Beyond-Birthday-Bound Security with HCTR2: Cascaded Construction and Tweak-based Key Derivation*

Yu Long Chen[1], Yukihito Hiraga[2], Nicky Mouha[3], Yusuke Naito[4], Yu Sasaki[5,6], and Takeshi Sugawara[2]

[1] KU Leuven, Leuven, Belgium, yulong.chen@esat.kuleuven.be
[2] The University of Electro-Communications, Tokyo, Japan, yukihito.hiraga@uec.ac.jp, sugawara@uec.ac.jp
[3] Independent Researcher, nicky@mouha.be
[4] Mitsubishi Electric Corporation, Kanagawa, Japan, Naito.Yusuke@ce.MitsubishiElectric.co.jp
[5] NTT Social Informatics Laboratories, Tokyo, Japan, yusk.sasaki@ntt.com
[6] Associate of National Institute of Standards and Technology, Gaithersburg, USA

Abstract. The block cipher (BC) mode for realizing a variable-input-length strong tweakable pseudorandom permutation (VIL-STPRP), also known as the accordion mode, is a rapidly growing research field driven by NIST's standardization project, which considers AES as a primitive. Widely used VIL-STPRP modes, such as HCTR2, have birthday-bound security and provide only 64-bit security with AES. To provide higher security, NIST is considering two directions: to develop new modes with beyond-birthday-bound (BBB) security and to use Rijndael-256-256 with HCTR2. This paper pursues the first direction while maintaining compatibility with HCTR2. In particular, we provide two solutions to achieve BBB security for two different approaches: (i) general cases without any conditions on the tweak and (ii) under the condition that the same tweak is not repeated too often as adopted in *bbb-ddd-AES* recently presented at Eurocrypt 2025. For the first approach, we propose a new mode, CHCTR2, that iterates HCTR2 with two independent keys, which achieves $2n/3$-bit security in the multi-user (mu) setting and satisfies NIST's requirements. For the second approach, we prove mu security of HCTR2, which allows us to apply the tweak-based key derivation (TwKD) to HCTR2 in a provable manner. When the number of BC calls processed by a single tweak is upper-bounded by $2^{n/3}$, HCTR2-TwKD achieves $2n/3$-bit mu security. By benchmarking optimized software implementations, we show that CHCTR2 with AES-256 outperforms HCTR2 with Rijndael-256-256 on all twelve processor models examined. Similarly, HCTR2-TwKD outperforms *bbb-ddd-AES* in general cases, and it is even comparable to *bbb-ddd-AES* rigorously optimized for tweak-repeating use cases using precomputation.

Keywords: Mode · Accordion · Beyond-Birthday-Bound security · Multi-user security · HCTR2 · Cascaded HCTR2 · Backward Compatibility

1 Introduction

A block cipher (BC) converts a fixed-length plaintext into a fixed-length ciphertext. BCs are used inside modes of operation to provide various functionalities.

In 2023, the National Institute of Standards and Technology (NIST) announced the development of a new BC mode: an "accordion mode" [33]. It corresponds to a variable-input-length strong tweakable pseudorandom permutation (VIL-STPRP) or tweakable wide-block encryption [32]. A VIL-STPRP provides an STPRP for a plaintext of any length, e.g., changing a single bit of the plaintext (resp. ciphertext) or the tweak randomizes the ciphertext (resp. plaintext).

In a VIL-STPRP, the encryption of a plaintext results in a ciphertext of the same length. This property is crucial in many applications. For example, in storage encryption, ciphertext expansion is not allowed because the data is divided into equal-sized data units, and the ciphertext is stored instead of the plaintext. To avoid ECB-style leakage, the sector

*This paper is the full version of [7] published in Asiacrypt 2025 by the same authors.

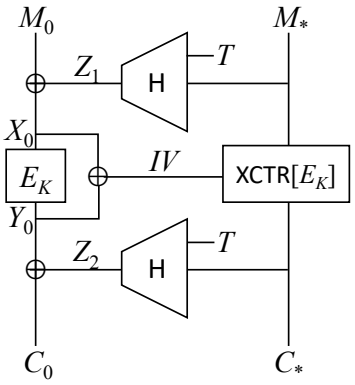


Fig. 1. HCTR2. The hash key is defined as $E_K(0)$ and the mask value of XCTR $L = E_K(1)$ is included in XCTR.

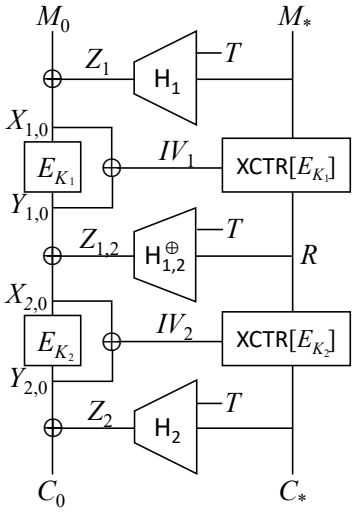


Fig. 2. CHCTR2. The second-round hash function is $Z_{1,2} = \mathsf{H}^{\oplus}_{1,2}(T, R) = \mathsf{H}_1(T, R) \oplus \mathsf{H}_2(T, R)$.

number is used as a tweak. Hence, the mode needs to be tweakable. Moreover, if a tweak of any length is acceptable, the encode-then-encipher paradigm [3] can be used: bits of zeros are padded to the plaintext and are checked during decryption. This provides robust authentication encryption (AE) [22], which mitigates nonce misuse in nonce-based AE.

Several BC-based VIL-STPRP modes have been designed, such as CMC [20], EME [21], HCH [4], TET [19], and HCTR2 [8]. Among them, HCTR2 has a wide adoption, e.g., the Linux kernel [36] and file encryption in Android [2]. HCTR2 is based on HCTR [38] and is proved to be secure up to the birthday bound. It adopts the hash-encrypt-hash structure depicted in Fig. 1: a plaintext is split into n-bit M_0 and remaining M_*, where n is the block size of a BC E_K with a k-bit key K. First, a polynomial hash function H, i.e., POLYVAL, a variant of GHASH, processes M_* and a tweak T, and the output is XORed to M_0. Next, the left n-bit state is updated using E_K, an IV is extracted for encryption, and the XCTR mode (a variant of the CTR mode) encrypts M_*. Finally, H processes C_* and T, and its output is XORed into the left n bits to produce C_0. The cost per block is roughly 1 BC call and 2 n-bit field multiplications (FMULs).

In NIST's standardization, NIST requires using AES [31] as a primitive [33]. In the NIST Workshop [33], a concern was raised that existing VIL-STPRP modes, such as HCTR2, only provide birthday security and only provide 64-bit security when combined with AES. NIST then proposed two directions [6]; one is to develop VIL-STPRP modes with beyond-birthday-bound (BBB) security, and the other is to standardize a BC with a 256-bit block size such as Rijndael-256-256 [28,37] and use it in a birthday-secure mode. A concern for the second direction is that Rijndael-256-256 has not been standardized yet, and the transition to a new BC takes a long time. Consequently, this paper pursues the first direction of designing a new BBB mode, which can take full advantage of established security evaluation and implementation techniques of AES. **

In general, there are two approaches to achieve BBB. The first approach typically enlarges the state size so that the number of queries required to cause undesirable events,

**The initial preliminary draft of NIST SP 800-197A [34], released after the initial submission of this paper, proposes to develop BBBAcc, a beyond-birthday-bound accordion based on AES and HCTR2 technique.

such as collisions, can increase. The second approach, originally proposed for AES-GCM-SIV, derives keys based on the tweak or nonce, assuming that each tweak or nonce is not repeated many times [17].

The first approach has the advantage of achieving BBB unconditionally for query values. NIST prefers this property, noted as *"NIST expects the BBB security to hold for all queries, including multiple queries with the same tweak"* [6]. Consequently, several designs presented at the NIST Workshop in 2024 [33], including ACCOR-L [13] and DbHCTR [26], followed this approach. However, ACCOR-L has the limitation that only AES-256 can be used as a primitive, and the construction relies on the assumption of the related-key security of AES-256. The presentation of DbHCTR was limited to the introductory idea, and a complete specification, a concrete bound, and an accompanying proof were not presented. Currently, there are no BBB-secure VIL-STPRP modes with proofs that meet NIST's requirements.

A large computational overhead is the major drawback of the BBB-secure modes, and the second approach can provide better performance, even comparable to HCTR2. However, the assumption on the tweak/nonce repetition is valid for particular use cases only. The assumption could be valid for nonce-based AEAD where the tweak includes a nonce, while it may be invalid for other applications, particularly storage encryption where the same disk unit is updated without a nonce. At the NIST workshop [33], beyond-birthday-bound docked-double-decker AES (*bbb-ddd-AES*) was presented to achieve $2n/3$-bit security when *"... assuming it is not used with the same tweak too often ..."*, which has recently been presented at Eurocrypt 2025 [10, 11]. *bbb-ddd-AES* requires 1 AES call to derive tweak-dependent mask values for encryption in addition to almost the same cost as HCTR2. Hence, without any precomputation, the cost per block is 2 BC calls and 2 FMULs. The designers [11] suggest the implementation technique to precompute all the mask values, which works in the limited use cases where the tweak value does not change. This reduces the cost to 1 BC call and 2 FMULs, the same as that of HCTR2. However, precomputation is unavailable for some applications, such as disk encryption that uses a sector number as a tweak.

Multi-user (mu) security is considered important by NIST [6]. In the classic single-user (su) security, an attacker could only query one user with a fixed key, but in the mu security, an attacker can query multiple users with independent keys, and the goal is to break any of them. This reflects real-world usage where multiple users with independent keys exist for a single service. Meanwhile, to the best of our knowledge, there are no existing VIL-STPRP modes with proved mu security except those obtained by the classical conversion from su bounds. Dedicated mu bounds are important because *"NIST prefers to standardize an accordion with better multi-user security than is provided by generic bounds"* [6].

Note that a VIL-STPRP with more than 64-bit security could be achieved by using primitives other than AES, e.g., modes with tweakable BCs [9, 12] or permutation-based modes [18] with a large permutation. Such an approach is outside the scope of this paper, because it is not the direction of NIST.

1.1 Our Contributions

First, with a novel idea of cascading HCTR2, we propose a new VIL-STPRP mode CHCTR2 achieving BBB security in the mu setting without imposing additional conditions on the tweak, which meets NIST's requirements at a high level. Second, we prove mu security of HCTR2, which further allows us to apply the tweak-based key derivation (TwKD) to HCTR2 in a provable manner. When the same tweak is not iterated many times, as recently discussed with *bbb-ddd-AES*, HCTR2-TwKD achieves BBB security. More details are explained in the following.

Table 1. Comparison of VIL-STPRP modes. #Primitive refers to the number of BC and 128-bit field multiplication (FMUL) calls to process each 128-bit block.

Mode	Model	Security Level		#Keys		#Primitives		Ref.
		General	Conditional	Hash	BC	BC	FMUL	
HCTR2	su	$n/2$	$n/2$	0	1	1	2	[8]
HCTR2†	mu	$n/2$	$n/2$	0	1	1	2	Ours
CHCTR2*	mu	$2n/3$	$2n/3$	0	2	2	4	Ours
bbb-ddd-AES w/o PC	su	$n/2$	$2n/3$	1	1	2	2	[11]
bbb-ddd-AES w/ PC‡	su	$n/2$	$2n/3$	1	1	1	2	[11]
HCTR2-TwKD	mu	$n/2$	$2n/3$	0	1	1	2	Ours

†The security level holds when $\sigma_{\mathsf{u}} \approx \sigma$. If $\sigma_{\mathsf{u}} \ll \sigma$, HCTR2 has BBB security.
‡Precomputation (PC) available under certain prerequisites reduces the number of BCs by 1 in *bbb-ddd-AES*. Section 10 provides more details.
*The security level holds when $\ell\sigma_{\mathsf{u}}^2 \approx \sigma^2$. If $\ell\sigma_{\mathsf{u}}^2 \ll \sigma^2$, CHCTR2 has beyond $2n/3$-bit security.

CHCTR2: A BBB-Secure VIL-STPRP Mode. We consider a cascaded construction that sequentially computes HCTR2 twice with two independent keys, as shown in Fig. 2. CHCTR2 can call HCTR2 as a black box, and thus maintains backward compatibility with existing implementations of HCTR2. With such black-box calls, the structure is hash-encrypt-hash-hash-encrypt-hash, which costs two HCTR2 calls i.e., two BC and four FMUL calls for each block. Further optimization is possible in dedicated CHCTR2 implementations by evaluating two consecutive POLYVAL calls, i.e., $\mathsf{H}^{\oplus}_{1,2}$ in Fig. 2, using Winograd's inner product algorithm [39], which reduces the per-block cost to two BC and $3.5{+}1/b$ FMUL calls where b denotes the number of loop unrolling. Regarding security, we prove the following mu-VIL-STPRP security bound of CHCTR2 under the standard mu-SPRP assumption: $O\left(\frac{\sigma}{2^n} + \frac{\ell\sigma_{\mathsf{u}}^2\sigma}{2^{2n}}\right)$, where σ is the number of n-bit data blocks (tweak and plaintext blocks) in all queries, σ_{u} is the maximum number of n-bit data blocks per user, and ℓ is the maximum number of n-bit data blocks per query. Assuming $\ell\sigma_{\mathsf{u}}^2 \le \sigma^2$, CHCTR2 achieves $2n/3$-bit mu-VIL-STPRP security. Furthermore, if $\ell\sigma_{\mathsf{u}}^2 \ll \sigma^2$, CHCTR2 has the beyond $2n/3$-bit mu-VIL-STPRP security.

mu-Security Proof of HCTR2. We next derive the first mu-VIL-STPRP-security bound of HCTR2 in the mu-SPRP assumption. Our bound is $O\left(\frac{\sigma_{\mathsf{u}}\sigma}{2^n}\right)$, showing that HCTR2 achieves $(n - \log_2 \sigma_{\mathsf{u}})$-bit security. If $\sigma \approx \sigma_{\mathsf{u}}$, i.e., when the adversary sends the most queries to a particular user, which is rather closer to the su setting, then HCTR2 achieves birthday-bound security, and if $\sigma_{\mathsf{u}} \ll 2^{n/2}$, it achieves BBB security. Note that XCTR of HCTR2 adds not only a counter but also a secret mask $L = E_K(1)$ to IV, while our proof targets a generalized construction such that the counter addition can be instantiated both with L and without L. Our achieved bound suggests that L is unnecessary and can be removed from the specification without degrading mu-VIL-STPRP security.

HCTR2-TwKD: HCTR2 with Tweak-Based Key Derivation. The mu-security proof of HCTR2 allows us to apply the tweak-based key derivation (TwKD) to HCTR2 in a provable manner, in which HCTR2's key is derived from a PRF that takes the tweak as an input. Similar to CHCTR2, this method calls HCTR2 as a black box, and thus maintains backward compatibility with existing implementations of HCTR2. Let σ_{tw} be the maximum

number of data blocks under the same tweak and user. The resulting mu bound is $\frac{\sigma_{\mathsf{tw}}\sigma}{2^n}$. Under the condition $\sigma_{\mathsf{tw}} \leq 2^{\frac{n}{3}}$, HCTR2-TwKD achieves $2n/3$-bit mu security.

Compared to *bbb-ddd-AES* that achieves $2n/3$-bit su security under the condition that the number of tweak repetitions is $2^{n/3}$ or less, HCTR2-TwKD achieves $2n/3$-bit mu security under the condition that the number of BC invocations for a single tweak is upper-bounded by $2^{n/3}$, therefore the security of HCTR2-TwKD is comparable in quantity and better in quality. Regarding performance, the overhead of HCTR2-TwKD over HCTR2 is negligible, i.e., the cost per block stays 1 BC and 2 FMULs, which outperforms *bbb-ddd-AES* with 2 BCs and 2 FMULs in the general case. Although *bbb-ddd-AES* can reduce the cost to 1 BC and 2 FMULs by precomputing tweak-dependent mask values in advance, this rigorous optimization is possible for limited use cases that repeat the same tweak; HCTR2-TwKD achieves the same performance without such restrictions and the cost of precomputation.

Software Performance Evaluation. We compare the software performance of the proposed methods with the state-of-the-art modes by instantiating them using BCs with a 256-bit key, as required by NIST [6]. Each implementation is optimized with dedicated CPU instructions [14], and their cycle-per-byte (cpb) performances are evaluated on 12 processors from four different vendors, including x86 and ARM architectures, and in local and cloud environments [1]. The first category evaluates CHCTR2 instantiated with AES-256 that achieves BBB security without any restriction on the tweak. The counterpart is HCTR2 instantiated with Rijndael-256-256 and a 256-bit universal hash function that concatenates two POLYVAL instances. While Rijndael-256-256 benefits from the AES instructions, the overhead of additional instructions needed for Rijndael-256-256 [15] is significant. As a result, the baseline CHCTR2 implementation, which simply invokes HCTR2 twice, outperforms HCTR2 with Rijndael-256-256 on 11 of the 12 target processors, and the optimized CHCTR2 implementation using Winograd's algorithm outperforms it on all of them.

The second comparison compares HCTR2-TwKD and *bbb-ddd-AES*, both instantiated with AES-256, that achieve the same BBB security when the same tweak is not repeated too often. We implement two versions of *bbb-ddd-AES* with and without precomputation. As a result, HCTR2-TwKD completely outperforms the other without precomputation because *bbb-ddd-AES* needs one additional AES-256 for each 128-bit block. Moreover, HCTR2-TwKD achieves a similar performance compared to *bbb-ddd-AES* with precomputation. In summary, the relative performances of HCTR2-TwKD and *bbb-ddd-AES* follow those suggested by the numbers of primitive calls in Table 1, and HCTR2-TwKD achieves better performance in general cases.

1.2 Paper Organization

Section 2 gives basic notation. Section 3 defines the specifications of HCTR2, CHCTR2, and HCTR2-TwKD. Section 4 summarizes security models. Section 5 shows the security bound of CHCTR2, which is proved in Section 6. Section 8 proves the mu security of HCTR2. Section 10 shows the implementation results.

2 Basic Notations

Let ε be the empty string and $\emptyset$ the empty set. For integers $i \leq j$, let $[i,j] \coloneqq \{i, i+1, \ldots, j\}$, and $[j] \coloneqq [1,j]$. If $j < i$ then $[i,j] \coloneqq \emptyset$. Let $\{0,1\}^i$ be the set of all i-bit strings and $\{0,1\}^0 \coloneqq \{\varepsilon\}$. Let $\{0,1\}^{\leq n} \coloneqq \cup_{i\in[0,n]}\{0,1\}^i$ and $\{0,1\}^* \coloneqq \cup_{i\geq 0}\{0,1\}^i$. Let $\{0,1\}^{n*} \coloneqq$

Algorithm 1 HCTR2

Encryption $\mathsf{HCTR2}[E_K, \mathsf{H}](T, M)$

1: $H \leftarrow E_K(\mathtt{bin}_n(0))$; $(M_0, M_*) \xleftarrow{n,|M|-n} M$
2: $Z_1 \leftarrow \mathsf{H}_H(T, M_*)$; $X_0 \leftarrow Z_1 \oplus M_0$
3: $Y_0 \leftarrow E_K(X_0)$; $IV \leftarrow X_0 \oplus Y_0$; $C_* \leftarrow \mathsf{XCTR}[E_K](IV, M_*)$;
4: $Z_2 \leftarrow \mathsf{H}_H(T, C_*)$; $C_0 \leftarrow Z_2 \oplus Y_0$
5: $C \leftarrow C_0 \| C_*$; **return** C

Decryption $\mathsf{HCTR2}^{-1}[E_K^{\pm}, \mathsf{H}](T, C)$

1: $H \leftarrow E_K(\mathtt{bin}_n(0))$; $(C_0, C_*) \xleftarrow{n,|C|-n} C$
2: $Z_2 \leftarrow \mathsf{H}_H(T, C_*)$; $Y_0 \leftarrow Z_2 \oplus C_0$
3: $X_0 \leftarrow E_K^{-1}(Y_0)$; $IV \leftarrow X_0 \oplus Y_0$; $M_* \leftarrow \mathsf{XCTR}[E_K](IV, C_*)$;
4: $Z_1 \leftarrow \mathsf{H}_H(T, M_*)$; $M_0 \leftarrow Z_1 \oplus X_0$
5: $M \leftarrow M_0 \| M_*$; **return** M

Subroutine $\mathsf{XCTR}[E_K](IV, D)$

1: **for** $i = 1, \ldots, |D|_n$ **do** $Y_i \leftarrow E_K(\mathsf{add}(IV, i))$ **end for**
2: $Y \leftarrow \mathsf{msb}_{|D|}(Y_1 \| \cdots \| Y_{|D|_n})$; $D' \leftarrow D \oplus Y$; **return** D'

$\cup_{i \geq 0}\{0,1\}^{in}$. Let $\mathsf{msb}_i(X)$ and $\mathsf{lsb}_i(X)$ denote the most and least significant i bits of a bit string X, respectively. If $i \geq |X|$, then $\mathsf{msb}_i(X) = X$ and $\mathsf{lsb}_i(X) = X$. Let $\mathtt{bin}_n(i)$ denote an n-bit representation of an integer $i \geq 0$. Let $|D|_n \coloneqq \lceil |D|/n \rceil$ denote the n-bit block length of D. For a value or set X, $Y \leftarrow X$ means that Y is set to X. $(D_1, D_2) \xleftarrow{n,|D|-n} D$ means that a bit string D is partitioned into the n-bit string D_1 and the $(|D|-n)$-bit string D_2 such that $D = D_1 \| D_2$, and $(D_1, \ldots, D_\ell) \xleftarrow{n} D$ means that D is partitioned into bit strings $D_1, \ldots, D_\ell$ such that $|D_i| = n$ for $i = 1, \ldots, \ell - 1$, $|D_\ell| \leq n$, and $D = D_1 \| \cdots \| D_\ell$. For a variable X and an integer i, $X \xleftarrow{+} i$ means that $X \leftarrow X + i$. $T \xleftarrow{\$} \mathcal{T}$ means that an element is chosen uniformly at random from a non-empty set $\mathcal{T}$ and assigned to T. For two sets $\mathcal{T}_1$ and $\mathcal{T}_2$, $\mathcal{T}_1 \xleftarrow{\cup} \mathcal{T}_2$ means that $\mathcal{T}_1 \leftarrow \mathcal{T}_1 \cup \mathcal{T}_2$. For a non-negative integer j and a set $\mathcal{S}$, let $\mathsf{UConf}_j(\mathcal{S}) \coloneqq \{\{S_1, S_2, \ldots, S_j\} \mid \forall i_1 \in [j], i_2 \in [j] \backslash \{i_1\} : S_{i_1} \neq S_{i_2} \wedge S_{i_1}, S_{i_2} \in \mathcal{S}\}$.

3 Specifications of HCTR2, CHCTR2, and HCTR2-TwKD

3.1 Primitives

Block Cipher (BC). We fix the key and block sizes of the underlying BCs to k and n. Let $\mathcal{BC}$ be the set of all block ciphers with k-bit keys and n-bit blocks. For a key $K \in \{0,1\}^k$ and $E \in \mathcal{BC}$, the BC is denoted by $E_K(\cdot) = E(K, \cdot)$. The decryption is denoted by E_K^{-1}. Let $E_K^{\pm} \coloneqq (E_K, E_K^{-1})$.

Keyed Hash Function. HCTR2 and its extensions CHCTR2 and HCTR2-TwKD use a family of keyed hash functions where the key space is $\{0,1\}^n$. Let $\mathcal{T} \subset \{0,1\}^*$ and $\mathcal{M}_* \coloneqq \{M \in \{0,1\}^* \mid 0 \leq |M| \leq r_{\mathsf{max}} n\}$ be the sets of tweaks and plaintexts/ciphertexts of the right part, respectively. Let $\mathcal{K}_{\mathsf{H}} \coloneqq \{0,1\}^n$ be the hash key space. Let $\{\mathsf{H}_H\}_{H \in \mathcal{K}_{\mathsf{H}}} : \mathcal{T} \times \mathcal{M}_* \to \{0,1\}^n$ be a family of keyed hash functions with n-bit keys H in $\{0,1\}^n$. We assume that there exists a constant $C \in \{0,1\}^n$ such that for any $T \in \mathcal{T}$ and $D \in \mathcal{M}_*$, $\mathsf{H}_{0^n}(T, D) = C$. Polynomial hash functions have this property.

3.2 HCTR2

HCTR2 is a tweakable wide-block encryption mode based on a BC E_K and a keyed hash function H_H. Let $K \in \{0,1\}^k$ be a BC key. The hash key is defined as $H = E_K(\mathtt{bin}_n(0))$.

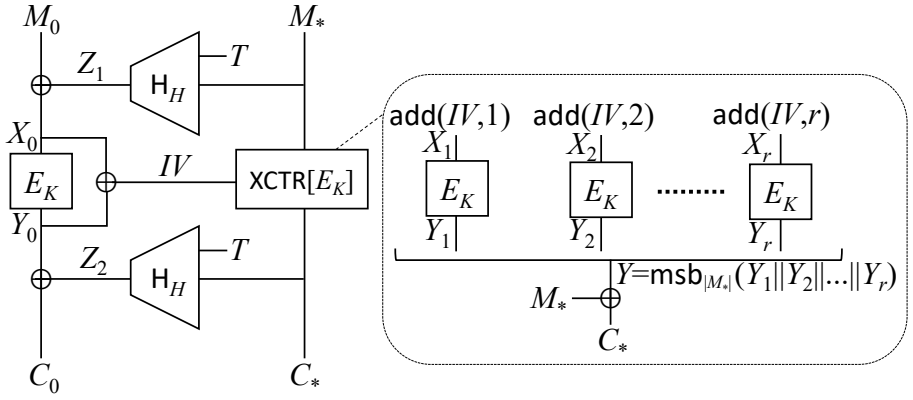


Fig. 3. HCTR2 and XCTR. The hash key is $H = E_K(\mathtt{bin}_n(0))$. $r = |M_*|_n$.

HCTR2 has three rounds. In the first round, H_H is performed. In the second round, E_K is performed once to generate IV of the encryption scheme $\mathsf{XCTR}[E_K]$, and then $\mathsf{XCTR}[E_K]$ is performed. In the third round, H_H is performed.

Let m_{max} be the maximum n-bit block length of plaintexts and $r_{\mathsf{max}} \coloneqq m_{\mathsf{max}} - 1$ be the maximum block length of plaintexts of the right part. The set of plaintexts of HCTR2 is defined as $\mathcal{M} \coloneqq \{M \in \{0,1\}^* \mid n \le |M| \le (r_{\mathsf{max}}+1)n\}$. Hence, HCTR2 accepts plaintexts of length at least n bits. Let $\mathsf{add} : \{0,1\}^n \times [r_{\mathsf{max}}] \to \{0,1\}^n$ be a function used in XCTR that takes an n-bit initial value IV and a counter i, and returns the plaintext at the i-th BC call in XCTR. We assume that add is a linear function defined as $\mathsf{add}(IV, i) = IV \oplus A_i$ where A_i is an n-bit value associated with i such that $\forall IV, i, j$ s.t. $i \neq j$: $\mathsf{add}(IV, i) \neq \mathsf{add}(IV, j)$. Note that XCTR uses the BC's key-dependent value defined as $A_i = E_K(\mathtt{bin}_n(1)) \oplus \mathtt{bin}_n(i)$, which is covered by the definition of add.

HCTR2 consists of encryption and decryption algorithms. These algorithms and the underlying mode XCTR are defined in Algorithm 1 and depicted in Fig. 3. HCTR2 is the encryption algorithm that takes a plaintext $M \in \mathcal{M}$ and a tweak $T \in \mathcal{T}$, and returns the ciphertext C. $\mathsf{HCTR2}^{-1}$ is the decryption one.

3.3 CHCTR2

We present a new tweakable wide-block encryption mode CHCTR2 for BBB security. CHCTR2 has a cascaded construction of HCTR2. It has two BC keys K_1 and K_2, which are respectively used for the first and second HCTR2 calls. The encryption CHCTR2 and the decryption $\mathsf{CHCTR2}^{-1}$ are defined as

$$\mathsf{CHCTR2}[E_{K_1}, E_{K_2}, \mathsf{H}](T, M) \coloneqq \mathsf{HCTR2}[E_{K_2}, \mathsf{H}]\left(T, \mathsf{HCTR2}[E_{K_1}, \mathsf{H}](T, M)\right),$$
$$\mathsf{CHCTR2}^{-1}[E^{\pm}_{K_1}, E^{\pm}_{K_2}, \mathsf{H}](T, C) \coloneqq \mathsf{HCTR2}^{-1}[E^{\pm}_{K_1}, \mathsf{H}]\left(T, \mathsf{HCTR2}^{-1}[E^{\pm}_{K_2}, \mathsf{H}](T, C)\right).$$

In CHCTR2, for $i \in [2]$ and $j \in [r]$, the add function of the XCTR invocation at the i-th round uses the BC's key-dependent counter. Let A_i be an n-bit value associated with i such that $\forall IV, i_1, i_1$ s.t. $i_1 \neq i_2$: $A_{i_1} \neq A_{i_2}$. Then, add of CHCTR2 is defined as $\mathsf{add}(IV, j) = IV \oplus A_j \oplus L_i$ where $L_i = E_{K_i}(\mathtt{bin}_n(1))$. Let $(X_{1,0}, Y_{1,0})$ and $(X_{2,0}, Y_{2,0})$ be respectively the pairs of plaintext and ciphertext blocks of E_{K_1} and E_{K_2} in the left part. For $j \in [r]$, let $(X_{1,j}, Y_{1,j})$ and $(X_{2,j}, Y_{2,j})$ be, respectively, the j-th pairs of E_{K_1} and E_{K_2} in the XCTR invocation of the right part. Let $Z_1, Z_{1,2}$, and Z_2 be respectively the hash

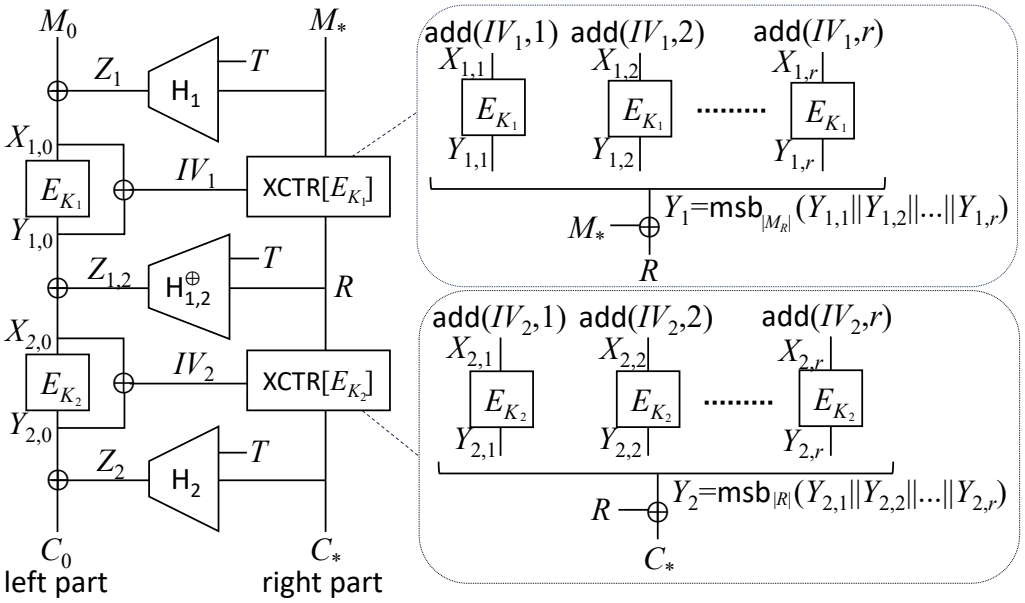


Fig. 4. CHCTR2. $r = |M_*|_n$. $\mathsf{H}_1 \coloneqq \mathsf{H}_{H_1}$ and $\mathsf{H}_2 \coloneqq \mathsf{H}_{H_2}$. The hash keys are defined as $H_1 = E_{K_1}(\mathtt{bin}_n(0))$ and $H_2 = E_{K_2}(\mathtt{bin}_n(0))$. The second hash value is defined as $Z_{1,2} = \mathsf{H}^{\oplus}_{1,2}(T, R) = \mathsf{H}_1(T, R) \oplus \mathsf{H}_2(T, R)$. For $i \in [2]$ and $j \in [r]$, $\mathsf{add}(IV_i, j) = A_j \oplus L_i$ where $L_i = E_{K_i}(\mathtt{bin}_n(1))$.

values of the first, second, and third hash layers. Note that $Z_{1,2}$ is the result of the sum of two hash values. The detailed construction of CHCTR2 is given in Fig. 4.

3.4 HCTR2-TwKD

For specific applications wherein the number of repetitions of the same tweak is limited, we consider HCTR2-TwKD which is HCTR2 with a tweak-based key derivation function (KDF). Let $F : \mathcal{L} \times \{0,1\}^{t_0} \to \{0,1\}^k$ be a KDF wherein $\mathcal{L}$ is the key space of the function. In HCTR2-TwKD, a tweak $T \in \mathcal{T}$ such that $|T| \geq t_0$ is partitioned into the t_0-bit part T_0 and the remaining part T_* such that $T = T_0 \| T_*$ and the number of repetitions of the same pair of a key L and a tweak T_0 is limited. For a key $L \in \mathcal{L}$, F with L is denoted by F_L. The encryption HCTR2-TwKD and the decryption HCTR2-TwKD^{-1} are defined as

$$\mathsf{HCTR2\text{-}TwKD}[F_L, E, \mathsf{H}](T, M) \coloneqq \mathsf{HCTR2}[E_{F_L(T_0)}, \mathsf{H}]\,(T_*, M)\,,$$
$$\mathsf{HCTR2\text{-}TwKD}^{-1}[F_L, E^{\pm}, \mathsf{H}](T, C) \coloneqq \mathsf{HCTR2}^{-1}[E^{\pm}_{F_L(T_0)}, \mathsf{H}]\,(T_*, C)\,.$$

4 Security Model and Tool

4.1 Distinguishing Advantage

We consider distinguishing-type security notions for BCs and VIL-STPRPs. For the security notions, we define the following distinguishing advantage of an adversary $\mathbf{A}$ that has access to either $\mathcal{O}_1$ or $\mathcal{O}_2$ and returns a decision bit. For $i \in [2]$, let $\mathbf{A}^{\mathcal{O}_i} = 1$ be an event that $\mathbf{A}$ with $\mathcal{O}_i$ returns 1. The distinguishing advantage function of $\mathbf{A}$ is defined as

$$\mathbf{Adv}^{\mathsf{dist}}_{\mathcal{O}_1,\mathcal{O}_2}(\mathbf{A}) \coloneqq \Pr\left[\mathbf{A}^{\mathcal{O}_1} = 1\right] - \Pr\left[\mathbf{A}^{\mathcal{O}_2} = 1\right].$$

4.2 Security Model for CHCTR2, HCTR2, and HCTR2-TwKD

We prove the security of CHCTR2, HCTR2, and HCTR2-TwKD in the sense of multi-user variable-input-length tweakable SPRP (mu-VIL-STPRP) security.

Let u be the number of users. Let $\Sigma \in \{\mathsf{CHCTR2}, \mathsf{HCTR2}, \mathsf{HCTR2\text{-}TwKD}\}$ be the target scheme. Let $\mathcal{K}$ be the key space of Σ. Let Σ_K be the encryption of the target scheme with a key $K \in \mathcal{K}$ and Σ_K^{-1} be the decryption. Let $\widetilde{\mathsf{WPerm}}$ be the set of all tweakable variable-length permutations with the same interface as Σ_K. In the mu-VIL-STPRP game, an adversary interacts with either the real-world or ideal-world oracles: $\mathcal{O}_{\mathsf{real}} \coloneqq (\Sigma_{K^{[\nu]}}, \Sigma^{-1}_{K^{[\nu]}})_{\nu\in[u]}$ and $\mathcal{O}_{\mathsf{ideal}} \coloneqq (\widetilde{\Psi}_\nu, \widetilde{\Psi}_\nu^{-1})_{\nu\in[u]}$, where $\forall \nu \in [u]$: $K^{[\nu]} \xleftarrow{\$} \mathcal{K}$ (real world) and $\widetilde{\Psi}_\nu \xleftarrow{\$} \widetilde{\mathsf{WPerm}}$ (ideal world). $K^{[\nu]}$ is the ν-th user's key and $\widetilde{\Psi}_\nu$ is the ν-th user's VIL-tweakable RP. At the end of this game, $\mathbf{A}$ returns a decision bit in $\{0,1\}$. Let $\mathbf{A}^{\mathcal{O}} \in \{0,1\}$ be an output of $\mathbf{A}$ with access to a set of oracles $\mathcal{O}$. Then, the mu-VIL-STPRP advantage function of $\mathbf{A}$ is defined as

$$\mathbf{Adv}^{\mathsf{mu\text{-}vil\text{-}stprp}}_{\Sigma}(\mathbf{A}) \coloneqq \mathbf{Adv}^{\mathsf{dist}}_{\mathcal{O}_{\mathsf{real}},\mathcal{O}_{\mathsf{ideal}}}(\mathbf{A}).$$

4.3 Proof Tool: Expectation Method

We refer to the set of query-response tuples (and additional values) that an adversary obtains in a security game as a "transcript." Let T_R be a transcript obtained by sampling in the real world, i.e., sampling of hash keys and RPs. Let T_I be a transcript obtained by sampling in the ideal world, i.e., sampling of $\widetilde{\Psi}_1, \ldots, \widetilde{\Psi}_u$ (and additional values). We call a transcript τ *valid* if $\Pr[\mathsf{T}_I = \tau] > 0$. Let $\mathcal{T}$ be the set of all valid transcripts such that $\forall \tau \in \mathcal{T} : \Pr[\mathsf{T}_R = \tau] \le \Pr[\mathsf{T}_I = \tau]$. Then, we have $\mathsf{SD}(\mathsf{T}_R, \mathsf{T}_I) \coloneqq \sum_{\tau\in\mathcal{T}}(\Pr[\mathsf{T}_I = \tau] - \Pr[\mathsf{T}_R = \tau])$.

We derive the upper-bound of the statistical distance using the expectation method of Hoang and Tessaro [23], which is a generalization of the coefficient H technique [35]. Here, $\mathcal{T}$ is partitioned into good transcripts $\mathcal{T}_{\mathsf{good}}$ and bad transcripts $\mathcal{T}_{\mathsf{bad}}$.

Lemma 1. *Let $\mathcal{T}_{\mathsf{good}}$ and $\mathcal{T}_{\mathsf{bad}}$ be good and bad transcripts, respectively. If there exists a non-negative function $g(\tau)$, which maps any valid transcript to a non-negative real value, such that for any $\tau \in \mathcal{T}_{\mathsf{good}}$, $\frac{\Pr[\mathsf{T}_R=\tau]}{\Pr[\mathsf{T}_I=\tau]} \ge 1 - g(\tau)$, then $\mathsf{SD}(\mathsf{T}_R, \mathsf{T}_I) \le \Pr[\mathsf{T}_I \in \mathcal{T}_{\mathsf{bad}}] + \mathsf{Ex}[g(\tau)]$.*

We thus show the following three points: (1) define good and bad transcripts; (2) upper-bound $\Pr[\mathsf{T}_I \in \mathcal{T}_{\mathsf{bad}}]$; (3) lower-bound $\frac{\Pr[\mathsf{T}_R=\tau]}{\Pr[\mathsf{T}_I=\tau]}$ using some non-negative function g and evaluate the expected value $\mathsf{Ex}[g(\tau)]$. Putting the bound and the expected value into the above lemma, we obtain the upper-bound.

4.4 Multi-user SPRP (mu-SPRP) Assumption

In our proofs, the underlying BCs are assumed to be mu-SPRP secure. Under the assumption, BC instantiations with independent keys are securely replaced with independent random permutations (RPs). Let $\mathsf{Perm}(n)$ be the set of n-bit permutations. Let u be the number of users. In the mu-SPRP game, an adversary $\mathbf{A}$ interacts with either the real-world oracles $(E^{\pm}_{K^{[1]}}, \ldots, E^{\pm}_{K^{[u]}})$ or the ideal-world oracles $(\pi_1^{\pm}, \ldots, \pi_u^{\pm})$, where $\forall \nu \in [u] : K^{[\nu]} \xleftarrow{\$} \{0,1\}^k$, $\pi_\nu \xleftarrow{\$} \mathsf{Perm}(n)$, and $\pi^{\pm} \coloneqq (\pi_\nu, \pi_\nu^{-1})$. At the end of this game, $\mathbf{A}$ returns a decision bit in $\{0,1\}$. The mu-SPRP advantage function of $\mathbf{A}$ is defined as

$$\mathbf{Adv}^{\mathsf{mu\text{-}sprp}}_{E}(\mathbf{A}) \coloneqq \mathbf{Adv}^{\mathsf{dist}}_{(E^{\pm}_{K_1},\ldots,E^{\pm}_{K_u}),(\pi_1^{\pm},\ldots,\pi_u^{\pm})}(\mathbf{A}).$$

For all possible adversaries $\mathbf{A}$ that have access to u users, make at most q queries, and run in time t, the maximum advantage is defined as

$$\mathbf{Adv}_E^{\mathsf{mu\text{-}sprp}}(u,q,t) \coloneqq \max_{\mathbf{A}} \mathbf{Adv}_E^{\mathsf{mu\text{-}sprp}}(\mathbf{A}).$$

4.5 Multi-user PRF (mu-PRF) Assumption

In our proof of HCTR2-TwKD, we assume that the KDF F is mu-PRF secure. Let u be the number of users. Let $\mathsf{Func}(t_0,k)$ be the set of all functions from $\{0,1\}^{t_0}$ to $\{0,1\}^k$. In the mu-PRF-security game, an adversary $\mathbf{A}$ has access to either real-world oracles $(F_{L^{[1]}},\ldots,F_{L^{[u]}})$ or ideal-world ones $(\mathcal{R}_1,\ldots,\mathcal{R}_u)$, where $\forall \nu \in [u] : L^{[\nu]} \xleftarrow{\$} \mathcal{L}$ and $\mathcal{R}_\nu \xleftarrow{\$} \mathsf{Func}(t_0,k)$. At the end of this game, $\mathbf{A}$ returns a decision bit. Then, the advantage function of $\mathbf{A}$ is defined as

$$\mathbf{Adv}_F^{\mathsf{mu\text{-}prf}}(\mathbf{A}) \coloneqq \mathbf{Adv}^{\mathsf{dist}}_{(F_{L^{[\nu]}})_{\nu\in[u]},(\mathcal{R}_\nu)_{\nu\in[u]}}(\mathbf{A}).$$

For all possible adversaries $\mathbf{A}$ that have access to u users, make at most q queries, and run in time t, the maximum advantage is defined as

$$\mathbf{Adv}_F^{\mathsf{mu\text{-}prf}}(u,q,t) \coloneqq \max_{\mathbf{A}} \mathbf{Adv}_F^{\mathsf{mu\text{-}prf}}(\mathbf{A}).$$

4.6 Universal Hash Function

Let $\{\mathsf{H}_H\}_{H\in\mathcal{K}_\mathsf{H}}$ be a family of hash functions with key space $\mathcal{K}_\mathsf{H} \coloneqq \{0,1\}^n$, input space $\mathcal{T}\times\{0,1\}^*$, and hash space $\{0,1\}^n$. In our proofs, we use the following assumptions on the hash functions.

Definition 1 (Regular Hash, Almost XOR Universal Hash). *Let $\omega \geq 1$. $\{\mathsf{H}_H\}_{H\in\mathcal{K}_\mathsf{H}}$ is said to be ω-regular if for any $Y\in\{0,1\}^n$, $(T,M)\in\mathcal{T}\times\mathcal{M}_*$, and $b\in\{0,1\}$,*

$$\Pr_{H\xleftarrow{\$}\{0,1\}^n}[\mathsf{H}_H(T,M) = Y \oplus bH] \leq \frac{\omega(|T|_n + |M|_n + 1)}{2^n}.$$

$\{\mathsf{H}_H\}_{H\in\mathcal{K}_\mathsf{H}}$ is said to be ω-AXU if for any $Y\in\{0,1\}^n$ and $\{(T,M),(T',M')\}\in\mathsf{UConf}_2(\mathcal{T}\times\mathcal{M}_*)$,

$$\Pr_{H\xleftarrow{\$}\{0,1\}^n}[\mathsf{H}_H(T,M)\oplus\mathsf{H}_H(T',M') = Y] \leq \frac{\omega\left(\max\{|T|_n+|M|_n,|T'|_n+|M'|_n\}+1\right)}{2^n}.$$

POLYVAL and GHASH satisfy $\omega = 1$.

5 Security of CHCTR2

5.1 Security Bound

In the proof of CHCTR2, we use the following assumption for the hash family $\{\mathsf{H}_H\}_{H\in\mathcal{K}_\mathsf{H}}$.

Definition 2 (Injective Property of First Input Block). *The injective property of $\{\mathsf{H}_H\}_{H\in\mathcal{K}_\mathsf{H}}$ is such that for any $H\in\mathcal{K}_\mathsf{H}\backslash\{0^n\}$, $T\in\mathcal{T}$, $D,D'\in\{0,1\}^n$, and $D_*\in\{0,1\}^*$ such that $D\neq D'$, $\mathsf{H}_H(T,D\|D_*)\neq\mathsf{H}_H(T,D'\|D_*)$*

Polynomial hash functions such as POLYVAL and GHASH have the injective property.

The following theorem shows the mu-VIL-STPRP-security bound of CHCTR2.

Theorem 1. *Assume that* $\{\mathsf{H}_H\}_{H\in\mathcal{K}_\mathsf{H}}$ *is* ω*-regular and* ω*-AXU, and has the above injective property. For any adversary* $\mathbf{A}$ *running in time* t *and making* q *queries to* u *users such that the total number of* n*-bit data (tweak and plaintext/ciphertext) blocks in all queries is* σ*, the maximum number of* n*-bit data blocks per user is* σ_u*, the maximum number of queries per user is* q_u*, the maximum number of* n*-bit plaintext blocks per query is* m_max*, and the maximum number of* n*-bit data blocks per query is* ℓ*, we have*

$$\mathbf{Adv}^{\mathsf{mu\text{-}vil\text{-}stprp}}_{\mathsf{CHCTR2}}(\mathbf{A}) \le \frac{2m_\mathsf{max}\sigma_\mathsf{u}^2\sigma}{2^{2n}} + \frac{35\omega^2\ell\sigma_\mathsf{u}\min\{\sigma_\mathsf{u}q, q_\mathsf{u}\sigma\}}{2^{2n}} + \frac{15\ell m_\mathsf{max}\sigma_\mathsf{u}\sigma}{2^{2n}} + \frac{12\omega\sigma}{2^n} + \mathbf{Adv}^{\mathsf{mu\text{-}sprp}}_{E}(2u, \sigma, t + O(\sigma)).$$

An overview of the proof is given below. The proof is given in Section 6.

5.2 Proof Overview

We prove the mu-VIL-STPRP security of CHCTR2 by using the expectation method (a generalization of the coefficient H technique) with the resampling method [5, 30]. The resampling method is a novel way of defining dummy internal values in the ideal world. In the proofs with this method, the (dummy) internal values and (dummy) keys are revealed to an adversary after finishing all queries. We explain the resampling method with the sum of two RPs SoP [27], since the right part of CHCTR2 can be seen as an iteration of SoP. Let $\dot\pi$ and $\ddot\pi$ be independent RPs. For an input $X \in \{0,1\}^n$, the output T of SoP is defined as $T = Y_1 \oplus Y_2$ where $Y_1 = \dot\pi(X)$ and $Y_2 = \ddot\pi(X)$. We consider the single-user-PRF-security of SoP (i.e., $u = 1$), i.e., indistinguishability between a single instance of SoP and an n-bit RF $\mathcal{R}$.

In the ideal world where an adversary interacts with an n-bit RF, when all queries are finished, or put differently, when all input-output pairs $((X^{(1)}, T^{(1)}), \ldots, (X^{(q)}, T^{(q)}))$ are fixed, we sample each dummy ciphertext of $\dot\pi$. For each α-th query $X^{(\alpha)}$, we first sample the (dummy) ciphertext $Y_1^{(\alpha)}$ of $\dot\pi$ as $Y_1^{(\alpha)} \xleftarrow{\$} \{0,1\}^n$ and then the other one $Y_2^{(\alpha)}$ is defined as $Y_2^{(\alpha)} = T^{(\alpha)} \oplus Y_1^{(\alpha)}$ according to the structure of SoP. Since all outputs of $\mathcal{R}$ are independently chosen, a collision may occur for the ciphertexts, i.e., $\exists\beta \in [\alpha-1], i \in [2]$ s.t. $X^{(\alpha)} \neq X^{(\beta)} \wedge Y_i^{(\alpha)} = Y_i^{(\beta)}$. The collision is incompatible with the property of RPs, and the collision probability is about $O(\frac{\alpha}{2^n})$. Hence, only with the first sampling, the ciphertexts are distinguishable from those in the real world with the birthday bound $\sum_{\alpha\in[q]} \frac{\alpha}{2^n} = O(\frac{q^2}{2^n})$.

The resampling method can improve the birthday bound by introducing the second sampling. After the first sampling fails for the α-th query, we discard the ciphertexts of the first sampling and freshly sample $Y_1^{(\alpha)}$ again. Then, the other one $Y_2^{(\alpha)}$ is defined as $Y_2^{(\alpha)} = T^{(\alpha)} \oplus Y_1^{(\alpha)}$. For each $\alpha \in [q]$, the collision probability of the second sampling is $\frac{2(\alpha-1)}{2^n}$, giving a $2n/3$-bit security bound $\sum_{\alpha\in[q]}(\frac{2(\alpha-1)}{2^n})^2 = O(\frac{q^3}{2^{2n}})$.[††]

For good transcripts τ, we lower-bound $\frac{\Pr[\mathsf{T}_R=\tau]}{\Pr[\mathsf{T}_I=\tau]}$, the ratio of the real-world and ideal-world probabilities. Although the resampling method reduces the collision probability explained above, each dummy ciphertext is sampled at most twice, and the number of chances to satisfy the target good transcript is increased, i.e., $\Pr[\mathsf{T}_I = \tau]$ is increased. In the real

[††]Note that if the number of samplings is three (or more) times, we can improve the collision probability to $O(\frac{q^4}{2^{3n}})$. However, the $2n/3$-bit term $\frac{q^3}{2^{2n}}$ appears in the bound of the ratio for good transcripts. In Section 5.3, we show that if the number of samplings is at most 2, the ratio is ≥ 1. However, when the third sampling is introduced, the probability that the first and second samplings fail is introduced, which is about $\sum_{\alpha\in[q]}\left(\frac{2(\alpha-1)}{2^n}\right)^2 = O(\frac{q^3}{2^{2n}})$, which cannot be canceled out by the budget of the ideal world.

world, all pairs of plaintext and ciphertext are defined by RPs $(\dot{\pi}, \ddot{\pi})$. In the ideal world, responses to construction queries are defined by an RF $\mathcal{R}$, and each dummy ciphertext is chosen uniformly at random from $\{0,1\}^n$. Hence, the sampling space of the ideal world is larger than that of the real world, which we call a "budget" of the ideal world. As in the proofs of triple encryption [29] and of key-alternating ciphers [5], the impact of the increase can be canceled out by using the budget of the ideal world, and providing the nice lower-bound $\frac{\Pr[\mathsf{T}_R=\tau]}{\Pr[\mathsf{T}_I=\tau]} \geq 1$. The details are given in Section 5.3.

In our proof of CHCTR2, we update the sampling method according to the structure of CHCTR2. In CHCTR2, the right part consists of two XCTR invocations, but the IVs IV_1 and IV_2 are defined by using two encrypted keyed hash functions. Hence, we need to consider collisions of the hash functions. Moreover, one IV IV_i is defined by using the other one IV_{3-i} via XCTR and the hash function in the middle layer of CHCTR2. Hence, we carefully define the sampling algorithm by considering collisions of the hash functions and the dependency between IV_1 and IV_2. The proof is given in Section 6, where we successfully define the sampling algorithm that offers the $2n/3$-bit security bound of CHCTR2.

5.3 Good Transcript Evaluation for PRF-Security of SoP

Let τ be a target good transcript. We evaluate $\frac{\Pr[\mathsf{T}_R=\tau]}{\Pr[\mathsf{T}_I=\tau]}$, the ratio of the real-world and ideal-world probabilities.

Real-world probability. For each $\alpha \in [q]$, $Y_1^{(\alpha)}$ and $Y_2^{(\alpha)}$ are respectively chosen uniformly at random from $2^n - (\alpha - 1)$ elements in $\{0,1\}^n$ by RPs $\dot{\pi}$ and $\ddot{\pi}$. Hence, the probability that $Y_1^{(\alpha)}$ and $Y_2^{(\alpha)}$ satisfy τ is $\left(\frac{1}{2^n-(\alpha-1)}\right)^2$.

Ideal-world probability. For each $\alpha \in [q]$, the probability that the α-th output $T^{(\alpha)}$ satisfies τ is $\frac{1}{2^n}$. Regarding $Y_1^{(\alpha)}$, since there are two chances of sampling $Y_1^{(\alpha)}$, we evaluate the probability for each case.

- The probability that $Y_1^{(\alpha)}$ satisfies τ is $\frac{1}{2^n}$. The probability that $T^{(\alpha)}$ and $Y_1^{(\alpha)}$ defined by the first sampling satisfy τ is $\left(\frac{1}{2^n}\right)^2$.
- The probability that the first sampling fails (a collision occurs at the first sampling on either $Y_1^{(\alpha)}$ or $Y_2^{(\alpha)}$) is $\frac{2(\alpha-1)}{2^n}$. Hence, the probability that $T^{(\alpha)}$ and $Y_1^{(\alpha)}$ by the second sampling satisfy τ is $\left(\frac{1}{2^n}\right)^2 \cdot \frac{2(\alpha-1)}{2^n}$.

So, the probability that $T^{(\alpha)}$ and $Y_1^{(\alpha)}$ (and $Y_2^{(\alpha)}$) satisfy τ is $\left(\frac{1}{2^n}\right)^2 \cdot \left(1 + \frac{2(\alpha-1)}{2^n}\right)$.

Lower-Bound of $\frac{\Pr[\mathsf{T}_R=\tau]}{\Pr[\mathsf{T}_I=\tau]}$. By using the above probabilities, the ratio of the real-world and ideal-world probabilities for good transcripts is

$$\begin{aligned}
\prod_{\alpha\in[q]} \frac{\left(\frac{1}{2^n-(\alpha-1)}\right)^2}{\left(\frac{1}{2^n}\right)^2 \cdot \left(1 + \frac{2(\alpha-1)}{2^n}\right)} &= \prod_{\alpha\in[q]} \left(\left(1 - \frac{\alpha-1}{2^n}\right)^2 \cdot \left(1 + \frac{2(\alpha-1)}{2^n}\right)\right)^{-1} \\
&= \prod_{\alpha\in[q]} \left(\left(1 - \frac{2(\alpha-1)}{2^n} + \left(\frac{\alpha-1}{2^n}\right)^2\right) \cdot \left(1 + \frac{2(\alpha-1)}{2^n}\right)\right)^{-1} \\
&= \prod_{\alpha\in[q]} \left(1 - 3\left(\frac{\alpha-1}{2^n}\right)^2 + 2\left(\frac{\alpha-1}{2^n}\right)^3\right)^{-1}.
\end{aligned}$$

The term $2\left(\frac{\alpha-1}{2^n}\right)^3$ is introduced due to the second sampling that increases the number of chances to satisfy a good transcript. On the other hand, the term "$3\left(\frac{\alpha-1}{2^n}\right)^2$," can be used as the "budget" of the ideal world. Since $3\left(\frac{\alpha-1}{2^n}\right)^2 \geq 2\left(\frac{\alpha-1}{2^n}\right)^3$, the term $2\left(\frac{\alpha-1}{2^n}\right)^3$ is canceled out by the budget, and we obtain the lower bound $\frac{\Pr[\mathsf{T}_R=\tau]}{\Pr[\mathsf{T}_I=\tau]} \geq 1$.

6 Proof of Theorem 1

Without loss of generality, we assume that the adversary is deterministic and makes no repeated or trivial query.

We first replace the BCs $(E^{\pm}_{K_1^{[\nu]}}, E^{\pm}_{K_2^{[\nu]}})_{\nu\in[u]}$ with RPs $(\dot{\pi}_\nu^{\pm}, \ddot{\pi}_\nu^{\pm})_{\nu\in[u]}$ where $\forall\nu \in [u]$: $\dot{\pi}_\nu \xleftarrow{\$} \mathsf{Perm}(n)$, $\ddot{\pi}_\nu \xleftarrow{\$} \mathsf{Perm}(n)$, $\dot{\pi}_\nu^{\pm} \coloneqq (\dot{\pi}_\nu, \dot{\pi}_\nu^{-1})$, $\ddot{\pi}_\nu^{\pm} \coloneqq (\ddot{\pi}_\nu, \ddot{\pi}_\nu^{-1})$, introducing the $\mathsf{mu\text{-}SPRP}$ advantage function $\mathbf{Adv}^{\mathsf{mu\text{-}sprp}}_E(2u, \sigma, t+O(\sigma))$ in the bound. Hence, in the following evaluation, the oracles $\mathcal{O}_{\mathsf{real}} = \left(\mathsf{CHCTR2}[\dot{\pi}_\nu, \ddot{\pi}_\nu, \mathsf{H}], \mathsf{CHCTR2}^{-1}[\dot{\pi}_\nu^{\pm}, \ddot{\pi}_\nu^{\pm}, \mathsf{H}]\right)_{\nu\in[u]}$ are the real-world ones.

6.1 Definitions

Let q_ν be the number of queries to the ν-th user. Without loss of generality, we assume that $\forall\nu \in [u] : q_\nu \geq 1$. Let σ_ν be the total number of n-bit data blocks in queries to the ν-th user. Hence, $\sigma = \sum_{\nu\in[u]} \sigma_\nu$. We refer to the stage in which an adversary makes queries as the *query stage* and the stage after the query stage as the *decision stage*. The values $X_{1,0}$ and $Y_{2,0}$ in the left part of $\mathsf{CHCTR2}$ are called *edge values*.

For $\alpha \in [q]$, values corresponding to the α-th query are denoted using the superscript (α), e.g., $M^{(\alpha)}$, $T^{(\alpha)}$, $X^{(\alpha)}_{i,j}$, $Y^{(\alpha)}_{i,j}$, etc. Let $t_\alpha \coloneqq |T^{(\alpha)}|_n$ be the n-bit block length of the tweak $T^{(\alpha)}$. Let $r_\alpha = |M^{(\alpha)}_*|_n$ be the n-bit block length of $M^{(\alpha)}_*$. Let $m_\alpha \coloneqq |M^{(\alpha)}|_n$ be the block length of the plaintext $M^{(\alpha)}$, i.e., $m_\alpha = r_\alpha + 1$. Let $\ell_\alpha \coloneqq t_\alpha + m_\alpha$ be the number of n-bit data blocks in the α-th query. Let $m^+_\alpha \coloneqq nm_\alpha - |M^{(\alpha)}|$ be the number of truncated bits in XCTR in the α-th query. For a plaintext $M^{(\alpha)}_*$ (resp. ciphertext $C^{(\alpha)}_*$) in the right part, let $M^{(\alpha)}_i$ (resp. $C^{(\alpha)}_i$) be the i-th n-bit plaintext (resp. ciphertext) block, i.e., $M^{(\alpha)}_* = M^{(\alpha)}_1 \| \cdots \| M^{(\alpha)}_{r_\alpha}$ and $C^{(\alpha)}_* = C^{(\alpha)}_1 \| \cdots \| C^{(\alpha)}_{r_\alpha}$. Note that $0 < |M^{(\alpha)}_{r_\alpha}| \leq n$ and $0 < |C^{(\alpha)}_{r_\alpha}| \leq n$. Let $\mathsf{u}_\alpha \in [u]$ be the user index corresponding to the α-th query, i.e., $\mathsf{u}_\alpha = \nu$ when the query is sent to the ν-th user. For $\nu \in [u]$ and $i \in [2]$, let $H^{[\nu]}_i$ and $L^{[\nu]}_i$ be the i-th round hash key and the i-th round mask value of the ν-th user. For $\nu \in [u]$, let $\mathcal{Q}^{[\nu]} \coloneqq \{\alpha \in [q] \mid \mathsf{u}_\alpha = \nu\}$ be the set of query indexes for the ν-th user. For $\alpha \in [q]$, let query_α be a predicate indicating the type of the α-th query: when the α-th query is an encryption query, we write $\mathsf{query}_\alpha \coloneqq \mathsf{enc}$; otherwise, $\mathsf{query}_\alpha \coloneqq \mathsf{dec}$. For $\nu \in [u]$, let $\mathcal{P}^{[\nu]}_1 \coloneqq \{(X^{(\alpha)}_{1,j}, Y^{(\alpha)}_{1,j}) \mid \alpha \in \mathcal{Q}^{[\nu]}, j \in [0, r_\alpha]\}$ and $\mathcal{P}^{[\nu]}_2 \coloneqq \{(X^{(\alpha)}_{2,j}, Y^{(\alpha)}_{2,j}) \mid \alpha \in \mathcal{Q}^{[\nu]}, j \in [0, r_\alpha]\}$ be the sets of all pairs of plaintext and ciphertext blocks of $\dot{\pi}_\nu^{\pm}$ and $\ddot{\pi}_\nu^{\pm}$ in queries to the ν-th user, respectively. For $i \in [2]$, if $(X, Y) \in \mathcal{P}^{[\nu]}_i$, then $\mathcal{P}^{[\nu]}_i(X) \coloneqq Y$ and $\mathcal{P}^{-1[\nu]}_i(Y) \coloneqq X$. If $(X, *) \notin \mathcal{P}^{[\nu]}_i$ (resp. $(*, Y) \notin \mathcal{P}^{[\nu]}_i$), then $\mathcal{P}^{[\nu]}_i(X) \coloneqq \varepsilon$ (resp. $\mathcal{P}^{-1[\nu]}_i(Y) \coloneqq \varepsilon$). For $i \in [2]$, let $\mathcal{X}^{[\nu]}_i \coloneqq \{X \mid (X, *) \in \mathcal{P}^{[\nu]}_i\}$ and $\mathcal{Y}^{[\nu]}_i \coloneqq \{Y \mid (*, Y) \in \mathcal{P}^{[\nu]}_i\}$. For $\nu \in [u]$, let $\mathsf{H}^{[\nu]}_1 \coloneqq \mathsf{H}_{H^{[\nu]}_1}$ and $\mathsf{H}^{[\nu]}_2 \coloneqq \mathsf{H}_{H^{[\nu]}_2}$. For $\nu \in [u]$ and query indexes $\alpha, \beta \in \mathcal{Q}^{[\nu]}$, we denote the query order by $\alpha \prec \beta$ (resp. $\alpha \preceq \beta$) if the internal values of the β-th query are defined after those of the α-th query are defined (resp. $\alpha \prec \beta$ or $\alpha = \beta$). Note that in the real world, $\alpha < \beta \Rightarrow \alpha \prec \beta$ and $\alpha \leq \beta \Rightarrow \alpha \preceq \beta$.

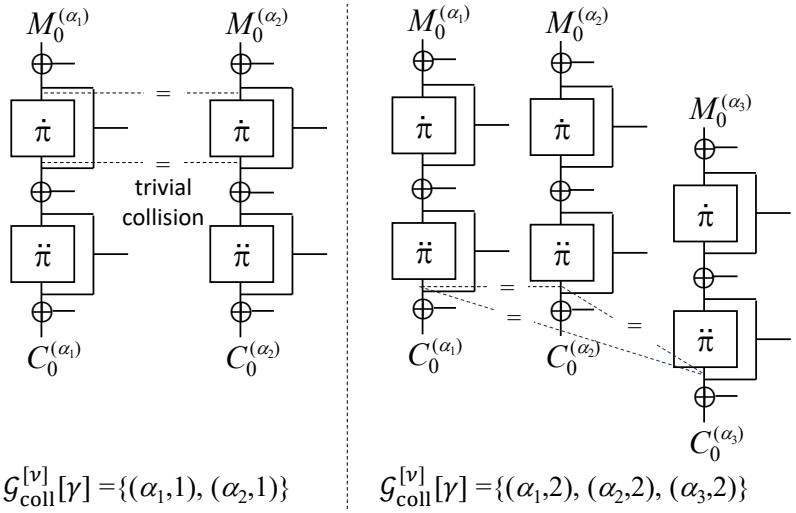


Fig. 5. Collision sets with a single collision (left) and with a 3-multi-collision (right) where the right parts of CHCTR2 are omitted.

6.2 Collision Set

For each $\nu \in [u]$, we define collision sets for edge values $(X^{(\alpha)}_{1,0}, Y^{(\alpha)}_{2,0})_{\alpha \in \mathcal{Q}^{[\nu]}}$. For each $\nu \in [u]$, a (γ-th) collision set $\mathcal{G}^{[\nu]}_{\mathsf{coll}}[\gamma]$ includes query indexes of the ν-th user with the same collisions for either $X_{1,0}$ or $Y_{2,0}$.

A (γ-th) collision set $\mathcal{G}^{[\nu]}_{\mathsf{coll}}[\gamma]$ is a set of pairs of query and round indexes with a collision of an edge value. For example, for distinct query indexes $\alpha_1, \alpha_2 \in \mathcal{Q}^{[\nu]}$ with the collision $X^{(\alpha_1)}_{1,0} = X^{(\alpha_2)}_{1,0}$, the collision set is defined as $\mathcal{G}^{[\nu]}_{\mathsf{coll}}[\gamma] \coloneqq \{(\alpha_1, 1), (\alpha_2, 1)\}$. If a multi-collision occurs, then the collision set keeps all query indexes with the multi-collision. For example, for distinct query indexes $\alpha_1, \alpha_2, \ldots, \alpha_j \in \mathcal{Q}^{[\nu]}$ with the multi-collision $Y^{(\alpha_1)}_{2,0} = Y^{(\alpha_2)}_{2,0} = \cdots = Y^{(\alpha_j)}_{2,0}$, the collision set is defined as $\mathcal{G}^{[\nu]}_{\mathsf{coll}}[\gamma] \coloneqq \{(\alpha_1, 2), (\alpha_2, 2), \ldots, (\alpha_j, 2)\}$. See Fig. 5.

For a collision set $\mathcal{G}^{[\nu]}_{\mathsf{coll}}[\gamma] \coloneqq \{(\alpha_1, i), (\alpha_2, i), \ldots, (\alpha_j, i)\}$ where $i \in [2]$, the (j-multi-)collision ensures that if $i = 1$, then the (j-multi-)collision $Y^{(\alpha_1)}_{1,0} = Y^{(\alpha_2)}_{1,0} = \cdots = Y^{(\alpha_j)}_{1,0}$ occurs; if $i = 2$, then the (j-multi-)collision $X^{(\alpha_1)}_{2,0} = X^{(\alpha_2)}_{2,0} = \cdots = X^{(\alpha_j)}_{2,0}$ occurs. Such collisions are called "trivial collisions".

Let $\mathcal{G}^{[\nu]}_{\mathsf{coll}}$ be the family of collision sets of the ν-th user. For a collision set $\mathcal{G}^{[\nu]}_{\mathsf{coll}}$ and $(\dot{\mathrm{a}}, *) \in \mathcal{G}^{[\nu]}_{\mathsf{coll}}$, we say $\dot{\mathrm{a}}$ is a longest query index if the block length of the plaintext $M^{(\dot{\mathrm{a}})}$ is maximum in the collision set, i.e.,$\forall (\alpha, *) \in \mathcal{G}^{[\nu]}_{\mathsf{coll}} \backslash \{(\dot{\mathrm{a}}, *)\} : |M^{(\alpha)}| \leq |M^{(\dot{\mathrm{a}})}|$. We say the other queries are non-longest queries. For each $\nu \in [u]$, we fix the indexes of the collision sets in $\mathcal{G}^{[\nu]}_{\mathsf{coll}}$ such that the plaintext length of the longest query of the γ-th collision set is greater than or equal to the plaintext length of the longest query of the $(\gamma - 1)$-th collision set. Note that query indexes with both $X_{1,0}$ and $Y_{2,0}$ collisions are considered in the bad event ecoll defined later.

Let $\mathcal{Q}^{[\nu]}_{\mathsf{coll}}[\gamma] \coloneqq \{\alpha \mid (\alpha, *) \in \mathcal{G}^{[\nu]}_{\mathsf{coll}}[\gamma]\}$ be query indexes in $\mathcal{G}^{[\nu]}_{\mathsf{coll}}[\gamma]$. Let $\mathcal{Q}^{[\nu]}_{\mathsf{coll}} \coloneqq \cup_{\gamma \in [|\mathcal{G}^{[\nu]}_{\mathsf{coll}}|]} \mathcal{Q}^{[\nu]}_{\mathsf{coll}}[\gamma]$. Let $\mathcal{Q}^{[\nu]}_{\mathsf{non\text{-}coll}} \subseteq \mathcal{Q}^{[\nu]}$ be query indexes of the ν-th user without collisions of the edge values. For $\mathcal{Q}^{[\nu]}_{\mathsf{non\text{-}coll}}$, we use the symbols $\mathrm{a}_1, \ldots, \mathrm{a}_{|\mathcal{Q}^{[\nu]}_{\mathsf{coll}}|}$ such that $\mathcal{Q}^{[\nu]}_{\mathsf{non\text{-}coll}} \coloneqq \{\mathrm{a}_1, \ldots, \mathrm{a}_{|\mathcal{Q}^{[\nu]}_{\mathsf{non\text{-}coll}}|}\}$ and $\forall i \in [|\mathcal{Q}^{[\nu]}_{\mathsf{non\text{-}coll}}| - 1] : |M^{(\mathrm{a}_i)}| \leq |M^{(\mathrm{a}_{i+1})}|$, i.e., the symbols are sorted by the plaintext lengths.

Algorithm 2 Main Sampling Algorithm for CHCTR2

1: **for** $\nu \in [u]$ **do**
2: $H_1^{[\nu]} \xleftarrow{\$} \{0,1\}^n$; $H_2^{[\nu]} \xleftarrow{\$} \{0,1\}^n$; $L_1^{[\nu]} \xleftarrow{\$} \{0,1\}^n$; $L_2^{[\nu]} \xleftarrow{\$} \{0,1\}^n$; $\mathcal{P}_1^{[\nu]} \leftarrow \emptyset$; $\mathcal{P}_2^{[\nu]} \leftarrow \emptyset$
3: **for** $\alpha \in \mathcal{Q}^{[\nu]}$ **do**
4: $\mathsf{sample}_\alpha \leftarrow 0$; **for** $i \in [2]$; $j \in [0, r_{\mathsf{max}}]$ **do** $(X_{i,j}^{(\alpha)}, Y_{i,j}^{(\alpha)}) \leftarrow (\varepsilon, \varepsilon)$ **end for**
5: $Z_1^{(\alpha)} \leftarrow \mathsf{H}_1^{[\nu]}(T^{(\alpha)}, M_*^{(\alpha)})$; $X_{1,0}^{(\alpha)} \leftarrow Z_1^{(\alpha)} \oplus M_0^{(\alpha)}$
6: $Z_2^{(\alpha)} \leftarrow \mathsf{H}_2^{[\nu]}(T^{(\alpha)}, C_*^{(\alpha)})$; $Y_{2,0}^{(\alpha)} \leftarrow Z_2^{(\alpha)} \oplus C_0^{(\alpha)}$
7: **if** $\mathsf{query}_\alpha = \mathsf{enc}$ **then**
$C_+^{(\alpha)} \xleftarrow{\$} \{0,1\}^{m_\alpha^+}$; $\hat{C}^{(\alpha)} \leftarrow C^{(\alpha)} \| C_+^{(\alpha)}$; $\hat{M}^{(\alpha)} \leftarrow M^{(\alpha)} \| 0^{m_\alpha^+}$
8: **else** $M_+^{(\alpha)} \xleftarrow{\$} \{0,1\}^{m_\alpha^+}$; $\hat{M}^{(\alpha)} \leftarrow M^{(\alpha)} \| M_+^{(\alpha)}$; $\hat{C}^{(\alpha)} \leftarrow C^{(\alpha)} \| 0^{m_\alpha^+}$ **end if**
9: **end for**
10: **for** $\gamma = 1, \ldots, |\mathcal{G}_{\mathsf{coll}}^{[\nu]}|$ **do** ▷ Sampling for collision sets $\mathcal{G}_{\mathsf{coll}}^{[\nu]}$
11: Let $(\dot{\mathrm{a}}, s) \in \mathcal{G}_{\mathsf{coll}}[\gamma]$ s.t. $\forall (\alpha, s) \in \mathcal{G}_{\mathsf{coll}}[\gamma] : |M^{(\alpha)}| \leq |M^{(\dot{\mathrm{a}})}|$
12: **run** $\mathsf{CHCTR2\text{-}Sample}_1(\dot{\mathrm{a}})$
13: **for** $(\alpha, s) \in \mathcal{G}_{\mathsf{coll}}[\gamma] \backslash \{(\dot{\mathrm{a}}, s)\}$ **do run** $\mathsf{CHCTR2\text{-}Sample}_2(\alpha, \dot{\mathrm{a}}, s)$ **end for**
14: **for** $(\alpha, *) \in \mathcal{G}_{\mathsf{coll}}[\gamma]$ **do** $\mathsf{sample}_\alpha \xleftarrow{+} 1$; $\mathcal{P}_1^{[\nu]} \xleftarrow{\cup} \mathcal{P}_1^{(\alpha)}$; $\mathcal{P}_2^{[\nu]} \xleftarrow{\cup} \mathcal{P}_2^{(\alpha)}$ **end for**
15: **end for**
16: **for** $i = 1, \ldots, |\mathcal{Q}_{\mathsf{non\text{-}coll}}^{[\nu]}|$ **do** ▷ Sampling for non-collision queries
17: **run** $\mathsf{CHCTR2\text{-}Sample}_1(\mathrm{a}_i)$
18: **if** $\mathsf{sample}_{\mathrm{a}_i} = 1$ **then**
19: **run** $\mathsf{CHCTR2\text{-}Sample}_1(\mathrm{a}_i)$
20: **end if**
21: $\mathcal{P}_1^{[\nu]} \xleftarrow{\cup} \mathcal{P}_1^{(\mathrm{a}_i)}$; $\mathcal{P}_2^{[\nu]} \xleftarrow{\cup} \mathcal{P}_2^{(\mathrm{a}_i)}$
22: **end for**
23: **end for**
24: **if** $\exists \nu \in [u], i \in [2], (X_{i,j_1}^{(\alpha)}, Y_{i,j_1}^{(\alpha)}), (X_{i,j_2}^{(\beta)}, Y_{i,j_2}^{(\beta)}) \in \mathcal{P}_i^{[\nu]}$ s.t.
$(X_{i,j_1}^{(\alpha)} = X_{i,j_2}^{(\beta)} \wedge Y_{i,j_1}^{(\alpha)} \neq Y_{i,j_2}^{(\beta)}) \vee (X_{i,j_1}^{(\alpha)} \neq X_{i,j_2}^{(\beta)} \wedge Y_{i,j_1}^{(\alpha)} = Y_{i,j_2}^{(\beta)})$ **then**
25: **for** $(\alpha, i, j) \in [q] \times [2] \times [0, r_{\mathsf{max}}]$ **do** $(X_{i,j}^{(\alpha)}, Y_{i,j}^{(\alpha)}) \leftarrow (\varepsilon, \varepsilon)$ **end for**
26: **end if**

6.3 Sampling Algorithm for CHCTR2

Using Algorithms 2, 3, and 4, we define dummy keys and dummy internal input-output tuples in the decision stage of the ideal world.

Algorithm 2. This is the main algorithm, which we explain below.
Lines 2 to 9 (Preliminary Procedure). First, the hash keys and the mask values of the ν-th user are defined, and the RP lists are initialized. Once the hash keys are fixed, all edge values are fixed, and thus the family of collision set $\mathcal{G}_{\mathsf{coll}}^{[\nu]}$ and the non-collision set $\mathcal{Q}_{\mathsf{non\text{-}coll}}^{[\nu]}$ are also fixed. For each $\alpha \in \mathcal{Q}^{[\nu]}$, a counter sample_α is introduced to count the number of failures of the sampling steps for the α-th query. For $(\alpha, *) \in \mathcal{G}_{\mathsf{coll}}^{[\nu]}$, sample_α is incremented by 1 at the end of the sampling procedure for the α-th query to account for the edge collision of the α-th query. For $\alpha \in \mathcal{Q}_{\mathsf{non\text{-}coll}}^{[\nu]}$, sample_α is initialized to 0. Since the block size of CHCTR2 is n bits, if $\mathsf{query}_\alpha = \mathsf{enc}$, a plaintext $M^{(\alpha)}$ and ciphertext $C^{(\alpha)}$ are respectively extended by appending the m_α^+-bit zeros $0^{m_\alpha^+}$ and an m_α^+-bit random value $C_+^{(\alpha)}$; if $\mathsf{query}_\alpha = \mathsf{dec}$, they are respectively extended by appending an m_α^+-bit random value $M_+^{(\alpha)}$ and the m_α^+-bit zeros $0^{m_\alpha^+}$. $\hat{M}^{(\alpha)}$ and $\hat{C}^{(\alpha)}$ are the extended plaintext and ciphertext. Let $\hat{M}_{i-1}^{(\alpha)}$ (resp. $\hat{C}_{i-1}^{(\alpha)}$) be the i-th n-bit block of $\hat{M}$ (resp. $\hat{C}$).
Lines 10-15 (Sampling Procedure for $\mathcal{G}_{\mathsf{coll}}^{[\nu]}$). For each $\gamma \in [|\mathcal{G}_{\mathsf{coll}}^{[\nu]}|]$ and the index $\dot{\mathrm{a}}$ of the longest query, the first sampling is executed. Using $\mathsf{CHCTR2\text{-}Sample}_1$, all plaintext-ciphertext block pairs $(X_{i,j}^{(\dot{\mathrm{a}})}, Y_{i,j}^{(\dot{\mathrm{a}})})$ are defined according to the structure of CHCTR2. Based on the

Algorithm 3 Sampling Algorithm

Subroutine $\mathsf{CHCTR2\text{-}Sample}_1(\alpha)$

1: $\nu \leftarrow \mathsf{u}_\alpha$; $Y_{1,0}^{(\alpha)} \xleftarrow{\$} \{0,1\}^n$; $X_{2,0}^{(\alpha)} \xleftarrow{\$} \{0,1\}^n$; $\hat{\mathsf{i}} \leftarrow 0$
2: **if** $X_{1,0}^{(\alpha)} \in \mathcal{X}_1^{[\nu]}$ **then** $Y_{1,0}^{(\alpha)} \leftarrow \mathcal{P}_1^{[\nu]}(X_{1,0}^{(\alpha)})$; $\mathsf{sample}_\alpha \xleftarrow{+} 3$ **end if**
3: **if** $Y_{2,0}^{(\alpha)} \in \mathcal{Y}_2^{[\nu]}$ **then** $X_{2,0}^{(\alpha)} \leftarrow \mathcal{P}_2^{-1[\nu]}(Y_{2,0}^{(\alpha)})$; $\hat{\mathsf{i}} \leftarrow 2$; $\mathsf{sample}_\alpha \xleftarrow{+} 3$ **end if**
4: $IV_1^{(\alpha)} \leftarrow X_{1,0}^{(\alpha)} \oplus Y_{1,0}^{(\alpha)}$; **for** $j \in [r_\alpha]$ **do** $X_{1,j}^{(\alpha)} \leftarrow IV_1^{(\alpha)} \oplus A_j \oplus L_1^{[\nu]}$ **end for**
5: $IV_2^{(\alpha)} \leftarrow X_{2,0}^{(\alpha)} \oplus Y_{2,0}^{(\alpha)}$; **for** $j \in [r_\alpha]$ **do** $X_{2,j}^{(\alpha)} \leftarrow IV_2^{(\alpha)} \oplus A_j \oplus L_2^{[\nu]}$ **end for**
6: **if** $\hat{\mathsf{i}} = 0 \wedge \exists j \in [r_\alpha]$ s.t. $X_{\hat{\mathsf{i}},j}^{(\alpha)} \in \mathcal{X}_2^{[\nu]} \vee X_{2,0}^{(\alpha)} = X_{2,j}^{(\alpha)}$ **then** $\hat{\mathsf{i}} \leftarrow 2$ **end if**
7: **if** $\hat{\mathsf{i}} = 0$ **then** $\hat{\mathsf{i}} \leftarrow 1$ **end if**
8: **for** $j \in [r_\alpha]$ **do**
9: $\quad Y_{\hat{\mathsf{i}},j}^{(\alpha)} \xleftarrow{\$} \{0,1\}^n$; **if** $X_{\hat{\mathsf{i}},j}^{(\alpha)} \in \mathcal{X}_{\hat{\mathsf{i}}}^{[\nu]}$ **then** $Y_{\hat{\mathsf{i}},j}^{(\alpha)} \leftarrow \mathcal{P}_{\hat{\mathsf{i}}}^{[\nu]}(X_{\hat{\mathsf{i}},j}^{(\alpha)})$ **end if**
10: $\quad$ **if** $X_{\hat{\mathsf{i}},j}^{(\alpha)} = X_{\hat{\mathsf{i}},0}^{(\alpha)}$ **then** $Y_{\hat{\mathsf{i}},j}^{(\alpha)} \leftarrow Y_{\hat{\mathsf{i}},0}^{(\alpha)}$ **end if**
11: $\quad Y_{3-\hat{\mathsf{i}},j}^{(\alpha)} \leftarrow Y_{\hat{\mathsf{i}},j}^{(\alpha)} \oplus \hat{M}_j^{(\alpha)} \oplus \hat{C}_j^{(\alpha)}$
12: **end for**
13: $R^{(\alpha)} \leftarrow \mathsf{msb}_{|M_*^{(\alpha)}|}(Y_{1,1}^{(\alpha)} \| \cdots \| Y_{1,r_\alpha}^{(\alpha)}) \oplus M_*^{(\alpha)}$
14: **if** $\hat{\mathsf{i}} = 1$ **then** $X_{2,0}^{(\alpha)} \leftarrow Y_{1,0}^{(\alpha)} \oplus \mathsf{H}_1^{[\nu]}(T^{(\alpha)}, R^{(\alpha)}) \oplus \mathsf{H}_2^{[\nu]}(T^{(\alpha)}, R^{(\alpha)})$ **end if**
15: **if** $\hat{\mathsf{i}} = 2$ **then** $Y_{1,0}^{(\alpha)} \leftarrow X_{2,0}^{(\alpha)} \oplus \mathsf{H}_1^{[\nu]}(T^{(\alpha)}, R^{(\alpha)}) \oplus \mathsf{H}_2^{[\nu]}(T^{(\alpha)}, R^{(\alpha)})$ **end if**
16: $IV_{3-\hat{\mathsf{i}}}^{(\alpha)} \leftarrow X_{3-\hat{\mathsf{i}},0}^{(\alpha)} \oplus Y_{3-\hat{\mathsf{i}},0}^{(\alpha)}$; **for** $j \in [r_\alpha]$ **do** $X_{3-\hat{\mathsf{i}},j}^{(\alpha)} \leftarrow IV_{3-\hat{\mathsf{i}}}^{(\alpha)} \oplus A_j \oplus L_{3-\hat{\mathsf{i}}}^{[\nu]}$ **end for**
17: **if** $(X_{2,0}^{(\alpha)} \in \mathcal{X}_2^{[\nu]} \wedge \hat{\mathsf{i}} = 1)$
$\quad \vee \Big(\exists i \in [2], j \in [0, r_\alpha]$ s.t. $(X_{i,j}^{(\alpha)}, Y_{i,j}^{(\alpha)}) \notin \mathcal{P}_i^{[\nu]} \wedge Y_{i,j}^{(\alpha)} \in \mathcal{Y}_i^{[\nu]} \wedge (i,j) \neq (2,0)\Big)$
$\quad \vee \Big(\exists i \in [2], j \in [r_\alpha], j' \in [0, j-1]$ s.t. $Y_{i,j}^{(\alpha)} = Y_{i,j'}^{(\alpha)}\Big)$ **then** $\mathsf{sample}_\alpha \xleftarrow{+} 1$ **end if**
18: **for** $i \in [2]$ **do** $\mathcal{P}_i^{(\alpha)} \leftarrow \Big\{(X_{i,j}^{(\alpha)}, Y_{i,j}^{(\alpha)}) \mid j \in [0, r_\alpha] \wedge (X_{i,j}^{(\alpha)}, Y_{i,j}^{(\alpha)}) \notin \mathcal{P}_i^{[\nu]}\Big\}$ **end for**
19: **if** $\mathsf{sample}_\alpha = 3$ **then** $\mathsf{sample}_\alpha \leftarrow 0$ **end if**

Algorithm 4 Assignment Algorithm

Subroutine $\mathsf{CHCTR2\text{-}Sample}_2(\alpha, \dot{\mathsf{a}}, s)$

1: $\nu \leftarrow \mathsf{u}_\alpha$; $(X_{s,0}^{(\alpha)}, Y_{s,0}^{(\alpha)}) \leftarrow (X_{s,0}^{(\dot{\mathsf{a}})}, Y_{s,0}^{(\dot{\mathsf{a}})})$; $IV_s^{(\alpha)} \leftarrow X_{s,0}^{(\alpha)} \oplus Y_{s,0}^{(\alpha)}$
2: **for** $j \in [r_\alpha]$ **do**
3: $\quad X_{s,j}^{(\alpha)} \leftarrow IV_s^{(\alpha)} \oplus A_j \oplus L_s^{[\nu]}$; $Y_{s,j}^{(\alpha)} \leftarrow Y_{s,j}^{(\dot{\mathsf{a}})}$; $Y_{3-s,j}^{(\alpha)} \leftarrow Y_{s,j}^{(\alpha)} \oplus \hat{M}_j^{(\alpha)} \oplus \hat{C}_j^{(\alpha)}$
4: **end for**
5: $R^{(\alpha)} \leftarrow \mathsf{msb}_{|M_*^{(\alpha)}|}(Y_{1,1}^{(\alpha)} \| \cdots \| Y_{1,r_\alpha}^{(\alpha)}) \oplus M_*^{(\alpha)}$
6: $Z_{1,2}^{(\alpha)} \leftarrow \mathsf{H}_1^{[\nu]}(T^{(\alpha)}, R^{(\alpha)}) \oplus \mathsf{H}_2^{[\nu]}(T^{(\alpha)}, R^{(\alpha)})$
7: **if** $s = 1$ **then** $X_{2,0}^{(\alpha)} \leftarrow Y_{1,0}^{(\alpha)} \oplus Z_{1,2}^{(\alpha)}$ **else** $Y_{1,0}^{(\alpha)} \leftarrow X_{2,0}^{(\alpha)} \oplus Z_{1,2}^{(\alpha)}$ **end if**
8: $IV_{3-s}^{(\alpha)} \leftarrow X_{3-s,0}^{(\alpha)} \oplus Y_{3-s,0}^{(\alpha)}$; **for** $j \in [r_\alpha]$ **do** $X_{3-s,j}^{(\alpha)} \leftarrow IV_{3-s}^{(\alpha)} \oplus A_j \oplus L_{3-s}^{[\nu]}$ **end for**
9: **if** $(X_{2,0}^{(\alpha)} \in \mathcal{X}_2^{[\nu]} \wedge s = 1)$
$\quad \vee \Big(\exists i \in [2], j \in [0, r_\alpha]$ s.t. $(X_{i,j}^{(\alpha)}, Y_{i,j}^{(\alpha)}) \notin \mathcal{P}_i^{[\nu]} \wedge Y_{i,j}^{(\alpha)} \in \mathcal{Y}_i^{[\nu]} \wedge (i,j) \neq (2,0)\Big)$
$\quad \vee \Big(\exists j' \in [0, j-1]$ s.t. $Y_{i,j}^{(\alpha)} = Y_{i,j'}^{(\alpha)}\Big)$ **then** $\mathsf{sample}_\alpha \xleftarrow{+} 1$ **end if**
10: **for** $i \in [2]$ **do** $\mathcal{P}_i^{(\alpha)} \leftarrow \Big\{(X_{i,j}^{(\alpha)}, Y_{i,j}^{(\alpha)}) \mid j \in [0, r_\alpha] \wedge (X_{i,j}^{(\alpha)}, Y_{i,j}^{(\alpha)}) \notin \mathcal{P}_i^{[\nu]}\Big\}$ **end for**

plaintext-ciphertext pairs of the $\dot{\mathsf{a}}$-th query, the remaining pairs within the same chain are then determined according to the structure of CHCTR2 through $\mathsf{CHCTR2\text{-}Sample}_2$. Specifically, at the s-th round, the pairs associated with the $\dot{\mathsf{a}}$-th query are propagated to the other queries as a result of collisions of the edge values.

Lines 16-22 (Sampling Procedure for $\mathcal{Q}_{\mathsf{non\text{-}coll}}^{[\nu]}$). For $i \in [|\mathcal{Q}_{\mathsf{non\text{-}coll}}^{[\nu]}|]$, The first sampling is always performed; the second is performed if the first fails, i.e., the first sampling for the a_i-th query runs $\mathsf{CHCTR2\text{-}Sample}_1$, which generates all pairs of plaintext and ciphertext

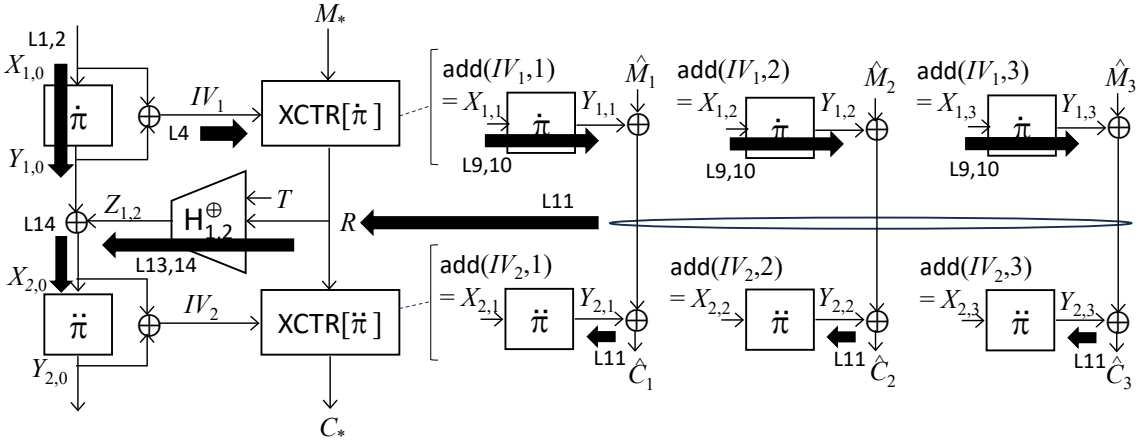


Fig. 6. Procedure of CHCTR2-Sample$_1$, where $X_{1,0}$ is fresh and $\hat{\mathsf{i}} = 1$. $\mathsf{H}^{\oplus}_{1,2}$ is the sum of H_1 and H_2.

blocks in a manner consistent with CHCTR2. If the first sampling fails, i.e., $\mathsf{sample}_{\mathrm{a}_i} = 1$, then the second sampling is executed.

Lines 24-26. In these lines, if there are pairs of plaintext and ciphertext blocks that are inconsistent with respect to RPs, then all pairs are set to $(\varepsilon, \varepsilon)$.

Algorithm 3 (CHCTR2-Sample$_1$). For $\alpha \in \mathcal{Q}^{[\nu]}$, the algorithm defines pairs of plaintext and ciphertext blocks $(X^{(\alpha)}_{i,j}, Y^{(\alpha)}_{i,j})_{i\in[2],j\in[0,r_\alpha]}$ and checks the pairs for inconsistencies for ciphertext blocks or collision for the edge values. If a pair is inconsistent or the edge collision occurs, then the counter sample_α is incremented by 1. Fig. 6 shows a diagram of the procedure.

Lines 1-7. In these lines, the flag $\hat{\mathsf{i}}$ is introduced. The flag is a round index wherein ciphertext blocks of the right part are sampled (and ciphertext blocks at the $(3-\hat{\mathsf{i}})$ are defined by using the ciphertext blocks at the $\hat{\mathsf{i}}$-th round, plaintext blocks and ciphertext blocks. If $X^{(\alpha)}_{1,0}$ is fresh (resp. not fresh), then $Y^{(\alpha)}_{1,0}$ (resp. $X^{(\alpha)}_{2,0}$) is defined and the flag $\hat{\mathsf{i}}$ is set to 1 (resp. 2), and $X^{(\alpha)}_{2,0}$ (resp. $Y^{(\alpha)}_{1,0}$) is defined after all other pairs of the α-th query are fixed so that all pairs of the α-th query are consistent with the structure of CHCTR2. Using these values, the IV and the plaintext blocks at the $\hat{\mathsf{i}}$-th round are fixed. In these lines, if an edge collision occurs, all plaintext blocks are deterministically fixed, and thus one cannot resample the corresponding value $Y_{1,0}$ or $X_{2,0}$. For handling the edge collision, sample_α is set to 3. If no edge collision occurs but a collision of plaintext blocks at the second round occurs, then $\hat{\mathsf{i}}$ is set to 2. Finally, if no above collision occurs, then $\hat{\mathsf{i}}$ is set to 1.

Lines 8-13. In these lines, all ciphertext blocks $(Y^{(\alpha)}_{1,j}, Y^{(\alpha)}_{2,j})_{j\in[r_\alpha]}$ of the right part are defined according to the structure of CHCTR2, i.e., $\hat{C}^{(\alpha)}_j = Y^{(\alpha)}_{1,j} \oplus Y^{(\alpha)}_{2,j} \oplus \hat{M}^{(\alpha)}_j$. For each j, if $X_{\hat{\mathsf{i}},j}$ is not fresh, then the corresponding ciphertext block that is previously defined is assigned to $Y_{\hat{\mathsf{i}},j}$. These values determine an input to the hash functions at the middle layer, $R^{(\alpha)}$.

Lines 14-16. Using the hash values $\mathsf{H}^{[\nu]}_1(T^{(\alpha)}, R^{(\alpha)}), \mathsf{H}^{[\nu]}_2(T^{(\alpha)}, R^{(\alpha)})$ and $Y^{(\alpha)}_{1,0}$ (resp. $X^{(\alpha)}_{2,0}$) if $\hat{\mathsf{i}} = 1$ (resp. $\hat{\mathsf{i}} = 2$), the other value $X^{(\alpha)}_{2,0}$ (resp. $Y^{(\alpha)}_{1,0}$) is determined. Hence, using the freshness of $Y^{(\alpha)}_{1,0}$ or $X^{(\alpha)}_{2,0}$, all pairs of the α-th query can be consistently defined with respect to the structure of CHCTR2. Using these values, the IV and the plaintext blocks at the $(3-\hat{\mathsf{i}})$-th round of the right part are fixed.

Lines 17-19. Once all pairs are defined, the values $Y_{1,0}^{(\alpha)}, X_{2,0}^{(\alpha)}$, and $\{Y_{i,j}^{(\alpha)} \mid (i,j) \in [2] \times [r_\alpha]\}$ are checked for inconsistencies with the RPs. Any collision detected among these values causes sample_α to be incremented by 1. Then, for each $i \in [2]$, all fresh pairs at the i-th round of the α-th query are stored in $\mathcal{P}_i^{(\alpha)}$. Finally, if $\mathsf{sample}_\alpha = 3$, then one edge collision occurs but all fresh pairs at the i-th round of the α-th query are consistently defined, thus sample_α is reset to 0. Note that if $\mathsf{sample}_\alpha = 4$, then an edge collision and a collision with the condition on line 17 occurs and the resampling step is not executed. This case is considered in the bad event $\mathsf{1edge}$. if $\mathsf{sample}_\alpha = 6$, then two edge collisions occur and the resapling step is not executed. This case is considered in the bad event $\mathsf{2edge}$.

Algorithm 4 ($\mathsf{CHCTR2\text{-}Sample_2}$). This algorithm takes the target query index α, the longest query index $\dot{a}$ whose pairs are all defined, and the round index s of the edge collision, and defines the pairs of the α-th query.
Lines 1-4. First, in the left part, the s-th-round values of the α-th query are fixed to the s-th-round values of the $\dot{a}$-th query due to the edge collision. Then, in the right part, the s-th-round values of the α-th query are fixed according to the s-th-round values of the $\dot{a}$-th query, and the $(3-s)$-th-round ciphertext blocks are defined according to the structure of $\mathsf{CHCTR2}$.
Lines 5-6. Using the ciphertext blocks defined in the previous lines, the inputs to the hash functions at the middle layer, $R^{(\alpha)}$, and the sum of the hash values $Z_{1,2}^{(\alpha)}$ are defined.
Lines 7-8. In these lines, the remaining plaintext or ciphertext blocks at the $(3-s)$-th round are defined.
Lines 9-10. After all pairs of the α-th query are defined, the new values $Y_{1,0}^{(\alpha)}, X_{2,0}^{(\alpha)}$, and $\{Y_{i,j}^{(\alpha)} \mid (i,j) \in [2] \times [r_\alpha]\}$ are checked for consistency with the RPs. A collision causes sample_α to be incremented by 1. Finally, for each $i \in [2]$, all fresh pairs at the i-th round of the α-th query are stored in $\mathcal{P}_i^{(\alpha)}$.

Summary. Our algorithms define dummy hash keys and pairs of plaintext and ciphertext blocks according to the structure of $\mathsf{CHCTR2}$. Our algorithm records the number of collisions of ciphertext blocks $Y_{i,j}$ with $(i,j) \neq (2,0)$ or $X_{2,0}$ by using sample_α. The inconsistency from the other collisions of $Y_{2,0}$ and $Y_{i,j}$ with $(i,j) \neq (2,0)$ is considered in the bad events $\mathsf{2edge}$, $\mathsf{1edge}$, and $\mathsf{2Xcoll}$ defined in Section 6.5. Hence, for any $\alpha \in [q]$, if $\mathsf{sample}_\alpha \leq 1$ and these events do not occur, then all pairs of plaintext and ciphertext blocks are consistently defined.

6.4 Adversary's View

We first define the extended plaintexts $(\hat{M}^{(\alpha)})_{\alpha\in[q]}$ and the extended ciphertexts $(\hat{C}^{(\alpha)})_{\alpha\in[q]}$ in the real world. For each $\alpha \in [q]$, if $\mathsf{query}_\alpha = \mathsf{enc}$, then $\hat{M}^{(\alpha)} := M^{(\alpha)} \| 0^{m_\alpha^+}$ and $\hat{C}^{(\alpha)} := C^{(\alpha)} \| \mathsf{lsb}_{m_\alpha^+}\big(Y_{1,r_\alpha}^{(\alpha)} \oplus Y_{2,r_\alpha}^{(\alpha)}\big)$, where $Y_{1,r_\alpha}^{(\alpha)} \oplus Y_{2,r_\alpha}^{(\alpha)}$ is the XOR of the truncated ciphertext blocks that is not used during encryption. If $\mathsf{query}_\alpha = \mathsf{dec}$, then $\hat{M}^{(\alpha)} := M^{(\alpha)} \| \mathsf{lsb}_{m_\alpha^+}\big(Y_{1,r_\alpha}^{(\alpha)} \oplus Y_{2,r_\alpha}^{(\alpha)}\big)$ and $\hat{C}^{(\alpha)} := C^{(\alpha)} \| 0^{m_\alpha^+}$. In the ideal world, these extended values are defined by Algorithm 2.

In the decision stage, all users' keys and pairs of plaintext and ciphertext blocks are revealed to the adversary $\mathbf{A}$. Then, the transcript τ of $\mathbf{A}$ consists of (dummy) hash keys $\mathcal{H} = (H_1^{[\nu]}, H_2^{[\nu]})_{\nu\in[u]}$, (dummy) mask values $\mathcal{L} = (L_1^{[\nu]}, L_2^{[\nu]})_{\nu\in[u]}$, query-response tuples $\{(T^{(\alpha)}, \hat{M}^{(\alpha)}, \hat{C}^{(\alpha)}) \mid \alpha \in [q]\}$, and (dummy) pairs $(\mathcal{P}_i^{[\nu]})_{\nu\in[u],i\in[2]}$.

6.5 Definitions of Good and Bad Transcripts

Conditions (or events) on bad transcripts are defined below. The set of bad transcripts $\mathcal{T}_{\mathsf{bad}}$ is a subset of $\mathcal{T}$ such that one of the conditions holds. The set of good transcripts is defined as $\mathcal{T}_{\mathsf{good}} \coloneqq \mathcal{T} \backslash \mathcal{T}_{\mathsf{bad}}$, which is the subset of $\mathcal{T}$ such that the conditions do not hold.

$\mathsf{fail}_{\mathsf{sample}}$. This condition is defined as

$$\mathsf{fail}_{\mathsf{sample}} \Leftrightarrow X_{1,0}^{(1)} = \varepsilon.$$

The condition implies that some internal values are inconsistently defined with respect to an RP. In other words, if this condition does not hold, then all internal values are consistently defined. Note that this condition does not hold in the real world.

For avoiding the case that $\mathsf{fail}_{\mathsf{sample}}$ occurs by the birthday bound, we introduce a resampling phase if a collision with ciphertext blocks $Y_{i,j}$ with $(i,j) \neq (2,0)$ or $X_{2,0}$ occurs, then the internal values are resampled. For the other collisions, we consider the following bad conditions $\mathsf{2edge}$, $\mathsf{1edge}$, $\mathsf{2Xcoll}$, and $\mathsf{hit}_{0,1}$. The condition $\mathsf{2edge}$ considers collisions of two edges $X_{1,0}$ and $Y_{2,0}$ of the same query. The condition $\mathsf{1edge}$ considers a collision of one edge $X_{1,0}$ or $Y_{2,0}$, and a collision of plaintext blocks $X_{i,j}$ with $(i,j) \neq (1,0)$ or ciphertext blocks $Y_{i,j}$ with $(i,j) \neq (2,0)$. The condition $\mathsf{2Xcoll}$ considers two collisions of plaintext blocks $X_{i,j}$ with $(i,j) \neq (1,0)$ or ciphertext blocks $Y_{i,j}$ with $(i,j) \neq (2,0)$ with the distinct rounds. Note that a collision of plaintext blocks $X_{i,j}$ does not break the consistency, since in our algorithm, if a collision of a plaintext block occurs, then the ciphertext block is set to the previous ciphertext one. The condition $\mathsf{hit}_{0,1}$ considers a collision of pairs for hash keys or mask values.

$\mathsf{2edge}$. The condition is defined as

$$\begin{aligned}\mathsf{2edge} \Leftrightarrow \exists \nu \in [u], \alpha, \beta_1, \beta_2 \in \mathcal{Q}^{[\nu]} \text{ s.t.} \\ \big((X_{1,0}^{(\alpha)} = X_{1,0}^{(\beta_1)} \wedge \alpha \neq \beta_1) \vee (\exists j_1 \in [r_{\beta_1}] \text{ s.t. } X_{1,0}^{(\alpha)} = X_{1,j_1}^{(\beta_1)} \wedge \beta_1 \prec \alpha)\big) \wedge \\ \big((Y_{2,0}^{(\alpha)} = Y_{2,0}^{(\beta_2)} \wedge \alpha \neq \beta_2) \vee (\exists j_2 \in [r_{\beta_2}] \text{ s.t. } Y_{2,0}^{(\alpha)} = Y_{2,j_2}^{(\beta_2)} \wedge \beta_2 \prec \alpha)\big).\end{aligned}$$

The condition implies that for some α, collisions with both of the edge values $X_{1,0}^{(\alpha)}$ and $Y_{2,0}^{(\alpha)}$ in the left part occur. Assuming that $\mathsf{2edge}$ does not occur, for any query index α in $\mathcal{Q}_{\mathsf{coll}}^{[\nu]}[\gamma]$, no collision occurs for at least one of the edge values.

$\mathsf{1edge}$. The condition is defined as

$$\mathsf{1edge} \Leftrightarrow \exists \nu \in [u], \alpha, \beta_1, \beta_2 \in \mathcal{Q}^{[\nu]} \text{ s.t. } \mathsf{1edge}_1 \vee \mathsf{1edge}_2,$$

where $\mathsf{1edge}_1$ and $\mathsf{1edge}_2$ are defined as follows.

$$\begin{aligned}\mathsf{1edge}_1 \Leftrightarrow \Big((X_{1,0}^{(\alpha)} = X_{1,0}^{(\beta_1)} \wedge \alpha \neq \beta_1) \vee (\exists j_1 \in [r_{\beta_1}] \text{ s.t. } X_{1,0}^{(\alpha)} = X_{1,j_1}^{(\beta_1)} \wedge \beta_1 \prec \alpha)\Big) \wedge \\ \Big(\exists j_2 \in [0, r_\alpha], j_2' \in [0, r_{\beta_2}] \text{ s.t. } (X_{2,j_2}^{(\alpha)} = X_{2,j_2'}^{(\beta_2)} \vee Y_{2,j_2}^{(\alpha)} = Y_{2,j_2'}^{(\beta_2)}) \wedge \beta_2 \preceq \alpha \wedge (\alpha, j_2) \neq (\beta_2, j_2')\Big). \\ \mathsf{1edge}_2 \Leftrightarrow \Big((Y_{2,0}^{(\alpha)} = Y_{2,0}^{(\beta_1)} \wedge \alpha \neq \beta_1) \vee (\exists j_1 \in [r_{\beta_1}] \text{ s.t. } Y_{2,0}^{(\alpha)} = Y_{2,j_1}^{(\beta_1)} \wedge \beta_1 \prec \alpha)\Big) \wedge \\ \Big(\exists j_2 \in [0, r_\alpha], j_2' \in [0, r_{\beta_2}] \text{ s.t. } (X_{1,j_2}^{(\alpha)} = X_{1,j_2'}^{(\beta_2)} \vee Y_{1,j_2}^{(\alpha)} = Y_{1,j_2'}^{(\beta_2)}) \wedge \beta_2 \preceq \alpha \wedge (\alpha, j_2) \neq (\beta_2, j_2')\Big).\end{aligned}$$

The condition implies that for some α and $i_0 \in [2]$, an edge collision at the i_0-th round occurs and a collision of X values at the $(3-i_0)$-th round occurs, where $i_0 = 1$ for $\mathsf{1edge}_1$ and $i_0 = 2$ for $\mathsf{1edge}_2$.

2Xcoll. The condition is defined as

$$\begin{aligned}\mathsf{2Xcoll} \Leftrightarrow \exists \nu \in [u], \alpha, \beta_1, \beta_2 \in \mathcal{Q}^{[\nu]}, j_1, j_2 \in [r_\alpha], j_1' \in [0, r_{\beta_1}], j_2' \in [0, r_{\beta_2}] \text{ s.t.} \\ X_{1,j_1}^{(\alpha)} = X_{1,j_1'}^{(\beta_1)} \wedge X_{2,j_2}^{(\alpha)} = X_{2,j_2'}^{(\beta_2)} \\ \wedge\, \beta_1 \preceq \alpha \wedge \beta_2 \preceq \alpha \wedge (\alpha, j_1) \neq (\beta_1, j_1') \wedge (\alpha, j_2) \neq (\beta_2, j_2').\end{aligned}$$

The condition implies that for some α, collisions of X values at both the first and second rounds occur.

$\mathsf{hit}_{0,1}$. The condition is defined as

$$\begin{aligned}\mathsf{hit}_{0,1} \Leftrightarrow \exists \alpha \in [q], i \in [2], j \in [0, r_\alpha] \text{ s.t. } X_{i,j}^{(\alpha)} = \mathtt{bin}_n(0) \vee Y_{i,j}^{(\alpha)} = H_i^{[\mathsf{u}_\alpha]} \vee \\ X_{i,j}^{(\alpha)} = \mathtt{bin}_n(1) \vee Y_{i,j}^{(\alpha)} = L_i^{[\mathsf{u}_\alpha]} \vee \\ H_i^{[\mathsf{u}_\alpha]} = L_i^{[\mathsf{u}_\alpha]}\end{aligned}$$

The condition implies that for some α and i, the hash key $H_i^{[\mathsf{u}_\alpha]}$ or the mask value $L^{[\mathsf{u}_\alpha]}$ is used in the process of CHCTR2, except for the hash key or the mask generation. In other words, assuming that $\mathsf{hit}_{0,1}$ does not occur, all hash keys and mask values can be independent of the other RP invocations in CHCTR2.

6.6 Evaluation for Bad Transcript: Upper-Bounding $\Pr[\mathsf{T}_I \in \mathcal{T}_{\mathsf{bad}}]$

For each $\mathsf{E} \in \{\mathsf{hit}_{0,1}, \mathsf{2edge}, \mathsf{1edge}, \mathsf{2Xcoll}, \mathsf{fail}_{\mathsf{sample}}\}$, let E^* be an event that E occurs before the other events in $\{\mathsf{hit}_{0,1}, \mathsf{2edge}, \mathsf{2Xcoll}, \mathsf{1edge}, \mathsf{fail}_{\mathsf{sample}}\} \backslash \{\mathsf{E}\}$ occur. We then have

$$\begin{aligned}\Pr[\mathsf{T}_I \in \mathcal{T}_{\mathsf{bad}}] &\leq \Pr[\mathsf{hit}_{0,1}^*] + \Pr[\mathsf{2edge}^*] + \Pr[\mathsf{1edge}^*] + \Pr[\mathsf{2Xcoll}^*] + \Pr[\mathsf{fail}_{\mathsf{sample}}^*] \\ &\leq \frac{12\omega\sigma}{2^n} + \frac{2\omega^2\ell(q_{\mathsf{u}}+1)\min\{q_{\mathsf{u}}\sigma, \sigma_{\mathsf{u}}q\}}{2^{2n}} \\ &\quad + \frac{24\omega^2\ell\sigma_{\mathsf{u}}\min\{q_{\mathsf{u}}\sigma, \sigma_{\mathsf{u}}q\} + 15\ell m_{\mathsf{max}}\sigma_{\mathsf{u}}\sigma}{2^{2n}} \\ &\quad + \frac{m_{\mathsf{max}}\sigma_{\mathsf{u}}^2\sigma}{2^{2n}} + \frac{m_{\mathsf{max}}\sigma_{\mathsf{u}}^2\sigma + 3\omega^2\ell q_{\mathsf{u}}\min\{q_{\mathsf{u}}\sigma, \sigma_{\mathsf{u}}q\}}{2^{2n}} \\ &\leq \frac{2m_{\mathsf{max}}\sigma_{\mathsf{u}}^2\sigma}{2^{2n}} + \frac{30\omega^2\ell\sigma_{\mathsf{u}}\min\{\sigma_{\mathsf{u}}q, q_{\mathsf{u}}\sigma\}}{2^{2n}} + \frac{15\ell m_{\mathsf{max}}\sigma_{\mathsf{u}}\sigma}{2^{2n}} + \frac{12\omega\sigma}{2^n}. \qquad (1)\end{aligned}$$

These bounds are given below.

Evaluation of $\Pr[\mathsf{hit}_{0,1}^*]$. Fix $\alpha \in [q]$. We first evaluate the probability of the collisions for the hash keys $X_{i,j}^{(\alpha)} = \mathtt{bin}_n(0) \vee Y_{i,j}^{(\alpha)} = H_i^{[\mathsf{u}_\alpha]}$ occurs. We consider the following cases.

– For the collisions with $X_{1,0}^{(\alpha)}$, $Y_{1,0}^{(\alpha)}$, $X_{2,0}^{(\alpha)}$, or $Y_{2,0}^{(\alpha)}$, the regular hash functions ensure that

$$\begin{aligned}\Pr[X_{1,0}^{(\alpha)} = \mathtt{bin}_n(0)] \leq \frac{\omega\ell_\alpha}{2^n}, \qquad \Pr[Y_{1,0}^{(\alpha)} = H_1^{[\mathsf{u}_\alpha]}] \leq \frac{\omega\ell_\alpha}{2^n}, \\ \Pr[X_{2,0}^{(\alpha)} = \mathtt{bin}_n(0)] \leq \frac{\omega\ell_\alpha}{2^n}, \qquad \Pr[Y_{2,0}^{(\alpha)} = H_2^{[\mathsf{u}_\alpha]}] \leq \frac{\omega\ell_\alpha}{2^n}.\end{aligned}$$

– For the collisions for $X_{i,j}^{(\alpha)}$ values such that $j \geq 1$, by the randomness of $L_i^{[\nu]}$, we have

$$\Pr[X_{i,j}^{(\alpha)} = \mathtt{bin}_n(0)] \leq \frac{1}{2^n}.$$

– For the collision $Y_{i,j}^{(\alpha)} = H_i^{[\nu]}$ such that $j \geq 1$, by the randomness of $Y_{i,j}^{(\alpha)}$ or $H_i^{[\nu]}$, we have

$$\Pr[Y_{i,j}^{(\alpha)} = H_i^{[\nu]} \mid X_{i,j}^{(\alpha)} \neq \mathtt{bin}_n(0)] \leq \frac{1}{2^n}.$$

Regarding the collisions of the mask values $X_{i,j}^{(\alpha)} = \mathtt{bin}_n(1) \vee Y_{i,j}^{(\alpha)} = L_i^{[\mathtt{u}_\alpha]}$, the evaluation is the same as that of the hash keys.

– For the collisions with $X_{1,0}^{(\alpha)}$, $Y_{1,0}^{(\alpha)}$, $X_{2,0}^{(\alpha)}$, or $Y_{2,0}^{(\alpha)}$, the regular hash functions ensure that

$$\Pr[X_{1,0}^{(\alpha)} = \mathtt{bin}_n(1)] \leq \frac{\omega\ell_\alpha}{2^n}, \qquad \Pr[Y_{1,0}^{(\alpha)} = L_1^{[\mathtt{u}_\alpha]}] \leq \frac{\omega\ell_\alpha}{2^n},$$
$$\Pr[X_{2,0}^{(\alpha)} = \mathtt{bin}_n(1)] \leq \frac{\omega\ell_\alpha}{2^n}, \qquad \Pr[Y_{2,0}^{(\alpha)} = L_2^{[\mathtt{u}_\alpha]}] \leq \frac{\omega\ell_\alpha}{2^n}.$$

– For the collisions for $X_{i,j}^{(\alpha)}$ values such that $j \geq 1$, by the randomness of $L_i^{[\nu]}$, we have

$$\Pr[X_{i,j}^{(\alpha)} = \mathtt{bin}_n(0)] \leq \frac{1}{2^n}.$$

– For the collision $Y_{i,j}^{(\alpha)} = L_i^{[\nu]}$ such that $j \geq 1$, by the randomness of $Y_{i,j}^{(\alpha)}$ or $L_i^{[\nu]}$, we have

$$\Pr[Y_{i,j}^{(\alpha)} = L_i^{[\nu]} \mid X_{i,j}^{(\alpha)} \neq \mathtt{bin}_n(1)] \leq \frac{1}{2^n}.$$

Regarding the collision $H_i^{[\mathtt{u}_\alpha]} = L_i^{[\mathtt{u}_\alpha]}$, since $H_i^{[\mathtt{u}_\alpha]}$ is chosen uniformly at random from $\{0,1\}^n$, we have

$$\Pr[H_i^{[\mathtt{u}_\alpha]} = L_i^{[\mathtt{u}_\alpha]}] \leq \frac{1}{2^n}.$$

Summing these bounds over all possible cases, we have the following bound.

$$\begin{aligned}\Pr[\mathsf{hit}_{0,1}] &\leq \sum_{\nu\in[u]} \left(\frac{1}{2^n} + \sum_{\alpha\in\mathcal{Q}^{[\nu]}} 2\left(\frac{4\omega\ell_\alpha}{2^n} + r_\alpha \cdot \frac{2}{2^n}\right)\right) \\ &\leq \sum_{\nu\in[u]} \frac{12\omega\sigma_\nu}{2^n} \\ &\leq \frac{12\omega\sigma}{2^n}.\end{aligned}$$

Evaluation of Pr[2edge*]. Fix $\nu \in [u]$, and $\alpha, \beta_1, \beta_2 \in \mathcal{Q}^{[\nu]}$. We then evaluate $\mathsf{2edge}^*$ occurs due to the α-th, β_1-th, and β_2-th queries. Note that $X_{i,j}$ values have the following structures.

$$\begin{aligned}X_{1,0} &= M_0 \oplus \mathsf{H}_1^{[\nu]}(T, M_*), \\ X_{2,0} &= Y_{1,0} \oplus \mathsf{H}_1^{[\nu]}(T,R) \oplus \mathsf{H}_2^{[\nu]}(T,R), \\ X_{1,j} &= X_{1,0} \oplus \underbrace{Y_{1,0} \oplus \mathsf{H}_1^{[\nu]}(T,R) \oplus \mathsf{H}_2^{[\nu]}(T,R)}_{=Y_{1,0}} \oplus A_j \oplus L_1^{[\nu]} \ (j \geq 1), \\ X_{2,j} &= X_{2,0} \oplus \underbrace{C_0 \oplus \mathsf{H}_2^{[\nu]}(T, C_*)}_{=Y_{2,0}} \oplus A_j \oplus L_2^{[\nu]} \ (j \geq 1).\end{aligned}$$

– Regarding $X_{1,0}^{(\alpha)} = X_{1,0}^{(\beta_1)}$ such that $\alpha \neq \beta_1$, by the AXU hash $\mathsf{H}_1^{[\nu]}$, we have

$$\Pr[X_{1,0}^{(\alpha)} = X_{1,0}^{(\beta_1)}] \leq \frac{\omega \max\{\ell_\alpha, \ell_{\beta_1}\}}{2^n}.$$

– Regarding $X_{1,0}^{(\alpha)} = X_{1,j_1}^{(\beta_1)}$ such that $\beta_1 \prec \alpha$ and $j_1 \geq 1$, by the randomness of the mask value $L_1^{[\nu]}$, we have

$$\Pr[X_{1,0}^{(\alpha)} = X_{1,j_1}^{(\beta_1)}] \leq \frac{1}{2^n}.$$

– Regarding $Y_{2,0}^{(\alpha)} = Y_{2,0}^{(\beta_2)}$ such that $\alpha \neq \beta_2$, by the AXU hash $\mathsf{H}_2^{[\nu]}$, we have

$$\Pr[Y_{2,0}^{(\alpha)} = Y_{2,0}^{(\beta_2)}] \leq \frac{\omega \max\{\ell_\alpha, \ell_{\beta_2}\}}{2^n}.$$

– Regarding $Y_{2,0}^{(\alpha)} = Y_{2,j_2}^{(\beta_2)}$ such that $\beta_2 \prec \alpha$ such that $j_2 \geq 1$, by the randomness of $Y_{2,j_2}^{(\beta_2)}$, we have

$$\Pr[Y_{2,0}^{(\alpha)} = Y_{2,j_2}^{(\beta_2)}] \leq \frac{1}{2^n}.$$

Summing the bound over each possible cases, we have

$$\begin{aligned}
\Pr[\mathsf{2edge}^*] \leq \sum_{\nu\in[u]} \Bigg(& \sum_{\substack{\alpha,\beta_1,\beta_2\in\mathcal{Q}^{[\nu]} \\ \text{s.t. } \beta_1\neq\alpha\neq\beta_2}} \left(\frac{\omega \max\{\ell_\alpha, \ell_{\beta_1}\}}{2^n} \cdot \frac{\omega \max\{\ell_\alpha, \ell_{\beta_2}\}}{2^n}\right) \\
& + \sum_{\substack{\alpha,\beta_1,\beta_2\in\mathcal{Q}^{[\nu]} \\ \text{s.t. } \beta_1\prec\alpha,\alpha\neq\beta_2}} \left(\frac{\omega \max\{\ell_\alpha, \ell_{\beta_2}\}}{2^n} \cdot \frac{r_{\beta_1}}{2^n}\right) \\
& + \sum_{\substack{\alpha,\beta_1,\beta_2\in\mathcal{Q}^{[\nu]} \\ \text{s.t. } \beta_2\prec\alpha,\alpha\neq\beta_1}} \left(\frac{\omega \max\{\ell_\alpha, \ell_{\beta_1}\}}{2^n} \cdot \frac{r_{\beta_2}}{2^n}\right) \\
& + \sum_{\substack{\alpha,\beta_1,\beta_2\in\mathcal{Q}^{[\nu]} \\ \text{s.t. } \beta_1\prec\alpha,\beta_2\prec\alpha}} \left(\frac{r_{\beta_1}}{2^n} \cdot \frac{r_{\beta_2}}{2^n}\right)\Bigg) \\
\leq \sum_{\nu\in[u]} & \frac{2\omega^2\ell(q_\nu-1)^2\sigma_\nu + 2\omega\ell q_\nu^2\sigma_\nu + m_{\mathsf{max}} q_\nu \sigma_\nu}{2^{2n}} \\
\leq & \frac{2\omega^2\ell(q_{\mathsf{u}}+1)\min\{q_{\mathsf{u}}\sigma, \sigma_{\mathsf{u}} q\}}{2^{2n}}.
\end{aligned}$$

Evaluation of Pr[1edge*]. Fix $\nu \in [u]$, and $\alpha, \beta_1, \beta_2 \in \mathcal{Q}^{[\nu]}$.

We first evaluate the probability of the collisions on the condition $\mathsf{1edge}_1$ due to the α-th, β_1-th, and β_2-th queries. Note that $X_{1,0}$ and $X_{2,j}$ have the following structures.

$$\begin{aligned}
X_{1,0} &= M_0 \oplus \mathsf{H}_1^{[\nu]}(T, M_*), \\
X_{2,0} &= Y_{1,0} \oplus \mathsf{H}_1^{[\nu]}(T, R) \oplus \mathsf{H}_2^{[\nu]}(T, R), \\
X_{2,j} &= X_{2,0} \oplus \underbrace{C_0 \oplus \mathsf{H}_2^{[\nu]}(T, C_*)}_{=Y_{2,0}} \oplus A_j \oplus L_2^{[\nu]} \ (j \geq 1).
\end{aligned}$$

We then consider the following cases. Let $\gamma = \max\{\alpha, \beta_1\}$, i.e., the γ-th query is made after the other query is made.

- Regarding $X_{1,0}^{(\alpha)} = X_{1,0}^{(\beta_1)}$ such that $\alpha \neq \beta_1$, we consider the following cases.
  - $\mathsf{query}_\gamma = \mathsf{enc}$: By the AXU hash $\mathsf{H}_1^{[\nu]}$, we have
$$\Pr[X_{1,0}^{(\alpha)} = X_{1,0}^{(\beta_1)}] \leq \frac{\omega \max\{\ell_\alpha, \ell_{\beta_1}\}}{2^n}.$$
  - $\mathsf{query}_\gamma = \mathsf{dec}$: By the randomness of $M_0^{(\gamma)}$ that is chosen uniformly at random from at least $2^n - q_{\mathsf{u}} \geq 2^n/2$ elements in $\{0,1\}^n$, we have
$$\Pr[X_{1,0}^{(\alpha)} = X_{1,0}^{(\beta_1)}] \leq \frac{2}{2^n}.$$
- Regarding $X_{1,0}^{(\alpha)} = X_{1,j_1}^{(\beta_1)}$ such that $j_1 \geq 1$ and $\beta_1 \prec \alpha$, by the randomness of the mask value $L_1^{[\nu]}$, we have
$$\Pr[X_{1,0}^{(\alpha)} = X_{1,j_1}^{(\beta_1)}] \leq \frac{1}{2^n}.$$
- Regarding $X_{2,j_2}^{(\alpha)} = X_{2,j_2'}^{(\beta_2)}$, we consider the following cases.
  - $\alpha = \beta_2$: By the counter values of XCTR, $j_2 = 0 \wedge j_2' \geq 1$ must hold. By the randomness of $L_2^{[\nu]}$, we have
$$\Pr[X_{2,0}^{(\alpha)} = X_{2,j_2'}^{(\alpha)}] \leq \frac{1}{2^n}.$$
  - $\beta_2 \prec \alpha \wedge (j_2, j_2') = (0,0)$: By the AXU hash $\mathsf{H}_2^{[\nu]}$, we have
$$\Pr[X_{2,0}^{(\alpha)} = X_{2,0}^{(\beta_2)}] \leq \frac{\omega \max\{\ell_\alpha, \ell_{\beta_2}\}}{2^n}.$$
  - $\beta_2 \prec \alpha \wedge ((j_2 = 0 \wedge j_2' \geq 1) \vee (j_2 \geq 1 \wedge j_2' = 0))$: By the randomness of $L_2^{[\nu]}$, we have
$$\Pr[X_{2,j_2}^{(\alpha)} = X_{2,j_2'}^{(\beta_2)}] \leq \frac{1}{2^n}.$$
  - $\beta_2 \prec \alpha \wedge j_2 \geq 1 \wedge j_2' \geq 1$: We consider the following cases.
    - $\beta_1 = \beta_2 \wedge \mathsf{query}_\alpha = \mathsf{enc}$: In this case, $\gamma = \alpha$. By the randomness of $C_0^{(\alpha)}$ that is chosen uniformly at random from at least $2^n - q \geq 2^n/2$ elements in $\{0,1\}^n$, we have
$$\Pr[X_{2,j_2}^{(\alpha)} = X_{2,j_2'}^{(\beta_2)}] \leq \frac{2}{2^n}.$$
    - $\beta_1 = \beta_2 \wedge \mathsf{query}_\alpha = \mathsf{dec}$: In this case, $\gamma = \alpha$. By the AXU hash $\mathsf{H}_1^{[\nu]}$, we have
$$\Pr[X_{2,j_2}^{(\alpha)} = X_{2,j_2'}^{(\beta_2)}] \leq \frac{\omega \max\{\ell_\alpha, \ell_{\beta_2}\}}{2^n}.$$
    - $\beta_1 \neq \beta_2$: By the randomness of $Y_{1,0}^{(\alpha)}$ or $Y_{1,1}^{(\alpha)}$ with the injective property of the hash functions, we have
$$\Pr[X_{2,j_2}^{(\alpha)} = X_{2,j_2'}^{(\beta_2)}] \leq \frac{1}{2^n}.$$

– Regarding $Y_{2,j_2}^{(\alpha)} = Y_{2,j_2'}^{(\beta_2)}$ such that $\beta_2 \preceq \alpha$, $j_2 \geq 1$ must hold. Since $Y_{2,j_2}^{(\alpha)} = Y_{1,j_2}^{(\alpha)} \oplus \hat{M}_{j_2}^{(\alpha)} \oplus \hat{C}_{j_2}^{(\alpha)}$ and either $\hat{M}_{j_2}^{(\alpha)}$ or $\hat{C}_{j_2}^{(\alpha)}$ is chosen uniformly at random from at least $2^n - q_\nu \geq 2^n/2$ elements, by the randomness of $\hat{M}_{j_2}^{(\alpha)}$, $\hat{C}_{j_2}^{(\alpha)}$, $Y_{1,j_2}^{(\alpha)}$ or $Y_{1,j_2'}^{(\beta_2)}$, we have

$$\Pr[Y_{2,j_2}^{(\alpha)} = Y_{2,j_2'}^{(\beta_2)}] \leq \frac{2}{2^n}.$$

We next evaluate the probability of the collisions on the condition $\mathsf{1edge}_2$. Let $\gamma = \max\{\alpha, \beta_1\}$. Note that $Y_{2,0}$ and $X_{1,j}$ have the following structures.

$$\begin{aligned}
Y_{2,0} &= C_0 \oplus \mathsf{H}_2^{[\nu]}(T, C_*),\\
X_{1,0} &= M_0 \oplus \mathsf{H}_1^{[\nu]}(T, M_*),\\
X_{1,j} &= X_{1,0} \oplus \underbrace{X_{2,0} \oplus \mathsf{H}_1^{[\nu]}(T, R) \oplus \mathsf{H}_2^{[\nu]}(T, R)}_{=Y_{1,0}} \oplus A_j \oplus L_1^{[\nu]} \ (j \geq 1).
\end{aligned}$$

We then consider the following cases.

– Regarding $Y_{2,0}^{(\alpha)} = Y_{2,0}^{(\beta_1)}$ such that $\alpha \neq \beta_1$, we consider the following cases.
  • $\mathsf{query}_\gamma = \mathsf{enc}$: By the randomness of $C_0^{(\gamma)}$ that is chosen uniformly at random from at least $2^n - q \geq 2^n/2$ elements in $\{0,1\}^n$, we have

$$\Pr[Y_{2,0}^{(\alpha)} = Y_{2,0}^{(\beta_1)}] \leq \frac{2}{2^n}.$$

  • $\mathsf{query}_\gamma = \mathsf{dec}$: By the AXU hash $\mathsf{H}_2^{[\nu]}$, we have

$$\Pr[Y_{2,0}^{(\alpha)} = Y_{2,0}^{(\beta_1)}] \leq \frac{\omega \max\{\ell_\alpha, \ell_{\beta_1}\}}{2^n}.$$

– Regarding $Y_{2,0}^{(\alpha)} = Y_{2,j_1}^{(\beta_1)}$ such that $\beta_2 \prec \alpha$ and $j_1 \geq 1$, by the randomness of $Y_{2,j_1}^{(\beta_1)}$, we have

$$\Pr[Y_{2,0}^{(\alpha)} = Y_{2,j_1}^{(\beta_1)}] \leq \frac{1}{2^n}.$$

– Regarding $X_{1,j_2}^{(\alpha)} = X_{1,j_2'}^{(\beta_2)}$ such that $\beta_2 \preceq \alpha$, $(j_2, j_2') \neq (0,0)$ must hold. We then consider the following cases.
  • $\alpha = \beta_2$: In this case, $j_2 = 0$ and $j_2' \geq 1$ must hold. By the randomness of $L_1^{[\nu]}$, we have

$$\Pr[X_{1,0}^{(\alpha)} = X_{1,j_2'}^{(\alpha)}] \leq \frac{1}{2^n}.$$

  • $\alpha \neq \beta_2 \wedge j_2 = 0$: In this case, $j_2' \geq 1$ must hold. By the randomness of $L_1^{[\nu]}$, we have

$$\Pr[X_{1,0}^{(\alpha)} = X_{1,j_2'}^{(\alpha)}] \leq \frac{1}{2^n}.$$

  • $\alpha \neq \beta_2 \wedge ((j_2 = 0 \wedge j_2' \geq 1) \vee (j_2 \geq 1 \wedge j_2' = 0))$: By the randomness of $L_1^{[\nu]}$, we have

$$\Pr[X_{1,j_2}^{(\alpha)} = X_{1,j_2'}^{(\beta_2)}] \leq \frac{1}{2^n}.$$

- $\alpha \neq \beta_2 \wedge j_2 \geq 1 \wedge j_2' \geq 1$: We consider the following cases.
  * $\beta_1 = \beta_2 \wedge \mathsf{query}_\alpha = \mathsf{enc}$: By the AXU hash $\mathsf{H}_2^{[\nu]}$, we have
$$\Pr[X_{1,j_2}^{(\alpha)} = X_{1,j_2'}^{(\beta_1)}] \leq \frac{\omega \max\{\ell_\alpha, \ell_{\beta_1}\}}{2^n}.$$
  * $\beta_1 = \beta_2 \wedge \mathsf{query}_\alpha = \mathsf{dec}$: By the randomness of $M_0^{(\alpha)}$ that is chosen uniformly at random from at least $2^n - q \geq 2^n/2$ elements in $\{0,1\}^n$, we have
$$\Pr[X_{1,j_2}^{(\alpha)} = X_{1,j_2'}^{(\beta_1)}] \leq \frac{2}{2^n}.$$
  * $\beta_1 \neq \beta_2$: By the randomness of $X_{2,0}^{(\alpha)}$ or $Y_{2,1}^{(\alpha)}$ with the injective property of the hash functions, we have
$$\Pr[X_{1,j_2}^{(\alpha)} = X_{1,j_2'}^{(\beta_2)}] \leq \frac{1}{2^n}.$$

– Regarding $Y_{1,0}^{(\alpha)} = Y_{1,j_2'}^{(\beta_2)}$ such that $\beta_2 \preceq \alpha$, we consider the following cases.
  - $j_2' = 0$: By the AXU hash $\mathsf{H}_1^{[\nu]}$, we have
$$\Pr[Y_{1,0}^{(\alpha)} = Y_{1,0}^{(\beta_2)}] \leq \frac{\omega \max\{\ell_\alpha, \ell_{\beta_2}\}}{2^n}.$$
  - $j_2' \geq 1$: By the randomness of $Y_{1,j_2'}^{(\beta_2)}$, we have
$$\Pr[Y_{1,0}^{(\alpha)} = Y_{1,j_2'}^{(\beta_2)}] \leq \frac{1}{2^n}.$$

– Regarding $Y_{1,j_2}^{(\alpha)} = Y_{1,j_2'}^{(\beta_2)}$ such that $j_2 \geq 1 \vee j_2' \geq 1$, since $Y_{2,j_2}^{(\alpha)} = Y_{1,j_2}^{(\alpha)} \oplus \hat{M}_{j_2}^{(\alpha)} \oplus \hat{C}_{j_2}^{(\alpha)}$ and either $\hat{M}_{j_2}^{(\alpha)}$ or $\hat{C}_{j_2}^{(\alpha)}$ is chosen uniformly at random from at least $2^n - q_\nu \geq 2^n/2$ elements. By the randomness of $\hat{M}_{j_2}^{(\alpha)}$, $\hat{C}_{j_2}^{(\alpha)}$, $Y_{1,j_2}^{(\alpha)}$ or $Y_{1,j_2'}^{(\beta_2)}$, we have
$$\Pr[Y_{1,j_2}^{(\alpha)} = Y_{1,j_2'}^{(\beta_2)}] \leq \frac{2}{2^n}.$$

Summing the bounds over each possible cases, we have

$$\begin{aligned}
\Pr[\mathsf{1edge}^*] &\leq \sum_{\nu\in[u]} 2 \cdot \Bigg(\sum_{\substack{\alpha,\beta_1,\beta_2 \in \mathcal{Q}^{[\nu]} \text{ s.t.} \\ \alpha\neq\beta_1 \wedge \beta_2 \preceq \alpha, \beta_1 \neq \beta_2}} \left(\frac{\omega \max\{\ell_\alpha, \ell_{\beta_1}\}}{2^n} + \frac{r_{\beta_1}+2}{2^n} \right) \\
&\qquad \times \left(\frac{3(r_\alpha+1)(r_{\beta_2}+1)}{2^n} + \frac{\omega \max\{\ell_\alpha, \ell_{\beta_2}\}}{2^n} \right) \\
&\quad + \sum_{\substack{\alpha,\beta_1 \in \mathcal{Q}^{[\nu]} \text{ s.t.} \\ \beta_1 \prec \alpha}} \left(\frac{\omega \max\{\ell_\alpha, \ell_{\beta_1}\} + r_{\beta_1}}{2^n} \cdot \frac{4(r_\alpha+1)(r_{\beta_1}+1) + \omega \max\{\ell_\alpha, \ell_{\beta_1}\}}{2^n} \right. \\
&\qquad \left. + \frac{r_{\beta_1}+2}{2^n} \cdot \frac{(r_\alpha+1)(r_{\beta_1}+1)(\omega \max\{\ell_\alpha, \ell_{\beta_1}\}+2)}{2^n} \right) \Bigg) \\
&\leq \sum_{\nu\in[u]} 2 \cdot \left(\frac{9\omega^2 \ell q_\nu \sigma_\nu^2 + 3\ell q_\nu^2 \sigma_\nu}{2^{2n}} + \frac{4\ell\sigma_\nu^2 + 2\ell q_\nu \sigma_\nu + 1.5\ell m_{\mathsf{max}} \sigma_\nu^2}{2^{2n}} \right) \\
&\leq \frac{24\omega^2 \ell \sigma_{\mathsf{u}} \min\{q_{\mathsf{u}}\sigma, \sigma_{\mathsf{u}} q\} + 15\ell m_{\mathsf{max}} \sigma_{\mathsf{u}} \sigma}{2^{2n}}.
\end{aligned}$$

Evaluation of Pr[2Xcoll*]. Fix $\nu \in [u], \alpha, \beta_1, \beta_2 \in \mathcal{Q}^{[\nu]}, j_1, j_2 \in [0, r_\alpha], j'_1 \in [0, r_{\beta_1}], j'_2 \in [0, r_{\beta_2}]$ s.t. $\beta_1 \preceq \alpha \wedge \beta_2 \preceq \alpha$. We evaluate the probability that $\mathsf{2Xcoll}^*$ occurs due to the α-th, β_1-th and β_2-th queries.

- Regarding $X^{(\alpha)}_{1,j_1} = X^{(\beta_1)}_{1,j'_1}$, $j_1 \geq 1$ must hold. We then consider the following cases.
  - $\alpha = \beta_1$: In this case, $j'_1 \geq 1$ must hold. By the counter of XCTR, we have
$$\Pr[X^{(\alpha)}_{1,j_1} = X^{(\alpha)}_{1,j'_1}] = 0.$$
  - $\beta_1 \prec \alpha \wedge j'_1 = 0$: By the randomness of $L^{[\nu]}_1$, we have
$$\Pr[X^{(\alpha)}_{1,j_1} = X^{(\beta_1)}_{1,j'_1}] \leq \frac{1}{2^n}.$$
  - $\beta_1 \prec \alpha \wedge j'_1 \geq 1$: By the randomness of $Y^{(\alpha)}_{1,0}$, we have
$$\Pr[X^{(\alpha)}_{1,j_1} = X^{(\beta_1)}_{1,j'_1}] \leq \frac{1}{2^n}.$$
- Regarding $X^{(\alpha)}_{2,j_2} = X^{(\beta_2)}_{2,j'_2}$, we consider the following cases.
  - $\alpha = \beta_2$: In this case, $j_2 = 0$ and $j'_2 \geq 1$ must hold. By the randomness of $L^{[\nu]}_2$, we have
$$\Pr[X^{(\alpha)}_{2,0} = X^{(\beta_2)}_{2,j'_2}] \leq \frac{1}{2^n}.$$
  - $\beta_2 \prec \alpha$: By the randomness of $Y^{(\alpha)}_{1,0}$ or $X^{(\alpha)}_{2,0}$, we have
$$\Pr[X^{(\alpha)}_{2,j_2} = X^{(\beta_2)}_{2,j'_2}] \leq \frac{1}{2^n}.$$

  Summing the bounds for each possible cases, we have
$$\begin{aligned}\Pr[\mathsf{2Xcoll}^*] &\leq \sum_{\nu\in[u]} \sum_{\substack{\alpha,\beta_1,\beta_2 \text{ s.t.} \\ \beta_1\preceq\alpha\wedge\beta_2\preceq\alpha}} \frac{r_\alpha(r_{\beta_1}+1)}{2^n} \cdot \frac{r_\alpha(r_{\beta_2}+1)}{2^n} \\ &\leq \sum_{\nu\in[u]} \frac{m_{\mathsf{max}}\sigma_\nu^3}{2^{2n}} \\ &\leq \frac{m_{\mathsf{max}}\sigma_{\mathsf{u}}^2\sigma}{2^{2n}}.\end{aligned}$$

Evaluation of $\Pr[\mathsf{fail}^*_{\mathsf{sample}}]$. We evaluate the probability that $X^{(1)}_{1,0} = \varepsilon$ holds, i.e., some internal values derived by our algorithm are inconsistent with respect to an RP. Our algorithm defines internal values $X_{i,j}$ and $Y_{i,j}$. The inconsistency occurs by collisions of plaintext blocks $X_{i,j}$ or ciphertext blocks $Y_{i,j}$. The inconsistency from collisions of plaintext blocks $X_{i,j}$ such that $(i,j) \neq (2,0)$ or $Y_{2,0}$ is covered by the bad conditions $\mathsf{2edge}$, $\mathsf{1edge}$, and $\mathsf{2Xcoll}$. Hence, in this part, we consider the inconsistency from collisions of ciphertext blocks $Y_{i,j}$ such that $(i,j) \neq (2,0)$ or $X_{2,0}$. In our algorithm, for some $\alpha \in [q]$, if one of ciphertext blocks $Y^{(\alpha)}_{i,j}$ such that $(i,j) \neq (2,0)$ and $X^{(\alpha)}_{2,0}$ occurs, then the internal values are resampled. The two sampling steps of our algorithm ensure that if $\mathsf{fail}_{\mathsf{sample}}$

holds, and 2edge, 1edge and 2Xcoll do not occur, then $\exists\alpha \in [q]$ s.t. $\mathsf{sample}_\alpha \geq 2$. We thus have

$$\Pr[\mathsf{fail}^*_{\mathsf{sample}}] \leq \Pr[(\exists\alpha \in [q] \text{ s.t. } \mathsf{sample}_\alpha \geq 2) \wedge \neg(\mathsf{hit}_{0,1} \vee \mathsf{2edge} \vee \mathsf{1edge} \vee \mathsf{2Xcoll}).$$

We evaluate the probability by using the following conditions.

$$\mathsf{fail}^{*1}_{\mathsf{sample}} \Leftrightarrow (\exists\nu \in [\mathsf{u}], \alpha \in \mathcal{Q}^{[\nu]}_{\mathsf{coll}} \text{ s.t. } \mathsf{sample}_\alpha \geq 2) \wedge \neg(\mathsf{hit}_{0,1} \vee \mathsf{2edge} \vee \mathsf{1edge} \vee \mathsf{2Xcoll}),$$
$$\mathsf{fail}^{*2}_{\mathsf{sample}} \Leftrightarrow (\exists\nu \in [\mathsf{u}], \alpha \in \mathcal{Q}^{[\nu]}_{\mathsf{non\text{-}coll}} \text{ s.t. } \mathsf{sample}_\alpha \geq 2) \wedge \neg(\mathsf{hit}_{0,1} \vee \mathsf{2edge} \vee \mathsf{1edge} \vee \mathsf{2Xcoll}).$$

By $\neg$2coll, $\mathcal{Q}^{[\nu]} = \mathcal{Q}^{[\nu]}_{\mathsf{coll}} \cup \mathcal{Q}^{[\nu]}_{\mathsf{non\text{-}coll}}$, and thus we have

$$\begin{aligned}\Pr[\mathsf{fail}^*_{\mathsf{sample}}] &\leq \Pr[(\exists\alpha \in [q] \text{ s.t. } \mathsf{sample}_\alpha \geq 2) \wedge \neg(\mathsf{hit}_{0,1} \vee \mathsf{2edge} \vee \mathsf{1edge} \vee \mathsf{2Xcoll})] \\ &\leq \Pr[\mathsf{fail}^{*1}_{\mathsf{sample}}] + \Pr[\mathsf{fail}^{*2}_{\mathsf{sample}}].\end{aligned}$$

Since $\mathsf{fail}^{*1}_{\mathsf{sample}}$ is the sub-case of 1edge, we have $\Pr[\mathsf{fail}^{*1}_{\mathsf{sample}}] = 0$. We thus derive the bound of $\Pr[\mathsf{fail}^{*2}_{\mathsf{sample}}]$.

Fix $\nu \in [u]$ and $\alpha, \beta_1, \beta_2 \in \mathcal{Q}^{[\nu]}$ such that $\beta_1 \preceq \alpha$ and $\beta_2 \preceq \alpha$. We evaluate the probability that $\mathsf{fail}^{*2}_{\mathsf{sample}}$ occurs at the α-th query with collisions of the β_1-th query and the β_2-th query, where the collision at the first sampling occurs between the α-th and β_1-th queries and the one at the second sampling occurs between the α-th and β_2-th queries such that no edge collision occurs between α and β_1 and between α and β_2. The event is denoted by $\mathbf{E}_2(\alpha, \beta_1, \beta_2)$. The event consists of the following events $\mathbf{E}_{21}(\alpha, \beta_1, \beta_2)$ and $\mathbf{E}_{22}(\alpha, \beta_1, \beta_2)$, i.e., $\mathbf{E}_2(\alpha, \beta_1, \beta_2) = \mathbf{E}_{21}(\alpha, \beta_1, \beta_2) \wedge \mathbf{E}_{22}(\alpha, \beta_1, \beta_2)$. For $s \in [2]$, the event $\mathbf{E}_{2s}(\alpha, \beta_1, \beta_2)$ considers the collision between the α-th and β_s-th queries at the s-th sampling.

- $\mathbf{E}_{2s}(\alpha, \beta_1, \beta_2)$ (s-th sampling):
  - $X^{(\alpha)}_{2,0} = X^{(\beta_s)}_{2,0} \vee$
  - $\exists i \in [2], ((X^{(\alpha)}_{i,j_1}, Y^{(\alpha)}_{i,j_1}), (X^{(\beta_s)}_{i,j_2}, Y^{(\beta_s)}_{i,j_2})) \in \mathsf{UConf}_2(\mathcal{P}^{[\nu]}_i)$ s.t. $(Y^{(\alpha)}_{i,j_1} = Y^{(\beta_s)}_{i,j_2} \wedge (i, j_1, j_2) \neq (2, 0, 0))$.

Fix $s \in [2]$, and $((X^{(\alpha)}_{i,j_1}, Y^{(\alpha)}_{i,j_1}), (X^{(\beta_s)}_{i,j_2}, Y^{(\beta_s)}_{i,j_2})) \in \mathsf{UConf}_2(\mathcal{P}^{[\nu]}_i)$. We consider the following cases.

- Regarding the collision $X^{(\alpha)}_{2,0} = X^{(\beta_s)}_{2,0}$, by the AXU hash $\mathsf{H}^{[\nu]}_1$ or $\mathsf{H}^{[\nu]}_2$ we have

$$\Pr[X^{(\alpha)}_{2,0} = X^{(\beta_s)}_{2,0}] \leq \frac{\omega \max\{\ell_\alpha, \ell_{\beta_s}\}}{2^n}.$$

- Regarding the collision $Y^{(\alpha)}_{i,j_1} = Y^{(\beta_s)}_{i,j_2}$, $(i, j_1) \neq (2, 0)$ must hold. by using the randomness of $Y^{(\alpha)}_{i,j_1}$ (when $j_1 \geq 1$) or $Y^{(\alpha)}_{1,0}$ (when $(i, j_1) = (1, 0)$), we have

$$\Pr[Y^{(\alpha)}_{i,j_1} = Y^{(\beta_s)}_{i,j_2}] \leq \frac{1}{2^n}.$$

Summing the bound over all possible cases, we have

$$\begin{aligned}\Pr[\mathsf{fail}^{*2}_{\mathsf{sample}}] &\le \sum_{\nu} \sum_{\substack{\alpha,\beta_1,\beta_2 \text{ s.t.} \\ \beta_1 \preceq \alpha, \beta_2 \preceq \alpha}} \Bigg(\frac{\omega^2 \max\{\ell_\alpha, \ell_{\beta_1}\} \max\{\ell_\alpha, \ell_{\beta_2}\}}{2^{2n}} \\ &\qquad + \sum_{i\in[2]} \frac{\omega \max\{\ell_\alpha, \ell_{\beta_1}\}}{2^n} \cdot \frac{(r_{\beta_2}+1)}{2^n} \\ &\qquad + \sum_{i\in[2]} \frac{\omega \max\{\ell_\alpha, \ell_{\beta_2}\}}{2^n} \cdot \frac{(r_{\beta_1}+1)}{2^n} \\ &\qquad + \sum_{i\in[2]} \frac{(r_\alpha+1)(r_{\beta_1}+1)}{2^n} \cdot \frac{(r_\alpha+1)(r_{\beta_2}+1)}{2^n} \Bigg) \\ &\le \sum_{\nu} \frac{\omega^2 \ell q_\nu^2 \sigma_\nu + 2\omega q_\nu \sigma_\nu^2 + m_{\mathsf{max}} \sigma_\nu^3}{2^{2n}} \\ &\le \frac{m_{\mathsf{max}} \sigma_{\mathsf{u}}^2 \sigma + 3\omega^2 \ell q_{\mathsf{u}} \min\{q_{\mathsf{u}}\sigma, \sigma_{\mathsf{u}} q\}}{2^{2n}},\end{aligned}$$

providing the following bound.

$$\Pr[\mathsf{fail}^{*}_{\mathsf{sample}}] \le \frac{m_{\mathsf{max}} \sigma_{\mathsf{u}}^2 \sigma + 3\omega^2 \ell q_{\mathsf{u}} \min\{q_{\mathsf{u}}\sigma, \sigma_{\mathsf{u}} q\}}{2^{2n}}.$$

6.7 Evaluation for Good Transcript: Lower-Bounding $\frac{\Pr[\mathsf{T}_R=\tau]}{\Pr[\mathsf{T}_I=\tau]}$

As explained in Section 5.2, our resampling-based proof increases the number of chances of satisfying the target good transcript, i.e., $\Pr[\mathsf{T}_I = \tau]$ is increased, since our algorithm resamples the internal values $Y^{(\alpha)}_{1,0}, X^{(\alpha)}_{2,0}$, and $(Y^{(\alpha)}_{i,j})_{i\in[2],j\in[r_\alpha]}$ if a collision occurs. On the other hand, in the real world, the internal values are defined by RPs $(\dot{\pi}^{\pm}_\nu, \ddot{\pi}^{\pm}_\nu)_{\nu\in[u]}$, and in the ideal world, the internal values are chosen uniformly at random from $\{0,1\}^n$. Hence, the sampling space of the ideal world is larger than that of the real world. Similarly to the evaluation for good transcripts of the security proof of SoP, by using the "budget" of the ideal world, the impact of the increase can be canceled out, and we can prove that there exists a function $g(\tau)$ such that for any good transcript τ,

$$\begin{aligned}&\frac{\Pr[\mathsf{T}_R = \tau]}{\Pr[\mathsf{T}_I = \tau]} \ge 1 - g(\tau) \\ &\text{such that } \mathsf{Ex}(g(\tau)) \le \frac{\sigma_{\mathsf{u}}^2 \sigma}{2^{2n}} + \frac{4 m_{\mathsf{max}} q_{\mathsf{u}} \max\{\sigma_{\mathsf{u}} q, q_{\mathsf{u}} \sigma\}}{2^{2n}}.\end{aligned}$$

The detailed evaluation is given in Section 7.

6.8 Conclusion

Summing the mu-SPRP-advantage, the bound in Eq. (1) for bad transcripts, and the bound $\frac{\sigma_{\mathsf{u}}\sigma^2}{2^{2n}} + \frac{4m_{\mathsf{max}} q_{\mathsf{u}} \max\{\sigma_{\mathsf{u}} q, q_{\mathsf{u}}\sigma\}}{2^{2n}}$ for good transcripts, we obtain the bound in Theorem 1.

7 Proof of Theorem 1: Lower-Bounding $\frac{\Pr[\mathsf{T}_R=\tau]}{\Pr[\mathsf{T}_I=\tau]}$

In this section, we show that there exists a function $g(\tau)$ such that for any good transcript τ,

$$\frac{\Pr[\mathsf{T}_R = \tau]}{\Pr[\mathsf{T}_I = \tau]} \ge 1 - g(\tau) \text{ and } \mathsf{Ex}(g(\tau)) \le \frac{\sigma_{\mathsf{u}}^2 \sigma}{2^{2n}} + \frac{4 m_{\mathsf{max}} q_{\mathsf{u}} \max\{\sigma_{\mathsf{u}} q, q_{\mathsf{u}} \sigma\}}{2^{2n}}.$$

Fix a good transcript τ that consists of the hash keys $\mathcal{H}^\star = (H_1^{[\nu]\star}, H_2^{[\nu]\star})_{\nu\in[u]\}}$, the mask values $\mathcal{L}^\star = (L_1^{[\nu]\star}, L_2^{[\nu]\star})_{\nu\in[u]\}}$, and internal pairs $(\mathcal{P}_1^{[\nu]\star})_{\nu\in[u]}$ and $(\mathcal{P}_2^{[\nu]\star})_{\nu\in[u]}$. We use the symbol "$\star$" for values and sets corresponding to the good transcript τ. For a set $\mathcal{L}$, let $\mathsf{T}_R \vdash \mathcal{L}$ (resp. $\mathsf{T}_I \vdash \mathcal{L}$) be an event that T_R (resp. T_I) satisfies all elements in $\mathcal{L}$.

7.1 Evaluation of $\Pr[\mathsf{T}_W \vdash \mathcal{H}^\star]$ for $W \in \{R, I\}$

In the real world, for each $\nu \in [u]$, the hash keys $H_1^{[\nu]}$ and $H_2^{[\nu]}$ are defined as $H_1^{[\nu]} = \dot{\pi}_\nu(\mathtt{bin}_n(0))$ and $H_2^{[\nu]} = \ddot{\pi}_\nu(\mathtt{bin}_n(0))$. Hence, the hash keys are respectively chosen uniformly at random from $\{0,1\}^n$. In the ideal world, each hash key $H_i^{[\nu]}$ is chosen uniformly at random from $\{0,1\}^n$. We thus have

$$\Pr[\mathsf{T}_R \vdash \mathcal{H}^\star] = \Pr[\mathsf{T}_I \vdash \mathcal{H}^\star].$$

7.2 Evaluation of $\Pr[\mathsf{T}_W \vdash \mathcal{L}^\star \mid \mathsf{T}_W \vdash \mathcal{H}^\star]$ for $W \in \{R, I\}$

In the real world, for each $\nu \in [u]$, the mask values $L_1^{[\nu]}$ and $L_2^{[\nu]}$ are defined as $L_1^{[\nu]} = \dot{\pi}_\nu(\mathtt{bin}_n(1))$ and $L_2^{[\nu]} = \ddot{\pi}_\nu(\mathtt{bin}_n(1))$. Hence, the mask values are respectively chosen uniformly at random from $\{0,1\}^n$ excluding the hash key. In the ideal world, for each $\nu \in [u]$, the mask values $L_1^{[\nu]}$ and $L_2^{[\nu]}$ are respectively chosen uniformly at random from $\{0,1\}^n$. We thus have

$$\Pr[\mathsf{T}_R \vdash \mathcal{L}^\star \mid \mathsf{T}_R \vdash \mathcal{H}^\star] \geq \Pr[\mathsf{T}_I \vdash \mathcal{L}^\star \mid \mathsf{T}_I \vdash \mathcal{H}^\star].$$

7.3 Evaluation of $\Pr[\mathsf{T}_W \vdash (\mathcal{P}_1^{[\nu]\star}, \mathcal{P}_2^{[\nu]\star})_{\nu\in[u]} \mid \mathsf{T}_W \vdash (\mathcal{H}^\star, \mathcal{L}^\star)]$ for $W \in \{R, I\}$

Fix $\nu \in [u]$. We evaluate the real-world and ideal-world probabilities for the ν-th user. Note that by $\neg\mathsf{hit}_{0,1}$, for each $\nu \in [u]$, $(\mathtt{bin}_n(0), H_1^{[\nu]})$ and $(\mathtt{bin}_n(1), L_1)$ are not in $\mathcal{P}_1^{[\nu]\star}$, and $(\mathtt{bin}_n(0), H_2^{[\nu]})$ and $(\mathtt{bin}_n(1), L_2)$ are not in $\mathcal{P}_2^{[\nu]\star}$.

In the ideal world, if $\alpha \in \mathcal{Q}_{\mathsf{coll}}^{[\nu]}$, then each value for the α-th query is sampled once. On the other hand, if $\alpha \in \mathcal{Q}_{\mathsf{non\text{-}coll}}^{[\nu]}$, then each value of the α-th query is sampled at most twice. Hence, the probability that sampled values satisfy $\mathcal{P}_1^{(\alpha)\star}$ and $\mathcal{P}_2^{(\alpha)\star}$ for $\alpha \in \mathcal{Q}_{\mathsf{non\text{-}coll}}^{[\nu]}$ is greater than the probability for $\alpha \in \mathcal{Q}_{\mathsf{coll}}^{[\nu]}$. We thus assume that for each $\nu \in [u]$, $\mathcal{Q}_{\mathsf{coll}}^{[\nu]} = \emptyset$ and $\mathcal{Q}_{\mathsf{non\text{-}coll}}^{[\nu]} = \mathcal{Q}^{[\nu]}$.

For $\nu \in [u]$, $v \in [q_\nu]$, and $i \in [2]$, let

$$z_i^{(v)} \coloneqq \Big|\{(X_{i,j}^{(\mathrm{a}_w)\star}, Y_{i,j}^{(\mathrm{a}_w)\star}) \mid (X_{i,j}^{(\mathrm{a}_v)\star}, Y_{i,j}^{(\mathrm{a}_v)\star}) \in \mathcal{P}_i^{(\mathrm{a}_v)\star} \wedge \forall w \in [v-1] : (X_{i,j}^{(\mathrm{a}_v)\star}, Y_{i,j}^{(\mathrm{a}_v)\star}) \notin \mathcal{P}_i^{(\mathrm{a}_w)\star}]\}\Big|$$

be the number of pairs of $\dot{\pi}_\nu$ (if $i = 1$); $\ddot{\pi}_\nu$ (if $i = 2$) defined at the a_v-th query, and

$$z_i^{(<v)} \coloneqq \sum_{w\in[v-1]} z_i^{(w)}$$

be the number of internal pairs of $\dot{\pi}_\nu$ (if $i = 1$); $\ddot{\pi}_\nu$ (if $i = 2$) defined by queries to the ν-th user before the a_v-th query (in the ideal world). Note that the pairs for hash keys and mask values are not included. Let

$$\mathcal{T}^{(<v)} \coloneqq \left\{w \in [v-1] \mid T^{(\mathrm{a}_w)} = T^{(\mathrm{a}_v)}\right\} \text{ and } t^{(<v)} \coloneqq \left|\mathcal{T}^{(<v)}\right|.$$

Real-World Probability. In the real world, by $\neg\mathsf{fail}_{\mathsf{sample}}$, for each $\nu \in [u]$, we have

$$\begin{aligned}&\Pr[\mathsf{T}_R \vdash (\mathcal{P}_1^{[\nu]\star}, \mathcal{P}_2^{[\nu]\star})]\\ &= \prod_{v\in[q_\nu]} \left(\left(\prod_{i\in[z_1^{(v)}]} \frac{1}{2^n - z_1^{(<v)} - i - 1} \right) \left(\prod_{i\in[z_2^{(v)}]} \frac{1}{2^n - z_2^{(<v)} - i - 1} \right) \right).\end{aligned}$$

Ideal-World Probability. For $v \in [q_\nu]$, if the a_v-th query is an encryption (resp. a decryption) one, then $C^{(\mathrm{a}_v)}$ (resp. $M^{(\mathrm{a}_v)}$) is chosen uniformly at random from at least $2^{m_{\mathrm{a}_v}} - t^{(<v)}$ elements of m_{a_v} bits. The remaining $m_{\mathrm{a}_v}^+$-bit value $C_+^{(\mathrm{a}_v)}$ or $M_+^{(\mathrm{a}_v)}$ is chosen uniformly at random from $\{0,1\}^{m_{\mathrm{a}_v}^+}$. For each $v \in [q_\nu]$, we have

$$\begin{aligned}&\Pr[\mathsf{T}_I \vdash \{(T^{(\mathrm{a}_v)\star}, \hat{M}^{(\mathrm{a}_v)\star}, \hat{C}^{(\mathrm{a}_v)\star}) \mid v \in [q_\nu]\}]\\ &\le \prod_{v\in[q_\nu]} \frac{1}{2^{m_{\mathrm{a}_v}^+}(2^{m_{\mathrm{a}_v}} - t^{(<v)})} = \frac{1}{2^{(r_{\mathrm{a}_v}+1)n} - t^{(<v)} 2^{m_{\mathrm{a}_v}^+}}.\end{aligned}$$

We assume that $\mathsf{T}_I \vdash \{(T^{(\mathrm{a}_v)\star}, \hat{M}^{(\mathrm{a}_v)\star}, \hat{C}^{(\mathrm{a}_v)\star}) \mid v \in [q_\nu]\}$.

Next, for each $v \in [q_\nu]$, we evaluate $\Pr[\forall i \in [0, r_\alpha] : Y_{1,i}^{(\mathrm{a}_v)} = Y_{1,i}^{(\mathrm{a}_v)\star}]$, the probability that the remaining values satisfy the target values in τ. Note that the assumption $\mathsf{T}_I \vdash \{(T^{(\mathrm{a}_v)\star}, \hat{M}^{(\mathrm{a}_v)\star}, \hat{C}^{(\mathrm{a}_v)\star})$ ensures that $\forall i \in [0, r_{\mathrm{a}_v}] : Y_{1,i}^{(\mathrm{a}_v)} = Y_{1,i}^{(\mathrm{a}_v)\star} \Rightarrow \forall i \in [0, r_{\mathrm{a}_v}] : Y_{2,i}^{(\mathrm{a}_v)} = Y_{2,i}^{(\mathrm{a}_v)\star}$, and the number of the remaining new ciphertext blocks $Y_{1,i}^{(\mathrm{a}_v)}$ is $z_1^{(v)} + z_2^{(v)} - (r_{\mathrm{a}_v} + 1)$. For $Y_{1,j}^{(\mathrm{a}_v)}$, if the input is new, then $Y_{1,j}^{(\mathrm{a}_v)}$ is chosen uniformly at random from $\{0,1\}^n$ and we have $\Pr[Y_{1,j}^{(\mathrm{a}_v)} = Y_{1,j}^{(\mathrm{a}_v)\star}] \le \frac{1}{2^n}$. Since each fresh value is sampled at most twice, we consider the following two cases.

- **Case 1**: The first sampling of the a_v-th query succeeds.
  The probability that $\forall i \in [0, r_{\mathrm{a}_v}] : Y_{1,i}^{(\mathrm{a}_v)} = Y_{1,i}^{(\mathrm{a}_v)\star}$ is at most

$$\left(\frac{1}{2^n}\right)^{z_1^{(v)} + z_2^{(v)} - (r_{\mathrm{a}_v}+1)}.$$

- **Case 2**: The first sampling of the a_v-th query fails and the second sampling succeeds. The second sampling is performed if a collision with $Y_{1,0}^{(\mathrm{a}_v)}, X_{2,0}^{(\mathrm{a}_v)}$ or $Y_{i,j}^{(\mathrm{a}_v)}$ ($(i,j) \in [2] \times [r_{\mathrm{a}_v}]$) occurs. We evaluate the collision probabilities below.
  - We evaluate the probability that a collision with $Y_{i,j}^{(\mathrm{a}_v)}$ occurs such that $(i,j) \in [2] \times [r_{\mathrm{a}_v}]$, i.e.,

$$p_1 \coloneqq \Pr\left[\exists (i,j) \in [2] \times [r_{\mathrm{a}_v}] \text{ s.t. } Y_{i,j}^{(\mathrm{a}_v)} \in \mathcal{Y}_i^{[\nu]} \cup \{Y_{i,j'}^{(\mathrm{a}_v)} \mid j' \in [0, j-1]\})\right],$$

    where $\mathcal{Y}_i^{[\nu]}$ is the set of ciphertext blocks of the ν-th user defined before the sampling of the a_v-th query. For each $Y_{i,j'}^{(\beta)}$ that is defined before $Y_{i,j}^{(\mathrm{a}_v)}$ is defined, since $Y_{i,j}^{(\mathrm{a}_v)}$ is chosen uniformly at random from $\{0,1\}^n$, we have $\Pr[Y_{i,j}^{(\mathrm{a}_v)} = Y_{i,j'}^{(\beta)}] \le \frac{1}{2^n}$ and

$$p_1 \le \sum_{i\in[2]} \frac{z_i^{(<v)} r_{\mathrm{a}_v} + \binom{r_{\mathrm{a}_v}}{2}}{2^n}.$$

- We evaluate the probability that a collision with $Y_{1,0}^{(\mathrm{a}_v)}$ occurs, i.e.,

$$p_2 \coloneqq \Pr\left[Y_{1,0}^{(\mathrm{a}_v)} \in \mathcal{Y}_1^{[\nu]}\right].$$

  Since $Y_{1,0}^{(\mathrm{a}_v)}$ is chosen uniformly at random from $\{0,1\}^n$, we have

$$p_2 \leq \frac{z_1^{(<v)}}{2^n}.$$

- We evaluate the probability that a collision with $X_{2,0}^{(\mathrm{a}_v)}$ occurs and no collision with $Y_{1,0}^{(\mathrm{a}_v)}$ occurs, i.e.,

$$p_3 \coloneqq \Pr\left[X_{2,0}^{(\mathrm{a}_v)} \in \mathcal{X}_2^{[\nu]} \wedge Y_{1,0}^{(\mathrm{a}_v)} \notin \mathcal{Y}_1^{[\nu]}\right].$$

  We consider the following cases.
  * For $X_{2,j}^{(\beta)} \in \mathcal{X}_2^{[\nu]}$ such that $j \geq 1$ or $T^{(\mathrm{a}_v)} \neq T^{(\beta)}$, since $X_{2,0}^{(\mathrm{a}_v)}$ is chosen uniformly at random from $\{0,1\}^n$, we have

$$\Pr\left[X_{2,0}^{(\mathrm{a}_v)} = X_{2,j}^{(\beta)} \mid j \geq 1 \vee T^{(\mathrm{a}_v)} \neq T^{(\beta)}\right] \leq \frac{1}{2^n}.$$

  * For $X_{2,j}^{(\beta)} \in \mathcal{X}_2^{[\nu]}$ such that $j = 0$ and $T^{(\alpha)} = T^{(\beta)}$, the collision has the following structure:

$$\begin{aligned} X_{2,0}^{(\mathrm{a}_v)} = X_{2,0}^{(\beta)} \Leftrightarrow\ & Y_{1,0}^{(\mathrm{a}_v)} \oplus \mathsf{H}_1^{[\nu]}(T^{(\mathrm{a}_v)}, R^{(\mathrm{a}_v)}) \oplus \mathsf{H}_2^{[\nu]}(T^{(\mathrm{a}_v)}, R^{(\mathrm{a}_v)}) \\ & = Y_{1,0}^{(\beta)} \oplus \mathsf{H}_1^{[\nu]}(T^{(\beta)}, R^{(\beta)}) \oplus \mathsf{H}_2^{[\nu]}(T^{(\beta)}, R^{(\beta)}). \end{aligned}$$

    If no collision with $Y_{1,0}^{(\mathrm{a}_v)}$ occurs, then $Y_{1,0}^{(\mathrm{a}_v)} \neq Y_{1,0}^{(\beta)}$ must hold. On the other hand, $X_{2,0}^{(\mathrm{a}_v)} = X_{2,0}^{(\beta)} \wedge R^{(\mathrm{a}_v)} = R^{(\beta)} \wedge T^{(\mathrm{a}_v)} = T^{(\beta)} \Rightarrow Y_{1,0}^{(\mathrm{a}_v)} = Y_{1,0}^{(\beta)}$. Hence, $R^{(\mathrm{a}_v)} \neq R^{(\beta)}$ must hold. Since $|Y_{1,0}^{(\mathrm{a}_v)}| + |R^{(\mathrm{a}_v)}| = (r_{\mathrm{a}_v}+1)n - m_{\mathrm{a}_v}^+$ and there is at least one choice for $Y_{1,0}^{(\mathrm{a}_v)} = Y_{1,0}^{(\beta)}$ in the $2^{(r_{\mathrm{a}_v}+1)n-m_{\mathrm{a}_v}^+}$ pairs of $(Y_{1,0}^{(\mathrm{a}_v)}, R^{(\mathrm{a}_v)})$, we have

$$\Pr[X_{2,0}^{(\mathrm{a}_v)} = X_{2,0}^{(\beta)} \mid T^{(\mathrm{a}_v)} = T^{(\beta)}] \leq \frac{2^{r_{\mathrm{a}_v}n-m_{\mathrm{a}_v}^+} - 1}{2^{(r_{\mathrm{a}_v}+1)n-m_{\mathrm{a}_v}^+}}.$$

  By using the above bounds, we have the following bound:

$$\begin{aligned} p_3 &\leq \frac{z_2^{(<v)} - t^{(<v)}}{2^n} + \frac{t^{(<v)}(2^{r_{\mathrm{a}_v}n-m_{\mathrm{a}_v}^+} - 1)}{2^{(r_{\mathrm{a}_v}+1)n-m_{\mathrm{a}_v}^+}} \\ &= \frac{z_2^{(<v)} - \frac{t^{(<v)}}{2^{r_{\mathrm{a}_v}n-m_{\mathrm{a}_v}^+}}}{2^n}. \end{aligned}$$

Hence, the probability that the first sampling of the a_v-th query fails and the second sampling succeeds: $\forall i \in [0, r_{\mathrm{a}_v}] : Y_{1,i}^{(\mathrm{a}_v)} = Y_{1,i}^{(\mathrm{a}_v)\star}$ is at most

$$\begin{aligned} & \left(\frac{1}{2^n}\right)^{z_1^{(v)}+z_2^{(v)}-(r_{\mathrm{a}_v}+1)} \cdot (p_1 + p_2 + p_3) \\ & \leq \left(\frac{1}{2^n}\right)^{z_1^{(v)}+z_2^{(v)}-(r_{\mathrm{a}_v}+1)} \cdot \left(\left(\sum_{i\in[2]} \frac{z_i^{(<v)}(r_{\mathrm{a}_v}+1) + \binom{r_{\mathrm{a}_v}}{2}}{2^n}\right) - \frac{\frac{t^{(<v)}}{2^{r_{\mathrm{a}_v}n-m_{\mathrm{a}_v}^+}}}{2^n}\right) \end{aligned}$$

By the above bounds, for each $v \in [q_\nu]$, we have the following bound.

$$\Pr[\mathsf{T}_I \vdash (\mathcal{P}_1^{[\nu]\star}, \mathcal{P}_2^{[\nu]\star})] \leq \prod_{v\in[q_\nu]} \frac{1}{2^{(r_{\mathrm{a}_v}+1)n} - t^{(<v)}2^{m^+_{\mathrm{a}_v}}} \cdot \left(\frac{1}{2^n}\right)^{z_1^{(v)}+z_2^{(v)}-(r_{\mathrm{a}_v}+1)} \times \left(1 + \left(\left(\sum_{i\in[2]} \frac{z_i^{(<v)}(r_{\mathrm{a}_v}+1) + \binom{r_{\mathrm{a}_v}}{2}}{2^n}\right) - \frac{\frac{t^{(<v)}}{2^{r_{\mathrm{a}_v}n-m^+_{\mathrm{a}_v}}}}{2^n}\right)\right).$$

Deriving the Lower Bound. By using the above bounds, we have the following bound:

$$\begin{aligned}
\frac{\Pr[\mathsf{T}_R = \tau]}{\Pr[\mathsf{T}_I = \tau]} &= \frac{\Pr[\mathsf{T}_R \vdash (\mathcal{P}_1^{[\nu]\star}, \mathcal{P}_2^{[\nu]\star})_{\nu\in[u]}]}{\Pr[\mathsf{T}_I \vdash (\mathcal{P}_1^{[\nu]\star}, \mathcal{P}_2^{[\nu]\star})_{\nu\in[u]}]} \\
&\geq \prod_{\nu\in[u]}\prod_{v\in[q_\nu]} \left(\left(\prod_{i\in[z_1^{(v)}]} \frac{1}{2^n - z_1^{(<v)} - i - 1}\right)\left(\prod_{i\in[z_2^{(v)}]} \frac{1}{2^n - z_2^{(<v)} - i - 1}\right)\right. \\
&\quad \times \left(\frac{1}{2^{(r_{\mathrm{a}_v}+1)n} - t^{(<v)}2^{m^+_{\mathrm{a}_v}}} \cdot \left(\frac{1}{2^n}\right)^{z_1^{(v)}+z_2^{(v)}-(r_{\mathrm{a}_v}+1)}\right)^{-1} \\
&\qquad \left.\times \left(1 + \left(\sum_{i\in[2]} \frac{z_i^{(<v)}(r_{\mathrm{a}_v}+1) + \binom{r_{\mathrm{a}_v}}{2}}{2^n}\right) - \frac{\frac{t^{(<v)}}{2^{r_{\mathrm{a}_v}n-m^+_{\mathrm{a}_v}}}}{2^n}\right)^{-1}\right) \\
&= \prod_{\nu\in[u]}\prod_{v\in[q_\nu]} \left(\left(\prod_{i\in[z_1^{(v)}]} \left(1 - \frac{z_1^{(<v)} + i + 1}{2^n}\right)\right)\left(\prod_{i\in[z_2^{(v)}]} \left(1 - \frac{z_2^{(<v)} + i + 1}{2^n}\right)\right)\right. \\
&\quad \left.\times \underbrace{\left(1 + \frac{t^{(<v)}2^{m^+_{\mathrm{a}_v}}}{2^{n(r_{\mathrm{a}_v}+1)} - t^{(<v)}2^{m^+_{\mathrm{a}_v}}}\right)\left(1 + \left(\sum_{i\in[2]} \frac{z_i^{(<v)}(r_{\mathrm{a}_v}+1) + \binom{r_{\mathrm{a}_v}}{2}}{2^n}\right) - \frac{\frac{t^{(<v)}}{2^{r_{\mathrm{a}_v}n-m^+_{\mathrm{a}_v}}}}{2^n}\right)}_{E}\right)^{-1},
\end{aligned}$$

where

$$\begin{aligned}
E &= 1 + \left(\sum_{i\in[2]} \frac{z_i^{(<v)}(r_{\mathrm{a}_v}+1) + \binom{r_{\mathrm{a}_v}}{2}}{2^n}\right) - \frac{t^{(<v)}}{2^{n(r_{\mathrm{a}_v}+1)-m^+_{\mathrm{a}_v}}} \\
&\quad + \frac{t^{(<v)}2^{m^+_{\mathrm{a}_v}}}{2^{n(r_{\mathrm{a}_v}+1)} - t^{(<v)}2^{m^+_{\mathrm{a}_v}}} + \frac{t^{(<v)}2^{m^+_{\mathrm{a}_v}}}{2^{n(r_{\mathrm{a}_v}+1)} - t^{(<v)}2^{m^+_{\mathrm{a}_v}}} \cdot \left(\sum_{i\in[2]} \frac{z_i^{(<v)}(r_{\mathrm{a}_v}+1) + \binom{r_{\mathrm{a}_v}}{2}}{2^n}\right) \\
&\quad - \frac{t^{(<v)}}{2^{n(r_{\mathrm{a}_v}+1)-m^+_{\mathrm{a}_v}}} \cdot \frac{t^{(<v)}2^{m^+_{\mathrm{a}_v}}}{2^{n(r_{\mathrm{a}_v}+1)} - t^{(<v)}2^{m^+_{\mathrm{a}_v}}}. \\
&\leq 1 + \sum_{i\in[2]} \left(\frac{z_i^{(<v)}(r_{\mathrm{a}_v}+1) + \binom{r_{\mathrm{a}_v}}{2}}{2^n} + \frac{2t^{(<v)}}{2^{n(r_{\mathrm{a}_v}+1)-m^+_{\mathrm{a}_v}}} \cdot \frac{z_i^{(<v)}(r_{\mathrm{a}_v}+1) + \binom{r_{\mathrm{a}_v}}{2}}{2^n}\right).
\end{aligned}$$

Let $\bar{z}_i^{(v)} = r_{\mathrm{a}_v} - z_i^{(v)} + 1$. By using the relation: $(1-a)(1+b) \le 1-a+b$ for any $a, b \ge 0$, we have the following bound:

$$\begin{aligned}
\frac{\Pr[\mathsf{T}_R = \tau]}{\Pr[\mathsf{T}_I = \tau]} &\ge \prod_{\nu\in[u]}\prod_{v\in[q_\nu]}\left(1 - \sum_{i\in[z_1^{(v)}]}\frac{z_1^{(<v)}+i+1}{2^n} - \sum_{i\in[z_2^{(v)}]}\frac{z_2^{(<v)}+i+1}{2^n}\right.\\
&\qquad\left. + \sum_{i\in[2]}\left(\frac{z_i^{(<v)}(r_{\mathrm{a}_v}+1)+\binom{r_{\mathrm{a}_v}}{2}}{2^n} + \frac{2t^{(<v)}}{2^{n(r_{\mathrm{a}_v}+1)-m^+_{\mathrm{a}_v}}}\cdot\frac{z_i^{(<v)}(r_{\mathrm{a}_v}+1)+\binom{r_{\mathrm{a}_v}}{2}}{2^n}\right)\right)^{-1}\\
&\ge \prod_{\nu\in[u]}\prod_{v\in[q_\nu]}\left(1 - \sum_{i\in[2]}\left(\frac{z_i^{(<v)}\bar{z}_i^{(v)} + \bar{z}_i^{(v)}r_{\mathrm{a}_v}}{2^n}\right.\right.\\
&\qquad\left.\left. + \frac{2t^{(<v)}}{2^{n(r_{\mathrm{a}_v}+1)-m^+_{\mathrm{a}_v}}}\cdot\frac{z_i^{(<v)}(r_{\mathrm{a}_v}+1)+0.5r_{\mathrm{a}_v}^2}{2^n}\right)\right)\\
&\ge 1 - \underbrace{\underbrace{\sum_{\nu\in[u]}\sum_{v\in[q_\nu]}\sum_{i\in[2]}\left(\frac{z_i^{(<v)}\bar{z}_i^{(v)} + \bar{z}_i^{(v)}r_{\mathrm{a}_v}}{2^n} + \frac{2q_\nu}{2^n}\cdot\frac{z_i^{(<v)}(r_{\mathrm{a}_v}+1)+0.5r_{\mathrm{a}_v}^2}{2^n}\right)}_{=:f(\tau)}}_{=:g(\tau)},
\end{aligned}$$

where

$$\begin{aligned}
g(\tau) &\le f(\tau) + \sum_{\nu\in[u]}\frac{2m_{\mathsf{max}}q_\nu\sigma_\nu^2 + 2\sigma_\nu^2}{2^{2n}}\\
&\le f(\tau) + \frac{4m_{\mathsf{max}}\max\{\sigma_{\mathsf{u}}q, q_{\mathsf{u}}\sigma\}}{2^{2n}}\\
&\le f(\tau) + \frac{4m_{\mathsf{max}}q_{\mathsf{u}}\max\{\sigma_{\mathsf{u}}q, q_{\mathsf{u}}\sigma\}}{2^{2n}}.
\end{aligned}$$

Regarding $\mathsf{Ex}(f(\tau))$, we bound $\mathsf{Ex}(\bar{z}_i^{(v)})$ as follows. $\bar{z}_i^{(v)}$ is the number of non-fresh plaintext blocks of the right part at the i-th round of the a_v-th query. Since $Y_{1,0}^{(\alpha)}$ is chosen uniformly at random from $\{0,1\}^n$, we have

$$\begin{aligned}
\mathsf{Ex}(\bar{z}_i^{(v)}) &\le \frac{\mathsf{Ex}(z_i^{(<v)})r_{\mathrm{a}_v}}{2^n}\\
&\le \frac{(\sum_{w\in[v-1]}(r_{\mathrm{a}_w}+1))r_{\mathrm{a}_v}}{2^n}.
\end{aligned}$$

Let $\ell_{<v} = \sum_{w\in[v-1]}(r_{\mathrm{a}_w}+1)$. By using the bound, we have

$$\begin{aligned}
\mathsf{Ex}(f(\tau)) &\le \sum_{\nu\in[u]}\sum_{v\in[q_\nu]}\sum_{i\in[2]}\frac{\ell_{<v}^2 r_{\mathrm{a}_v} + \ell_{<v}r_{\mathrm{a}_v}^2}{2^{2n}}\\
&\le \sum_{\nu\in[u]}\frac{0.5\sigma_\nu^3 + 0.5\sigma_\nu^3}{2^{2n}}\\
&\le \frac{\sigma_{\mathsf{u}}^2\sigma}{2^{2n}}
\end{aligned}$$

Hence, there exists a function $g(\tau)$ such that for any good transcript τ,

$$\frac{\Pr[\mathsf{T}_R = \tau]}{\Pr[\mathsf{T}_I = \tau]} \ge 1 - g(\tau) \text{ s.t. } \mathsf{Ex}(g(\tau)) \le \frac{\sigma_{\mathsf{u}}^2\sigma}{2^{2n}} + \frac{4m_{\mathsf{max}}q_{\mathsf{u}}\max\{\sigma_{\mathsf{u}}q, q_{\mathsf{u}}\sigma\}}{2^{2n}}.$$

8 Security of HCTR2 and HCTR2-TwKD

For the evaluations, a keyed hash function is decomposed into poly_H and pad defined as follows.

- $\{\mathsf{poly}_H\}_{H\in\mathcal{K}_\mathsf{H}}$ is a family of keyed hash functions with the key space $\mathcal{K}_\mathsf{H}$ such that $\mathsf{poly}_H : \{0,1\}^{n*} \to \{0,1\}^n$ accepts an input of length multiple of n and returns an n-bit hash value.
- $\mathsf{pad} : \mathcal{T} \times \mathcal{M} \to \{0,1\}^{n*}$ is an injective padding function such that $\forall (T, D) \in \mathcal{T} \times \mathcal{M} : (1 \le |\mathsf{pad}(T, D)|_n \le |T|_n + |D|_n + 1) \wedge (\mathsf{pad}(T, D) \notin \cup_{i\in\mathbf{N}}\{0^{in}\})$.

Then, H_H is defined as follows: for a key $H \in \mathcal{K}_\mathsf{H}$ and an input pair $(T, D) \in \mathcal{T} \times \mathcal{M}$, $\mathsf{H}_H(T, D) \coloneqq \mathsf{poly}_H(\mathsf{pad}(T, D))$.

We assume that pad has the following structures that captures the feature of the hash function of HCTR2. Let $\mathsf{zp}_n : \{0,1\}^* \to \{0,1\}^{n*}$ be a padding function such that for an input $D \in \{0,1\}^*$, the output is defined as $\mathsf{zp}_n \coloneqq D\|0^{n|D|_n-|D|}$. Let $\mathsf{ozp}_n : \{0,1\}^* \to \{0,1\}^{n*}$ be a padding function such that for an input $D \in \{0,1\}^*$, the output is defined as $\mathsf{ozp}_n(D) \coloneqq D\|10^{n|D\|1|_n-|D|-1}$. Let $\mathsf{pad}_\mathsf{pf} : [0, 2^{n-3} - 1] \times \{0,1\} \to \{0,1\}^n\backslash\{0^n\}$ be an injective function.

- pad is an injective padding function such that for an input pair $(T, D) \in \mathcal{T} \times \mathcal{M}$, the output is defined as
  - if n **divides** $|D|$ then $\mathsf{pad}(T, D) \coloneqq (\mathsf{pad}_\mathsf{pf}(|T|, 1)\|\mathsf{zp}_n(T)\|D)$, and
  - otherwise $\mathsf{pad}(T, D) \coloneqq (\mathsf{pad}_\mathsf{pf}(|T|, 0)\|\mathsf{zp}_n(T)\|\mathsf{ozp}_n(D)$.

In this proof, we use the following property as well as AXU and regular.

Definition 3 (Almost 2XOR Universal). *$\{\mathsf{H}_H\}_{H\in\mathcal{K}_\mathsf{H}}$ is ω-A2XU if for any $Y \in \{0,1\}^n$, pairs (T, M), $(T', M') \in \mathcal{T} \times \mathcal{M}$, and $C \in \{0,1\}^{|M|}$ such that $\mathsf{lsb}_{|\mathsf{pad}(T',M')|}(\mathsf{pad}(T, M) \oplus \mathsf{pad}(T, C)) \neq \mathsf{pad}(T', M')$,*

$$\Pr[H \xleftarrow{\$} \mathcal{K}_\mathsf{H}, \mathsf{H}_H(T, M) \oplus \mathsf{H}_H(T, C) \oplus \mathsf{H}_H(T', M') = Y] \le \frac{\omega\,(\max\{|T|_n + |M|_n, |T'|_n + |M'|_n\} + 1)}{2^n}.$$

The POLYVAL hash function used in HCTR2 is 1-A2XU.

The following theorem shows the mu-STPRP-security bound of HCTR2.

Theorem 2 (Security of HCTR2). *Assume that $\{\mathsf{H}_H\}_{H\in\mathcal{K}_\mathsf{H}}$ is ω-regular, ω-AXU, and ω-A2XU. For any adversary $\mathbf{A}$ running in time t and making q queries to u users such that the total number of n-bit data blocks in all queries is σ and the maximum number of data blocks per user is σ_u, we have*

$$\mathbf{Adv}^{\mathsf{mu\text{-}vil\text{-}stprp}}_{\mathsf{HCTR2}}(\mathbf{A}) \le \frac{4\omega\sigma_\mathsf{u}\sigma}{2^n} + \frac{4n\sigma}{2^n} + \mathbf{Adv}^{\mathsf{mu\text{-}sprp}}_{E}(u, \sigma, t + O(\sigma)).$$

The proof is given in Section 9. An overview of the proof is given after the following theorem.

Regarding HCTR2-TwKD, we use mu-PRF-secure functions for TwKD. Then, for each user $\nu \in [u]$, TwKD with a fresh tweak generates a fresh key of HCTR2. Hence, the maximum number of data blocks per tweak σ_tw becomes the maximum number of data blocks per key of HCTR2. Replacing σ_u in the above bound with σ_tw, we obtain the following bound of HCTR2-TwKD.

Theorem 3 (Security of HCTR2-TwKD). *Assume that* $\{\mathsf{H}_H\}_{H\in\mathcal{K}_\mathsf{H}}$ *is* ω*-regular,* ω*-AXU, and* ω*-A2XU. For any adversary* $\mathbf{A}$ *running in time* t *and making* q *queries to* u *users such that the total number of* n*-bit data blocks in all queries is* σ*, the maximum number of* n*-bit data blocks per pair of user and tweak* T_0 *is* σ_{tw}*, and the total number of pairs of user and tweak* T_0 *is at most* N_{tw}*, we have*

$$\mathbf{Adv}^{\mathsf{mu\text{-}vil\text{-}stprp}}_{\mathsf{HCTR2\text{-}TwKD}}(\mathbf{A}) \le \frac{4\omega\sigma_{\mathsf{tw}}\sigma}{2^n} + \frac{4n\sigma}{2^n} + \mathbf{Adv}^{\mathsf{mu\text{-}sprp}}_{E}(N_{tw}, \sigma, t + O(\sigma)) + \mathbf{Adv}^{\mathsf{mu\text{-}prf}}_{F}(u, q, t + O(\sigma)).$$

Overview of the Proof of Theorem 2. The bound in Theorem 2 can be obtained by simplifying the proof of CHCTR2. The proof of CHCTR2 considers two collisions of internal plaintext blocks or ciphertext blocks for BBB security. On the other hand, for the proof of HCTR2, we can consider only a single collision within the same user for birthday-bound security. By the birthday analysis with ω-regular, ω-AXU, and ω-A2XU, for each user $\nu \in [u]$, the collision probability is at most $\frac{4\omega\sigma_\nu^2}{2^n} + \frac{4n\sigma_\nu}{2^n}$ where σ_ν is the number of n-bit data blocks in queries to the ν-th user. Summing the bound for each user, we obtain the bound in Theorem 2.

9 Proof of Theorem 2

Without loss of generality, we assume that the adversary is deterministic and makes no repeated or trivial query.

We first replace the BCs $(E^{\pm}_{K^{[\nu]}})_{\nu\in[u]}$ with RPs $(\pi^{\pm}_\nu)_{\nu\in[u]}$ where $\forall\nu \in [u] : \pi_\nu \xleftarrow{\$} \mathsf{Perm}(n)$ and $\pi^{\pm} := (\pi_\nu, \pi_\nu^{-1})$, thereby introducing the advantage function $\mathbf{Adv}^{\mathsf{mu\text{-}sprp}}_{E}(u, \sigma, t+O(\sigma))$ in the bound. Hence, the following evaluation considers the oracles $\mathcal{O}_{\mathsf{real}} = (\mathsf{HCTR2}[\pi_\nu, \mathsf{H}], \mathsf{HCTR2}^{-1}[\pi^{\pm}_\nu, \mathsf{H}])_{\nu\in[u]}$ as the real-world ones.

9.1 Definitions

Let q_ν be the number of queries to the ν-th user. Without loss of generality, we assume that $\forall\nu \in [u] : q_\nu \ge 1$. Let σ_ν be the number of n-bit data blocks in queries to the ν-th user. Note that $\sigma = \sum_{\nu\in[u]} \sigma_\nu$. We call the stage that an adversary makes queries the "query stage" and the stage after the query stage the "decision stage".

For $\alpha \in [q]$, values corresponding to the α-th query are denoted by using the superscript symbol (α), e.g., $M^{(\alpha)}$, $T^{(\alpha)}$, $X_i^{(\alpha)}$, $Y_i^{(\alpha)}$, etc. Let $t_\alpha := |T^{(\alpha)}|_n$. Let $m_\alpha := |M^{(\alpha)}|$ be the length of the plaintext $M^{(\alpha)}$. Let $r_\alpha = \lceil |M_*^{(\alpha)}|/n \rceil$ be the block length of the right part of the plaintext $M^{(\alpha)}$. Let $m_\alpha^+ := n(r_\alpha + 1) - m_\alpha$ be the number of truncated bits in CTR for the α-th query and $\ell_\alpha := t_\alpha + r_\alpha + 1$ the total number of data blocks in the α-th query. Let $r_{\mathsf{max}} := \max\{r_\alpha \mid \alpha \in [q]\}$. For the plaintext $M_*^{(\alpha)}$ (resp. ciphertext $C_*^{(\alpha)}$) of the right part, let $M_i^{(\alpha)}$ (resp. $C_i^{(\alpha)}$) be the i-th n-bit plaintext (resp. ciphertext) block, i.e., $M_*^{(\alpha)} = M_1^{(\alpha)} \| \cdots \| M_{r_\alpha}^{(\alpha)}$ and $C_*^{(\alpha)} = C_1^{(\alpha)} \| \cdots \| C_{r_\alpha}^{(\alpha)}$. Note that the length of the last block is $\le n$. Let $\mathsf{u}_\alpha \in [u]$ be the user index corresponding to the α-th query, i.e., if the query is to the ν-th user, then $\mathsf{u}_\alpha = \nu$. For $\nu \in [u]$, let $\mathcal{Q}^{[\nu]} := \{\alpha \in [q] \mid \mathsf{u}_\alpha = \nu\}$ be query indexes for the ν-th user. For $\alpha \in [q]$, let $\mathsf{query}^{(\alpha)}$ be a predicate for the α-th query: if the α-th query is an encryption one, then $\mathsf{query}^{(\alpha)} := \mathsf{enc}$, and else $\mathsf{query}^{(\alpha)} := \mathsf{dec}$. For $\nu \in [u]$, let $\mathcal{P}^{[\nu]} := \{(X_i^{(\alpha)}, Y_i^{(\alpha)}) \mid \alpha \in \mathcal{Q}^{[\nu]}, i \in [0, r_\alpha]\}$ be the set of all input-output tuples of $\pi^{\pm}_\nu$ for the ν-th user.

Algorithm 5 Sampling Algorithm for HCTR2

1: **for** $\nu \in [u]$; $\alpha \in \mathcal{Q}^{[\nu]}$ **do**
2: $Z_1^{(\alpha)} \leftarrow \mathsf{H}_{H^{[\nu]}}(T^{(\alpha)}, M_*^{(\alpha)})$; $Z_2^{(\alpha)} \leftarrow \mathsf{H}_{H^{[\nu]}}(T^{(\alpha)}, C_*^{(\alpha)})$
3: $X_0^{(\alpha)} \leftarrow Z_1^{(\alpha)} \oplus M_0^{(\alpha)}$; $Y_0^{(\alpha)} \leftarrow Z_2^{(\alpha)} \oplus C_0^{(\alpha)}$; $IV^{(\alpha)} \leftarrow X_0^{(\alpha)} \oplus Y_0^{(\alpha)}$
4: **for** $j \in [r_\alpha - 1]$ **do** $X_j^{(\alpha)} \leftarrow \mathsf{add}(IV^{(\alpha)}, j)$; $Y_j^{(\alpha)} \leftarrow M_j^{(\alpha)} \oplus C_j^{(\alpha)}$ **end for**
5: **if** $r_\alpha \geq 1$ **then** $X_{r_\alpha}^{(\alpha)} \leftarrow \mathsf{add}(IV^{(\alpha)}, r_\alpha)$; $Y^* \xleftarrow{\$} \{0,1\}^{n-|M_{r_\alpha}^{(\alpha)}|}$; $Y_{r_\alpha}^{(\alpha)} \leftarrow (M_{r_\alpha}^{(\alpha)} \oplus C_{r_\alpha}^{(\alpha)}) \| Y^*$ **end if**
6: **end for**

9.2 Sampling Algorithm for HCTR2

In the decision stage of the ideal world, we define dummy hash keys $(H^{[\nu]})_{\nu \in [u]}$ and dummy internal pairs of plaintext and ciphertext blocks $\{(X_i^{(\alpha)}, Y_i^{(\alpha)}) \mid \alpha \in [q], i \in [0, r_\alpha]\}$ according to the structure of HCTR2. The sampling algorithm is defined in Algorithm 5.

9.3 Adversary's View

In the decision stage, all the user's keys and all internal pairs defined for each query are revealed to the adversary $\mathbf{A}$. Then, $\mathbf{A}$'s transcript τ consists of

- (dummy) user's keys $\mathcal{K}_{\mathsf{H}} = \left\{H^{[\nu]} \mid \nu \in [u]\right\}$,
- query-response tuples $\left\{(T^{(\alpha)}, M^{(\alpha)}, C^{(\alpha)} \mid \alpha \in [q]\right\}$, and
- (dummy) pairs $\left\{(X_i^{(\alpha)}, Y_i^{(\alpha)} \mid \alpha \in [q], i \in [0, r_\alpha]\right\}$.

9.4 Good and Bad Transcripts

We define three conditions on bad transcripts in the following. The set of bad transcripts $\mathcal{T}_{\mathsf{bad}}$ is a subset of $\mathcal{T}$ such that one of the conditions holds. The set of good transcripts is defined as $\mathcal{T}_{\mathsf{good}} := \mathcal{T} \backslash \mathcal{T}_{\mathsf{bad}}$, which is the subset of $\mathcal{T}$ such that no condition holds.

hit_0. The condition is defined as

$$\mathsf{hit}_0 \Leftrightarrow \exists \alpha \in [q], i \in [0, r_\alpha] \text{ s.t. } X_i^{(\alpha)} = \mathtt{bin}_n(0) \vee Y_i^{(\alpha)} = H^{[\mathsf{u}_\alpha]}.$$

In the condition, for some α, the hash key $H^{[\mathsf{u}_\alpha]}$ is used in the process of HCTR2 except for the hash key generation. In other words, assuming that hit_0 does not occur, all hash keys are independent of the process of HCTR2.

coll_X. The condition is defined as

$$\mathsf{coll}_X \Leftrightarrow \exists \nu \in [u], \alpha, \beta \in \mathcal{Q}^{[\nu]}, i \in [0, r_\alpha], j \in [0, r_\beta] \text{ s.t. } (\alpha, i) \neq (\beta, j) \wedge X_i^{(\alpha)} = X_j^{(\beta)}.$$

This condition considers a collision of plaintext blocks in the same user. Assuming that coll_X does not occur, for each user, the plaintext blocks are all distinct.

coll_Y. The condition is defined as

$$\mathsf{coll}_Y \Leftrightarrow \exists \nu \in [u], \alpha, \beta \in \mathcal{Q}^{[\nu]}, i \in [0, r_\alpha], j \in [0, r_\beta] \text{ s.t. } (\alpha, i) \neq (\beta, j) \wedge Y_i^{(\alpha)} = Y_j^{(\beta)}.$$

This condition considers a collision of ciphertext blocks in the same user. Assuming that coll_Y does not occur, for each user, the ciphertext blocks are all distinct.

9.5 Upper-Bounding $\Pr[\mathsf{T}_I \in \mathcal{T}_{\mathsf{bad}}]$

With the bad events hit_0, coll_X and coll_Y, we have

$$\begin{aligned}\Pr[\mathsf{T}_I \in \mathcal{T}_{\mathsf{bad}}] &\le \Pr[\mathsf{hit}_0] + \Pr[\mathsf{coll}_X] + \Pr[\mathsf{coll}_Y] \\ &\le \frac{\omega\sigma_{\mathsf{u}}\sigma}{2^n} + \frac{4n\sigma}{2^n} + \frac{2\omega\sigma_{\mathsf{u}}\sigma}{2^n} + \frac{\omega\sigma_{\mathsf{u}}\sigma}{2^n} \\ &= \frac{4\omega\sigma_{\mathsf{u}}\sigma}{2^n} + \frac{4n\sigma}{2^n}.\end{aligned}$$

The bounds of $\Pr[\mathsf{hit}_0]$, $\Pr[\mathsf{coll}_X]$, and $\Pr[\mathsf{coll}_Y]$ are given below.

Evaluation of $\Pr[\mathsf{hit}_0]$. For each $\alpha \in [q]$ and $i \in [0, r_\alpha]$, $X_i^{(\alpha)}$ and $Y_0^{(\alpha)}$ are defined by using the regular hash $\mathsf{H}_{H^{[\mathsf{u}_\alpha]}}$, and we have the following bounds:

$$\Pr[X_i^{(\alpha)} = \mathtt{bin}_n(0)] \le \frac{\omega(t_\alpha + r_\alpha + 1)}{2^n} \text{ and } \Pr[Y_0^{(\alpha)} = H^{[\mathsf{u}_\alpha]}] \le \frac{\omega(t_\alpha + r_\alpha + 1)}{2^n}.$$

For each $\alpha \in [q]$ and $i \in [r_\alpha]$, since $Y_i^{(\alpha)}$ is chosen uniformly at random from $\{0,1\}^n$, we have the following bound:

$$\Pr[Y_i^{(\alpha)} = H^{[\mathsf{u}_\alpha]}] \le \frac{1}{2^n}.$$

Summing these bounds for each α, i, we have

$$\begin{aligned}\Pr[\mathsf{hit}_0] &\le \sum_{\nu\in[u]} \sum_{\alpha\in\mathcal{Q}^{[\nu]}} \left(\frac{(r_\alpha+1)\cdot\omega(t_\alpha+r_\alpha+1)}{2^n} + \frac{\omega(t_\alpha+r_\alpha+1)}{2^n} + \frac{r_\alpha}{2^n} \right) \\ &\le \sum_{\nu\in[u]} \frac{\omega\sigma_\nu^2}{2^n} \\ &\le \frac{\omega\sigma_{\mathsf{u}}\sigma}{2^n}.\end{aligned}$$

Evaluation of $\Pr[\mathsf{coll}_X]$. We first define the following event for evaluating the collision $X_0^{(\alpha)} = X_j^{(\beta)}$. The event considers the case that for some query, the sum of the plaintext and the ciphertext of the right part has the structure of pad.

- $\mathsf{hit}_{\mathsf{pf}} : \exists \alpha \in [q], T' \in \mathcal{T}, D' \in \mathcal{M}$ s.t.
  - $M_*^{(\alpha)} \oplus C_*^{(\alpha)} = \mathsf{pad}_{\mathsf{pf}}(|T'|, 1) \| \mathsf{zp}_n(T') \| D'$ if $|D'| \bmod n = 0$;
  - $M_*^{(\alpha)} \oplus C_*^{(\alpha)} = \mathsf{pad}_{\mathsf{pf}}(|T'|, 0) \| \mathsf{zp}_n(T') \| \mathsf{ozp}_n(D')$ if $|D'| \bmod n \neq 0$.

For each $\alpha \in [q]$, there are at most $2n\ell_\alpha$ candidates for the prefix value $\mathsf{pad}_{\mathsf{pf}}(|T'|, b)$ where $b \in \{0,1\}$ and either $\mathsf{msb}_n(M_*^{(\alpha)})$ or $\mathsf{msb}_n(C_*^{(\alpha)})$ is chosen uniformly at random from at least $2^n - q$ elements, we have

$$\Pr[\mathsf{hit}_{\mathsf{pf}}] \le \sum_{\alpha\in[q]} \frac{2n\ell_\alpha}{2^n - q} \le \frac{4n\sigma}{2^n},$$

assuming that $q \le 2^{n-1}$. Hereafter, we assume that $\mathbf{A}$ aborts just after $\mathsf{hit}_{\mathsf{pf}}$ occurs.

For each of $\nu \in [u], \alpha, \beta \in \mathcal{Q}^{[\nu]}$, and $j \in [0, r_\beta]$, we consider the following cases regarding the collision $X_0^{(\alpha)} = X_j^{(\beta)}$, where X_0 and X_j $(j \ge 1)$ are respectively defined as

$$X_0 = M_0 \oplus \mathsf{H}_{H^{[\nu]}}(T, M_*) \text{ and } X_j = M_0 \oplus \mathsf{H}_{H^{[\nu]}}(T, M_*) \oplus C_0 \oplus \mathsf{H}_{H^{[\nu]}}(T, C_*) \oplus A_j.$$

– If $j = 0$, then $\alpha \neq \beta$ must hold, and by the AXU hash $\mathsf{H}_{H^{[\nu]}}$ we have

$$\Pr[X_0^{(\alpha)} = X_0^{(\beta)}] \leq \frac{\omega(\max\{t_\alpha + r_\alpha, t_\beta + r_\beta)\} + 1)}{2^n}.$$

– If $j \geq 1$, then we consider the following cases.
  - If $|\mathsf{pad}(T^{(\alpha)}, M_*^{(\alpha)})|_n \geq |\mathsf{pad}(T^{(\beta)}, M_*^{(\beta)})|_n$, then the prefix value $\mathsf{pad}_{\mathsf{pf}}(|T^{(\beta)}|, b)$ ($\neq 0^n$) ensures that $\mathsf{lsb}_{|\mathsf{pad}(T^{(\beta)}, M_*^{(\beta)})|}(\mathsf{pad}(T^{(\alpha)}, M_*^{(\alpha)}) \oplus \mathsf{pad}(T^{(\alpha)}, C_*^{(\alpha)})) \neq \mathsf{pad}(T^{(\beta)}, M_*^{(\beta)})$ holds, and hence by the A2XU hash, we have the following bound

$$\Pr[X_0^{(\alpha)} = X_j^{(\beta)}] \leq \frac{\omega(t_\alpha + r_\alpha + 1)}{2^n}.$$

  - If $|\mathsf{pad}(T^{(\alpha)}, M_*^{(\alpha)})|_n < |\mathsf{pad}(T^{(\beta)}, M_*^{(\beta)})|_n$, then since $\mathsf{hit}_{\mathsf{pf}}$ does not occur, $\mathsf{lsb}_{|\mathsf{pad}(T^{(\beta)}, M_*^{(\beta)})|}(\mathsf{pad}(T^{(\alpha)}, M_*^{(\alpha)}) \oplus \mathsf{pad}(T^{(\alpha)}, C_*^{(\alpha)})) \neq \mathsf{pad}(T^{(\beta)}, M_*^{(\beta)})$ holds. Hence, the A2XU hash ensures that

$$\Pr[X_0^{(\alpha)} = X_j^{(\beta)}] \leq \frac{\omega(\max\{t_\alpha + r_\alpha, t_\beta + r_\beta\} + 1)}{2^n}.$$

By the above bounds, we have

$$\Pr[X_0^{(\alpha)} = X_j^{(\beta)}] \leq \frac{\omega(\max\{t_\alpha + r_\alpha, t_\beta + r_\beta)\} + 1)}{2^n}.$$

We next consider a collision $X_i^{(\alpha)} = X_j^{(\beta)}$, where $\nu \in [u], \alpha, \beta \in \mathcal{Q}^{[\nu]}$, $i \in [r_\alpha]$, and $j \in [r_\beta]$. Without loss of generality, we assume that $\alpha \leq \beta$. Since $M^{(\beta)}$ (resp. $C^{(\beta)}$) is chosen uniformly at random from at least $2^n - q$ elements in $\{0,1\}^n$, we have

$$\Pr\left[\begin{array}{l}\exists (i,j) \in [r_\alpha] \times [r_\beta] \text{ s.t.} \\ \quad X_i^{(\alpha)} = X_j^{(\beta)} \wedge (\alpha, i) \neq (\beta, j)\end{array}\right] \leq \begin{cases} 0 & \text{if } \alpha = \beta, \\ \dfrac{r_\alpha r_\beta}{2^n - q} \leq \dfrac{2 r_\alpha r_\beta}{2^n} & \text{if } \alpha \neq \beta, \end{cases}$$

assuming $q \leq 2^{n-1}$.

By the above bounds, we have

$$\begin{aligned}\Pr[\mathsf{coll}_X] &\leq \frac{4n\sigma}{2^n} + \sum_{\nu \in [u]} \left(\sum_{\alpha \in \mathcal{Q}^{[\nu]}} \left(\frac{r_\alpha^2}{2^n} + \frac{\omega(t_\alpha + r_\alpha + 1) r_\alpha}{2^n} \right) \right. \\ &\qquad \left. + \sum_{\substack{\alpha, \beta \in \mathcal{Q}^{[\nu]} \\ \text{s.t. } \alpha < \beta}} \left(\frac{\omega(\max\{t_\alpha + r_\alpha, t_\beta + r_\beta)\} + 1)}{2^n} + \frac{2 r_\alpha r_\beta}{2^n} \right) \right) \\ &\leq \frac{4n\sigma}{2^n} + \sum_{\nu \in [u]} \left(\frac{2\omega\sigma_\nu^2}{2^n} \right) \\ &\leq \frac{4n\sigma}{2^n} + \frac{2\omega\sigma_{\mathsf{u}}\sigma}{2^n}.\end{aligned}$$

Evaluation of $\Pr[\mathsf{coll}_Y]$. For each $\nu \in [u], \alpha, \beta \in \mathcal{Q}^{[\nu]}$, the AXU hash function $\mathsf{H}_{H^{[\nu]}}$ ensures that

$$\Pr[Y_0^{(\alpha)} = Y_0^{(\beta)}] \leq \frac{\omega(\max\{t_\alpha + r_\alpha, t_\beta + r_\beta)\} + 1)}{2^n}.$$

For each $i \in [r_\alpha], j \in [0, r_\beta]$, since $Y_i^{(\alpha)}$ is chosen uniformly at random from $\{0,1\}^n$, we have the following bound:

$$\Pr[Y_i^{(\alpha)} = Y_j^{(\beta)}] \le \frac{1}{2^n}.$$

Summing the bounds for each ν, α, β, we have the following bound:

$$\begin{aligned}\Pr[\mathsf{coll}_Y] &\le \sum_{\nu\in[u]} \sum_{\substack{\alpha,\beta\in\mathcal{Q}^{[\nu]} \\ \text{s.t. } \alpha<\beta}} \frac{\omega(\max\{t_\alpha + r_\alpha, t_\alpha + r_\alpha)\} + 1) + r_\alpha(r_\beta + 1)}{2^n} \\ &\le \sum_{\nu\in[u]} \frac{\omega\sigma_\nu^2}{2^n} \\ &\le \frac{\omega\sigma_{\mathsf{u}}\sigma}{2^n}.\end{aligned}$$

9.6 Lower-Bounding $\frac{\Pr[\mathsf{T}_R=\tau]}{\Pr[\mathsf{T}_I=\tau]}$

For each good transcript τ, we show that $\frac{\Pr[\mathsf{T}_R=\tau]}{\Pr[\mathsf{T}_I=\tau]} \ge 1$. In this evaluation, we use the symbol "$\star$" for values and sets corresponding to a transcript τ. For a set $\mathcal{L}$, let $\mathsf{T}_R \vdash \mathcal{L}$ (resp. $\mathsf{T}_I \vdash \mathcal{L}$) be an event that T_R (resp. T_I) satisfies all elements in $\mathcal{L}$.

Fix a good transcript τ that consists of the hash keys $\mathcal{K}_{\mathsf{H}}$ and pairs of plaintext and ciphertext blocks.

Evaluation of $\Pr[\mathsf{T}_W \vdash \mathcal{K}_{\mathsf{H}}^\star]$ for $W \in \{R, I\}$. In the real world, each hash key $H^{[\nu]}$ is chosen uniformly at random from $\{0,1\}^n$ by $H^{[\nu]} = \pi(\mathtt{bin}_n(0))$. In the ideal world, each hash key $H_i^{[\nu]}$ is chosen uniformly at random from $\{0,1\}^n$. We thus have $\Pr[\mathsf{T}_R \vdash \mathcal{K}_{\mathsf{H}}^\star] = \Pr[\mathsf{T}_I \vdash \mathcal{K}_{\mathsf{H}}^\star]$.

Evaluation of $\Pr[\mathsf{T}_W \vdash (\mathcal{P}^{[\nu]\star})_{\nu\in[u]} \mid \mathcal{K}_{\mathsf{H}}^\star]$ for $W \in \{R, I\}$. For $\nu \in [u]$, let $\tau_\nu \coloneqq \{(X_i^{(\alpha)\star}, Y_i^{(\alpha)\star}) \mid \alpha \in \mathcal{Q}^{[\nu]}, i \in [0, r_\alpha]\}$ be the set of internal pairs of the ν-th user in τ, and $N_\nu \coloneqq |\tau_\nu|$.

Real-World Probability. By $\neg\mathsf{coll}_X$ and $\neg\mathsf{coll}_Y$, the real-world probability is not 0. We then have

$$\Pr[\mathsf{T}_R \vdash \tau_\nu] = \prod_{i\in[N_\nu]} \frac{1}{2^n - i}.$$

Ideal-World Probability. In the ideal world, for each $\alpha \in \mathcal{Q}^{[\nu]}$, if $\mathsf{query}_\alpha = \mathsf{enc}$ (resp. $\mathsf{query}_\alpha = \mathsf{dec}$), then $C^{(\alpha)}$ (resp. $M^{(\alpha)}$) is chosen uniformly at random from at least $2^{|M^{(\alpha)}|} - (\alpha - 1)$ elements such that $|M^{(\alpha)}| \ge n$ and the truncated value Y^* is chosen uniformly at random from $\{0,1\}^{n-|M_{r_\alpha}^{(\alpha)}|}$, thus each internal pair in the ideal world is defined from the space larger than the real world. We thus have

$$\Pr[\mathsf{T}_I \vdash \tau_\nu] \le \Pr[\mathsf{T}_R \vdash \tau_\nu].$$

Deriving the Lower Bound. Using the above bounds, we have

$$\frac{\Pr[\mathsf{T}_R = \tau]}{\Pr[\mathsf{T}_I = \tau]} \ge \prod_{\nu\in[u]} \frac{\Pr[\mathsf{T}_R \vdash \tau_\nu]}{\Pr[\mathsf{T}_I \vdash \tau_\nu]} \ge 1.$$

10 Software Performance Evaluation

This section benchmarks optimized software implementations of the proposed methods and the state-of-the-art modes. We instantiate the BBB-secure modes (CHCTR2, HCTR2-TwKD, and *bbb-ddd-AES*) using the 128-bit primitives, AES-256 and POLYVAL. Meanwhile, the birthday-bounded HCTR2 is instantiated with the 256-bit block cipher Rijndael-256-256, which benefits from AES instructions and is being considered for standardization by NIST [28, 37]. For instantiating HCTR2 with 256-bit primitives, we use a simple 256-bit universal hash function that concatenates two POLYVAL instances with different hash keys [26]. This comparison aims at comparing different approaches for achieving more than 64-bit security [6], and the security levels are not strictly aligned: HCTR2 with Rijndael-256-256 achieves 128-bit security while the others have $\frac{2}{3} \cdot 128$-bit security.

Our evaluation considers tweaks shorter than 128 bits to align with *bbb-ddd-AES*. In particular, we use CENC [24] for deriving 256-bit keys in HCTR2-TwKD, and the key derivation function F_L is given by

$$F_L : T \mapsto (E_L(00||T) \oplus E_L(01||T))||(E_L(00||T) \oplus E_L(10||T)),$$

wherein L is a 256-bit key, E_L is encryption with AES-256, and T is a 126-bit tweak.

10.1 Implementation

The software implementations of the target schemes are optimized using dedicated instructions for the x86 and ARM instruction sets. The descriptions below summarize the techniques for implementing each primitive on the x86 platform. The implementations on the ARM platform are based on the same techniques using the corresponding dedicated instructions.

AES-256. AES-256 can be realized simply with AES-NI [14], which implements AES rounds, and our AES-256 implementation comprises fifteen AVX instructions: (i) thirteen `vaesenc`, (ii) one `vaesenclast`, and (iii) one `vpxor` for the first key addition.

POLYVAL. The carry-less multiplication instruction [16], `vpclmulqdq`, is used for implementing FMUL in POLYVAL and the 256-bit hash function. FMUL involves textbook multiplication for obtaining a 256-bit polynomial and polynomial reduction to bring it back to a 128-bit polynomial. Our FMUL implementation uses aggregated polynomial reduction [25] to run several textbook multiplications in parallel and reduce the cost of the reduction step.

Rijndael-256-256. Rijndael-256-256 also benefits from AES-NI because it uses the same SubBytes and MixColumns as AES. By following the technique by Gueron [15], we use the shuffling (`vpshufb`) and blending (`vpblendvb`) instructions to fill the gap between AES and Rijndael-256-256, i.e., ShiftRows over a 256-bit state.

Optimization across two POLYVAL instances. Evaluating $\mathsf{H}^{\oplus}_{1,2}(T, R) = \mathsf{H}_1(T, R) \oplus \mathsf{H}_2(T, R)$ involves two POLYVAL computations on the same input under different hash keys (see Fig. 2). Since POLYVAL can be viewed as an inner product between a vector of message blocks and a vector of powers of the hash key, these two computations are inner products sharing one argument, and can thus be computed jointly using Winograd's inner product algorithm [39].[‡‡] Combined with Horner's method, this reduces

[‡‡]Winograd's inner product algorithm computes an inner product $\sum_{i=1}^{2n} a_i b_i$ using the identity $\sum_{i=1}^{2n} a_i b_i = \sum_{k=1}^{n} ((a_{2k-1} + b_{2k})(a_{2k} + b_{2k-1}) - a_{2k-1}a_{2k} - b_{2k-1}b_{2k})$. Since the term $a_{2k-1}a_{2k}$ depends only on the a_i, it can be reused to compute another inner product $\sum_{i=1}^{2n} a_i c_i$. Furthermore, if the b_i and c_i are known in advance, the terms $b_{2k-1}b_{2k}$ and $c_{2k-1}c_{2k}$ can be precomputed. In our setting, the a_i are the message blocks, and the b_i and c_i are the powers of the two hash keys.

the number of FMUL calls per block, excluding precomputation, from 2 to $1.5 + 1/b$, where b denotes the number of unrolled blocks in Horner's method. The first term, 1.5, comes from Winograd's inner product algorithm, while the second term, $1/b$, accounts for the one additional FMUL call per b blocks required by Horner's method.

Our software implementation unrolls the loops to improve performance by saturating pipelined execution units in the target processors. Our implementations process eight and twelve consecutive message blocks at once in block cipher encryption and in POLYVAL, respectively, which are the minimum unrolling factors with which the cpb performance reaches plateaus.

We implement two versions of *bbb-ddd-AES* with and without precomputation. Encryption and decryption of *bbb-ddd-AES* require a series of tweak-dependent masks given by $E_{K_2}(2||T||i)$, where K_2 is a key, T is a tweak, and i is a block index. When using the same tweak T repeatedly, we can improve performance by precomputing a table containing $E_{K_2}(2||T||i)$ for possible i [11]. This technique significantly impacts the performance by reducing the number of BC calls for each message block, as summarized in Table 1. Meanwhile, such an optimization is not always available; in the case of disk encryption, for example, the tweak is uncontrollable because that depends on a sector number.

10.2 Evaluation

We determine cpb performance by recording the values in the hardware performance monitors while the target implementation processes an 80 kB message. This procedure is performed 200 times, and the average is taken as the representative value. Table 2 shows the benchmark results for eight x86 processors from Intel and AMD, as well as four ARM processors from Apple and AWS. The evaluation environment includes both local machines and the Amazon AWS EC2 instances in the cloud [1].

We evaluate the performance of (i) a baseline implementation that invokes HCTR2 twice and (ii) the optimized implementation based on Winograd's algorithm. By comparing CHCTR2 with AES-256 and HCTR2 with Rijndael-256-256, which achieve more than 64-bit security without any restrictions on the tweak, the baseline implementation of CHCTR2 achieves better performance than HCTR2 with Rijndael-256-256 in all CPUs except for Ryzen 5 9600X; this negative result is explained by relatively slower POLYVAL performances in the AMD platform.

The optimized implementation of CHCTR2 using Winograd's algorithm improves the cpb by 0.9–9.5% over the baseline implementation. As a result of this optimization, the performance of CHCTR2 is better than HCTR2 with Rijndael-256-256 in all 12 CPUs, the reduction of cpb distributes between 78.0–96.9%. This result can be explained by the number of instructions as follows. For processing a 256-bit message chunk, CHCTR2 requires four AES-256 calls, involving 56 ($= 14 \times 4$) AES rounds, i.e., instructions. HCTR2, on the other hand, uses more instructions because a single Rijndael-256-256 needs 84 instructions for 28 ($= 14 \times 2$) AES rounds, 28 `vpblendvb` instructions, and 28 `vpshufb` instructions. Moreover, CHCTR2 invokes a smaller number of FMUL calls: CHCTR2 and HCTR2 use 7.25 (2 for H_1, $3.25 = 2 \cdot (1.5 + 1/8)$ for $\mathsf{H}^{\oplus}_{1,2}$ with an unrolling factor of 8, and 2 for H_2) and 8 FMUL calls for processing a 256-bit chunk, respectively.

Next, we compare HCTR2-TwKD and *bbb-ddd-AES*, which achieve BBB security when the same tweak is not repeated too often. First, Table 2 shows that HCTR2-TwKD outperforms *bbb-ddd-AES* without precomputation in all the processor models. Second, HCTR2-TwKD still achieves a similar performance compared with *bbb-ddd-AES* using precomputation. These results are aligned with the number of primitive calls shown in Table 1. In summary,

Table 2. Cycles-per-byte (cpb) performances of the VIL-STPRP schemes instantiated for more than 64-bit security in twelve processor models. The entries with † are measured on Amazon AWS instances on the cloud. *bbb-ddd-AES* has two implementations with and without precomputation (PC).

Processor Model	Architecture	CHCTR2		HCTR2	HCTR2-TwKD	*bbb-ddd-AES*	
		Baseline	Winograd			w/o PC	w/ PC
Intel Core Ultra 7 265K	Arrow Lake-S	1.748	1.732	2.116	0.884	1.526	1.098
Intel Xeon w5-3423	Sapphire Rapids	2.022	1.923	2.198	1.019	1.747	1.175
Intel Xeon 8488C†	Sapphire Rapids	2.021	1.927	2.197	1.020	1.749	1.184
Intel Xeon 8375C†	Ice Lake	2.031	1.981	2.172	1.028	1.981	1.377
AMD Ryzen 5 9600X	Zen5	3.068	2.910	3.004	1.548	2.354	1.746
AMD Ryzen 5 7600X	Zen4	3.095	2.953	3.144	1.558	2.557	1.914
AMD EPYC 9R14†	Zen4	3.095	2.952	3.144	1.558	2.586	1.896
AMD EPYC 7R13†	Zen3	3.101	2.925	3.630	1.561	2.738	2.063
Apple M2	Avalanche	1.539	1.435	1.842	0.784	1.314	1.023
Apple M1	Firestorm	1.588	1.473	1.841	0.809	1.387	1.058
AWS Graviton4†	Noverse-V2	1.872	1.708	2.265	0.953	1.634	1.245
AWS Graviton3†	Noverse-V1	2.025	1.832	2.341	1.029	1.717	1.393

HCTR2-TwKD is better than *bbb-ddd-AES* when the tweak cannot be repeated. Even when the tweak is repeated, HCTR2-TwKD achieves a similar performance without paying the memory cost for storing the precomputed values while maintaining backward compatibility with HCTR2.

References

[1] Amazon Web Services, Inc.: Amazon EC2 instance types. https://aws.amazon.com/ec2/instance-types/ (2024)

[2] Android Open Source Project: File-based encryption. https://source.android.com/docs/security/features/encryption/file-based (2023)

[3] Bellare, M., Rogaway, P.: Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In: ASIACRYPT 2000. pp. 317–330 (2000)

[4] Chakraborty, D., Sarkar, P.: HCH: A new tweakable enciphering scheme using the hash-encrypt-hash approach. In: INDOCRYPT 2006. LNCS, vol. 4329, pp. 287–302. Springer (2006)

[5] Chen, S., Steinberger, J.P.: Tight security bounds for key-alternating ciphers. In: EUROCRYPT 2014. LNCS, vol. 8441, pp. 327–350 (2014)

[6] Chen, Y.L., Davidson, M., Dworkin, M., Kelsey, J., Sasaki, Y., Turan, M.S., Chang, D., Mouha, N., Thompson, A.: Requirements for Cryptographic Accordions. NIST Interagency Report, NIST IR 8552 (Apr 2025), https://csrc.nist.gov/pubs/ir/8552/final

[7] Chen, Y.L., Hiraga, Y., Mouha, N., Naito, Y., Sasaki, Y., Sugawara, T.: Beyond-birthday-bound security with hctr2: Cascaded construction and tweak-based key derivation. In: ASIACRYPT 2025. LNCS, vol. 16245, pp. 3–34 (2025)

[8] Crowley, P., Huckleberry, N., Biggers, E.: Length-preserving encryption with HCTR2. IACR Cryptol. ePrint Arch. (2021), https://eprint.iacr.org/2021/1441

[9] Datta, N., Dutta, A., Ghosh, S., Nandi, H.: Optimally secure TBC based accordion mode. Cryptology ePrint Archive, Paper 2024/2053 (2024), https://eprint.iacr.org/2024/2053

[10] Dobraunig, C., Matusiewicz, K., Mennink, B., Tereschenko, A.: Efficient instances of docked double decker with AES. IACR Cryptol. ePrint Arch. p. 84 (2024), https://eprint.iacr.org/2024/084

[11] Dobraunig, C., Matusiewicz, K., Mennink, B., Tereschenko, A.: Efficient instances of docked double decker with aes, and application to authenticated encryption. In: EUROCRYPT 2025. pp. 62–92 (2025)

[12] Dutta, A., Nandi, M.: Tweakable HCTR: A BBB secure tweakable enciphering scheme. In: INDOCRYPT 2018. LNCS, vol. 11356, pp. 47–69. Springer (2018)

[13] Duy, H.N., Fernández, P.G., Udovenko, A., Biryukov, A.: Accordion mode based on hash-encrypt-hash. Submitted to Workshop (Jun 2024), https://csrc.nist.gov/csrc/media/Events/2024/accordion-cipher-mode-workshop-2024/documents/papers/accordion-mode-based-hash-encrypt-hash.pdf

[14] Gueron, S.: Intel advanced encryption standard (AES) new instructions set. https://www.intel.com/content/dam/doc/white-paper/advanced-encryption-standard-new-instructions-set-paper.pdf (2010)

[15] Gueron, S.: Constructions based on the aes round and polynomial multiplications that are efficient on modern processor architectures. https://csrc.nist.gov/csrc/media/Presentations/2023/constructions-based-on-the-aes-round/images-media/sess-5-gueron-bcm-workshop-2023.pdf (2023)

[16] Gueron, S., Kounavis, M.E.: Intel carry-less multiplication instruction and its usage for computing the GCM mode. https://www.intel.com/content/dam/develop/external/us/en/documents/clmul-wp-rev-2-02-2014-04-20.pdf (2014)

[17] Gueron, S., Lindell, Y.: Better Bounds for Block Cipher Modes of Operation via Nonce-Based Key Derivation. In: CCS 2017. pp. 1019–1036. ACM (2017)

[18] Gunsing, A., Daemen, J., Mennink, B.: Deck-based wide block cipher modes and an exposition of the blinded keyed hashing model. IACR Trans. Symmetric Cryptol. **2019**(4), 1–22 (2019)

[19] Halevi, S.: Invertible universal hashing and the TET encryption mode. In: CRYPTO 2007. LNCS, vol. 4622, pp. 412–429. Springer (2007)

[20] Halevi, S., Rogaway, P.: A tweakable enciphering mode. In: CRYPTO 2003. LNCS, vol. 2729, pp. 482–499. Springer (2003)

[21] Halevi, S., Rogaway, P.: A parallelizable enciphering mode. In: CT-RSA 2004. LNCS, vol. 2964, pp. 292–304. Springer (2004)

[22] Hoang, V.T., Krovetz, T., Rogaway, P.: Robust authenticated-encryption AEZ and the problem that it solves. In: EUROCRYPT 2015. LNCS, vol. 9056, pp. 15–44 (2015)

[23] Hoang, V.T., Tessaro, S.: Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In: CRYPTO 2016. vol. 9814, pp. 3–32. Springer (2016)

[24] Iwata, T.: New blockcipher modes of operation with beyond the birthday bound security. In: FSE 2006. LNCS, vol. 4047, pp. 310–327. Springer (2006)

[25] Jankowski, K., Laurent, P.: Packed AES-GCM algorithm suitable for AES/PCLMULQDQ instructions. IEEE Trans. Computers **60**(1), 135–138 (2011)

[26] Lee, B.: A BBB secure accordion mode from HCTR. Workshop Presentation (Jun 2024), https://csrc.nist.gov/csrc/media/Presentations/2024/a-bbb-secure-accordion-mode-from-hctr/images-media/sess-8-lee-acm-workshop-2024.pdf
[27] Lucks, S.: The sum of prps is a secure PRF. In: EUROCRYPT 2000. LNCS, vol. 1807, pp. 470–484. Springer (2000)
[28] Morris, D.: NIST plans. Posted on NIST Ciphermodes Forum (Aug 2024), https://groups.google.com/a/list.nist.gov/g/ciphermodes-forum/c/D5qni2KDoms
[29] Naito, Y., Sasaki, Y., Sugawara, T.: The exact multi-user security of (tweakable) key alternating ciphers with a single permutation. In: EUROCRYPT 2024, Part I. LNCS, vol. 14651, pp. 97–127. Springer (2024)
[30] Naito, Y., Sasaki, Y., Sugawara, T., Yasuda, K.: The multi-user security of triple encryption, revisited: Exact security, strengthening, and application to TDES. In: CCS 2022. pp. 2323–2336. ACM (2022)
[31] NIST: Advanced encryption standard (AES). FIPS 197 (2023), published November 26, 2001; Updated May 9, 2023, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf
[32] NIST: The third NIST workshop on block cipher modes of operation 2023. https://csrc.nist.gov/events/2023/third-workshop-on-block-cipher-modes-of-operation (2023)
[33] NIST: NIST workshop on the requirements for an accordion cipher mode 2024. https://csrc.nist.gov/Events/2024/accordion-cipher-mode-workshop-2024 (2024)
[34] NIST: Pre-draft call for comments: Nist launches development of cryptographic accordions. https://csrc.nist.gov/pubs/sp/800/197/a/iprd (2025)
[35] Patarin, J.: The "Coefficients H" technique. In: SAC 2008. vol. 5381, pp. 328–345. Springer (2008)
[36] The Linux Kernel: hctr2.c. https://github.com/torvalds/linux/blob/master/crypto/hctr2.c, accessed: 2025-05-15
[37] Turan, M.S.: Open discussion next steps. Discussed at NIST Workshop (Jun 2024), https://csrc.nist.gov/csrc/media/Presentations/2024/open-discussion/images-media/sess-9-turan-acm-workshop-2024.pdf
[38] Wang, P., Feng, D., Wu, W.: HCTR: A variable-input-length enciphering mode. In: Information Security and Cryptology, First SKLOIS Conference, CISC 2005. pp. 175–188 (2005)
[39] Winograd, S.: A new algorithm for inner product. IEEE Transactions on Computers **C-17**(7), 693–694 (1968). https://doi.org/10.1109/TC.1968.227420