Key Committing Security of HCTR2, Revisited

Donghoon Chang[1,2], Yu Long Chen[3], Yukihito Hiraga[4], Kazuhiko Minematsu[5,6], Nicky Mouha[7], Yusuke Naito[8], Yu Sasaki[1,9], and Takeshi Sugawara[4]

[1] Associate of National Institute of Standards and Technology
[2] FWI, Fairfax, USA
[3] KU Leuven, Leuven, Belgium
[4] The University of Electro-Communications, Tokyo, Japan
[5] NEC Corporation, Kawasaki, Japan
[6] The University of Osaka, Suita, Japan
[7] KeyCryptic, Bethesda, USA
[8] Mitsubishi Electric Corporation, Kanagawa, Japan
[9] NTT Social Informatics Laboratories, Tokyo, Japan

Abstract. This paper presents improved attacks and proofs for the key committing security of EtE-HCTR2, a robust authenticated encryption scheme constructed from HCTR2 and the Encode-then-Encipher (EtE) framework, in light of the ongoing standardization effort of cryptographic accordions by NIST. We improve attacks on the instantiations with two common encodings, where zeros are either appended or prepended to the message, namely $\mathsf{EtE_A}$-HCTR2 and $\mathsf{EtE_P}$-HCTR2. Compared with the state-of-the-art attack by Chen et al. in ToSC 2023(4), our $\mathsf{EtE_A}$-HCTR2 attack reduces the complexity from $O(2^{\tau/2})$ to $O(2^{\max\{\tau/3,\tau-n\}})$ for an n-bit block cipher and τ-bit zero padding, which degrades $\mathsf{EtE_A}$-HCTR2's security below the birthday bound. Meanwhile, our $\mathsf{EtE_P}$-HCTR2 attack reduces the complexity from $O(2^{\min\{n/2,\tau\}})$ to $O(2^{\tau/2})$, which is tight with our new security proof. We verify these computationally-bounded attacks by experimentally generating concrete vectors for both $\mathsf{EtE_A}$-HCTR2 with $\tau = 96$ and $\mathsf{EtE_P}$-HCTR2 with $\tau = 64$, each instantiated with $n = 128$, in less than 15 minutes. We consider yet another padding scheme that appends zeros to the first message block, namely $\mathsf{EtE_S}$-HCTR2, and prove that it has a tight committing security bound of $O(2^{\tau/2})$ by avoiding the issue in $\mathsf{EtE_A}$-HCTR2.

Keywords: HCTR2 · EtE · Key commitment · Generic attack · Proof

1 Introduction

Cryptographic accordions [9], block-cipher (BC) modes for realizing a variable-input-length strong tweakable pseudorandom permutation (VIL-STPRP), are drawing attention since the National Institute of Standards and Technology (NIST) started the development project in 2023. HCTR2 [12] is a cryptographic

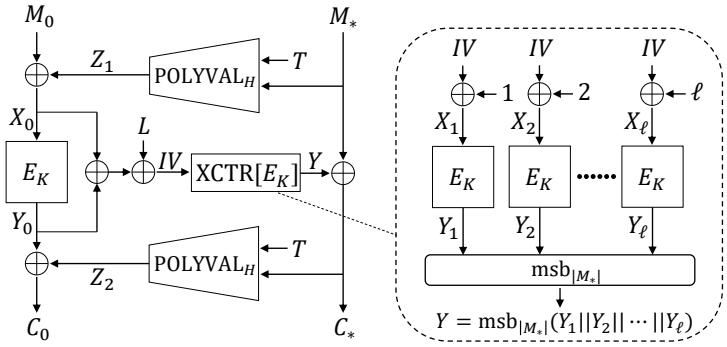

Fig. 1. HCTR2 [12]. $H = E_K(\texttt{bin}(0))$. $L = E_K(\texttt{bin}(1))$. msb_l returns the most significant l bits of the input.

accordion that achieves provable security up to the birthday bound [12], designed by Crowley, Huckleberry, and Biggers as an improvement of its predecessor HCTR [36]. HCTR2 is based on the hash-encrypt-hash structure shown in Fig. 1, which can be viewed as a 3-round unbalanced Feistel construction consisting of a long data part and a short one-block part. In the first round, the long side is hashed and mixed into the short side. In the second round, the short side is encrypted, and the resulting value is used to transform the entire long side in a CTR-like manner. In the third round, the transformed long side is hashed again and mixed back into the short side. As a result, both sides become mutually dependent. Therefore, even a one-bit change in the input or output causes the entire output or input, respectively, to change randomly. At the same time, the overall construction remains an invertible length-preserving transformation. HCTR2 comprising a polynomial hash function POLYVAL [21, 22] and XCTR typically instantiated with AES. HCTR2 has seen widespread deployment, including the Linux kernel [35] and Android file-based encryption [4], and NIST recently proposed to develop accordions based on the HCTR2 technique [33].

According to *Requirements for Cryptographic Accordions* (NIST IR 8552) [9], accordion modes are expected to support several derived functions, including authenticated encryption with associated data (AEAD) and deterministic authenticated encryption (DAE). In particular, the efficient realization of Robust Authenticated Encryption (RAE) [3, 25] that generalizes both DAE and misuse-resistant AE (MRAE) [34] is a significant advantage of cryptographic accordions. An RAE can be built simply using the Encode-then-Encipher (EtE) framework [7] that applies special encoding to a message verified during decryption. We are interested in the instantiations using the widely-used encoding schemes that append and prepend zeros to the message, namely EtE_A and EtE_P, respectively.

Committing security is a relatively new AEAD security property that restricts decryption of a ciphertext to a certain set of parameters, such as a key,

nonce, associated data (AD), and plaintext. It is relevant to the security of real-world systems, such as end-to-end encrypted messaging [13,18,27], key rotation for key management services, envelope encryption [5,23,24], Google's Subscribe with Google (SwG) [2], and password-based encryption and key exchange [30]. Committing security considers an adversary who knows or even chooses a key, which is beyond the scope of conventional security definitions of AEAD, including RAE. Consequently, Farshim et al. [16] formalized the security notion, and it has evolved over the years [2, 10, 13, 16, 18, 31]; we refer to Menda et al.'s work [31] for a survey of these developments and the relationships among the various definitions. This work considers a slightly different definition of key commitment security, since the syntax of EtE does not necessarily involve a nonce N or AD A. The encryption takes (K, T, M) as input, where K is a key, T is a tweak, and M is a message, and returns a ciphertext C. In the CMT_k notion, the adversary wins if it returns (K, T, M) and (K', T', M') that encrypt to the same ciphertext and $K \neq K'$. The $\mathsf{CMT}^*_\mathsf{k}$ notion is stronger and requires $K \neq K'$ and $T = T'$. Assuming that (N, A) is encoded into a tweak, our $\mathsf{CMT}^*_\mathsf{k}$ and CMT_k notions correspond to those defined by Menda et al. [31]. The attacks in this paper target the $\mathsf{CMT}^*_\mathsf{k}$ notion, while the proofs target the CMT_k notion.

NIST requires committing security for accordion-based AEADs.[10] Along with the research direction of studying the committing security of general accordion modes [26,32], the committing security of EtE-$\mathsf{HCTR2}$, an RAE constructed from $\mathsf{HCTR2}$ and EtE, has become a new challenge. Figs. 2 and 3 summarize the previous and proposed attacks on EtE_P-$\mathsf{HCTR2}$ and EtE_A-$\mathsf{HCTR2}$, respectively, for the bit length τ of zero padding and the block length n of an underlying BC. Fig. 2-(1) shows a trivial attack that works with $O(2^\tau)$ complexity by an exhaustive search of K that generates a correctly encoded message, while fixing C, T, T'. Chen et al. [10] proposed attacks with birthday-bound complexity, where the CMT_k attack is reduced to a collision-finding problem. Their EtE_P-$\mathsf{HCTR2}$ attack works up to $\tau \leq n$ and has $O(2^{\min\{n/2,\tau\}})$ complexity; the attack is advantageous over the trivial attack for $\tau > \frac{n}{2}$, as depicted in Fig. 2-(2). Meanwhile, their EtE_A-$\mathsf{HCTR2}$ attack has $O(2^{\tau/2})$ complexity, which is depicted in Fig. 3-(4).

Our goal is to address significant open research problems necessary for determining the sufficient amount of padding τ to achieve a target level of committing security with EtE-$\mathsf{HCTR2}$. Can we further improve the CMT_k and/or $\mathsf{CMT}^*_\mathsf{k}$ attacks of EtE-$\mathsf{HCTR2}$? In particular, Chen et al.'s attack on EtE_P-$\mathsf{HCTR2}$ [10] requires $O(2^{n/2})$ complexity, which is in contrast with the EtE_A-$\mathsf{HCTR2}$ attack with $O(2^{\tau/2})$ complexity. Can we improve the attack on EtE_P-$\mathsf{HCTR2}$ for $\tau \leq n$? Furthermore, can we lower-bound the security with a security proof, ideally with a tight bound? There has been no proof for the committing security of $\mathsf{HCTR2}$, unlike $\mathsf{Adiantum}$ [11] or AEZ [25], which were proved by Chen et al. [10].

[10] NIST IR 8552 [9, Sect. 4.4.2]: "Because key commitment is important in many real-world applications, NIST expects to require this property in the derived functions that support authentication".

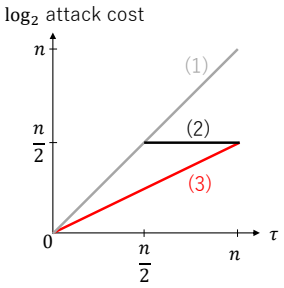

Fig. 2. Comparison of attack complexities for $\mathsf{EtE_P}$-HCTR2:
(1) trivial $O(2^\tau)$ attack
(2) previous $O(2^{n/2})$ attack
(3) our $O(2^{\tau/2})$ attack

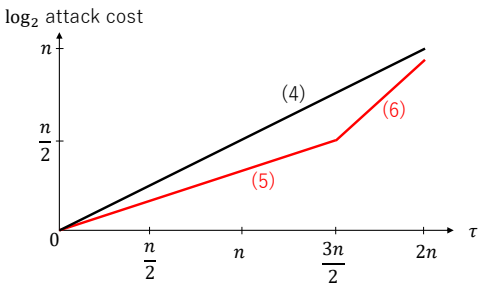

Fig. 3. Comparison of attack complexities for $\mathsf{EtE_A}$-HCTR2:
(4) previous $O(2^{\tau/2})$ attack
(5) our $O(2^{\tau/3})$ attack
(6) our $O(2^{\tau-n})$ attack with multi-collisions

Table 1. Comparison of attacks with an approximate complexity.

Ref.	$\mathsf{EtE_P}$-HCTR2	$\mathsf{EtE_A}$-HCTR2	$\mathsf{EtE_S}$-HCTR2
Trivial	$O(2^\tau)$	$O(2^\tau)$	$O(2^\tau)$
Chen et al. [10]	$O(2^{\min\{n/2,\tau\}})$	$O(2^{\tau/2})$	–
Ours	$O(2^{\tau/2})^*$	$O(2^{\max\{\tau/3,\tau-n\}})$	$O(2^{\tau/2})^{*\dagger}$

*: Tight attacks due to our proofs.
†: This is a trivial application of the attack against $\mathsf{EtE_A}$-HCTR2 by [10].

Our Contributions. This paper addresses the open research questions by studying improved attacks and new proofs of $\mathsf{EtE_P}$-HCTR2 and $\mathsf{EtE_A}$-HCTR2, along with another padding scheme $\mathsf{EtE_S}$ that appends zeros to the first message block, similar to the one used for Adiantum in the previous work [10]. Table 1 summarizes our attack complexities and their tightness in comparison with the state-of-the-art. Note that $\mathsf{EtE_S}$-HCTR2 can be attacked with $O(2^{\tau/2})$ complexity with a trivial application of the previous attack on $\mathsf{EtE_A}$-HCTR2 [10]. Our results are generic in the sense that both attacks and proofs work for any BC.

Attacks (Section 4). We present $\mathsf{CMT}^*_\mathsf{k}$ attacks on $\mathsf{EtE_A}$-HCTR2 (Section 4.2) and $\mathsf{EtE_P}$-HCTR2 (Section 4.3) improved from the previous work [10]. They are computationally-bounded (cf. information-theoretic) attacks: we count not only the BC calls, but also all the involved computations as a necessary cost.

$\mathsf{EtE_A}$-HCTR2. Our new attack works with $O(2^{\tau/3})$ complexity for $\tau \leq \frac{3n}{2}$, as depicted in Fig. 3-(5). When $\frac{3n}{2} \leq \tau < 2n$, the attack complexity becomes $O(2^{\tau-n})$ with a simple extension. The attack can be improved by introducing multicollisions, reducing the attack cost by a few bits when $\tau \gg \frac{3n}{2}$ and slightly

improving the generic birthday-bound attack on τ even at $\tau = 2n$, which is depicted in Fig. 3-(6). Note that $\tau = \frac{3n}{2}$, which is 192 with 128-bit BCs like AES, is already large. The impact of achieving the attack below the birthday bound for such a wide range of τ is significant. It is also expected that the practical encoding size is up to $\tau = 2n$, and we successfully improve the attack in any practical τ. The approach to break the birthday bound is a divide-and-conquer attack. We choose the message size so that the first $\frac{2\tau}{3}$ encoded bits are located in the ith block, and the remaining $\frac{\tau}{3}$ bits are located in the $(i+1)$th block. A collision attack is performed on the $\frac{2\tau}{3}$ bits in the ith block, and the brute-force attack is applied in the $(i+1)$th block.

EtE$_\mathsf{P}$-HCTR2. Our new attack, on the other hand, works with $O(2^{\tau/2})$ complexity for $\tau \leq n$, which is depicted in Fig. 2-(3). This upper bound is more intuitive and simpler than the previous results. Moreover, it shows that achieving higher security than the birthday bound on τ is impossible. Concretely, for a small encoding size such as $\tau \leq \frac{n}{2}$, our $O(2^{\tau/2})$ attack improves the trivial $O(2^{\tau})$ attack for the first time. Such a small encoding size is practically relevant, for example, AES-CCM in NIST SP800-38C [14] allows 64-bit tags, which corresponds to $\tau = \frac{n}{4}$ when an underlying BC has $n = 256$ such as Rijndael-256. The attack is based on a new approach searching for paired internal state values having the same input and output difference of the BC on the left-hand side.

Finally, we provide experimental verification of the proposed attacks in Section 4.4. To support our performance claims, concrete vectors are generated for HCTR2 instantiated with AES-128. Our optimized implementations using AVX instructions generate the input vectors for EtE$_\mathsf{A}$-HCTR2 with $\tau = 96$ in 652 seconds and EtE$_\mathsf{P}$-HCTR2 with $\tau = 64$ in 347 seconds.

Proof (Sections 5, 6, and 7). We show the tight security bounds with the matching proofs for EtE$_\mathsf{S}$-HCTR2 (Theorem 1 and the proof in Section 6) and EtE$_\mathsf{P}$-HCTR2 (Theorem 2 and the proof in Section 7). Note that EtE$_\mathsf{A}$-HCTR2 is omitted because it only provides below-birthday-bound security, making it an impractical option. More specifically, EtE$_\mathsf{S}$-HCTR2 and EtE$_\mathsf{P}$-HCTR2 are CMT$_\mathsf{k}$ secure with the bounds approximately $\max\{\frac{p^2}{2^\tau}, \frac{p}{2^n}\}$ in the ideal cipher (IC) model, where p is the number of queries to an IC.

EtE$_\mathsf{S}$-HCTR2. Our proof starts by reducing EtE$_\mathsf{S}$-HCTR2's CMT$_\mathsf{k}$ security to the collision resistance of a partial function called DMXCTR that accepts a pair (K, X) and is composed of Davies-Meyer (DM) and XCTR, as shown in Fig. 4-(left) (Lemma 1 in Section 6, see its proof in Section 6.1 for the rationale behind this simplification). Regarding the collision resistance of DMXCTR with $\tau \leq n$, if there is no collision at the n-bit middle point between DM and XCTR with an inverse operation of the BC, the BC's outputs of XCTR are defined by forward operations. The collision probabilities of the middle point by an inverse operation and of the outputs of XCTR by forward operations are, respectively, by the birthday analysis, about $\frac{p^2}{2^n}$ and $\frac{p^2}{2^\tau}$. When $n < \tau$, a BC is called twice in XCTR. For handling the two output blocks, we introduce an additional query to an IC, where if one of the output blocks in XCTR is defined, then the other

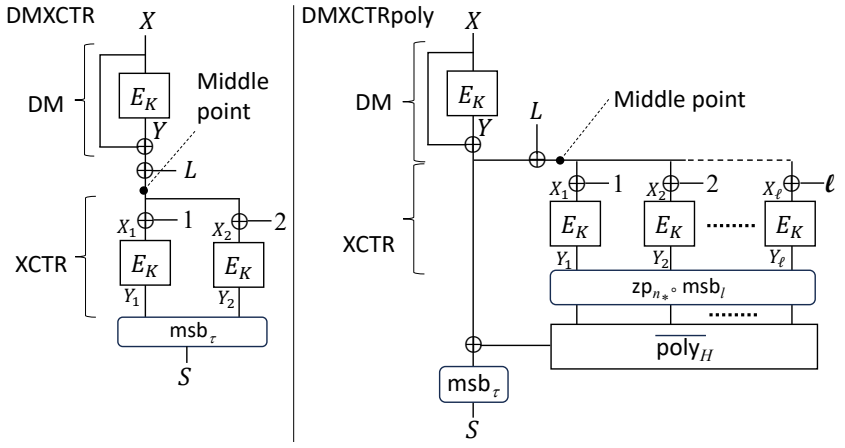

Fig. 4. DMXCTR with $\tau \leq 2n$ (left) and DMXCTRpoly with $\tau \leq n$ (right). $L = E_K(1)$. msb_l returns the most significant l bits of the input. zp_{n*} is a padding function that appends a zero string so that the last block is a full one.

output block is defined by a forward query to an IC. The additional query ensures that each collision probability of the middle point by an inverse operation and of the outputs of XCTR can be about $\max\{\frac{p^2}{2^\tau}, \frac{p}{2^n}\}$. Hence, we have the bound $\max\{\frac{p^2}{2^\tau}, \frac{p}{2^n}\}$ of the $\mathsf{CMT_k}$ security of $\mathsf{EtE_S}$-HCTR2.

EtE$_\mathsf{P}$-HCTR2. In proving the security, on the other hand, the main technical challenge is handling of Z_1 and Z_2, the outputs from POLYVAL in HCTR2 that are XORed with the τ padded value (see Fig. 1). An attacker may try to break $\mathsf{CMT_k}$ security by arbitrarily choosing Z_1 and Z_2 through the freeness of the inputs, but we show that the structure of $\mathsf{EtE_P}$-HCTR2 prohibits such attacks, which is the key discovery. Our proof starts by reducing $\mathsf{EtE_P}$-HCTR2's $\mathsf{CMT_k}$ security to the collision resistance of a partial function called DMXCTRpoly in Fig. 4-(right), which accepts an input tuple (l, K, X) and returns a τ-bit output S (Lemma 3 in Section 7, see Section 7.1 for its proof). DMXCTRpoly is composed of DM, XCTR, and $\overline{\mathsf{poly}}$ (polynomial hash) that takes the outputs from the BCs only. If there is no collision at the n-bit middle point between DM and XCTR with an inverse operation of the BC, all inputs to $\overline{\mathsf{poly}}$ are outputs of the BCs defined by forward operations. Hence, the output from $\overline{\mathsf{poly}}$ cannot be controlled by the adversary and is seen as a random value, and the collision probability on the middle point is approximately $\frac{p^2}{2^n}$. Without such a collision, the outputs from DMXCTRpoly become τ-bit random values, ensuring that the collision probability on the outputs is approximately $\frac{p^2}{2^\tau}$. Hence, we have the bound $\frac{p^2}{2^\tau}$ of the $\mathsf{CMT_k}$ security of $\mathsf{EtE_P}$-HCTR2.

2 Preliminaries

Notation. Let ε be the empty string and $\emptyset$ be the empty set. For an integer $i \geq 0$, let $\{0,1\}^i$ be the set of all i-bit strings and $\{0,1\}^0 \coloneqq \{\varepsilon\}$. Let $\{0,1\}^{\geq i}$ be the set of all bit strings of length $\geq i$. For a bit string X and an integer i, let $\mathsf{msb}_i(X)$ and $\mathsf{lsb}_i(X)$ be the most and least significant i bits of X, respectively. For integers $0 \leq i \leq j$, let $[i,j] \coloneqq \{i, i+1, \ldots, j\}$, and $[j] \coloneqq [1,j]$. If $j < i$ then $[i,j] \coloneqq \emptyset$. Let $\mathtt{ozp}_{n*}$ (resp. $\mathtt{zp}_{n*}$) be a padding function that accepts an arbitrary-length bit-string X and returns $X\|10^s$ (resp. $X\|0^s$) such that s is the minimum non-negative integer such that $|X\|10^s| \bmod n = 0$ (resp. $|X\|0^s| \bmod n = 0$). For a non-negative integer i, let $\mathtt{bin}_n(i)$ be the little-endian conversion of i to an n-bit string. For a value or a set X, $Y \leftarrow X$ means that X is assigned to Y. For a positive integer n and a bit string D of length $\geq n$, $(D_1, D_2) \xleftarrow{n,|D|-n} D$ means that 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 the ℓ 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 non-empty set $\mathcal{T}$, $T \xleftarrow{\$} \mathcal{T}$ means that an element is chosen uniformly at random from $\mathcal{T}$ 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 $\mathsf{UConf}_j(\mathcal{S}) \coloneqq \{\{S_1, S_2, \ldots, S_j\} \subseteq \mathcal{S}^j \mid \forall (i_1, i_2) \in [j]^2 : i_1 \neq i_2 \Rightarrow S_{i_1} \neq S_{i_2}\}$. For a set $\mathcal{S}$ of tuples of l elements and $j \in [l]$, let $(y_1, \ldots, y_{j-1}, *, y_{j+1}, \ldots, y_l) \in \mathcal{S}$ be a condition that $\exists y$ s.t. $(y_1, \ldots, y_{j-1}, y, y_{j+1}, \ldots, y_l) \in \mathcal{S}$.

Block Cipher. In this paper, we fix the key and block sizes of the underlying block cipher to k and n, respectively. 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 keyed block cipher is denoted by $E_K(\cdot) = E(K, \cdot)$. The decryption is denoted by E_K^{-1}.

Our proofs are given in the ideal (block) cipher model, wherein an ideal cipher is defined as $E \xleftarrow{\$} \mathcal{BC}$, and all users and adversaries have access to E and E^{-1}. Adversary's queries to an ideal cipher are called "primitive queries". In our proofs, an ideal cipher is realized by the lazy sampling method. The method of realizing the lazy sampled IC is given in the following.

Let $\mathcal{E}$ be a table that is initially empty and keeps query-response tuples of $E^{\pm} = (E, E^{-1})$. Let $\mathcal{E}(K,X) \coloneqq Y$ and $\mathcal{E}^{-1}(K,Y) \coloneqq X$ if $(K,X,Y) \in \mathcal{E}$, and $\mathcal{E}(K,X) \coloneqq \varepsilon$ (resp. $\mathcal{E}^{-1}(K,Y) \coloneqq \varepsilon$) if $(K,X,*) \notin \mathcal{E}$ (resp. $(K,*,Y) \notin \mathcal{E}$). Let $\mathcal{E}(K,*) \coloneqq \{Y \mid (K,*,Y) \in \mathcal{E}\}$ and $\mathcal{E}^{-1}(K,*) \coloneqq \{X \mid (K,X,*) \in \mathcal{E}\}$.

- For a query (K,X) to E, the ciphertext Y is defined as follows:
  $Y \leftarrow \mathcal{E}(K,X)$; **if** $Y = \varepsilon$ **then** $Y \xleftarrow{\$} \{0,1\}^n \backslash \mathcal{E}(K,*)$; $\mathcal{E} \xleftarrow{\cup} \{(K,X,Y)\}$.
- For a query (K,Y) to E^{-1}, the plaintext X is defined as follows:
  $X \leftarrow \mathcal{E}^{-1}(K,Y)$; **if** $X = \varepsilon$ **then** $X \xleftarrow{\$} \{0,1\}^n \backslash \mathcal{E}^{-1}(K,*)$; $\mathcal{E} \xleftarrow{\cup} \{(K,X,Y)\}$.

Key Committing Security. Committing security of symmetric-key encryption was introduced by Farshim et al. [16]. Later, the concept was expanded by

providing a fine-grained class of notions in a number of works such as [2, 6, 31]. The key-committing security denotes the difficulty of finding distinct input tuples that encrypt to an identical ciphertext under different keys. The syntax of AEAD built by applying EtE to a tweakable BC does not necessarily contain nonce and AD, thus we define our key-commitment security for EtE based on Menda et al. [31].

Definition 1. *Let* $\mathsf{EtE}[P]$ *be an EtE AE scheme based on an internal idealized primitive* P *that takes a key* K*, a tweak* T*, and a plaintext* M*, and returns a ciphertext. The* $\mathsf{CMT}^*_{\mathsf{k}}$ *and* $\mathsf{CMT}_{\mathsf{k}}$ *advantages of* EtE *are defined as*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{cmt}^*_{\mathsf{k}}}_{\mathsf{EtE}[P]}(\mathbf{A}) &\coloneqq \Pr\big[((K,T,M),(K',T',M')) \leftarrow \mathbf{A}^P,\\ &\qquad K \neq K', T = T', \mathsf{EtE}_K(T,M) = \mathsf{EtE}_{K'}(T',M')\big],\\ \mathbf{Adv}^{\mathsf{cmt}_{\mathsf{k}}}_{\mathsf{EtE}[P]}(\mathbf{A}) &\coloneqq \Pr\big[((K,T,M),(K',T',M')) \leftarrow \mathbf{A}^P,\\ &\qquad K \neq K', \mathsf{EtE}_K(T,M) = \mathsf{EtE}_{K'}(T',M')\big].\end{aligned}$$

Assuming that a pair of a nonce and AD is encoded into a tweak, our $\mathsf{CMT}^*_{\mathsf{k}}$ and $\mathsf{CMT}_{\mathsf{k}}$ notions correspond to those defined by Menda et al. Since $\mathsf{CMT}^*_{\mathsf{k}}$ requires the tweak values in the input tuple to be identical, while $\mathsf{CMT}_{\mathsf{k}}$ does not, hence, $\mathsf{CMT}_{\mathsf{k}}$ is a stronger notion. The attacks studied in this paper target the $\mathsf{CMT}^*_{\mathsf{k}}$ security notion, while the proofs target the $\mathsf{CMT}_{\mathsf{k}}$ security notion. The previous attacks [10] allow tweaks to be different, hence the attacks in this paper are performed in the more challenging setting than that of [10].

3 Specification of HCTR2

HCTR2. The specification of HCTR2 [12] is given in Algorithm 1. HCTR2 is an accordion mode with a hash-encrypt-hash structure, where the hashing parts use the polynomial hash function POLYVAL, and the encryption part uses XCTR. $\mathsf{POLYVAL}_H$ is a hash function with a key $H \in \{0,1\}^n$ that takes a tweak $T \in \{0,1\}^*$ and a part of plaintext/ciphertext in $\{0,1\}^*$ and returns an n-bit hash value. In HCTR2, the hash key is defined as $H = E_K(\mathtt{bin}_n(0))$. $\mathsf{XCTR}[E_K]$ is a mode of stream encryption that takes an n-bit IV and the data length and returns a key stream of the specified length.

$\mathsf{HCTR2.enc}[E_K] : \{0,1\}^* \times \{0,1\}^{\geq n} \to \{0,1\}^{\geq n}$ is the encryption of HCTR2 that takes a tweak $T \in \{0,1\}^*$ and a plaintext $M \in \{0,1\}^{\geq n}$, and returns a ciphertext C such that $|M| = |C|$. In the encryption, a plaintext M is divided into an n-bit left part of the plaintext M_0 and the remaining right part of the plaintext M_*. In the first hashing part, $\mathsf{POLYVAL}_H$ with T and M_* is performed to define the output Z_1. In the encrypting part, the value $X_0 = M_0 \oplus Z_1$ is encrypted by using the BC E_K and the value $IV = X_0 \oplus Y_0 \oplus L$, where Y_0 is the output of the BC, is used as an IV of XCTR. The right part of the plaintext M_* is encrypted by using XCTR with IV, and the right part of the ciphertext C_* is obtained. Finally, the second hashing part is performed, where $\mathsf{POLYVAL}_H$ with T and C_* is performed, and the left n-bit part of the ciphertext C_0 is defined by XORing the hash value Z_2 with the output Y_0. The ciphertext is defined as $C = C_0 \| C_*$.

Algorithm 1 HCTR2

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

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

Decryption $\mathsf{HCTR2.dec}[E_K^{\pm}](T, C)$

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

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

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

Subroutine $\mathsf{POLYVAL}_H(T, D)$

1: **if** n **divides** $|D|$ **then** $Z \leftarrow \overline{\mathsf{poly}}_H(\mathtt{bin}(2|T|+2) \| \mathtt{zp}_{n*}(T) \| D)$
else $Z \leftarrow \overline{\mathsf{poly}}_H(\mathtt{bin}(2|T|+3) \| \mathtt{zp}_{n*}(T) \| \mathtt{ozp}_{n*}(D))$ **end if**
2: **return** Z

Subroutine $\overline{\mathsf{poly}}_H(\tilde{D})$

1: $D_1, \ldots, D_\ell \xleftarrow{n} \tilde{D}$; $\overline{H} \leftarrow H \cdot \mathtt{x}^{-n}$
2: $Z \leftarrow D_1 \cdot \overline{H}^{\ell} \oplus D_2 \cdot \overline{H}^{\ell-1} \oplus \cdots \oplus D_{\ell-1} \cdot \overline{H}^{2} \oplus D_\ell \cdot \overline{H}$; **return** Z

$\mathsf{HCTR2.dec}[E_K] : \{0,1\}^* \times \{0,1\}^{\geq n} \to \{0,1\}^{\geq n}$ is the decryption of HCTR2 that takes a tweak $T \in \{0,1\}^*$ and a ciphertext $C \in \{0,1\}^{\geq n}$, and returns a plaintext M such that $|M| = |C|$. In the decryption, a plaintext C is divided into an n-bit left part of the ciphertext C_0 and the remaining right part of the ciphertext C_*. Then, the inverse of the encryption procedures is performed and the plaintext M is obtained.

POLYVAL is a polynomial hash function and accepts $T \in \{0,1\}^*$ and $D \in \{0,1\}^*$, and returns an n-bit value. The multiplications are done over the Galois field $GF(2^n)$ with an irreducible polynomial $p(\mathtt{x})$. The irreducible polynomial of HCTR2 is defined as $p(\mathtt{x}) = \mathtt{x}^{128} + \mathtt{x}^{127} + \mathtt{x}^{126} + \mathtt{x}^{121} + 1$. In the multiplications, an n-bit string $a = a_0 \| a_1 \| \cdots \| a_{n-2} \| a_{n-1} \in \{0,1\}^n$ is regarded as an element in $GF(2^n)$: $a_{n-1}\mathtt{x}^{n-1} + a_{n-2}\mathtt{x}^{n-2} + \cdots + a_1\mathtt{x} + a_0$, where $\forall i \in [0, n-1] : a_i \in \{0,1\}$.

EtE-HCTR2. We consider three EtE-based AE schemes with HCTR2, namely $\mathsf{EtE_A}$-HCTR2, $\mathsf{EtE_P}$-HCTR2, and $\mathsf{EtE_S}$-HCTR2. Let τ be the size of ciphertext expansion, i.e., 0^τ is embedded into an original plaintext. The encoded value is input to HCTR2.

The encryption of those AE schemes takes a tweak $T \in \{0,1\}^*$ and a plaintext $M \in \{0,1\}^*$. In $\mathsf{EtE_A}$-HCTR2, 0^τ is appended to a plaintext, and in $\mathsf{EtE_P}$-HCTR2, 0^τ is prepended to a plaintext. In $\mathsf{EtE_S}$-HCTR2, 0^τ is inserted between the most significant n bits of a plaintext and the other bits assuming $|M| \geq n$. Those are defined as $\mathsf{EtE_A}\text{-}\mathsf{HCTR2.enc}[E_K](T, M) \coloneqq \mathsf{HCTR2.enc}[E_K](T, M\|0^\tau)$,
$\mathsf{EtE_P}\text{-}\mathsf{HCTR2.enc}[E_K](T, M) \coloneqq \mathsf{HCTR2.enc}[E_K](T, 0^\tau\|M)$,
$\mathsf{EtE_S}\text{-}\mathsf{HCTR2.enc}[E_K](T, M) \coloneqq \mathsf{HCTR2.enc}[E_K](T, M_0\|0^\tau\|M_r)$,
where $M_0 = \mathsf{msb}_n(M)$ and $M_r = \mathsf{lsb}_{|M|-n}(M)$.

4 Attacks on $\mathsf{CMT}^*_\mathsf{k}$ Security of EtE-HCTR2

4.1 Inverting POLYVAL

It is widely known that for a given hash key H, input data D, and an output value Z, it is easy to find a tweak T such that $Z = \mathsf{POLYVAL}_H(T, D)$. Chen et al. [10] explained the procedure as follows. Note that $|T|$ and $|D|$ are supposed to be multiples of n for simplicity. The extension to the other cases is trivial.

$$\begin{aligned}
\mathsf{POLYVAL}_H(T, D) = Z &\Leftrightarrow \overline{\mathsf{poly}}_H(\mathtt{bin}(2|T|+2)\|T\|D) = Z \\
&\Leftrightarrow (2|T|+2) \cdot \overline{H}^{\frac{|T|+|D|}{n}+1} \oplus \overline{\mathsf{poly}}_H(T) \cdot \overline{H}^{\frac{|D|}{n}} \oplus \overline{\mathsf{poly}}_H(D) = Z \\
&\Leftrightarrow \overline{\mathsf{poly}}_H(T) = \left(Z \oplus (2|T|+2) \cdot \overline{H}^{\frac{|T|+|D|}{n}+1} \oplus \overline{\mathsf{poly}}_H(D)\right) \cdot \overline{H}^{-\frac{|D|}{n}}
\end{aligned}$$

where $H \neq 0$ is known, and the right-hand side can be computed down to a single value. One can arbitrarily set every block of T except for one and compute the required value for the remaining block using a simple algebra.

It is easy to extend this analysis to find a common tweak $T = T'$ such that $\mathsf{POLYVAL}_H(T, D) = Z$ and $\mathsf{POLYVAL}_{H'}(T', D') = Z'$ are simultaneously satisfied for given values H, D, Z, H', D', and Z'. For this, the adversary needs to solve a system of two linear equations, and the solution can be found if at least two blocks of $T(= T')$ can be controlled by the adversary.

4.2 Attacks on $\mathsf{EtE_A}$-HCTR2

Previous Attack on $\mathsf{EtE_A}$-HCTR2. The attack by Chen et al. [10] first chooses any K and K', and searches for two distinct input values to the BC on the left-hand side, denoted by X_0 and X'_0 in Fig. 5, that collide on the last τ bits of the key stream i.e., $\mathsf{lsb}_\tau(Y) = \mathsf{lsb}_\tau(Y')$. This is the bottleneck, which requires $O(2^{\tau/2})$ computations. Then, M_0 and M_* are fixed to any values, and T is computed by inverting the first hash layer $X_0 \oplus M_0 = \mathsf{POLYVAL}_H(T, M_*\|0^\tau)$, which further fixes C_0 and C_*. Their goal is to attack the $\mathsf{CMT_k}$ notion, so $C'_0 = C_0$ and $C'_* = C_*$ must hold, though T' can differ from T. The attack computes T' by inverting the second hash layer $E_{K'}(X'_0) \oplus C_0 = \mathsf{POLYVAL}_{H'}(T', C_*)$. Finally, M'_0 and M'_* are generated by decrypting C'.

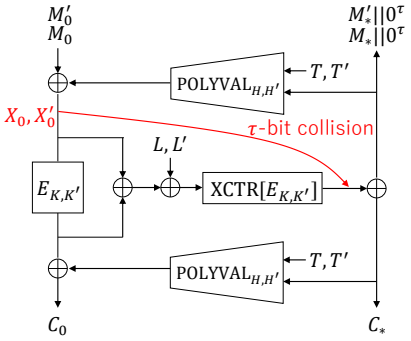

Fig. 5. $O(2^{\tau/2})$ attack for $\mathsf{EtE_A}$-HCTR2.

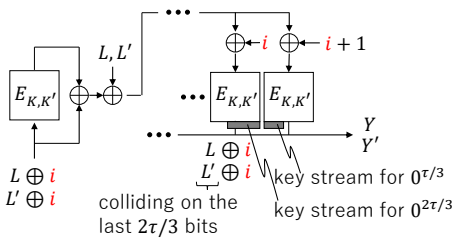

Fig. 6. Core of our $O(2^{\tau/3})$ attack.

Application to $\mathsf{EtE_S}$-HCTR2. Before we explain our attack on $\mathsf{EtE_A}$-HCTR2, we note that the previous attack against $\mathsf{EtE_A}$-HCTR2 is trivially applied to $\mathsf{EtE_S}$-HCTR2, where 0^τ is prepended to M_*. The core of the above attack against $\mathsf{EtE_A}$-HCTR2 is to generate a collision on the key stream for encrypting the appended 0^τ. This configuration is exactly the same for attacking $\mathsf{EtE_S}$-HCTR2. The only difference is that the collision must occur at the first τ bits of the key stream instead of the last τ bits i.e., $\mathsf{msb}_\tau(Y) = \mathsf{msb}_\tau(Y')$ instead of $\mathsf{lsb}_\tau(Y) = \mathsf{lsb}_\tau(Y')$. Hence, $\mathsf{EtE_S}$-HCTR2 can be attacked with $O(2^{\tau/2})$ computations.

$O(2^{\tau/3})$ Attack with $\tau \leq \frac{3n}{2}$. We present an attack on $\mathsf{EtE_A}$-HCTR2 that is faster than the birthday attack. The complexity is $O(2^{\tau/3})$ and is applied when $\tau \leq \frac{3n}{2}$. The key observation of our attack is that "When the output value of the BC on the left-hand side is $L \oplus \mathtt{bin}_n(i)$, the key stream value of the ith block of the XCTR mode is $L \oplus \mathtt{bin}_n(i)$." Namely, we have the following Proposition 1.

Proposition 1. $Y_0 = L \oplus \mathtt{bin}_n(i) \Rightarrow Y_i = L \oplus \mathtt{bin}_n(i)$.

Proof. The input value of the BC on the left-hand side is $E_K^{-1}(L \oplus \mathtt{bin}_n(i))$. Hence, the input value to the ith block is $E_K^{-1}(L \oplus \mathtt{bin}_n(i)) \oplus (L \oplus \mathtt{bin}_n(i)) \oplus L \oplus \mathtt{bin}_n(i)$, which is $E_K^{-1}(L \oplus \mathtt{bin}_n(i))$. Hence, Y_i is $E_K\big(E_K^{-1}(L \oplus \mathtt{bin}_n(i))\big)$, which is $L \oplus \mathtt{bin}_n(i)$. □

The attack diagram using Prop. 1 is depicted in Fig. 6. $|M|$ is set to $(i-1)n + \frac{\tau}{3}$ bits. After 0^τ is appended, $\frac{2\tau}{3}$ bits of zeros are placed in the ith block and $\frac{\tau}{3}$ bits are placed in the $(i+1)$th block. The attacker's goal is to find a pair of inputs with $K \neq K'$ such that the last τ bits of the key streams collide.

In the ith block, to ensure the collision on the $\frac{2\tau}{3}$ bits, we find two keys K, K' that produce L, L' colliding on the last $\frac{2\tau}{3}$ bits, namely $\mathsf{lsb}_{\frac{2\tau}{3}}(E_K(\mathtt{bin}_n(1))) = \mathsf{lsb}_{\frac{2\tau}{3}}(E_{K'}(\mathtt{bin}_n(1)))$. This ensures that the $\frac{2\tau}{3}$ bits of $L \oplus i$ and $L' \oplus i$ also collide for any i. The property that the collision on the ith block is ensured independently of the choice of i enables a divide-and-conquer attack.

In the $(i+1)$th block, the attacker finds a value of i that collides on the $\frac{\tau}{3}$ bits in a brute-force manner. Finally, (K, T, M) and (K', T, M') are generated by inverting POLYVAL. In summary, the attack consists of the following three phases, where the complexity is $O(2^{\tau/3})$ each. The pseudo-code is given in Alg. 2.

Phase 1. Searching for a pair (K, K') that collides on the last $\frac{2\tau}{3}$ bits of (L, L').
Phase 2. Searching for i that derives a collision on $\mathsf{msb}_{\tau/3}(Y_{i+1})$.[11]
Phase 3. Generating an input pair with a length of $O(2^{\tau/3})$ blocks.

Regarding the success probability, the attack consists of generating a collision on $\frac{2\tau}{3}$ bits at **Phase 1** and the brute-force search on $\frac{\tau}{3}$ bits at **Phase 2**. One iteration of **Phase 1** requires 1 BC call, hence, with a cost of $2^{\tau/3}$ BC invocations, the success probability is $1 - e^{-1/2} \approx 0.39$, which increases to $1 - e^2 \approx 0.84$ by spending $2 \cdot 2^{\tau/3}$ BC invocations. One iteration of **Phase 2** requires 4 BC calls and with a cost of $4 \cdot 2^{\tau/3}$ BC invocations, the success probability is $1 - e^{-1} \approx 0.63$. The probability that both phases succeed is $0.84 \times 0.63 \approx 0.54$.

Note that 1 encryption of $\mathsf{EtE_A}$-HCTR2 is more expensive than 1 BC due to POLYVAL, XCTR, etc. It is unclear how many $\mathsf{EtE_A}$-HCTR2 invocations our attack corresponds to. For simplicity, we focus on the asymptotic complexity of $O(2^{\tau/3})$, rather than a detailed analysis of the constant factors.

A Basic Extension for $\frac{3n}{2} \leq \tau < 2n$. When $\tau > \frac{3n}{2}$, the $O(2^{\tau/3})$ attack becomes inapplicable because the number of the colliding bits in the ith block is limited to at most n, which is now less than $\frac{2\tau}{3}$ for this range of τ. Hence, at **Phase 1**, we find K and K' yielding a full collision on L, i.e., $E_K(\mathtt{bin}_n(1)) = E_{K'}(\mathtt{bin}_n(1))$, with a complexity of $O(2^{n/2})$ and at **Phase 2** search for i that derives an identical key stream on the first $\tau - n$ bits in the $(i+1)$th block with complexity $O(2^{\tau-n})$.

The logarithm of the attack complexity increases linearly from $n/2$ to n as τ increases from $\frac{3n}{2}$ to $2n$. With $\tau = 2n$, the attack complexity matches that of the $O(2^{\tau/2})$ attack by Chen et al. [10]. However, by introducing multi-collisions, our attack can be better even if $\tau = 2n$, as explained in the following paragraph.

Further Improvement with Multi-collisions. When τ gets larger than $\frac{3n}{2}$ and approaches $2n$, the cost of **Phase 1**, which requires $O(2^{n/2})$ to find K, K' deriving the same L, is much lower than that of **Phase 2**, which requires $O(2^{\tau-n})$. We show that the attack can be improved by balancing those two costs, particularly by generating multi-collisions at **Phase 1**. The detailed analysis will be given in Appendix A. Here, we discuss the intuition of this improvement.

In **Phase 1**, we find r keys $\{K^{(1)}, \ldots, K^{(r)}\}$ colliding on L, more specifically, $E_{K^{(1)}}(\mathtt{bin}_n(1)) = \cdots = E_{K^{(r)}}(\mathtt{bin}_n(1))$. To find such an r-collision, we consider choosing $2^{\frac{r-1}{r}n + \frac{\log_2(r!)+1}{r}}$ keys and computing their L, under the assumption that

[11] This attack cannot be applied to $\mathsf{EtE_S}$-HCTR2 because i must be 1 for $\mathsf{EtE_S}$-HCTR2, and it cannot exploit the advantage of considering many choices of i.

Algorithm 2 $O(2^{\tau/3})$ Attack against $\mathsf{EtE_A}$-HCTR2 with $\tau \leq \frac{3n}{2}$

Require: τ satisfying $\tau \leq \frac{3n}{2}$
Ensure: (K, T, M), (K', T, M')
s.t. $\mathsf{HCTR2.enc}[E_K](T, M\|0^\tau) = \mathsf{HCTR2.enc}[E_{K'}](T, M'\|0^\tau)$

Phase 1. Find K, K' partially colliding on L
for $i = 1, \ldots, 2^{\tau/3}$ **do**
  $K^{(i)} \xleftarrow{\$} \{0,1\}^k$; $L^{(i)} \leftarrow E_{K^{(i)}}(\mathtt{bin}_n(1))$
end for
Find $i1$ and $i2$ such that $L^{(i1)}$ and $L^{(i2)}$ collide on the last $\frac{2\tau}{3}$ bits.
Set $K \leftarrow K^{(i1)}, K' \leftarrow K^{(i2)}$ and derive L, L', H, H'.

Phase 2. Search for i colliding in the $(i+1)$th block
$\ell \leftarrow \emptyset$; $i = 1$
while $\ell = \emptyset$ **do**
  $IV \leftarrow E_K^{-1}(L \oplus \mathtt{bin}_n(i)) \oplus \mathtt{bin}_n(i)$; $Y_{i+1} \leftarrow E_K(IV \oplus \mathtt{bin}_n(i+1))$
  $IV' \leftarrow E_{K'}^{-1}(L' \oplus \mathtt{bin}_n(i)) \oplus \mathtt{bin}_n(i)$; $Y'_{i+1} \leftarrow E_{K'}(IV' \oplus \mathtt{bin}_n(i+1))$
  if $\mathsf{msb}_{\tau/3}(Y_{i+1}) = \mathsf{msb}_{\tau/3}(Y'_{i+1})$ **then** $\ell \leftarrow i$
  end if
  $i \leftarrow i + 1$
end while

Phase 3. Generate paired input tuples
Set the message length $|M|$ to be $|M| \leftarrow n + \ell n - \frac{2\tau}{3}$.
$C \xleftarrow{\$} \{0,1\}^{|M|} \| \mathsf{lsb}_{2\tau/3}\big(E_K(IV \oplus \mathtt{bin}_n(\ell))\big) \| \mathsf{msb}_{\tau/3}\big(E_K(IV \oplus \mathtt{bin}_n(\ell+1))\big)$
for $i = 1, \cdots, \ell - 1$ **do**
  $Y_i \leftarrow E_K(IV \oplus \mathtt{bin}_n(i))$; $Y'_i \leftarrow E_{K'}(IV' \oplus \mathtt{bin}_n(i))$
end for
$M_* \leftarrow Y \oplus C_*$; $M'_* \leftarrow Y' \oplus C_*$
Find T to satisfy both $C_0 \oplus (L \oplus \mathtt{bin}_n(\ell)) = \mathsf{POLYVAL}_H(T, C_*)$ and $C_0 \oplus (L' \oplus \mathtt{bin}_n(\ell)) = \mathsf{POLYVAL}_{H'}(T, C_*)$ by inverting $\mathsf{POLYVAL}$.
$M_0 \leftarrow \mathsf{POLYVAL}_H(T, M_*) \oplus E_K^{-1}(L \oplus \mathtt{bin}_n(\ell))$
$M'_0 \leftarrow \mathsf{POLYVAL}_{H'}(T, M'_*) \oplus E_{K'}^{-1}(L' \oplus \mathtt{bin}_n(\ell))$
return $(K, T, \mathsf{msb}_{|M|}(M_0\|M_*))$, $(K', T, \mathsf{msb}_{|M|}(M'_0\|M'_*))$

$\frac{r-1}{r}n + \frac{\log_2(r!)+1}{r} \leq k$. In **Phase 2**, to examine the collision on other $\tau - n$ bits by changing i, we have r choices for each i, and it is sufficient to find a collision on any two of r choices. Hence, the complexity for each i increases by a factor of r compared to the basic extension, while the probability of generating a collision for each i increases to $\frac{\binom{r}{2}}{2^{\tau-n}} = \frac{r(r-1)}{2^{\tau-n+1}}$, namely, it increases roughly by a factor of r^2, and the number of i that needs to be tested is reduced by the same factor. In the end, the complexity of **Phase 2** is reduced by a factor of r. **Phase 3** does not change and is always cheaper than **Phase 2**. Hence, the optimal attack complexity can be computed by carefully choosing r for a given τ.

Table 2 presents several comparisons for $n = 128$. Complexities are evaluated by including constant factors, e.g., the number of BC calls in the defined computations. Figure 7 illustrates how the complexity of our attack and that of

Table 2. Complexity comparison for $\tau \geq \frac{3n}{2}$ with $n = 128$. The complexity of our attack is given by Eq. (1) and the birthday attack is evaluated by $(1 + \lceil \frac{\tau}{n} \rceil)2^{\tau/2}$.

τ	Attack with r-collision	Generic birthday attack [10]
192	$2^{66.58}$ $(r = 2)$	$2^{97.59}$
213	$2^{87.00}$ $(r = 2)$	$2^{108.09}$
214	$2^{87.78}$ $(r = 3)$	$2^{108.59}$
224	$2^{97.00}$ $(r = 3)$	$2^{113.59}$
225	$2^{98.41}$ $(r = 4)$	$2^{114.09}$
256	$2^{126.51}$ $(r = 22)$	$2^{129.59}$
260	$2^{130.16}$ $(r = 43)$	$2^{132.00}$

the birthday attack increase as τ increases for $n = 128$. Even when $\tau \geq 256$, our

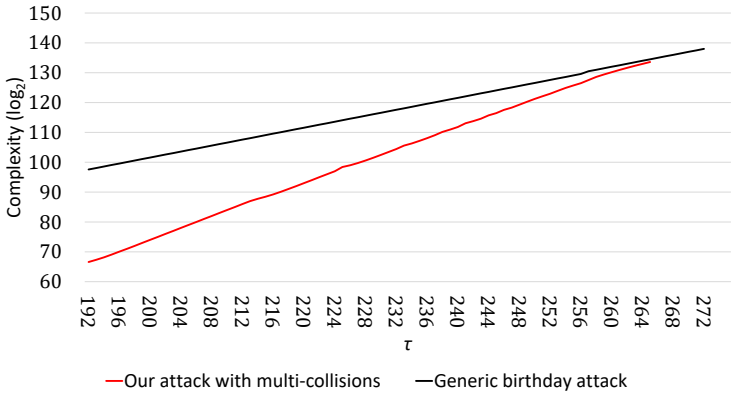

Fig. 7. Impact of Using Multi-collisions when $n = 128$.

attack achieves an improvement of a few bits.

Extension for Restricted Message Length. The attacks so far assume that there is no restriction on the input message length. One may wonder about the impact on the attack when the maximum input message length is limited.

Assume that $\tau \leq \frac{2n}{3}$. Let $M_{\max}$ denote the maximum input message length. The $O(2^{\tau/3})$ attack requires that $M_{\max} \geq 2^{\tau/3}$ blocks, thus placing $0^{\frac{2\tau}{3}}$ in the

ith block and $0^{\tau/3}$ in the $i+1$th block, and generating collisions for $0^{\frac{2\tau}{3}}$ by the birthday paradox and $0^{\frac{\tau}{3}}$ via a brute-force search by changing i from 1 to $2^{\tau/3}$. The attack can be generalized by parameterizing the number of message blocks examined in the $i+1$th block as X. The attack places $0^{\tau-\log_2 X}$ in the ith block and $0^{\log_2 X}$ in the $i+1$th block. Then, we generate collisions for $0^{\tau-\log_2 X}$ by the birthday paradox and $0^{\log_2 X}$ via a brute-force search by changing i from 1 to X, leading to the attack complexity of $O(2^{\frac{\tau-\log_2 X}{2}})+X$.

If $M_{\max} \geq 2^{\tau/3}$ blocks, the complexity is optimized with $X = 2^{\tau/3}$, and this is exactly our $O(2^{\tau/3})$ attack. If $M_{\max} \ll 2^{\tau/3}$ blocks, we have $O(2^{\frac{\tau-\log_2 X}{2}}) \gg X$ and the complexity is minimized when $X = M_{\max}$, which gives $O(2^{\frac{\tau-\log_2 M_{\max}}{2}})$. As long as $M_{\max}$ is meaningfully large, the attack is faster than $O(2^{\tau/2})$.

4.3 Attacks on EtE$_\mathsf{P}$-HCTR2

Previous Attack on EtE$_\mathsf{P}$-HCTR2. For EtE$_\mathsf{P}$-HCTR2, intuitively, the attacker can trivially ensure a collision between C_* and C'_* by properly choosing M_* and M'_*. Hence, the most challenging part is to generate a collision between C_0 and C'_0, while ensuring the presence of the encoded 0^τ in M_0 and M'_0.

The attack by Chen et al. [10] first chooses any K, K', and fixes M_* and M'_* to 0^n. They observed that once X_0 is chosen and K and M_0 are fixed, a function mapping X_0 to C_0 is uniquely determined. The analysis is as follows. When X_0 is chosen, the output of XCTR Y is computed as $Y = \mathsf{XCTR}(X_0 \oplus E_K(X_0) \oplus L)$. The computation of the first hash layer is $\mathsf{POLYVAL}_H(T, M_*)$, which can be decomposed into $\mathsf{POLYVAL}_H(T, C_*) \oplus \overline{\mathsf{poly}}_H(Y)$. This is because

$$\begin{aligned}
\mathsf{POLYVAL}_H(T, M_*) &= \mathsf{POLYVAL}_H(T, C_* \oplus Y) = \overline{\mathsf{poly}}_H(\mathtt{bin}(2|T|+2) \| T \| C_* \oplus Y) \\
&= (2|T|+2) \cdot \overline{H}^{\frac{|T|+|D|}{n}+1} \oplus \overline{\mathsf{poly}}_H(T) \cdot \overline{H}^{\frac{|D|}{n}} \oplus \overline{\mathsf{poly}}_H(C_* \oplus Y) \\
&= (2|T|+2) \cdot \overline{H}^{\frac{|T|+|D|}{n}+1} \oplus \overline{\mathsf{poly}}_H(T) \cdot \overline{H}^{\frac{|D|}{n}} \oplus \overline{\mathsf{poly}}_H(C_*) \oplus \overline{\mathsf{poly}}_H(Y) \\
&= \mathsf{POLYVAL}_H(T, C_*) \oplus \overline{\mathsf{poly}}_H(Y),
\end{aligned}$$

where the attacker chooses $|T|$ and $|C_*|$ to be multiples of n in advance. This transformation is depicted in Fig. 8. Namely, the term $\mathsf{poly}_H(Y)$ is computed solely by Y independently of C_*, and the term $\mathsf{POLYVAL}_H(T, C_*)$ becomes exactly the same as the computation of the second hash layer. From the first hash layer, we have $M_0 \oplus \mathsf{POLYVAL}_H(T, C_*) \oplus \overline{\mathsf{poly}}_H(Y) = X_0$. Since $M_0 = 0^n$, we have $\mathsf{POLYVAL}_H(T, C_*) = \overline{\mathsf{poly}}_H(Y) \oplus X_0$. From the second hash layer, we have $Y_0 \oplus \mathsf{POLYVAL}_H(T, C_*) = C_0$, which yields $C_0 = Y_0 \oplus X_0 \oplus \overline{\mathsf{poly}}_H(Y)$. By expressing Y_0 and Y in terms of X_0, we finally have the function mapping X_0 to C_0 as $C_0 = X_0 \oplus E_K(X_0) \oplus \overline{\mathsf{poly}}_H(\mathsf{XCTR}(X_0 \oplus E_K(X_0) \oplus L))$.

The entire attack is depicted in Fig. 9. The attacker chooses $2^{n/2}$ values for X_0 and X'_0 and computes the corresponding C_0 and C'_0. This allows her to find a collision between C_0 and C'_0 while satisfying the requirement for the encoded 0^τ in M_0 and M'_0 with $O(2^{n/2})$ complexity. Note that C_* can be fixed to any,

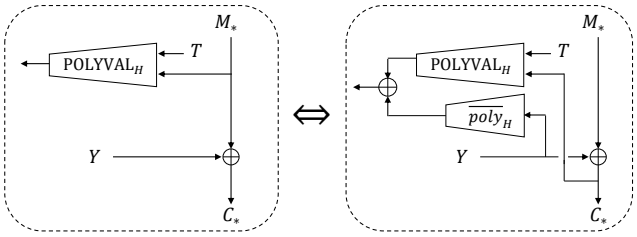

Fig. 8. Decomposition of $\mathsf{POLYVAL}_H(T, M_*)$. Left is a part of the standard HCTR2 description that can be seen in Fig. 1. Right is the description used in the attack.

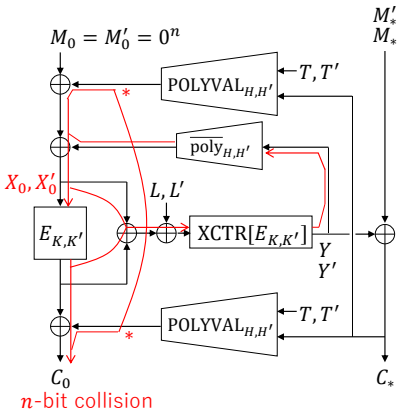

Fig. 9. $O(2^{n/2})$ attack for $\mathsf{EtE_P}$-HCTR2.

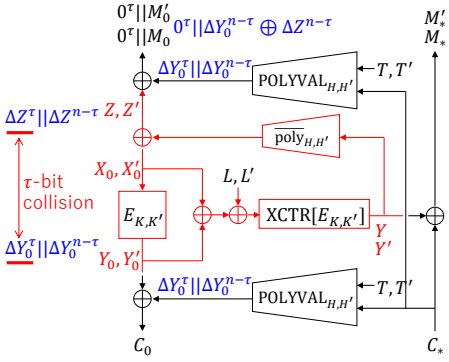

Fig. 10. $O(2^{\tau/2})$ Attack for $\mathsf{EtE_P}$-HCTR2. The red color is related to the collision generation, and the blue color denotes differences.

then $M_* \leftarrow Y \oplus C_*$ and $M_*' \leftarrow Y' \oplus C_*$ are computed, respectively. The output of POLYVAL is controlled by inverting POLYVAL thanks to the unfixed T, T'.

For $\tau = n$, this achieves the birthday-bound attack. However, even for $\tau < n$, the complexity stays $O(2^{n/2})$. This led Chen et al. [10] to conclude that $\mathsf{CMT_k}$ security of $\mathsf{EtE_A}$-HCTR2 is at most $O(2^{\tau/2})$ while $\mathsf{CMT_k}$ security of $\mathsf{EtE_P}$-HCTR2 is at most $O(2^{n/2})$. Although it is not explained in [10], when $\tau < n/2$, the attack is worse than the trivial $O(2^\tau)$ attack; for a fixed C, T, find K such that $\mathsf{msb}_\tau(M)$ are 0^τ. The correct attack complexity is given by $O(2^{\min\{n/2,\tau\}})$.

New Attack on $\mathsf{EtE_P}$-HCTR2. The previous attack requires a complexity of $O(2^{n/2})$ even if $\tau < n$. We observe that one potential improvement is to avoid fixing M_0 to 0^n for $\tau < n$ allowing the attacker to exploit the property that $n-\tau$ bits of M_0 can be chosen freely. To exploit the property, we carefully analyze the impact of two hash layers that depend on each other.

Our analysis is depicted in Fig. 10. As the previous attack, we first fix K, K'. Then, BC's input X_0 is chosen, which fixes BC's output Y_0 and XCTR's output Y, and then $\overline{\mathsf{poly}}_H(Y)$ in the first hash layer is independently computed from

$\mathsf{POLYVAL}_H(T, C_*)$. Let Z denote the value of $X_0 \oplus \overline{\mathsf{poly}}_H(Y)$. Similarly, for any X_0', the corresponding Y_0', Y', and Z' are computed. Let Δ denote the difference between the values derived from X_0 and X_0'. Also, let Δh denote the output difference from $\mathsf{POLYVAL}$, i.e. $\Delta h = \mathsf{POLYVAL}_H(T, C_*) \oplus \mathsf{POLYVAL}_{H'}(T, C_*)$. Δh is common to both the first and second hash layers. The goal is to ensure $\Delta C_0 = 0$ while maintaining the 0^τ padding in M_0 and M_0'. From the second hash layer, we have $\Delta Y_0 = \Delta h$ to achieve $\Delta C_0 = 0$. Then from the first hash layer, the condition with respect to the difference is

$$\big(\mathsf{msb}_\tau(\Delta Z) = \mathsf{msb}_\tau(\Delta Y_0)\big) \wedge \big(\mathsf{lsb}_{n-\tau}(\Delta Z) \oplus \mathsf{lsb}_{n-\tau}(\Delta Y_0) = \mathsf{lsb}_{n-\tau}(\Delta M_0)\big).$$

The τ-bit condition for the MSB can only be satisfied probabilistically since both Z and Y_0 behave as random values determined by X_0, while the $n-\tau$-bit condition for the LSB can be deterministically satisfied by properly choosing M_0 and M_0'. Therefore, the above condition on differences can be satisfied with $O(2^{\tau/2})$ complexity by searching for a collision between $\mathsf{msb}_\tau(\Delta Z)$ and $\mathsf{msb}_\tau(\Delta Y_0)$. Regarding the values of 0^τ, after Z, Z', M_0, M_0' are fixed, 0^τ can be ensured by inverting $\mathsf{POLYVAL}$ in the first hash layer.

The pseudo-code of our attack is given in Alg. 3. The message length can be arbitrary; however, for simplicity, we fix it such that the total length becomes $2n$ bits after encoding the τ bits of zeros. Namely, the input message consists of $n-\tau$ bits of M_0 and n bits of M_*. The corresponding output is denoted by $C_0 \| C_*$. The tweak length can also be any, but we fix it to $2n$ bits for simplicity.

Regarding the success probability, the probabilistic part of the attack is to generate a collision on τ bits between $2^{\tau/2}$ values computed for one key and $2^{\tau/2}$ values computed for the other key. The success probability is $1 - e^{-1} \approx 0.63$. Note that this success probability is the same as the previous $O(2^{n/2})$ attack.

4.4 Experimental Verification

We verify the proposed attacks by experimentally generating vectors for $\mathsf{HCTR2}$ instantiated with $\mathsf{AES\text{-}128}$. Our implementations of Algorithms 2 and 3 optimize the basic $\mathsf{HCTR2}$ components using the AVX instructions, including AES-NI [19] and CLMUL [20]. We run it on a single core of the Intel Xeon w5-3423 processor, while measuring the wall-clock time, and the obtained input vectors are further verified with the original $\mathsf{HCTR2}$ implementation by Google [17]. The concrete vectors are shown in Appendix B.

Particular collision-finding tasks (the lines 1–6 in Algorithm 2 and the lines 4–14 in Algorithm 3) are optimized using Floyd's cycle-finding algorithm [28] to reduce costly storage access. Given a target function F and an initial value x_0, we incrementally search $i > 0$ satisfying $F^{(i)}(x_0) = F^{(2i)}(x_0)$, wherein $F^{(i)}(\cdot)$ denotes applications of F for i times. Such an $F^{(i)}(x_0)$ is in the middle of the cycle in the sequence obtained by repeated applications of F, and we can find the beginning of the cycle with the minimum integer j satisfying $F^{(j)}(x_0) = F^{(i+j)}(x_0)$. As a result, $F^{(j-1)}(x_0) \neq F^{(i+j-1)}(x_0)$ is a colliding pair.

Our Algorithm 2 implementation finds $K \neq K'$ satisfying $\mathsf{lsb}_{2\tau/3}\big(E_K(\mathtt{bin}_n(1))\big) = \mathsf{lsb}_{2\tau/3}\big(E_{K'}(\mathtt{bin}_n(1))\big)$ using $F(x) = \mathsf{lsb}_{2\tau/3}\big(E_{0^{n-2\tau/3}\|x}(\mathtt{bin}_n(1))\big)$. In order

Algorithm 3 $O(2^{\tau/2})$ attack against $\mathsf{EtE_P}$-$\mathsf{HCTR2}$ with $\tau \leq n$

Require: τ satisfying $\tau \leq n$
Ensure: (K, T, M), (K', T, M')
s.t. $\mathsf{HCTR2.enc}[E_K](T, 0^\tau \| M) = \mathsf{HCTR2.enc}[E_{K'}](T, 0^\tau \| M')$

1: Set the message length to $2n - \tau$ bits and the tweak length to $2n$ bits.
2: Fix $K, K' \in \{0,1\}^k$, $M_0 \in \{0,1\}^{n-\tau}$, $C_* \in \{0,1\}^n$ to arbitrarily chosen values.
3: $H \leftarrow E_K(\mathtt{bin}_n(0))$; $L \leftarrow E_K(\mathtt{bin}_n(1))$; $H' \leftarrow E_{K'}(\mathtt{bin}_n(0))$; $L' \leftarrow E_{K'}(\mathtt{bin}_n(1))$

4: **for** $i = 1, \ldots, 2^{\tau/2}$ **do**
5: $\quad X_0^{(i)} \xleftarrow{\$} \{0,1\}^n$; $Y_0^{(i)} \leftarrow E_K(X_0^{(i)})$; $Y^{(i)} \leftarrow E_K(X_0^{(i)} \oplus Y_0^{(i)} \oplus L \oplus \mathtt{bin}_n(1))$
6: $\quad Z^{(i)} \leftarrow X_0^{(i)} \oplus \overline{\mathsf{poly}}_H(Y^{(i)})$
7: $\quad$ Compute $\mathsf{msb}_\tau(Y_0^{(i)} \oplus Z^{(i)})$ and store it along with $Y_0^{(i)}$, $Z^{(i)}$, $Y^{(i)}$.
8: **end for**
9: **for** $j = 1, \ldots, 2^{\tau/2}$ **do**
10: $\quad X_0'^{(j)} \xleftarrow{\$} \{0,1\}^n$; $Y_0'^{(j)} \leftarrow E_{K'}(X_0'^{(j)})$; $Y'^{(j)} \leftarrow E_{K'}(X_0'^{(j)} \oplus Y_0'^{(j)} \oplus L' \oplus \mathtt{bin}_n(1))$
11: $\quad Z'^{(j)} \leftarrow X_0'^{(j)} \oplus \overline{\mathsf{poly}}_{H'}(Y'^{(j)})$
12: $\quad$ Compute $\mathsf{msb}_\tau(Y_0'^{(j)} \oplus Z'^{(j)})$ and store it along with $Y_0'^{(j)}$, $Z'^{(j)}$, $Y'^{(j)}$.
13: **end for**
14: Find $i*$ and $j*$ such that $\mathsf{msb}_\tau(Y_0^{(i*)} \oplus Z^{(i*)}) = \mathsf{msb}_\tau(Y_0'^{(j*)} \oplus Z'^{(j*)})$.
15: $(Y_0, Z, Y) \leftarrow (Y_0^{(i*)}, Z^{(i*)}, Y^{(i*)})$, $(Y_0', Z', Y') \leftarrow (Y_0^{(j*)}, Z^{(j*)}, Y^{(j*)})$
16: $M_* \leftarrow Y \oplus C_*$; $M_*' \leftarrow Y' \oplus C_*$

17: $M_0' \leftarrow M_0 \oplus \Delta Y_0^{n-\tau} \oplus \Delta Z^{n-\tau}$
18: Find T to satisfy both $Z \oplus 0^\tau \| M_0 = \mathsf{POLYVAL}_H(T, C_*)$ and $Z' \oplus 0^\tau \| M_0' = \mathsf{POLYVAL}_{H'}(T, C_*)$ by inverting $\mathsf{POLYVAL}$.
19: $C_0 \leftarrow Y_0 \oplus \mathsf{POLYVAL}_H(T, C_*)$
20: **return** $(K, T, M_0 \| M_*)$, $(K', T, M_0' \| M_*')$

to implement Algorithm 3 as a cycle-finding problem, we modify Algorithm 3 so that we search for values of K and K' while fixing both X_0 and X_0' to $\mathtt{bin}_n(1)$. Then, the function for the cycle-finding algorithm is defined as $F(x) = \mathsf{msb}_\tau\big(\overline{\mathsf{poly}}_{E_{0^{n-\tau}\|x}(\mathtt{bin}_n(0))}(E_{0^{n-\tau}\|x}(\mathtt{bin}_n(0)))\big)$. Since we have $Y = E_K\big(\mathtt{bin}_n(1) \oplus E_K(\mathtt{bin}_n(1)) \oplus L \oplus \mathtt{bin}_n(1)\big) = E_K(\mathtt{bin}_n(0))$, one can see $\mathsf{msb}_\tau(Y_0 \oplus Z) = \mathsf{msb}_\tau(Y_0' \oplus Z')$ when $K = 0^{n-\tau}\|x, K' = 0^{n-\tau}\|x'$ such that $F(x) = F(x')$.

For the attack on $\mathsf{EtE_A}$-$\mathsf{HCTR2}$ with $\frac{\tau}{3}$-bit complexity, we target $\tau = 96$ bits, which is a tag length supported by AES-GCM in NIST SP800-38D [15], and the entire computation ends in 652 seconds. For the attack on $\mathsf{EtE_P}$-$\mathsf{HCTR2}$ with $\frac{\tau}{2}$-bit complexity, on the other hand, we target $\tau = 64$ bits considering a short tag included in several industrial standards, such as 5G [1], AES-CCM in NIST SP800-38C [14], and the Internet of Things (IoT) and media encryption applications [8], and the concrete vectors are found successfully in 347 seconds.

5 Key Committing Security of EtE-HCTR2

5.1 Security Results

We give the $\mathsf{CMT_k}$-Security bounds of $\mathsf{EtE_S}$-HCTR2, approximately $\max\{\frac{p^2}{2^\tau}, \frac{p}{2^n}\}$, and of $\mathsf{EtE_P}$-HCTR2, approximately $\frac{p^2}{2^{\min\{\tau,n\}}}$, in the IC model that are respectively given in Theorems 1 and 2. The bounds show that $\mathsf{EtE_S}$-HCTR2 (resp. $\mathsf{EtE_P}$-HCTR2) are respectively $\mathsf{CMT_k}$-secure up to about $2^{\frac{\min\{\tau,2n\}}{2}}$ (resp. $2^{\frac{\min\{\tau,n\}}{2}}$) queries. The proof of Theorem 1 (resp. 2) is given in Section 6 (resp. 7).

Theorem 1 ($\mathsf{EtE_S}$-HCTR2). *For any computationally-unbounded* $\mathsf{CMT_k}$ *adversary against* EtE-HCTR2 *making* p *queries,* $\mathbf{Adv}^{\mathsf{cmt_k}}_{\mathsf{EtE_S}\text{-}\mathsf{HCTR2}}(\mathbf{A}) \le \epsilon$ *such that*

if $\tau \le n$, $\epsilon = \dfrac{5p^2}{2^\tau - \frac{p}{2^{n-\tau}}}$;

if $n < \tau \le 2n$, $\epsilon = \dfrac{5n}{(1-\frac{4p}{2^n})\log_2 n} \cdot \dfrac{p^2}{2^\tau} + \dfrac{2}{1-\frac{2p}{2^n}} \cdot \dfrac{p}{2^n} + \left(\dfrac{6(\log_2 n)^2}{n(1-\frac{2p}{2^n})} \cdot \dfrac{p}{2^n}\right)^{\frac{n}{\log_2 n}}$.

Theorem 2 ($\mathsf{EtE_P}$-HCTR2). *Assume that* $n \ge 2$ *and the irreducible polynomial* $p(\mathtt{x})$ *of* POLYVAL *contains* $\mathtt{x}^{n-1}$. *For any computationally-unbounded* $\mathsf{CMT_k}$ *adversary against* $\mathsf{EtE_P}$-HCTR2 *making* p *queries,*

$$\mathbf{Adv}^{\mathsf{cmt_k}}_{\mathsf{EtE_P}\text{-}\mathsf{HCTR2}}(\mathbf{A}) \le \frac{2n^2}{(1-\frac{2p}{2^n})\log_2 n} \cdot \frac{p^2}{2^{\min\{\tau,n\}} - \frac{2p}{2^{n-\tau}}} + \left(\frac{6\log_2 n}{\sqrt{n}(1-\frac{2p}{2^n})} \cdot \frac{p}{2^{n/2}}\right)^{\frac{n}{\log_2 n}}.$$

5.2 Definitions for Proofs of Theorems 1 and 2

Let $\mathcal{Q}$ be the set of all query-response tuples of $E^{\pm} = (E, E^{-1})$. Let $\mathcal{Q}^{\mathsf{fwd}}$ (resp. $\mathcal{Q}^{\mathsf{inv}}$) be the set of query-response tuples in $\mathcal{Q}$ that are defined by forward (resp. inverse) queries. For each $K \in \{0,1\}^k$, let $\mathcal{Q}_K \coloneqq \{(X,Y) \mid (K,X,Y) \in \mathcal{Q}\}$ and $p_K \coloneqq |\mathcal{Q}_K|$. For each $K \in \{0,1\}^k$, let $\mathcal{Q}^{\mathsf{fwd}}_K \coloneqq \{(X,Y) \mid (K,X,Y) \in \mathcal{Q}^{\mathsf{fwd}}\}$ and $\mathcal{Q}^{\mathsf{inv}}_K \coloneqq \{(X,Y) \mid (K,X,Y) \in \mathcal{Q}^{\mathsf{inv}}\}$. For $(\dot{K},\dot{X},\dot{Y}), (\ddot{K},\ddot{X},\ddot{Y}) \in \mathcal{Q}$, let $(\dot{K},\dot{X},\dot{Y}) \prec (\ddot{K},\ddot{X},\ddot{Y})$ be the condition that $(\ddot{K},\ddot{X},\ddot{Y})$ is defined after $(\dot{K},\dot{X},\dot{Y})$ is defined and $(\dot{K},\dot{X},\dot{Y}) \preceq (\ddot{K},\ddot{X},\ddot{Y})$ be a condition that $(\dot{K},\dot{X},\dot{Y}) \prec (\ddot{K},\ddot{X},\ddot{Y}) \vee (\dot{K},\dot{X},\dot{Y}) = (\ddot{K},\ddot{X},\ddot{Y})$ holds. Note that K is omitted if it is clear from the context. Let $\mu \coloneqq \frac{n}{\log_2 n}$.

6 Proof of Theorem 1: $\mathsf{CMT_k}$ Security of $\mathsf{EtE_S}$-HCTR2

We consider the partial function of $\mathsf{EtE_S}$-HCTR2 shown in Algorithm 4 and Fig. 4(left). We call the partial function "DMXCTR". DMXCTR consists of two parts: DM and XCTR with two blocks.

Algorithm 4 DMXCTR

Function $\mathsf{DMXCTR}[E](K, X)$

1: $L \leftarrow E_K(\mathtt{bin}_n(1))$; $Y \leftarrow E_K(X)$; $IV \leftarrow X \oplus Y$
2: **for** $i \in [2]$ **do** $X_i \leftarrow IV \oplus L \oplus \mathtt{bin}_n(i)$; $Y_i \leftarrow E_K(X_i)$ **end for**
3: $S \leftarrow \mathsf{msb}_{\min\{\tau,2n\}}(Y_1 \| Y_2)$; **return** S

The following lemma shows that the CMT_k security of $\mathsf{EtE_S}$-HCTR2 is reduced to the collision resistance of DMXCTR. The advantage function of an adversary $\mathbf{B}$ that breaks the collision resistance of DMXCTR is defined as

$$\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{DMXCTR}}(\mathbf{B}) \coloneqq \Pr\big[((K^\dagger, X^\dagger), (K^\ddagger, X^\ddagger)) \leftarrow \mathbf{B}^{E^\pm} \text{ s.t. } \mathsf{DMXCTR}[E](K^\dagger, X^\dagger) = \mathsf{DMXCTR}[E](K^\ddagger, X^\ddagger) \wedge K^\dagger \neq K^\ddagger\big].$$

In $\mathsf{EtE_S}$-HCTR2, the first τ bits of M_* of HCTR2 are zeros, and thus the first $\min\{\tau, 2n\}$ bits of C_* are equal to BC's outputs $\mathsf{msb}_{\min\{\tau,2n\}}(Y_1 \| Y_2)$ in XCTR. Hence, the CMT_k security of $\mathsf{EtE_S}$-HCTR2 is reduced to the collision resistance of DMXCTR.

Lemma 1. *For any* CMT_k *adversary against* $\mathsf{EtE_S}$-HCTR2 *that makes* p *queries, there exists a collision-finding adversary* $\mathbf{B}$ *against* DMXCTR *making* p *queries such that*

$$\mathbf{Adv}^{\mathsf{cmt_k}}_{\mathsf{EtE_S}\text{-}\mathsf{HCTR2}}(\mathbf{A}) \leq \mathbf{Adv}^{\mathsf{coll}}_{\mathsf{DMXCTR}}(\mathbf{B}).$$

The following lemma shows the bounds of the collision resistance of DMXCTR.

Lemma 2. *For any adversary* $\mathbf{B}$ *against* DMXCTR *making* p *queries,* $\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{DMXCTR}}(\mathbf{B}) \leq \epsilon$ *such that*

if $\tau \leq n$, $\epsilon = \dfrac{5p^2}{2^\tau - \frac{p}{2^{n-\tau}}}$;

if $n < \tau \leq 2n$, $\epsilon = \dfrac{5n}{(1 - \frac{4p}{2^n}) \log_2 n} \cdot \dfrac{p^2}{2^\tau} + \dfrac{2}{1 - \frac{2p}{2^n}} \cdot \dfrac{p}{2^n} + \left(\dfrac{6(\log_2 n)^2}{n(1 - \frac{2p}{2^n})} \cdot \dfrac{p}{2^n} \right)^{\frac{n}{\log_2 n}}$.

Combining these lemmas, we have the bounds in Theorem 1.

6.1 Proof of Lemma 1

Assume that $\mathbf{A}$ is a CMT_k adversary against $\mathsf{EtE_S}$-HCTR2 making at most p primitive queries. We construct a collision-finding adversary $\mathbf{B}$ against DMXCTR making at most p primitive queries. The adversary $\mathbf{B}$ is defined in Algorithm 5. The adversary $\mathbf{B}$ simulates $\mathbf{A}$'s IC by using $\mathbf{B}$'s IC $E^\pm$. After finishing all queries of $\mathbf{A}$, $\mathbf{B}$ calculates the internal values $X_0^\dagger$ and $X_0^\ddagger$ of HCTR2.enc with $\mathbf{A}$'s outputs and $\mathbf{A}$'s query-response tuples.

Since

$$C^\dagger = C^\ddagger \Rightarrow \mathsf{msb}_\tau(C_\tau^\dagger) = \mathsf{msb}_\tau(C^\ddagger) \Rightarrow \mathsf{msb}_\tau(Y_1^\dagger \| Y_2^\dagger) = \mathsf{msb}_\tau(Y_1^\ddagger \| Y_2^\ddagger),$$

Algorithm 5 Collision Finding Adversary $\mathbf{B}$ with a CMT_k adversary $\mathbf{A}$

For $\mathbf{A}$'s query $(\tilde{K}, \tilde{X})$ to E, $\mathbf{B}$ returns $\tilde{X}$ to $\mathbf{A}$, which is defined as follows.

1: $\mathbf{B}$ makes the query $(\tilde{K}, \tilde{X})$ to E and obtains the response $\tilde{Y}$.

For $\mathbf{A}$'s query $(\tilde{K}, \tilde{Y})$ to E^{-1}, $\mathbf{B}$ returns $\tilde{Y}$ to $\mathbf{A}$, which is defined as follows.

1: $\mathbf{B}$ makes the query $(\tilde{K}, \tilde{Y})$ to E^{-1} and $\mathbf{B}$ obtains the response $\tilde{X}$.

For $\mathbf{A}$'s output tuple $((K^\dagger, T^\dagger, M^\dagger), (K^\ddagger, T^\ddagger, M^\ddagger))$, $\mathbf{B}$ returns pairs $((K^\dagger, X^\dagger), (K^\ddagger, X^\ddagger))$, which is defined as follows.

1: $\mathbf{B}$ calculates the internal values X_0 of $\mathsf{HCTR2.enc}[E_{K^\dagger}](T^\dagger, M_0^\dagger \| 0^\tau \| M_r^\dagger)$ and $X_0^\ddagger$ of $\mathsf{HCTR2.enc}[E_{K^\ddagger}](T^\ddagger, M_0^\ddagger \| 0^\tau \| M_r^\ddagger))$, respectively, by using the query-response tuples from $\mathbf{A}$ to $E^{\pm}$

2: $(X^\dagger, X^\ddagger) \leftarrow (X_0^\dagger, X_0^\ddagger)$

if $\mathbf{A}$ breaks the CMT_k security of $\mathsf{EtE_S}$-HCTR2, then $\mathbf{B}$ finds a collision of DMXCTR.

□

6.2 Proof of Lemma 2

Without loss of generality, we assume that $\mathbf{B}$ is deterministic and does not make repeated queries. We use the definitions in Section 5.2.

Complete Set for DMXCTR. For $(K, X, X) \in \mathcal{Q}$, we define a complete set $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})$ that keeps internal values of $\mathsf{DMXCTR}[E](K, X)$.

$$\begin{aligned}\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}) \coloneqq \{&(0, X, Y), (1, X_1, Y_1), \boxed{(2, X_2, Y_2)} \mid \\ &X_1 = X \oplus Y \oplus L \oplus \mathtt{bin}_n(1) \wedge \boxed{X_2 = X \oplus Y \oplus L \oplus \mathtt{bin}_n(2) \wedge} \\ &(\mathtt{bin}_n(0), L), (X, Y), (X_1, Y_1), \boxed{(X_2, Y_2)} \in \mathcal{Q}_K\},\end{aligned}$$

where if $\tau \leq n$ (resp. $\tau > n$), the boxed statements are removed (resp. included). See Fig. 4(left) for the correspondence between the structure of DMXCTR and the complete set. Let $f_{\mathsf{msb}_\tau}(\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})) \coloneqq \mathsf{msb}_\tau(Y_1 \| Y_2)$ be a function that returns an output of $\mathsf{DMXCTR}[E](K, X)$. For $i \in [2]$, let $f_i(\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})) \coloneqq Y_i$.

Deriving the Bound of Lemma 2. We first define a condition coll.

$$\begin{aligned}\mathsf{coll} \Leftrightarrow \exists (\dot{K}, \dot{X}, \dot{Y}), (\ddot{K}, \ddot{X}, \ddot{Y}) \in \mathcal{Q} \text{ s.t.} \\ \dot{K} \neq \ddot{K} \wedge f_{\mathsf{msb}_\tau}(\mathcal{S}^{\mathsf{comp}}_{(\dot{K},\dot{X})}(\mathcal{Q})) = f_{\mathsf{msb}_\tau}(\mathcal{S}^{\mathsf{comp}}_{(\ddot{K},\ddot{X})}(\mathcal{Q})).\end{aligned}$$

By using the event, we have $\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{DMXCTR}}(\mathbf{B}) \leq \Pr[\mathsf{coll}]$.

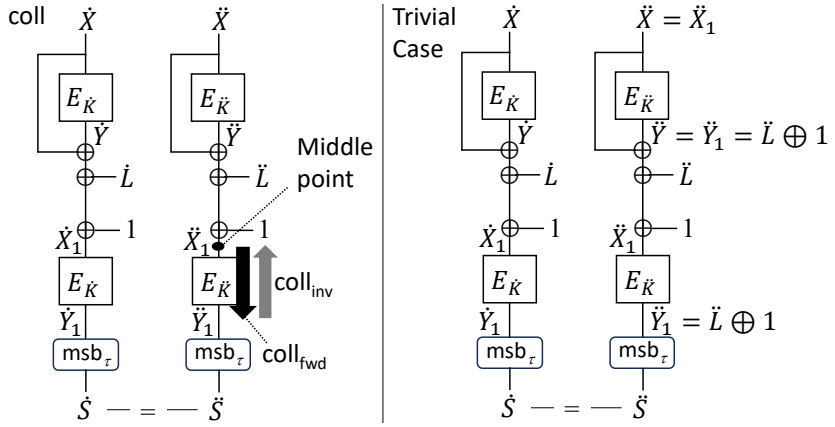

Fig. 11. coll with $\tau \leq n$ (left) and trivial case in $\mathsf{coll}_{\mathsf{inv}}$ with $\tau \leq n$ (right).

In Section 6.3 (resp. Section 6.4), the bound with $\tau \leq n$ (resp. $n < \tau \leq 2n$) is derived. Let $\mathcal{S}^{\mathsf{comp}}_{(\dot{K},\dot{X})}(\mathcal{Q}) = \{(0,\dot{X},\dot{Y}),(1,\dot{X}_1,\dot{Y}_1),\boxed{(2,\dot{X}_2,\dot{Y}_2)}\}$ and $\mathcal{S}^{\mathsf{comp}}_{(\ddot{K},\ddot{X})}(\mathcal{Q}) = \{(0,\ddot{X},\ddot{Y}),\ (1,\ddot{X}_1,\ddot{Y}_1),\boxed{(2,\ddot{X}_2,\ddot{Y}_2)}\}$.

6.3 Upper-Bounding $\Pr[\mathsf{coll}]$ with $\tau \leq n$

We first define the following two events such that $\mathsf{coll} = \mathsf{coll}_{\mathsf{fwd}} \vee \mathsf{coll}_{\mathsf{inv}}$:

$$\mathsf{coll}_{\mathsf{fwd}} \Leftrightarrow \mathsf{coll} \wedge (\ddot{X}_1,\ddot{Y}_1) \in \mathcal{Q}^{\mathsf{fwd}}_{\ddot{K}} \qquad \mathsf{coll}_{\mathsf{inv}} \Leftrightarrow \mathsf{coll} \wedge (\ddot{X}_1,\ddot{Y}_1) \in \mathcal{Q}^{\mathsf{inv}}_{\ddot{K}}.$$

Fig. 11(left) shows these events. Without loss of generality, we assume that $(\dot{K}_1,\dot{X}_1,\dot{Y}_1) \prec (\ddot{K}_1,\ddot{X}_1,\ddot{Y}_1)$. By using these events, we have

$$\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{DMXCTR}}(\mathbf{B}) \leq \Pr[\mathsf{coll}] \leq \Pr[\mathsf{coll}_{\mathsf{fwd}}] + \Pr[\mathsf{coll}_{\mathsf{inv}}].$$

Evaluation for $\mathsf{coll}_{\mathsf{fwd}}$. Since $\ddot{Y}_1$ is chosen uniformly at random from at least $2^n - p$ elements, we have

$$\begin{aligned}\Pr[\mathsf{coll}_{\mathsf{fwd}}] &\leq \sum_{\substack{\{\dot{K},\ddot{K}\}\in \\ \mathsf{UConf}_2(\{0,1\}^k)}} p_{\dot{K}}p_{\ddot{K}} \cdot \sum_{V\in\{0,1\}^{n-\tau}} \Pr\begin{bmatrix}\dot{Y}_1 \oplus \ddot{Y}_1 \\ = (0^\tau \| V)\end{bmatrix} \\ &\leq \sum_{\substack{\{\dot{K},\ddot{K}\}\in \\ \mathsf{UConf}_2(\{0,1\}^k)}} \frac{p_{\dot{K}}p_{\ddot{K}} \cdot 2^{n-\tau}}{2^n - p}.\end{aligned}$$

Evaluation for $\mathsf{coll}_{\mathsf{inv}}$. We evaluate $\Pr[\mathsf{coll}_{\mathsf{inv}}]$, the probability that for some pairs $\big((\dot{K},\dot{X},\dot{Y}),(\dot{K},\dot{X}_1,\dot{Y}_1)\big)$ and $\big((\ddot{K},\ddot{X},\ddot{Y}),(\ddot{K},\ddot{X}_1,\ddot{Y}_1)\big)$ such that $\dot{K}\neq\ddot{K}$, $(\ddot{K},\ddot{X}_1,\ddot{Y}_1)\in\mathcal{Q}^{\mathsf{inv}}$ and $(\dot{K}_1,\dot{X}_1,\dot{Y}_1)\prec(\ddot{K}_1,\ddot{X}_1,\ddot{Y}_1)$, the pairs are complete set, and a collision occurs, i.e., $f_{\mathsf{msb}_\tau}(\mathcal{S}^{\mathsf{comp}}_{(\dot{K},\dot{X})}(\mathcal{Q}))=f_{\mathsf{msb}_\tau}(\mathcal{S}^{\mathsf{comp}}_{(\ddot{K},\ddot{X})}(\mathcal{Q}))$.

We introduce the following event: $\mathsf{hit}_1\Leftrightarrow\exists(*,\mathtt{bin}_n(1),*)\in\mathcal{Q}^{\mathsf{inv}}$, wherein a mask L is defined by an inverse query. For each $(K,X,Y)\in\mathcal{Q}^{\mathsf{inv}}$, X is chosen from at least 2^n-p elements, we have $\Pr[\mathsf{hit}_1]\leq\frac{p}{2^n-p}$. Hereafter, we assume that hit_1 does not occur, ensuring that each mask L is defined by a forward query.

In this evaluation, we consider a trivial case where the XCTR part is trivially connected with the DM part,[12] i.e., the pair $((\ddot{K},\ddot{X},\ddot{Y}),(\ddot{K}_1,\ddot{X}_1,\ddot{Y}_1))$ satisfies $\mathsf{C}_{\mathsf{triv}}\coloneqq((\ddot{X}_1,\ddot{Y}_1)=(\ddot{X},\ddot{Y})\wedge\ddot{Y}_1=\ddot{L}\oplus\mathtt{bin}_n(1))$. Fig. 11(right) shows the trivial case. We then evaluate the probability that $\mathsf{coll}_{\mathsf{inv}}$ occurs below.

- $\mathsf{coll}_{\mathsf{inv}}\wedge\neg\mathsf{C}_{\mathsf{triv}}$: The probability of $\mathsf{coll}_{\mathsf{inv}}\wedge\neg\mathsf{C}_{\mathsf{triv}}$ is reduced into a probability p_1 that $\mathcal{S}^{\mathsf{comp}}_{(\ddot{K},\ddot{X})}(\mathcal{Q})$ is defined by connecting the XCTR part with the DM part at the middle point, i.e.,
$$p_1=\Pr[\ddot{X}_1=\ddot{X}\oplus\ddot{Y}\oplus\ddot{L}\oplus\mathtt{bin}_n(1)\mid(\ddot{K},\ddot{X},\ddot{Y})\in\mathcal{Q}^{\mathsf{inv}}\wedge\neg\mathsf{C}_{\mathsf{triv}}].$$
  - If $(\ddot{X},\ddot{Y})=(\mathtt{bin}_n(1),\ddot{L})$, since $\ddot{X}_1=\ddot{X}\oplus\ddot{Y}\oplus\ddot{L}\oplus\mathtt{bin}_n(1)\Rightarrow\ddot{X}_1=0^n$, and $\dot{X}_1$ is chosen from at least 2^n-p elements, we have $p_1\leq\sum_{\ddot{K}\in\{0,1\}^k}p_{\ddot{K}}\cdot\frac{1}{2^n-p}$.
  - If $(\ddot{X},\ddot{Y})\neq(\mathtt{bin}_n(1),\ddot{L})\wedge(\ddot{X}_1,\ddot{Y}_1)\prec(\ddot{X},\ddot{Y})$, since $\ddot{L}$ and $\ddot{X}$ or $\ddot{Y}$ are chosen from at least 2^n-p elements, we have $p_1\leq\sum_{\ddot{K}\in\{0,1\}^k}p_{\ddot{K}}(p_{\ddot{K}}-1)\cdot\frac{1}{2^n-p}$.
  - If $(\ddot{X},\ddot{Y})\prec(\ddot{X}_1,\ddot{Y}_1)$, since $\ddot{X}_1$ and $\ddot{L}$ are chosen from at least 2^n-p elements, we have $p_1\leq\sum_{\ddot{K}\in\{0,1\}^k}p_{\ddot{K}}(p_{\ddot{K}}-1)\cdot\frac{1}{2^n-p}$.
  - If $(\ddot{X},\ddot{Y})=(\ddot{X}_1,\ddot{Y}_1)$, $\ddot{X}_1=\ddot{X}\oplus\ddot{Y}\oplus\ddot{L}\oplus\mathtt{bin}_n(1)\Rightarrow\ddot{Y}_1=\ddot{L}\oplus\mathtt{bin}_n(1)$ but $\ddot{Y}_1=\ddot{L}\oplus W$ s.t. $W\neq\mathtt{bin}_n(1)$ by $\neg\mathsf{C}_{\mathsf{triv}}$. We thus have $p_1=0$.
- $\mathsf{coll}_{\mathsf{inv}}\wedge\mathsf{C}_{\mathsf{triv}}$: We evaluate p_2, the probability of $\mathsf{coll}_{\mathsf{inv}}\wedge\mathsf{C}_{\mathsf{triv}}$ as follows.
  - If $(\dot{X}_1,\dot{Y}_1)\in\mathcal{Q}^{\mathsf{fwd}}_{\ddot{K}}$, since $\mathsf{msb}_\tau(\dot{Y}_1)=\mathsf{msb}_\tau(\ddot{Y}_1)\Rightarrow\mathsf{msb}_\tau(\dot{Y}_1)=\mathsf{msb}_\tau(\ddot{L}\oplus\mathtt{bin}_n(1))$, and $\dot{Y}_1$ and $\ddot{L}$ are chosen from at least 2^n-p elements, we have $p_2\leq\sum_{\dot{K}\neq\ddot{K}}p_{\dot{K}}p_{\ddot{K}}\cdot\frac{2^{n-\tau}}{2^n-p}$.
  - If $(\dot{X}_1,\dot{Y}_1)\in\mathcal{Q}^{\mathsf{inv}}_{\ddot{K}}$ such that $\dot{Y}_1=\dot{L}\oplus\mathtt{bin}_n(1)$, since $\mathsf{msb}_\tau(\dot{Y})=\mathsf{msb}_\tau(\ddot{Y})\Rightarrow\mathsf{msb}_\tau(\dot{L})=\mathsf{msb}_\tau(\ddot{L})$, and $\dot{L}$ and $\ddot{L}$ are chosen from at least 2^n-p elements, we have $p_2\leq\sum_{\{\dot{K},\ddot{K}\}\in\mathsf{UConf}_2(\{0,1\}^k)\text{ s.t. }p_{\dot{K}},p_{\ddot{K}}>0}\frac{2^{n-\tau}}{2^n-p}$.
  - If $(\dot{X}_1,\dot{Y}_1)\in\mathcal{Q}^{\mathsf{inv}}_{\ddot{K}}$ such that $\dot{Y}_1\neq\dot{L}\oplus\mathtt{bin}_n(1)$, then the evaluation is the same as that of $\mathsf{coll}_{\mathsf{inv}}\wedge\neg\mathsf{C}_{\mathsf{triv}}$ given above. We thus have $p_2\leq\sum_{\dot{K}\in\{0,1\}^k}\frac{2p^2_{\dot{K}}-p_{\dot{K}}}{2^n-p}$

[12] The same condition appears in Proposition 1 in the attack section (Section 4.2).

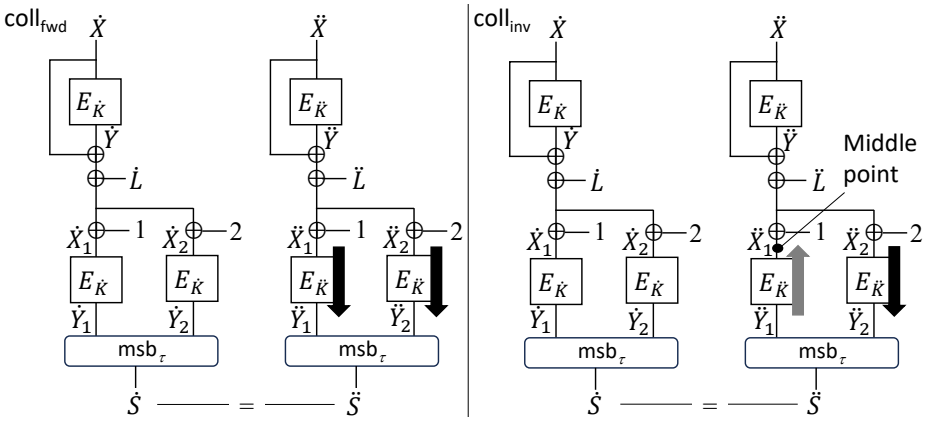

Fig. 12. coll with $n < \tau \leq 2n$: $\mathsf{coll}_{\mathsf{fwd}}$ (left) and $\mathsf{coll}_{\mathsf{inv}}$ (right).

Summing these bounds, we have

$$\Pr[\mathsf{coll}_{\mathsf{inv}}] \leq \frac{p}{2^n - p} + \sum_{\ddot{K} \in \{0,1\}^k} \frac{2p_{\ddot{K}}^2 - p_{\ddot{K}}}{2^n - p} + \sum_{\substack{\{\dot{K}, \ddot{K}\} \in \mathsf{UConf}_2(\{0,1\}^k) \\ \text{s.t. } p_{\dot{K}} p_{\ddot{K}} > 0}} \frac{2^{n-\tau}(p_{\dot{K}} p_{\ddot{K}} + 1)}{2^n - p}$$
$$+ \sum_{\ddot{K} \in \{0,1\}^k} p_{\ddot{K}}^2 \cdot \frac{1}{2^n - p} + \sum_{\dot{K} \in \{0,1\}^k} \frac{2p_{\dot{K}}^2 - p_{\dot{K}}}{2^n - p}.$$

Conclusion of the Evaluation. By using the above bounds, we have

$$\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{DMXCTR}}(\mathbf{B}) \leq \sum_{\substack{\{\dot{K}, \ddot{K}\} \in \\ \mathsf{UConf}_2(\{0,1\}^k)}} \frac{3p_{\dot{K}} p_{\ddot{K}}}{2^\tau - \frac{p}{2^{n-\tau}}} + \sum_{K \in \{0,1\}^k} \frac{5p_K^2}{2^n - p} \leq \frac{5p^2}{2^\tau - \frac{p}{2^{n-\tau}}}.$$

6.4 Upper-Bounding $\Pr[\mathsf{coll}]$ with $n < \tau \leq 2n$

In this proof, for each forward query (K, X) (resp. inverse query (K, Y)), $\mathbf{B}$ is permitted to make an additional query as well as the original query. If the original query is a forward one, then $\mathbf{B}$ obtains two responses $Y = E_K(X)$ and $Y' = E_K(X \oplus \mathtt{bin}_n(3))$. If the query is an inverse one, then $\mathbf{B}$ obtains two responses $X = E_K^{-1}(Y)$ and $Y' = E_K(X \oplus \mathtt{bin}_n(3))$. Hence, the additional response is defined by a forward query. By the additional responses, at least one of the output blocks of the XCTR part is defined by a forward query, and $\mathbf{B}$ obtains at most $2p$ query-response tuples.

We can derive the bound of $\Pr[\mathsf{coll}]$ by incorporating the additional block Y_2 with the evaluation of the case $\tau \leq n$. The structures of the collision are given

in Fig. 12. Fig. 12(left) shows a collision by a forward query with an additional query, and Fig. 12(right) shows a collision by an inverse query (at the first block) with an additional response. Here, we explain an intuition of the evaluation. The details are given in Section 6.5.

By the additional queries, for each complete set $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})$, one of the output blocks Y_1 and Y_2 of the XCTR part is defined by a forward query. If Y_i ($i \in [2]$) is defined by the additional query, then we can use the n-bit randomness of Y_i, and $\Pr[\mathsf{coll}]$ can be bounded by the multiplying the factor $O(\frac{n/\log_2 n}{2^n})$ with the bound of $\Pr[\mathsf{coll}]$ with $\tau \leq n$. Note that our proof uses a multi-collision technique to handle the number of target values that collide with the additional response. The factor $n/\log_2 n$ comes from the multi-collision. As a result, we have the bound in Lemma 2.

6.5 Detail Evaluation for Upper-Bounding $\Pr[\mathsf{coll}]$ with $n < \tau \leq 2n$

We define two events $\mathsf{coll}_{\mathsf{fwd}}$ and $\mathsf{coll}_{\mathsf{inv}}$ such that $\mathsf{coll} = \mathsf{coll}_{\mathsf{fwd}} \vee \mathsf{coll}_{\mathsf{inv}}$:

$$\mathsf{coll}_{\mathsf{fwd}} \Leftrightarrow \mathsf{coll} \wedge (\ddot{X}_1, \ddot{Y}_1), (\ddot{X}_2, \ddot{Y}_2) \in \mathcal{Q}^{\mathsf{fwd}}_{\ddot{K}}$$
$$\mathsf{coll}_{\mathsf{inv}} \Leftrightarrow \mathsf{coll} \wedge \exists i \in [2] \text{ s.t. } (\ddot{X}_i, \ddot{Y}_i) \in \mathcal{Q}^{\mathsf{inv}}_{\ddot{K}} \wedge (\ddot{X}_{3-i}, \ddot{Y}_{3-i}) \in \mathcal{Q}^{\mathsf{fwd}}_{\ddot{K}}.$$

See Fig. 12 for these events. Without loss of generality, we assume that $(\dot{K}_1, \dot{X}_1, \dot{Y}_1) \prec (\ddot{K}_1, \ddot{X}_1, \ddot{Y}_1)$. Note that by additional queries, $(\ddot{K}_1, \ddot{X}_1, \ddot{Y}_1)$ or $(\ddot{K}_2, \ddot{X}_2, \ddot{Y}_2)$ is defined by a forward query. We evaluate the probabilities for $\mathsf{coll}_{\mathsf{fwd}}$ and $\mathsf{coll}_{\mathsf{inv}}$.

Evaluating the probability for $\mathsf{coll}_{\mathsf{fwd}}$

Fro each $(\dot{K}, \dot{X}_1, \dot{Y}_1), (\dot{K}, \dot{X}_2, \dot{Y}_2) \in \mathcal{Q}$ and $(\ddot{K}, \ddot{X}_1, \ddot{Y}_1), (\ddot{K}, \ddot{X}_2, \ddot{Y}_2) \in \mathcal{Q}^{\mathsf{fwd}}$ such that $\dot{K} \neq \ddot{K}$ and $(\dot{K}, \dot{X}_1, \dot{Y}_1) \prec (\ddot{K}, \ddot{X}_1, \ddot{Y}_1)$, since $\ddot{Y}_1$ and $\ddot{Y}_2$ are chosen uniformly at random from at least $2^n - 2p$ elements, we have

$$\begin{aligned}\Pr[\mathsf{msb}_\tau(\dot{Y}_1 \| \dot{Y}_2) = \mathsf{msb}_\tau(\ddot{Y}_1 \| \ddot{Y}_2)] &\leq \sum_{S \in \{0,1\}^{2n-\tau}} \Pr[\dot{Y}_1 \| \dot{Y}_2 = \ddot{Y}_1 \| \ddot{Y}_2 \oplus (0^\tau || S)] \\ &\leq 2^{2^n - \tau} \left(\frac{1}{2^n - 2p}\right)^2,\end{aligned}$$

providing the following bound:

$$\begin{aligned}\Pr[\mathsf{coll}_{\mathsf{fwd}}] &\leq \sum_{\{\dot{K}, \ddot{K}\} \in \mathsf{UConf}_2(\{0,1\}^k)} 2p_{\dot{K}} p_{\ddot{K}} \cdot 2^{2^n - \tau} \left(\frac{1}{2^n - 2p}\right)^2 \\ &\leq \sum_{\{\dot{K}, \ddot{K}\} \in \mathsf{UConf}_2(\{0,1\}^k)} \frac{2p_{\dot{K}} p_{\ddot{K}}}{2^\tau (1 - \frac{4p}{2^n})} \\ &\leq \frac{p^2}{2^\tau (1 - \frac{4p}{2^n})}.\end{aligned}$$

Evaluating the Probability for $\mathsf{coll}_{\mathsf{inv}}$

Additional Events. We define two events $\mathsf{hit}_{0,1}$ and mcoll for evaluating the probabilities for $\mathsf{coll}_{\mathsf{fwd}}$ and $\mathsf{coll}_{\mathsf{inv}}$.

$\underline{\mathsf{hit}_{0,1}}$. This event is defined as

$$\mathsf{hit}_{0,1} \Leftrightarrow \exists i \in \{0,1\}, (*, \mathtt{bin}_n(i), *) \in \mathcal{Q}^{\mathsf{inv}},$$

wherein some mask L is defined by an inverse query. For each $(K, X, Y) \in \mathcal{Q}^{\mathsf{inv}}$, X is chosen from at least $2^n - 2p$ elements, we have

$$\Pr[\mathsf{hit}_{0,1}] \leq \frac{2p}{2^n - 2p}.$$

$\underline{\mathsf{mcoll}}$. This is a multi-collision event for f_1 or f_2. See Section 6.2 for the definitions of f_1 and f_2. The event is used for upper-bounding the number of complete sets, and the upper-bound is used for summing the probability of $\mathsf{coll}_{\mathsf{inv}}$ for each pair of complete sets. mcoll is defined as follows.[13]

$\mathsf{mcoll} \Leftrightarrow$
$\exists Z \in \{0,1\}^n, i \in [2], (K^{(1)}, X^{(1)}, Y^{(1)}), \ldots, (K^{(\mu)}, X^{(\mu)}, Y^{(\mu)}) \in \mathcal{Q}$ s.t.
$\Big(\forall j \in [\mu] : f_i(\mathcal{S}^{\mathsf{comp}}_{(K^{(j)}, X^{(j)})}(\mathcal{Q})) = Z\Big) \wedge \Big(\forall \{i,j\} \in \mathsf{UConf}_2([\mu]) : K^{(i)} \neq K^{(j)}\Big)$.

Fix $Z \in \{0,1\}^n$, $i \in [2]$, and $(K, X, Y) \in \mathcal{Q}$. We evaluate the probability that the following event occurs where $L = E_K(\mathtt{bin}_n(1))$. Note that by $\neg\mathsf{hit}_{0,1}$, L is defined by a forward query.

$$\exists (X_i, Y_i) \in \mathcal{Q}_K \text{ s.t. } Y_i = Z \wedge X_i = X \oplus Y \oplus L \oplus \mathtt{bin}_n(i).$$

- If $(X_i, Y_i) \in \mathcal{Q}^{\mathsf{fwd}}_K$, then since Y_i is chosen uniformly at random from at least $2^n - 2p$ elements, we have

$$\Pr[Y_i = Z \wedge X_i = X \oplus Y \oplus L \oplus \mathtt{bin}_n(i)] \leq \Pr[Y_i = Z] \leq \frac{1}{2^n - 2p}.$$

- If $(X_i, Y_i) \in \mathcal{Q}^{\mathsf{inv}}_K$, then we consider the following cases.
  - If $Y_i = L \oplus \mathtt{bin}_n(i)$, then since L is chosen uniformly at random from at least $2^n - 2p$ elements, we have

$$\begin{aligned}&\Pr[Y_i = Z \wedge X_i = X \oplus Y \oplus L \oplus \mathtt{bin}_n(i)] \\ &\leq \Pr[Y_i = Z] = \Pr[L \oplus \mathtt{bin}_n(i) = Z] \leq \frac{1}{2^n - 2p}.\end{aligned}$$

[13] μ appears in both mcoll and $\mathsf{coll}_{\mathsf{inv}}$, and we choose μ so that the probabilities for these two events are balanced. Under this requirement, the optimal choice becomes $\mu = n/\log_2(n)$.

- If $Y_i \neq L \oplus \mathtt{bin}_n(i) \wedge (X, Y) = (X_i, Y_i)$, then letting $Y_i \coloneqq L \oplus W$ such that $W \neq \mathtt{bin}_n(i)$, we have

$$\begin{aligned}&\Pr[Y_i = Z \wedge X_i = X \oplus Y \oplus L \oplus \mathtt{bin}_n(i)]\\ &\leq \Pr[X_i = X \oplus Y \oplus L \oplus \mathtt{bin}_n(i)] = \Pr[W = \mathtt{bin}_n(i)] = 0.\end{aligned}$$

- If $Y_i \neq L \oplus \mathtt{bin}_n(i) \wedge (X, Y) \neq (X_i, Y_i)$, then since X_i, L, and either X or Y are chosen uniformly at random from at least $2^n - 2p$ elements, we have

$$\begin{aligned}&\Pr[Y_i = Z \wedge X_i = X \oplus Y \oplus L \oplus \mathtt{bin}_n(i)]\\ &\leq \Pr[X_i = X \oplus Y \oplus L \oplus \mathtt{bin}_n(i)] \leq \frac{1}{2^n - 2p}.\end{aligned}$$

By summing the bound for each possible tuple $(i, Z, (K^{(1)}, X^{(1)}, Y^{(1)}), \ldots, (K^{(\mu)}, X^{(\mu)}, Y^{(\mu)}))$ and using Stirling's approximation, we have

$$\begin{aligned}\Pr[\mathsf{mcoll}] &\leq \sum_{\substack{i\in[2],\\ Z\in\{0,1\}^n}} \sum_{\substack{(K^{(1)},X^{(1)},Y^{(1)}),\ldots,\\ (K^{(\mu)},X^{(\mu)},Y^{(\mu)})\in\mathcal{Q}\text{ s.t.}\\ \forall\{j_1,j_2\}\in\mathsf{UConf}_2([\mu]):K^{(j_1)}\neq K^{(j_2)}}} \Pr\left[\begin{array}{l}\forall j \in [\mu] :\\ f_i(\mathcal{S}^{\mathsf{comp}}_{(K^{(j)},X^{(j)})}(\mathcal{Q})) = Z\end{array}\right]\\ &\leq \sum_{\substack{i\in[2],\\ Z\in\{0,1\}^n}} \binom{p}{\mu} \cdot \left(\frac{1}{2^n - 2p}\right)^{\mu}\\ &\leq 2 \cdot 2^n \cdot \binom{p}{\mu} \cdot \left(\frac{1}{2^n - 2p}\right)^{\mu}\\ &\leq \left((2\log_2 n) \cdot \frac{ep}{\frac{n}{\log_2 n} \cdot (2^n - 2p)}\right)^{\frac{n}{\log_2 n}}\\ &\leq \left(\frac{6(\log_2 n)^2}{n(1 - \frac{2p}{2^n})} \cdot \frac{p}{2^n}\right)^{\frac{n}{\log_2 n}}.\end{aligned}$$

Deriving the Bound of the Probability for $\mathsf{coll}_{\mathsf{inv}}$. Let $\mathsf{coll}^*_{\mathsf{inv}}$ be an event that $\mathsf{coll}_{\mathsf{inv}}$ occurs before $\mathsf{hit}_{0,1}$ and mcoll occur. We evaluate the probability that $\mathsf{coll}^*_{\mathsf{inv}}$ occurs.

Fix

$$\begin{aligned}&i \in [2], \{\dot{K}, \ddot{K}\} \in \mathsf{UConf}_2(\{0,1\}^k),\\ &(\dot{X}_1, \dot{Y}_1), (\dot{X}_2, \dot{Y}_2) \in \mathcal{Q}_{\dot{K}},\\ &(\ddot{X}, \ddot{Y}), (\ddot{X}_1, \ddot{Y}_1), (\ddot{X}_2, \ddot{Y}_2), (\mathtt{bin}_n(1), \ddot{L}) \in \mathcal{Q}_{\ddot{K}}.\end{aligned}$$

such that

$$(\ddot{X}_i, \ddot{Y}_i) \in \mathcal{Q}_{\ddot{K}}^{\mathsf{inv}} \text{ (i.e., } (\ddot{X}_{3-i}, \ddot{Y}_{3-i}) \in \mathcal{Q}_{\ddot{K}}^{\mathsf{fwd}}),$$
$$\dot{X}_1 \oplus \dot{X}_2 = \ddot{X}_1 \oplus \ddot{X}_2 = \mathtt{bin}_n(3), \text{ and}$$
$$(\dot{K}, \dot{X}_1, \dot{Y}_1) \prec (\ddot{K}, \ddot{X}_1, \ddot{Y}_1).$$

We then evaluate the following probability.

$$\Pr\left[\mathsf{msb}_\tau(\dot{Y}_1 \| \dot{Y}_2) = \mathsf{msb}_\tau(\ddot{Y}_1 \| \ddot{Y}_2) \wedge \ddot{X}_i = \ddot{X} \oplus \ddot{Y} \oplus \ddot{L} \oplus \mathtt{bin}_n(i)\right].$$

In this evaluation, we consider the following trivial case[14] where

$$\mathsf{C}_{\mathsf{triv}} \coloneqq ((\ddot{X}_i, \ddot{Y}_i) = (\ddot{X}, \ddot{Y}) \wedge \ddot{Y}_i = \ddot{L} \oplus \mathtt{bin}_n(i)),$$

which is a 2-block version of the trivial case in the evaluation with $\tau \leq n$. In the trivial case, the DM part and the i-th block in the XCTR part are trivially connected, i.e.,

$$\begin{aligned}&(\ddot{X}_i, \ddot{Y}_i) = (\ddot{X}, \ddot{Y}) \wedge \ddot{Y}_i = \ddot{L} \oplus \mathtt{bin}_n(i)\\ &\Rightarrow \ddot{X} \oplus \ddot{Y} \oplus \ddot{L} \oplus \mathtt{bin}_n(i) = \ddot{X}_i \oplus (\ddot{L} \oplus \mathtt{bin}_n(i)) \oplus \ddot{L} \oplus \mathtt{bin}_n(i) = \ddot{X}_i.\end{aligned}$$

Evaluation with $\neg\mathsf{C}_{\mathsf{triv}}$. Fix $V \in \{0,1\}^{2n-\tau}$. We evaluate the following probability:

$$p^* \coloneqq \Pr\left[\dot{Y}_1 \| \dot{Y}_2 \oplus \ddot{Y}_1 \| \ddot{Y}_2 = (0^{\tau-n} \| V) \wedge \ddot{X}_i = \ddot{X} \oplus \ddot{Y} \oplus \ddot{L} \oplus \mathtt{bin}_n(i) \mid \neg\mathsf{C}_{\mathsf{triv}}\right].$$

We first evaluate the probability that $\ddot{X}_i = \ddot{X} \oplus \ddot{Y} \oplus \ddot{L} \oplus \mathtt{bin}_n(i)$ holds.

- If $(\ddot{X}, \ddot{Y}) \prec (\ddot{X}_i, \ddot{Y}_i)$, then since $\ddot{X}_i$ is chosen from at least $2^n - p$ elements, $\Pr[\ddot{X} \oplus \ddot{Y} \oplus \ddot{L} \oplus \mathtt{bin}_n(i) = \ddot{X}_i] \leq \frac{1}{2^n - 2p}$.
- If $(\ddot{X}, \ddot{Y}) = (\ddot{X}_i, \ddot{Y}_i)$, then by $\neg\mathsf{C}_{\mathsf{triv}}$, we have $\ddot{Y}_i = \ddot{L} \oplus W$ such that $W \neq \mathtt{bin}_n(i)$, and thus $\Pr[\ddot{X}_i = \ddot{X} \oplus \ddot{Y} \oplus \ddot{L} \oplus \mathtt{bin}_n(i)] = 0$.
- If $(\ddot{X}_i, \ddot{Y}_i) \prec (\ddot{X}, \ddot{Y}) \wedge (\ddot{X}, \ddot{Y}) \neq (\mathtt{bin}_n(1), \ddot{L})$, then since $\ddot{L}$ and $\ddot{X}$ or $\ddot{Y}$ are chosen from at least $2^n - p$ elements, $\Pr[\ddot{X} \oplus \ddot{Y} \oplus \ddot{L} \oplus \mathtt{bin}_n(i) = \ddot{X}_i] \leq \frac{1}{2^n - 2p}$.
- If $(\ddot{X}, \ddot{Y}) = (\mathtt{bin}_n(1), \ddot{L})$, then $\ddot{X} \oplus \ddot{Y} \oplus \ddot{L} \oplus \mathtt{bin}_n(1) = 0^n$, and by $\neg\mathsf{hit}_{0,1}$, $\Pr[\ddot{X}_i = \ddot{X} \oplus \ddot{Y} \oplus \ddot{L} \oplus \mathtt{bin}_n(i)] = 0$.

Regarding the collision for $\dot{Y}_{3-i}$ and $\ddot{Y}_{3-i}$, since $\ddot{Y}_{3-i}$ is defined by a forward query and is chosen uniformly at random from at least $2^n - p$ elements, for each $V \in \{0,1\}^{2n-\tau}$, we have $\Pr[\dot{Y}_1 \| \dot{Y}_2 \oplus \ddot{Y}_1 \| \ddot{Y}_2 = (0^{\tau-n} \| V)] \leq \frac{1}{2^n - 2p}$.

By using the above bounds, we have $p^* \leq \left(\frac{1}{2^n - 2p}\right)^2$.

[14] The same condition appears in Proposition 1 in the attack section (Section 4.2).

For any $W \in \{0,1\}^{2n-\tau}$, the number of complete sets $\mathcal{S}^{\mathsf{comp}}_{(\dot{K},\dot{Y})}$ such that $\dot{Y}_i = \ddot{Y}_i \oplus (0^{\tau-n}\|W)$ is at most $\mu - 1$ by $\neg\mathsf{mcoll}$. Hence, we have

$$\begin{aligned}
&\Pr[\mathsf{coll}^*_{\mathsf{inv}} \mid \neg\mathsf{C}_{\mathsf{triv}}] \\
&\le \sum_{i\in[2]} 2^{2n-\tau} \cdot (\mu-1) \cdot \sum_{\ddot{K}\in\{0,1\}^k} \sum_{(\ddot{X},\ddot{Y})\in\mathcal{Q}_{\ddot{K}}} p_{\ddot{K}} \cdot (2p_{\ddot{K}}) \cdot \left(\frac{1}{2^n-2p}\right)^2 \\
&\le \sum_{\ddot{K}\in\{0,1\}^k} \sum_{(\ddot{X},\ddot{Y})\in\mathcal{Q}_{\ddot{K}}} \frac{4\cdot 2^{2n-\tau}(\mu-1)p^2_{\ddot{K}}}{(2^n-2p)^2} \\
&\le 4\left(\frac{n}{\log_2 n} - 1\right) \cdot \frac{p^2}{2^\tau(1-\frac{4p}{2^n})}.
\end{aligned}$$

Evaluation with $\mathsf{C}_{\mathsf{triv}}$. We evaluate the probability that $\mathsf{coll}^*_{\mathsf{inv}}$ with $\mathsf{C}_{\mathsf{triv}}$. In this case, $\ddot{Y}_i = \ddot{L} \oplus \mathtt{bin}_n(i)$ and $\ddot{L}$ is defined by a forward query. Since $\ddot{L}$ and $\ddot{Y}_{3-i}$ are respectively chosen uniformly at random from $2^n - 2p$ elements, we have

$$\begin{aligned}
\Pr[\mathsf{msb}_\tau(\dot{Y}_1\|\dot{Y}_2) = \mathsf{msb}_\tau(\ddot{Y}_1\|\ddot{Y}_2)] &\le \sum_{S\in\{0,1\}^{2n-\tau}} \Pr[\dot{Y}_1\|\dot{Y}_2 \oplus \ddot{Y}_1\|\ddot{Y}_2 = (0^\tau\|S)] \\
&\le 2^{2n-\tau} \cdot \left(\frac{1}{2^n-2p}\right)^2.
\end{aligned}$$

Hence, we have

$$\begin{aligned}
\Pr[\mathsf{coll}^*_{\mathsf{inv}} \mid \mathsf{C}_{\mathsf{triv}}] &\le \sum_{i\in[2]} \sum_{\substack{\{\dot{K},\ddot{K}\}\in \\ \mathsf{UConf}_2(\{0,1\}^k)}} \sum_{\substack{(\dot{X},\dot{Y})\in\mathcal{Q}_{\dot{K}} \\ (\ddot{X}_i,\ddot{L}\oplus\mathtt{bin}_n(i))\in\mathcal{Q}_{\ddot{K}}}} 2^{2n-\tau} \cdot \left(\frac{1}{2^n-2p}\right)^2 \\
&\le \sum_{i\in[2]} \frac{0.5\cdot 2^{2n-\tau}\cdot p^2}{(2^n-2p)^2} \\
&= \frac{2^{2n-\tau}\cdot p^2}{(2^n-2p)^2}.
\end{aligned}$$

Deriving the Bound of $\Pr[\mathsf{coll}_{\mathsf{inv}}]$. By using the above bounds, we have

$$\begin{aligned}
\Pr[\mathsf{coll}_{\mathsf{inv}}] &\le 4\left(\frac{n}{\log_2 n} - 1\right) \cdot \frac{p^2}{2^\tau(1-\frac{4p}{2^n})} + \frac{2p^2}{2^\tau(1-\frac{4p}{2^n})} \\
&\le \frac{4n}{(\log_2 n)(1-\frac{4p}{2^n})} \cdot \frac{p^2}{2^\tau}.
\end{aligned}$$

Algorithm 6 DMXCTRpoly

Function $\mathsf{DMXCTRpoly}[E](l, K, X)$ where $l \le 2^{n/2}$

1: $\ell \leftarrow \lceil l/n \rceil$; $H \leftarrow E_K(\mathtt{bin}_n(0))$; $L \leftarrow E_K(\mathtt{bin}_n(1))$; $Y \leftarrow E_K(X)$; $IV \leftarrow Y \oplus L$
2: **for** $i \in [\ell]$ **do** $X_i \leftarrow IV \oplus \mathtt{bin}_n(i)$; $Y_i \leftarrow E_K(X_i)$ **end for**
3: $S \leftarrow X \oplus Y \oplus \overline{\mathsf{poly}}_H \circ \mathsf{zp}_{n*} \circ \mathsf{msb}_l(Y_1 \| \cdots \| Y_{\ell-1} \| Y_\ell)$; **return** $\mathsf{msb}_{\min\{\tau,n\}}(S)$

Conclusion of the Evaluation

By using the above bounds, we have

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{DMXCTR}}(\mathbf{B}) &\le \frac{2p}{2^n - 2p} + \left(\frac{6(\log_2 n)^2}{n(1-\frac{2p}{2^n})} \cdot \frac{p}{2^n}\right)^{\frac{n}{\log_2 n}} \\
&\quad + \frac{1}{1-\frac{4p}{2^n}} \cdot \frac{p^2}{2^\tau} + \frac{4n}{(\log_2 n)(1-\frac{4p}{2^n})} \cdot \frac{p^2}{2^\tau} \\
&\le \frac{5n}{(1-\frac{4p}{2^n})\log_2 n} \cdot \frac{p^2}{2^\tau} + \frac{2}{(1-\frac{2p}{2^n})} \cdot \frac{p}{2^n} \\
&\quad + \left(\frac{6(\log_2 n)^2}{n(1-\frac{2p}{2^n})} \cdot \frac{p}{2^n}\right)^{\frac{n}{\log_2 n}}.
\end{aligned}$$

7 Proof of Theorem 2: CMT_k Security of EtE_P-HCTR2

Let m_{max} be the maximum length in blocks of the plaintexts of EtE_P-HCTR2. We assume that $m_{\mathsf{max}} \le 2^{n/2}$.

We consider a partial function of EtE_P-HCTR2 called "DMXCTRpoly". The specification is shown in Algorithm 6 and Fig. 4(right). DMXCTRpoly consists of three parts: DM, XCTR, and $\overline{\mathsf{poly}}$. DMXCTRpoly takes a length l and a pair of a key K and an input block X, and returns a $\min\{\tau, n\}$-bit value S.

The following lemma shows that the CMT_k security of EtE_P-HCTR2 is reduced to the collision resistance of DMXCTRpoly. The advantage function of an adversary $\mathbf{B}$ that breaks the collision resistance of DMXCTRpoly is defined as

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{DMXCTRpoly}}(\mathbf{B}) \coloneqq \Pr\big[(l,(K^\dagger,X^\dagger),(K^\ddagger,X^\ddagger)) \leftarrow \mathbf{B}^{E,E^{-1}} \text{ s.t. } K^\dagger \ne K^\ddagger \wedge \\
\mathsf{DMXCTRpoly}[E](l,K^\dagger,X^\dagger) = \mathsf{DMXCTRpoly}[E](l,K^\ddagger,X^\ddagger)\big].
\end{aligned}$$

Lemma 3. *For any* CMT_k *adversary* $\mathbf{A}$ *against* EtE_P-HCTR2 *that makes* p *queries, there exists a collision-finding adversary* $\mathbf{B}$ *against* DMXCTRpoly *making at most* p *primitive queries such that*

$$\mathbf{Adv}^{\mathsf{cmt}_\mathsf{k}}_{\mathsf{EtE}_\mathsf{P}\text{-}\mathsf{HCTR2}}(\mathbf{A}) \le \mathbf{Adv}^{\mathsf{coll}}_{\mathsf{DMXCTRpoly}}(\mathbf{B}).$$

The following lemma shows the bound of the collision resistance of DMXCTRpoly.

Algorithm 7 Collision Finding Adversary $\mathbf{B}$ with a CMT_k adversary $\mathbf{A}$

For $\mathbf{A}$'s query $(\tilde{K}, \tilde{X})$ to E, $\mathbf{B}$ returns $\tilde{Y}$ to $\mathbf{A}$, which is defined as follows.

1: $\mathbf{B}$ makes the query $(\tilde{K}, \tilde{X})$ to E and obtains the response $\tilde{Y}$.

For $\mathbf{A}$'s query $(\tilde{K}, \tilde{Y})$ to E^{-1}, $\mathbf{B}$ returns $\tilde{X}$ to $\mathbf{A}$, which is defined as follows.

1: $\mathbf{B}$ makes the query $(\tilde{K}, \tilde{Y})$ to E^{-1} and $\mathbf{B}$ obtains the response $\tilde{X}$.

For $\mathbf{A}$'s output tuple $((K^\dagger, T^\dagger, M^\dagger), (K^\ddagger, T^\ddagger, M^\ddagger))$, $\mathbf{B}$ returns an output tuple $(l, (K^\dagger, X^\dagger), (K^\ddagger, X^\ddagger))$, which is defined as follows.

1: $l \leftarrow \tau + |M^\dagger| - n$

2: $\mathbf{B}$ calculates the internal values $X_0^\dagger$ of $\mathsf{HCTR2.enc}[E_{K^\dagger}](T^\dagger, 0^\tau \| M^\dagger)$ and $X_0^\ddagger$ of $\mathsf{HCTR2.enc}[E_{K^\ddagger}](T^\ddagger, 0^\tau \| M^\ddagger))$, respectively, by using the query-response tuples from $\mathbf{A}$ to $E^{\pm}$

3: $(X^\dagger, X^\ddagger) \leftarrow (X_0^\dagger, X_0^\ddagger)$

Lemma 4. *For any adversary* $\mathbf{B}$ *against* $\mathsf{DMXCTRpoly}$ *making* p *queries,*

$$\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{DMXCTRpoly}}(\mathbf{B}) \le \frac{2n^2}{(1-\frac{2p}{2^n})\log_2 n} \cdot \frac{p^2}{2^\tau - \frac{2p}{2^{n-\tau}}} + \left(\frac{6\log_2 n}{\sqrt{n}(1-\frac{2p}{2^n})} \cdot \frac{p}{2^{n/2}} \right)^{\frac{n}{\log_2 n}}.$$

Combining these lemmas, we have the bound in Theorem 2.

7.1 Proof of Lemma 3

Assume that $\mathbf{A}$ is a CMT_k adversary against EtE_P-$\mathsf{HCTR2}$ making at most p primitive queries. We construct a collision-finding adversary $\mathbf{B}$ against $\mathsf{DMXCTRpoly}$ making at most p primitive queries. The adversary $\mathbf{B}$ is defined in Algorithm 7. The adversary $\mathbf{B}$ simulates $\mathbf{A}$'s IC by using $\mathbf{B}$'s IC $E^{\pm}$. After finishing all queries of $\mathbf{A}$, $\mathbf{B}$ calculates the internal values $X_0^\dagger$ and $X_0^\ddagger$ of $\mathsf{HCTR2.enc}$ with $\mathbf{A}$'s outputs and $\mathbf{A}$'s query-response tuples. Let $t \coloneqq \min\{\tau, n\}$.

We show that a collision of EtE_P-$\mathsf{HCTR2.enc}$ yields a collision of $\mathsf{DMXCTRpoly}$. Let us consider the following two equations to show the $\mathsf{POLYVAL}$ evaluations of $\mathsf{HCTR2.enc}[E_K](T, 0^\tau \| M))$:

$$0^t = \mathsf{msb}_t(X_0 \oplus \mathsf{POLYVAL}_H(T, M_*)),\ C_0 = Y_0 \oplus \mathsf{POLYVAL}_H(T, C_*).$$

Let $l \coloneqq |M_*|$ and $\ell \coloneqq \lceil |M_*|/n \rceil$. With these equations, we have

$$\begin{aligned}\mathsf{msb}_t(C_0) &= \mathsf{msb}_t(X_0 \oplus Y_0 \oplus \overline{\mathsf{poly}}_H \circ \mathsf{zp}_{n*}(M_* \oplus C_*)) \\ &= \mathsf{msb}_t(X_0 \oplus Y_0 \oplus \overline{\mathsf{poly}}_H \circ \mathsf{zp}_{n*} \circ \mathsf{msb}_l(Y_1 \| \cdots \| Y_\ell)).\end{aligned}$$

Thus, for two input-output tuples of EtE_P-$\mathsf{HCTR2.enc}$ $(K^\dagger, T^\dagger, M^\dagger, C^\dagger)$ and $(K^\ddagger, T^\ddagger, M^\ddagger, C^\ddagger)$, we have

$$\begin{aligned}C_0^\dagger = C_0^\ddagger &\Rightarrow \mathsf{msb}_t(C_0^\dagger) = \mathsf{msb}_t(C_0^\ddagger) \\ &\Rightarrow \mathsf{DMXCTRpoly}[E](l, K^\dagger, X_0^\dagger) = \mathsf{DMXCTRpoly}[E](l, K^\ddagger, X_0^\ddagger),\end{aligned}$$

ensuring that if $\mathbf{A}$ breaks the CMT_k security of EtE_P-$\mathsf{HCTR2}$, then $\mathbf{B}$ finds a collision of $\mathsf{DMXCTRpoly}$. □

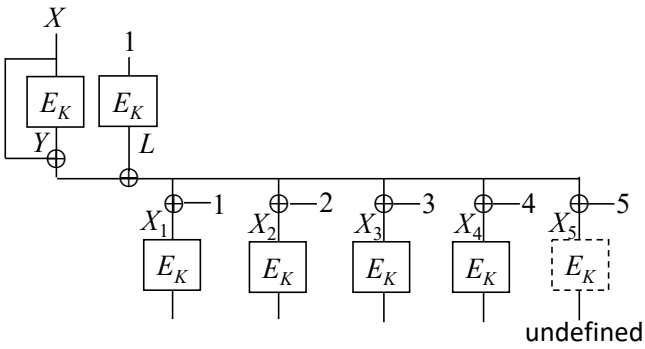

Fig. 13. Complete Set $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}) = \{(0, X, Y), (1, X_1, *), (2, X_2, *), (3, X_3, *), (4, X_4, *)\}$.

7.2 Proof of Lemma 4

Without loss of generality, we assume that $\mathbf{B}$ is deterministic and does not make repeated queries. We use the definitions in Section 5.2 In this proof, for each forward query (K, X) (resp. inverse query (K, Y)), $\mathbf{B}$ is permitted to obtain an additional response associated with K before obtaining the response of the query: if the hash key $H = E_K(\mathtt{bin}_n(0))$ is not responded to $\mathbf{B}$, then $\mathbf{B}$ obtains H that is defined by a forward query. By the additional query-response tuples, $\mathbf{B}$ obtains at most $2p$ query-response tuples.

Complete Set for DMXCTRpoly. For $(K, X, Y) \in \mathcal{Q}$, we define a complete set $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})$ for DMXCTRpoly that keeps internal tuples of DMXCTRpoly such that the DM part is (K, X).

$$\begin{aligned}\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}) \coloneqq \{&(0, X, Y), (1, X_1, Y_1), \ldots, (\ell, X_\ell, Y_\ell) \mid \\ &(\mathtt{bin}_n(1), L), (X, Y), (X_1, Y_1), \ldots, (X_\ell, Y_\ell) \in \mathcal{Q}_K \wedge \\ &(X_{\ell+1}, *) \notin \mathcal{Q}_K \wedge (\forall i \in [\ell + 1] : X_i = X \oplus Y \oplus L \oplus \mathtt{bin}_n(i))\}.\end{aligned}$$

Fig. 13 illustrates the structure of $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})$. Note that if $\ell = 0$, then $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}) = \{(0, X, Y)\}$. We define notations and subsets of $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})$ in the following.

- Let f_{poly} be a function that takes l and $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})$, and returns an output of $\mathsf{DMXCTRpoly}[E](l, K, X)$, i.e.,
  - if $l \in [0, n\ell]$, then $f_{\mathsf{poly}}(l, \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})) \coloneqq \mathsf{msb}_\tau(X \oplus Y \oplus \overline{\mathsf{poly}}_H \circ \mathsf{zp}_{n*} \circ \mathsf{msb}_l(Y_1 \| \cdots \| Y_\ell))$;
  - otherwise $f_{\mathsf{poly}}(l, \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})) = \varepsilon$, where $H = E_K(\mathtt{bin}_n(0))$.
- Let $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{fwd}}) \coloneqq \{(i, X_i, Y_i) \in \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}) \mid (X_i, Y_i) \in \mathcal{Q}^{\mathsf{fwd}}_K\}$ and $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{inv}}) \coloneqq \{(i, X_i, Y_i) \in \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}) \mid (X_i, Y_i) \in \mathcal{Q}^{\mathsf{inv}}_K\}$ be subsets of $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})$ that respectively keep tuples defined forward queries and inverse ones.

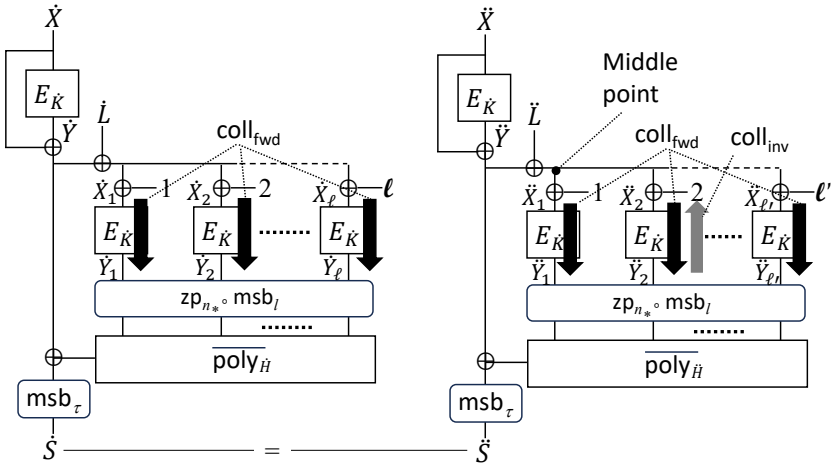

Fig. 14. $\mathsf{coll}_{\mathsf{fwd}}$ and $\mathsf{coll}_{\mathsf{inv}}$ where $\dot{K} \neq \ddot{K}$.

- Let $\ell_{(K,X)} \coloneqq |\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})| - 1$.
- For $(K, X, Y) \in \mathcal{Q}$ and $i \in [\ell_{(K,X)}]$, let $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})[i] \coloneqq \{(j, X_j, Y_j) \in \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}) \mid j \in [i]\}$ be a subset of $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})$ that keeps tuples with the indexes from 1 to i and $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})[0] \coloneqq \emptyset$. Similarly, let $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{fwd}})[i] \coloneqq \{(j, X_j, Y_j) \in \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{fwd}}) \mid j \in [i]\}$ and $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{inv}})[i] \coloneqq \{(j, X_j, Y_j) \in \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{inv}}) \mid j \in [i]\}$.

Deriving the Bound of Lemma 4. We define a collision event coll such that $\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{DMXCTRpoly}}(\mathbf{B}) \leq \Pr[\mathsf{coll}]$.

$$\begin{aligned}\mathsf{coll} \Leftrightarrow \exists(\dot{K}, \dot{X}, \dot{Y}), (\ddot{K}, \ddot{X}, \ddot{Y}) \in \mathcal{Q}, l \in [0, nm_{\mathsf{max}}] \text{ s.t.} \\ f_{\mathsf{poly}}(l, \mathcal{S}^{\mathsf{comp}}_{(\dot{K},\dot{X})}(\mathcal{Q})) = f_{\mathsf{poly}}(l, \mathcal{S}^{\mathsf{comp}}_{(\ddot{K},\ddot{X})}(\mathcal{Q})) \neq \varepsilon.\end{aligned}$$

Let $\mathcal{S}^{\mathsf{comp}}_{(\dot{K},\dot{X})}(\mathcal{Q}) = \{(0, \dot{X}, \dot{Y}), (1, \dot{X}_1, \dot{Y}_1), \ldots, (\ell_{(\dot{X},\dot{Y})}, \dot{X}_{\ell_{(\dot{X},\dot{Y})}}, \dot{Y}_{\ell_{(\dot{X},\dot{Y})}})\}$ and $\mathcal{S}^{\mathsf{comp}}_{(\ddot{K},\ddot{X})}(\mathcal{Q}) = \{(0, \ddot{X}, \ddot{Y}), (1, \ddot{X}_1, \ddot{Y}_1), \ldots, (\ell_{(\ddot{X},\ddot{Y})}, \ddot{X}_{\ell_{(\ddot{X},\ddot{Y})}}, \ddot{Y}_{\ell_{(\ddot{X},\ddot{Y})}})\}$.

We define the following sub events for coll such that $\mathsf{coll} = \mathsf{coll}_{\mathsf{fwd}} \vee \mathsf{coll}_{\mathsf{inv}}$. These events are shown in Fig. 14.

$$\begin{aligned}\mathsf{coll}_{\mathsf{fwd}} \Leftrightarrow \mathsf{coll} \wedge \forall(i, \dot{X}_i, \dot{Y}_i) \in \mathcal{S}^{\mathsf{comp}}_{(\dot{K},\dot{X})}(\mathcal{Q}) : (\dot{X}_i, \dot{Y}_i) \in \mathcal{Q}^{\mathsf{fwd}}_{\dot{K}} \\ \wedge \forall(i, \ddot{X}_i, \ddot{Y}_i) \in \mathcal{S}^{\mathsf{comp}}_{(\ddot{K},\ddot{X})}(\mathcal{Q}) : (\ddot{X}_i, \ddot{Y}_i) \in \mathcal{Q}^{\mathsf{fwd}}_{\ddot{K}},\end{aligned}$$

$$\begin{aligned}\mathsf{coll}_{\mathsf{inv}} \Leftrightarrow \mathsf{coll} \wedge \Big(\exists(i, \dot{X}_i, \dot{Y}_i) \in \mathcal{S}^{\mathsf{comp}}_{(\dot{K},\dot{X})}(\mathcal{Q}) \text{ s.t. } (\dot{X}_i, \dot{Y}_i) \in \mathcal{Q}^{\mathsf{inv}}_{\dot{K}} \\ \vee \big(\exists(i, \ddot{X}_i, \ddot{Y}_i) \in \mathcal{S}^{\mathsf{comp}}_{(\ddot{K},\ddot{X})}(\mathcal{Q}) \text{ s.t. } (\ddot{X}_i, \ddot{Y}_i) \in \mathcal{Q}^{\mathsf{inv}}_{\ddot{K}}\big)\Big),\end{aligned}$$

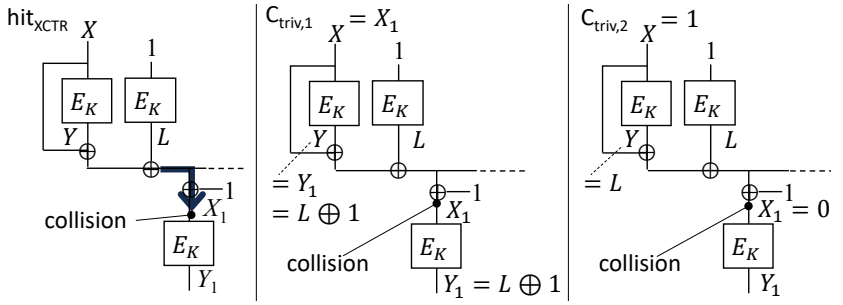

Fig. 15. $\mathsf{hit}_{\mathsf{XCTR}}$ (left), trivial case $\mathsf{C}_{\mathsf{triv},1}$ (center), and trivial case $\mathsf{C}_{\mathsf{triv},2}$ (right). In the left figure, $(X_1, Y_1) \prec (\mathtt{bin}_n(1), L) \vee (X_1, Y_1) \prec (X, Y) \vee (X_1, Y_1) \in \mathcal{Q}_K^{\mathsf{inv}}$.

$\mathsf{coll}_{\mathsf{fwd}}$ implies that all elements are defined by forward queries, and $\mathsf{coll}_{\mathsf{inv}}$ implies that some element is defined by an inverse query.

Additional Events for $\mathsf{coll}_{\mathsf{fwd}}$ and $\mathsf{coll}_{\mathsf{inv}}$. We define three events $\mathsf{hit}_{0,1}$, $\mathsf{hit}_{\mathsf{XCTR}}$, and $\mathsf{mcoll}_{\mathsf{DM}}$ for evaluating the probabilities of $\mathsf{coll}_{\mathsf{fwd}}$ and $\mathsf{coll}_{\mathsf{inv}}$.

$\underline{\mathsf{hit}_{0,1}}$. $\mathsf{hit}_{0,1}$ is an event that for some inverse query, the response hits $\mathtt{bin}_n(0)$ or $\mathtt{bin}_n(1)$:

$$\mathsf{hit}_{0,1} \Leftrightarrow \exists i \in \{0,1\}, (*, \mathtt{bin}_n(i), *) \in \mathcal{Q}^{\mathsf{inv}}.$$

For each $(K, X, Y) \in \mathcal{Q}^{\mathsf{inv}}$, since X is chosen from at least $2^n - 2p$ elements, we have

$$\Pr[\mathsf{hit}_{0,1}] \le \sum_{\alpha \in [p]} \frac{2}{2^n - 2p} = \frac{2p}{2^n - 2p}.$$

Hereafter, we assume that $\mathsf{hit}_{0,1}$ does not occur.

$\underline{\mathsf{hit}_{\mathsf{XCTR}}}$. $\mathsf{hit}_{\mathsf{XCTR}}$ is an event that there exists a complete set such that the DM part collides with the first block of the XCTR part. See Fig. 15(left). Note that $\mathsf{hit}_{\mathsf{XCTR}}$ is defined so that the trivial cases $\mathsf{C}_{\mathsf{triv},1}$[15] and $\mathsf{C}_{\mathsf{triv},2}$ are excluded.

$$\begin{aligned}\mathsf{hit}_{\mathsf{XCTR}} \Leftrightarrow \exists K \in \{0,1\}^k, (\mathtt{bin}_n(1), L), (X, Y), (X_1, Y_1) \in \mathcal{Q}_K \text{ s.t.} \\ \Big((X_1, Y_1) \preceq (\mathtt{bin}_n(1), L) \vee (X_1, Y_1) \preceq (X, Y) \vee (X_1, Y_1) \in \mathcal{Q}_K^{\mathsf{inv}}\Big) \\ \wedge \big(X_1 = X \oplus Y \oplus L \oplus \mathtt{bin}_n(1)\big) \wedge \neg(\mathsf{C}_{\mathsf{triv},1} \vee \mathsf{C}_{\mathsf{triv},2})\end{aligned}$$

where

$$\begin{aligned}&\mathsf{C}_{\mathsf{triv},1} \Leftrightarrow (X_1, Y_1) \in \mathcal{Q}_K^{\mathsf{inv}} \wedge (X_1, Y_1) = (X, Y) \wedge Y_1 = L \oplus \mathtt{bin}_n(1), \\ &\mathsf{C}_{\mathsf{triv},2} \Leftrightarrow X = \mathtt{bin}_n(1) (Fig.\ 15(right)).\end{aligned}$$

[15] The same condition appears in Proposition 1 in the attack section (Section 4.2).

See Figs. 15(center)(right) for the trivial cases.

We evaluate the probability $\Pr[\mathsf{hit}_{\mathsf{XCTR}}]$. Fix $K \in \{0,1\}^k$, (X,Y), (X_1,Y_1), $(\mathtt{bin}_n(1),L) \in \mathcal{Q}_K$ such that $\neg(\mathsf{C}_{\mathsf{triv},1} \vee \mathsf{C}_{\mathsf{triv},2})$ holds. We then consider the condition $X_1 = X \oplus Y \oplus L \oplus \mathtt{bin}_n(1)$.

- If $(X_1,Y_1),(X,Y) \prec (\mathtt{bin}_n(1),L)$, since L is not canceled out by $\neg\mathsf{C}_{\mathsf{triv},2}$ and is defined by a forward query due to $\neg\mathsf{hit}_{0,1}$, we have $\Pr[X_1 = X \oplus Y \oplus L \oplus \mathtt{bin}_n(1)] \le \frac{1}{2^n-2p}$.
- If $(X_1,Y_1),(\mathtt{bin}_n(1),L) \prec (X,Y)$, since X or Y is chosen from at least 2^n-2p elements, we have $\Pr[X_1 = X \oplus Y \oplus L \oplus \mathtt{bin}_n(1)] \le \frac{1}{2^n-2p}$.
- If $(\mathtt{bin}_n(1),L) \prec (X_1,Y_1) = (X,Y) \in \mathcal{Q}_K^{\mathsf{fwd}}$, since Y is chosen from at least 2^n-2p elements, we have $\Pr[X_1 = X \oplus Y \oplus L \oplus \mathtt{bin}_n(1)] \le \frac{1}{2^n-2p}$.
- If $(\mathtt{bin}_n(1),L) \prec (X_1,Y_1) = (X,Y) \in \mathcal{Q}_K^{\mathsf{inv}}$, letting $Y_1 = L \oplus W$ such that $W \neq \mathtt{bin}_n(1)$ (by $\neg\mathsf{C}_{\mathsf{triv},1}$), we have $\Pr[X_1 = X \oplus Y \oplus L \oplus \mathtt{bin}_n(1)] = 0$.
- If $(X,Y) \prec (X_1,Y_1) = (\mathtt{bin}_n(1),L)$, since L is chosen uniformly at random from at least 2^n-2p elements, $\Pr[X_1 = X \oplus Y \oplus L \oplus \mathtt{bin}_n(1)] \le \frac{1}{2^n-2p}$.
- If $(X,Y),(\mathtt{bin}_n(1),L) \prec (X_1,Y_1) \in \mathcal{Q}_K^{\mathsf{inv}}$, since Y_1 is chosen from at least 2^n-2p elements, we have $\Pr[X_1 = X \oplus Y \oplus L \oplus \mathtt{bin}_n(1)] \le \frac{1}{2^n-2p}$.

For each $K \in \{0,1\}^k$, the number of possible triples $(X,Y),(X_1,Y_1)$, and $(\mathtt{bin}_n(1),L)$ is at most p_K^2. Thus,

$$\Pr[\mathsf{hit}_{\mathsf{XCTR}}] \le \sum_{K\in\{0,1\}^k} p_K^2 \cdot \frac{1}{2^n-2p} \le \frac{p^2}{2^n-2p}.$$

$\underline{\mathsf{mcoll}_{\mathsf{DM}}}$. $\mathsf{mcoll}_{\mathsf{DM}}$ is a multi-collision event of the least significant half of DM's outputs for upper-bounding the number of complete sets. The upper-bound is used for summing the collision probability for each pair of complete sets. $\mathsf{mcoll}_{\mathsf{DM}}$ is defined as follows.

$$\mathsf{mcoll}_{\mathsf{DM}} \Leftrightarrow \exists K \in \{0,1\}^k, Z \in \{0,1\}^{n/2}, \{(X_1,Y_1),\ldots,(X_\mu,Y_\mu)\} \in \mathsf{UConf}_\mu(\mathcal{Q}_K) \\ \text{s.t. } \forall i \in [\mu] : \mathsf{lsb}_{n/2}(X_i \oplus Y_i) = Z.$$

Assuming that $\mathsf{mcoll}_{\mathsf{DM}}$ does not occur, for each $K \in \{0,1\}^k$ and $(\dot{X},\dot{Y}) \in \mathcal{Q}_K$, the number of other tuples $(\ddot{X},\ddot{Y}) \in \mathcal{Q}_K$ such that $\mathcal{S}^{\mathsf{comp}}_{(K,\dot{X})}(\mathcal{Q}) \cap \mathcal{S}^{\mathsf{comp}}_{(K,\ddot{X})}(\mathcal{Q}) \neq \emptyset$ is at most $\mu - 1$.[16]

For each K, Z, and $(X_1,Y_1),\ldots,(X_\mu,Y_\mu)$, we have

$$\Pr[\forall i \in [\mu] : \mathsf{lsb}_{n/2}(X_i \oplus Y_i) = Z] \le \left(\frac{2^{n/2}}{2^n-2p}\right)^\mu.$$

[16] μ appears in both $\mathsf{mcoll}_{\mathsf{DM}}$ and $\mathsf{coll}_{\mathsf{fwd}}$, and we choose μ so that these two terms are balanced. Under this requirement, the optimal choice becomes $\mu = n/\log_2(n)$.

Summing the bounds for each K, Z, and μ pairs, and using Stirling's approximation, we have

$$\Pr[\mathsf{mcoll}_{\mathsf{DM}}] \le \binom{2p}{\mu} \cdot 2^{n/2} \cdot \left(\frac{2^{n/2}}{2^n - 2p}\right)^{\mu} \le 2^{n/2} \cdot \left(\frac{2ep2^{n/2}}{\frac{n}{\log_2 n} \cdot (2^n - 2p)}\right)^{\frac{n}{\log_2 n}}$$
$$\le \left(\frac{6\sqrt{n}p2^{n/2}}{\frac{n}{\log_2 n} \cdot (2^n - 2p)}\right)^{\frac{n}{\log_2 n}} \le \left(\frac{6\log_2 n}{\sqrt{n}(1 - \frac{2p}{2^n})} \cdot \frac{p}{2^{n/2}}\right)^{\frac{n}{\log_2 n}}.$$

Evaluation for $\mathsf{coll}_{\mathsf{fwd}}$. Let $\mathsf{coll}^*_{\mathsf{fwd}}$ be an event that $\mathsf{coll}_{\mathsf{fwd}}$ occurs before one of hit_1, $\mathsf{hit}_{\mathsf{XCTR}}$, and $\mathsf{mcoll}_{\mathsf{DM}}$ occurs. Fix $S \in \{0,1\}^\tau$, $(K, X, Y) \in \mathcal{Q}$, and $l \in [0, n \cdot \ell_{(K,X)}]$. Let $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}) = \{(0, X, Y), (1, X_1, Y_1), \ldots, (\ell_{(K,X)}, X_{\ell_{(K,X)}}, Y_{\ell_{(K,X)}})\}$ such that $\forall i \in [\ell_{(K,X)}] : (X_i, Y_i) \in \mathcal{Q}^{\mathsf{fwd}}_K$. We then evaluate the probability that $S = f_{\mathsf{poly}}(l, \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}))$.

- If $l = 0$, then since X or Y is randomly chosen from at least $2^n - 2p$ elements, we have $\Pr[f_{\mathsf{poly}}(0, \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})) = S] = \Pr[\mathsf{msb}_\tau(X \oplus Y) = S] \le \frac{2^{n-\tau}}{2^n - 2p}$.
- If $1 \le l < n$, then $f_{\mathsf{poly}}(l, \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})) = \mathsf{msb}_\tau(X \oplus Y \oplus \overline{H} \cdot (\mathsf{zp}_{n*} \circ \mathsf{msb}_l(Y_1)))$, and by $\mathsf{zp}_{n*} \circ \mathsf{msb}_l(Y_1)) \neq \mathtt{x}^n$, we have
$$\overline{H} \cdot (\mathsf{zp}_{n*} \circ \mathsf{msb}_l(Y_1)) = H \cdot \mathtt{x}^{-n} \cdot (\mathsf{zp}_{n*} \circ \mathsf{msb}_l(Y_1)) \neq H.$$
By $\neg\mathsf{hit}_{\mathsf{XCTR}}$, $(K, X, Y) \prec (K, X_1, Y_1)$. Hence, the randomness of X or Y is not canceled out by $H \cdot \mathtt{x}^{-n}$ (even if $X = \mathtt{bin}_n(1)$). By the randomness of Y_1 and either X or Y, we have $\Pr[f_{\mathsf{poly}}(l, \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})) = S] \le \frac{2^{n-\tau}}{2^n - 2p}$.
- If $n \le l$, then $Y_1, \ldots, Y_{\ell_{(K,X)}}$ are respectively chosen from at least $2^n - 2p$ elements, ensuring that $\Pr[f_{\mathsf{poly}}(l, \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})) = S] \le \frac{2^{n-\tau}}{2^n - 2p}$.

By $\neg\mathsf{mcoll}_{\mathsf{DM}}$, for each $K \in \{0,1\}^k$ and $(X_*, Y_*) \in \mathcal{Q}_K$, the number of complete sets that include (K, X_*, Y_*) in the XCTR part is at most $\mu - 1$. Hence, we have

$$\forall K \in \{0,1\}^k : \sum_{(X,Y) \in \mathcal{Q}_K} |\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}))| \le (\mu - 1)p_K.$$

Summing the above bounds for each pair of outputs, we have

$$
\begin{aligned}
&\Pr[\mathsf{coll}^*_{\mathsf{fwd}}] \\
&\le \sum_{\substack{\{\dot{K},\ddot{K}\}\in \\ \mathsf{UConf}_2(\{0,1\}^k)}} \sum_{\substack{(\dot{X},\dot{Y})\in\mathcal{Q}_{\dot{K}},\\ (\ddot{X},\ddot{Y})\in\mathcal{Q}_{\ddot{K}}}} \sum_{\substack{0\le l\le \\ n\cdot\ell_{(K,X)}}} \sum_{S\in\{0,1\}^\tau} \Pr\begin{bmatrix} S = f_{\mathsf{poly}}(l,\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})) \\ = f_{\mathsf{poly}}(l,\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}))\end{bmatrix} \\
&\le \sum_{\substack{\{\dot{K},\ddot{K}\}\in \\ \mathsf{UConf}_2(\{0,1\}^k)}} \sum_{\substack{(\dot{X},\dot{Y})\in\mathcal{Q}_{\dot{K}},\\ (\ddot{X},\ddot{Y})\in\mathcal{Q}_{\ddot{K}}}} \sum_{\substack{S\in \\ \{0,1\}^\tau}} n\cdot\min\left\{|\mathcal{S}^{\mathsf{comp}}_{(\dot{K},\dot{X})}(\mathcal{Q})|, |\mathcal{S}^{\mathsf{comp}}_{(\ddot{K},\ddot{X})}(\mathcal{Q})|\right\}\left(\frac{2^{n-\tau}}{2^n-2p}\right)^2 \\
&\le \sum_{K\in\{0,1\}^k} p\cdot(n(\mu-1)p_K)\cdot 2^\tau\left(\frac{2^{n-\tau}}{2^n-2p}\right)^2 \\
&\le \frac{n\left(\frac{n}{\log_2 n}-1\right)}{1-\frac{2p}{2^n}}\cdot\frac{p^2}{2^\tau-\frac{2p}{2^{n-\tau}}}.
\end{aligned}
$$

Evaluation for $\mathsf{coll}_{\mathsf{inv}}$. We define a sub-event of $\mathsf{coll}_{\mathsf{inv}}$, wherein there exist complete set $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})$ such that for some $(i_0, X_{i_0}, Y_{i_0}) \in \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{inv}})$, all elements in $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{fwd}})[i_0]$, (X_{i_1}, Y_{i_1}) are defined before (X_{i_0}, Y_{i_0}) is defined, or all elements in $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})[i_0]$ are defined by inverse queries:

$$
\begin{aligned}
&\mathsf{hit}_{\mathsf{inv}} \Leftrightarrow \exists (K,X,Y)\in\mathcal{Q}, (i_0,X_{i_0},Y_{i_0})\in\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{inv}}) \text{ s.t.} \\
&\quad\left(\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{fwd}})[i_0]\neq\emptyset \wedge \forall (i,X_i,Y_i)\in\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{fwd}})[i_0] : (X_i,Y_i)\prec(X_{i_0},Y_{i_0})\right) \\
&\qquad\qquad \vee (i_0\ge 2 \wedge \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{inv}})[i_0] = \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})[i_0]).
\end{aligned}
$$

Note that in the event, $X_i = X \oplus Y \oplus L \oplus \mathtt{bin}_n(i)$ from the definition of the complete set.

Let $\mathsf{coll}^*_{\mathsf{inv}}$ be an event that $\mathsf{coll}_{\mathsf{inv}}$ occurs before one of $\mathsf{hit}_{\mathsf{inv}}$, $\mathsf{hit}_{0,1}$, $\mathsf{hit}_{\mathsf{XCTR}}$, and $\mathsf{mcoll}_{\mathsf{DM}}$ occurs. Let $\mathsf{hit}^*_{\mathsf{inv}}$ be an event that $\mathsf{hit}_{\mathsf{inv}}$ occurs before one of $\mathsf{hit}_{0,1}$, $\mathsf{hit}_{\mathsf{XCTR}}$, and $\mathsf{mcoll}_{\mathsf{DM}}$ occurs. We then have

$$
\Pr[\mathsf{coll}_{\mathsf{inv}}] \le \Pr[\mathsf{coll}^*_{\mathsf{inv}}] + \Pr[\mathsf{hit}^*_{\mathsf{inv}}].
$$

These probabilities are evaluated in the following.

Evaluating $\Pr[\mathsf{coll}^*_{\mathsf{inv}}]$. Fix $S \in \{0,1\}^\tau$ and $(K,X,Y) \in \mathcal{Q}$. By $\neg\mathsf{hit}_{\mathsf{inv}}$ and $\neg\mathsf{hit}_{\mathsf{XCTR}}$, for $(i_0, X_{i_0}, Y_{i_0}) \in \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{inv}})$, if $i_0 \ge 2$, $\exists (i_1, X_{i_1}, Y_{i_1}) \in \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{fwd}})[i_0]$ s.t. $(X_{i_0}, Y_{i_0}) \preceq (X_{i_1}, Y_{i_1})$; and if $i_0 = 1$, $Y_1 = L \oplus \mathtt{bin}_n(1)$ holds.

Regarding the probability that $f_{\mathsf{poly}}(l, \mathcal{S}^{\mathsf{comp}}_{(\dot{K},\dot{X})}(\mathcal{Q})) = S$, the evaluation is the same as that of $\mathsf{coll}_{\mathsf{fwd}}$. By using the randomness of Y_{i_1} (if $i_0 > 1$) or L (if $i_0 = 1$) that is defined by a forward query and is chosen uniformly at random from at least $2^n - 2p$ elements, for each l, $\Pr[f_{\mathsf{poly}}(l, \mathcal{S}^{\mathsf{comp}}_{(\dot{K},\dot{X})}(\mathcal{Q})) = S] \le \frac{2^{n-\tau}}{2^n-2p}$.

Summing the bound for all possible complete sets (as the evaluation of $\Pr[\mathsf{coll}^*_{\mathsf{fwd}}]$), we have

$$\Pr[\mathsf{coll}^*_{\mathsf{inv}}] \le \frac{n\left(\frac{n}{\log_2 n}-1\right)}{1-\frac{2p}{2^n}} \cdot \frac{p^2}{2^\tau - \frac{2p}{2^{n-\tau}}}.$$

Evaluating $\Pr[\mathsf{hit}^*_{\mathsf{inv}}]$. We first consider the case that $i_0 \ge 2 \wedge \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{inv}})[i_0] = \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})[i_0]$. Since $\forall (i, X_i, Y_i) \in \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})[i_0] : (X_i, Y_i) \in \mathcal{Q}^{\mathsf{inv}}_K \Rightarrow X_1 \oplus X_2 = \mathtt{bin}_n(3)$ and X_1 and X_2 are chosen uniformly at random from at least $2^n - 2p$ elements, $\Pr[X_1 \oplus X_2 = \mathtt{bin}_n(3)] \le \frac{1}{2^n - 2p}$.

We consider the other case: $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{fwd}})[i_0] \neq \emptyset \wedge \forall (i, X_i, Y_i) \in \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{fwd}})[i_0] : (X_i, Y_i) \prec (X_{i_0}, Y_{i_0})$. Fix $i_0 \ge 1$, $(K, X, Y) \in \mathcal{Q}, (X_1, Y_1), \ldots, (X_{i_0-1}, Y_{i_0-1}) \in \mathcal{Q}_K, (X_{i_0}, Y_{i_0}) \in \mathcal{Q}^{\mathsf{inv}}_K$ s.t. $\forall i \in [i_0 - 1] : X \oplus Y \oplus L = \mathtt{bin}_n(i) \oplus X_i$, where $L = E_K(\mathtt{bin}_n(1))$. Hence, $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q})[i_0 - 1] = \{(1, X_1, Y_1), \ldots, (i_0 - 1, X_{i_0-1}, Y_{i_0-1})\}$. Note that by $\mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{fwd}})[i_0] \neq \emptyset$, $i_0 \ge 2$ holds. We then evaluate the probability that $X \oplus Y \oplus L = \mathtt{bin}_n(i_0) \oplus X_{i_0}$ holds.

- If $(X, Y) \prec (X_{i_0}, Y_{i_0})$ or $(X_{i_0}, Y_{i_0}) \prec (X, Y)$, then by the randomness of X_{i_0}, L, or $X \oplus Y$ that is chosen from at least $2^n - p$ elements, we have $\Pr[X \oplus Y \oplus L = \mathtt{bin}_n(i_0) \oplus X_{i_0}] \le \frac{1}{2^n - 2p}$.
- If $(X, Y) = (X_{i_0}, Y_{i_0})$, then $(X, Y) \neq (X_1, Y_1)$, thus $Y_1 \neq L \oplus \mathtt{bin}_n(1)$. Hence, we have $\mathsf{hit}_{\mathsf{inv}} \Rightarrow (X_1, Y_1) \prec (X, Y)$. However, it contradicts the condition $\neg \mathsf{hit}_{\mathsf{XCTR}}$.

Without loss of generality, assume that for any $(i', X_{i'}, Y_{i'}) \in \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{inv}})[i_0 - 1]$, $\mathsf{hit}_{\mathsf{inv}}$ does not occur, i.e., $\exists (i, X_i, Y_i) \in \mathcal{S}^{\mathsf{comp}}_{(K,X)}(\mathcal{Q}^{\mathsf{fwd}})[i']$ s.t. $(X_{i'}, Y_{i'}) \prec (X_i, Y_i)$. Then, for each $(K, X, Y) \in \mathcal{Q}$, the corresponding index i_0 is uniquely fixed.

Summing the bound for each possible choices, we have

$$\Pr[\mathsf{hit}^*_{\mathsf{inv}}] \le \binom{p}{2} \cdot \frac{1}{2^n - 2p} + \sum_{K \in \{0,1\}^k} \binom{p_K}{2} \cdot \frac{1}{2^n - 2p} \le \frac{3p(p-1)}{2^n - 2p}.$$

Conclusion of the Evaluation. Summing the above bounds, we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{DMXCTR}}(\mathbf{B}) &\le \Pr[\mathsf{coll}^*_{\mathsf{fwd}}] + \Pr[\mathsf{coll}^*_{\mathsf{inv}}] + \Pr[\mathsf{hit}^*_{\mathsf{inv}}] \\ &\quad + \Pr[\mathsf{hit}_{0,1}] + \Pr[\mathsf{hit}_{\mathsf{XCTR}}] + \Pr[\mathsf{mcoll}_{\mathsf{DM}}] \\ &\le \frac{2n^2}{(1-\frac{2p}{2^n})\log_2 n} \cdot \frac{p^2}{2^\tau - \frac{2p}{2^{n-\tau}}} + \left(\frac{6\log_2 n}{\sqrt{n}(1-\frac{2p}{2^n})} \cdot \frac{p}{2^{n/2}}\right)^{\frac{n}{\log_2 n}}.\end{aligned}$$

8 Conclusion

This paper studies the open research problems necessary to determine the appropriate amount of padding τ to achieve a target level of committing security with EtE-HCTR2. As a first contribution, our new EtE$_\mathsf{P}$-HCTR2 attack improves the complexity from $O(2^{\min\{n/2,\tau\}})$ to $O(2^{\tau/2})$, reducing the attack cost

in $\tau \leq n$, particularly relevant for Acc256, an instantiation using a block cipher with $n = 256$. The new attack is tight with our new security proof. Second, we show that $\mathsf{EtE_A}$-HCTR2 only provides below the birthday bound security, with a new attack that succeeds with $O(2^{\max\{\tau/3, \tau-n\}})$ complexity, outperforming the previous birthday-bound attack [10]. In order to avoid the special handling of the first message block that causes the efficient attack against $\mathsf{EtE_A}$-HCTR2, we study the security of another padding scheme that appends zeros to the first message block, namely $\mathsf{EtE_S}$-HCTR2. It achieves the tight committing security bound of $O(2^{\tau/2})$, the same level of security as $\mathsf{EtE_P}$-HCTR2. We still have open research problems regarding the committing security of EtE-HCTR2. In particular, the security of $\mathsf{EtE_P}$-HCTR2 and $\mathsf{EtE_S}$-HCTR2 is tight up to a constant depending on n, and there is room for further optimization.

References

1. 3GPP TS 33 501: Security architecture and procedures for 5G System. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3169 (2024)
2. Albertini, A., Duong, T., Gueron, S., Kölbl, S., Luykx, A., Schmieg, S.: How to abuse and fix authenticated encryption without key commitment. In: Butler, K.R.B., Thomas, K. (eds.) USENIX Security 2022. pp. 3291–3308. USENIX Association (Aug 2022), https://www.usenix.org/conference/usenixsecurity22/presentation/albertini
3. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Mouha, N., Yasuda, K.: How to securely release unverified plaintext in authenticated encryption. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part I. LNCS, vol. 8873, pp. 105–125. Springer, Berlin, Heidelberg (Dec 2014). https://doi.org/10.1007/978-3-662-45611-8_6
4. Android Open Source Project: File-based encryption. https://source.android.com/docs/security/features/encryption/file-based (2023)
5. AWS Key Management Service: AWS KMS concepts. https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html (Accessed on 12/02/2026)
6. Bellare, M., Hoang, V.T.: Efficient schemes for committing authenticated encryption. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part II. LNCS, vol. 13276, pp. 845–875. Springer, Cham (May / Jun 2022). https://doi.org/10.1007/978-3-031-07085-3_29
7. Bellare, M., Rogaway, P.: Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 317–330. Springer, Berlin, Heidelberg (Dec 2000). https://doi.org/10.1007/3-540-44448-3_24
8. Campagna, M., Maximov, A., Mattsson, J.P.: Galois Counter Mode with Secure Short Tags (GCM-SST), draft-mattsson-cfrg-aes-gcm-sst-18. https://datatracker.ietf.org/doc/draft-mattsson-cfrg-aes-gcm-sst/18/ (2023)
9. 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 (2025). https://doi.org/10.6028/NIST.IR.8552
10. Chen, Y.L., Flórez-Gutiérrez, A., Inoue, A., Ito, R., Iwata, T., Minematsu, K., Mouha, N., Naito, Y., Sibleyras, F., Todo, Y.: Key committing security of AEZ and more. IACR Trans. Symm. Cryptol. **2023**(4), 452–488 (2023). https://doi.org/10.46586/tosc.v2023.i4.452-488

11. Crowley, P., Biggers, E.: Adiantum: length-preserving encryption for entry-level processors. IACR Trans. Symm. Cryptol. **2018**(4), 39–61 (2018). https://doi.org/10.13154/tosc.v2018.i4.39-61
12. Crowley, P., Huckleberry, N., Biggers, E.: Length-preserving encryption with HCTR2. Cryptology ePrint Archive, Report 2021/1441 (2021), https://eprint.iacr.org/2021/1441
13. Dodis, Y., Grubbs, P., Ristenpart, T., Woodage, J.: Fast message franking: From invisible salamanders to encryptment. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part I. LNCS, vol. 10991, pp. 155–186. Springer, Cham (Aug 2018). https://doi.org/10.1007/978-3-319-96884-1_6
14. Dworkin, M.: NIST Special Publication 800-38C: Recommendation for block cipher modes of operation: the CCM mode for authentication and confidentiality. https://csrc.nist.gov/pubs/sp/800/38/c/upd1/final (2007)
15. Dworkin, M.: NIST Special Publication 800-38D: Recommendation for block cipher modes of operation: Galois/counter mode (GCM) and GMAC. https://csrc.nist.gov/pubs/sp/800/38/d/final (2007)
16. Farshim, P., Orlandi, C., Roşie, R.: Security of symmetric primitives under incorrect usage of keys. IACR Trans. Symm. Cryptol. **2017**(1), 449–473 (2017). https://doi.org/10.13154/tosc.v2017.i1.449-473
17. Google: HCTR2 specification and test vectors. https://github.com/google/hctr2, accessed: 2025-09-24
18. Grubbs, P., Lu, J., Ristenpart, T.: Message franking via committing authenticated encryption. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part III. LNCS, vol. 10403, pp. 66–97. Springer, Cham (Aug 2017). https://doi.org/10.1007/978-3-319-63697-9_3
19. Gueron, S.: Intel advanced encryption standard (AES) new instructions set (2010)
20. Gueron, S., Kounavis, M.E.: Intel carry-less multiplication instruction and its usage for computing the GCM mode (2014)
21. Gueron, S., Langley, A., Lindell, Y.: AES-GCM-SIV: Specification and analysis. Cryptology ePrint Archive, Paper 2017/168 (2017), https://eprint.iacr.org/2017/168
22. Gueron, S., Langley, A., Lindell, Y.: AES-GCM-SIV: nonce misuse-resistant authenticated encryption. RFC **8452**, 1–42 (2019). https://doi.org/10.17487/RFC8452, https://doi.org/10.17487/RFC8452
23. Hirose, S., Minematsu, K.: Envelope encryption in the symmetric-key setting: A formalization and generic constructions. Cryptology ePrint Archive, Paper 2023/1727 (2023), https://eprint.iacr.org/2023/1727
24. Hirose, S., Minematsu, K.: A formal treatment of envelope encryption. In: Kim, J., Park, J., Lee, W. (eds.) ICISC 2024. LNCS, vol. 15596, pp. 96–110. Springer (2024), https://doi.org/10.1007/978-981-96-5566-3_6
25. Hoang, V.T., Krovetz, T., Rogaway, P.: Robust authenticated-encryption AEZ and the problem that it solves. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part I. LNCS, vol. 9056, pp. 15–44. Springer, Berlin, Heidelberg (Apr 2015). https://doi.org/10.1007/978-3-662-46800-5_2
26. Hoang, V.T., Menda, S.: Robust AE with committing security. In: ASIACRYPT 2024, Part IX. Lecture Notes in Computer Science, vol. 15492, pp. 312–342. Springer (2024)
27. Isobe, T., Ito, R., Minematsu, K.: Security analysis of SFrame. In: Bertino, E., Shulman, H., Waidner, M. (eds.) ESORICS 2021, Part II. LNCS, vol. 12973, pp. 127–146. Springer, Cham (Oct 2021). https://doi.org/10.1007/978-3-030-88428-4_7

28. Knuth, D.E.: The art of computer programming, volume 2 (3rd ed.): Seminumerical Algorithms. Addison-Wesley Longman Publishing Co., Inc., USA (1997)
29. Lefevre, C., Mennink, B.: Tight preimage resistance of the Sponge construction. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part IV. LNCS, vol. 13510, pp. 185–204. Springer, Cham (Aug 2022). https://doi.org/10.1007/978-3-031-15985-5_7
30. Len, J., Grubbs, P., Ristenpart, T.: Partitioning oracle attacks. In: Bailey, M., Greenstadt, R. (eds.) USENIX Security 2021. pp. 195–212. USENIX Association (Aug 2021), https://www.usenix.org/conference/usenixsecurity21/presentation/len
31. Menda, S., Len, J., Grubbs, P., Ristenpart, T.: Context discovery and commitment attacks - how to break CCM, EAX, SIV, and more. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part IV. LNCS, vol. 14007, pp. 379–407. Springer, Cham (Apr 2023). https://doi.org/10.1007/978-3-031-30634-1_13
32. Naito, Y., Sasaki, Y., Takeshi, T.: Committing wide encryption mode with minimum ciphertext expansion. IACR Trans. Symm. Cryptol. **2025**(1), 44–69 (2025). https://doi.org/10.46586/tosc.v2025.i1.44-69
33. National Institute of Standards and Technology: NIST launches development of cryptographic accordions. https://csrc.nist.gov/News/2025/nist-launches-development-cryptographic-accordions (2025)
34. Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 373–390. Springer, Berlin, Heidelberg (May / Jun 2006). https://doi.org/10.1007/11761679_23
35. The Linux Kernel: hctr2.c. https://github.com/torvalds/linux/blob/master/crypto/hctr2.c, accessed: 2025-05-15
36. Wang, P., Feng, D., Wu, W.: HCTR: A variable-input-length enciphering mode. In: Information Security and Cryptology, First SKLOIS Conference, CISC 2005. LNCS, vol. 3822, pp. 175–188. Springer (2005)

A Rigorous evaluation of attacks against $\mathsf{EtE_A}$-HCTR2 using multi-collisions

Phase 1. We aim to find an r-collision $\{K^{(1)}, \ldots, K^{(r)}\}$ of keys such that $E_{K^{(1)}}(\mathtt{bin}_n(1)) = E_{K^{(2)}}(\mathtt{bin}_n(1)) = \cdots = E_{K^{(r)}}(\mathtt{bin}_n(1))$. Later, we will explain how to choose r. To find such an r-collision, we choose $2^{\frac{r-1}{r}n+\frac{\log_2(r!)+1}{r}}$ keys and compute their encryptions of $\mathtt{bin}_n(1)$, assuming that $\frac{r-1}{r}n + \frac{\log_2(r!)+1}{r} \leq k$. The total number of r-sized subsets from this collection is $\binom{2^{\frac{r-1}{r}n+\frac{\log_2(r!)+1}{r}}}{r}$. The probability that a fixed set of r keys forms an r-collision (i.e., all r encryptions are equal) is $\frac{1}{2^{(r-1)n}}$. Note that when $x^2 \leq N$, $\prod_{i=0}^{x-1}(N-i) \geq \frac{N^x}{2}$ [29]. Therefore, when $r^2 \leq 2^{\frac{r-1}{r}n+\frac{\log_2(r!)+1}{r}}$, we can get the lower bound of the number of r-sized subsets:

$$\binom{2^{\frac{r-1}{r}n+\frac{\log_2(r!)+1}{r}}}{r} \geq \frac{\left(2^{\frac{r-1}{r}n+\frac{\log_2(r!)+1}{r}}\right)^r}{2 \cdot r!} = \frac{2^{(r-1)n} \cdot 2^{\log_2(r!)+1}}{2 \cdot r!} = 2^{(r-1)n}.$$

Hence, the expected number of r-collisions is at least 1, meaning that with high probability, such an r-collision exists among the sampled keys.

Phase 2. From **Phase 1**, we have obtained an r-collision $\{K^{(1)}, \ldots, K^{(r)}\}$ such that $E_{K^{(1)}}(\mathtt{bin}_n(1)) = E_{K^{(2)}}(\mathtt{bin}_n(1)) = \cdots = E_{K^{(r)}}(\mathtt{bin}_n(1))$. Let $L = E_{K^{(1)}}(\mathtt{bin}_n(1))$. From Prop. 1, n bits of Y_i collide for any i. Now, we find an index i and two distinct keys $K^{(a)}$ and $K^{(b)}$ (with $1 \leq a \neq b \leq r$) such that $\mathsf{msb}_{\tau-n}(Y_{i+1}^{(a)}) = \mathsf{msb}_{\tau-n}(Y_{i+1}^{(b)})$.

Given an r-collision $\{K^{(1)}, \ldots, K^{(r)}\}$, for each index i, we can compute $Y_{i+1}^{(1)}, \ldots, Y_{i+1}^{(r)}$ through αr computations of the block cipher, where $\alpha = \lceil \frac{\tau}{n} \rceil$. There are $\binom{r}{2} = \frac{r(r-1)}{2}$ pairs of (a, b), and the expected probability that we have $\mathsf{msb}_{\tau-n}(Y_{i+1}^{(a)}) = \mathsf{msb}_{\tau-n}(Y_{i+1}^{(b)})$ among these is $\frac{r(r-1)}{2^{\tau-n+1}}$. Therefore, we need to examine i from 1 to $\frac{2^{\tau-n+1}}{r(r-1)}$, where $\frac{2^{\tau-n+1}}{r(r-1)} \leq 2^n$ is required to ensure i to fit within an n-bit counter. The total number of block cipher calls in this step is $\alpha r \cdot \frac{2^{\tau-n+1}}{r(r-1)} = \frac{\alpha \cdot 2^{\tau-n+1}}{r-1}$. Once we find $\mathsf{msb}_{\tau-n}(Y_{i+1}^{(a)}) = \mathsf{msb}_{\tau-n}(Y_{i+1}^{(b)})$, let $Y = Y^{(a)}$.

Phase 3. This step does not change from that of the basic extension. When computing XCTR with two keys, we need at most $2 \cdot \frac{2^{\tau-n+1}}{r(r-1)}$ evaluations of E, because $i \leq \frac{2^{\tau-n+1}}{r(r-1)}$.

Attack Complexity. Overall, the attack complexity is as follows.

Phase 1. $2^{\left(\frac{r-1}{r} \cdot n + \frac{\log_2(r!)+1}{r}\right)}$
Phase 2. $\alpha r \cdot \frac{2^{\tau-n+1}}{r(r-1)}$, where $(\alpha - 1)n < \tau \leq \alpha n$
Phase 3. at most $2 \cdot \frac{2^{\tau-n+1}}{r(r-1)}$

Best Choice of r and τ. When $\tau \geq \frac{3n}{2}$ and $\alpha = \lceil \frac{\tau}{n} \rceil$, the attack complexity is minimized by choosing τ and r such that $\frac{2r-1}{r} \cdot n + \frac{\log_2(r!)+1}{r} - \log_2(2\alpha) + \log_2(r-1) \leq \tau < \frac{2(r+1)-1}{r+1} \cdot n + \frac{\log_2((r+1)!)+1}{r+1} - \log_2(2\alpha) + \log_2 r$, where $r \geq 2$ and $\frac{2^{\tau-n+1}}{r(r-1)} \leq 2^n$.

The reason is as follows. The complexity of **Phase 3** is always less than or equal to that of **Phase 2**. Therefore, it is important to minimize the complexity of the sum of **Phases 1** and **2**, which is written as

$$2^{\left(\frac{r-1}{r} \cdot n + \frac{\log_2(r!)+1}{r}\right)} + \alpha r \cdot \frac{2^{\tau-n+1}}{r(r-1)}. \tag{1}$$

In the case of **Phase 1**, its complexity depends only on r, not τ.

Given r from **Phase 1**, when

$$\tau = \frac{2r-1}{r} \cdot n + \frac{\log_2(r!)+1}{r} - \log_2(2\alpha) + \log_2(r-1),$$

the complexities of **Phases 1** and **2** are equal:

$$2^{\frac{r-1}{r} \cdot n + \frac{\log_2(r!)+1}{r}} = \alpha r \cdot \frac{2^{\tau-n+1}}{r(r-1)}.$$

Therefore, as long as **Phase 1** uses an r-collision (not an $(r+1)$-collision), τ is chosen to be less than

$$\frac{2(r+1)-1}{r+1} \cdot n + \frac{\log_2((r+1)!)+1}{r+1} - \log_2(2\alpha) + \log_2 r.$$

In other words, depending on the value of τ, we can choose r accordingly to minimize the total computational cost. Table 2 presents several comparisons for $n = 128$. Figure 7 illustrates how the complexity of our attack and that of the generic attack increases as τ increases for $n = 128$. Even for some values of τ with $\tau \geq 256$, our attack achieves a few bits of improvement over the generic attack complexity.

Comparison with O-notation. Based on the method of selecting r and τ as described above, our new attack shows that when $\tau = O(\frac{2r-1}{r}n)$, the $\mathsf{CMT}^*_{\mathsf{k}}$ security can be broken with $O(2^{\frac{r-1}{r}n})$ block cipher computations. In contrast, a generic attack under the same condition, $t = O(\frac{2r-1}{r}n)$, requires $3 \cdot 2^{\tau/2} = O(2^{\frac{2r-1}{2r}n})$ block cipher computations.

B Concrete Vectors

This Supplementary Material shows concrete vectors generated in the experiment in Section 4.4. Note that hexadecimal values are shown with typewriter fonts. Table 3 shows the vectors for breaking $\mathsf{EtE_A}$-HCTR2 with $\tau = 96$ bits generated in 652 seconds. Note that Table 3 does not show $0^{64}\|M$ and $0^{64}\|M'$ because they are huge, and the readers can reproduce those values by decrypting (K, T, C) and (K', T, C). Meanwhile, Table 4 shows the vectors for breaking $\mathsf{EtE_P}$-HCTR2 with $\tau = 64$ bits generated in 347 second.

Table 3. Concrete vector for $\mathsf{EtE_A}$-HCTR2 with $\tau = 96$ bits. The messages $M\|0^{96}$ and $M'\|0^{96}$ can be obtained by decrypting (K, T, C) and (K', T, C).

Parameter	Value
K	`00000000000000000c7d87ee8847a50f`
K'	`00000000000000000074466e3edc153ddc`
T	`906327a5dc3786e96660a782ec10d8b6...`
	`d9c454da34527885a669584af7f83c81`
C	$0^{343340172736}\|$`729a8c996f47da415de93271`

Table 4. Concrete vector for $\mathsf{EtE_P}$-HCTR2 with $\tau = 64$ bits.

Parameter	Value
K	`813757abdec7b9440000000000000000`
$0^{64}\|M$	$0^{64}\|$`f98f07789515d746792cec14d835417ff11214ebd00ce016`
K'	`6c1def7ba9eea7b70000000000000000`
$0^{64}\|M'$	$0^{64}\|$`44881a620fd50503b69a215279d072b4c6e12aaca03b0cbb`
T	`c78e3737723fa2b319a4b1e9e12a42cf...`
	`561481d35551d73fce33d3460184dd9e`
C	`e6fcd939c22b2c7a0e9bc5ea07d49d87...`
	`d7e28da01174599ec57cc8f8bd91cfea`