers efficient parallelizability by XORing the block cipher output with a sum-of-permutations generated keystream; Docked Double Decker [14] by Dobraunig et al., which also suggests a BBB-secure authenticated encryption mode for block ciphers like AES using a CENC-like construction; and more recently, the eGCM and eGCM-SIV proposals by Chung et al. [10], which leverage CENC-like constructions (eCTR and HteC) to upgrade the security of the original GCM family.

The second primary methodological approach for achieving BBB security employs TBCs, a primitive formally introduced by Liskov, Rivest, and Wagner [24], where an additional input, the tweak (T), selects a unique permutation from a family; by assigning a unique tweak value to every block encryption, TBCs rigorously enforce domain separation, ensuring non-repeating inputs to the underlying block cipher and thereby theoretically extending the security boundary up to the full codebook size of $O(2^n)$. A critical nuance is in TBCs guaranteeing that the tweak provides a fresh, distinct random permutation for each encryption operation. This TBC approach has driven significant innovation, including the OCB (Offset Codebook) family (versions 1 through 3) by Rogaway et al. [29, 22], which uses an efficient XEX-based construction to simulate a TBC, and its enhanced version XOCB [2] which achieves BBB security by hybridizing the TBC principle with XORP. Further advanced TBC-based modes include ZCZ by Bhaumik, List, and Nandi [7], which offers full n-bit security, and THCTR by Dutta and Nandi [15], which achieves n-bit security dependent on tweak repetition, while the more recent HCTR+ by Datta et al. [12] is a BBB-secure mode that distinctively

achieves n-bit security independent of tweak repetition.

The third major methodological approach for circumventing the birthday bound is fresh re-keying, a procedural technique extensively analyzed by Abdalla and Bellare [1], and subsequently refined by Mennink [27], which mandates the frequent and periodic replacement of the underlying block cipher key, typically before the volume of processed data approaches the inherent birthday bound ($2^{n/2}$ blocks). By strictly limiting the number of blocks encrypted under any single, static key, and often employing a robust Key Derivation Function (KDF) to generate a new key, the probability of cryptographic collision is periodically reset, which allows the overall, aggregate data volume for a long session to significantly exceed the birthday bound limit while successfully preserving a high security margin against trade-off attacks. This principle contrasts with the structural modifications of the XORP and TBC methods by focusing on the temporal dimension of key usage constraints, and has been demonstrated as a viable technique, such as in Mennink's proposal for achieving another BBB secure version of the OCB mode [27].

Despite the theoretical and practical advancements in BBB security modes employing XORP, TBC, or Fresh Re-keying methodologies, none of these innovative designs have yet been formally standardized for widespread adoption. Consequently, recognizing the imperative to meet evolving high performance and high-security requirements, NIST has initiated a focused standardization effort for a new class of cryptographic mode known as the Accordion Mode.

The NIST Accordion Mode project transcends the objective of merely developing a superior iteration of established standards like GCM; it is formally defining a novel class of cryptographic primitive. The term Accordion functions as a descriptive metaphor for its operational capacity: the cipher accepts a message of arbitrary length and dynamically adjusts its internal transformation, compressing or extending, to produce a ciphertext of the same length while operating on the message as a single variable-length unit. Technically, the Accordion Mode is defined as a Tweakable Variable-Input-Length Strong Pseudorandom Permutation (VIL-SPRP) , representing a conceptual shift where the cipher dynamically expands and contracts to match the entire message length, thereby treating the entire message as a single, indivisible block. The construction must satisfy the rigorous requirement of being indistinguishable from a random permutation under adaptive encryption and decryption queries. Operating on the complete variable-length input in this manner provides the Wide Block behavior required of an Accordion construction. Furthermore, the primitive accepts a secondary

public input, the tweak, which serves to randomize the specific permutation within the family; this ensures that encrypting the identical message with the same key but a distinct tweak yields a completely uncorrelated ciphertext, a property critical for applications such as disk encryption (where the tweak represents the sector address) and network protocols (where the tweak often incorporates the sequence number). Nevertheless, this project expects the submission of multiple variants, including a construction that offers BBB security in conjunction with the primary mode that furnishes the indispensable functional features. This requirement guarantees that the probability of internal cryptographic collisions remains statistically negligible, even as the cumulative volume of processed data approaches the theoretical maximum security limit of the underlying primitive.

To meet these requirements, NIST proposed the development of variants of the HCTR2 algorithm [11] for the Accordion Mode standardization effort in June 2025. HCTR2 is considered a practical and flexible design that provides a suitable basis for incorporating the properties required from an Accordion construction. Consequently, in this article we present HCTR^{++}, a variant of HCTR2 designed to achieve a beyond-birthday-bound security profile under the stated hash, message-length, and tweak-repetition conditions. HCTR^{++} replaces the underlying block-cipher calls with fresh re-keying operations and the n-bit keyed almost-XOR-universal (AXU) hash functions with a $2n$-bit ε-uniform-AXU keyed hash family. The construction retains compatibility with standard block ciphers having matching key and block sizes, in particular AES-128. Moreover, the design shown in Figure 3 exhibits a symmetric structure between its encryption and decryption procedures, reducing the structural complexity of implementation.

The remainder of this paper is organized as follows: Section 2 delineates established methodology fresh re-keying for achieving BBB security. Section 3 elucidates the architectural mechanics of the HCTR2 algorithm, the proposed design HCTR^{++} BBB-secure variant. Finally, Section 4 presents the formal security analysis and establishes the SPRP security bound of the construction under the stated assumptions.

2 Achieving Beyond Birthday Bound Security

To exceed the conventional birthday bound of $O(2^{n/2})$, a mode of operation must disrupt the collision probability inherent in using a fixed permutation over a growing data set. As previously mentioned, there are three dominant approaches to achieve this: XOR of Permutations, tweakable block ciphers,

and fresh re-keying. In this work, we utilize fresh re-keying method to construct a BBB secure variant of HCTR2. We explain this method in more detail below.

2.1 Fresh Re-keying

In the landscape of BBB cryptography, Fresh Re-keying stands out as a radical approach. Rather than modifying the mode of operation or the internal structure of the cipher, Fresh Re-keying mitigates the birthday bound by ensuring that the underlying block cipher instance is never used enough times for collisions to become statistically probable.

Its methodology addresses the birthday bound by fundamentally limiting the operational lifespan of any single key instance, K. As the quantity of data blocks (q) processed by a static block cipher E_K increases, the probability of an adversary distinguishing the resulting ciphertext from a truly random permutation grows according to the Birthday Paradox. Critically, the birthday bound, $q \approx 2^{n/2}$, applies specifically to the number of blocks encrypted under a single, fixed key. Fresh re-keying effectively neutralizes this constraint by generating a new, ephemeral session key for every individual block or small chunk of blocks. The encryption of a message block M using a master key MK and a unique nonce/counter r is formally defined by the sequence $K_r = F_{MK}(r)$ for key derivation and $C = E_{K_r}(M)$ for the encryption operation, where F is a Pseudorandom Function (PRF) and E is the underlying block cipher. With a secure re-keying function, distinct values of r generate ephemeral keying material that behaves as fresh key material up to the distinguishing and collision probabilities guaranteed by the re-keying construction. Consequently, the underlying block cipher is not repeatedly exercised under a single static key.

Fresh re-keying constructions are commonly analyzed by separating the security of the key-derivation mechanism from that of the underlying block cipher, often using an ideal-cipher model or corresponding computational assumptions. With a static key, an adversary observes q outputs derived from one permutation, and the security is limited by the collision probability $O(q^2/2^n)$. In contrast, a secure re-keying construction distributes the evaluations across freshly derived cipher instances, so that the adversary no longer observes q evaluations under a single static permutation. The resulting security therefore depends on the collision and distinguishing bounds of the re-keying construction rather than only on the birthday bound associated with repeated use of a single key.

This methodological approach was initially proposed by Abdalla and Bel-

lare [1] to enhance the permissible usage count of the master key. Subsequently, Dobraunig et al. [13] advanced this concept by proposing two distinct re-keying schemes, one of which achieves BBB security through the strategic leveraging of a TBC. Building upon this foundation, Mennink [27] conducted further research into optimizing the efficiency of combining BBB-secure TBCs with re-keying, leading to the proposal of three distinct fresh re-keying schemes. These schemes, detailed in Mennink's work, are explicitly based on: the Mennink TBC construction [26], one of the TBC variants proposed by Wang et al. [31], and the generalized XHX construction by Jha et al [20].

2.2 Mennink's Fresh Re-keying R3 Scheme

The proposed construction employs Mennink's R3 fresh re-keying scheme as its internal re-keying primitive. As illustrated in Figure 1, the R3 scheme is architecturally influenced by the tweakable block cipher paradigm. For an n-bit re-keying input r, the scheme evaluates a $2n$-bit keyed hash as $u \parallel v \leftarrow H_{\overline{K}}(r)$, where the first n bits form the ephemeral block-cipher key u and the remaining n bits form the whitening value v. The resulting operation $\widetilde{E}_{\overline{K}}(r, X) = E_u(X \oplus v) \oplus v$ therefore refreshes the effective block-cipher instance as the re-keying input changes. This mechanism is used as the fresh re-keying component of HCTR^{++}.

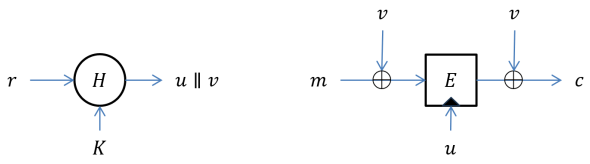


Figure 1: Re-keying Construction R3

3 HCTR2 Mechanics and BBB Variant Construction

Driven by the limitations inherent in current modes of operation, NIST has identified HCTR2 [11] as a candidate for the development of new cryptographic modes within the Accordion Mode standardization effort. HCTR2 is a variable-input-length, tweakable strong pseudorandom permutation, a

requisite property for the proposed new standard. Structurally, the algorithm is based on the Hash-Encrypt-Hash (HEH) construction. To facilitate its use as an Accordion mode, researchers have proposed developing variants that incorporate additional security and functional properties. This section details the operational mechanics of HCTR2 and subsequently introduces the proposed BBB-secure HCTR^{++} variant, which combines Mennink's R3 fresh re-keying scheme with a $2n$-bit ε-uniform-AXU keyed hash family.

3.1 Working Mechanism of HCTR2

HCTR2, which is given in the Figure 2, employs a three-pass structure reminiscent of a Feistel network, utilizing the HEH paradigm to efficiently process plaintext of arbitrary length while maintaining high security bounds. The algorithm operates under a single block cipher key, K. To manage internal hashing and enforce domain separation without necessitating additional key material, HCTR2 derives two internal sub-keys during initialization: a polynomial hashing key, $\overline{h} \leftarrow E_K(\text{bin}(0))$, and a masking key, $L \leftarrow E_K(\text{bin}(1))$ where $\text{bin}(x)$ is little-endian n-bit binary representation of x.

For a given tweak T and plaintext P, the input is partitioned into a fixed-size leading block, P_1 (n bits), and a variable-length remainder, $\overline{P}$. The process initiates with a polynomial hash function, $H_{\overline{h}}$, which processes the tweak T and the remainder $\overline{P}$. The resulting digest is XORed with the leading block P_1 to generate an intermediate value, PP. This value is subsequently encrypted via the underlying block cipher to produce CC.

A critical component of HCTR2 is the generation of the internal state J (acting as the initialization vector for the XCTR phase), which is derived by XORing the input and output of the block cipher encryption with the masking key L (i.e., $J = PP \oplus CC \oplus L$). The XCTR mode then generates a keystream by encrypting a sequence of counters XORed with J (denoted as $E_K(J \oplus 1) \parallel E_K(J \oplus 2) \dots$). This keystream is XORed with the remainder $\overline{P}$ to yield the ciphertext remainder $\overline{C}$. In the final phase, the algorithm re-hashes the tweak T and the newly generated ciphertext remainder $\overline{C}$; this hash digest is XORed with CC to derive the final leading ciphertext block, C_1. The complete ciphertext is formed by the concatenation $C = C_1 \parallel \overline{C}$.

3.2 BBB Variant HCTR2: HCTR^{++}

Securing the HCTR2 algorithm at the BBB level requires modifications to the two components that dominate the internal collision analysis. First, the n-bit keyed hash function of HCTR2 is replaced by a $2n$-bit keyed hash family

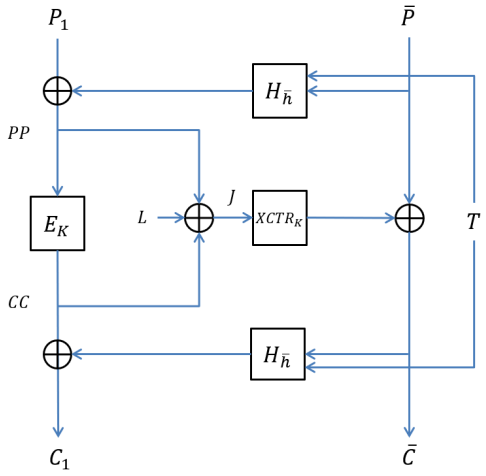


Figure 2: HCTR2 Algorithm

$H : \mathcal{K}_H \times \mathcal{D} \to \{0,1\}^{2n}$ satisfying the ε-uniform-AXU conditions of Definition 2. Second, the block-cipher calls in the HCTR2 data path are replaced by Mennink's R3 fresh re-keying construction [27]. Accordingly, $\mathcal{D}$ denotes the full input domain of H used by HCTR^{++}, comprising the pairs (X, T) supplied to the outer and central hash evaluations and the n-bit re-keying inputs supplied to R3. The security analysis does not treat the resulting re-keyed block-cipher instances as deterministically independent. Instead, possible collisions in their derivation are bounded through the ε-uniform-AXU property of H, while their deviation from ideal re-keyed permutations is bounded by the R3 security result. The security theorem in Section 4 is stated for messages containing at least two n-bit blocks.

Detailed in Figure 3, HCTR^{++} modifies the HCTR2 construction in three principal respects. It replaces the static block-cipher calls with Mennink's R3 re-keying operations, replaces the n-bit universal hash function with the $2n$-bit ε-uniform-AXU hash family H of Definition 2, and replaces XCTR with the R3-based CTR^{++} construction. A final inverse R3 operation is also introduced in the leading-block path so that both halves of the outer hash output affect the transformation while preserving the symmetry between encryption and decryption. As in HCTR2, HCTR^{++} takes a single master key K, from which the internal hash keys $\overline{h}_1$, $\overline{h}_2$, and $\overline{K}$ are derived. The same hash family H is used for the outer, central, and re-keying evaluations described above, under the corresponding internal hash keys, while the central hash output $r \parallel J$ provides the values used by CTR^{++}. The complete

encryption and decryption procedures, including the symmetric placement of the forward and inverse R3 operations, are specified in Figure 4.

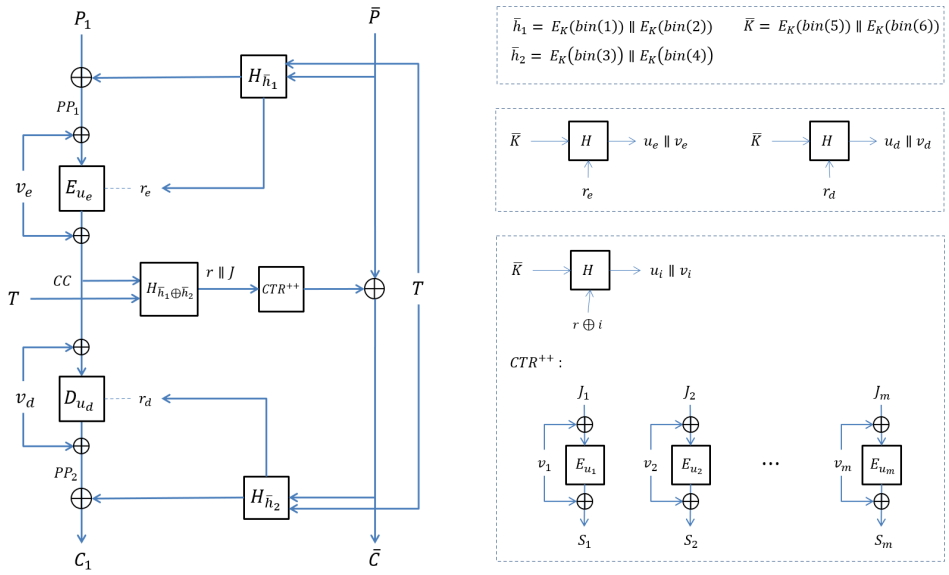


Figure 3: HCTR^{++} Algorithm

3.3 Design Rationale

HCTR^{++} demonstrates how established cryptographic primitives can be integrated to achieve beyond-birthday-bound security. The design is intended to retain compatibility with standard block ciphers having matching key and block sizes, in particular AES-128. Accordingly, the proposed construction supports widely deployed block-cipher primitives without requiring a native tweakable block cipher.

The fresh re-keying methodology was adopted because a TBC-based design would require either a dedicated tweakable block cipher or an additional construction for building one from a standard block cipher, reducing the direct compatibility of the design with primitives such as AES. Moreover, an XOR-sum-of-permutations approach would require more substantial structural modifications to preserve the reversible encryption and decryption paths of HCTR2. Fresh re-keying therefore provides a natural means of improving the security bound while preserving the basic architecture of HCTR2 and its use of a conventional block cipher.

Algorithm 1 $\mathbf{Enc}(K, \mathbf{T}, P)$

$P_1 \leftarrow \mathrm{MSB}_n(P)$
$\overline{P} \leftarrow \mathrm{LSB}_{|P|-n}(P)$
$\overline{h}_1 \leftarrow E_K(bin(1)) \parallel E_K(bin(2))$
$\overline{h}_2 \leftarrow E_K(bin(3)) \parallel E_K(bin(4))$
$\overline{K} \leftarrow E_K(bin(5)) \parallel E_K(bin(6))$
$PP_1 \leftarrow P_1 \oplus \mathrm{MSB}_n(H_{\overline{h}_1}(\overline{P}, T))$
$r_e \leftarrow \mathrm{LSB}_n(H_{\overline{h}_1}(\overline{P}, T))$
$u_e \parallel v_e \leftarrow H_{\overline{K}}(r_e)$
$CC \leftarrow E_{u_e}(PP_1 \oplus v_e) \oplus v_e$
$r \parallel J \leftarrow H_{\overline{h}_1 \oplus \overline{h}_2}(CC, T)$
$S \leftarrow \mathrm{CTR}^{++}(J)$
  $s = |\overline{P}|, \quad m = \lceil s/n \rceil, \quad i = 1$
  for $i \leq m$ **do**
    $J_i = J \oplus i, \quad r_i = r \oplus i$
    $u_i \parallel v_i \leftarrow H_{\overline{K}}(r_i)$
    $S_i \leftarrow E_{u_i}(J_i \oplus v_i) \oplus v_i$
    $i \leftarrow i + 1$
  end for
  $S = \mathrm{MSB}_s(S_1 \parallel S_2 \parallel \cdots \parallel S_m)$
$\overline{C} \leftarrow \overline{P} \oplus S$
$r_d \leftarrow \mathrm{LSB}_n(H_{\overline{h}_2}(\overline{C}, T))$
$u_d \parallel v_d \leftarrow H_{\overline{K}}(r_d)$
$PP_2 \leftarrow D_{u_d}(CC \oplus v_d) \oplus v_d$
$C_1 \leftarrow PP_2 \oplus \mathrm{MSB}_n(H_{\overline{h}_2}(\overline{C}, T))$
$C \leftarrow C_1 || \overline{C}$
return C

Algorithm 2 $\mathbf{Dec}(K, \mathbf{T}, C)$

$C_1 \leftarrow \mathrm{MSB}_n(C)$
$\overline{C} \leftarrow \mathrm{LSB}_{|C|-n}(C)$
$\overline{h}_1 \leftarrow E_K(bin(1)) \parallel E_K(bin(2))$
$\overline{h}_2 \leftarrow E_K(bin(3)) \parallel E_K(bin(4))$
$\overline{K} \leftarrow E_K(bin(5)) \parallel E_K(bin(6))$
$PP_1 \leftarrow C_1 \oplus \mathrm{MSB}_n(H_{\overline{h}_2}(\overline{C}, T))$
$r_e \leftarrow \mathrm{LSB}_n(H_{\overline{h}_2}(\overline{C}, T))$
$u_e \parallel v_e \leftarrow H_{\overline{K}}(r_e)$
$CC \leftarrow E_{u_e}(PP_1 \oplus v_e) \oplus v_e$
$r \parallel J \leftarrow H_{\overline{h}_1 \oplus \overline{h}_2}(CC, T)$
$S \leftarrow \mathrm{CTR}^{++}(J)$
  $s = |\overline{C}|, \quad m = \lceil s/n \rceil, \quad i = 1$
  for $i \leq m$ **do**
    $J_i = J \oplus i, \quad r_i = r \oplus i$
    $u_i \parallel v_i \leftarrow H_{\overline{K}}(r_i)$
    $S_i \leftarrow E_{u_i}(J_i \oplus v_i) \oplus v_i$
    $i \leftarrow i + 1$
  end for
  $S = \mathrm{MSB}_s(S_1 \parallel S_2 \parallel \cdots \parallel S_m)$
$\overline{P} \leftarrow \overline{C} \oplus S$
$r_d \leftarrow \mathrm{LSB}_n(H_{\overline{h}_1}(\overline{P}, T))$
$u_d \parallel v_d \leftarrow H_{\overline{K}}(r_d)$
$PP_2 \leftarrow D_{u_d}(CC \oplus v_d) \oplus v_d$
$P_1 \leftarrow PP_2 \oplus \mathrm{MSB}_n(H_{\overline{h}_1}(\overline{P}, T))$
$P \leftarrow P_1 || \overline{P}$
return P

Figure 4: HCTR^{++} Pseudocode

The design incorporates several modifications required by the integration. In particular, replacing the original n-bit hash output with a $2n$-bit ε-uniform-AXU hash output requires both halves of the hash value to influence the transformation. For this purpose, an additional R3 operation is introduced in the leading-block path. This operation is chosen as an inverse R3 call, preserving the symmetric relationship between the encryption and decryption procedures while allowing the complete outer-hash output to affect the resulting ciphertext.

Despite the additional computational cost introduced by re-keying, HCTR^{++} retains compatibility with standard block ciphers while achieving a beyond-birthday-bound security profile for bounded message lengths, sufficiently

small hash parameter ε, and sufficiently restricted tweak repetition. Compared with block-cipher-based constructions such as bbb-ddd-AES, CHCTR, and HCTR2-TwKD, HCTR^{++} offers a different trade-off between security level, primitive-call complexity, and key requirements. TBC-based schemes such as THCTR and HCTR+ provide comparable security objectives but rely on a tweakable block-cipher primitive rather than directly using a conventional block cipher. These distinctions are summarized in Table 1, which compares the schemes according to their underlying primitives, security bounds, operation counts, and key requirements.

Table 1: Comparison with BBB Secure Tweakable Enciphering Schemes

Design		Security	Cipher	Operation Count	Key Count
TCT_2	[30]	$O(2^{2n/3})$	BC	$(6m+6)$BC + $(7m+7)$HF	$m+s+19$
THCTR	[15]	$O(2^n)^{\ddagger}$	TBC	mTBC + 3HF	3
bbb-ddd-AES	[14]	$O(2^{2n/3})^{\ddagger}$	BC	$(2m+4)$BC + 2HF	3
Db-HCTR	[23]	$O(2^{2n/3})$	TBC/BC	1TBC + $(m-1)$BC + 2HF	3
HCTR+	[12]	$O(2^n)$	TBC	$(m+2)$TBC + 2HF	1
CHCTR	[9]	$O(2^{2n/3})$	BC	$2m$BC + 4HF	2
HCTR2-TwKD	[9]	$O(2^{2n/3})^{\ddagger}$	BC	mBC + 2HF + 1KDF	1
HCTR^{++}		$\boldsymbol{O(2^n)^{\ddagger}}$	**BC**	$\boldsymbol{(m+7)}$**BC** + $\boldsymbol{(m+4)}$**HF**	**1**

BC: Block Cipher, TBC: Tweakable Block Cipher, HF: Hash Function, KDF : Key Derivation Function, n: block size, m: number of blocks of a specific message

‡The security levels of these algorithms depend on the number of data blocks under the same tweaks.

4 Security

In this section, we establish the security of HCTR^{++} within the framework of strong pseudorandom permutations as defined by Bellare et al. [3]. The proof follows the code-based game-playing methodology used in the original analysis, but the hash function is retained throughout the games and is analyzed only through its uniformity and almost-XOR-universality properties.

Definition 1. Let $\mathcal{K}$ and $\mathcal{M}$ denote the key space and message space, respectively. Let $\mathrm{Perm}(\mathcal{M})$ denote the set of all length-preserving permutations on $\mathcal{M}$. Consider a block cipher $E : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$. The SPRP advantage of a distinguishing adversary $\mathcal{A}$ against E is defined as

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

Here, $K \xleftarrow{\$} \mathcal{K}$ denotes sampling a key uniformly at random, and the notation "$\Rightarrow 1$" denotes the event that the adversary outputs 1.

This advantage quantifies the adversary's ability to distinguish the block cipher from a uniform random permutation given adaptive access to both encryption and decryption oracles. A negligible advantage implies security against both chosen-plaintext and chosen-ciphertext attacks.

Definition 2 (ε-uniform-AXU hash family)**.** Let $\mathcal{K}_H = \{0,1\}^{2n}$ be the hash-key space, and let $\mathcal{D}$ be the complete input domain on which H is invoked by HCTR^{++}. In particular, $\mathcal{D}$ comprises the pairs $(\overline{P}, T)$ and $(\overline{C}, T)$ used in the outer-hash evaluations, the pairs (CC, T) used in the central-hash evaluations, and the n-bit inputs used by the R3 re-keying operations. Let $H : \mathcal{K}_H \times \mathcal{D} \longrightarrow \{0,1\}^{2n}$ be a keyed hash family.

The family H is called ε-uniform if, for every $x \in \mathcal{D}$ and every $y \in \{0,1\}^{2n}$,

$$\Pr_{h \xleftarrow{\$} \mathcal{K}_H}\left[H_h(x) = y\right] \leq \varepsilon.$$

It is called ε-almost-XOR-universal (ε-AXU) if, for every pair of distinct inputs $x, x' \in \mathcal{D}$ and every $\delta \in \{0,1\}^{2n}$,

$$\Pr_{h \xleftarrow{\$} \mathcal{K}_H}\left[H_h(x) \oplus H_h(x') = \delta\right] \leq \varepsilon.$$

We call H an ε-uniform-AXU hash family if it satisfies both properties.

We now state the security of HCTR^{++} for an arbitrary $2n$-bit ε-uniform-AXU keyed hash family satisfying Definition 2 over the input domains used by the construction. The theorem does not assume any particular realization of H, and all hash-dependent terms in the bound are therefore expressed in terms of ε. The analysis is stated in the ideal-cipher model underlying the R3 analysis of [27]. To obtain a standard-model statement for a concrete block cipher, the appropriate multi-key PRP distinguishing advantage must be added to the bound.

Theorem 1. *Let E be an ideal cipher with n-bit keys and n-bit blocks, and let H be a $2n$-bit ε-uniform-AXU keyed hash family as in Definition 2. Let* HCTR$^{++}[E, H]$ *denote the one-key construction over messages of at least*

two blocks. Consider an adversary $\mathcal{A}$ making at most q encryption and decryption queries, where at most μ queries use the same tweak. For the r-th construction query, let m_r be the number of blocks processed in the remainder, and define

$$Q := \sum_{r=1}^{q} (m_r + 2) \le q(a+1),$$

where a is the maximum number of blocks in one queried message. Thus, Q is the total number of forward and inverse calls to the R3 re-keying primitive. Let p be the primitive-query budget in the R3 ideal-cipher reduction. Then

$$\begin{aligned}\mathrm{Adv}^{\mathrm{SPRP}}_{\mathrm{HCTR}^{++}[E,H]}(\mathcal{A}) \le\ & 2Q(Q-1+p)\varepsilon + q\,2^n\varepsilon \\ & + \frac{Q+p+15+q(\mu-1)}{2^n} \\ & + \frac{2Q(Q-1)+q^2}{2^{2n+1}}.\end{aligned}$$

In particular, when $\varepsilon = O(L/2^{2n})$ for a bounded message-length factor L, $p = O(Q)$, and the tweak multiplicity μ is bounded, the theorem gives

$$\mathrm{Adv}^{\mathrm{SPRP}}_{\mathrm{HCTR}^{++}[E,H]}(\mathcal{A}) = O\left(\frac{Q^2 L}{2^{2n}} + \frac{Q}{2^n}\right),$$

and therefore yields a beyond-birthday-bound security profile.

Proof. Let $\mathcal{A}$ denote an adversary that makes at most q external queries. Each query contains at least two blocks and at most a blocks. Without loss of generality, we assume that $\mathcal{A}$ makes no redundant forward or inverse query. We define a sequence of games, beginning with the real construction and ending with the ideal random-function experiment used in the SPRP-to-SPRF transition. Throughout the proof, all hash keys are sampled or derived once at initialization and remain fixed for the complete interaction.

For the purpose of the game sequence, let $\widetilde{E} = \mathrm{R3}[E, H_{\overline{K}}]$ denote the concrete R3 re-keying primitive constructed from E and H. Accordingly, the one-key construction $\mathrm{HCTR}^{++}[E,H]$ may equivalently be viewed as $\mathrm{HCTR}^{++}[\widetilde{E},H]$.

Game0. This game embodies the real attack environment. The adversary $\mathcal{A}$ interacts with an oracle that faithfully simulates the $\mathrm{HCTR}^{++}[\widetilde{E},H]$ construction. The R3 re-keying primitive is

$$\widetilde{E}_{\overline{K}}(r,X) = E_u(X \oplus v) \oplus v, \qquad u \parallel v \leftarrow H_{\overline{K}}(r),$$

where the outer and central calls to H use the keys $\overline{h}_1$, $\overline{h}_2$, and $\overline{h}_1 \oplus \overline{h}_2$ derived as specified in Figure 4. Consequently,

$$\Pr[\mathcal{A}^{\mathrm{HCTR}^{++}[E,H]} \Rightarrow 1] = \Pr[\mathcal{A}^{\mathsf{Game0}} \Rightarrow 1].$$

Game0a. This game differs from Game0 only in the derivation of the three internal $2n$-bit keys. Instead of computing

$$\begin{aligned}\overline{h}_1 &= E_K(\mathrm{bin}(1)) \parallel E_K(\mathrm{bin}(2)),\\ \overline{h}_2 &= E_K(\mathrm{bin}(3)) \parallel E_K(\mathrm{bin}(4)),\\ \overline{K} &= E_K(\mathrm{bin}(5)) \parallel E_K(\mathrm{bin}(6)),\end{aligned}$$

the game samples $U_1, \ldots, U_6 \xleftarrow{\$} \{0,1\}^n$ and sets

$$\overline{h}_1 = U_1 \parallel U_2, \qquad \overline{h}_2 = U_3 \parallel U_4, \qquad \overline{K} = U_5 \parallel U_6.$$

These values are sampled once and remain fixed throughout the interaction. Thus, $\overline{h}_1$, $\overline{h}_2$, and $\overline{K}$ are mutually independent and uniform; in addition, every pair among $\overline{h}_1$, $\overline{h}_2$, and $\overline{h}_1 \oplus \overline{h}_2$ is independent and uniform.

To justify this transition while preserving the ideal-cipher interface, define $\mathsf{bad}_{\mathrm{KD}}$ to occur if two of $U_1, \ldots, U_6$ coincide, or if one of the Q internal block-cipher calls made by R3 or one of the p primitive queries is evaluated under the hidden master key K. If $\mathsf{bad}_{\mathrm{KD}}$ does not occur, the six values can be coupled with the six distinct outputs of the permutation E_K at the fixed derivation points, and every other ideal-cipher call is made under a key different from K. Therefore, the two games are identical until $\mathsf{bad}_{\mathrm{KD}}$. The collision event among the six sampled values satisfies

$$\Pr[\exists i \neq j : U_i = U_j] \leq \binom{6}{2} 2^{-n} = 15 \cdot 2^{-n}.$$

Conditioned on any transcript that has not yet set $\mathsf{bad}_{\mathrm{KD}}$, the hidden key K remains uniform and independent of the key used by the next non-derivation cipher call. Hence each of the at most $Q+p$ such calls hits K with probability 2^{-n}. A union bound therefore gives

$$\Pr[\mathsf{bad}_{\mathrm{KD}}] \leq \frac{Q+p+15}{2^n}.$$

It follows from the Fundamental Lemma of Game-Playing that

$$\left|\Pr[\mathcal{A}^{\mathsf{Game0}} \Rightarrow 1] - \Pr[\mathcal{A}^{\mathsf{Game0a}} \Rightarrow 1]\right| \leq \frac{Q+p+15}{2^n}.$$

Game1. Starting from Game0a, we replace the concrete R3 re-keying scheme $\widetilde{E}$ with an ideal re-keying permutation $\widetilde{\pi}$, while retaining the hash family H. This results in the construction $\mathrm{HCTR}^{++}[\widetilde{\pi}, H]$. The simulation makes

$$Q = \sum_{r=1}^{q} (m_r + 2)$$

forward or inverse construction queries to R3.

The restriction of H to the n-bit re-keying inputs satisfies the hash requirements of Theorem 3 of Mennink [27]. Its uniformity follows directly from 2. Moreover, full $2n$-bit ε-AXU implies the required n-partial-XOR-uniformity with the same parameter: for every pair of distinct $r, r' \in \{0,1\}^n$ and every $v \in \{0,1\}^n$,

$$\Pr_{h \xleftarrow{\$} \mathcal{K}_H} \left[H_h(r) \oplus H_h(r') = 0^n \parallel v \right] \leq \varepsilon,$$

since $0^n \parallel v$ is one complete $2n$-bit difference covered by the AXU condition. Hence, for Q construction queries and p primitive queries,

$$\mathrm{Adv}^{\mathrm{stprp}}_{\widetilde{E}}(Q, p) \leq Q(Q - 1 + 2p)\varepsilon.$$

Therefore, by a standard hybrid argument,

$$\begin{aligned} \mathrm{Adv}^{\mathrm{SPRP}}_{\mathrm{HCTR}^{++}[E,H]}(\mathcal{A}) \leq \frac{Q + p + 15}{2^n} + \mathrm{Adv}^{\mathrm{SPRP}}_{\mathrm{HCTR}^{++}[\widetilde{\pi},H]}(\mathcal{A}) \\ + \mathrm{Adv}^{\mathrm{stprp}}_{\widetilde{E}}(Q, p). \end{aligned}$$

The ideal re-keying permutation is simulated by lazy sampling, following Halevi and Rogaway [18]. Let D and R be the multisets of inputs and outputs, respectively, of the internal re-keying calls, including their re-keying indices. Let T be the multiset of central-hash inputs (T, CC). These multisets, the table defining $\widetilde{\pi}$, and the flag bad are initialized once before the first query and are retained throughout the interaction. The keys $\overline{h}_1$ and $\overline{h}_2$ are also sampled once at initialization. For an encryption query $(P_1^r \parallel \overline{P}^r, T^r)$, the three hash values used by the simulation are

$$\begin{aligned} X_1^r &\leftarrow H_{\overline{h}_1}(\overline{P}^r, T^r), \\ (r \parallel J)^r &\leftarrow H_{\overline{h}_1 \oplus \overline{h}_2}(CC^r, T^r), \\ X_2^r &\leftarrow H_{\overline{h}_2}(\overline{C}^r, T^r), \end{aligned}$$

and for a decryption query $(C_1^r \parallel \overline{C}^r, T^r)$ they are

$$\begin{aligned} X_1^r &\leftarrow H_{\overline{h}_2}(\overline{C}^r, T^r), \\ (r \parallel J)^r &\leftarrow H_{\overline{h}_1 \oplus \overline{h}_2}(CC^r, T^r), \\ X_2^r &\leftarrow H_{\overline{h}_1}(\overline{P}^r, T^r). \end{aligned}$$

All calls to $\widetilde{\pi}$ and $\widetilde{\pi}^{-1}$ are lazily sampled as described above. In particular, CC^r, the values S_i^r, and PP_2^r are sampled uniformly subject to permutation consistency. The flag bad is set whenever a newly sampled domain or range tuple collides with a previously defined tuple, or whenever a central-hash input repeats in T. The adversary has no visibility of this flag.

To facilitate the subsequent analysis, we use the standard SPRP-to-SPRF switching argument [5]. Since every queried message contains at least $2n$ bits,

$$\begin{aligned} \mathrm{Adv}^{\mathrm{SPRP}}_{\mathrm{HCTR}^{++}[E,H]}(\mathcal{A}) \le \frac{Q+p+15}{2^n} + \mathrm{Adv}^{\mathrm{SPRF}}_{\mathrm{HCTR}^{++}[\widetilde{\pi},H]}(\mathcal{A}) \\ + \mathrm{Adv}^{\mathrm{stprp}}_{\widetilde{E}}(Q,p) + \frac{q^2}{2^{2n+1}}. \end{aligned}$$

Game2. This game is derived from Game1 by eliminating the consistency checks that enforce the permutation property of $\widetilde{\pi}$. Every invocation of the internal forward or inverse primitive now receives a fresh uniformly random n-bit output. The flag bad is retained. It is set whenever a fresh assignment would collide with a domain or range tuple already defined in Game1, and also whenever a central-hash input repeats in T. By the Fundamental Lemma of Game-Playing,

$$\left|\Pr[\mathcal{A}^{\mathsf{Game1}} \Rightarrow 1] - \Pr[\mathcal{A}^{\mathsf{Game2}} \Rightarrow 1]\right| \le \Pr[\mathsf{Game2\ sets\ bad}].$$

Furthermore, the fresh uniform values S_i^r and PP_2^r propagate directly to the response returned to the adversary. The blocks S_i^r form the mask applied to the returned remainder, while PP_2^r masks the returned first block. Hence, conditioned on the preceding transcript and on the current nonredundant query, the response is a fresh uniform string of the appropriate length. This conclusion does not require the hash evaluations to be random-function outputs: once their values are fixed, the fresh masks S_i^r and PP_2^r still make the corresponding output components uniform. Consequently,

$$\Pr[\mathcal{A}^{\mathsf{Game2}} \Rightarrow 1] = \Pr[\mathcal{A}^{\mathrm{SPRF}} \Rightarrow 1].$$

Therefore,

$$\mathrm{Adv}^{\mathrm{SPRF}}_{\mathrm{HCTR}^{++}[\widetilde{\pi},H]}(\mathcal{A}) \le \Pr[\mathsf{Game2\ sets\ bad}].$$

Algorithm 3 Game1 – r^{th} Encryption Query $(P_1^r || \overline{P^r}, T^r)$

Shared Game1 initialization. Before the first encryption or decryption query, set $\mathsf{bad} \leftarrow \textbf{false}$, $\mathrm{D} \leftarrow \mathrm{R} \leftarrow \mathsf{T} \leftarrow \emptyset$, and $\widetilde{\pi} \leftarrow \textbf{undef}$. For $\rho \in \{0,1\}^n$, let $\overline{\mathrm{D}}_\rho = \{0,1\}^n \setminus \{X : (\rho, X) \in \mathrm{D}\}$ and $\overline{\mathrm{R}}_\rho = \{0,1\}^n \setminus \{Y : (\rho, Y) \in \mathrm{R}\}$; $\widetilde{\pi}^{-1}$ is maintained implicitly.

$X_1^r \leftarrow H_{\overline{h}_1}(\overline{P^r}, T^r)$
$r_e^r \leftarrow \mathrm{LSB}_n(X_1^r)$
$PP_1^r \leftarrow P_1^r \oplus \mathrm{MSB}_n(X_1^r)$
$CC^r \xleftarrow{\$} \{0,1\}^n$
if $(r_e^r, CC^r) \in \mathrm{R}$ **then** $\mathsf{bad} \leftarrow \textbf{true}$, $\boxed{CC^r \xleftarrow{\$} \overline{\mathrm{R}}_{r_e^r}}$ **end if**
if $(r_e^r, PP_1^r) \in \mathrm{D}$ **then** $\mathsf{bad} \leftarrow \textbf{true}$, $\boxed{CC^r \leftarrow \widetilde{\pi}(r_e^r, PP_1^r)}$ **end if**
$\widetilde{\pi}(r_e^r, PP_1^r) \leftarrow CC^r, \quad \mathrm{D} \leftarrow \mathrm{D} \uplus \{(r_e^r, PP_1^r)\}, \quad \mathrm{R} \leftarrow \mathrm{R} \uplus \{(r_e^r, CC^r)\}$
$(r \parallel J)^r \leftarrow H_{\overline{h}_1 \oplus \overline{h}_2}(CC^r, T^r)$
if $(T^r, CC^r) \in \mathsf{T}$ **then** $\mathsf{bad} \leftarrow \textbf{true}$ **end if**
$\mathsf{T} \leftarrow \mathsf{T} \uplus \{(T^r, CC^r)\}$
  $s^r \leftarrow |\overline{P^r}|, \quad m_r \leftarrow \lceil s^r / n \rceil$
  for $i = 1$ **to** m_r **do**
    $J_i^r \leftarrow J^r \oplus i, \quad r_i^r \leftarrow r^r \oplus i$
    $S_i^r \xleftarrow{\$} \{0,1\}^n$
    if $(r_i^r, S_i^r) \in \mathrm{R}$ **then** $\mathsf{bad} \leftarrow \textbf{true}$, $\boxed{S_i^r \xleftarrow{\$} \overline{\mathrm{R}}_{r_i^r}}$ **end if**
    if $(r_i^r, J_i^r) \in \mathrm{D}$ **then** $\mathsf{bad} \leftarrow \textbf{true}$, $\boxed{S_i^r \leftarrow \widetilde{\pi}(r_i^r, J_i^r)}$ **end if**
    $\widetilde{\pi}(r_i^r, J_i^r) \leftarrow S_i^r, \quad \mathrm{D} \leftarrow \mathrm{D} \uplus \{(r_i^r, J_i^r)\}, \quad \mathrm{R} \leftarrow \mathrm{R} \uplus \{(r_i^r, S_i^r)\}$
  end for
  $S^r \leftarrow \mathrm{MSB}_{s^r}(S_1^r || S_2^r || \cdots || S_{m_r}^r)$
$\overline{C^r} \leftarrow \overline{P^r} \oplus S^r$
$X_2^r \leftarrow H_{\overline{h}_2}(\overline{C^r}, T^r)$
$r_d^r \leftarrow \mathrm{LSB}_n(X_2^r)$
$PP_2^r \xleftarrow{\$} \{0,1\}^n$
if $(r_d^r, PP_2^r) \in \mathrm{D}$ **then** $\mathsf{bad} \leftarrow \textbf{true}$, $\boxed{PP_2^r \xleftarrow{\$} \overline{\mathrm{D}}_{r_d^r}}$ **end if**
if $(r_d^r, CC^r) \in \mathrm{R}$ **then** $\mathsf{bad} \leftarrow \textbf{true}$, $\boxed{PP_2^r \leftarrow \widetilde{\pi}^{-1}(r_d^r, CC^r)}$ **end if**
$\widetilde{\pi}^{-1}(r_d^r, CC^r) \leftarrow PP_2^r, \quad \mathrm{D} \leftarrow \mathrm{D} \uplus \{(r_d^r, PP_2^r)\}, \quad \mathrm{R} \leftarrow \mathrm{R} \uplus \{(r_d^r, CC^r)\}$
$C_1^r \leftarrow PP_2^r \oplus \mathrm{MSB}_n(X_2^r)$
$C^r \leftarrow C_1^r || \overline{C^r}$
return C^r

Game3. This game changes only the order in which the hidden values of Game2 are generated. During the interaction, conditioned on the preceding transcript, each nonredundant encryption or decryption query is answered with a fresh uniform string of the appropriate length. Once all q queries have been completed, the game reconstructs the internal values associated with the resulting transcript.

Algorithm 4 Game1 – r^{th} Decryption Query $(C_1^r||\overline{C^r}, T^r)$

$X_1^r \leftarrow H_{\overline{h}_2}(\overline{C^r}, T^r)$
$r_e^r \leftarrow \mathrm{LSB}_n(X_1^r)$
$PP_1^r \leftarrow C_1^r \oplus \mathrm{MSB}_n(X_1^r)$
$CC^r \xleftarrow{\$} \{0,1\}^n$
if $(r_e^r, CC^r) \in \mathrm{R}$ **then** bad $\leftarrow$ **true**, $\boxed{CC^r \xleftarrow{\$} \overline{\mathrm{R}}_{r_e^r}}$ **end if**
if $(r_e^r, PP_1^r) \in \mathrm{D}$ **then** bad $\leftarrow$ **true**, $\boxed{CC^r \leftarrow \widetilde{\pi}(r_e^r, PP_1^r)}$ **end if**
$\widetilde{\pi}(r_e^r, PP_1^r) \leftarrow CC^r, \quad \mathrm{D} \leftarrow \mathrm{D} \uplus \{(r_e^r, PP_1^r)\}, \quad \mathrm{R} \leftarrow \mathrm{R} \uplus \{(r_e^r, CC^r)\}$
$(r \parallel J)^r \leftarrow H_{\overline{h}_1 \oplus \overline{h}_2}(CC^r, T^r)$
if $(T^r, CC^r) \in \mathsf{T}$ **then** bad $\leftarrow$ **true end if**
$\mathsf{T} \leftarrow \mathsf{T} \uplus \{(T^r, CC^r)\}$
  $s^r \leftarrow |\overline{C^r}|, \quad m_r \leftarrow \lceil s^r/n \rceil$
  for $i = 1$ **to** m_r **do**
    $J_i^r \leftarrow J^r \oplus i, \quad r_i^r \leftarrow r^r \oplus i$
    $S_i^r \xleftarrow{\$} \{0,1\}^n$
    if $(r_i^r, S_i^r) \in \mathrm{R}$ **then** bad $\leftarrow$ **true**, $\boxed{S_i^r \xleftarrow{\$} \overline{\mathrm{R}}_{r_i^r}}$ **end if**
    if $(r_i^r, J_i^r) \in \mathrm{D}$ **then** bad $\leftarrow$ **true**, $\boxed{S_i^r \leftarrow \widetilde{\pi}(r_i^r, J_i^r)}$ **end if**
    $\widetilde{\pi}(r_i^r, J_i^r) \leftarrow S_i^r, \quad \mathrm{D} \leftarrow \mathrm{D} \uplus \{(r_i^r, J_i^r)\}, \quad \mathrm{R} \leftarrow \mathrm{R} \uplus \{(r_i^r, S_i^r)\}$
  end for
  $S^r \leftarrow \mathrm{MSB}_{s^r}(S_1^r||S_2^r||\cdots||S_{m_r}^r)$
$\overline{P^r} \leftarrow \overline{C^r} \oplus S^r$
$X_2^r \leftarrow H_{\overline{h}_1}(\overline{P^r}, T^r)$
$r_d^r \leftarrow \mathrm{LSB}_n(X_2^r)$
$PP_2^r \xleftarrow{\$} \{0,1\}^n$
if $(r_d^r, PP_2^r) \in \mathrm{D}$ **then** bad $\leftarrow$ **true**, $\boxed{PP_2^r \xleftarrow{\$} \overline{\mathrm{D}}_{r_d^r}}$ **end if**
if $(r_d^r, CC^r) \in \mathrm{R}$ **then** bad $\leftarrow$ **true**, $\boxed{PP_2^r \leftarrow \widetilde{\pi}^{-1}(r_d^r, CC^r)}$ **end if**
$\widetilde{\pi}^{-1}(r_d^r, CC^r) \leftarrow PP_2^r, \quad \mathrm{D} \leftarrow \mathrm{D} \uplus \{(r_d^r, PP_2^r)\}, \quad \mathrm{R} \leftarrow \mathrm{R} \uplus \{(r_d^r, CC^r)\}$
$P_1^r \leftarrow PP_2^r \oplus \mathrm{MSB}_n(X_2^r)$
$P^r \leftarrow P_1^r||\overline{P^r}$
return P^r

For every query r, the game samples $CC^r \xleftarrow{\$} \{0,1\}^n$ independently and computes the hash values under the fixed keys used by the construction. For an encryption transcript, these values are

$$X_1^r = H_{\overline{h}_1}(\overline{P}^r, T^r), \qquad (r^r \parallel J^r) = H_{\overline{h}_1 \oplus \overline{h}_2}(CC^r, T^r), \qquad X_2^r = H_{\overline{h}_2}(\overline{C}^r, T^r).$$

For a decryption transcript, the roles of $\overline{h}_1, \overline{P}^r$ and $\overline{h}_2, \overline{C}^r$ are exchanged as in Figure 4. The values r_e^r, PP_1^r, r_i^r, J_i^r, r_d^r, and PP_2^r are then recovered

using the corresponding relations of the construction, while

$$S^r = \overline{P}^r \oplus \overline{C}^r.$$

If the last remainder block is partial, the unused suffix of $S^r_{m_r}$ is sampled independently and uniformly so that every S^r_i is an n-bit value.

The game then forms the multisets D, R, and T. Here, $\uplus$ denotes multiset union and $\biguplus$ its indexed form; unlike ordinary set union, these operations preserve repeated occurrences, which are relevant to the definition of bad.

$$\begin{aligned}
D &:= \biguplus_{r=1}^{q} \left(\{(r^r_e, PP^r_1)\} \uplus \{(r^r_d, PP^r_2)\} \uplus \biguplus_{i=1}^{m_r} \{(r^r_i, J^r_i)\} \right), \\
R &:= \biguplus_{r=1}^{q} \left(\{(r^r_e, CC^r)\} \uplus \{(r^r_d, CC^r)\} \uplus \biguplus_{i=1}^{m_r} \{(r^r_i, S^r_i)\} \right), \\
\mathsf{T} &:= \biguplus_{r=1}^{q} \{(T^r, CC^r)\}.
\end{aligned}$$

The flag bad is set if some tuple occurs more than once in D, in R, or in T. By deferred decisions, this reordering preserves the joint distribution of the transcript, the reconstructed internal variables, and the event bad. Therefore,

$$\Pr[\mathsf{Game2 \text{ sets } bad}] = \Pr[\mathsf{Game3 \text{ sets } bad}].$$

Bad-event analysis. We now bound the probability that $\mathsf{Game3}$ sets bad. As throughout the proof, redundant forward and inverse queries are omitted. The complete query-response transcript and the independently sampled values $CC^1, \ldots, CC^q$ are independent of the internal hash keys. Conditioning on these values fixes every hash input and every target relation appearing below, while $\overline{h}_1$, $\overline{h}_2$, and $\overline{h}_1 \oplus \overline{h}_2$ retain their original pairwise-independent uniform distributions. No additional condition is imposed on

$$S^r = \overline{P}^r \oplus \overline{C}^r.$$

Possible repetitions of complete streams or individual stream blocks are included directly in the collision analysis of R. We also assume that the n-bit encodings of the counter indices used within each query are distinct. Recall that, counting multiplicities, $|D| = |R| = Q$.

Algorithm 5 Game3

QUERY-RESPONSE

$r^{\mathbf{th}}$ **Encryption Query** $(P_1^r \parallel \overline{P}^r, T^r)$

$qt^r \leftarrow$ **Enc**

$\ell^r \leftarrow |P_1^r \parallel \overline{P}^r|, \quad Y^r \xleftarrow{\$} \{0,1\}^{\ell^r}$

$C^r = C_1^r \parallel \overline{C}^r \leftarrow Y^r$

return C^r

$r^{\mathbf{th}}$ **Decryption Query** $(C_1^r \parallel \overline{C}^r, T^r)$

$qt^r \leftarrow$ **Dec**

$\ell^r \leftarrow |C_1^r \parallel \overline{C}^r|, \quad Y^r \xleftarrow{\$} \{0,1\}^{\ell^r}$

$P^r = P_1^r \parallel \overline{P}^r \leftarrow Y^r$

return P^r

INTERMEDIATE COMPUTATION

$\mathsf{bad} \leftarrow$ **false**, $\quad \mathrm{D} \leftarrow \mathrm{R} \leftarrow \mathsf{T} \leftarrow \emptyset$, where $\mathrm{D}, \mathrm{R}, \mathsf{T}$ are multisets

for $r = 1$ to q **do**

  if $qt^r =$ **Enc then**

    $X_1^r \leftarrow H_{\overline{h}_1}(\overline{P}^r, T^r)$

    $r_e^r \leftarrow \mathrm{LSB}_n(X_1^r), \quad PP_1^r \leftarrow P_1^r \oplus \mathrm{MSB}_n(X_1^r)$

    $CC^r \xleftarrow{\$} \{0,1\}^n$

    $\mathrm{D} \leftarrow \mathrm{D} \uplus \{(r_e^r, PP_1^r)\}, \quad \mathrm{R} \leftarrow \mathrm{R} \uplus \{(r_e^r, CC^r)\}$

    $(r^r \parallel J^r) \leftarrow H_{\overline{h}_1 \oplus \overline{h}_2}(CC^r, T^r)$

    $\mathsf{T} \leftarrow \mathsf{T} \uplus \{(T^r, CC^r)\}$

    $s^r \leftarrow |\overline{P}^r|, \quad m_r \leftarrow \lceil s^r/n \rceil, \quad S^r \leftarrow \overline{C}^r \oplus \overline{P}^r$

    $U^r \xleftarrow{\$} \{0,1\}^{m_r n - s^r}, \quad S_1^r \parallel \cdots \parallel S_{m_r}^r \leftarrow S^r \parallel U^r$

    for $i = 1$ to m_r **do**

      $J_i^r \leftarrow J^r \oplus i, \quad r_i^r \leftarrow r^r \oplus i$

      $\mathrm{D} \leftarrow \mathrm{D} \uplus \{(r_i^r, J_i^r)\}, \quad \mathrm{R} \leftarrow \mathrm{R} \uplus \{(r_i^r, S_i^r)\}$

    end for

    $X_2^r \leftarrow H_{\overline{h}_2}(\overline{C}^r, T^r)$

    $r_d^r \leftarrow \mathrm{LSB}_n(X_2^r), \quad PP_2^r \leftarrow C_1^r \oplus \mathrm{MSB}_n(X_2^r)$

    $\mathrm{D} \leftarrow \mathrm{D} \uplus \{(r_d^r, PP_2^r)\}, \quad \mathrm{R} \leftarrow \mathrm{R} \uplus \{(r_d^r, CC^r)\}$

  else if $qt^r =$ **Dec then**

    $X_1^r \leftarrow H_{\overline{h}_2}(\overline{C}^r, T^r)$

    $r_e^r \leftarrow \mathrm{LSB}_n(X_1^r), \quad PP_1^r \leftarrow C_1^r \oplus \mathrm{MSB}_n(X_1^r)$

    $CC^r \xleftarrow{\$} \{0,1\}^n$

    $\mathrm{D} \leftarrow \mathrm{D} \uplus \{(r_e^r, PP_1^r)\}, \quad \mathrm{R} \leftarrow \mathrm{R} \uplus \{(r_e^r, CC^r)\}$

    $(r^r \parallel J^r) \leftarrow H_{\overline{h}_1 \oplus \overline{h}_2}(CC^r, T^r)$

    $\mathsf{T} \leftarrow \mathsf{T} \uplus \{(T^r, CC^r)\}$

    $s^r \leftarrow |\overline{C}^r|, \quad m_r \leftarrow \lceil s^r/n \rceil, \quad S^r \leftarrow \overline{P}^r \oplus \overline{C}^r$

    $U^r \xleftarrow{\$} \{0,1\}^{m_r n - s^r}, \quad S_1^r \parallel \cdots \parallel S_{m_r}^r \leftarrow S^r \parallel U^r$

    for $i = 1$ to m_r **do**

      $J_i^r \leftarrow J^r \oplus i, \quad r_i^r \leftarrow r^r \oplus i$

      $\mathrm{D} \leftarrow \mathrm{D} \uplus \{(r_i^r, J_i^r)\}, \quad \mathrm{R} \leftarrow \mathrm{R} \uplus \{(r_i^r, S_i^r)\}$

    end for

    $X_2^r \leftarrow H_{\overline{h}_1}(\overline{P}^r, T^r)$

    $r_d^r \leftarrow \mathrm{LSB}_n(X_2^r), \quad PP_2^r \leftarrow P_1^r \oplus \mathrm{MSB}_n(X_2^r)$

    $\mathrm{D} \leftarrow \mathrm{D} \uplus \{(r_d^r, PP_2^r)\}, \quad \mathrm{R} \leftarrow \mathrm{R} \uplus \{(r_d^r, CC^r)\}$

  end if

end for

BAD ANALYSIS

$\mathsf{bad} \leftarrow$ **true if** some tuple occurs more than once in D, R, or T

We first record two direct consequences of 2. For every $x \in \mathcal{D}$ and every $c \in \{0,1\}^n$,

$$\Pr_{h \xleftarrow{\$} \mathcal{K}_H} \left[\mathrm{LSB}_n(H_h(x)) = c\right] \leq 2^n \varepsilon,$$

and, for every pair of distinct inputs $x, x' \in \mathcal{D}$,

$$\Pr_{h \xleftarrow{\$} \mathcal{K}_H} \left[\mathrm{LSB}_n\big(H_h(x) \oplus H_h(x')\big) = c\right] \leq 2^n \varepsilon.$$

The same bounds hold with MSB_n in place of LSB_n. These inequalities follow by summing the relevant full-output bound over the 2^n possible values of the unconstrained half. Moreover, if h and h' are independent uniform hash keys, conditioning on one evaluation and applying uniformity to the other gives the same $2^n\varepsilon$ bound for any prescribed relation between corresponding n-bit halves of $H_h(x)$ and $H_{h'}(x')$.

Collisions in D. Consider first two endpoint tuples, each of which is of one of the forms

$$(r_e, PP_1) \qquad \text{or} \qquad (r_d, PP_2).$$

Equality of the two tuples fixes one complete $2n$-bit XOR difference of the corresponding outer-hash outputs. If the evaluations use the same hash key on distinct inputs, AXU bounds the event by ε. If they use pairwise-independent hash keys, conditioning on one output fixes a target for the other, and uniformity again bounds the event by ε.

It remains to consider evaluations under the same key on an identical input. If both values arise from initial outer-hash evaluations of queries of the same type, tuple equality either makes the external queries redundant or is impossible because their first blocks differ. In every other such case, at least one complete outer value is response-derived; equality of the outer-hash input together with equality of the masked first coordinate then requires a uniform response of length at least $2n$ bits to match a fixed string. The probability of this event is at most 2^{-2n}.

An endpoint counter collision fixes a complete relation between an outer-hash output and a central-hash output. The corresponding keys are pairwise independent, and uniformity bounds the event by ε. For two counter tuples (r_i, J_i), distinct indices within one query cannot collide by injectivity of the counter encoding. Across queries, distinct central-hash inputs are handled by AXU, whereas identical central-hash inputs are charged to $\mathrm{Coll}(\mathsf{T})$. Consequently,

$$\Pr[\mathrm{Coll}(D) \wedge \neg\, \mathrm{Coll}(\mathsf{T})] \leq \binom{Q}{2}\big(\varepsilon + 2^{-2n}\big).$$

Collisions in R. We first isolate the two endpoint tuples contributed by the same external query:

$$(r_e^r, CC^r), \qquad (r_d^r, CC^r).$$

Their second coordinates are already identical. The first coordinates are obtained under the independent keys $\overline{h}_1$ and $\overline{h}_2$; therefore, by the half-output consequence above,

$$\Pr[r_e^r = r_d^r] \le 2^n \varepsilon.$$

The q same-query pairs contribute at most $q2^n\varepsilon$.

Consider any remaining endpoint–endpoint, endpoint–counter, or counter–counter pair that is not one of the identical-input cases treated separately below. Let $\mathcal{E}_2$ denote the equality required between the second coordinates. Over the random transcript and the independently sampled values $CC^1, \ldots, CC^q$, one may expose all values except one fresh n-bit response block or one fresh CC value; hence

$$\Pr[\mathcal{E}_2] \le 2^{-n}.$$

For a partial final stream block, the same bound holds because its unused suffix is completed by independent uniform bits in Game3.

Conditioned on the transcript, on $CC^1, \ldots, CC^q$, and on $\mathcal{E}_2$, equality of the first coordinates fixes a prescribed relation between two n-bit halves of hash outputs. Same-key evaluations on distinct inputs are bounded by the half-output consequence of AXU, and evaluations under pairwise-independent keys are bounded by uniformity. In either case, the conditional probability is at most $2^n\varepsilon$. Thus, the joint probability of the two coordinate equalities is at most

$$2^{-n} \cdot 2^n \varepsilon = \varepsilon.$$

Equal central-hash inputs across distinct queries are charged to $\mathrm{Coll}(\mathsf{T})$, while counter tuples with distinct indices within one query cannot collide.

If two initial outer-hash evaluations of the same query type repeat the same outer-hash input, their endpoint indices repeat deterministically. A range collision then requires equality of the corresponding CC values. Because the repeated outer-hash input includes the same tweak, this event is also a collision in T and is charged there. Every other endpoint pair evaluated under the same key on an identical input contains at least one response-derived remainder. A range collision then requires both an n-bit remainder match and an independent CC collision, and is therefore bounded

by 2^{-2n}. Consequently,

$$\Pr[\mathrm{Coll}(R) \wedge \neg\, \mathrm{Coll}(\mathsf{T})] \le \binom{Q}{2}\left(\varepsilon + 2^{-2n}\right) + q2^n\varepsilon.$$

Collisions in T. The multiset T contains the tuples (T^r, CC^r). Only pairs sharing the same tweak can collide. The number of such pairs is at most $q(\mu-1)/2$, and the corresponding independent CC values collide with probability 2^{-n}. Therefore,

$$\Pr[\mathrm{Coll}(\mathsf{T})] \le \frac{q(\mu-1)}{2^{n+1}}.$$

Using a union bound and charging identical central-hash inputs to $\mathrm{Coll}(\mathsf{T})$ gives

$$\Pr[\text{Game3 sets bad}] \le 2\binom{Q}{2}\left(\varepsilon + 2^{-2n}\right) + q2^n\varepsilon + \frac{q(\mu-1)}{2^{n+1}}.$$

For consistency with the slightly looser statement of Theorem 1, we may further replace the last term by $q(\mu-1)/2^n$.

Combining this estimate with the key-derivation hop, the SPRP-to-SPRF switching term, and the R3 bound established in $\mathsf{Game1}$, we obtain

$$\begin{aligned}
\mathrm{Adv}^{\mathrm{SPRP}}_{\mathrm{HCTR}^{++}[E,H]}(\mathcal{A}) &\le \frac{Q+p+15}{2^n} + Q(Q-1+2p)\varepsilon \\
&\quad + 2\binom{Q}{2}\left(\varepsilon + 2^{-2n}\right) + q2^n\varepsilon \\
&\quad + \frac{q(\mu-1)}{2^n} + \frac{q^2}{2^{2n+1}} \\
&= 2Q(Q-1+p)\varepsilon + q2^n\varepsilon \\
&\quad + \frac{Q+p+15+q(\mu-1)}{2^n} \\
&\quad + \frac{2Q(Q-1)+q^2}{2^{2n+1}},
\end{aligned}$$

which is the bound stated in Theorem 1.

□

5 Conclusion

The exponential growth of data transmission rates and the ubiquity of high-throughput encryption environments have rendered the traditional birthday bound limit of $O(2^{n/2})$ an increasingly relevant security limitation rather than merely a theoretical abstraction. As detailed in this work, established standards such as GCM and CBC are increasingly ill-equipped to secure terabit-scale data streams without risking distinguishing attacks or plaintext compromise. In response to NIST's call for next-generation Accordion modes that ensure robust security margins, this article has presented HCTR^{++}, a novel variant of the HCTR2 algorithm designed to achieve a beyond-birthday-bound security profile under the stated assumptions.

HCTR^{++} successfully bridges the gap between the operational efficiency of HCTR2 and the rigorous security demands of modern infrastructure by synthesizing the Fresh Re-keying methodology with a Wide Block Encryption structure. By integrating the R3 re-keying scheme, replacing the standard encryption primitives with dynamic re-keying operations, and the n-bit universal hash function with a $2n$-bit ε-uniform-AXU keyed hash family, the proposed construction reduces and bounds the internal collision probabilities that dominate birthday-bound analyses that constrain standard modes. Furthermore, the structural innovations introduced, specifically the introduction of a decryption operation on the initial block and the symmetric derivation of the internal state J, ensure that the ciphertext depends on the entire hash output while simultaneously harmonizing the encryption and decryption logic for implementation ease.

Ultimately, HCTR^{++} demonstrates that it is possible to retain the parallelizability and versatility of the HCTR2 design while achieving a full-block-scale, beyond-birthday-bound security profile under the stated hash, message-length, and tweak-repetition conditions. As the cryptographic community and NIST move toward standardizing the Accordion Mode, HCTR^{++} stands as a formidable candidate. HCTR^{++} provides a candidate direction for constructing Accordion oriented Wide Block Encryption from standard block cipher primitives. Under the stated assumptions, the construction achieves the SPRP security bound established in this work while retaining the structural advantages of HCTR2.

Acknowledgement

We would like to thank Mustafa Khairallah for identifying an issue in an earlier version of the construction, and Michele Montedoro for identifying an issue in the original formulation and proof of Theorem 1. Their comments helped improve the correctness and presentation of this work.The authors acknowledge the use of AI Tools in drafting and refining this paper.

References

[1] Michel Abdalla and Mihir Bellare. Increasing the lifetime of a key: A comparative analysis of the security of re-keying techniques. In Tatsuaki Okamoto, editor, *Advances in Cryptology — ASIACRYPT 2000*, pages 546–559, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[2] Zhenzhen Bao, Seongha Hwang, Akiko Inoue, Byeonghak Lee, Jooyoung Lee, and Kazuhiko Minematsu. Xocb: Beyond-birthday-bound secure authenticated encryption mode with rate-one computation. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, pages 532–561, Cham, 2023. Springer Nature Switzerland.

[3] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 394–403, 1997.

[4] M. Bellare and R. Impagliazzo. A tool for obtaining tighter security analyses of pseudorandom function based constructions, with applications to PRP to PRF conversion. Cryptology ePrint Archive, Paper 1999/024, 1999.

[5] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences*, 61(3):362–399, 2000.

[6] Karthikeyan Bhargavan and Gaëtan Leurent. On the practical (in-)security of 64-bit block ciphers: Collision attacks on http over tls and openvpn. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 456–467, New York, NY, USA, 2016. Association for Computing Machinery.

[7] Ritam Bhaumik, Eik List, and Mridul Nandi. Zcz–achieving n-bit sprp security with a minimal number of tweakable-block-cipher calls. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 336–366. Springer, 2018.

[8] Yu Long Chen, Michael Davidson, Morris Dworkin, John Kelsey, Yu Sasaki, Meltem Sönmez Turan, Donghoon Chang, Nicky Mouha, and Alyssa Thompson. Requirements for cryptographic accordions. 2025.

[9] Yu Long Chen, Yukihito Hiraga, Nicky Mouha, Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Beyond-birthday-bound security with HCTR2: Cascaded construction and tweak-based key derivation. Cryptology ePrint Archive, Paper 2026/085, 2026.

[10] Woohyuk Chung, Seongha Hwang, Seongkwang Kim, Byeonghak Lee, and Jooyoung Lee. Making gcm great again: Toward full security and longer nonces. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025*, pages 33–61, Cham, 2025. Springer Nature Switzerland.

[11] Paul Crowley, Nathan Huckleberry, and Eric Biggers. Length-preserving encryption with HCTR2. Cryptology ePrint Archive, Paper 2021/1441, 2021.

[12] Nilanjan Datta, Avijit Dutta, Shibam Ghosh, Eik List, and Hrithik Nandi. HCTR+: An optimally secure TBC-based accordion mode. Cryptology ePrint Archive, Paper 2024/2053, 2024.

[13] Christoph Dobraunig, François Koeune, Stefan Mangard, Florian Mendel, and François-Xavier Standaert. Towards fresh and hybrid re-keying schemes with beyond birthday security. In Naofumi Homma and Marcel Medwed, editors, *Smart Card Research and Advanced Applications*, pages 225–241, Cham, 2016. Springer International Publishing.

[14] Christoph Dobraunig, Krystian Matusiewicz, Bart Mennink, and Alexander Tereschenko. Efficient instances of docked double decker with AES, and application to authenticated encryption. Cryptology ePrint Archive, Paper 2024/084, 2024.

[15] Avijit Dutta and Mridul Nandi. Tweakable HCTR: A BBB secure tweakable enciphering scheme. Cryptology ePrint Archive, Paper 2019/1324, 2019.

[16] Morris Dworkin. Recommendation for block cipher modes of operation: Methods and techniques, nist sp 800-38a, 2001. `https://https://doi.org/10.6028/NIST.SP.800-38A`.

[17] Morris Dworkin. Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac, nist sp 800-38d, 2007. `https://doi.org/10.6028/NIST.SP.800-38D`.

[18] Shai Halevi and Phillip Rogaway. A tweakable enciphering mode. Cryptology ePrint Archive, Paper 2003/148, 2003.

[19] Tetsu Iwata. New blockcipher modes of operation with beyond the birthday bound security. Cryptology ePrint Archive, Paper 2006/188, 2006.

[20] Ashwin Jha, Eik List, Kazuhiko Minematsu, Sweta Mishra, and Mridul Nandi. Xhx – a framework for optimally secure tweakable block ciphers from classical block ciphers and universal hashing. In Tanja Lange and Orr Dunkelman, editors, *Progress in Cryptology – LATINCRYPT 2017*, pages 207–227, Cham, 2019. Springer International Publishing.

[21] Antoine Joux. Authentication failures in nist version of gcm. *NIST Comment*, 3, 2006.

[22] Ted Krovetz and Phillip Rogaway. The software performance of authenticated-encryption modes. In Antoine Joux, editor, *Fast Software Encryption*, pages 306–327, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[23] Byeonghak Lee. A BBB-secure accordion mode from HCTR. Presentation at the NIST Workshop on the Requirements for an Accordion Cipher Mode, June 2024. Accessed: 2025-05-22.

[24] Moses Liskov, Ronald L Rivest, and David Wagner. Tweakable block ciphers. In *Advances in Cryptology—CRYPTO 2002: 22nd Annual International Cryptology Conference Santa Barbara, California, USA, August 18–22, 2002 Proceedings 22*, pages 31–46. Springer, 2002.

[25] Stefan Lucks. The sum of prps is a secure prf. In *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 470–484. Springer, 2000.

[26] Bart Mennink. Optimally secure tweakable blockciphers. In *International Workshop on Fast Software Encryption*, pages 428–448. Springer, 2015.

[27] Bart Mennink. Beyond birthday bound secure fresh rekeying: Application to authenticated encryption. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, pages 630–661, Cham, 2020. Springer International Publishing.

[28] Jacques Patarin. A proof of security in o (2 n) for the xor of two random permutations. In *International Conference on Information Theoretic Security*, pages 232–248. Springer, 2008.

[29] Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes ocb and pmac. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 16–31. Springer, 2004.

[30] 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*, pages 405–423, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[31] Lei Wang, Jian Guo, Guoyan Zhang, Jingyuan Zhao, and Dawu Gu. How to build fully secure tweakable blockciphers from classical blockciphers. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016*, pages 455–483, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.