# AES-based CCR Hash with High Security and Its Application to Zero-Knowledge Proofs

Hongrui Cui<sup>1</sup>, Chun Guo<sup>2,3,4</sup>, Xiao Wang<sup>5</sup>, Chenkai Weng<sup>6</sup>, Kang Yang<sup>7</sup>, and Yu Yu<sup>1,8</sup>

<sup>1</sup> Shanghai Jiao Tong University, Shanghai, China {rickfreeman,yyuu}@sjtu.edu.cn

<sup>2</sup> School of Cyber Science and Technology, Shandong University, Qingdao, Shandong,

China

<sup>3</sup> Key Laboratory of Cryptologic Technology and Information Security of Ministry of Education, Shandong University, Qingdao, Shandong, 266237, China

<sup>4</sup> Shandong Research Institute of Industrial Technology, Jinan, Shandong, 250102,

China

chun.guo.sc@gmail.com

<sup>5</sup> Northwestern University, Evanston, USA wangxiao@northwestern.edu

<sup>6</sup> Arizona State University, Tempe, USA

car1ckweng@gmail.com

<sup>7</sup> State Key Laboratory of Cryptology, Beijing, China

yangk@sklc.org

<sup>8</sup> Shanghai Qi Zhi Institute, Shanghai, China

Abstract. The recent VOLE-based interactive zero-knowledge (VOLE-ZK) protocols along with non-interactive zero-knowledge (NIZK) proofs based on MPC-in-the-Head (MPCitH) and VOLE-in-the-Head (VOLEitH) extensively utilize the commitment schemes, which adopt a circular correlation robust (CCR) hash function as the core primitive. Nevertheless, the state-of-the-art CCR hash construction by Guo et al. (S&P'20), building from random permutations, can only provide 128-bit security, when it is instantiated from AES. This brings about a gap between AES-based CCR hash function and high security (beyond 128-bit security).

In this paper, we fill this gap by constructing a new CCR hash function from AES, supporting three security levels (i.e., 128, 192 and 256). Using the AES-based CCR hash function, we present an all-but-one vector commitment (AVC) scheme, which constitutes a computationally intensive part of the NIZK proofs from MPCitH and VOLEitH, where these NIZK proofs can in turn be transformed into the promising post-quantum signature candidates. Furthermore, we obtain an efficient VOLE-ZK protocol with security levels higher than 128 from the CCR hash function. Our benchmark results show that the AES-based CCR hash function has a comparable performance with CCR hash functions based on Rijndael with larger block sizes, which is not standardized and has a limited application range. In the AVC context, the expensive commitment component instantiated with our AES-based CCR hash function improves the running time by a factor of  $7 \sim 30 \times$ , compared to the SHA3-based instantiation used in the recent post-quantum signature algorithm FAEST.

**Keywords:** AES-based Circular Correlation Robust Hash Functions · All-but-one Vector Commitment · Interactive Zero Knowledge Protocols · Non-Interactive Zero Knowledge Proofs · High Security Levels

### 1 Introduction

Zero-knowledge (ZK) proofs allow a prover to convince a verifier that a statement is true, in a way that the verifier learns nothing beyond the validity of the statement. ZK proofs have a wide range of applications in, e.g., cryptography, blockchain and machine learning. It is a central objective to improve the efficiency of ZK proofs while achieving the specified security level (e.g., 128, 192, or 256). ZK proofs often adopt the classical "commit-then-prove" paradigm, where a commitment on the witness is first generated, and then the statement is proved in zero knowledge.

A powerful and efficient technique to design non-interactive zero-knowledge (NIZK) proofs is the MPC-in-the-Head (MPCitH) framework [43]. MPCitH enjoys the high efficiency for small to medium-sized circuits, and has been used to design a series of post-quantum signature schemes based on a variety of one-way functions (OWFs) such as AES [26,11,27,44], MPC-friendly OWFs [45,31,47], syndrome decoding [33,3,22] and the multivariate quadratic problem [15,32], including the NIST post-quantum signature candidates [67,2,46,34]. MPCitH adopts an all-but-one vector commitment (AVC) scheme as a crucial building block, where all components of a vector except for one would be opened. Recently, a flurry of interactive ZK protocols [60,30,12,63,35,61,29,62,8,66,28,20,49,65] use subfield vector oblivious linear evaluation (sVOLE) correlations as additively homomorphic commitments (AHCs). Such ZK protocols are called VOLE-ZK, and enjoy the efficiency features of blazing fast end-to-end runtime and scalability to very large circuits. Compared to MPCitH with public verifiability, VOLE-ZK has the shorter proof size and faster running time, but is limited to the designatedverifier setting. Very recently, Baum et al. [7] proposed a new NIZK framework, called VOLE-in-the-head (VOLEitH), which bears resemblance to MPCitH in the spirit of philosophy and makes VOLE-ZK be publicly verifiable. They use VOLEitH to construct a post-quantum signature scheme based solely on AES, referred to as FAEST [6], which has the shorter signature size and comparable signing/verification time, compared to MPCitH. Note that VOLEitH also builds upon the AVC scheme.

Both AVC and AHC schemes described as above build upon the classical GGM tree [36] or the recent optimized GGM tree (called Half-Tree) [41]. <sup>9</sup>

<sup>&</sup>lt;sup>9</sup> For the applications of MPCitH, VOLEitH and VOLE-ZK, compared to GGM tree [36], Half-Tree [41] can reduce the number of AES calls by  $25\% \sim 50\%$ . For the VOLE-ZK application, Half-Tree is also able to reduce the communication of generating AHCs by a half.

Compared to the GGM construction with a pseudorandom generator, the performance of GGM tree can benefit from the circular correlation robust (CCR) hash function, according to the state-of-the-art implementation [39]. Moreover, Half-Tree adopts the CCR hash function as a core primitive. Informally, a CCR hash function H guarantees that for a uniform key  $\Delta$ , it is infeasible to distinguish  $H(x_i \oplus \Delta) \oplus b_i \Delta$  for a set of pairs  $\{(x_i, b_i)\}$  from random strings, where  $(x_i, 0)$  and  $(x_i, 1)$  cannot be queried simultaneously. The CCR hash function is able to be constructed straightforwardly from cryptographic hash functions (e.g., SHA256 and SHA3) in the random oracle model [6,2]. However, given the support of hardware instructions, prior works [13,39] have demonstrated that AES is significantly faster than SHA256 and SHA3. In particular, the benchmark result [39] shows that AES is about 50× faster than SHA256 and 120× faster than SHA3. Therefore, using AES to construct CCR hash functions is a better choice.

The CCR hash functions [39,38,24,56,40], which are constructed with AES in the random-permutation model, have been applied in the efficient AVC and AHC schemes such as [64,63,60,18,56,62,41,66,7,21]. However, due to the fact that AES has a fixed block length of 128 bits for three security levels (i.e., 128, 192 and 256 bits), the existing AES-based CCR constructions [39,38,24,56,40] have a severe limitation, i.e., they cannot provide a security level higher than 128 bits where the key  $\Delta$  has a length of at most 128 bits. Thus, there exists a gap between the standardized AES algorithm and construction of CCR hash functions for both 192-bit and 256-bit security levels.

#### 1.1 Our Contributions

In this paper, we fill the above gap by presenting an AES-based construction of the CCR hash function with three security levels (i.e., 128, 192 and 256 bits). By observing that the MPCitH and VOLEitH frameworks require that the AVC scheme satisfies the so-called extractable binding property, we formulate a new extractability for CCR hash functions, which may be of independent interest. Then, we rigorously prove both CCR and extractable properties of the proposed construction in the ideal-cipher model. Following the work [39], the CCR hash function with both 192-bit and 256-bit security levels can be constructed from the Rijndael algorithm supporting larger block lengths [25], which is not standardized and less studied than AES. In comparison, our CCR construction adopts the standard AES algorithm with better time-tested security, and enjoys the hardware acceleration in the use cases where hardware instructions are only available for the "whole" AES. When the hardware instructions of AES round functions and Rijndael permutations are available to accelerate the Rijndael algorithm, our construction has a comparable performance as indicated by the experiment in Section 7.1.

We apply the AES-based CCR hash function with extractability to design an efficient AVC scheme with three security levels. The AVC scheme integrates the Half-Tree optimization, and can be used to design NIZK proofs and postquantum signature schemes in the MPCitH and VOLEitH frameworks. In the



Fig. 1: The AES-based CCR hash function construction diagram.

AVC scheme of the VOLEitH-based signature scheme FAEST [6], each leaf node in the GGM tree is hashed with SHA3. As a by-product, we replace SHA3 with the AES-based CCR hash function, and achieve an improvement of  $7 \sim 30 \times$ in terms of running time as indicated by our experiment in Section 7.2. The recent work [21] to improve the AVC scheme based on Half-Tree uses Rijndael to achieve the 192-bit and 256-bit security levels, and gives a specified construction in the random permutation model. In comparison, our AVC scheme enjoys the standard AES algorithm to achieve security levels beyond 128 bits as well as a modular and generic construction using extractable CCR hash functions. In addition, the techniques underlying our AVC scheme can also be used to improve the performance of another recent AVC scheme [5].

We also apply the AES-based CCR hash function to construct an sVOLE protocol in the PCG framework [17,19], by combining it with the learning parity with noise (LPN) problem. The sVOLE protocol enjoys the complete picture of three security levels and the high efficiency by integrating the Half-Tree optimization. When using sVOLE correlations as AHCs, we can directly obtain an efficient VOLE-ZK protocol following prior works [60,30,12,63,35,29,62,66,20].

### 2 Technical Overview

The main technical difficulty is that AES only has 128-bit block size regardless of the security parameter. Therefore, we have to design hash functions working on an input length longer than the block size of the underlying block cipher. We solve the problem by constructing the hash output from multiple blocks, as shown in Fig. 1. For a block cipher with *n*-bit block size and  $\lambda$ -bit key size, the left *n*-bit input  $x_L$  of the CCR hash function serves as the input of the block cipher (after applying an orthomorphism) while the rest  $(\lambda - n)$ -bit input  $x_R$  of the CCR hash function serves as part of the block cipher secret key. Since our focus is the AES algorithm, which has a fixed block size 128 bits and variable key lengths of 128, 192 and 256 bits, we turn our attention to the  $n \leq \lambda \leq 2n$  parameter range. For  $\lambda = n$ , one block cipher invocation is sufficient and the construction is identical to the CCR hash in [39]. For  $n < \lambda \leq 2n$ , we need two invocations of the block cipher keyed with the input value  $x_R$  concatenated with distinct public constants.

For input length  $\lambda \in \mathbb{N}$ , the circular correlation robust (CCR) property for a hash function  $H : \{0,1\}^{\lambda} \to \{0,1\}^{\lambda}$  states that for a uniformly random key  $\Gamma \in \{0,1\}^{\lambda}$ , the oracle  $\mathcal{O}(x,b) = H(x \oplus \Gamma) \oplus b \cdot \Gamma$  should be indistinguishable from an ideal random oracle, where  $x \in \{0,1\}^{\lambda}, b \in \{0,1\}^{.1}$  By applying the domain separation technique as mentioned above, the key  $\Gamma$  is distributed into both the input block and the secret key of the block cipher. Intuitively, by working in the ideal cipher model, information about the key  $\Gamma$  can only be leaked when an ideal cipher query made by the adversary collides with some internal ideal cipher invocation made by the CCR oracle, or two internal ideal cipher invocations made by two distinct CCR oracle calls collides. Since the key  $\Gamma$  is distributed to both the secret key and the input block of the ideal cipher, a collision implies recovering the key  $\Gamma$ , which happens with probability  $2^{-\lambda}$ . By applying a union bound on all possible collision events, we conclude that the construction in Fig. 1 is indeed CCR-secure. The above intuition is formally argued using the H-coefficient technique [51] in Section 4.2.

Motivated by the extractable binding property of AVC schemes [7], we formulate the extractable property of CCR hash functions. Intuitively, the property states that a hash function H is extractable, if given two purported hash outputs with correlated inputs  $y = \text{Com}(x) = \text{H}(x \oplus \text{C}_2) || \text{H}(x \oplus \text{C}_3)$  (C<sub>2</sub> and C<sub>3</sub> are two public constants) and the ideal cipher transcript, there exists an efficient extractor that can extract the input x. Moreover, when the extraction fails, any efficient adversary cannot provide a valid x such that the correlation holds.

Intuitively, since in the ideal cipher model the adversary has no way of acquiring the ideal cipher output other than querying the ideal cipher oracle, all information about a hash function call made by the adversary is recorded in the ideal cipher transcript, from which the extractor can easily acquire the possible inputs. Nevertheless, extraction would fail if adversary finds collision in hash function output or succeeds in inverting a hash image that has not been queried before. We bound both probabilities by using the technique in [48]. In particular, by granting the adversary "free" queries, we manage to separate the ideal cipher queries into "piles" such that the **Com** invocation on a set of input values can only incur queries within this pile. Therefore, further analysis can be conducted on the pile-level. Moreover, to capture the fact that the adversary can learn almost all information about the permutation related to a single key when it acquires too many input-output pairs, we further grant the adversary with a "super query" such that when half of the inputs related to a single key

<sup>&</sup>lt;sup>1</sup> A technical detail is that the distinguisher should not query the same x with both 0 and 1 to prevent a trivial attack.

have been queried, the remaining half and the corresponding output values are given to the adversary as well. Crucially, the additional input-output pairs do not count towards the total number of queries that the adversary makes to the ideal cipher. The intuition is that for a pile returned by a normal query, much information about the related key is unknown to the adversary and the pile itself has sufficient entropy such that collision and inversion is unlikely. On the other hand, a pile returned by a super query has not much entropy to the adversary, but to get a super query the adversary must spend many normal queries and the total number of super queries is limited. By formalizing this intuition, we upper bound the probability of collision to be  $O(q_E/2^{\lambda} + q_E^2/2^{2\lambda})$  and the probability of pre-image inversion to be  $O(q_E/2^{\lambda})$ , where  $q_E$  is the number of ideal cipher queries of the adversary.

Using the derived bounds, we are able to capture the the extraction failure probability of our CCR hash construction and derive the precise security bounds of the proposed CCR hash function with respect to the new extractable CCR definition. We note that, since our construction reduces to the extended MMO construction when the block size equals to the security parameter, our proof also implies the extractable security of the hash function in [39] under the random permutation model.

We demonstrate two applications for the AES-based extractable CCR hash function. The first one is an AVC scheme that supports security levels of 128, 192 and 256 bits from AES. In particular, we instantiate the correlated GGM tree construction with the CCR hash function while using the CCR-based commitment scheme Com in the leaf nodes. For AVC's hiding property, we show that the adversary's view in the hiding game can be simulated by calling the oracle in the CCR game. For the binding property, the definition of CCR's extractability allows the extractor of the AVC's extractable binding game to invoke the extractor of the CCR's extractability game in the ideal cipher model. Moreover, the extraction failure bound of the AVC's extractable binding game can also be derived straightforwardly from a union bound on the extraction failure events of each leaf node commitment. Therefore, the extractable binding property of the AVC scheme can be reduced to the extractability property of the CCR hash function. In the second application, we instantiate the sVOLE protocol in [41] with the AES-based CCR hash function. Since the sVOLE protocol in [41] builds on CCR hash functions in a black-box manner, we automatically get an AES-based sVOLE protocol with three security levels.

### **3** Preliminaries

We list the notations of this paper in Section 3.1. In Section 3.2 we recall the definition of circular correlation robust hash functions and in Section 3.3 we recall the H-coefficient technique which underlies our proofs.

#### 3.1 Notation

We use  $\lambda$  to denote the computational security parameter. We use log to denote logarithms in base 2. We define  $[a, b] = \{a, \ldots, b-1\}$  and write  $[a, b] = \{a, \ldots, b\}$ . We write  $x \leftarrow S$  to denote sampling x uniformly at random from a finite set S. We use  $\{x_i\}_{i \in S}$  to denote the set that consists of all elements with indices in set S. When the context is clear, we abuse the notation and use  $\{x_i\}$  to denote such a set.

We use bold lower-case letters like  $\boldsymbol{a}$  for column vectors and bold uppercase letters like  $\boldsymbol{A}$  for matrices. We let  $a_i$  denote the *i*-th component of  $\boldsymbol{a}$  (with  $a_0$ the first entry) and  $\boldsymbol{a}[i, j]$  denote the sub-vector of  $\boldsymbol{a}$  with indices [i, j].

Let *E* denote the encryption algorithm of a block cipher with a block size of *n* bits. Since we only consider the AES algorithm in this work, we assume that the key size always equals the security parameter  $\lambda$ . Let  $[x]_n \in \{0,1\}^n$  be the binary representation of an integer  $x \in \mathbb{Z}_{2^n}$ . In our construction we also utilize an orthomorphism  $\sigma : \{0,1\}^n \to \{0,1\}^n$ , which means that

- $-\sigma$  is linear,
- $-\sigma$  is a permutation,
- $\sigma \oplus id$  is a permutation.

We can instantiate  $\sigma$  as  $\sigma(x_L || x_R) = x_R \oplus x_L || x_L$  where  $x_L, x_R$  are the left and right halves of the input x.

#### 3.2 Security Definition

We recall the definition of circular correlation robustness of [39] (in the ideal cipher model) in Definition 1.

**Definition 1.** Let  $\mathsf{H}^E : \{0,1\}^{\lambda} \to \{0,1\}^{\lambda}$  be a function defined upon an ideal cipher E. For  $\Gamma \in \{0,1\}^{\lambda}$ , define  $\mathcal{O}_{\Gamma}^{\mathsf{ccr}}(w,b) = \mathsf{H}^E(w \oplus \Gamma) \oplus b \cdot \Gamma$ . We don't allow the distinguisher to query the same w with both b = 0 and b = 1 to avoid the trivial attack. For a distinguisher  $\mathcal{D}$ , we define the following advantage

$$\mathsf{Adv}_{\mathsf{H}}^{\mathsf{ccr}} := \left| \Pr_{\Gamma \leftarrow \{0,1\}^{\lambda}} [\mathcal{D}_{\Gamma}^{\mathcal{O}_{\Gamma}^{\mathsf{ccr}}(\cdot), E(\cdot, \cdot), E^{-1}(\cdot, \cdot)}(1^{\lambda}) = 1] - \Pr_{f \leftarrow \mathcal{F}_{\lambda+1, \lambda}} [\mathcal{D}^{f(\cdot), E(\cdot, \cdot), E^{-1}(\cdot, \cdot)}(1^{\lambda}) = 1] \right|$$

where  $\mathcal{F}_{\lambda+1,\lambda}$  denotes the set of all functions mapping  $(\lambda+1)$ -bit inputs to  $\lambda$ -bit outputs. H is  $(q_E, q_C, \epsilon)$ -circular correlation robust if for all  $\mathcal{D}$  making at most  $q_E$  queries to E and  $E^{-1}$  and at most  $q_C$  queries to the oracle we have  $\mathsf{Adv}_{\mathsf{H}}^{\mathsf{ccr}} \leq \epsilon$ .

#### 3.3 The H-coefficient Technique

We will employ Patarin's H-coefficient technique [51] to prove the circular correlation robustness and pseudorandomness of our new building blocks. We provide a brief overview of its main ingredients here (we refer to [51,23] for complete elaborations). Our presentation borrows heavily from that of [23]. Fix a distinguisher D that makes a bounded number of queries to its oracles. As in the security definition presented above, D's aim is to distinguish between two worlds: a "real world" and an "ideal world". Assume wlog that D is deterministic. The execution of D defines a *transcript* that includes the sequence of queries and answers received from its oracles; D's output is a deterministic function of its transcript. Thus, if  $T_{\rm re}, T_{\rm id}$  denote the probability distributions on transcripts induced by the real and ideal worlds, respectively, then D's distinguishing advantage is upper bounded by the statistical distance

$$\Delta(T_{\rm re}, T_{\rm id}) := \frac{1}{2} \sum_{\mathcal{Q}} \left| \Pr[T_{\rm re} = \mathcal{Q}] - \Pr[T_{\rm id} = \mathcal{Q}] \right|,\tag{1}$$

where the sum is taken over all possible transcripts Q.

Let  $\mathcal{T}$  denote the set of all transcripts such that  $\Pr[T_{\mathsf{id}} = \mathcal{Q}] > 0$  for all  $\mathcal{Q} \in \mathcal{T}$ . We look for a partition of  $\mathcal{T}$  into two sets  $\mathcal{T}_1$  and  $\mathcal{T}_2$  of "good" and "bad" transcripts, respectively, along with a constant  $\epsilon_1 \in [0, 1)$  such that

$$\mathcal{Q} \in \mathcal{T}_1 \implies \Pr[T_{\mathsf{re}} = \mathcal{Q}] / \Pr[T_{\mathsf{id}} = \mathcal{Q}] \ge 1 - \epsilon_1.$$
 (2)

It is then possible to show (see [23] for details) that

$$\Delta(T_{\mathsf{re}}, T_{\mathsf{id}}) \le \epsilon_1 + \Pr[T_{\mathsf{id}} \in \mathcal{T}_2] \tag{3}$$

is an upper bound on the distinguisher's advantage. One should think of  $\epsilon_1$  and  $\Pr[T_{id} \in \mathcal{T}_2]$  as "small", so "good" transcripts have nearly the same probability of appearing in the real world and the ideal world, whereas "bad" transcripts have low probability of occurring in the ideal world.

An appeal of the H-coefficient technique is that it handles arbitrary adaptive distinguishers even though the technical calculations involve a *posteriori* probabilities computed with respect to finalized transcripts.

### 4 AES-based Extractable CCR Hash Function

In this section, we first present a block-cipher-based CCR hash function  $H_{CCR}$  with a bit security level higher than the block size in Section 4.1. We then prove our CCR security claim in Section 4.2. Then, in Section 4.3 we propose and prove an extractability notion of  $H_{CCR}$  so that it becomes a multi-purpose construction.

#### 4.1 Construction of $H_{CCR}$

Our AES-based CCR hash functions are described in Fig. 2, where we treat the three possible key lengths of AES (i.e. 128, 192, or 256) separately. To analyze their provable security, in Fig. 3, we present a theoretical CCR hash construction built upon an ideal cipher of *n*-bit block size and  $\lambda$ -bit keys, where  $\lambda$  is also the security parameter.

Fig. 2: The AES-based CCR hash function. Here  $C_0$  and  $C_1$  are two distinct 128-bit public constants, r is a  $\lambda$ -bit value,  $[\cdot]_{128}$  denotes the 128-bit binary representation,  $\sigma$  is an orthomorphism on 128-bit values,  $\mathsf{left}_n$  (resp. right<sub>n</sub>) denotes taking the left (resp. right) n bits of the input.

#### 4.2 The Circular Correlation Robustness of H<sub>CCR</sub>

We prove the security of the abstract construction in Fig. 3 using the H-coefficient technique. We refer unfamiliar readers to Section 3.3 for a brief introduction to this proof technique.

**Lemma 1.** If we model E as an ideal cipher, then the hash function  $\mathsf{H}^{E}_{\mathsf{CCR}}$  defined in Fig. 3 is  $(q_E, q_C, \varepsilon)$ -circular correlation robust, where

$$\varepsilon \le \frac{4q_E q_C}{2^{\lambda}} + \frac{q_C(q_C - 1)}{2^n}.$$
(4)

*Proof.* Fix a deterministic distinguisher D making queries to two oracles. The first is an ideal cipher oracle (and its inverse) where the key size is  $\lambda$ -bit and the block space is n-bit. In the real world, the second oracle is  $\mathcal{O}_{\Gamma}^{\mathsf{CCR}}(w,b) = \mathsf{H}_{\mathsf{CCR}}(\Gamma \oplus w) \oplus b \cdot \Gamma$  (for  $\Gamma$  uniformly sampled from  $\{0,1\}^{\lambda}$ ); Whereas in the ideal world it is an independent random function from  $\{0,1\}^{\lambda+1}$  to  $\{0,1\}^{\lambda}$ .

Recall from Fig. 3 that  $n \leq \lambda \leq 2n$ ,  $r_L = \mathsf{left}_n(r)$  and  $r_R = \mathsf{right}_{\lambda-n}(r)$ . We (have to) employ a special treatment for the case where  $n < \lambda < 2n$  (which means  $n \nmid \lambda$ ). Concretely, in this case, instead of defining  $z_L \leftarrow E(r_R \| \mathsf{C}_0, \sigma(r_L)) \oplus \sigma(r_L)$ and  $z_R \leftarrow \mathsf{left}_{\lambda-n}(E(r_R \| \mathsf{C}_1, \sigma(r_L)) \oplus \sigma(r_L))$  as in Fig. 3, we define  $z_L \leftarrow E(r_R \| \mathsf{C}_0, \sigma(r_L)) \oplus \sigma(r_L)$  and  $z_R \leftarrow E(r_R \| \mathsf{C}_1, \sigma(r_L)) \oplus \sigma(r_L)$ , i.e., the truncation is removed; and further  $(\mathcal{O}')_{\Gamma}^{\mathsf{ccr}}(x, b) = (z_L \| z_R) \oplus b \cdot (\Gamma \| [0]_{2n-\lambda})$ , i.e., the output of  $(\mathcal{O}')_{\Gamma}^{\mathsf{ccr}}$  is of 2n bits. We focus on indistinguishability of  $(\mathcal{O}')_{\Gamma}^{\mathsf{ccr}}$  and an expanding random function  $f' : \{0,1\}^{\lambda} \times \{0,1\} \to \{0,1\}^{2n}$ : note that this implies indistinguishability of the original  $\mathcal{O}_{\Gamma}^{\mathsf{ccr}}$  and  $f : \{0,1\}^{\lambda} \times \{0,1\} \to \{0,1\}^{\lambda}$ .

With the above in mind, let  $Q_E$  and  $Q_O$  be the transcripts of ideal cipher and construction queries and responses respectively, where  $Q_E = \{(k_1, x_1, y_1), \ldots\}$ 

Algorithm  $\mathsf{H}^{E}_{\mathsf{CCR}}(1^{\lambda}, 1^{n}, r)$ //  $\lambda$ : security parameter and key-size of E; n: block size of E. // Note that  $|r| = \lambda$ . - If  $n \leq \lambda \leq 2n \ll n + \lambda$ : 1.  $r_{L} \leftarrow \mathsf{left}_{n}(r), r_{R} \leftarrow \mathsf{right}_{\lambda-n}(r)$ 2.  $z_{L} \leftarrow E(r_{R} || \mathsf{C}_{0}, \sigma(r_{L})) \oplus \sigma(r_{L})$ 3.  $z_{R} \leftarrow \mathsf{left}_{\lambda-n}(E(r_{R} || \mathsf{C}_{1}, \sigma(r_{L})) \oplus \sigma(r_{L}))$  // omit if  $n = \lambda$ 4. return  $z_{L} || z_{R}$ 

Fig. 3: The abstract model for the AES-based CCR hash function. Here  $C_0$  and  $C_1$  are two distinct *n*-bit public constants, *r* is a  $\lambda$ -bit value,  $\sigma$  is an orthomorphism on *n*-bit values, left<sub>n</sub> (resp. right<sub>n</sub>) denotes taking the left (resp. right) *n* bits of the input.

records the queries/answers to/from E or  $E^{-1}$  (with  $(k, x, y) \in Q_E$  meaning E(k, x) = y) and  $Q_{\mathcal{O}} = \{(w_1, b_1, z_1), \ldots\}$  records the queries/answers to/from the second oracle.

Denote the transcript of D's interaction by  $\mathcal{Q} = (\mathcal{Q}_E, \mathcal{Q}_O, \Gamma)$ , where  $\mathcal{Q}_E = \{(k_1, x_1, y_1), \ldots\}$  records the queries/answers to/from E or  $E^{-1}$  (with  $(k, x, y) \in \mathcal{Q}_E$  meaning E(k, x) = y) and  $\mathcal{Q}_O = \{(w_1, b_1, z_1), \ldots\}$  records the queries/answers to/from the second oracle. Following [23,39], a key  $\Gamma$  is appended to the transcript (even though it is not part of the view of D) to facilitate the analysis: in the real world, this is the key used by the second oracle, whereas in the ideal world, it is simply an independent key sampled from  $\{0, 1\}^{\lambda}$ . A transcript  $\mathcal{Q}$  is attainable for some fixed D if there exist some oracles such that the interaction of D with those oracles would lead to transcript  $\mathcal{Q}$ .

For notational simplicity we define  $\Gamma_L := \mathsf{left}_n(\Gamma)$  and  $\Gamma_R := \mathsf{right}_{\lambda-n}(\Gamma)$ . We assume that the key size of E always equals the security parameter  $\lambda$ . We also recall that for each  $(w, b, z) \in \mathcal{Q}_{\mathcal{O}}$  in the real world we have

$$z_{L} = E(w_{R} \oplus \Gamma_{R} \| \mathsf{C}_{0}, \sigma(w_{L} \oplus \Gamma_{L})) \oplus \sigma(w_{L} \oplus \Gamma_{L}) \oplus b \cdot \Gamma_{L}$$
  
$$z_{R} = E(w_{R} \oplus \Gamma_{R} \| \mathsf{C}_{1}, \sigma(w_{L} \oplus \Gamma_{L})) \oplus \sigma(w_{L} \oplus \Gamma_{L}) \oplus b \cdot (\Gamma_{R} \| [0]_{2n-\lambda})$$

(Recall that we have removed the truncation in the case where  $n < \lambda < 2n$ , so that  $|z_R|$  always equals n.)

We say a transcript  $(\mathcal{Q}_E, \mathcal{Q}_{\mathcal{O}}, \Gamma)$  is *bad* if either of the following conditions is fulfilled:

- (B-1) There is a query  $(w, b, z) \in \mathcal{Q}_{\mathcal{O}}$  and a query of the following form in  $\mathcal{Q}_E$ . •  $(w_R \oplus \Gamma_R \| \mathsf{C}_0, \sigma(w_L \oplus \Gamma_L), \star)$ 
  - $(w_R \oplus \Gamma_R \| \mathsf{C}_1, \sigma(w_L \oplus \Gamma_L), \star)$  in the case of  $n < \lambda \leq 2n$
  - $(w_R \oplus \Gamma_R \| \mathsf{C}_0, \star, z_L \oplus \sigma(w_L \oplus \Gamma_L) \oplus b \cdot \Gamma_L)$
  - $(w_R \oplus \Gamma_R \| \mathsf{C}_1, \star, z_R \oplus \sigma(w_L \oplus \Gamma_L) \oplus b \cdot (\Gamma_R \| [0]_{2n-\lambda}))$  in the case of  $n < \lambda \leq 2n$

- (B-2) There are distinct  $(w^i, b^i, z^i)$ ,  $(w^j, b^j, z^j) \in \mathcal{Q}_{\mathcal{O}}$  subject to  $w_R^i = w_R^j$  where either of the following two cases occurs.
  - $\sigma(w_L^i) \oplus b^i \cdot \Gamma_L \oplus z_L^i = \sigma(w_L^j) \oplus b^j \cdot \Gamma_L \oplus z_L^j$
  - $\sigma(w_L^i) \oplus b^i \cdot (\Gamma_R || [0]_{2n-\lambda}) \oplus z_R^i = \sigma(w_L^j) \oplus b^j \cdot (\Gamma_R || [0]_{2n-\lambda}) \oplus z_R^j$  in the case of  $n < \lambda \leq 2n$ . (Again, since we have removed the truncation, it always holds  $|z_R^j|, |z_R^j| = n$ .)

We now bound the probabilities of these events in the ideal world, beginning with (B-1). For every pair of records  $((w, b, z), (k, x, y)) \in \mathcal{Q}_{\mathcal{O}} \times \mathcal{Q}_{E}$ , we bound the four cases separately.

**Case 1.** The bad event is  $(w_R \oplus \Gamma_R || C_0 = k) \land (\sigma(w_L \oplus \Gamma_L) = x)$ . Using the linear property of  $\sigma$  and rearranging the terms, this is equivalent to

$$(\Gamma_R = \mathsf{left}_{\lambda-n}(k) \oplus w_R) \land (\sigma(\Gamma_L) = x \oplus \sigma(w_L))$$

Since  $\Gamma$  is uniformly sampled from  $\{0,1\}^{\lambda}$  and  $\sigma$  is a permutation, we have

$$\Pr\left[\left(\Gamma_R = \mathsf{left}_{\lambda-n}(k) \oplus w_R\right) \land \left(\sigma(\Gamma_L) = x \oplus \sigma(w_L)\right)\right] \le 2^{-\lambda}$$

**Case 2.** The bad event is  $(w_R \oplus \Gamma_R || C_1 = k) \land (\sigma(w_L \oplus \Gamma_L) = x)$ . Using the same argument as the first case, we conclude that the probability of this event is bounded by  $2^{-\lambda}$ .

**Case 3.** The bad event is  $(w_R \oplus \Gamma_R || \mathsf{C}_0 = k) \land (z_L \oplus \sigma(w_L \oplus \Gamma_L) \oplus b \cdot \Gamma_L = y)$ . Consider the following two sub-cases. If b = 0 then the event is equivalent to  $(\Gamma_R = w_R \oplus \mathsf{left}_{\lambda-n}(k)) \land (\sigma(\Gamma_L) = z_L \oplus \sigma(w_L) \oplus y)$ , which happens except with probability  $2^{-\lambda}$ . Otherwise we have b = 1 and the event is  $(\Gamma_R = w_R \oplus \mathsf{left}_{\lambda-n}(k)) \land (\sigma(\Gamma_L) \oplus \Gamma_L = z_L \oplus \sigma(w_L) \oplus y)$ . Since  $\sigma$  is an orthomorphism, the map  $x \mapsto \sigma(x) \oplus x$  is a permutation. We conclude that this event also happens except with probability  $2^{-\lambda}$ .

**Case 4.** The bad event is equivalent with  $(w_R \oplus \Gamma_R || C_1 = k) \land (z_R \oplus \sigma(w_L \oplus \Gamma_L) \oplus b \cdot (\Gamma_R || [0]_{2n-\lambda}) = y)$ . Rearranging the terms we have

$$\begin{aligned} &\Pr\left[\left(w_{R}\oplus\Gamma_{R}\|\mathsf{C}_{1}=k\right)\wedge\left(z_{R}\oplus\sigma(w_{L}\oplus\Gamma_{L})\oplus b\cdot\Gamma_{R}=y\right)\right]\\ &=\Pr\left[\Gamma_{R}=\mathsf{left}_{\lambda-n}(k)\oplus w_{R}\right]\cdot\\ &\Pr\left[\sigma(\Gamma_{L})=z_{R}\oplus\sigma(w_{L})\oplus b\cdot\left(\Gamma_{R}\|[0]_{2n-\lambda}\right)\oplus y\mid\Gamma_{R}=\mathsf{left}_{\lambda-n}(k)\oplus w_{R}\right]\\ &<2^{-\lambda}\end{aligned}$$

Taking a union bound over all four cases and summing over the  $q_E q_C$  pairs in  $\mathcal{Q}_{\mathcal{O}} \times \mathcal{Q}_E$ , we conclude that the probability of (B-1) in the ideal world is bounded by  $\frac{4q_E q_C}{2\lambda}$ .

For (B-2), consider distinct  $(w^i, b^i, z^i), (w^j, b^j, z^j) \in \mathcal{Q}_{\mathcal{O}}$ . Note that even if we take condition on the value of  $\Gamma$ , the values  $z^i, z^j$  are uniform and independent. Thus, for side  $\in \{L, R\}$ 

$$\Pr[\sigma(w^i_{\mathsf{side}} \oplus w^j_{\mathsf{side}}) \oplus (b^i_{\mathsf{side}} \oplus b^j_{\mathsf{side}}) \cdot \Gamma_{\mathsf{side}} = z^i_{\mathsf{side}} \oplus z^j_{\mathsf{side}}] = 2^{-n}.$$

Taking a union bound over side  $\in \{L, R\}$  and the  $\binom{q_C}{2} = q_C(q_C - 1)/2$  pairs of distinct queries, the probability of (B-2) is at most  $\frac{q_C(q_C - 1)}{2^n}$ .

Fix a good transcript  $(\mathcal{Q}_E, \mathcal{Q}_O, \Gamma)$ . The probability that the ideal world is consistent with this transcript is

$$\begin{cases} \frac{1}{2^{n}} \cdot \frac{1}{(2^{n})_{q_{E,1}} \cdot \dots \cdot (2^{n})_{q_{E,\mu}}} \cdot \frac{1}{2^{q_{C}n}} & \text{if } \lambda = n \\ \frac{1}{2^{\lambda}} \cdot \frac{1}{(2^{n})_{q_{E,1}} \cdot \dots \cdot (2^{n})_{q_{E,\mu}}} \cdot \frac{1}{2^{2q_{C}n}} & \text{if } n < \lambda \le 2n \end{cases}$$
(5)

where  $\mu \in [1, q_E]$  denote the number of distinct keys in  $\mathcal{Q}_E$  and  $q_{E,1}, ..., q_{E,\mu}$  denotes the number of entries in  $\mathcal{Q}_E$  under the respective keys. Again such a result is obtained due to removing the truncation in the case  $n \nmid \lambda$ .

The probability that the real world is consistent with this transcript is

$$\begin{cases} \frac{1}{2^{\lambda}} \cdot \frac{1}{(2^{n})_{q_{E,1}} \cdots (2^{n})_{q_{E,\mu}}} \cdot \Pr[\forall(w,b,z) \in \mathcal{Q}_{\mathcal{O}} : \mathcal{O}_{\Gamma}^{\mathsf{CCR}}(w,b) = z \mid E \vdash \mathcal{Q}_{E}] & \text{if } n \mid \lambda \\ \frac{1}{2^{\lambda}} \cdot \frac{1}{(2^{n})_{q_{E,1}} \cdots (2^{n})_{q_{E,\mu}}} \cdot \Pr[\forall(w,b,z) \in \mathcal{Q}_{\mathcal{O}} : (\mathcal{O}')_{\Gamma}^{\mathsf{CCR}}(w,b) = z \mid E \vdash \mathcal{Q}_{E}] & \text{if } n \nmid \lambda \end{cases}$$

Recall that  $(\mathcal{O}')_{\Gamma}^{\mathsf{CCR}}$  is the construction oracle obtained by removing the truncation.

We can express the last term of the above as

$$\begin{cases} \prod_{i=1}^{q_C} \Pr[\mathcal{O}_{\Gamma}^{\mathsf{CCR}}(w^i, b^i) = z^i \mid E \vdash \mathcal{Q}_E \land \forall j < i : \mathcal{O}_{\Gamma}^{\mathsf{CCR}}(w^j, b^j) = z^j] & \text{if } n \mid \lambda \\ \prod_{i=1}^{q_C} \Pr[(\mathcal{O}')_{\Gamma}^{\mathsf{CCR}}(w^i, b^i) = z^i \mid E \vdash \mathcal{Q}_E \land \forall j < i : (\mathcal{O}')_{\Gamma}^{\mathsf{CCR}}(w^j, b^j) = z^j] & \text{if } n \nmid \lambda \end{cases}$$

Consider the "original" oracle  $\mathcal{O}_{\Gamma}^{\mathsf{CCR}}$  first. Note that  $\mathcal{O}_{\Gamma}^{\mathsf{CCR}}(w^{i}, b^{i}) = z^{i}$  iff  $\mathsf{H}_{\mathsf{CCR}}(\Gamma \oplus w^{i}) \oplus b^{i} \cdot \Gamma = z^{i}$ , i.e.,

$$(z_i)_L = E((w_i)_R \oplus \Gamma_R \| \mathsf{C}_0, \sigma((w_i)_L \oplus \Gamma_L)) \oplus \sigma((w_i)_L \oplus \Gamma_L) \oplus b_i \cdot \Gamma_L (z_i)_R = E((w_i)_R \oplus \Gamma_R \| \mathsf{C}_1, \sigma((w_i)_L \oplus \Gamma_L)) \oplus \sigma((w_i)_L \oplus \Gamma_L) \oplus b_i \cdot \Gamma_R \text{ (if } \lambda = 2n)$$

Since the transcript is good there is no query of the form  $((w_i)_R \oplus \Gamma_R || [0]_n, \sigma(w_L \oplus (w_i)_L), \star)$  or  $((w_i)_R \oplus \Gamma_R || [1]_n, \sigma(w_L \oplus (w_i)_L), \star)$  in  $\mathcal{Q}_E$  (since (B-1) does not occur), nor is  $E((w_i)_R \oplus \Gamma_R || [0]_n, \sigma((w_i)_L \oplus \Gamma_L))$  or  $E((w_i)_R \oplus \Gamma_R || [1]_n, \sigma((w_i)_L \oplus \Gamma_L))$  determined by the fact that  $\mathcal{O}_{\Gamma}^{\mathsf{CCR}}(w_j, b_j) = z_j$  for all j < i (since D does not make two queries to  $\mathcal{O}_{\Gamma}^{\mathsf{CCR}}$  with the same  $w_i$ ).

Similarly, there is no query of the form  $((w_i)_R \oplus \Gamma_R || [0]_n, \star, \sigma((w_i)_L \oplus \Gamma_L) \oplus b_i \cdot \Gamma_L \oplus z_i)$  or  $((w_i)_R \oplus \Gamma_R || [1]_n, \star, z_R \oplus \sigma((w_i)_L \oplus \Gamma_L) \oplus b_i \cdot \Gamma_R)$  in  $\mathcal{Q}_E$  (since (B-1) does not occur), nor is  $E^{-1}((w_i)_R \oplus \Gamma_R || [0]_n, \sigma((w_i)_L \oplus \Gamma_L) \oplus b_i \cdot \Gamma_L \oplus (z_i)_L)$  or  $E^{-1}((w_i)_R \oplus \Gamma_\Gamma || [1]_n, \sigma((w_i)_L \oplus \Gamma_L) \oplus b_i \cdot \Gamma_R \oplus (z_i)_R)$  determined by the fact that  $\mathcal{O}_{\Gamma}^{\mathsf{CCR}}(w_j, b_j) = z_j$  for all j < i (since (B-2) does not occur). Thus, for all i we have

$$\begin{aligned} &\Pr[\mathcal{O}_{\Gamma}^{\mathsf{CCR}}(w^{i},b^{i})=z^{i}\mid E\vdash\mathcal{Q}_{E}\wedge\forall j< i:\mathcal{O}_{\Gamma}^{\mathsf{CCR}}(w^{j},b^{j})=z^{j}]\\ &\geq \frac{1}{2^{n}-q_{E}-i+1}\cdot\frac{1}{2^{\lambda-n}-q_{E}-i+1} \geq \frac{1}{2^{\lambda}}\qquad (\lambda\in\{n,2n\}). \end{aligned}$$

On the other hand,  $(\mathcal{O}')^{\mathsf{CCR}}_{\Gamma}(w^i, b^i) = z^i$  iff

$$(z_i)_L = E((w_i)_R \oplus \Gamma_R \| \mathsf{C}_0, \sigma((w_i)_L \oplus \Gamma_L)) \oplus \sigma((w_i)_L \oplus \Gamma_L) \oplus b_i \cdot \Gamma_L (z_i)_R = E((w_i)_R \oplus \Gamma_R \| \mathsf{C}_1, \sigma((w_i)_L \oplus \Gamma_L)) \oplus \sigma((w_i)_L \oplus \Gamma_L) \oplus b_i \cdot (\Gamma_R \| [0]_{2n-\lambda}).$$

In a similar vein to the above, it can be seen

$$\Pr[(\mathcal{O}')_{\Gamma}^{\mathsf{CCR}}(w^i, b^i) = z^i \mid E \vdash \mathcal{Q}_E \land \forall j < i : (\mathcal{O}')_{\Gamma}^{\mathsf{CCR}}(w^j, b^j) = z^j]$$
$$\geq \frac{1}{2^n - q_E - i + 1} \cdot \frac{1}{2^n - q_E - i + 1} \geq \frac{1}{2^{2n}}.$$

It follows that

$$\Pr[\forall (w, b, z) \in \mathcal{Q}_{\mathcal{O}} : \mathcal{O}_{\Gamma}^{\mathsf{CCR}}(w, b) = z \mid E \vdash \mathcal{Q}_{E}] \ge 1/2^{q_{C}\lambda} \quad (\lambda \in \{n, 2n\}),$$
  
$$\Pr[\forall (w, b, z) \in \mathcal{Q}_{\mathcal{O}} : (\mathcal{O}')_{\Gamma}^{\mathsf{CCR}}(w, b) = z \mid E \vdash \mathcal{Q}_{E}] \ge 1/2^{2q_{C}n},$$

and so the probability that the real world is consistent with the transcript is always at least (5). This completes the proof.

#### 4.3 The Extractable Binding Property of H<sub>CCR</sub>

We show that in the ideal cipher model, we can extract the pre-image of  $H_{CCR}$  given output of sufficient length. In particular, we define an extractability property for a hash function H and then prove that the construction  $H_{CCR}$  in Fig. 3 satisfies this property.

**Definition 2.** Let  $\mathsf{H}^E : \{0,1\}^{\lambda} \to \{0,1\}^{\lambda}$  be a hash function and  $C_2, C_3 \in \{0,1\}^{\lambda}$  be two public distinct constants. Let  $\mathsf{Com}(m) := \mathsf{H}^E(m \oplus C_2) \| \mathsf{H}^E(m \oplus C_3)$  be a commitment scheme and  $\mathsf{Ext}^E(c, \mathcal{Q}_E)$  be an extractor that outputs the preimage of a purported commitment c. We say that the hash function  $\mathsf{H}^E$  satisfies  $(q_E, \varepsilon, \delta)$ -extractable binding if for any stateful PPT adversary  $\mathcal{A}$  making at most  $q_E$  queries to E in the following experiment  $\mathbf{Exp}_{\mathsf{H}^E}^{\mathrm{ebind}}(\mathcal{A}, \mathsf{Ext})$ , we have  $\Pr[\mathbf{Exp}_{\mathsf{H}^E}^{\mathrm{ebind}}(\mathcal{A}, \mathsf{Ext}) = \mathtt{fail}] \leq \delta$  and  $\Pr[\mathbf{Exp}_{\mathsf{H}^E}^{\mathrm{ebind}}(\mathcal{A}, \mathsf{Ext}) = \mathtt{coll}] \leq \varepsilon$ .

Experiment  $\operatorname{Exp}_{H^{E}}^{\operatorname{ebind}}(\mathcal{A}, \mathsf{Ext})$ 

 (c, Q<sub>E</sub>) ← A<sup>E</sup>(commit) // Q<sub>E</sub> is the transcript of E-queries and answers of A<sup>E</sup>
 m<sup>\*</sup> ← Ext<sup>E</sup>(c, Q<sub>E</sub>)
 m ← A<sup>E</sup>(open)
 Output fail if Com(m) = c ∧ m<sup>\*</sup> = ⊥
 Output coll if Com(m) = c ∧ m<sup>\*</sup> ≠ ⊥ ∧ m ≠ m<sup>\*</sup>

Remark 1. To the best of our knowledge, all existing ZK schemes in the MPC-inthe-Head and VOLE-in-the-Head frameworks employ commitment with a similar structure as in Definition 2. E.g. in [9,50] the commitment consists of the concatenation of hashes the message XOR'ed with 0 and 1 but the hash function is required to be a random oracle. In [21] the construction is the same. Remark 2. We note that the extractable binding property in Definition 2 will only be used in the GGM-style vector commitment scheme to be introduced in Section 5, in which the message to hide is already uniformly random given the partial opening and we shall argue the vector commitment's hiding property using the CCR property of  $H_{CCR}$ . Therefore, we do not explicitly define the hiding property of the Com in Definition 2.

 $H_{CCR}$  is collision resistant. We will prove that our construction  $H_{CCR}^E(1^{\lambda}, 1^n, r)$  in Fig. 3 is extractable binding (Lemma 5). To this end, we need to establish its collision resistance, which is the focus of this subsection.

We distinguish between the cases of  $\lambda = n$  and  $n < \lambda \leq 2n$ . When  $\lambda = n$ , the construction  $\mathsf{H}_{\mathsf{CCR}}(m \oplus \mathsf{C}_2) \| \mathsf{H}_{\mathsf{CCR}}(m \oplus \mathsf{C}_3)$  is essentially a special case of the Hirose double-block-length compression function [42], which maps  $r \in \{0,1\}^{3n}$  to  $E_{\mathsf{right}_{2n}(r)}(\mathsf{left}_n(r)) \| E_{\mathsf{right}_{2n}(r)}(\mathsf{left}_n(r) \oplus [1]_n) \oplus \mathsf{left}_n(r) \oplus [1]_n$  using a block cipher  $E : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$ . A collision security bound of  $O(q_E^2/2^n)$ was given by Hirose [42, Theorem 4]. However, this is a bit buggy, since using  $q_E$  queries an "internal collision" of the form  $\mathsf{H}^E_{\mathsf{CCR}}(m \oplus \mathsf{C}_2) \| \mathsf{H}^E_{\mathsf{CCR}}(m \oplus \mathsf{C}_3) =$  $\mathsf{H}^E_{\mathsf{CCR}}(m \oplus \mathsf{C}_3) \| \mathsf{H}^E_{\mathsf{CCR}}(m \oplus \mathsf{C}_2)$  can be found with a success probability of  $O(q_E/2^n)$ . On the other hand, Berti et al. [14, Lemma 1] proved that the probability of finding a collision for the truncated construction  $\mathsf{H}^E_{\mathsf{CCR}}(m \oplus \mathsf{C}_2) \| \mathsf{left}_{n-1}(\mathsf{H}^E_{\mathsf{CCR}}(m \oplus \mathsf{C}_3))$  is bounded by  $6q_E/2^n$ . Clearly, this implies a collision security bound for the untruncated construction.

**Lemma 2.** Consider the case  $\lambda = n$ , and let  $C_2 = C_0$ ,  $C_3 = C_1$ . If we model  $E : \{0,1\}^{\lambda} \times \{0,1\}^n \to \{0,1\}^n$  as an ideal cipher, then the function Com defined in Definition 2 is  $(q_E, 6q_E/2^n)$ -collision-resistant.

When  $n < \lambda \leq 2n$ , the construction  $\mathsf{H}_{\mathsf{CCR}}(m \oplus \mathsf{C}_2) \| \mathsf{H}_{\mathsf{CCR}}(m \oplus \mathsf{C}_3)$  resembles the multi-block-length compression function of Abed et al. [1], which maps  $r \in \{0,1\}^{(w+1)n}$  to

$$E_{\mathsf{right}_{wn}(r)}(\mathsf{left}_n(r) \oplus [0]_n) \oplus \mathsf{left}_n(r) \oplus [0]_n \| \dots \| E_{\mathsf{right}_{wn}(r)}(\mathsf{left}_n(r) \oplus [w-1]_n) \oplus \mathsf{left}_n(r) \oplus [w-1]_n \oplus$$

using a long-key blockcipher  $E : \{0,1\}^{wn} \times \{0,1\}^n \to \{0,1\}^n$ . Though, the *E*-calls made by  $\mathsf{H}_{\mathsf{CCR}}(m \oplus \mathsf{C}_2) \| \mathsf{H}_{\mathsf{CCR}}(m \oplus \mathsf{C}_3)$  are using two distinct keys, and the outputs are truncated to  $\lambda$  bits. For rigorousness, we provide a lemma and its complete proof as follows.

**Lemma 3.** Consider the case  $n < \lambda \leq 2n$ , and let  $C_2 = C_0 ||[0]_{\lambda-n}, C_3 = C_1 ||[0]_{\lambda-n}$ . If we model  $E : \{0,1\}^{\lambda} \times \{0,1\}^n \to \{0,1\}^n$  as an ideal cipher, then the function Com defined in Definition 2 is  $(q_E, \varepsilon)$ -collision-resistant, where

$$\varepsilon \le \frac{20q_E}{2^{\lambda}} + \frac{640q_E^2}{2^{2\lambda}} \tag{6}$$

*Proof.* Our proof follows [1, Theorem 3] (which was motivated by [48]).

**Proof setup.** Since  $C_2 = C_0 ||[0]_{\lambda-n}$ ,  $C_3 = C_1 ||[0]_{\lambda-n}$ , it holds  $(x_L || x_R) \oplus C_2 = (x_L \oplus C_0 || x_R)$  and  $(x_L || x_R) \oplus C_3 = (x_L \oplus C_1 || x_R)$ . For convenience, define  $\Delta_k := [0]_{\lambda-n} ||(C_0 \oplus C_1)$  and  $\Delta_p := \sigma(C_0) \oplus \sigma(C_1)$ : note that  $(r_R || C_b) \oplus \Delta_k = (r_R || C_{\overline{b}})$  for any  $r_R \in \{0, 1\}^{\lambda-n}$  and  $b \in \{0, 1\}$ , and  $\sigma(x \oplus C_b) \oplus \Delta_p = \sigma(x \oplus C_{\overline{b}})$  for any  $x \in \{0, 1\}^n$  and  $b \in \{0, 1\}$  (since  $\sigma$  is linear).

Consider the interaction between the adversary  $\mathcal{A}$  and the ideal cipher E, and let  $\mathcal{Q}_E = \{(k_1, x_1, y_1), (k_2, x_2, y_2), ...\}$  be the query transcript of  $\mathcal{A}$  (which is similar to Section 4.2). Then, for any  $r_R \in \{0,1\}^{\lambda-n}$  and  $r_L \in \{0,1\}^n$ , the set

$$\left\{ \begin{array}{l} (r_{R} \| \mathsf{C}_{0}, \sigma(r_{L} \oplus \mathsf{C}_{0}), y_{0}^{0}), (r_{R} \| \mathsf{C}_{1}, \sigma(r_{L} \oplus \mathsf{C}_{0}), y_{0}^{1}), \\ (r_{R} \| \mathsf{C}_{0}, \sigma(r_{L} \oplus \mathsf{C}_{1}), y_{1}^{0}), (r_{R} \| \mathsf{C}_{1}, \sigma(r_{L} \oplus \mathsf{C}_{1}), y_{1}^{1}) \right\}$$
(7)

is called a *pile*. We stress that it is an *orderless* set. The involved keys  $r_R || C_0 \in \{0,1\}^{\lambda}$  and  $r_R || C_1 \in \{0,1\}^{\lambda}$  are called *the (two) keys of the pile*. For convenience, we further define

$$\mathcal{Q}_E[k] := \left\{ (x, y) : (k, x, y) \in \mathcal{Q}_E \right\}$$
(8)

for the set of query records in  $Q_E$  with the specific key k. The definition of such a structure stems from the 4 ideal cipher queries underlying the commitment evaluation  $\mathsf{Com}(r_L || r_R)$ . This constitutes our motivation of introducing this terminology.

Following [48], on top of the  $q_E$  queries the adversary  $\mathcal{A}$  wants to make, we give it several queries "for free", to ensure that all piles in the query history are "complete". The involved "free" queries are as follows.

- Normal forward query. If the adversary queries E(k, x) for some key  $k \in \{0, 1\}^{\lambda}$  and  $x \in \{0, 1\}^{n}$ , we also give it for free the answer to the 3 queries  $E(k, x \oplus \Delta_p), E(k \oplus \Delta_k, x), E(k \oplus \Delta_k, x \oplus \Delta_p).$
- Normal backward query. If the adversary queries  $E^{-1}(k, y) \to x$  for some key  $k \in \{0, 1\}^{\lambda}$  and  $y \in \{0, 1\}^n$ , we also give it for free the answer to the 3 queries  $E(k, x \oplus \Delta_p), E(k \oplus \Delta_k, x), E(k \oplus \Delta_k, x \oplus \Delta_p)$ .

Note that the adversary is not charged for the above "free" queries: they do not count towards the maximum of  $q_E$  queries that the adversary is allowed. However, these queries become part of the adversary's query history, just like other queries. We follow the standard assumption in ideal cipher proofs that "the adversary never makes a query that will result in a triple (k, x, y) which is already present in the query history". This means the adversary is not allowed, later, to remake the "free" queries "on its own". We remark that "free" queries are a common tool for analyzing the security of hash functions [42,4,48].

Due to the above "free" queries, records in the adversary's query history  $\mathcal{Q}$  can be grouped into "piles" as defined in Eq. (7). We now give further "free" queries to  $\mathcal{A}$ . In detail, after each pile has been completed (namely, after  $\mathcal{A}$  has received their responses), we check whether the two keys k and  $k \oplus \Delta_k$  of the pile is such that the (current) query history  $\mathcal{Q}$  contains exactly  $2^n/2$  query records with each of them, and give all remaining queries under them for free to  $\mathcal{A}$  if so.

We insert these  $2 \times (2^n/2) = 2^n$  free query records into Q. Since A is assumed never to make a query to which it knows the answer, it cannot make any more queries under this key after these free queries are inserted into Q.

When  $2 \times (2^n/2)$  such free query records are given to  $\mathcal{A}$ , we say that a super query occurs. This can be summed up as follows:

- Super query. When the query history  $\mathcal{Q}_E$  has  $|\mathcal{Q}_E[k]| = 2^n/2$  for some  $k \in \{0,1\}^{\lambda}$  (which also means  $|\mathcal{Q}_E[k \oplus \Delta_k]| = 2^n/2$ ), all the remaining queries of the form  $E(k, \cdot), E(k \oplus \Delta_k, \cdot)$  are given for free.

This means the adversary needs to take  $2^n/4$  queries to trigger a super query. By this, if  $\mathcal{A}$  makes  $q_E$  adversarial queries during the interaction then the following holds:

- There are exactly  $4q_E$  "normal queries" constituting  $q_E$  distinct piles;
- There are at most  $\frac{q_E}{2^n/4} \leq \frac{4q_E}{2^n}$  "super queries";
- For each  $k \in \{0, 1\}^{\lambda}$ , if the corresponding super query occurs then the number of piles in the super query is exactly  $\frac{2^n}{4}$ . This also means the number of piles in all super queries is at most  $\frac{4q_E}{2^n} \times \frac{2^n}{4} = q_E$  in total.

**Proof intuition: Decomposing the collision event.** We now serve some intuitions for the remaining arguments. For each  $r_R \in \{0,1\}^{\lambda-n}$ , all the  $2 \times 2^n$  inputs of  $E(r_R || C_0, \cdot), E(r_R || C_1, \cdot)$  constitute  $2^n/2$  disjoint piles (i.e., no input is shared among distinct piles). Meanwhile, each pile of Eq. (7) provides 2 possible commitment evaluations, i.e.,

$$\begin{aligned} \mathsf{Com}(r_L \oplus \mathsf{C}_0, r_R) &= \Big( \begin{array}{c} \sigma(r_L \oplus \mathsf{C}_0) \oplus E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_0)), \\ &\quad \mathsf{left}_{\lambda - n} \big( \sigma(r_L \oplus \mathsf{C}_0) \oplus E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_0)) \big), \\ \sigma(r_L \oplus \mathsf{C}_1) \oplus E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_1)), \\ &\quad \mathsf{left}_{\lambda - n} \big( \sigma(r_L \oplus \mathsf{C}_1) \oplus E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_1)) \big) \big), \end{aligned}$$

$$\begin{aligned} \mathsf{Com}(r_L \oplus \mathsf{C}_1, r_R) &= \Big( \begin{array}{c} \sigma(r_L \oplus \mathsf{C}_1) \oplus E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_1)), \\ &\quad \mathsf{left}_{\lambda - n} \big( \sigma(r_L \oplus \mathsf{C}_1) \oplus E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_1)), \\ &\quad \mathsf{left}_{\lambda - n} \big( \sigma(r_L \oplus \mathsf{C}_1) \oplus E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_1)) \big), \\ \sigma(r_L \oplus \mathsf{C}_0) \oplus E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_0)), \\ &\quad \mathsf{left}_{\lambda - n} \big( \sigma(r_L \oplus \mathsf{C}_0) \oplus E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_0)) \big) \big), \end{aligned}$$

Collisions may occur among distinct commitment evaluations from a single pile, e.g.,  $\operatorname{Com}(r_L \oplus \mathsf{C}_0, r_R) = \operatorname{Com}(r_L \oplus \mathsf{C}_1, r_R)$ , which will be referred as *internal collisions in piles*. Collisions can also occur among two distinct piles. With these in mind, we will decompose the collision event into *internal* and *external collisions*, and bound their probabilities in turn. **Decomposing and bounding the collision event.** As discussed above, we decompose the collision event as follows:

- (B-1)  $\mathcal{A}$  obtains a pile

$$\{ (r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_0), y_0^0), (r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_0), y_0^1), \\ (r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_1), y_1^0), (r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_1), y_1^1) \}$$

that has internal collisions, i.e.,  $\operatorname{Com}(r_L \oplus \mathsf{C}_0, r_R) = \operatorname{Com}(r_L \oplus \mathsf{C}_1, r_R)$ . - (B-2)  $\mathcal{A}$  obtains two distinct piles

$$\left\{ \begin{array}{l} (r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_0), y_0^0), (r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_0), y_0^1), \\ (r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_1), y_1^0), (r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_1), y_1^1) \end{array} \right\}$$

and

$$\begin{cases} (r'_{R} \| \mathsf{C}_{0}, \sigma(r'_{L} \oplus \mathsf{C}_{0}), y_{0}^{0'}), (r'_{R} \| \mathsf{C}_{1}, \sigma(r'_{L} \oplus \mathsf{C}_{0}), y_{0}^{1'}), \\ (r'_{R} \| \mathsf{C}_{0}, \sigma(r'_{L} \oplus \mathsf{C}_{1}), y_{1}^{0'}), (r'_{R} \| \mathsf{C}_{1}, \sigma(r'_{L} \oplus \mathsf{C}_{1}), y_{1}^{1'}) \end{cases}$$

that have *external collisions*, i.e., there exists  $b, b' \in \{0, 1\}$  such that  $\mathsf{Com}(r_L \oplus \mathsf{C}_b, r_R) = \mathsf{Com}(r'_L \oplus \mathsf{C}_{b'}, r'_R)$ .

We analyze them in turn.

**Event (B-1).** The collision  $\text{Com}(r_L \oplus C_0, r_R) = \text{Com}(r_L \oplus C_1, r_R)$  is equivalent with 2 equalities as follows:

1.  $\sigma(r_L \oplus \mathsf{C}_0) \oplus y_0^0 = \sigma(r_L \oplus \mathsf{C}_1) \oplus y_1^0$ , 2.  $\mathsf{left}_{\lambda-n} (\sigma(r_L \oplus \mathsf{C}_0) \oplus y_0^1) = \mathsf{left}_{\lambda-n} (\sigma(r_L \oplus \mathsf{C}_1) \oplus y_1^1)$ .

With the above in mind, we let (B-11) denote the event that the adversary obtains such a pile *after a normal query*, and (B-12) the event that the adversary obtains such a pile *after a super query*. We consider the two sub-events in turn.

<u>(B-11)</u>. First, consider the case  $\mathcal{A}$  making a forward query  $E(r_R || \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_0)) \to y_0^0$ . Then, at most  $2^n/2 - 2$  queries (counting free queries) have been previously answered with each of the two keys  $r_R || \mathsf{C}_0$  and  $r_R || \mathsf{C}_1$  since otherwise a super query for that key would have occurred. Thus the 2 answers  $E(r_R || \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_0)) \to y_0^1, E(r_R || \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_1)) \to y_1^1$  are distinct values uniformly picked from a set of size at least  $2^n/2 + 2$ . By these,

- The probability to have  $\sigma(r_L \oplus \mathsf{C}_0) \oplus y_0^0 = \sigma(r_L \oplus \mathsf{C}_1) \oplus y_1^0$  is at most  $1/(2^n/2 + 2) \le 2/2^n$ ;
- Conditioned on  $E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_0)) = y_0^1$ , it holds

$$\begin{aligned} &\Pr\left[\mathsf{left}_{\lambda-n}(\sigma(r_L \oplus \mathsf{C}_0) \oplus y_0^1) = \mathsf{left}_{\lambda-n}(\sigma(r_L \oplus \mathsf{C}_1) \oplus y_1^1)\right] \\ &= \sum_{t \in \{0,1\}^{2n-\lambda}} \Pr\left[y_1^1 = \left(\mathsf{left}_{\lambda-n}(\sigma(r_L \oplus \mathsf{C}_0) \oplus y_0^1 \oplus \sigma(r_L \oplus \mathsf{C}_1))\right) \| t\right] \\ &\leq \frac{2^{2n-\lambda}}{2^n/2+1} \leq \frac{2}{2^{\lambda-n}}. \end{aligned}$$

By these, the probability that the event (B-11) occurs w.r.t. the pile is at most  $\frac{2}{2^n} \times \frac{2}{2^{\lambda-n}} \leq 4/2^{\lambda}$ .

When  $\mathcal{A}$  makes a backward query  $E^{-1}(r_R \| \mathsf{C}_0, y_0^0) \to x = \sigma(r_L \oplus \mathsf{C}_0)$ , the analysis is similar by symmetry: the associated forward queries  $E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_0))$  and  $E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_1))$  have answers uniformly distributed in a set of size at least  $2^n/2$ . Therefore, the probability that (B-11) occurs w.r.t. the pile of these four queries is at most  $4/2^{\lambda}$ .

When  $\mathcal{A}$  queries  $E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_0)) \to y_0^1$  or  $E^{-1}(r_R \| \mathsf{C}_1, y_0^1) \to x$ , the associated forward queries  $E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_0))$  and  $E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_1))$  have answers uniformly distributed in a set of size at least  $2^n/2$ , and the same bound  $4/2^{\lambda}$  holds. Finally, since  $\mathcal{A}$  itself makes at most  $q_E$  queries in total, we have

$$\Pr\left[(\text{B-11})\right] \le \frac{4q_E}{2^{\lambda}}.\tag{9}$$

(B-12). Suppose a super query is about to occur on  $r_R \in \{0,1\}^{\lambda-n}$ , meaning that the value of  $E(r_R || \mathsf{C}_0, \cdot)$  is already known on exactly  $2^n/2$  points. Let  $\sigma(r_L \oplus \mathsf{C}_0), \sigma(r_L \oplus \mathsf{C}_1)$  be in the domain of the super query. (We say that a point  $x \in \{0,1\}^n$  is "in the domain of the super query" if  $E(r_R || \mathsf{C}_0, x)$  is not yet known, and will be queried as part of the super query; note that a point  $\sigma(r_L \oplus \mathsf{C}_0) \in \{0,1\}^n$  is in the domain of the super query if and only if  $\sigma(r_L \oplus \mathsf{C}_1)$  is also in the domain of the super query.) Once  $y_0^0 = E(r_R || \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_0))$  is known, the value of  $y_1^0 = E(r_R || \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_1))$  returned by the super query is uniformly picked from  $2^n/2 - 1$  possibilities. Therefore,

- The probability to have  $\sigma(r_L \oplus \mathsf{C}_0) \oplus y_0^0 = \sigma(r_L \oplus \mathsf{C}_1) \oplus y_1^0$  is at most  $\frac{1}{2^{n/2-1}} \leq \frac{4}{2n}$ ;

 $\frac{\frac{4}{2^n}}{2^n};$ - Conditioned on  $E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_0)) = y_0^1$ , it holds

$$\Pr\left[\mathsf{left}_{\lambda-n}(\sigma(r_L \oplus \mathsf{C}_0) \oplus y_0^1) = \mathsf{left}_{\lambda-n}(\sigma(r_L \oplus \mathsf{C}_1) \oplus y_1^1)\right]$$
$$= \sum_{t \in \{0,1\}^{2n-\lambda}} \Pr\left[y_1^1 = \left(\mathsf{left}_{\lambda-n}(\sigma(r_L \oplus \mathsf{C}_0) \oplus y_0^1 \oplus \sigma(r_L \oplus \mathsf{C}_1))\right) \|t\right]$$
$$\leq \frac{2^{2n-\lambda}}{2^n/2 - 1} \leq \frac{4}{2^{\lambda-n}}. \quad (\text{since } n \geq 2 \text{ implies } 2^n - 2 \leq 2^n/2)$$

Therefore, the probability that the super query returns values such that the pile  $\{(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_0), y_0^0), ..., (r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_1), y_1^1)\}$  has internal collisions is at most  $\frac{4}{2^n} \times \frac{4}{2^{\lambda-n}} \leq \frac{16}{2^{\lambda}}$ . The number of choices for this pile is  $2^n/4$  for the value  $r_R \in \{0, 1\}^{\lambda-n}$ . By a union bound, the probability that (B-12) occurs w.r.t. the super query for  $r_R \in \{0, 1\}^{\lambda-n}$  is bounded by

$$\frac{16}{2^{\lambda}} \times \frac{2^n}{4} \le \frac{4 \cdot 2^n}{2^{\lambda}}$$

As discussed, at most  $4q_E/2^n$  super queries can occur. Thus

$$\Pr\left[(\text{B-12})\right] \le \frac{4 \cdot 2^n}{2^\lambda} \times \frac{4q_E}{2^n} \le \frac{16q_E}{2^\lambda}.$$
(10)

Summary for (B-1). Summing over the two sub-events yields

$$\Pr[(B-1)] \le \Pr[(B-11)] + \Pr[(B-12)] \le \frac{4q_E}{2^{\lambda}} + \frac{16q_E}{2^{\lambda}} \le \frac{20q_E}{2^{\lambda}}.$$
 (11)

**Event (B-2).** The condition that the two piles are distinct means that  $(r'_L, r'_R) \notin \{(r_L, r_R), (r_L \oplus \Delta_p, r_R)\}$  and they consist of 8 distinct queries. Wlog assume that the pile  $\{(r_R || \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_0), y_0^0), ..., (r_R || \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_1), y_1^1)\}$  appears later in  $\mathcal{Q}_E$ . We let (B-21) denote the event that the adversary obtains such two piles after a normal query, and (B-22) the event that the adversary obtains such two piles after a super query. We consider the two sub-events in turn.

 $\begin{array}{l} (B-21). \text{ Consider the case where } \mathcal{A} \text{ makes a forward query } E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_0)) \to y_0^0 \text{ first. As argued before, the response } y_0^0 \text{ comes uniformly from at least } 2^n/2 + 2 \geq 2^n/2 \text{ possibilities. Furthermore, the 3 associated forward queries } E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_0)) \to y_0^1, E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_1)) \to y_1^0 \text{ and } E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_1)) \to y_1^1 \text{ have } y_1^1, y_1^0, y_1^1 \text{ uniformly distributed in at least } 2^n/2 \text{ possibilities. By these, for any pair of bits } (b, b') \in (\{0, 1\})^2 \text{ the probability to have } \mathsf{Com}(r_L \oplus \mathsf{C}_b, r_R) = \mathsf{Com}(r'_L \oplus \mathsf{C}_b', r'_R) \text{ is at most } \frac{16}{2^{2\lambda}}, \text{ since:} \end{array}$ 

- The probability to have  $\sigma(r_L \oplus \mathsf{C}_b) \oplus E(r_R || \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_b)) = \sigma(r'_L \oplus \mathsf{C}_{b'}) \oplus E(r'_R || \mathsf{C}_0, \sigma(r'_L \oplus \mathsf{C}_{b'}))$  and  $\sigma(r_L \oplus \mathsf{C}_{\overline{b}}) \oplus E(r_R || \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_{\overline{b}})) = \sigma(r'_L \oplus \mathsf{C}_{\overline{b'}}) \oplus E(r'_R || \mathsf{C}_0, \sigma(r'_L \oplus \mathsf{C}_{\overline{b'}}))$  is at most  $\frac{1}{2^{n/2}} \times \frac{1}{2^{n/2}} \leq 4/2^{2n}$ ;
- Conditioned on the ideal cipher outputs  $E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_b)), E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_{b'})), E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_{\overline{b'}}))$  and  $E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_{\overline{b'}}))$ , the probability to have  $\mathsf{left}_{\lambda-n}(\sigma(r_L \oplus \mathsf{C}_{\overline{b}}) \oplus E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_b))) = \mathsf{left}_{\lambda-n}(\sigma(r'_L \oplus \mathsf{C}_{b'}) \oplus E(r'_R \| \mathsf{C}_1, \sigma(r'_L \oplus \mathsf{C}_{b'}))))$  and  $\mathsf{left}_{\lambda-n}(\sigma(r_L \oplus \mathsf{C}_{\overline{b}}) \oplus E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_{\overline{b'}}))) = \mathsf{left}_{\lambda-n}(\sigma(r'_L \oplus \mathsf{C}_{\overline{b'}}))) = \mathsf{left}_{\lambda-n}(\sigma(r'_L \oplus \mathsf{C}_{\overline{b'}})))$  and  $\mathsf{left}_{\lambda-n}(\sigma(r'_L \oplus \mathsf{C}_{\overline{b'}}) \oplus E(r'_R \| \mathsf{C}_1, \sigma(r'_L \oplus \mathsf{C}_{\overline{b'}}))) = \mathsf{left}_{\lambda-n}(\sigma(r'_L \oplus \mathsf{C}_{\overline{b'}}) \oplus E(r'_R \| \mathsf{C}_1, \sigma(r'_L \oplus \mathsf{C}_{\overline{b'}}))))$  is at most  $(\frac{2}{2^{\lambda-n}})^2$ ;

When  $\mathcal{A}$  makes a backward query  $E^{-1}(r_R \| \mathsf{C}_0, y_0^0) \to x$ , the response  $x = \sigma(r_L \oplus \mathsf{C}_0)$  is uniform in a set of size at least  $2^n/2 + 2 \ge 2^n/2$ . Again, the 3 subsequent forward queries  $E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_0))$ ,  $E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_1))$  and  $E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_1))$  have answers uniformly distributed in at least  $2^n/2$  possibilities, and it yields the same bound  $16/2^{2\lambda}$ .

The number of choices for  $\{(r_R \| C_0, \sigma(r_L \oplus C_0), y_0^0), ..., (r_R \| C_1, \sigma(r_L \oplus C_1), y_1^1)\}$ (piles in normal queries) is at most  $q_E$ , while the number of choices for

$$\left\{ (r'_{R} \| \mathsf{C}_{0}, \sigma(r'_{L} \oplus \mathsf{C}_{0}), y^{0'}_{0}), ..., (r'_{R} \| \mathsf{C}_{1}, \sigma(r'_{L} \oplus \mathsf{C}_{1}), y^{1'}_{1}) \right\}$$

(piles in normal or super queries) is at most  $q_E + q_E = 2q_E$ . Moreover, the number of choices for the pair of bits  $(b, b') \in (\{0, 1\})^2$  is at most 4. Therefore,

$$\Pr[(B-21)] \le 4 \cdot q_E \cdot 2q_E \times \frac{16}{2^{2\lambda}} \le \frac{128q_E^2}{2^{2\lambda}}.$$
 (12)

(B-22). As argued before, for any  $r_L$  "in the domain of the super query",

- The two queries  $E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_0))$  and  $E(r_R \| \mathsf{C}_0, \sigma(r_L \oplus \mathsf{C}_1))$  have answers uniformly distributed in  $2^n/2$  and  $2^n/2 - 2 + 1 \ge 2^n/4$  possibilities respectively;
- The two queries  $E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_0))$  and  $E(r_R \| \mathsf{C}_1, \sigma(r_L \oplus \mathsf{C}_1))$  have answers uniformly distributed in  $2^n/2$  and  $2^n/2 - 2 + 1 \ge 2^n/4$  possibilities respectively.

Therefore, using the same argument as the previous case, for any pair of bits  $(b,b') \in (\{0,1\})^2$  the probability of having  $\operatorname{Com}(r_L \oplus \mathsf{C}_b, r_R) = \operatorname{Com}(r'_L \oplus \mathsf{C}_{b'}, r'_R)$  is at most  $(8/2^{\lambda})^2$ .

The number of choices for  $\{(r_R \| C_0, \sigma(r_L \oplus C_0), y_0^0), ..., (r_R \| C_1, \sigma(r_L \oplus C_1), y_1^1)\}$ (piles in super queries) is at most  $q_E$ , while the number of choices for

$$\left\{\left(r_{R}'\|\mathsf{C}_{0},\sigma(r_{L}'\oplus\mathsf{C}_{0}),y_{0}^{0}\right),...,\left(r_{R}'\|\mathsf{C}_{1},\sigma(r_{L}'\oplus\mathsf{C}_{1}),y_{1}^{1}\right)\right\}$$

(piles in normal or super queries) is at most  $q_E + q_E = 2q_E$ . Moreover, the number of choices for the pair of bits  $(b, b') \in (\{0, 1\})^2$  is at most 4. Therefore,

$$\Pr\left[(B-22)\right] \le 4 \cdot q_E \cdot 2q_E \times \frac{64}{2^{\lambda}} \le \frac{512q_E^2}{2^{\lambda}}.$$
(13)

Summary for (B-2). Summing over the two subevents yields

$$\Pr\left[(B-2)\right] \le \Pr\left[(B-21)\right] + \Pr\left[(B-22)\right] \le \frac{128q_E^2}{2^{2\lambda}} + \frac{512q_E^2}{2^{\lambda}} \le \frac{640q_E^2}{2^{2\lambda}}.$$
 (14)

**Summary.** Gathering Eq. (11) and (14), the probability that  $\mathcal{A}$  finds a collision from its query history is bounded by

$$\frac{20q_E}{2^{\lambda}} + \frac{640q_E^2}{2^{2\lambda}}$$
(15)

as claimed.

By Lemmas 2 and 3, the bound  $\frac{20q_E}{2^{\lambda}} + \frac{640q_E^2}{2^{2\lambda}}$  provides a universal collision security bound for all  $n \leq \lambda \leq 2n$ .

**Lemma 4.** Assume that  $n \leq \lambda \leq 2n$ , and let  $C_2 = C_0 ||[0]_{\lambda-n}$  and  $C_3 = C_1 ||[0]_{\lambda-n}$ . If we model  $E : \{0,1\}^{\lambda} \times \{0,1\}^n \to \{0,1\}^n$  as an ideal cipher, then the function Com defined in Definition 2 is  $(q_E, \varepsilon)$ -collision-resistant, where

$$\varepsilon \le \frac{20q_E}{2^{\lambda}} + \frac{640q_E^2}{2^{2\lambda}}.$$
(16)

 $H_{CCR}$  is extractable binding. With the help of the above results, we are now able to prove our extractable binding claim.

**Lemma 5.** The hash function  $H_{CCR}$  in Fig. 3 has  $(q_E, \frac{80q_E}{2^{\lambda}} + \frac{10240q_E^2}{2^{2\lambda}} + \frac{160q_E}{2^{2\lambda}}, \frac{80q_E}{2^{\lambda}} + \frac{10240q_E^2}{2^{2\lambda}})$  extractable binding property in Definition 2.

**Input.** The commitment  $c \in \{0,1\}^{2\lambda}$  and the ideal cipher transcript  $Q_E$ . **Output.** Message m such that Com(m) = c or  $\bot$ . **Oracle.** Ext has oracle access to E.

- 1. Ext initializes an empty set  $Q_P = \emptyset$ .
- 2. While  $\mathcal{Q}_E \neq \emptyset$ :
  - Pick the first element  $(k, x, y) \in \mathcal{Q}_E$ .
  - $\mathcal{Q}_P = \mathcal{Q}_P \cup \mathsf{Pile}(\mathcal{Q}_E, (k, x, y)).$
- 3. Now  $Q_P$  contains all possible piles from  $Q_E$ . Ext tries to find the preimage of c from  $Q_P$  as follows.
  - If  $\lambda = n$ , then for each entry in  $\mathcal{Q}_P$  denoted as

$$\{(k = \mathsf{C}_0, x_0, y_0), (k = \mathsf{C}_0, x_1 = x_0 \oplus \Delta_p, y_1)\}$$

Ext defines  $m_0 = \sigma^{-1}(x_0) \oplus \mathsf{C}_0$  and  $m_1 = \sigma^{-1}(x_1) \oplus \mathsf{C}_0$ . If  $c = x_0 \oplus y_0 || x_1 \oplus y_1$ , then Ext sets  $m^* = m_0$ . If  $c = x_1 \oplus y_1 || x_0 \oplus y_0$ , then Ext sets  $m^* = m_1$ . - If  $n < \lambda \leq 2n$ , then for each pile in  $\mathcal{Q}_P$  denoted as

$$\left\{ (k_0 = x_R \| \mathsf{C}_0, x_0, y_0^0), (k_1 = k_0 \oplus \Delta_k, x_0, y_0^1), (k_0, x_1 = x_0 \oplus \Delta_p, y_1^0), (k_1, x_1, y_1^1) \right\}$$

Ext defines  $m_0 = \sigma^{-1}(x_0) \oplus \mathsf{C}_0 ||_{x_R}$  and  $m_1 = \sigma^{-1}(x_1) \oplus \mathsf{C}_0 ||_{x_R}$ . If  $c = x_0 \oplus y_0^0 ||\mathsf{left}_{\lambda-n}(x_0 \oplus y_0^1) ||_{x_1} \oplus y_1^0 ||\mathsf{left}_{\lambda-n}(x_1 \oplus y_1^1)$ , then Ext sets  $m^* = m_0$ . If  $c = x_1 \oplus y_1^0 ||\mathsf{left}_{\lambda-n}(x_1 \oplus y_1^1) ||_{x_0} \oplus y_0^0 ||\mathsf{left}_{\lambda-n}(x_0 \oplus y_0^1)$ , then Ext sets  $m^* = m_1$ .

4. If  $m^*$  is undefined or multiply defined, then Ext outputs  $\perp$ . Otherwise, Ext outputs the unique  $m^*$  value.

**Procedure**  $\mathsf{Pile}(\mathcal{Q}_E, (k, x, y))$ :

Case 1: λ = n
If k ≠ C₀ then remove (k, x, y) from Q<sub>E</sub> and return Ø.
Define x' = x ⊕ Δ<sub>p</sub> and check if there is an entry (k, x', \*) ∈ Q<sub>E</sub>.
If so, denote the entry as (k, x', y<sub>1</sub>) and remove it from Q<sub>E</sub>.
Otherwise, query the oracle E to get y<sub>1</sub> = E(k, x').
Return {(k, x, y), (k, x', y<sub>1</sub>)}.
Case 2: n < λ ≤ 2n</li>
If right<sub>n</sub>(k) ∉ {C₀, C₁} then remove (k, x, y) from Q<sub>E</sub> and return Ø.
Define x' = x ⊕ Δ<sub>p</sub>, k' = k ⊕ Δ<sub>k</sub>. For the *i*-th element (k̃, x̃) ∈ ((k', x), (k, x'), (k', x')), check if there exists an entry (k̃, x̃, \*) ∈ Q<sub>E</sub>.
If so, denote the respective entries as (k̃, x̃, y<sub>i</sub>) and remove it from Q<sub>E</sub>.
Otherwise, query the oracle E to get y<sub>i</sub> = E(k̃, x̃).

Fig. 4: The extractor  $\mathsf{Ext}^E$  for our proof of Lemma 5.

<sup>3.</sup> Return  $\{(k, x, y), (k', x, y_1), (k, x', y_2), (k', x', y_3)\}$ .

*Proof.* Let c be the commitment output by  $\mathcal{A}$  and  $\mathcal{Q}_E$  be  $\mathcal{A}$ 's transcript of *E*-queries and answers. We follow the technique in the proof of Lemma 3 and define *piles*. Concretely, a pile is an orderless set with size  $\theta = 2$  in the case of  $\lambda = n$  and  $\theta = 4$  in the case of  $n < \lambda \leq 2n$ . We also define the constants  $\Delta_k = [0]_{\lambda-n} \| (\mathsf{C}_0 \oplus \mathsf{C}_1) \text{ and } \Delta_p = \sigma(\mathsf{C}_0) \oplus \sigma(\mathsf{C}_1).$ 

- If  $\lambda = n$ , then a pile is defined as

$$\{(k = \mathsf{C}_0, x_0, y_0), (k, x_0 \oplus \Delta_p, y_1)\}.$$

- If  $n < \lambda \leq 2n$ , then a pile is defined as

 $\{(k_0 = x_R \| \mathsf{C}_0, x_0, y_0^0), (k_1 = k_0 \oplus \Delta_k, x_0, y_0^1), (k_0, x_1 = x_0 \oplus \Delta_p, y_1^0), (k_1, x_1, y_1^1)\}.$ 

We define our extractor  $\mathsf{Ext}^{E}(c, \mathcal{Q}_{E})$  in Fig. 4. In the following we analyze the probabilities of  $\Pr[\mathsf{Exp}_{\mathsf{H}_{\mathsf{CR}}^{\mathsf{ebind}}}^{\mathsf{ebind}}(\mathcal{A},\mathsf{Ext}) = \mathtt{fail}] \leq \delta$  and  $\Pr[\mathsf{Exp}_{\mathsf{H}_{\mathsf{CR}}^{\mathsf{ebind}}}^{\mathsf{ebind}}(\mathcal{A},\mathsf{Ext}) = \mathtt{coll}] \leq \varepsilon$  in the extractable binding game (Definition 2). Let *m* be the purported Com pre-image returned by  $\mathcal{A}$ . Recall that by definition,

$$-\Pr[\mathbf{Exp}_{\mathsf{H}_{\mathsf{CCR}}^{\mathsf{ebind}}}^{\mathsf{ebind}}(\mathcal{A},\mathsf{Ext}) = \mathtt{fail}] = \Pr[\mathsf{Com}(m) = c \land m^* = \bot] \\ -\Pr[\mathbf{Exp}_{\mathsf{H}_{\mathsf{CCR}}^{\mathsf{ebind}}}^{\mathsf{ebind}}(\mathcal{A},\mathsf{Ext}) = \mathtt{coll}] = \Pr[\mathsf{Com}(m) = c \land m^* \neq \bot \land m \neq m^*]$$

Regarding the probability of outputting fail, note that Ext outputs  $m^* = \bot$ in two cases: first, it finds multiple candidate  $m^*$  values in  $\mathcal{Q}_E$  (denote this event by Multi); second, it does not find any candidate in  $\mathcal{Q}_E$  (denote this event by Null). It is easy to see

$$\begin{aligned} \Pr[\mathsf{Com}(m) &= c \land m^* = \bot] &= \Pr[\mathsf{Com}(m) = c \land \mathsf{Multi}] + \Pr[\mathsf{Com}(m) = c \land \mathsf{Null}] \\ &\leq \Pr[\mathsf{Multi}] + \Pr[\mathsf{Com}(m) = c \mid \mathsf{Null}] \end{aligned}$$

The event Multi indicates that the combination of  $\mathcal{A}$  and Ext succeeds in finding collisions on Com defined upon  $\mathsf{H}^{E}_{\mathsf{CCR}}$ . Since  $\mathcal{A}$  and Ext make at most  $4q_{E}$  queries to E in total, Lemma 4 indicates

$$\Pr\left[\mathsf{Multi}\right] \le rac{80q_E}{2^{\lambda}} + rac{10240q_E^2}{2^{2\lambda}}.$$

Regarding the event Com(m) = c, we show in Lemma 6 that

$$\Pr\bigl[\mathsf{Com}(m) = c \mid \mathsf{Null}\bigr] \le \frac{160q_E}{2^{2\lambda}}$$

By union bound, we conclude that  $\Pr[\mathbf{Exp}_{\mathsf{H}_{\mathsf{CCR}}^{ebind}}^{ebind}(\mathcal{A},\mathsf{Ext}) = \mathtt{fail}] \leq \delta = \frac{80q_E}{2^{\lambda}} + \frac{10240q_E^2}{2^{2\lambda}} + \frac{160q_E}{2^{2\lambda}}.$ 

The probability of outputting coll clearly does not exceed  $\Pr[\mathsf{Multi}]$ , yielding  $\Pr[\mathbf{Exp}_{\mathsf{H}_{\mathsf{CCR}}}^{\mathrm{ebind}}(\mathcal{A},\mathsf{Ext}) = \mathsf{coll}] \leq \varepsilon = \frac{80q_E}{2^{\lambda}} + \frac{10240q_E^2}{2^{2\lambda}}$ . These complete the proof.

**Lemma 6.** Consider the extractable binding game in Definition 2. If the adversary submitted a commitment c such that after the pile completion step in the proof of Lemma 5 no pre-image is found, then the probability that any adversary making at most  $q_E$  queries to E succeeds in providing a message  $m \in \{0,1\}^{\lambda}$  such that  $\operatorname{Com}(m) = c$  is bounded by  $\frac{160q_E}{2^{2\lambda}}$ .

*Proof.* Suppose that the adversary returns a valid message m such that Com(m) = c. We assume without loss of generality that the pile related to m appears in the adversary's transcript  $Q_E$  (since if not we can append that pile to  $Q_E$ ). We follow the same strategy as in the proof of Lemma 3 to grant the adversary free queries such that the entries in  $Q_E$  form piles and revealing all the input-output pairs associated with a key k if the number of queries related to k exceeds  $2^n/2$ . Consider the first pile

$$\left\{ (k_0 = x_R \| \mathsf{C}_0, x_0, y_0^0), (k_1 = k_0 \oplus \varDelta_k, x_0, y_0^1), (k_0, x_1 = x_0 \oplus \varDelta_p, y_1^0), (k_1, x_1, y_1^1) \right\}$$

and define  $m_0 = \sigma^{-1}(x_0) \oplus \mathsf{C}_0 || x_R$  and  $m_1 = \sigma^{-1}(x_1) \oplus \mathsf{C}_0 || x_R$ . By the definition of this pile, we have  $m = m_0$  or  $m = m_1$ . And since  $\mathsf{Com}(m) = c$ , we have the following two cases respectively.

$$c = x_0 \oplus y_0^0 \| \mathsf{left}_{\lambda-n}(x_0 \oplus y_0^1) \| x_1 \oplus y_1^0 \| \mathsf{left}_{\lambda-n}(x_1 \oplus y_1^1)$$
(17)

$$c = x_1 \oplus y_1^0 \| \mathsf{left}_{\lambda - n}(x_1 \oplus y_1^1) \| x_0 \oplus y_0^0 \| \mathsf{left}_{\lambda - n}(x_0 \oplus y_0^1) \tag{18}$$

Since this is the first pile being the pre-image of c, we have that  $y_0^0, y_0^1, y_1^0, y_1^1$ do not appear previously in  $Q_E$  and therefore are uniformly distributed over the non-queried positions. Consider the following two cases.

#### - Inversion After Free Query.

Since the values  $y_0^0, y_0^1, y_1^0, y_1^1$  are sampled from a set of size at least  $2^n/2 + 2$ , we have

$$\Pr[Eq. (17)] \le \frac{1}{2^n/2 + 2} \cdot \frac{2^{2n-\lambda}}{2^n/2 + 2} \cdot \frac{1}{2^n/2 + 1} \cdot \frac{2^{2n-\lambda}}{2^n/2 + 1} \le \frac{16}{2^{2\lambda}}$$

Similarly, we have  $\Pr[Eq. (18)] \leq \frac{16}{2^{2\lambda}}$ . By a union bound over all the  $q_E$  queries, we conclude that in this case  $\Pr[\mathsf{Com}(m) = c] \leq \frac{32q_E}{2^{2\lambda}}$ .

- Inversion After Super Query.

For each pile returned by the super query, we have

$$\Pr[Eq. (17)] \le \frac{1}{2^n/2} \cdot \frac{2^{2n-\lambda}}{2^n/2 - 1} \cdot \frac{1}{2^n/2} \cdot \frac{2^{2n-\lambda}}{2^n/2 - 1} \le \frac{64}{2^{2\lambda}}.$$

Notice that here we use the fact that since n > 1,  $2^n/2 - 1 \ge 2^n/4$ . We also have  $\Pr[Eq. (18)] \le \frac{64}{2^{2\lambda}}$ . By a union bound on all the  $2^n/4$  piles in a super query, the overall probability of inversion success in a super query is bounded by  $\frac{32 \cdot 2^n}{2^{2\lambda}}$ . Since the number of super query is bounded by  $\frac{4q_E}{2^n}$ , we have that in this case  $\Pr[\mathsf{Com}(m) = c] \le \frac{128q_E}{2^{2\lambda}}$ .

In summary, we have  $\Pr[\mathsf{Com}(m) = c] \leq \frac{32q_E}{2^{2\lambda}} + \frac{128q_E}{2^{2\lambda}} = \frac{160q_E}{2^{2\lambda}}$ .

Remark 3. We note that our construction of extractable CCR hash function in Fig. 3 generalizes the MMO-style CCR hash function in [39]. In particular, when  $\lambda = n = 128$ , our construction is identical to  $\mathsf{MMO}(x) := \pi(\sigma(x)) \oplus \sigma(x)$  if we view  $\pi(x) = E([0]_n, x)$ . Moreover, since the ideal cipher's key is fixed to be  $[0]_n$ , our proof in the ideal cipher model implies a proof in the random permutation model. We argue that this generalization demonstrates the utility of the new extractable CCR notation as one can derive a vector commitment scheme in the random permutation model by following our construction in Section 5.

### 5 The AES-based AVC Scheme

In this section, we present an AVC scheme from AES in the ideal cipher model. Our starting point is the correlated GGM tree proposed in [41]. Crucially, to support all-but-one opening, we need to derive the commitment information as well as the message for each of the leaf nodes in the tree. We formulate this process in Definition 3.

**Definition 3.** Let  $\mathsf{H}_{\mathsf{leaf}} : \{0,1\}^{\lambda} \to \{0,1\}^{\lambda} \times \{0,1\}^{2\lambda}$  be a function that maps the leaf node  $r \in \{0,1\}^{\lambda}$  into a message  $m \in \{0,1\}^{\lambda}$  and a commitment  $\mathsf{com} \in \{0,1\}^{2\lambda}$ .

In [10,9] the  $H_{leaf}$  function is instantiated by a random oracle, which due to its repeated invocation becomes a performance bottleneck.<sup>10</sup> On the other hand, in [21] the authors constructed  $H_{leaf}$  from a MMO-style CCR hash function [39] but its security analysis is tightly related to the structure of the CCR hash function. In this work, we try to design  $H_{leaf}$  and prove its security by a black box reduction to the extractable CCR security notion introduced in Section 4.

**Definition 4.** Let  $N, \lambda \in \mathbb{N}$  be the message length, vector dimension, and security parameter respectively. Let  $E, E^{-1}$  be the ideal cipher oracles. We define the syntax of the following four algorithms that constitute an AVC scheme.

- Setup $(1^{\lambda}, N) \rightarrow \text{crs:}$  Given security parameter  $\lambda$  and vector dimension N as input, output commitment key crs. If unnecessary, we can set crs =  $\bot$ .
- Commit(crs)  $\rightarrow$  (com, decom,  $(m_0, ..., m_{N-1})$ ): Given commitment key crs as input, output a commitment com, the opening information decom as well as N messages  $m_0, ..., m_{N-1} \in \{0, 1\}^{\lambda}$ .
- Open(crs, decom,  $\alpha$ )  $\rightarrow$  decom<sub> $\alpha$ </sub>: Given commitment key crs, opening information decom, and an opening index  $\alpha \in [0, N)$  output the decommitment for the index  $\alpha$ .
- Verify(crs, com,  $\alpha$ , decom<sub> $\alpha$ </sub>)  $\rightarrow$  { $(m_j)_{j\neq\alpha}$ }  $\cup$  { $\perp$ }: Given commitment key crs, commitment com, the index  $\alpha$ , and decommitment for  $\alpha$ , output either N-1 messages  $(m_j)_{j\neq\alpha}$  in case of accepting com or  $\perp$  in case of rejecting com.

<sup>&</sup>lt;sup>10</sup> We tested the reference implementation of FAEST on faest-128s instance and found that the hash function at the leaf level takes up 25.8% of the total signing time.

#### All four algorithms have oracle access to $E, E^{-1}$ .

We then propose a candidate AVC scheme in Fig. 5 by utilizing the correlated GGM construction in [41]. We define the required properties of the AVC scheme. In particular, we define the correctness property in Definition 5, the real-orrandom hiding property in Definition 6, and the extractable binding property in Definition 7.

#### Protocol cGGM-based AVC

Let  $\lambda, N = 2^d \in \mathbb{N}$  be the security parameter and vector dimension. Let  $\mathsf{H}_{\mathsf{CCR}}, \mathsf{H}_{\mathsf{leaf}}$ be two hash functions to be specified in the ideal cipher/random permutation model. Let  $\mathsf{H}_{\mathsf{RO}}$  be a random oracle. We define the following four algorithms.

- $\mathsf{Setup}(1^{\lambda}, N) \to \mathsf{crs}$ : Setup the hash functions.
- Commit(crs)  $\rightarrow$  (com, decom,  $(m_0, ..., m_{N-1})$ ):

  - 1. Sample a global key for this commitment  $\Gamma \leftarrow \{0,1\}^{\lambda}$ . 2. Sample the first layer nodes as  $r_0^1 \leftarrow \{0,1\}^{\lambda}$ ,  $r_1^1 = r_0^1 \oplus \Gamma$ . Let  $K_0^1 = r_0^1$ . 3. For  $i \in [2,d]$  we expand from the (i-1)-th layer to the *i*-th layer as follows. For  $j \in [0,2^{i-1})$ , compute  $r_{2j}^i = \mathsf{H}_{\mathsf{CCR}}(r_j^{i-1})$  and  $r_{2j+1}^i = r_{2j}^i \oplus r_j^{i-1}$ . Let  $K_{0}^{i} = \sum_{j \in 2^{i-1}} r_{2j}^{i}.$
  - 4. In the d-th layer, for  $i \in [0, 2^d)$ , we define  $(m_i, \operatorname{com}_i) = H_{\text{leaf}}(r_i^d)$  and  $com = H_{RO}(com_0, ..., com_{N-1})$ . Output  $(m_i)_{i \in [0,N)}$ , com, and decom =  $(\Gamma, \{K_0^i\}_{i \in [1,d]}, \{\operatorname{com}_i\}_{i \in [0,N)}).$
- Open(crs, decom,  $\alpha$ )  $\rightarrow$  decom<sub> $\alpha$ </sub>:
  - 1. Let  $\alpha_1, ..., \alpha_d$  be the binary decomposition of  $\alpha \in [0, 2^d)$  subject to  $\sum_{i \in [0,d)} \alpha_{i+1} \cdot 2^i = \alpha$ . Let  $\bar{\alpha}_i = \alpha_i \oplus 1$  be the negation of  $\alpha_i$ .
- 2. Output the opening information as  $\operatorname{decom}_{\alpha} = (\{K_0^i \oplus \bar{\alpha}_i \cdot \Gamma\}_{i \in [1,d]}, \operatorname{com}_{\alpha}).$ Verify(crs, com,  $\alpha$ , decom $_{\alpha}$ )  $\rightarrow \{(m_i)_{i \neq \alpha}\} \cup \{\bot\}$ :
  - 1. Parse the decommitment information as  $\operatorname{decom}_{\alpha} = (\{K_{\bar{\alpha}_i}^i\}_{i \in [1,d]}, \operatorname{com}_{\alpha}).$
- 2. Reconstruct the GGM tree using the information  $\{K_{\bar{\alpha}_i}^i\}_{i \in [1,d]}$ . Recover the last layer information as  $(m_j, \operatorname{com}_j) = \mathsf{H}_{\mathsf{leaf}}(r_j^d)$  for  $j \in [0, N), j \neq \alpha$ .
- 3. Re-compute the commitment as  $com' = H_{RO}(com_0, ..., com_{N-1})$  where  $com_{\alpha}$ is part of the decommitment information and the rest is generated in the previous step. Output  $\perp$  if  $\operatorname{com}' \neq \operatorname{com}$ . Otherwise, output  $(m_i)_{i\neq\alpha}$ .

Fig. 5: The candidate correlated GGM tree-based vector commitment scheme.

**Definition 5** (Correctness). An all-but-one vector commitment scheme AVC is (perfectly) correct if for all  $\lambda \in \mathbb{N}$  and  $N = poly(\lambda)$  the following condition holds.

 $\operatorname{crs} \leftarrow \operatorname{Setup}(1^{\lambda}, N), (\operatorname{com}, \operatorname{decom}, (m_0, ..., m_{N-1})) \leftarrow \operatorname{Commit}(\operatorname{crs}), \forall \alpha \in [0, N)$  $\operatorname{decom}_{\alpha} \leftarrow \operatorname{Open}(\operatorname{crs}, \operatorname{decom}, \alpha) : \operatorname{Verify}(\operatorname{crs}, \operatorname{com}, \alpha, \operatorname{decom}_{\alpha}) = (m_i)_{i \in [0,N), i \neq \alpha}.$ 

Algorithm H<sup>E</sup><sub>leaf</sub>(1<sup>λ</sup>, r)
// λ: security parameter;
// Note that |r| = λ.
1. Let H<sup>E</sup><sub>CCR</sub> be as defined in Fig. 3 and Com(r) = H<sup>E</sup><sub>CCR</sub>(r ⊕ C<sub>2</sub>) || H<sup>E</sup><sub>CCR</sub>(r ⊕ C<sub>3</sub>)
2. Let m = H<sup>E</sup><sub>CCR</sub>(r), com = Com(r)
3. Output (m, com).

Fig. 6: Theoretical model of  $\mathsf{H}^{E}_{\mathsf{leaf}}$ , where  $\mathsf{C}_2, \mathsf{C}_3$  are two  $\lambda$ -bit public constants.

**Definition 6 (Hiding).** Let AVC be an all-but-one vector commitment scheme following the syntax of Definition 4. The adaptive hiding experiment for AVC with  $N = 2^d = \text{poly}(\lambda)$  and stateful  $\mathcal{A}$  is defined as follows.

- 1. crs  $\leftarrow$  Setup $(1^{\lambda}, N), b^* \leftarrow \{0, 1\}$
- 2.  $(\text{com}, \text{decom}, (m_0^*, \dots, m_{N-1}^*)) \leftarrow \text{Commit}(\text{crs})$
- 3.  $\alpha \leftarrow \mathcal{A}(1^{\lambda}, \text{crs}, \text{com})$
- 4. decom<sub> $\alpha$ </sub>  $\leftarrow$  Open(crs, decom,  $\alpha$ )
- 5. Let  $m_i = m_i^*$  for  $i \in [0, N), i \neq \alpha$

6. For 
$$i = \alpha$$
, sample  $m_i \stackrel{\text{s}}{\leftarrow} \{0, 1\}^{\lambda}$  and set  $m_i = \begin{cases} m_i^* & \text{if } b^* = 0\\ m_i & \text{if } b^* = 1 \end{cases}$ 

7.  $b \leftarrow \mathcal{A}((m_i)_{i \in [0,N)}, \mathtt{decom}_{\alpha})$ 

8. Output 1 (success) if  $b = b^*$ , else 0 (failure).

In the selective hiding experiment,  $\mathcal{A}$  must choose  $\alpha$  prior to receiving com. The advantage AdvSelHide<sup>AVC</sup>( $\mathcal{A}$ ) (resp. AdvAdpHide<sup>AVC</sup>( $\mathcal{A}$ )) of an adversary  $\mathcal{A}$  is defined by  $|\Pr[\mathcal{A} \text{ wins}] - \frac{1}{2}|$  in the selective (resp. adaptive) hiding experiment. We say AVC is selectively (resp. adaptively) hiding if every PPT adversary  $\mathcal{A}$  has negligible advantage.

**Definition 7 (Binding).** Let AVC be an all-but-one vector commitment scheme following the syntax of Definition 4. Let  $\mathsf{Ext}(\mathsf{crs}, \mathsf{com}, Q_E) \to (m_i)_{i \in [0,N)}$  be an extraction function.

For any  $N = 2^d = \text{poly}(\lambda)$ , define the following extractable binding game for a stateful adversary  $\mathcal{A}$ .

- $\operatorname{crs} \leftarrow \operatorname{Setup}(1^{\lambda}, N)$
- $\texttt{com} \leftarrow \mathcal{A}(\texttt{crs})$
- $-(m_i^*)_{i\in[0,N)} = \mathsf{Ext}(\mathsf{crs},\mathsf{com},Q_E)$
- $(\texttt{decom}_{\alpha}, \alpha) \leftarrow \mathcal{A}()$
- Output 1 (success) if Verify(com,  $\alpha$ , decom<sub> $\alpha$ </sub>) =  $(m_i)_{i \in [0,N), i \neq \alpha}$  but  $m_i \neq m_i^*$  for some  $i \in [0, N), i \neq \alpha$ . Otherwise, output 0 (failure).

Let  $AdvEB^{AVC}(\mathcal{A}) = Pr[\mathcal{A} wins]$  be the advantage of the adversary in the above game. We say that AVC is extractable-binding if every PPT adversary  $\mathcal{A}$  has a negligible probability of winning the game.

#### 5.1 The Hiding Property of the AVC Scheme

In this section, we prove that the AVC scheme instantiated by the theoretical framework in Fig. 6 satisfies the selective hiding property in Definition 6.

**Lemma 7.** Let E be the ideal cipher oracle,  $\mathsf{H}^{E}_{\mathsf{CCR}}$  be a  $(q_E, q_C, \varepsilon_{\mathsf{CCR}})$ -circular correlation robust hash function, and  $\mathsf{H}_{\mathsf{leaf}}$  be defined as in Fig. 6. The AVC scheme in Fig. 5 with  $N = 2^d$  satisfies  $\mathsf{AdvSelHide}^{\mathsf{AVC}}(\mathcal{A}) \leq \varepsilon_{\mathsf{CCR}}$  for any adversary  $\mathcal{A}$  making  $q_E$  queries to E and  $q_C = d + 3$ .

*Proof.* Firstly, since we are proving the selective hiding property, we may assume the opening index  $\alpha = \alpha_1, ..., \alpha_d$  is fixed. Then, notice that

$$\begin{aligned} \Pr[b = b^*] &= \Pr[b = 1 \mid b^* = 1] \cdot \Pr[b^* = 1] + \Pr[b = 0 \mid b^* = 0] \cdot \Pr[b^* = 0] \\ &= \frac{1}{2} \Pr[b = 1 \mid b^* = 1] + \frac{1}{2} (1 - \Pr[b = 1 \mid b^* = 0]) \\ &= \frac{1}{2} + \frac{1}{2} (\Pr[b = 1 \mid b^* = 1] - \Pr[b = 1 \mid b^* = 0]) \end{aligned}$$

Therefore, we have

$$\Pr\left[\mathcal{A} \text{ wins}\right] - \frac{1}{2} = \frac{1}{2} \left|\Pr[b = 1 \mid b^* = 1] - \Pr[b = 1 \mid b^* = 0]\right|$$
(19)

We construct three hybrid experiments to upper-bound Eq. (19).

- **Hybrid**<sub>1</sub>. In this hybrid, the adversary is given the real commitment as well as the partial opening and the messages. This corresponds to the  $b^* = 0$  case.
- **Hybrid**<sub>2</sub>. In this hybrid, we generate  $com, (m_i)_{i \in [0,N)}, decom_{\alpha}$  using the following procedure.
  - First sample  $\operatorname{decom}_{\alpha}, \operatorname{com}_{\alpha}, m_{\alpha}$  uniformly at random.
  - Then compute  $\{m_i\}_{i\neq\alpha}$  and  $\{\operatorname{com}_i\}_{i\neq\alpha}$  using the Verify algorithm.
  - Finally, compute  $com = H_{RO}(com_0, ..., com_{N-1})$ .
- Hybrid<sub>3</sub>. In this hybrid, we generate com,  $(m_i)_{i \neq \alpha}$ , decom<sub> $\alpha$ </sub> using the algorithm Commit but samples  $m_{\alpha}$  uniformly at random. Notice that this corresponds to the  $b^* = 1$  case.

We derive the final result via two claims.

*Claim.* The difference between **Hybrid**<sub>1</sub> and **Hybrid**<sub>2</sub> is bounded by  $\varepsilon_{CCR}$ .

*Proof.* Let  $\mathcal{A}$  be an adversary that distinguishes **Hybrid**<sub>1</sub> and **Hybrid**<sub>2</sub>. We construct a distinguisher  $\mathcal{B}$  of the CCR game with the same advantage. In this way, we can prove that the difference between **Hybrid**<sub>1</sub> and **Hybrid**<sub>2</sub> is upperbounded by  $\varepsilon_{CCR}$ .

The algorithm  $\mathcal{B}$  proceeds as follows.

1. First samples  $r_{\bar{\alpha}_1}^1 \leftarrow \{0,1\}^{\lambda}$  and sets  $s_1 = r_{\bar{\alpha}_1}^1, K^1 = r_{\bar{\alpha}_1}^1$ 

- 2. For  $i \in [2, d]$ , let  $j^* = \alpha_1 \| ... \| \alpha_{i-1}$  computes  $r_{j^* + \bar{\alpha}_i}^i = \mathcal{O}^{\mathsf{CCR}}(s_{i-1}, \bar{\alpha}_i) \oplus \bar{\alpha}_i s_{i-1}$ . For all  $j \in [0, 2^{i-1})$  s.t.  $j \neq j^*$ , compute  $r_{2j}^i = \mathsf{H}_{\mathsf{CCR}}(r_j^i)$  and  $r_{2j+1}^i = \mathsf{H}_{\mathsf{CCR}}(r_j^i) \oplus r_j^i$ . Then computes  $K^i = \sum_{j \in [0, 2^{i-1})} r_{2j+\bar{\alpha}_i}^i$ . Finally, update  $s_i = s_{i-1} \oplus r_{j^* + \bar{\alpha}_i}^i$
- 3. For  $j \in [0, 2^d)$  s.t.  $j \neq \alpha$ , computes  $m_j = \mathsf{H}_{\mathsf{CCR}}(r_j^d)$  and  $\mathsf{com}_j = \mathsf{H}_{\mathsf{CCR}}(r_j^d \oplus \mathsf{C}_2) \|\mathsf{H}_{\mathsf{CCR}}(r_j^d \oplus \mathsf{C}_3)$ .
- 4. Computes  $m_{\alpha} = \mathcal{O}^{\mathsf{CCR}}(s_d, 0)$  and  $\mathsf{com}_{\alpha} = \mathcal{O}^{\mathsf{CCR}}(s^d \oplus \mathsf{C}_2, 0) \| \mathcal{O}^{\mathsf{CCR}}(s^d \oplus \mathsf{C}_3, 0)$
- 5. Computes  $\operatorname{com} = \operatorname{H}_{\operatorname{RO}}(\operatorname{com}_0, ..., \operatorname{com}_{N-1}).$
- 6. Output com,  $\{m_i\}_{i \in [0, 2^d)}$ , and  $\operatorname{decom}_{\alpha} = \{K^i\}_{i \in [1, d]}$

Notice that if  $\mathcal{O}^{\mathsf{CCR}}$  returns the real correlated randomness then the view of  $\mathcal{A}$  is identical to **Hybrid**<sub>1</sub>, whereas if  $\mathcal{O}^{\mathsf{CCR}}$  returns uniformly random values then the view of  $\mathcal{A}$  is identical to **Hybrid**<sub>2</sub>. Therefore, the advantage of  $\mathcal{A}$  is upper-bounded by  $\varepsilon_{\mathsf{CCR}}$ .

*Claim.* The difference between **Hybrid<sub>2</sub>** and **Hybrid<sub>3</sub>** is bounded by  $\varepsilon_{CCR}$ .

*Proof.* The argument is almost identical to the previous claim, except that we always sample  $m_{\alpha}$  uniformly at random in the two hybrids. Therefore, the advantage of the adversary in distinguishing the two hybrids is also bounded by  $\varepsilon_{CCR}$ .

Combining the above two claims, we conclude that in Eq. (19) we have

$$|\Pr[b=1 \mid b^*=1] - \Pr[b=1 \mid b^*=0]| \le 2\varepsilon_{\mathsf{CCR}}$$
(20)

and therefore

$$\mathsf{AdvSelHide}^{\mathsf{AVC}}(\mathcal{A}) = \left| \Pr\left[\mathcal{A} \text{ wins}\right] - \frac{1}{2} \right| \le \varepsilon_{\mathsf{CCR}}$$
(21)

#### 5.2 The Extractable Binding Property of H<sub>leaf</sub>

**Lemma 8.** Let  $\mathsf{H}_{\mathsf{CCR}}$  be a  $(q_E, \varepsilon_{\mathsf{EB}}, \delta_{\mathsf{EB}})$ -extractable binding CCR function in the ideal cipher model and  $\mathsf{H}_{\mathsf{RO}}$  be a random oracle. By instantiating  $\mathsf{H}_{\mathsf{leaf}}$  with the construction in Fig. 6, the vector commitment scheme in Fig. 5 with  $N = 2^d$  satisfies  $\mathsf{AdvEB}^{\mathsf{AVC}}(\mathcal{A}) \leq \frac{1+q_H(q_H-1)/2}{2^{2\lambda}} + N \cdot (\delta_{\mathsf{EB}} + \varepsilon_{\mathsf{EB}})$  for any adversary  $\mathcal{A}$  making  $q_E$  queries to the ideal cipher oracle and  $q_H$  queries to the random oracle.

*Proof.* We present the extractor first. The extractor first goes through the random oracle's transcript and looks for the pre-image of com. If none can be found or the pre-image is not unique, then it outputs  $\perp$ .

Assuming that the pre-image  $\operatorname{com}_0, ..., \operatorname{com}_{N-1}$  is unique then the extractor applies  $\operatorname{Ext}(\operatorname{com}_i, \mathcal{Q}_E)$  for every  $i \in [0, N)$  to get  $m_i$  (notice that  $m_i$  might be  $\bot$ ). Finally, the extractor outputs  $\{m_i\}_{i \in [0,N)}$ .

Now we bound the probability  $\mathsf{AdvEB}^{\mathsf{AVC}}(\mathcal{A})$ . Since  $\mathsf{H}_{\mathsf{RO}}$  is a random oracle, the probability of the extractor outputting  $\bot$  due to finding a collision is bounded by  $\frac{1}{2^{2\lambda}} + \frac{2}{2^{2\lambda}} + \cdots + \frac{q_H - 1}{2^{2\lambda}} = \frac{q_H(q_H - 1)}{2^{22\lambda}}$ . The probability of not being able to find a pre-image is bounded by  $\frac{1}{2^{2\lambda}}$ . Therefore, the extraction of com succeeds except with probability  $\frac{1+q_H(q_H-1)/2}{2^{2\lambda}}$ .

Furthermore, the event of extraction failure for  $i \neq \alpha$  would imply extraction failure for  $\mathsf{Ext}(\mathsf{com}_i, \mathcal{Q}_E)$ , which is bounded by  $\delta_{\mathsf{EB}} + \varepsilon_{\mathsf{EB}}$ . Therefore, by taking a union-bound on all the leaf nodes, we conclude that

$$\mathsf{AdvEB}^{\mathsf{AVC}}(\mathcal{A}) \le \frac{1 + q_H(q_H - 1)/2}{2^{2\lambda}} + N \cdot (\delta_{\mathsf{EB}} + \varepsilon_{\mathsf{EB}}).$$
(22)

### 6 The AES-based AHC Scheme

In this section, we demonstrate that using the AES-based CCR hash function, we can construct an AHC scheme that is extensively utilized in VOLE-ZK [12,63,29,62]. Let  $\mathbb{F}$  be a subfield and  $\mathbb{K}$  be its extension field. In such a commitment correlation, one party (Alice) gets two random vectors  $\boldsymbol{u} \in \mathbb{F}^N, \boldsymbol{w} \in \mathbb{K}^N$ while another party (Bob) gets a random vector and a random scalar  $\boldsymbol{v} \in \mathbb{K}^N, \Gamma \in \mathbb{K}$  such that  $\boldsymbol{v} = \boldsymbol{w} + \boldsymbol{u} \cdot \boldsymbol{\Delta}$ . Intuitively, Bob "authenticates" the vector  $\boldsymbol{u}$  since we can require that whenever Alice reveals  $\boldsymbol{u}$  (or some of its components) she would also have to provide the corresponding  $\boldsymbol{w}$  values. To change  $\boldsymbol{u}$ , Alice has to guess the exact value of  $f\Gamma$ , and her success probability equals to the inverse of the size of  $\mathbb{K}$ . Additionally, since the authentication global key  $\Gamma$  can be fixed, the relation  $\boldsymbol{v} = \boldsymbol{w} + \boldsymbol{u} \cdot \boldsymbol{\Delta}$  still holds under linear operations on  $\boldsymbol{u}$ . Therefore, the commitment is said to be additively homomorphic and commitment of random values can be changed to that of arbitrary values. We note that this correlation is also referred to as subfield Vector Oblivious Linear Evaluation (sVOLE) in the literature.

The state-of-the-art method for generating aforementioned additively homomorphic commitment relies on the Learning Parity with Noise (LPN) assumption [63,60,57,17]. Let  $m, N \in \mathbb{N}$  be two parameters and m < N. The LPN assumption [52] states that for a public random compressing matrix  $\mathbf{H} \in \mathbb{F}^{m \times N}$ and a secret sparse vector  $\mathbf{e} \in \mathbb{F}^N$ , the product  $\mathbf{y} = \mathbf{H} \cdot \mathbf{e}$  is pseudorandom. Therefore, an efficient method to generate additively homomorphic commitment for a random vector  $\mathbf{u}$  is to first generate many commitments for the *single point* vector  $\mathbf{u}'$ , and then using the additive homomorphic property of the commitment, we can add multiple single point vectors to get a sparse vector and then multiply it by the LPN compression matrix  $\mathbf{H}$  to get commitment of the pseudorandom vector  $\mathbf{u}$ . In summary, under the LPN assumption, the task of generating AHC correlations of random vectors.

The most efficient protocol to generate such correlations to date is the pseudorandom correlated GGM tree method of Guo et al. [41]. This protocol utilizes

circular correlation robust (CCR) hash function as a black box and therefore our AES-based CCR hash function can be integrated into this protocol seamlessly. Moreover, as previous AES-based instantiations utilizes the construction in [39], which is limited to 128-bit security, our construction pushes the AHC generation method of [41] to the high security domain.

We recall the security of the single point subfield VOLE protocol  $\Pi_{spsVOLE}$  with pseudorandom correlated GGM tree of [41] in Theorem 1. We include the descriptions of the ideal functionality generating the sVOLE and single point sVOLE correlations to Appendix A for completeness.

**Theorem 1 (Theorem 3 in [41]).** Given CCR function  $H : \mathbb{F}_{2^{\lambda}} \to \mathbb{F}_{2^{\lambda}}$ , function  $Convert_{\mathbb{K}} : \mathbb{F}_{2^{\lambda}} \to \mathbb{K}$ , and the pseudorandom correlated GGM tree (Fig. 7) for field  $\mathbb{K}$ , keyed hash function  $H_S(x) := H(S \oplus x)$  with key  $S \leftarrow \mathbb{F}_{2^{\lambda}}$ , and function  $Convert_{\mathbb{K}}$ , protocol  $\Pi_{spsVOLE-pcGGM}$  (Fig. 8) UC-realizes functionality  $\mathcal{F}_{spsVOLE}$  (Fig. 10) against any semi-honest adversary in the ( $\mathcal{F}_{COT}$ ,  $\mathcal{F}_{sVOLE}$ )hybrid model.

Remark 4. We note that Theorem 1 requires the CCR security of H and therefore by instantiating H with the construction in Section 4, we can acquire a secure  $\Pi_{spsVOLE}$  for all parameters choices  $\lambda \in \{128, 192, 256\}$  from AES.

### 7 Performance Evaluation

In this section, we test the performance of our proposed AES-based CCR hash function. We want to resolve two research questions via experiments in this section, namely:

- RQ1: In terms of performance, in the parameter ranges where the security parameter exceeds the block size, how does our AES-based CCR hash function compare with previous MMO-style CCR hash construction [39] instantiated with the non-standardized block ciphers?
- RQ2: To what extent can the commitment scheme required at the leaf nodes of the all-but-one vector commitment (AVC) scheme be optimized by using the new AES-based CCR hash function instead of standardized cryptographic hash functions?

For the first question, we instantiated the MMO-style CCR hash function [39] with the Rijndael encryption scheme [25] where the block size equals the security parameter and compared our AES-based CCR hash construction with it. We chose this particular configuration as the comparison baseline for multiple reasons. Firstly, the 128-bit block size version of Rijndael is chosen as the Advanced Encryption Standard and has withstood decades of cryptanalysis efforts. This brings confidence in the other configurations of the Rijndael encryption scheme. Secondly, the Rijndael-based CCR hash function appeared previously in the literature [21]. Finally, due to the structural similarity, both the Rijndael cipher and AES can utilize the AES-NI instruction sets, therefore offering a

**Parameters:** Tree depth  $d \in \mathbb{N}$ . Field  $\mathbb{K}$ . Keyed hash function  $\mathsf{H}_S : \mathbb{F}_{2^{\lambda}} \to \mathbb{F}_{2^{\lambda}}$ . Function  $\mathsf{Convert}_{\mathbb{K}} : \mathbb{F}_{2^{\lambda}} \to \mathbb{K}$ . pcGGM.FullEval( $\Delta, k$ ): Given  $(\Delta, k) \in \mathbb{F}_{2^{\lambda}}^2$  $\begin{array}{ll} 1: \ s_1^0 := k \in \mathbb{F}_{2^{\lambda}}, s_1^1 := \Delta \oplus k \in \mathbb{F}_{2^{\lambda}} \\ 2: \ \mathbf{for} \ i \in [2,d), j \in [0,2^{i-1}) \ \mathbf{do} \\ 3: \ \ s_i^{2j} := \mathsf{H}_S \left( s_{i-1}^j \right) \in \mathbb{F}_{2^{\lambda}}, s_i^{2j+1} := s_{i-1}^j \oplus s_i^{2j} \in \mathbb{F}_{2^{\lambda}}. \end{array}$ 4: for  $j \in [0, 2^{d-1}), \sigma \in \{0, 1\}$  do 5:  $s_n^{2j+\sigma} := \operatorname{Convert}_{\mathbb{K}} \left( \mathsf{H}_S\left(s_{d-1}^j \oplus \sigma\right) \right) \in \mathbb{K}$ 6:  $\boldsymbol{v} := \left(s_d^0, \dots, s_d^{2^d-1}\right) \in \mathbb{K}^{2^d}$ 7: for  $i \in [1, d)$  do  $K_0^i := \bigoplus_{j \in [0, 2^{i-1})} s_i^{2j} \in \mathbb{F}_{2^{\lambda}}$ 9:  $(K_0^d, K_1^d) := \left(\sum_{j \in [0, 2^{d-1})} s_d^{2j}, \sum_{j \in [0, 2^{d-1})} s_d^{2j+1}\right) \in \mathbb{K}^2$ 10: return  $\left( \boldsymbol{v}, \left\{ K_{0}^{i} \right\}_{i \in [1, d-1]}, \left( K_{0}^{d}, K_{1}^{d} \right) \right)$  $\mathsf{pcGGM}.\mathsf{PuncFullEval}(\alpha, \left\{K_{\bar{\alpha}_i}^i\right\}_{i \in [1,d]}, \gamma): \mathrm{Given}\left(\alpha, \left\{K_{\bar{\alpha}_i}^i\right\}_i, \gamma\right) \in \{0,1\}^d \times \mathbb{K}^d \times \mathbb{K}, \mathsf{pcGGM}.\mathsf{PuncFullEval}(\alpha, \left\{K_{\bar{\alpha}_i}^i\right\}_i, \gamma): \mathsf{Given}\left(\alpha, \left\{K_{\bar{\alpha}_i}^i\right\}_i, \gamma\right) \in \{0,1\}^d \times \mathbb{K}^d \times \mathbb{K}, \mathsf{pcGGM}.\mathsf{PuncFullEval}(\alpha, \left\{K_{\bar{\alpha}_i}^i\right\}_i, \gamma): \mathsf{Given}\left(\alpha, \left\{K_{\bar{\alpha}_i}^i\right\}_i, \gamma\right) \in \{0,1\}^d \times \mathbb{K}^d \times \mathbb{K}, \mathsf{pcGGM}.\mathsf{PuncFullEval}(\alpha, \left\{K_{\bar{\alpha}_i}^i\right\}_i, \gamma): \mathsf{Given}\left(\alpha, \left\{K_{\bar{\alpha}_i}^i\right\}_i, \gamma\right) \in \{0,1\}^d \times \mathbb{K}^d \times \mathbb{K}, \mathsf{pcGGM}.\mathsf{PuncFullEval}(\alpha, \left\{K_{\bar{\alpha}_i}^i\right\}_i, \gamma): \mathsf{Given}\left(\alpha, \left\{K_{\bar{\alpha}_i}^i\right\}_i, \gamma\right) \in \{0,1\}^d \times \mathbb{K}^d \times \mathbb{K}, \mathsf{pcGGM}.\mathsf{PuncFullEval}(\alpha, \left\{K_{\bar{\alpha}_i}^i\right\}_i, \gamma): \mathsf{Given}\left(\alpha, \left\{K_{\bar{\alpha}_i}^i\right\}_i, \gamma\right) \in \{0,1\}^d \times \mathbb{K}^d \times \mathbb{K}, \mathsf{pcGM}.\mathsf{pcGM}.\mathsf{pcGM}.\mathsf{pcGM}.\mathsf{pcGM}$ 1:  $s_1^{\bar{\alpha}_1} := K_{\bar{\alpha}_1}^1 \in \mathbb{F}_{2^{\lambda}}$ 2: for  $i \in [2, d)$  do 
$$\begin{split} \mathbf{f} & \mathbf{f} \in [2, a) \text{ do} \\ & \mathbf{for} \ j \in [0, 2^{i-1}), j \neq \alpha_1 \dots \alpha_{i-1} \text{ do} \\ & s_i^{2j} := \mathsf{H}_S \left( s_{i-1}^j \right) \in \mathbb{F}_{2^{\lambda}}, s_i^{2j+1} := s_{i-1}^j \oplus s_i^{2j} \in \mathbb{F}_{2^{\lambda}} \\ & s_i^{\alpha_1 \dots \alpha_{i-1} \bar{\alpha}_i} := K_{\bar{\alpha}_i}^i \oplus \left( \oplus_{j \in [0, 2^{i-1}), j \neq \alpha_1 \dots \alpha_{i-1}} s_i^{2j+\bar{\alpha}_i} \right) \in \mathbb{F}_{2^{\lambda}}. \end{split}$$
3: 4: 5: 6: for  $j \in [0, 2^{d-1}), j \neq \alpha_1 \dots \alpha_{d-1}, \sigma \in \{0, 1\}$  do 7:  $s_d^{2j+\sigma} := \text{Convert}_{\mathbb{K}} \left( \mathbb{H}_S \left( s_{d-1}^j \oplus \sigma \right) \right) \in \mathbb{K}$ 8:  $s_d^{\alpha_1 \dots \alpha_{d-1}\bar{\alpha}_d} := K_{\bar{\alpha}_d}^d - \sum_{j \in [0, 2^{d-1}), j \neq \alpha_1 \dots \alpha_{d-1}} s_d^{2j+\bar{\alpha}_d} \in \mathbb{K}$ 9:  $s_d^{\alpha} := \gamma - \sum_{j \in [0, 2^d), j \neq \alpha} s_d^j \in \mathbb{K}, \boldsymbol{w} := \left( s_d^0, \dots, s_d^{2^d-1} \right) \in \mathbb{K}^{2^d}$ 10: return w

Fig. 7: The pseudorandom correlated GGM tree algorithms.

fairer comparison in terms of performance. As for the second question, we chose to use the SHA3 algorithm as it is the latest standard of NIST and is utilized by some ZK-based post-quantum signatures [6,2].

We implemented our AES-based CCR hash function and commitment scheme based on the emp-toolkit [59]. We used the vectorized implementation of the AES algorithms in the AES-NI whitepaper [37]. As for Rijndael, we chose to use the vectorized implementation from FAEST [9]. We used the SHA3 implementation available at the original emp-toolkit project, which builds upon the widely used OpenSSL library [58]. Our code is available at the anonymized repository https: //anonymous.4open.science/r/aes\_ccr\_hash-6B7B. We conducted our experiments on a machine running Ubuntu 24.04 OS with i7-7700 CPU @ 3.60GHz and 16GB RAM. The compiler is GCC 13.2.0 with the '-O2' flag enabled. **Parameters:** Field  $\mathbb{F}$  and its extension field  $\mathbb{K}$ .

Init. This procedure is executed only once.

- 1.  $P_A$  and  $P_B$  send (init) to  $\mathcal{F}_{COT}$ , which returns  $\Delta \in \mathbb{F}_{2^{\lambda}}$  to  $P_A$ .
- 2.  $P_A$  and  $P_B$  send (init) to  $\mathcal{F}_{sVOLE}$ , which returns  $\Gamma \in \mathbb{K}$  to  $P_A$ .  $P_A$  outputs  $\Gamma$ .

**Extend.**  $\mathsf{P}_{\mathsf{A}}$  and  $\mathsf{P}_{\mathsf{B}}$  input  $N = 2^d$  and use pcGGM (c.f. Fig. 7) for  $d, \mathbb{K}$ , keyed hash function  $\mathsf{H}_S : \mathbb{F}_{2^{\lambda}} \to \mathbb{F}_{2^{\lambda}}$ , and function  $\mathsf{Convert}_{\mathbb{K}} : \mathbb{F}_{2^{\lambda}} \to \mathbb{K}$ .

- 1.  $\mathsf{P}_{\mathsf{A}}$  and  $\mathsf{P}_{\mathsf{B}}$  send (extend, d) to  $\mathcal{F}_{\mathsf{COT}}$ , which returns  $(\mathsf{K}[r_1], \ldots, \mathsf{K}[r_n]) \in \mathbb{F}_{2^{\lambda}}^d$  to  $\mathsf{P}_{\mathsf{A}}$  and  $((r_1, \ldots, r_d), (\mathsf{M}[r_1], \ldots, \mathsf{M}[r_d])) \in \mathbb{F}_2^d \times \mathbb{F}_{2^{\lambda}}^d$  to  $\mathsf{P}_{\mathsf{B}}$  such that  $\mathsf{M}[r_i] = \mathsf{K}[r_i] \oplus r_i \cdot \Delta$  for  $i \in [1, d]$ .
- 2.  $\mathsf{P}_{\mathsf{A}}$  and  $\mathsf{P}_{\mathsf{B}}$  send (extend, 1) to  $\mathcal{F}_{\mathsf{sVOLE}}$ , which returns  $\mathsf{K}[s] \in \mathbb{K}$  to  $\mathsf{P}_{\mathsf{A}}$  and  $(s,\mathsf{M}[s]) \in \mathbb{F} \times \mathbb{K}$  to  $\mathsf{P}_{\mathsf{B}}$  such that  $\mathsf{M}[s] = \mathsf{K}[s] + s \cdot \Gamma$ .
- 3.  $\mathsf{P}_{\mathsf{B}}$  samples  $\beta \leftarrow \mathbb{F}^*$ , sets  $\mathsf{M}[\beta] := \mathsf{M}[s]$ , and sends  $\delta := s \beta \in \mathbb{F}$  to  $\mathsf{P}_{\mathsf{A}}$ .  $\mathsf{P}_{\mathsf{A}}$  sets  $\mathsf{K}[\beta] := \mathsf{K}[s] + \delta \cdot \Gamma$  such that  $\mathsf{M}[\beta] = \mathsf{K}[\beta] + \beta \cdot \Gamma$ .
- 4.  $\mathsf{P}_{\mathsf{A}}$  samples  $(c_1, \mu) \leftarrow \mathbb{F}_{2^{\lambda}}^2$  and sets  $k := \mathsf{K}[r_1] \oplus c_1$ ,

$$\left(\boldsymbol{v}, \left\{K_0^i\right\}_{i\in[1,d-1]}, \left(K_0^d, K_1^d\right)\right) := \mathsf{pcGGM}.\mathsf{FullEval}(\varDelta, k),$$

 $c_{i} := \mathsf{K}[r_{i}] \oplus K_{0}^{i} \text{ for } i \in [2, d), c_{n}^{\sigma} := \mathsf{Convert}_{\mathbb{K}} (\mathsf{H}_{S}(\mu \oplus \mathsf{K}[r_{d}] \oplus \sigma \cdot \Delta)) + K_{\sigma}^{d} \text{ for } \sigma \in \{0, 1\}, \text{ and } \psi := K_{0}^{d} + K_{1}^{d} - \mathsf{K}[\beta]. \mathsf{P}_{\mathsf{A}} \text{ sends } (c_{1}, \dots, c_{d-1}, \mu, c_{d}^{0}, c_{d}^{1}, \psi) \text{ to } \mathsf{P}_{\mathsf{B}}.$ 5.  $\mathsf{P}_{\mathsf{B}} \text{ sets } \alpha = \alpha_{1} \dots \alpha_{d} := \bar{r}_{1} \dots \bar{r}_{d} \in [0, N), K_{\alpha_{i}}^{i} := \mathsf{M}[r_{i}] \oplus c_{i} \text{ for } i \in [1, d), K_{\bar{\alpha}}^{d}, := c_{d}^{r_{d}} - \mathsf{Convert}_{\mathbb{K}} (\mathsf{H}_{S}(\mu \oplus \mathsf{M}[r_{d}])), \text{ and}$ 

$$[1, a), \mathbf{K}_{\bar{\alpha}_d} := c_d^{\mathbb{Z}} - \operatorname{Convert}_{\mathbb{K}} (\mathbf{H}_S (\mu \oplus \operatorname{\mathsf{M}} [r_d])), \text{ and}$$

$$\boldsymbol{u} := \mathbf{Unit}_{\mathbb{F}}(N, \alpha, \beta), \boldsymbol{w} := \mathsf{pcGGM}.\mathsf{PuncFullEval}\left(\alpha, \left\{K^i_{\bar{\alpha}_i}\right\}_{i \in [1,d]}, \psi + \mathsf{M}[\beta]\right).$$

6.  $P_A$  outputs v and  $P_B$  outputs (u, w).

**Macro.** For  $N \in \mathbb{N}, \alpha \in [0, N), \beta \in \mathbb{F}$ ,  $\mathbf{Unit}_{\mathbb{F}}(N, \alpha, \beta)$  returns  $\boldsymbol{u} \in \mathbb{F}^N$  s.t.  $u_i = 0$  for  $i \neq \alpha$  and  $u_{\alpha} = \beta$ .

Fig. 8: The single point subfield oblivious vector linear evaluation protocol with pseudorandom correlated GGM tree algorithms.

#### 7.1 Performance of the AES-based CCR Hash Function

As in the use cases of AVC (Section 5) and AHC (Section 6) schemes, the CCR hash function is called on a large number of independent blocks. So we tested the performance of the CCR hash function running in this scenario. In particular, we iteratively ran the CCR hash function for many rounds, using the hash output of one invocation as the input of the next invocation. Then, we measured the total running time of the CCR computation and reported the average running time of a single invocation in Table 1. Notice that since when the security parameter is 128,

our AES-based construction becomes identical with MMO-style construction, so the performance of Rijndael-based construction was not tested in this setting.

We noticed that the running time of our construction is slower than the Rijndael-based counterpart for the security parameters of 192 and 256. The increased running time is most likely because our construction requires key scheduling for the security parameters 192 and 256 while the Rijndael-based construction uses fixed key. We also noticed that the 256 bit version of Rijndael runs slightly faster than the 192 bit version. By inspecting the compiled assembly codes, we discovered that the 192 bit Rijndael encryption function indeed has more more instructions in total compared to the 256 bit version. In particular, the vectorized implementation of Rijndael in FAEST [9] introduces operations to process the internal state so that the AES-NI instructions can be utilized. The 192-bit Rijndael requires more such instructions than the 256-bit case. Therefore, despite the 256-bit Rijndael has two additional rounds, it still has a smaller total instruction count.

Advantages of Constructions from AES. We would also like to stress that since AES is a standardized algorithm it has better time-tested security compared to other configurations of Rijndael. Moreover, due to the standardized status of AES, it is possible that in many realistic scenarios, satisfactory hardware support is only provided for the "whole" AES (via low-latency AES IPs [16,55,53] or side-channel protected AES IPs [54]), but not available for AES round functions nor Rijndael permutations. In these cases, constructions built upon the "whole" AES would better meet the relevant implementation requirements (either low-latency or side-channel security) than those built upon the AES round functions.

Table 1: The performance comparison between our AES-based CCR hash function and Rijndael-based MMO-style CCR hash function [39]. The running time is measured in CPU cycles and is averaged over 100,000,000 invocations.

| Construction | Sec   | urity Paran | Standardization |     |
|--------------|-------|-------------|-----------------|-----|
|              | 128   | 192         | 256             |     |
| Rijndael-MMO | -     | 150.02      | 148.74          | No  |
| Our Scheme   | 40.98 | 250.52      | 361.78          | Yes |

### 7.2 Performance of Leaf Node Commitment Schemes in AVC

We also performed experiments to investigate the performance gain of using AES-based commitment scheme as compared to commitment schemes from cryptographic hash functions to commit the leaf nodes in an all-but-one vector commitment scheme (AVC). Recall that in an AVC scheme the leaf node x has

length of the security parameter  $\lambda$  and in previous works [6,2] such commitment is defined by  $\text{Com}(x) = \text{H}_{\text{RO}}(x)$  where  $\text{H}_{\text{RO}}$  is a random oracle that maps  $\lambda$ -bit strings to  $2\lambda$ -bit strings and is typically instantiated by SHA3.

Following the methodology of the previous experiment, we ran the commitment schemes (either AES-based or SHA3-based) in an iterative manner, feeding the first half of the commitment output of one round as the input of the next commitment invocation. We reported the benchmark result in Table 2. We observed that the AES-based commitment scheme has  $7 \sim 30$  times improvement in terms of running time over the SHA3-based construction. This result confirms that the AES-based commitment scheme has significant performance benefits as compared to SHA3-based counterparts.

Table 2: The performance comparison between AES-based commitment scheme and SHA3-based commitment scheme. The running time is measured in CPU cycles and is averaged over 100,000,000 invocations.

| Construction | Security Parameter |         |         | Standardization |
|--------------|--------------------|---------|---------|-----------------|
|              | 128                | 192     | 256     |                 |
| SHA3-based   | 2842.11            | 2863.31 | 2892.89 | Yes             |
| AES-based    | 41.78              | 264.77  | 378.05  | Yes             |

### References

- Abed, F., Forler, C., List, E., Lucks, S., Wenzel, J.: Counter-bDM: A provably secure family of multi-block-length compression functions. In: Pointcheval, D., Vergnaud, D. (eds.) AFRICACRYPT 14. LNCS, vol. 8469, pp. 440–458. Springer, Heidelberg (May 2014). https://doi.org/10.1007/978-3-319-06734-6\_26
- Aguilar-Melchor, C., Feneuil, T., Gama, N., Gueron, S., Howe, J., Joseph, D., Joux, A., Persichetti, E., Randrianarisoa, T.H., Rivain, M., Yue, D.: SDitH — Syndrome Decoding in the Head. Tech. rep., National Institute of Standards and Technology (2023), available at https://csrc.nist.gov/Projects/pqc-dig-sig/ round-1-additional-signatures
- Aguilar Melchor, C., Hülsing, A., Joseph, D., Majenz, C., Ronen, E., Yue, D.: SDitH in the QROM. pp. 317–350. LNCS, Springer, Heidelberg (Dec 2023). https: //doi.org/10.1007/978-981-99-8739-9\_11
- Armknecht, F., Fleischmann, E., Krause, M., Lee, J., Stam, M., Steinberger, J.P.: The preimage security of double-block-length compression functions. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 233–251. Springer, Heidelberg (Dec 2011). https://doi.org/10.1007/978-3-642-25385-0\_13
- 5. Baum, C., Beullens, W., Mukherjee, S., Orsini, E., Ramacher, S., Rechberger, C., Roy, L., Scholl, P.: One tree to rule them all: Optimizing GGM trees and OWFs for post-quantum signatures. Cryptology ePrint Archive, Paper 2024/490 (2024), https://eprint.iacr.org/2024/490, https://eprint.iacr.org/2024/490

- Baum, C., Braun, L., de Saint Guilhem, C.D., Klooß, M., Majenz, C., Mukherjee, S., Orsini, E., Ramacher, S., Rechberger, C., Roy, L., Scholl, P.: FAEST. Tech. rep., National Institute of Standards and Technology (2023), available at https: //csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures
- Baum, C., Braun, L., Delpech de Saint Guilhem, C., Klooß, M., Orsini, E., Roy, L., Scholl, P.: Publicly verifiable zero-knowledge and post-quantum signatures from VOLE-in-the-head. pp. 581–615. LNCS, Springer, Heidelberg (Aug 2023). https: //doi.org/10.1007/978-3-031-38554-4\_19
- Baum, C., Braun, L., Munch-Hansen, A., Scholl, P.: MozZ<sub>2<sup>k</sup></sub> arella: Efficient vector-OLE and zero-knowledge proofs over Z<sub>2<sup>k</sup></sub>. In: CRYPTO 2022, Part IV. pp. 329–358. LNCS, Springer, Heidelberg (Aug 2022). https://doi.org/10.1007/ 978-3-031-15985-5\_12
- Baum, C., Braun, L., de Saint Guilhem, C.D., Klooß, M., Majenz, C., Mukherjee, S., Orsini, E., Ramacher, S., Rechberger, C., Roy, L., Scholl, P.: FAEST: Algorithm Specifications. Tech. rep., National Institute of Standards and Technology (2023), available at https://faest.info/faest-spec-v1.1.pdf
- Baum, C., Braun, L., de Saint Guilhem, C.D., Klooß, M., Orsini, E., Roy, L., Scholl, P.: Publicly verifiable zero-knowledge and post-quantum signatures from vole-in-the-head. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part V. LNCS, vol. 14085, pp. 581–615. Springer (2023). https://doi.org/10.1007/ 978-3-031-38554-4\_19, https://doi.org/10.1007/978-3-031-38554-4\_19
- Baum, C., Delpech de Saint Guilhem, C., Kales, D., Orsini, E., Scholl, P., Zaverucha, G.: Banquet: Short and fast signatures from AES. In: Garay, J. (ed.) PKC 2021, Part I. LNCS, vol. 12710, pp. 266–297. Springer, Heidelberg (May 2021). https://doi.org/10.1007/978-3-030-75245-3\_11
- Baum, C., Malozemoff, A.J., Rosen, M.B., Scholl, P.: Mac'n'cheese: Zero-knowledge proofs for boolean and arithmetic circuits with nested disjunctions. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part IV. LNCS, vol. 12828, pp. 92– 122. Springer, Heidelberg, Virtual Event (Aug 2021). https://doi.org/10.1007/ 978-3-030-84259-8\_4
- Bellare, M., Hoang, V.T., Keelveedhi, S., Rogaway, P.: Efficient garbling from a fixed-key blockcipher. In: 2013 IEEE Symposium on Security and Privacy. pp. 478–492. IEEE Computer Society Press (May 2013). https://doi.org/10.1109/ SP.2013.39
- 14. Berti, F., Guo, C., Pereira, O., Peters, T., Standaert, F.X.: TEDT: a leakageresistant AEAD mode. IACR TCHES 2020(1), 256-320 (2019). https://doi. org/10.13154/tches.v2020.i1.256-320, https://tches.iacr.org/index.php/ TCHES/article/view/8400
- Beullens, W.: Sigma protocols for MQ, PKP and SIS, and Fishy signature schemes. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part III. LNCS, vol. 12107, pp. 183–211. Springer, Heidelberg (May 2020). https://doi.org/10.1007/ 978-3-030-45727-3\_7
- Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knežević, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçin, T.: PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 208–225. Springer, Heidelberg (Dec 2012). https://doi.org/ 10.1007/978-3-642-34961-4\_14
- Boyle, E., Couteau, G., Gilboa, N., Ishai, Y.: Compressing vector OLE. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) ACM CCS 2018. pp. 896–912. ACM Press (Oct 2018). https://doi.org/10.1145/3243734.3243868

- Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Resch, N., Scholl, P.: Correlated pseudorandomness from expand-accumulate codes. In: CRYPTO 2022, Part II. pp. 603–633. LNCS, Springer, Heidelberg (Aug 2022). https://doi.org/ 10.1007/978-3-031-15979-4\_21
- Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Scholl, P.: Efficient pseudorandom correlation generators: Silent OT extension and more. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 489–518. Springer, Heidelberg (Aug 2019). https://doi.org/10.1007/ 978-3-030-26954-8\_16
- Bui, D., Chu, H., Couteau, G., Wang, X., Weng, C., Yang, K., Yu, Y.: An efficient ZK compiler from SIMD circuits to general circuits. Cryptology ePrint Archive, Paper 2023/1610 (2023), https://eprint.iacr.org/2023/1610
- Bui, D., Cong, K., de Saint Guilhem, C.D.: Improved all-but-one vector commitment with applications to post-quantum signatures. Cryptology ePrint Archive, Paper 2024/097 (2024), https://eprint.iacr.org/2024/097, https://eprint. iacr.org/2024/097
- Carozza, E., Couteau, G., Joux, A.: Short signatures from regular syndrome decoding in the head. pp. 532–563. LNCS, Springer, Heidelberg (Jun 2023). https://doi.org/10.1007/978-3-031-30589-4\_19
- Chen, S., Steinberger, J.P.: Tight security bounds for key-alternating ciphers. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 327–350. Springer, Heidelberg (May 2014). https://doi.org/10.1007/ 978-3-642-55220-5\_19
- Chen, Y.L., Tessaro, S.: Better security-efficiency trade-offs in permutation-based two-party computation. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part II. LNCS, vol. 13091, pp. 275–304. Springer, Heidelberg (Dec 2021). https: //doi.org/10.1007/978-3-030-92075-3\_10
- Daemen, J., Rijmen, V.: The block cipher rijndael. In: International Conference on Smart Card Research and Advanced Applications. pp. 277–284. Springer (1998)
- Delpech de Saint Guilhem, C., De Meyer, L., Orsini, E., Smart, N.P.: BBQ: Using AES in picnic signatures. In: Paterson, K.G., Stebila, D. (eds.) SAC 2019. LNCS, vol. 11959, pp. 669–692. Springer, Heidelberg (Aug 2019). https://doi.org/10. 1007/978-3-030-38471-5\_27
- Delpech de Saint Guilhem, C., Orsini, E., Tanguy, T.: Limbo: Efficient zeroknowledge MPCitH-based arguments. In: Vigna, G., Shi, E. (eds.) ACM CCS 2021. pp. 3022–3036. ACM Press (Nov 2021). https://doi.org/10.1145/3460120. 3484595
- Dittmer, S., Eldefrawy, K., Graham-Lengrand, S., Lu, S., Ostrovsky, R., Pereira, V.: Boosting the performance of high-assurance cryptography: Parallel execution and optimizing memory access in formally-verified line-point zero-knowledge. pp. 2098–2112. ACM Press (2023). https://doi.org/10.1145/3576915.3616583
- 29. Dittmer, S., Ishai, Y., Lu, S., Ostrovsky, R.: Improving line-point zero knowledge: Two multiplications for the price of one. pp. 829–841. ACM Press (2022). https: //doi.org/10.1145/3548606.3559385
- Dittmer, S., Ishai, Y., Ostrovsky, R.: Line-point zero knowledge and its applications. Cryptology ePrint Archive, Report 2020/1446 (2020), https://eprint. iacr.org/2020/1446
- Dobraunig, C., Kales, D., Rechberger, C., Schofnegger, M., Zaverucha, G.: Shorter signatures based on tailor-made minimalist symmetric-key crypto. pp. 843–857. ACM Press (2022). https://doi.org/10.1145/3548606.3559353

- 32. Feneuil, T.: Building MPCitH-based signatures from MQ, MinRank, and rank SD. In: ACNS 24, Part I. pp. 403–431. LNCS, Springer, Heidelberg (Jun 2024). https://doi.org/10.1007/978-3-031-54770-6\_16
- Feneuil, T., Joux, A., Rivain, M.: Syndrome decoding in the head: Shorter signatures from zero-knowledge proofs. In: CRYPTO 2022, Part II. pp. 541–572. LNCS, Springer, Heidelberg (Aug 2022). https://doi.org/10.1007/ 978-3-031-15979-4\_19
- Feneuil, T., Rivain, M.: MQOM MQ on my Mind. Tech. rep., National Institute of Standards and Technology (2023), available at https://csrc.nist.gov/ Projects/pqc-dig-sig/round-1-additional-signatures
- Franzese, N., Katz, J., Lu, S., Ostrovsky, R., Wang, X., Weng, C.: Constantoverhead zero-knowledge for RAM programs. In: Vigna, G., Shi, E. (eds.) ACM CCS 2021. pp. 178–191. ACM Press (Nov 2021). https://doi.org/10.1145/ 3460120.3484800
- Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions (extended abstract). In: 25th FOCS. pp. 464–479. IEEE Computer Society Press (Oct 1984). https://doi.org/10.1109/SFCS.1984.715949
- Gueron, S.: Intel advanced encryption standard (aes) new instructions set. Intel Corporation 128 (2010)
- Guo, C., Katz, J., Wang, X., Weng, C., Yu, Y.: Better concrete security for halfgates garbling (in the multi-instance setting). In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part II. LNCS, vol. 12171, pp. 793–822. Springer, Heidelberg (Aug 2020). https://doi.org/10.1007/978-3-030-56880-1\_28
- 39. Guo, C., Katz, J., Wang, X., Yu, Y.: Efficient and secure multiparty computation from fixed-key block ciphers. In: 2020 IEEE Symposium on Security and Privacy. pp. 825–841. IEEE Computer Society Press (May 2020). https://doi.org/10. 1109/SP40000.2020.00016
- 40. Guo, C., Wang, X., Yang, K., Yu, Y.: On tweakable correlation robust hashing against key leakages. Cryptology ePrint Archive, Paper 2024/163 (2024), https: //eprint.iacr.org/2024/163
- 41. Guo, X., Yang, K., Wang, X., Zhang, W., Xie, X., Zhang, J., Liu, Z.: Half-tree: Halving the cost of tree expansion in COT and DPF. In: EUROCRYPT 2023, Part I. pp. 330–362. LNCS, Springer, Heidelberg (Jun 2023). https://doi.org/ 10.1007/978-3-031-30545-0\_12
- Hirose, S.: Some plausible constructions of double-block-length hash functions. In: Robshaw, M.J.B. (ed.) FSE 2006. LNCS, vol. 4047, pp. 210–225. Springer, Heidelberg (Mar 2006). https://doi.org/10.1007/11799313\_14
- Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. In: Johnson, D.S., Feige, U. (eds.) 39th ACM STOC. pp. 21–30. ACM Press (Jun 2007). https://doi.org/10.1145/1250790.1250794
- Kales, D., Zaverucha, G.: Efficient lifting for shorter zero-knowledge proofs and post-quantum signatures. Cryptology ePrint Archive, Report 2022/588 (2022), https://eprint.iacr.org/2022/588
- 45. Katz, J., Kolesnikov, V., Wang, X.: Improved non-interactive zero knowledge with applications to post-quantum signatures. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) ACM CCS 2018. pp. 525–537. ACM Press (Oct 2018). https: //doi.org/10.1145/3243734.3243805
- 46. Kim, S., Cho, J., Cho, M., Ha, J., Kwon, J., Lee, B., Lee, J., Lee, J., Lee, S., Moon, D., Son, M., Yoon, H.: AIMer. Tech. rep., National Institute of Standards and Technology (2023), available at https://csrc.nist.gov/Projects/pqc-dig-sig/ round-1-additional-signatures

- 47. Kim, S., Ha, J., Son, M., Lee, B., Moon, D., Lee, J., Lee, S., Kwon, J., Cho, J., Yoon, H., Lee, J.: AIM: Symmetric primitive for shorter signatures with stronger security. pp. 401–415. ACM Press (2023). https://doi.org/10.1145/3576915.3616579
- Lee, J., Stam, M., Steinberger, J.P.: The security of Tandem-DM in the ideal cipher model. Journal of Cryptology 30(2), 495–518 (Apr 2017). https://doi.org/10. 1007/s00145-016-9230-z
- Lin, F., Xing, C., Yao, Y.: More efficient zero-knowledge protocols over Z<sub>2<sup>k</sup></sub> via galois rings. Cryptology ePrint Archive, Paper 2023/150, To appear at CRYPTO 2024 (2023), https://eprint.iacr.org/2023/150
- 50. Melchor, C.A., Feneuil, T., Gama, N., Gueron, S., Howe, J., Joseph, D., Joux, A., Persichetti, E., Randrianarisoa, T.H., Rivain, M., Yue, D.: The Syndrome Decoding in the Head (SD-in-the-Head) Signature Scheme. Tech. rep., National Institute of Standards and Technology (2023), available at https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/ round-1/spec-files/SDitH-spec-web.pdf
- Patarin, J.: The "coefficients H" technique (invited talk). In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 328–345. Springer, Heidelberg (Aug 2009). https://doi.org/10.1007/978-3-642-04159-4\_21
- Pietrzak, K.: Cryptography from learning parity with noise. In: SOFSEM. vol. 12, pp. 99–114. Springer (2012)
- 53. Rambus: AES-IP-36 AES ECB/CBC/CTR accelerators, https://www.rambus. com/security/crypto-accelerator-cores/aes-ip-36/
- 54. Reuse, D.:: AES-DPA-FIA SW library Advanced DPA- and FIAresistant FortiCrypt AES SW library, https://www.design-reuse.com/sip/ advanced-dpa-and-fia-resistant-forticrypt-aes-sw-library-ip-52392/
- 55. Reuse, D.:: CryptoCORE 400G 400G AES Encryption Core, https://www. design-reuse.com/sip/400g-aes-encryption-core-ip-46356/
- Roy, L.: SoftSpokenOT: Quieter OT extension from small-field silent VOLE in the minicrypt model. In: CRYPTO 2022, Part I. pp. 657–687. LNCS, Springer, Heidelberg (Aug 2022). https://doi.org/10.1007/978-3-031-15802-5\_23
- 57. Schoppmann, P., Gascón, A., Reichert, L., Raykova, M.: Distributed vector-OLE: Improved constructions and implementation. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019. pp. 1055–1072. ACM Press (Nov 2019). https: //doi.org/10.1145/3319535.3363228
- 58. The OpenSSL Project: OpenSSL, https://github.com/openssl/openssl
- 59. Wang, X., Malozemoff, A.J., Katz, J.: EMP-toolkit: Efficient MultiParty computation toolkit. https://github.com/emp-toolkit (2016)
- Weng, C., Yang, K., Katz, J., Wang, X.: Wolverine: Fast, scalable, and communication-efficient zero-knowledge proofs for boolean and arithmetic circuits. In: 2021 IEEE Symposium on Security and Privacy. pp. 1074–1091. IEEE Computer Society Press (May 2021). https://doi.org/10.1109/SP40001.2021.00056
- Weng, C., Yang, K., Xie, X., Katz, J., Wang, X.: Mystique: Efficient conversions for zero-knowledge proofs with applications to machine learning. In: Bailey, M., Greenstadt, R. (eds.) USENIX Security 2021. pp. 501–518. USENIX Association (Aug 2021)
- Weng, C., Yang, K., Yang, Z., Xie, X., Wang, X.: AntMan: Interactive zeroknowledge proofs with sublinear communication. pp. 2901–2914. ACM Press (2022). https://doi.org/10.1145/3548606.3560667
- 63. Yang, K., Sarkar, P., Weng, C., Wang, X.: QuickSilver: Efficient and affordable zero-knowledge proofs for circuits and polynomials over any field. In: Vigna, G.,

Shi, E. (eds.) ACM CCS 2021. pp. 2986–3001. ACM Press (Nov 2021). https://doi.org/10.1145/3460120.3484556

- Yang, K., Weng, C., Lan, X., Zhang, J., Wang, X.: Ferret: Fast extension for correlated OT with small communication. In: Ligatti, J., Ou, X., Katz, J., Vigna, G. (eds.) ACM CCS 2020. pp. 1607–1626. ACM Press (Nov 2020). https://doi. org/10.1145/3372297.3417276
- Yang, Y., Heath, D.: Two shuffles make a RAM: Improved constant overhead zero knowledge RAM. Cryptology ePrint Archive, Paper 2023/1115, To appear at USENIX Security 2024 (2023), https://eprint.iacr.org/2023/1115
- Yang, Y., Heath, D., Hazay, C., Kolesnikov, V., Venkitasubramaniam, M.: Batchman and robin: Batched and non-batched branching for interactive ZK. pp. 1452–1466. ACM Press (2023). https://doi.org/10.1145/3576915.3623169
- 67. Zaverucha, G., Chase, M., Derler, D., Goldfeder, S., Orlandi, C., Ramacher, S., Rechberger, C., Slamanig, D., Katz, J., Wang, X., Kolesnikov, V., Kales, D.: Picnic. Tech. rep., National Institute of Standards and Technology (2020), available at https://csrc.nist.gov/projects/post-quantum-cryptography/ post-quantum-cryptography-standardization/round-3-submissions

## A Functionalities for Generating Additively Homomorphic Commitment Correlations.

As mentioned in Section 6, the state-of-the-art protocol [12,63,29,62] reduces the generation of additively homomorphic commitment correlation (also known as subfield vector oblivious linear evaluation correlation) to the generation of commitment of single point vectors (also known as single point subfield vector oblivious linear evaluation correlation). For completeness, we recall the  $\mathcal{F}_{sVOLE}$ and  $\mathcal{F}_{spsVOLE}$  functionalities in Fig. 9 and Fig. 10 respectively.

#### Functionality $\mathcal{F}_{\mathsf{sVOLE}}$

**Init.** Upon receiving (init) from  $P_A$  and  $P_B$ , sample  $\Delta \leftarrow \mathbb{F}_{p^r}$  if  $P_B$  is honest or receive  $\Delta \in \mathbb{F}_{p^r}$  from the adversary otherwise. Store global key  $\Delta$  and send  $\Delta$  to  $P_B$ , and ignore all subsequent init commands.

 ${\bf Extend.}$  This procedure can be run multiple times. Upon receiving (extend,  $\ell)$  from  $\mathsf{P}_A$  and  $\mathsf{P}_B,$  do:

- 1. If  $\mathsf{P}_{\mathsf{B}}$  is honest, sample  $\mathsf{K}[\boldsymbol{x}] \leftarrow \mathbb{F}_{p^r}^{\ell}$ . Otherwise, receive  $\mathsf{K}[\boldsymbol{x}] \in \mathbb{F}_{p^r}^{\ell}$  from the adversary.
- 2. If  $\mathsf{P}_{\mathsf{A}}$  is honest, sample  $\boldsymbol{x} \leftarrow \mathbb{F}_{p}^{\ell}$  and compute  $\mathsf{M}[\boldsymbol{x}] := \mathsf{K}[\boldsymbol{x}] + \Delta \cdot \boldsymbol{x} \in \mathbb{F}_{p^{r}}^{\ell}$ . Otherwise, receive  $\boldsymbol{x} \in \mathbb{F}_{p}^{\ell}$  and  $\mathsf{M}[\boldsymbol{x}] \in \mathbb{F}_{p^{r}}^{\ell}$  from the adversary, and then recompute  $\mathsf{K}[\boldsymbol{x}] := \mathsf{M}[\boldsymbol{x}] \Delta \cdot \boldsymbol{x} \in \mathbb{F}_{p^{r}}^{\ell}$ .
- 3. Send  $(\boldsymbol{x}, \mathsf{M}[\boldsymbol{x}])$  to  $\mathsf{P}_{\mathsf{A}}$  and  $\mathsf{K}[\boldsymbol{x}]$  to  $\mathsf{P}_{\mathsf{B}}$ .

**Global-key query.** If  $\mathsf{P}_{\mathsf{A}}$  is corrupted, receive (guess,  $\Delta'$ ) from the adversary with  $\Delta' \in \mathbb{F}_{p^r}$ . If  $\Delta' = \Delta$ , send success to  $\mathsf{P}_{\mathsf{A}}$  and ignore any subsequent global-key query. Otherwise, send abort to both parties and abort.

Fig. 9: The subfield vector oblivious linear evaluation functionality.

#### Functionality $\mathcal{F}_{spsVOLE}$

**Init.** Upon receiving (init) from  $\mathsf{P}_{\mathsf{A}}$  and  $\mathsf{P}_{\mathsf{B}}$ , sample  $\Delta \leftarrow \mathbb{F}_{p^r}$  if  $\mathsf{P}_{\mathsf{B}}$  is honest and receive  $\Delta \in \mathbb{F}_{p^r}$  from the adversary otherwise. Store global key  $\Delta$ , send  $\Delta$  to  $\mathsf{P}_{\mathsf{B}}$ , and ignore all subsequent init commands.

**Extend.** Upon receiving (sp-extend, N), where  $N = 2^d$  for some  $d \in \mathbb{N}$ , from  $\mathsf{P}_{\mathsf{A}}$  and  $\mathsf{P}_{\mathsf{B}}$ , do:

- 1. If  $\mathsf{P}_{\mathsf{B}}$  is honest, sample  $v \leftarrow \mathbb{F}_{p^r}^N$ . Otherwise, receive  $v \in \mathbb{F}_{p^r}^N$  from the adversary.
- 2. If  $\mathsf{P}_{\mathsf{A}}$  is honest, then sample  $v \in u_p^N$ , other init, reserve  $v \in u_p^N$  from the adversary, and compute  $\boldsymbol{w} := \boldsymbol{v} + \Delta \cdot \boldsymbol{u} \in \mathbb{F}_p^N$ . Otherwise, receive  $\boldsymbol{u} \in \mathbb{F}_p^N$  (with at most one nonzero entry) and  $\boldsymbol{w} \in \mathbb{F}_{p^r}^N$  from the adversary, and recompute  $\boldsymbol{v} := \boldsymbol{w} - \Delta \cdot \boldsymbol{u} \in \mathbb{F}_{p^r}^N$ .
- 3. If  $\mathsf{P}_{\mathsf{B}}$  is corrupted, receive a set  $I \subseteq [0, N)$  from the adversary. Let  $\alpha \in [0, N)$  be the index of the nonzero entry of  $\boldsymbol{u}$ . If  $\alpha \in I$ , send success to  $\mathsf{P}_{\mathsf{B}}$  and continue. Otherwise, send abort to both parties and abort.
- 4. Send (u, w) to  $\mathsf{P}_{\mathsf{A}}$  and v to  $\mathsf{P}_{\mathsf{B}}$ .

**Global-key query.** If  $\mathsf{P}_{\mathsf{A}}$  is corrupted, receive (guess,  $\Delta'$ ) from the adversary with  $\Delta' \in \mathbb{F}_{p^r}$ . If  $\Delta' = \Delta$ , send success to  $\mathsf{P}_{\mathsf{A}}$  and ignore any subsequent global-key query. Otherwise, send abort to both parties and abort.

Fig. 10: The single point subfield vector oblivious linear evaluation functionality.