AES-based Multi-instance TCCR Hash with High Security, and Applications

Hongrui Cui[1], Chun Guo[2,3,4], Xiao Wang[5], Chenkai Weng[6], Kang Yang[7], and Yu Yu[1,8]

[1] Shanghai Jiao Tong University, Shanghai, China
{rickfreeman,yyuu}@sjtu.edu.cn
[2] School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, China
[3] Key Laboratory of Cryptologic Technology and Information Security of Ministry of Education, Shandong University, Qingdao, Shandong, 266237, China
[4] Shandong Research Institute of Industrial Technology, Jinan, Shandong, 250102, China
chun.guo.sc@gmail.com
[5] Northwestern University, Evanston, USA
wangxiao@northwestern.edu
[6] Arizona State University, Tempe, USA
car1ckweng@gmail.com
[7] State Key Laboratory of Cryptology, Beijing, China
yangk@sklc.org
[8] Shanghai Qi Zhi Institute, Shanghai, China

Abstract. Correlation robust hash function (CRHF) is an important class of building blocks that have found applications in many cryptographic applications, including various protocols for zero-knowledge (ZK) proof, circuit garbling, vector oblivious linear evaluation (VOLE), etc. It offers more efficient instantiations from block ciphers as compared to cryptographic hash functions like SHA-3. State-of-the-art instantiations work provide near-optimal security in the ideal cipher model that can in turn be efficiently instantiated using AES, which is standardized and provides instruction-level acceleration. However, since AES is only available for 128-bit block size, they cannot be generalized beyond 128-bit security while enjoying the same level of efficiency.

In this paper, we fill this gap by constructing a new multi-instance tweakable circular CRHF (miTCCR) from AES, supporting up to 256-bit security. Since miTCCR is the most general notion of CRHF, it directly leads to numerous applications.

1. The state-of-the-art optimization of GGM tree (Eurocrypt'23), which builds on circular CRHF as a black box, can be instantiated with our construction, boosting its security strength, as well as the security of the VOLE-ZK protocols building on top, from 128 bits to 256 bits.
2. With a new extractability notion, we use the miTCCR hash to instantiate the leaf commitment in an all-but-one vector commitment scheme, which is a core primitive of post-quantum digital signature

schemes and ZK in the VOLE-in-the-Head framework. Experiments show that our technique could reduce the end-to-end signing/verification time of FAEST (resp. FAEST-EM) by 13.1% $\sim$ 40.4% / 15.2% $\sim$ 46.4% (resp. 1.5% $\sim$ 25.0% / 2.9% $\sim$ 24.2%).

3. The multi-instance-secure half-gates garbling scheme (Crypto'20) uses miTCCR as a black box. Our construction can strengthen its security as well as the security of constant-round multiparty computation building on half-gates to 246-bit.

1 Introduction

Hash functions are fundamental building blocks for modern cryptography, covering many applications in multiparty computation (MPC) and zero knowledge (ZK). Due to their extensive usage, optimizing the efficiency has become an important research direction. Beginning with [42] and developed by a series of subsequent works [37,36,38], *correlation robustness* (CR) has been identified and recognized as a balance point between utility and efficiency. Informally, a hash function $\mathsf{H} : \{0,1\}^\lambda \to \{0,1\}^\lambda$ is CR if the oracle $\mathcal{O}^\Gamma(x) := \mathsf{H}(\Gamma \oplus x)$ is a pseudorandom function, where Γ is sampled from a high-entropic distribution. On the utility end, several primitives in MPC can be constructed from (different flavors of) this hash function, including free-XOR garbling schemes [66,53], oblivious transfer extensions [42,5,54], Beaver triples [45]. Notably, the Goldreich-Goldwasser-Micali (GGM) tree [35] can be optimized by the circular variant of CR (CCR) hash [39], giving rise to efficiency enhancement for numerous applications including function secret sharing [68,19], correlated OT [39], vector oblivious linear evaluation (VOLE) [63,39], designated-verifier ZK [58,30,13,61,33,59,29,60,11,65,27,20,48,64] and ZK frameworks like MPC in the Head [43] and VOLE in the Head [10].

More importantly, on the efficiency end, correlation robust hash functions can be constructed efficiently from blockciphers [37,36,24,54,38]. Building on the standardized AES cipher and benefiting from the ubiquitous AES-NI instruction set, those constructions demonstrate significantly better computational cost compared with cryptographic hash functions like SHA256 or SHA3 [14,37]. In particular, the benchmark result [37] shows that AES is about 50$\times$ faster than SHA256 and 120$\times$ faster than SHA3. We note that apart from the obvious efficiency advantage, the AES cipher is widely considered a mature choice in terms of security, as it has withstood decades of cryptanalysis effort.

Nevertheless, a problem with all existing constructions of correlation robust hash functions is that they all assume a permutation on λ bits where λ is the security parameter. For AES, this is only the case for $\lambda = 128$, as for 192-bit and 256-bit security levels, the block lengths are still 128 bits. Consequently, when using the AES cipher, current correlation robust hash functions only support applications with a 128-bit security parameter, which is insufficient for high-security scenarios (e.g., when considering adversaries with quantum capability.) Therefore, there is a gap between the standardized AES algorithm and existing CR hash constructions for 192-bit and 256-bit security levels.

1.1 Our Contributions

In this paper, we fill the above gap by presenting an AES-based correlation robust hash function with three security levels (i.e., 128, 192, and 256 bits). Moreover, we observed that many ZK protocols utilizing GGM trees require extractability of the leaf nodes [10,9,67,3]. Therefore, to better support these applications, we formalize a new extractability notion, which may be of independent interest. Specifically, we prove that in the ideal-cipher model, our new construction satisfies the most generic form of CR named multi-instance tweakable circular correlation robustness (miTCCR), which enables all existing variants of correlation robustness [36]. We also show that the new construction satisfies extractability.

We remark that CR and CCR hashes are much easier to construct than TCCR hashes, because they are not tweakable and so have much shorter inputs than TCCR hashes have. Using $\mathsf{AES}\text{-}\lambda$, the AES member with λ-bit keys, a somewhat trivial construction of a pseudorandom oracle with λ-bit inputs and 128ℓ-bit outputs is $PRO_\lambda(x) := \mathsf{AES}\text{-}\lambda(x,[0]_{128})\|...\|\mathsf{AES}\text{-}\lambda(x,[\ell-1]_{128})$. However, the security of this construction is capped at 128 bits,[9] falling short of instantiating highly secure constructions. Furthermore, this simple approach does not extend to constructing TCCR hash, which has much longer inputs. By these, our constructions of TCCR hash with high multi-instance security is non-trivial.

On the other hand, following the work [37], a TCCR hash function with both 192-bit and 256-bit security levels can be constructed from the 256-bit Rijndael blockcipher [26]. But the latter is not standardized and less studied than AES. Our construction, on the other hand, adopts the standard AES algorithm with better time-tested security. 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 [18,52,50] or side-channel protected AES IPs [51]), but not available for AES round functions nor Rijndael ciphers. 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.

When the AES-NI instructions (for AES round functions and key schedule) are available, both our construction and the Rijndael blockcipher enjoy accelerated implementations. In this case, our construction has a comparable performance as indicated by the experiment in Section 7.1.

We demonstrate the usefulness of the new construction and the extractability notion in three applications. Firstly, the state-of-the-art GGM tree optimization half-tree [39] utilizes CCR hash as a black box. The GGM tree serves as a basic primitive of a series of cryptographic protocols such as MPC [31,68], ZK [61], and privacy-preserving machine learning [40]. By applying our miTCCR hash

[9] A query to $PRO_\lambda(x)$ always gives rise to ℓ distinct 128-bit output blocks. Whereas for a true random oracle $\mathsf{H}_{\mathsf{RO}} : \{0,1\}^\lambda \mapsto \{0,1\}^{128\ell}$, an input with collided 128-bit output blocks can be obtained within approximately $2^{128}/\ell^2$ queries. By this, $PRO_\lambda(x)$ can be distinguished from a true random oracle with $2^{128}/\ell^2$ queries.

construction, we push the attainable security levels of half-tree as well as applications that rely on the GGM tree from 128 bits to up to 256 bits, from standard AES. Our experiment in Section 7.1 shows that the standardized AES cipher benefits substantially from VAES parallelism in the batched setting: at batch size 8, our construction achieves 5.88, 10.85, and 11.81 cycles per hash for $\lambda = 128, 192, 256$, outperforming the Rijndael-based miTCCR construction at all security levels (1.9×, 2.1×, and 1.6× faster, respectively). Compared to SHA-256 (SHA-NI) and SHA3 (XKCP with AVX2), our construction is $17 \sim 33\times$ and $6 \sim 11\times$ faster at batch size 8.

Secondly, we present an optimized all-but-one vector commitment scheme where we use the half-tree method to generate a binary tree and use miTCCR hash to commit the leaf nodes. We show that the security of the optimized scheme can be reduced to the newly proposed extractable property of the miTCCR hash function. To demonstrate its utility in improving the efficiency of ZK and signature schemes in the VOLE in the Head framework, we integrated it into the FAEST digital signature scheme [8]. Our experiment in Section 7.2 shows that our construction yields $13.1\% \sim 40.4\%$ / $15.2\% \sim 46.4\%$ (resp. $1.5\% \sim 25.0\%$ / $2.9\% \sim 24.2\%$) end-to-end reduction in signing / verification time for FAEST (resp. FAEST-EM).

Thirdly, we show that our miTCCR hash construction can serve as a drop-in replacement for the multi-instance-secure half-gate garbling scheme [36], giving rise to MPC applications [56,57,44,62,28,25]. As in the first application, our new hash function pushes the security levels of all downstream applications from 128 bits to up to 246 bits, from standard AES.

2 Technical Overview

In this section, we present an overview of the proof techniques, the new extractability notions, and the applications of this paper.

2.1 Multi-instance TCCR hash from AES

The main technical difficulty is that AES only has a 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 λ-bit key size, the left n-bit input x_L of the 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 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 [37]. For $n < \lambda \leq 2n$, we need two invocations

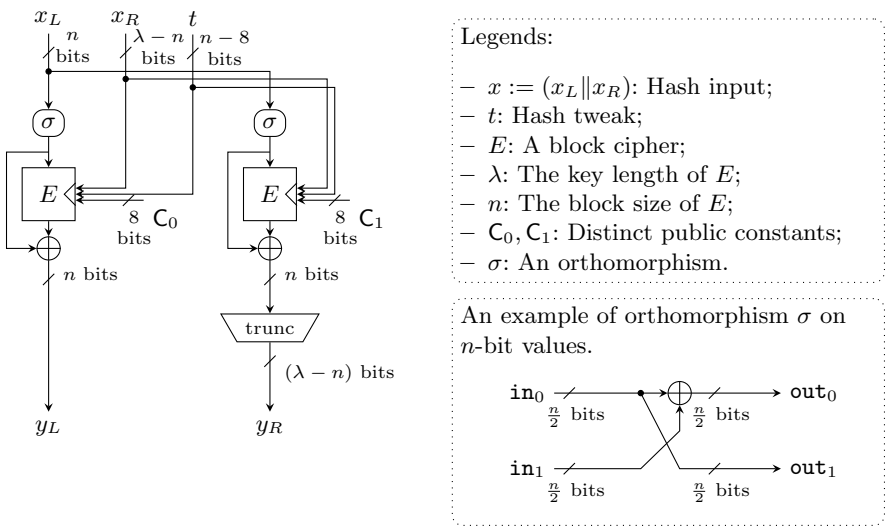

Fig. 1: The AES-based Multi-instance TCCR hash function construction diagram.

of the block cipher keyed with the input value x_R concatenated with the tweak and distinct public constants.

For input length $\lambda \in \mathbb{N}$ and tweak length $n_t \in \mathbb{N}$, the multi-instance tweakable circular correlation robust (miTCCR) property for a hash function $\mathsf{H} : \{0,1\}^\lambda \times \{0,1\}^{n_t} \to \{0,1\}^\lambda$ states that for u keys $\Gamma^1, ..., \Gamma^u \in \{0,1\}^\lambda$, the u *construction* oracles $\mathcal{O}_i(x, t, b) = \mathsf{H}(x \oplus \Gamma^i) \oplus b \cdot \Gamma^i$ for $i \in [u]$ should be indistinguishable from random functions.[1] In our construction, the key Γ 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 Γ can only be leaked when an ideal cipher query made by the adversary collides with some internal ideal cipher invocation made by the construction oracles, or two internal ideal cipher invocations made by two distinct construction oracle calls collides. Since the key Γ is distributed to both the secret key and the input block of the ideal cipher, a collision implies recovering the key Γ, which happens with negligible probability if the key is sampled from a high-entropic distribution. By applying a union bound on all possible collision events, we conclude that the construction in Fig. 1 is indeed miTCCR-secure. The above intuition is formally argued using the H-coefficient technique [49] in Section 4.2.

[1] A technical detail is that the distinguisher should not query the same (x, t) with both 0 and 1 to prevent a trivial attack.

2.2 Extractable miTCCR Hash Function

Motivated by the extractable binding property of all-but-one vector commitment (AVC) schemes [10], we formulate the extractable property of miTCCR hash functions. Intuitively, the property states that a hash function H is extractable, if given two purported hash outputs with correlated inputs $y = \mathsf{Com}(x) = \mathsf{H}(x \oplus \mathsf{C}_2)\|\mathsf{H}(x \oplus \mathsf{C}_3)$ (C_2 and C_3 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 the adversary finds a collision in the 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 [47]. 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 a 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 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. Moreover, the pile itself has sufficient entropy such that collision and inversion are unlikely over a union bound on all normal queries. On the other hand, although we grant the adversary many free queries in a super query, the adversary must spend many normal queries to get that super query and thus the total number of piles across all super queries is limited. The marginal distribution of every pile in a super query is uniform over half of the input space, and therefore the adversary's success probability is negligible over a union bound on all super queries. 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 $O(q_E/2^{2\lambda})$, where q_E is the number of ideal cipher queries of the adversary.

Using the derived bounds, we can capture the extraction failure probability of our construction and derive the precise security bounds of the proposed miTCCR hash function concerning the new extractability definition. Since our construction reduces to the MMO construction when the block size equals the security parameter, our proof also implies the extractable security of the hash function in [37] under the random permutation model.

2.3 Applications

We list three applications of the extractable miTCCR hash function with high security. As all existing variants of CR hash functions, namely correlated CR (CCR) and tweakable correlated CR (TCCR), can be derived from miTCCR, we assume in the following discussion that we have access to CCR and TCCR hash functions from AES with the same security guarantee.

Half-trees with high security from AES. In a GGM tree, each node in a binary tree is assigned a λ-bit label. For each node with label x, its two child nodes are defined by applying a length-doubling PRG on x. Given any fixed leaf node, all other leaf nodes can be derived from the *sibling nodes* on the path from the fixed leaf to the root. This feature is extensively utilized in many applications including VOLE, correlated OT (COT) [63,61,19,54], distributed point function (DPF), and distributed comparison function (DCF) [34,19,39].

Half-tree is an alternative method to derive a GGM tree using a hash function [39]. For any node with label x, its left node is computed as $\mathsf{H}(x)$ while its right node is $x \oplus \mathsf{H}(x)$. The intuition is that many applications (e.g., DPF, DCF, COT) only require the aforementioned sibling nodes to be pseudorandom. In the half-tree construction, those values can be computed by an oracle $\mathcal{O}(x, b) := \mathsf{H}(x \oplus \Gamma) \oplus b \cdot \Gamma$, which exactly corresponds to the construction oracle in the CCR definition. Therefore, the security of the half-tree reduces to the CCR security of H.

Previous constructions of CCR hash [37,36] only provide 128-bit CCR security from AES, limiting the available security levels of the half-tree optimization. Our work circumvents this barrier and enables the half-tree security levels to match AES's three security levels $(128, 192, 256)$. Notably, we can build highly efficient designated-verifier ZK protocols from VOLE correlations with *information theoretic* security [12]. Therefore, our new hash construction enables the composed VOLE-ZK protocols to enjoy both higher security levels and the efficiency offered by half-tree.

AVC from extractable miTCCR. All-but-one vector commitment (AVC) is a crucial building block of zero-knowledge proofs and digital signature schemes in the MPC-in-the-Head and VOLE-in-the-Head frameworks. In the commitment phase, a prover generates a random vector together with a commitment, while in the opening phase, the prover can disclose all but one position of the vector to the verifier. Security of the upper-level applications relies on the *extractable binding* property, which requires an extractor that outputs a vector or aborts given a commitment chosen by the malicious prover. This property asserts that with high probability, either the extractor's output vector matches any accepting outputs of the verifier, or the extractor aborts and the prover fails to open the commitment.[10]

[10] We note that the latest version of the FAEST signature scheme removes the extractable requirement of the AVC scheme [6]. Nevertheless, the new unforgeability proof is much more involved and requires specific tweaking of the signature scheme to fit with the new proof.

AVC is usually instantiated as follows. The prover first generates a GGM tree using a PRG (usually instantiated with AES counter mode). Then, for each leaf node with label x, a leaf commitment is computed. Finally, all the leaf commitments are hashed together with a random oracle H_{RO} to derive a total commitment of all leaves. In the 1st version of FAEST [10,9] and most MPC-in-the-Head digital signature/zero-knowledge schemes [3], the leaf commitment is derived with a random oracle typically instantiated by SHA3, which is substantially slower than AES-based hash functions, as shown in Table 1. In the 2nd version of FAEST-EM [8] and SDitH [2], the leaf commitment is derived by calling a PRG typically instantiated by AES counter mode, which requires non-black-box analysis in the ideal cipher model or non-standard properties of the PRG. The 2nd version of FAEST uses a Naor-style commitment scheme to derive leaf commitments and does not require non-standard PRG properties, but incurs a 1.5 times blow-up in commitment size. Bui et al. propose to use an MMO-style CCR hash function to derive leaf commitments [21]. Nevertheless, for security levels beyond 128 bits, current MMO-style CCR hash requires non-standardized Rijndael parameters [37].

With the extractable miTCCR hash function, we provide two improvements to the above construction. Firstly, instead of a vanilla GGM tree, we derive a half-tree with the new miTCCR hash function. Let N be a power of 2 and assume that the AES key scheduling and encryption take the same unit cost. The costs of generating a vanilla GGM tree with N leaf nodes using AES counter mode are $3(N-1)$, $4(N-1)$, and $5(N-1)$ at security levels 128, 192, and 256, respectively. When using our miTCCR hash function, the costs become $2(N-1)$, $4(N-1)$, and $4(N-1)$. Therefore, we achieve some degree of savings from this optimization. Secondly, instead of using random oracle or counter-mode PRG to compute the leaf commitments, we use two invocations of the miTCCR hash function. Our construction builds on standard AES, offers security beyond 128 bits, and is more efficient than current constructions.

Garbling Schemes. As the core component of constant-round two-party and multi-party computation, circuit garbling schemes extensively utilize hash functions in various row-reduction methods [66,53]. In [36], Guo et al. studied the concrete multi-instance security of the popular half-gate scheme [66]. They showed that miTCCR enables a concretely efficient variant of the half-gate garbling scheme, which has multi-instance security essentially matching its single-instance security.

Once again, existing miTCCR hash construction [36] only provides 128-bit security when instantiated from standard AES. By instantiating with our new construction, we get, for the first time, garbling schemes with security levels up to 246 bits from standard AES. We refer the readers to Appendix A for the half-gate variant of [36] and Section 6 for the security claim.

3 Preliminaries

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

3.1 Notation

We use λ 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 $\mathbf{A}$ 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. We define $n_t = n - 8$ as the tweak size relevant to the tweakable correlation robust hash construction. Since we only consider the AES algorithm in this work, we assume that the key size always equals the security parameter λ. 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

- σ is linear,
- σ is a permutation,
- $\sigma \oplus \mathsf{id}$ is a permutation, where id is the identity map.

We can instantiate σ 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 Definitions

We recall the definition of circular correlation robustness of [37] (in the ideal cipher model) in Definition 1 and the definition of multi-instance tweakable circular correlation robustness of [36] in Definition 2.

Definition 1 ([37]). *Let $\mathsf{H}^E : \{0,1\}^\lambda \to \{0,1\}^\lambda$ be a function defined upon an ideal cipher E and $\mathcal{R}$ be a distribution over $\{0,1\}^\lambda$. For $\Gamma \leftarrow \mathcal{R}$, define $\mathcal{O}^{\mathsf{ccr}}_\Gamma(w, b) = \mathsf{H}^E(w \oplus \Gamma) \oplus b \cdot \Gamma$. We do not allow the distinguisher to query the same w with both $b = 0$ and $b = 1$ to avoid the trivial attack. For a distinguisher D, we define the following advantage*

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

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

Definition 2 ([36]). *Let $\mathsf{H}^E : \{0,1\}^\lambda \times \{0,1\}^{n_t} \to \{0,1\}^\lambda$ be a function defined upon an ideal cipher E and $\mathcal{R}$ be a distribution over $\{0,1\}^\lambda$. Let $u, \mu \in \mathbb{N}^+$ be the number of keys and the maximum number of $\mathcal{O}^{\mathsf{miTCCR}}$-oracle queries on any fixed tweak. For $\Gamma \in \{0,1\}^\lambda$, $t \in \{0,1\}^{n_t}$, define $\mathcal{O}^{\mathsf{miTCCR}}_\Gamma(w,t,b) = \mathsf{H}^E(w \oplus \Gamma, t) \oplus b \cdot \Gamma$. We don't allow the distinguisher to query the same (w,t) with both $b=0$ and $b=1$ to avoid the trivial attack. For any tweak $t \in \{0,1\}^{n_t}$, the distinguisher is also not allowed to make $(\cdot,t,\cdot)$-queries across all the $\mathcal{O}^{\mathsf{miTCCR}}$-oracles for more than μ times. For a distinguisher D, we define the following advantage*

$$\mathsf{Adv}^{\mathsf{miTCCR}}_{\mathsf{H}}(D,u,\mu) := \Big| \Pr_{\Gamma^1,\ldots,\Gamma^u \leftarrow \mathcal{R}}[D^{E,\mathcal{O}^{\mathsf{miTCCR}}_{\Gamma^1}(\cdot),\ldots,\mathcal{O}^{\mathsf{miTCCR}}_{\Gamma^u}(\cdot)}(1^\lambda)=1] - \Pr_{f_1,\ldots,f_u \leftarrow \mathcal{F}_{\lambda+n_t+1,\lambda}}[D^{E,f_1(\cdot),\ldots,f_u(\cdot)}(1^\lambda)=1] \Big| ,$$

where $\mathcal{F}_{\lambda+n_t+1,\lambda}$ denotes the set of all functions mapping $(\lambda+n_t+1)$-bit inputs to λ-bit outputs. H *is $(q_E, q_C, u, \mu, \rho, \epsilon)$-multi-instance tweakable circular correlation robust if for all D making at most q_E queries to E and at most q_C queries to all the $\mathcal{O}^{\mathsf{miTCCR}}$-oracles and $\mathcal{R}$ with min-entropy at least ρ, we have $\mathsf{Adv}^{\mathsf{mitccr}}_{\mathsf{H}}(D,u,\mu) \leq \epsilon$.*

3.3 The H-coefficient Technique

We will employ Patarin's H-coefficient technique [49] 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 [49,23] 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 $T_{\mathsf{re}}, T_{\mathsf{id}}$ 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

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

where the sum is taken over all possible transcripts $\mathcal{Q}$.

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

$$\mathcal{Q} \in \mathcal{T}_1 \implies \Pr\left[T_{\mathsf{re}} = \mathcal{Q}\right] / \Pr\left[T_{\mathsf{id}} = \mathcal{Q}\right] \geq 1 - \epsilon_1. \tag{2}$$

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

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

is an upper bound on the distinguisher's advantage. One should think of ϵ_1 and $\Pr\left[T_{\mathsf{id}} \in \mathcal{T}_2\right]$ 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 Multi-instance Extractable TCCR Hash Function

In this section, we present the block-cipher-based multi-instance TCCR (miTCCR) hash constructions with bit security levels higher than the block sizes in Section 4.1. We then prove its security in Section 4.2. Then, in Section 4.3 we propose and prove an extractability notion of the hash function so that it becomes a more versatile construction. The plain TCCR property can be obtained by fixing the instance number to be one. Moreover, the CCR construction can be derived by fixing the tweak to be all zeros. Therefore, we only state the security claims of the CCR hash function in Section 4.4.

4.1 The Construction of $\mathsf{H}_{\mathsf{miTCCR}}$

Our AES-based miTCCR 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 the theoretical miTCCR hash constructions built upon an ideal cipher of n-bit block size and λ-bit keys, where λ is also the security parameter. In the miTCCR hash construction, we support tweak size $n_t = n - 8$ bits. (When instantiated with AES, the tweak size is 120 bits.) The CCR hash construction is obtained by fixing the tweak value in the miTCCR hash.

In our construction, we define two distinct constants $\mathsf{C}_0, \mathsf{C}_1 \in \mathbb{Z}_{2^8}$. Those constants, which can be expressed as 8-bit values, will be part of the ideal cipher key values to differentiate the two ideal cipher calls in a single hash computation.

4.2 The Multi-instance Tweakable Circular Correlation Robustness of $\mathsf{H}_{\mathsf{miTCCR}}$

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

Algorithm $\mathsf{H}_{\mathsf{miTCCR}}(1^\lambda, 1^n, r, t)$ // $t \in \{0,1\}^{120}$

- If $\lambda = 128$: return $\mathsf{AES\text{-}128}\big([\mathsf{C}_0]_8 \| t, \sigma(r)\big) \oplus \sigma(r)$
- If $\lambda = 192$:
  1. $r_L \leftarrow \mathsf{left}_{128}(r)$, $r_R \leftarrow \mathsf{right}_{64}(r)$
  2. return $\mathsf{AES\text{-}192}\big(r_R \| [\mathsf{C}_0]_8 \| t, \sigma(r_L)\big) \oplus \sigma(r_L)$
     $\big\| \mathsf{left}_{64}\Big(\mathsf{AES\text{-}192}\big(r_R \| [\mathsf{C}_1]_8 \| t, \sigma(r_L)\big) \oplus \sigma(r_L)\Big)$
- If $\lambda = 256$:
  1. $r_L \leftarrow \mathsf{left}_{128}(r)$, $r_R \leftarrow \mathsf{right}_{128}(r)$
  2. return $\mathsf{AES\text{-}256}\big(r_R \| [\mathsf{C}_0]_8 \| t, \sigma(r_L)\big) \oplus \sigma(r_L)$
     $\big\| \mathsf{AES\text{-}256}\big(r_R \| [\mathsf{C}_1]_8 \| t, \sigma(r_L)\big) \oplus \sigma(r_L)$

Fig. 2: The AES-based miTCCR hash function. Here C_0 and C_1 are two public constants that can be expressed in 8 bits. r is a λ-bit value, t is a 120-bit value, $[\cdot]_n$ denotes the n-bit binary representation, σ is an orthomorphism on 128-bit values, left_n (resp. right_n) denotes taking the left (resp. right) n bits of the input.

this proof technique. The proof follows the same high-level ideas as [37, Theorem 5], where we define the bad event as the internal ideal cipher query in the construction oracle query generating collisions either internally or with some explicit ideal cipher queries. Due to using more than one ideal cipher query for the $\lambda > n$ cases, the bad event is slightly more complicated than the proof in [37]. We defer the full proof to Section B.

Lemma 1. *If we model E as an ideal cipher, then the hash function $\mathsf{H}^E_{\mathsf{miTCCR}}$ defined in Fig. 3 is $(q_E, q_C, u, \mu, \rho, \epsilon)$-multi-instance tweakable circular correlation robust, where*

$$\epsilon \leq \frac{4q_E\mu}{2^\rho} + \frac{q_C(\mu-1)}{2^n}. \tag{4}$$

4.3 The Extractable Binding Property of $\mathsf{H}_{\mathsf{miTCCR}}$

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

Definition 3. *Let $\mathsf{H}^E : \{0,1\}^\lambda \times \{0,1\}^{n_t} \to \{0,1\}^\lambda$ be a hash function and $[\mathsf{C}_2]_\lambda, [\mathsf{C}_3]_\lambda \in \{0,1\}^\lambda$ be two public distinct constants. Let $m \in \{0,1\}^\lambda$ be a message and $t \in \{0,1\}^{n_t}$ be a tweak and let $\mathsf{Com}(m,t) := \mathsf{H}^E(m \oplus [\mathsf{C}_2]_\lambda, t) \| \mathsf{H}^E(m \oplus [\mathsf{C}_3]_\lambda, t)$ be a commitment scheme and $\mathsf{Ext}^E(c, t, \mathcal{Q}_E)$ be an extractor that outputs the pre-image of a purported commitment c. We say that the hash function H^E satisfies $(q_E, \varepsilon, \delta)$-*extractable binding *if for any stateful PPT adversary $\mathcal{A}$*

Algorithm $\mathsf{H}^E_{\mathsf{miTCCR}}(1^\lambda, 1^n, r, t)$
// t: a tweak of length $n-8$

– If $n \le \lambda \le 2n$:
  1. $r_L \leftarrow \mathsf{left}_n(r)$, $r_R \leftarrow \mathsf{right}_{\lambda-n}(r)$
  2. $z_L \leftarrow E\big(r_R \| [\mathsf{C}_0]_8 \| t, \sigma(r_L)\big) \oplus \sigma(r_L)$
  3. $z_R \leftarrow \mathsf{left}_{\lambda-n}\big(E\big(r_R \| [\mathsf{C}_1]_8 \| t, \sigma(r_L)\big) \oplus \sigma(r_L)\big)$ // omit if $n = \lambda$
  4. return $z_L \| z_R$

Fig. 3: The abstract model for the AES-based miTCCR hash function. Here C_0 and C_1 are two public constants that can be expressed in 8 bits. r is a λ-bit value, t is a $(n-8)$-bit value, $[\cdot]_n$ denotes the n-bit binary representation, σ is an orthomorphism on n-bit values, left_n (resp. right_n) denotes taking the left (resp. right) n bits of the input.

making at most q_E queries to E in the following experiment $\mathbf{Exp}^{\mathrm{ebind}}_{\mathsf{H}^E}(\mathcal{A}, \mathsf{Ext})$, *we have* $\Pr[\mathbf{Exp}^{\mathrm{ebind}}_{\mathsf{H}^E}(\mathcal{A}, \mathsf{Ext}) = \mathtt{fail}] \le \delta$ *and* $\Pr[\mathbf{Exp}^{\mathrm{ebind}}_{\mathsf{H}^E}(\mathcal{A}, \mathsf{Ext}) = \mathtt{coll}] \le \varepsilon$.

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

1. $(c, \mathcal{Q}_E) \leftarrow \mathcal{A}^E(\mathtt{commit})$
   // $\mathcal{Q}_E$ *is the transcript of E-queries and answers of* $\mathcal{A}^E$
2. $(m^*, t^*) \leftarrow \mathsf{Ext}^E(c, \mathcal{Q}_E)$
3. $(m', t') \leftarrow \mathcal{A}^E(\mathtt{open})$
4. *Output* `fail` *if* $\mathsf{Com}(m', t') = c \wedge m^* = \bot$
5. *Output* `coll` *if* $\mathsf{Com}(m', t') = c \wedge m^* \neq \bot \wedge (m', t') \neq (m^*, t)$

Remark 1. To the best of our knowledge, almost all existing ZK schemes in the MPC-in-the-Head and VOLE-in-the-Head frameworks employ commitment with a similar structure as in Definition 3. E.g. in [9,3] the commitment consists of the concatenation of hashes of the message XOR'ed with 0 and 1, but the hash function is required to be a random oracle. In [22] the construction is the same. One exception is the latest version of FAEST [6] where the extractability of the commitment scheme is relaxed to plain binding due to a more delicate analysis of the digital signature scheme which only requires soundness of the underlying ZK scheme rather than knowledge soundness.

Remark 2. We note that the extractable binding property in Definition 3 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 or TCCR properties of the hash functions. Therefore, we do not explicitly define the hiding property of the Com in Definition 3.

To facilitate a modular proof of the extractable binding property of $\mathsf{H}_{\mathsf{miTCCR}}$, we define the following collision resistant property with regard to hash functions.

Definition 4. *Let* H^E *be a hash function and* Com *be its induced commitment scheme as defined in Definition 3. We call this hash function* (q_E, ε)*-collision resistant if for an adversary making at most* q_E *queries to* E*, the probability of winning the following experiment is bounded by* ε*.*

Experiment $\mathbf{Exp}^{\mathrm{Coll}}_{\mathsf{Com}}(\mathcal{A})$

1. $(m, t, m', t') \leftarrow \mathcal{A}^E(\mathtt{commit})$
2. $\mathcal{A}$ *wins if* $\mathsf{Com}(m,t) = \mathsf{Com}(m',t') \wedge (m,t) \neq (m',t')$.

$\mathsf{H}_{\mathsf{miTCCR}}$ is collision-resistant. We distinguish between the cases of $\lambda = n$ and $n < \lambda \leq 2n$. When $\lambda = n$, the construction $\mathsf{H}_{\mathsf{miTCCR}}(m \oplus [\mathsf{C}_2]_\lambda, t) \| \mathsf{H}_{\mathsf{miTCCR}}(m \oplus [\mathsf{C}_3]_\lambda, t)$ is essentially a special case of the Hirose double-block-length compression function [41], 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 $O({q_E}^2/2^{2n})$ was given by Hirose [41, Theorem 4]. However, this is a bit buggy, since using q_E queries an "internal collision" of the form $\mathsf{H}^E_{\mathsf{miTCCR}}(m \oplus [\mathsf{C}_2]_\lambda, t) \| \mathsf{H}^E_{\mathsf{miTCCR}}(m \oplus [\mathsf{C}_3]_\lambda, t) = \mathsf{H}^E_{\mathsf{miTCCR}}(m \oplus [\mathsf{C}_3]_\lambda, t) \| \mathsf{H}^E_{\mathsf{miTCCR}}(m \oplus [\mathsf{C}_2]_\lambda, t)$ can be found with a success probability of $O(q_E/2^n)$. On the other hand, Berti et al. [16, Lemma 1] proved that the probability of finding a collision for the truncated construction $\mathsf{H}^E_{\mathsf{miTCCR}}(m \oplus [\mathsf{C}_2]_\lambda, t) \| \mathsf{left}_{n-1}\big(\mathsf{H}^E_{\mathsf{miTCCR}}(m \oplus [\mathsf{C}_3]_\lambda, t)\big)$ 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* $[\mathsf{C}_2]_n = [\mathsf{C}_0]_n$*,* $[\mathsf{C}_3]_n = [\mathsf{C}_1]_n$*. 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 3 is* $(q_E, 6q_E/2^n)$*-collision-resistant.*

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

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

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

Lemma 3. *Consider the case* $n < \lambda \leq 2n$*, and let* $[\mathsf{C}_2]_\lambda = [\mathsf{C}_0]_n \| [0]_{\lambda-n}$*,* $[\mathsf{C}_3]_\lambda = [\mathsf{C}_1]_n \| [0]_{\lambda-n}$*. 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 3 is* (q_E, ε)*-collision-resistant, where*

$$\varepsilon \leq \frac{20q_E}{2^\lambda} + \frac{640{q_E}^2}{2^{2\lambda}} \tag{5}$$

The proof follows and generalizes [1, Theorem 3] (which was motivated by [47]). Roughly speaking, Com can be viewed as making $2\lceil \frac{\lambda}{n} \rceil$ Davies-Meyer

invocations with correlated inputs and concatenating their outputs (with some truncation when $n \nmid \lambda$). For the simpler case of $n \mid \lambda$, each Davies-Meyer invocation "contributes" a term of $O\left(\frac{1}{2^n}\right)$ to the collision probability, and the probability of collision among every pair of $\mathtt{com}$-evaluations accumulates to $O\left(\left(\frac{1}{2^n}\right)^{2\lambda/n}\right) = O\left(\frac{1}{2^{2\lambda}}\right)$. Taking a union bound over the $O({q_E}^2)$ pairs of $\mathtt{com}$-evaluations yields the bound. The full proof is available in Appendix C.

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

Lemma 4. *Assume that $n \le \lambda \le 2n$, and let $[\mathsf{C}_2]_\lambda = [\mathsf{C}_0]_n \| [0]_{\lambda-n}$ and $[\mathsf{C}_3]_\lambda = [\mathsf{C}_1]_n \| [0]_{\lambda-n}$. 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 3 is (q_E, ε)-collision-resistant, where*

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

$\mathsf{H}_{\mathsf{miTCCR}}$ 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 $\mathsf{H}_{\mathsf{miTCCR}}$ in Fig. 3 has $(q_E, \frac{80q_E}{2^\lambda} + \frac{10240{q_E}^2}{2^{2\lambda}} + \frac{160q_E}{2^{2\lambda}}, \frac{80q_E}{2^\lambda} + \frac{10240{q_E}^2}{2^{2\lambda}})$ extractable binding property in Definition 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. Let t be the tweak provided by $\mathcal{A}$ and define $t_0 := [\mathsf{C}_0]_8 \| t$, $t_1 := [\mathsf{C}_1]_8 \| t$. 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 $n < \lambda \le 2n$. We also define the constants $\Delta_k := [0]_{\lambda-8} \| ([\mathsf{C}_0]_8 \oplus [\mathsf{C}_1]_8)$, $\Delta_\mathsf{C} := [\mathsf{C}_0]_n \oplus [\mathsf{C}_1]_n$ and $\Delta_p := \sigma(\Delta_\mathsf{C})$.

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

$$\left\{(k = [\mathsf{C}_0]_8 \| t, x_0, y_0), (k, x_0 \oplus \Delta_p, y_1)\right\}.$$

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

$$\left\{(k_0 = x_R \| [\mathsf{C}_0]_8 \| t, x_0, y_0^0), (k_1 = k_0 \oplus \Delta_k, x_0, y_0^1), (k_0, x_1 = x_0 \oplus \Delta_p, y_1^0), (k_1, x_1, y_1^1)\right\}.$$

We define our extractor $\mathsf{Ext}^E(c, \mathcal{Q}_E)$ in Fig. 4. In the following we show that $\Pr[\mathbf{Exp}^{\mathrm{ebind}}_{\mathsf{H}^E_{\mathsf{miTCCR}}}(\mathcal{A}, \mathsf{Ext}) = \mathtt{fail}] \le \delta$ and $\Pr[\mathbf{Exp}^{\mathrm{ebind}}_{\mathsf{H}^E_{\mathsf{miTCCR}}}(\mathcal{A}, \mathsf{Ext}) = \mathtt{coll}] \le \varepsilon$ in the extractable binding game (Definition 3). Let m be the purported Com pre-image returned by $\mathcal{A}$. Recall that by definition,

- $\Pr[\mathbf{Exp}^{\mathrm{ebind}}_{\mathsf{H}^E_{\mathsf{miTCCR}}}(\mathcal{A}, \mathsf{Ext}) = \mathtt{fail}] = \Pr[\mathsf{Com}(m', t') = c \wedge m^* = \bot]$
- $\Pr[\mathbf{Exp}^{\mathrm{ebind}}_{\mathsf{H}^E_{\mathsf{miTCCR}}}(\mathcal{A}, \mathsf{Ext}) = \mathtt{coll}] = \Pr[\mathsf{Com}(m', t') = c \wedge m^* \neq \bot \wedge (m', t') \neq (m^*, t^*)]$

Input. The commitment $c \in \{0,1\}^{2\lambda}$ and the ideal cipher transcript $\mathcal{Q}_E$.
Output. A message m and a tweak t such that $\mathsf{Com}(m,t) = c$ or $\perp$.
Oracle. Ext has oracle access to E.

1. Ext initializes an empty set $\mathcal{Q}_P = \emptyset$.
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 $\mathcal{Q}_P$ contains all possible piles from $\mathcal{Q}_E$. Ext tries to find the pre-image of c from $\mathcal{Q}_P$ as follows.
   - If $\lambda = n$, then for each entry in $\mathcal{Q}_P$ of the form
   $$\big\{(k = [\mathsf{C}_0]_8 \| t, x_0, y_0), (k = [\mathsf{C}_1]_8 \| t, x_1 = x_0 \oplus \Delta_p, y_1)\big\}$$
   Ext defines $m_0 = \sigma^{-1}(x_0) \oplus [\mathsf{C}_0]_n$ and $m_1 = \sigma^{-1}(x_1) \oplus [\mathsf{C}_0]_n$. 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 $m^* = m_1$. Ext also defines $t^* = t$.
   - If $n < \lambda \leq 2n$, then for each pile in $\mathcal{Q}_P$ of the form
   $$\left\{\begin{matrix}(k_0 = x_R \| [\mathsf{C}_0]_8 \| t, x_0, y_0^0), & (k_1 = k_0 \oplus \Delta_k, x_0, y_0^1), \\ (k_0, x_1 = x_0 \oplus \Delta_p, y_1^0), & (k_1, x_1, y_1^1)\end{matrix}\right\}$$
   Ext defines $m_0 = \sigma^{-1}(x_0) \oplus [\mathsf{C}_0]_n \| x_R$ and $m_1 = \sigma^{-1}(x_1) \oplus [\mathsf{C}_0]_n \| x_R$. 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 $m^* = m_1$. Ext also defines $t^* = t$.
4. If (m^*, t^*) is undefined or multiply defined, then Ext outputs $\perp$. Otherwise, Ext outputs the unique (m^*, t^*) value.

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

- Case 1: $\lambda = n$
  1. If $k \neq t_0$ then remove (k,x,y) from $\mathcal{Q}_E$ and return $\emptyset$.
  2. Define $x' = x \oplus \Delta_p$ and check if there is an entry $(k, x', \star) \in \mathcal{Q}_E$.
     - If so, denote the entry as (k, x', y_1) and remove it from $\mathcal{Q}_E$.
     - Otherwise, query the oracle E to get $y_1 = E(k, x')$.
  3. Return $\{(k,x,y), (k,x',y_1)\}$.
- Case 2: $n < \lambda \leq 2n$
  1. If $\mathsf{right}_n(k) \notin \{t_0, t_1\}$ then remove (k,x,y) from $\mathcal{Q}_E$ and return $\emptyset$.
  2. Define $x' = x \oplus \Delta_p, k' = k \oplus \Delta_k$. For the i-th element $(\tilde{k}, \tilde{x}) \in ((k',x),(k,x'),(k',x'))$, check if there exists an entry $(\tilde{k}, \tilde{x}, \star) \in \mathcal{Q}_E$.
     - If so, denote the respective entries as $(\tilde{k}, \tilde{x}, y_i)$ and remove it from $\mathcal{Q}_E$.
     - Otherwise, query the oracle E to get $y_i = E(\tilde{k}, \tilde{x})$.
  3. Return $\{(k,x,y), (k',x,y_1), (k,x',y_2), (k',x',y_3)\}$.

Fig. 4: The extractor Ext^E for our proof of Lemma 5.

Regarding the probability of outputting `fail`, note that Ext outputs $m^* = \perp$ 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

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

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

$$\Pr\big[\mathsf{Multi}\big] \leq \frac{80q_E}{2^\lambda} + \frac{10240{q_E}^2}{2^{2\lambda}}.$$

Regarding the event $\mathsf{Com}(m,t) = c$, we show in Lemma 6 that

$$\Pr\big[\mathsf{Com}(m,t) = c \mid \mathsf{Null}\big] \leq \frac{160q_E}{2^{2\lambda}}.$$

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

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

Lemma 6. *Consider the extractable binding game in Definition 3. 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 (m', t') such that $\mathsf{Com}(m', t') = c$ is bounded by $\frac{160q_E}{2^{2\lambda}}$.*

We can prove this lemma using the same proof technique in Lemma 4, i.e., by grouping the queries into piles and differentiating between the inversion event happening in a normal query and in a super query. We defer the proof to Section D.

4.4 The Security of the $\mathsf{H}_{\mathsf{CCR}}$ Construction

The circular correlation robustness and extractable properties of the $\mathsf{H}_{\mathsf{CCR}}$ construction inherit from the respective properties of the $\mathsf{H}_{\mathsf{miTCCR}}$ construction in Section 4.1. Therefore, we only state the results here without repeating the proofs.

Lemma 7. *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, ε)-circular correlation robust, where*

$$\varepsilon \leq \frac{4q_E q_C}{2^\lambda} + \frac{q_C(q_C - 1)}{2^n}. \tag{7}$$

Lemma 8. $\mathsf{H}_{\mathsf{CCR}}$ *in Fig. 3 has* $(q_E, \frac{80q_E}{2^\lambda} + \frac{10240{q_E}^2}{2^{2\lambda}} + \frac{160q_E}{2^{2\lambda}}, \frac{80q_E}{2^\lambda} + \frac{10240{q_E}^2}{2^{2\lambda}})$ *extractable binding property in Definition 3.*

Remark 3. We note that our extractable CCR hash function in Fig. 3 generalizes the MMO-style CCR hash function in [37]. In particular, when $\lambda = n = 128$, our CCR construction is identical to $\mathsf{MMO}(x) := \pi(\sigma(x)) \oplus \sigma(x)$ if we fix $t = [0]_{120}$ and let $\pi(x) = E([\mathsf{C}_0]_8 \| [0]_{120}, x)$.

Moreover, since the ideal cipher's key is fixed to be $[\mathsf{C}_0]_8 \| [0]_{120}$, 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 [39]. 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 5.

Definition 5. *Let* $\mathsf{H}_{\mathsf{leaf}} : \{0,1\}^\lambda \times \{0,1\}^{n_t} \to \{0,1\}^\lambda \times \{0,1\}^{2\lambda}$ *be a function that maps the leaf node* $r \in \{0,1\}^\lambda$*, a tweak value* $t \in \{0,1\}^{n_t}$ *into a message* $m \in \{0,1\}^\lambda$ *and a commitment* $\mathsf{com} \in \{0,1\}^{2\lambda}$*.*

We recall the syntax of all-but-one vector commitment in Definition 6. We note that the "one-tree" optimization utilizes its batched version [7,8]. We choose to use the vanilla version as it can be straightforwardly generalized to the batched version, and it can be coupled with a wider range of protocols, e.g., the relaxed vector commitment in [46].

Definition 6. *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 constituting an AVC scheme.*

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

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 [39]. We define the required properties of the AVC scheme. In particular, we define the correctness property in Definition 7, the real-or-random hiding property in Definition 8, and the extractable binding property in Definition 9.

Protocol cGGM-based AVC

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

- $\mathsf{Setup}(1^\lambda, N) \to \mathsf{crs}$: Sample $t \leftarrow \{0,1\}^{n_t}$ as the initial tweak.
- $\mathsf{Commit}(\mathsf{crs}) \to (\mathtt{com}, \mathtt{decom}, (m_0, ..., m_{N-1}))$:
  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 $j \in [0, 2^{i-1})$:
     (a) Compute $r_{2j}^i = \mathsf{H}_{\mathsf{miTCCR}}(r_j^{i-1}, t)$,
     (b) Compute $r_{2j+1}^i = r_{2j}^i \oplus r_j^{i-1}$,
     (c) Update $t = t + 1$.
     Let $K_0^i = \sum_{j \in 2^{i-1}} r_{2j}^i$.
  4. In the d-th layer, for $i \in [0, 2^d)$:
     (a) Compute $(m_i, \mathtt{com}_i) = \mathsf{H}_{\mathsf{leaf}}(r_i^d, t)$,
     (b) Update $t = t + 1$.
  5. Compute $\mathtt{com} = \mathsf{H}_{\mathsf{RO}}(\mathtt{com}_0, ..., \mathtt{com}_{N-1})$.
  6. Output $(m_i)_{i \in [0,N)}$, $\mathtt{com}$, and $\mathtt{decom} = (\Gamma, \{K_0^i\}_{i \in [1,d]}, \{\mathtt{com}_i\}_{i \in [0,N)})$.
- $\mathsf{Open}(\mathsf{crs}, \mathtt{decom}, \alpha) \to \mathtt{decom}_\alpha$:
  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 α_i.
  2. Output the opening information as $\mathtt{decom}_\alpha = (\{K_0^i \oplus \bar{\alpha}_i \cdot \Gamma\}_{i \in [1,d]}, \mathtt{com}_\alpha)$.
- $\mathsf{Verify}(\mathsf{crs}, \mathtt{com}, \alpha, \mathtt{decom}_\alpha) \to \{(m_j)_{j \neq \alpha}\} \cup \{\perp\}$:
  1. Parse the decommitment information as $\mathtt{decom}_\alpha = (\{K_{\bar{\alpha}_i}^i\}_{i \in [1,d]}, \mathtt{com}_\alpha)$.
  2. Reconstruct the GGM tree using the information $\{K_{\bar{\alpha}_i}^i\}_{i \in [1,d]}$. Recover the last layer information as $(m_j, \mathtt{com}_j) = \mathsf{H}_{\mathsf{leaf}}(r_j^d)$ for $j \in [0, N), j \neq \alpha$.
  3. Re-compute the commitment as $\mathtt{com}' = \mathsf{H}_{\mathsf{RO}}(\mathtt{com}_0, ..., \mathtt{com}_{N-1})$ where $\mathtt{com}_\alpha$ is part of the decommitment information and the rest is generated in the previous step. Output $\perp$ if $\mathtt{com}' \neq \mathtt{com}$. Otherwise, output $(m_i)_{i \neq \alpha}$.

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

Definition 7 (Correctness). *An all-but-one vector commitment scheme* AVC *is (perfectly) correct if for all* $\lambda \in \mathbb{N}$ *and* $N = \mathsf{poly}(\lambda)$ *the following condition*

Algorithm $\mathsf{H}^E_{\mathsf{leaf}}(1^\lambda, r, t)$
// λ: security parameter;
// Note that $|r| = \lambda$.

1. Let $\mathsf{H}^E_{\mathsf{miTCCR}}$ be as defined in Fig. 3 and $\mathsf{Com}(r,t) = \mathsf{H}^E_{\mathsf{miTCCR}}(r \oplus [\mathsf{C}_2]_\lambda, t) \| \mathsf{H}^E_{\mathsf{miTCCR}}(r \oplus [\mathsf{C}_3]_\lambda, t)$
2. Let $m = \mathsf{H}^E_{\mathsf{miTCCR}}(r,t)$, $\mathtt{com} = \mathsf{Com}(r,t)$
3. Output $(m, \mathtt{com})$.

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

holds.

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

Definition 8 (Hiding). *Let* AVC *be an all-but-one vector commitment scheme following the syntax of Definition 6. The adaptive hiding experiment for* AVC *with* $N = 2^d = \mathsf{poly}(\lambda)$ *and stateful* $\mathcal{A}$ *is defined as follows.*

1. $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, N), b^* \leftarrow \{0,1\}$
2. $(\mathtt{com}, \mathtt{decom}, (m_0^*, ..., m_{N-1}^*)) \leftarrow \mathsf{Commit}(\mathsf{crs})$
3. $\alpha \leftarrow \mathcal{A}^{E, \mathsf{H}_{\mathsf{RO}}}(1^\lambda, \mathsf{crs}, \mathtt{com})$
4. $\mathtt{decom}_\alpha \leftarrow \mathsf{Open}(\mathsf{crs}, \mathtt{decom}, \alpha)$
5. *Let* $m_i = m_i^*$ *for* $i \in [0, N), i \neq \alpha$
6. *For* $i = \alpha$, *sample* $m_i \xleftarrow{\$} \{0,1\}^\lambda$ *and set* $m_i = \begin{cases} m_i^* & \text{if } b^* = 0 \\ m_i & \text{if } b^* = 1 \end{cases}$
7. $b \leftarrow \mathcal{A}^{E, \mathsf{H}_{\mathsf{RO}}}((m_i)_{i \in [0,N)}, \mathtt{decom}_\alpha)$
8. *Output 1 (success) if* $b = b^*$, *else 0 (failure).*

In the selective hiding experiment, $\mathcal{A}$ *must choose* α *prior to receiving* com.

The advantage $\mathsf{AdvSelHide}^{\mathsf{AVC}}(\mathcal{A})$ *(resp.* $\mathsf{AdvAdpHide}^{\mathsf{AVC}}(\mathcal{A})$*) 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 9 (Binding). *Let* AVC *be an all-but-one vector commitment scheme following the syntax of Definition 6. Let* $\mathsf{Ext}(\mathsf{crs}, \mathtt{com}, Q_E) \to (m_i)_{i \in [0,N)}$ *be an extraction function.*

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

– $(\mathsf{crs}, \mathtt{com}, \mathcal{Q}_E, \mathcal{Q}_{\mathsf{H}_{\mathsf{RO}}}) \leftarrow \mathcal{A}^{E, \mathsf{H}_{\mathsf{RO}}}(1^\lambda)$
// $\mathcal{Q}_E$ *and* $\mathcal{Q}_{\mathsf{H}_{\mathsf{RO}}}$ *are the transcripts of* E *and* H_{RO} *respectively*

– $(m_i^*)_{i\in[0,N)} = \mathsf{Ext}(\mathsf{crs}, \mathtt{com}, \mathcal{Q}_E, \mathcal{Q}_{\mathsf{H_{RO}}})$
– $(\mathsf{crs}', \mathtt{decom}_\alpha, \alpha) \leftarrow \mathcal{A}^{E,\mathsf{H_{RO}}}(1^\lambda)$
– *Output 1 (success) if* $\mathsf{Verify}(\mathsf{crs}', \mathtt{com}, \alpha, \mathtt{decom}_\alpha) = (m_i)_{i\in[0,N),i\neq\alpha}$ *but* $m_i \neq m_i^*$ *for some* $i \in [0, N), i \neq \alpha$. *Otherwise, output 0 (failure).*

Let $\mathsf{AdvEB}^{\mathsf{AVC}}(\mathcal{A}) = \Pr[\mathcal{A}$ *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 adaptive hiding property in Definition 8.

Lemma 9. *Let* E *be the ideal cipher oracle,* $\mathsf{H}^E_{\mathsf{miTCCR}}$ *be a* $(q_E, q_C, u, \mu, \lambda, \varepsilon)$*-circular correlation robust hash function where* $q_C = d + 3$, $u = 1$, $\mu = 3$, *and* $\mathsf{H}_{\mathsf{leaf}}$ *be defined as in Fig. 6. The AVC scheme in Fig. 5 with* $N = 2^d$ *satisfies* $\mathsf{AdvAdpHide}^{\mathsf{AVC}}(\mathcal{A}) \leq 2\varepsilon + \frac{q_H}{2^{2\lambda}}$ *for any adversary* $\mathcal{A}$ *making* q_E *queries to* E *and* q_H *queries to the random oracle* $\mathsf{H_{RO}}$.

Proof. We can express the advantage of the adversary in terms of the difference between the adversary outputting 1 in the $b^* = 0$ case and the $b^* = 1$ case.

$$\left|\Pr[b = b*] - \frac{1}{2}\right| = \frac{1}{2}\left|\Pr[b = 1 \mid b^* = 1] - \Pr[b = 1 \mid b^* = 0]\right| \tag{8}$$

We construct three hybrid experiments to upper-bound Eq. (8).

– $\mathbf{Hybrid_1}$. The adversary is given the real commitment as well as the partial opening and the messages. This corresponds to the $b^* = 0$ case.
– $\mathbf{Hybrid_2}$. We abort the experiment if the adversary queries the pre-image of $\mathtt{com}$ before getting $\mathtt{decom}_\alpha$ in Step 4 of the experiment.
– $\mathbf{Hybrid_3}$. The challenger samples $\mathtt{com} \leftarrow \{0,1\}^{2\lambda}$ in the beginning of the experiment and programs $\mathsf{H_{RO}}(\mathtt{com}_0, ..., \mathtt{com}_{N-1}) = \mathtt{com}$.
– $\mathbf{Hybrid_4}$. The challenger generates $(m_i)_{i\in[0,N)}, \mathtt{decom}_\alpha$ using the following procedure.
  • First sample $\mathtt{decom}_\alpha, \mathtt{com}_\alpha, m_\alpha$ uniformly at random.
  • Then compute $\{m_i\}_{i\neq\alpha}$ and $\{\mathtt{com}_i\}_{i\neq\alpha}$ using the Verify algorithm.
– $\mathbf{Hybrid_5}$. The challenger generates $(m_i)_{i\neq\alpha}, \mathtt{decom}_\alpha$ using the algorithm Commit but sample m_α uniformly at random.
– $\mathbf{Hybrid_6}$. We generate $\mathtt{com}$ using the algorithm Commit.
– $\mathbf{Hybrid_7}$. We remove the abort condition introduced in $\mathbf{Hybrid_2}$. This hybrid experiment corresponds to the $b^* = 1$ case.

We derive the final result via a series of claims.

Claim. The difference between $\mathbf{Hybrid_1}$ and $\mathbf{Hybrid_2}$ as well as the difference between $\mathbf{Hybrid_6}$ and $\mathbf{Hybrid_7}$ is bounded by $\varepsilon + \frac{q_H}{2^{2\lambda}}$.

Proof. We prove that the abort event happens with probability at most $\varepsilon + \frac{q_H}{2^{2\lambda}}$, thus proving the claim. The preimage of $\mathtt{com}$ is $(\mathtt{com}_0, ..., \mathtt{com}_{N-1})$. As shown below, the adversary's view can be simulated using the construction oracle $\mathcal{O}^{\mathsf{miTCCR}}$. Let p be the probability of the abort event. The miTCCR property of the hash function ensures that by switching the construction oracle to a random function, the probability of the abort event is at least $p - \varepsilon$. However, by switching the construction oracle to a random function, the value $\mathtt{com}_\alpha$ is uniformly random over $\{0,1\}^{2\lambda}$. Therefore, we have $p - \varepsilon \leq \frac{q_H}{2^{2\lambda}}$, implying our claim.

Claim. The difference between $\mathbf{Hybrid_2}$ and $\mathbf{Hybrid_3}$ as well as the difference between $\mathbf{Hybrid_5}$ and $\mathbf{Hybrid_6}$ is zero.

Proof. In both switches between hybrid experiments, we ensure that the adversary cannot query the pre-image of $\mathtt{com}$ prior to learning the opening information. Therefore, sampling $\mathtt{com}$ either during the Com procedure or separately does not change its distribution.

Claim. The difference between $\mathbf{Hybrid_3}$ and $\mathbf{Hybrid_4}$ is bounded by ε.

Proof. Let $\mathcal{A}$ be an adversary that distinguishes $\mathbf{Hybrid_3}$ and $\mathbf{Hybrid_4}$. We construct a distinguisher $\mathcal{B}$ of the miTCCR game with the same advantage. In this way, we can prove that the difference between $\mathbf{Hybrid_3}$ and $\mathbf{Hybrid_4}$ is upper-bounded by ε.

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

1. First sample $r^1_{\bar{\alpha}_1} \leftarrow \{0,1\}^\lambda$, $t \leftarrow \{0,1\}^{n_t}$, and set $s_1 = r^1_{\bar{\alpha}_1}, K^1 = r^1_{\bar{\alpha}_1}$
2. For $i \in [2, d]$,
   (a) For $j \in [0, 2^{i-1})$:
      i. If $j = \alpha_1 \| ... \| \alpha_{i-1}$, then compute $r^i_{j^*+\bar{\alpha}_i} = \mathcal{O}^{\mathsf{miTCCR}}(s_{i-1}, t, \bar{\alpha}_i) \oplus \bar{\alpha}_i s_{i-1}$ and update $t = t + 1$.
      ii. Otherwise, compute $r^i_{2j} = \mathsf{H}_{\mathsf{miTCCR}}(r^i_j, t)$, $r^i_{2j+1} = r^i_{2j} \oplus r^i_j$, and update $t = t + 1$.
   (b) Compute $K^i = \sum_{j \in [0, 2^{i-1})} r^i_{2j+\bar{\alpha}_i}$.
   (c) Update $s_i = s_{i-1} \oplus r^i_{j^*+\bar{\alpha}_i}$
3. For $j \in [0, 2^d)$:
   (a) If $j \neq \alpha$, compute $m_j = \mathsf{H}_{\mathsf{miTCCR}}(r^d_j, t)$ and $\mathtt{com}_j = \mathsf{Com}(r^d, t)$, and update $t = t + 1$.
   (b) Otherwise, compute $m_\alpha = \mathcal{O}^{\mathsf{miTCCR}}(s_d, t, 0)$ and $\mathtt{com}_\alpha = \mathcal{O}^{\mathsf{miTCCR}}(s^d \oplus [\mathsf{C}_2]_\lambda, t, 0) \| \mathcal{O}^{\mathsf{CCR}}(s^d \oplus [\mathsf{C}_3]_\lambda, t, 0)$.
4. Compute $\mathtt{com} = \mathsf{H}_{\mathsf{RO}}(\mathtt{com}_0, ..., \mathtt{com}_{N-1})$.
5. Output $\mathtt{com}$, $\{m_i\}_{i \in [0, 2^d)}$, and $\mathtt{decom}_\alpha = \{K^i\}_{i \in [1,d]}$

Notice that if $\mathcal{O}^{\mathsf{miTCCR}}$ returns the real correlated randomness then the view of $\mathcal{A}$ is identical to $\mathbf{Hybrid_3}$, whereas if $\mathcal{O}^{\mathsf{miTCCR}}$ returns uniformly random values then the view of $\mathcal{A}$ is identical to $\mathbf{Hybrid_4}$. Moreover, the construction oracle involves only a single key Γ which follows the uniform distribution over $\{0,1\}^\lambda$. The reduction algorithm $\mathcal{B}$ only invokes the construction oracle for $q_C = d + 3$ times with a maximum of 3 calls on a single tweak. Therefore, the advantage of $\mathcal{A}$ is upper-bounded by ε.

Claim. The difference between $\mathbf{Hybrid_4}$ and $\mathbf{Hybrid_5}$ is bounded by ε.

Proof. The argument is almost identical to the previous claim, except that we always sample m_α uniformly at random in the two hybrids. Therefore, the advantage of the adversary in distinguishing the two hybrids is also bounded by ε.

Combining the above claims, we conclude that in Eq. (8) we have

$$|\Pr[b = 1 \mid b^* = 1] - \Pr[b = 1 \mid b^* = 0]| \leq 4\varepsilon + 2 \cdot \frac{q_H}{2^{2\lambda}} \tag{9}$$

and therefore

$$\mathsf{AdvAdpHide}^{\mathsf{AVC}}(\mathcal{A}) = \left| \Pr[b = b^*] - \frac{1}{2} \right| \leq 2\varepsilon + \frac{q_H}{2^{2\lambda}} \tag{10}$$

□

5.2 The Extractable Binding Property of the AVC Scheme

Lemma 10. *Let* $\mathsf{H}_{\mathsf{miTCCR}}$ *be a* $(q_E, \varepsilon_{\mathsf{EB}}, \delta_{\mathsf{EB}})$*-extractable binding* miTCCR *function in the ideal cipher model and* H_{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 $\mathcal{Q}_{\mathsf{H}_{\mathsf{RO}}}$ and looks for the pre-image of $\mathtt{com}$. If none can be found or the pre-image is not unique, then it outputs $\perp$.

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

Now we bound the probability $\mathsf{AdvEB}^{\mathsf{AVC}}(\mathcal{A})$. Since H_{RO} is a random oracle, the probability of the extractor outputting $\perp$ due to finding a collision is bounded by $\frac{1}{2^{2\lambda}} + \frac{2}{2^{2\lambda}} + \cdots + \frac{q_H-1}{2^{2\lambda}} = \frac{q_H(q_H-1)}{2\cdot 2^{2\lambda}}$. The probability of not being able to find a pre-image is bounded by $\frac{1}{2^{2\lambda}}$. Therefore, the extraction of $\mathtt{com}$ 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 $\mathsf{Ext}(\mathtt{com}_i, \mathcal{Q}_E)$, which is bounded by $\delta_{\mathsf{EB}} + \varepsilon_{\mathsf{EB}}$. Therefore, by taking a union-bound on all the leaf nodes, we conclude that

$$\mathsf{AdvEB}^{\mathsf{AVC}}(\mathcal{A}) \leq \frac{1 + q_H(q_H - 1)/2}{2^{2\lambda}} + N \cdot (\delta_{\mathsf{EB}} + \varepsilon_{\mathsf{EB}}). \tag{11}$$

□

6 High Security Half-Gate from AES

In [36], Guo et al. showed that the concrete security of half-gate garbling scheme is insufficient without incorporating a tweak in the hash input, motivating the use of miTCCR hash functions. In particular, Guo et al. derived concrete security bounds of half-gate garbling from the quantified miTCCR hash security. Focusing on constructions from standard AES, the concrete security in [36] is limited to 128 bits.

Building from the new miTCCR construction in Section 4, we can extend the result in [36] and push the concrete security of half-gate beyond 128 bits. Due to space limit and since the security theorems in [36] utilizes miTCCR hash in a black-box fashion, we defer the definition of the garbling scheme and the half-gates construction to sec-garbling-definition, and only state the composed result in this section.

Corollary 1. *Let E be an ideal cipher with key length λ and block length n satisfying $n \le \lambda \le 2n$. Then for any integer $\mu \ge 1$, the garbled circuit scheme* $\mathsf{HalfGates}^{\mathsf{H}}$ *is*

– (q_E, u, C, ϵ')*-private where*

$$\epsilon' \le \frac{4q_E\mu}{2^{\lambda-1}} + \frac{(2C+n_{\mathsf{out}})(\mu-1)}{2^n} + \frac{(2C+n_{\mathsf{out}})^{\mu+1}}{(\mu+1)!\cdot 2^{(n-8)\mu}}$$

– (q_E, u, C, ϵ')*-oblivious where*

$$\epsilon' \le \frac{4q_E\mu}{2^{\lambda-1}} + \frac{(2C+n_{\mathsf{out}})(\mu-1)}{2^n} + \frac{(2C+n_{\mathsf{out}})^{\mu+1}}{(\mu+1)!\cdot 2^{(n-8)\mu}}$$

– $(q_E, u, C, n_{\mathsf{out}}, \epsilon')$*-authentic where*

$$\epsilon' \le \frac{4q_E\mu}{2^{\lambda-1}} + \frac{(2C+n_{\mathsf{out}})(\mu-1)}{2^n} + \frac{(2C+n_{\mathsf{out}})^{\mu+1}}{(\mu+1)!\cdot 2^{(n-8)\mu}} + \frac{1}{2^\lambda}$$

Since the above bounds hold for any integer $\mu \ge 1$, it is possible to find an optimal μ. However, setting $\mu = n$ suffices for a good bound. Concretely, consider the privacy bound: when $\mu = n$, the first two terms become $\frac{4nq_E}{2^{\lambda-1}} + \frac{(2C+n_{\mathsf{out}})(n-1)}{2^n}$, while the third term becomes

$$\frac{(2C+n_{\mathsf{out}})^{n+1}}{(n+1)!\cdot 2^{(n-8)n}} \le \frac{1}{(n+1)!}\cdot\left(\frac{2(2C+n_{\mathsf{out}})}{2^{n-8}}\right)^{n+1},$$

which is bounded by $\frac{2(2C+n_{\mathsf{out}})}{(n+1)!\cdot 2^{n-8}}$ when $2(2C+n_{\mathsf{out}}) \le 2^{n-8}$ (which likely holds in practice). The computational security is thus determined by the term $\frac{4nq_E}{2^{\lambda-1}}$. Therefore, using our new hash function, the half-gate scheme ensures $\lambda - 3 - \log_2 n$ bit computational security. When $\lambda = 256$ and $n = 128$, the concrete security is 246 bits.

7 Performance Evaluation

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

- RQ1: For security parameters beyond 128 bits, how does the performance of our AES-based miTCCR hash function compare with previous MMO-style construction [37,36] instantiated with the non-standardized Rijndael variants?
- RQ2: How does our miTCCR-based all-but-one vector commitment (AVC) compare against the AES-CTR-based baseline in [8]?

For the first question, we instantiated the MMO-style miTCCR hash function [36] with the Rijndael encryption scheme [26] where the block size equals the security parameter and compared our AES-based 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 miTCCR hash function appeared previously in the literature [21]. Finally, due to the structural similarity, all Rijndael parameter sets could to some extent utilize the AES-NI and VAES instruction sets (which are designed specifically for AES), therefore offering a fairer comparison in terms of performance. As for the second question, we chose the optimized implementation of the 2nd version of the FAEST signature scheme as baseline and tested the performance gain of the batched all-but-one vector commitment component as well as the end-to-end sign/verify performance, when using the half-tree-based AVC construction in Section 5.

We implemented our AES-based miTCCR hash function as well as the Rijndael-based MMO-style miTCCR hash function [36] using the VAES instruction set. We used the XKCP implementation [17] for the vectorized implementation of SHA3 baseline. We implement the SHA-256 function using the SHA-NI instruction set. Our code is available at the anonymized repository `https://anonymous.4open.science/r/mitccr-artifact-776E`. We conducted our experiments on a machine running Ubuntu 26.04 LTS with an AMD Ryzen 9 9955HX3D 16-Core Processor (16 cores and 32 hardware threads) and 30 GiB RAM. The CPU supports VAES, SHA-NI, and AVX-512 instructions. The compiler is GCC 15.2.0, and the benchmark code was compiled in release mode with '-O3' and '-march=native' enabled. We used CMake 4.2.3 and OpenSSL 3.5.5.

7.1 Performance of the AES-based miTCCR Hash Functions

We benchmarked batched miTCCR hashing with batch sizes $1, 2, 4, 8$, reporting average cycles per hash. The SHA-256 and SHA3 baselines hash the tweak and input together and truncate the digest to the desired output length, where we implement the SHA-256 function using the SHA-NI instruction set. For SHA3,

we use the vectorized SHAKE128 (resp. SHAKE256) implementation for $\lambda = 128$ (resp. $\lambda = 192, 256$) from XKCP [17], which internally uses the AVX2 instruction set. For the miTCCR construction in [36] instantiated with Rijndael, we use the implementation in emp-toolkit [55] for $\lambda = 128$ and implement the $\lambda = 192, 256$ cases by ourselves. Results are shown in Table 1.

Overall trends. All VAES-accelerated constructions benefit substantially from batching. Our AES-based construction achieves 4.2× speedup from batch 1 to batch 8 at $\lambda = 128$ (24.78 to 5.88 cycles), and 2.8× speedup at $\lambda = 192, 256$ (30.31 to 10.85, 33.28 to 11.81). Our Rijndael-based construction also has 5.3× and 6.3× gains at $\lambda = 192, 256$ (121.11 to 23.02, 118.19 to 18.79). The Rijndael-based construction in emp-tool [55], which only offers the 128-bit variant, follows a different implementation strategy, and has limited speed-up at the batched setting. SHA-256, accelerated by SHA-NI, is essentially flat across batch sizes (219–223 at batch 1 to 191–195 at batch 8), as SHA-NI does not offer the same kind of data-parallelism as VAES. SHA3, using the XKCP implementation with AVX2, drops steeply from 600–604 cycles at batch 1 to 65–67 cycles at batch 8, but even at its best remains an order of magnitude above the AES-based constructions.

$\lambda = 128$. At batch 1, the Rijndael-based construction is faster than ours (15.69 vs. 24.78 cycles), due to implementation differences. However, from batch 2 onward our construction overtakes (12.75 vs. 13.27) and the gap widens at batch 4 (6.76 vs. 11.63, 1.7×) and batch 8 (5.88 vs. 11.09, 1.9×), as the VAES-native 128-bit AES layout amortizes the per-instance overhead more effectively than the Rijndael layout. At batch 8, our construction is 33× and 11× faster than SHA-256 and SHA3, respectively.

$\lambda = 192, 256$. Our construction outperforms the Rijndael-based one at every batch size. At batch 1, the gap is 4.0× (30.31 vs. 121.11) for $\lambda = 192$ and 3.6× (33.28 vs. 118.19) for $\lambda = 256$. The Rijndael cipher with 192/256-bit blocks operates on wider internal states that do not align well with the 128-bit VAES lanes, incurring significant data-rearrangement overhead. At batch 8, our advantage narrows to 2.1× (10.85 vs. 23.02) at $\lambda = 192$ and 1.6× (11.81 vs. 18.79) at $\lambda = 256$, as the Rijndael implementations' larger batching gains partially compensate, but our construction remains clearly ahead. At batch 8, our construction is 18×/17× faster than SHA-256 and 6× faster than SHA3 for the 192/256-bit settings.

Summary. The batched results confirm that our AES-based miTCCR construction benefits decisively from VAES parallelism and is the fastest option across all security parameters at batch sizes ≥ 2, while remaining competitive even at batch 1 for the 128-bit case.

7.2 Performance of the miTCCR-based BAVC in FAEST

We integrated the batched version of the miTCCR-based AVC scheme from Section 5 into the FAEST signature scheme [8] and measured the performance gain against the original architecture-optimized implementation [32]. We test both the FAEST (which uses a Naor-style leaf commitment) and FAEST-EM

Table 1: The batched performance of hash functions for different security parameters. The running time is measured in CPU cycles per hash.

Construction	Security	Batch 1	Batch 2	Batch 4	Batch 8
Rijndael-based [36]	128	15.69	13.27	11.63	11.09
SHA-256-based		223.46	199.86	191.51	194.45
SHA3-based		600.33	208.37	124.49	65.80
Ours (AES-based)		24.78	12.75	6.76	5.88
Rijndael-based [36]	192	121.11	61.54	30.84	23.02
SHA-256-based		219.32	201.38	192.43	194.90
SHA3-based		602.17	209.61	128.23	66.51
Ours (AES-based)		30.31	16.43	14.87	10.85
Rijndael-based [36]	256	118.19	59.31	30.00	18.79
SHA-256-based		220.54	199.50	193.44	194.91
SHA3-based		604.12	210.17	130.66	67.18
Ours (AES-based)		33.28	17.98	14.86	11.81

(which uses AES-CTR as its leaf commitment) variants. The experiments cover all three security parameters ($\lambda = 128, 192, 256$) and both the "fast" (f) and "small" (s) parameter sets. All measurements are mean wall time in milliseconds, collected via Catch2 over at least 100,000 iterations.

BAVC Component Performance. Table 2 reports the commit and verification time of the BAVC component. For FAEST, the miTCCR-based BAVC reduces commit time by 29.3% $\sim$ 68.0% and verify time by 29.8% $\sim$ 66.6%. For FAEST-EM, the reduction is 4.0% $\sim$ 41.7% for commit and 3.7% $\sim$ 39.8% for verify.

End-to-End Signature Performance. Table 3 reports the end-to-end signing and verification time. For FAEST, replacing the BAVC component with the TCCR-based construction reduces signing time by 13.1% $\sim$ 40.4% and verification time by 15.2% $\sim$ 46.4%, with a slight reduction in signature size (e.g., 4506 B to 4330 B for 128s) due to reducing the leaf commitment size from 3λ bits to 2λ bits. For FAEST-EM, the reduction is 1.5% $\sim$ 25.0% for signing and 2.9% $\sim$ 24.2% for verification, with unchanged signature size.

Acknowledgements. The authors used Codex and Claude Code to help develop the implementation codes in Section 7. We thank the anonymous reviewers of Asiacrypt 2025 for their valuable suggestions.

References

1. 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 (May 2014). https://doi.org/10.1007/978-3-319-06734-6_26

Table 2: The runtime comparison of the original BAVC component in FAEST v2 vs. our construction. The percentage in the parenthesis shows reduction compared to the original scheme.

Scheme	Instance	Commit (ms)		Verify (ms)	
		Original	Ours	Original	Ours
FAEST	128s	1.048	0.719 (31.39%)	1.074	0.748 (30.35%)
	128f	0.099	0.070 (29.29%)	0.104	0.073 (29.81%)
	192s	5.429	1.789 (67.05%)	5.537	1.847 (66.64%)
	192f	0.677	0.217 (67.95%)	0.691	0.231 (66.57%)
	256s	5.536	2.960 (46.53%)	5.726	3.047 (46.79%)
	256f	0.582	0.297 (48.97%)	0.590	0.309 (47.63%)
FAEST-EM	128s	0.749	0.719 (4.01%)	0.777	0.748 (3.73%)
	128f	0.074	0.070 (5.41%)	0.079	0.073 (7.59%)
	192s	2.988	1.789 (40.13%)	3.057	1.847 (39.58%)
	192f	0.372	0.217 (41.67%)	0.384	0.231 (39.84%)
	256s	3.455	2.960 (14.33%)	3.584	3.047 (14.98%)
	256f	0.364	0.297 (18.41%)	0.376	0.309 (17.82%)

Table 3: End-to-end performance of FAEST v2 between the original version and the modified scheme using our miTCCR-based BAVC. The percentage in the parenthesis shows reduction compared to the original scheme.

Scheme	Instance	Sign (ms)		Verify (ms)		Size (B)	
		Original	Ours	Original	Ours	Original	Ours
FAEST	128s	1.869	1.578 (15.57%)	1.525	1.193 (21.77%)	4506	4330 (3.91%)
	128f	0.244	0.212 (13.11%)	0.204	0.173 (15.20%)	5924	5668 (4.32%)
	192s	9.315	5.548 (40.44%)	7.894	4.229 (46.43%)	11260	10876 (3.41%)
	192f	1.531	1.059 (30.83%)	1.361	0.897 (34.09%)	14948	14372 (3.85%)
	256s	9.233	6.542 (29.14%)	9.081	6.477 (28.68%)	20696	19992 (3.40%)
	256f	1.394	1.110 (20.37%)	1.306	1.006 (22.97%)	26548	25524 (3.86%)
FAEST-EM	128s	1.393	1.338 (3.95%)	1.156	1.122 (2.94%)	3906	3906
	128f	0.199	0.196 (1.51%)	0.165	0.159 (3.64%)	5060	5060
	192s	6.477	4.860 (24.97%)	6.027	4.567 (24.22%)	9340	9340
	192f	1.039	0.886 (14.73%)	0.919	0.763 (16.97%)	12380	12380
	256s	7.061	6.475 (8.30%)	6.739	6.169 (8.46%)	17984	17984
	256f	1.132	1.069 (5.57%)	1.025	0.954 (6.93%)	23476	23476

2. Aguilar Melchor, C., Bettaieb, S., Bidoux, L., Feneuil, T., Gaborit, P., Gama, N., Gueron, S., Howe, J., Hülsing, A., Joseph, D., Joux, A., Kulkarni, M., Persichetti, E., Randrianarisoa, T.H., Rivain, M., Yue, D.: SDitH — Syndrome Decoding in the Head. Tech. rep., National Institute of Standards and Technology (2024), available at https://csrc.nist.gov/Projects/pqc-dig-sig/

round-2-additional-signatures
3. 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
4. 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 (Dec 2011). https://doi.org/10.1007/978-3-642-25385-0_13
5. Asharov, G., Lindell, Y., Schneider, T., Zohner, M.: More efficient oblivious transfer and extensions for faster secure computation. In: Sadeghi, A.R., Gligor, V.D., Yung, M. (eds.) ACM CCS 2013. pp. 535–548. ACM Press (Nov 2013). https://doi.org/10.1145/2508859.2516738
6. Baum, C., Beullens, W., Braun, L., de Saint Guilhem, C.D., Klooß, M., Majenz, C., Mukherjee, S., Orsini, E., Ramacher, S., Rechberger, C., et al.: Faest v2: Algorithm specifications (2025)
7. 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. In: ASIACRYPT 2024, Part I. pp. 463–493. LNCS (Dec 2024). https://doi.org/10.1007/978-981-96-0875-1_15
8. Baum, C., Braun, L., Beullens, W., Delpech de Saint Guilhem, C., 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 (2024), available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures
9. Baum, C., Braun, L., Delpech de Saint Guilhem, C., 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
10. 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 (Aug 2023). https://doi.org/10.1007/978-3-031-38554-4_19
11. Baum, C., Braun, L., Munch-Hansen, A., Scholl, P.: Moz$\mathbb{Z}_{2^k}$arella: Efficient vector-OLE and zero-knowledge proofs over $\mathbb{Z}_{2^k}$. In: CRYPTO 2022, Part IV. pp. 329–358. LNCS (Aug 2022). https://doi.org/10.1007/978-3-031-15985-5_12
12. Baum, C., Dittmer, S., Scholl, P., Wang, X.: Sok: vector OLE-based zero-knowledge protocols **91**(11), 3527–3561 (2023). https://doi.org/10.1007/s10623-023-01292-8
13. 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. Virtual Event (Aug 2021). https://doi.org/10.1007/978-3-030-84259-8_4
14. 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

15. Bellare, M., Hoang, V.T., Rogaway, P.: Foundations of garbled circuits. In: Yu, T., Danezis, G., Gligor, V.D. (eds.) ACM CCS 2012. pp. 784–796. ACM Press (Oct 2012). https://doi.org/10.1145/2382196.2382279
16. Berti, F., Guo, C., Pereira, O., Peters, T., Standaert, F.X.: TEDT: a leakage-resistant 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
17. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V., Keer, R.V.: XKCP: eXtended Keccak Code Package. https://github.com/XKCP/XKCP
18. 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 (Dec 2012). https://doi.org/10.1007/978-3-642-34961-4_14
19. 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 (Aug 2022). https://doi.org/10.1007/978-3-031-15979-4_21
20. 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
21. Bui, D., Cong, K., Delpech de Saint Guilhem, C.: Faster VOLEitH signatures from all-but-one vector commitment and half-tree. pp. 205–223. LNCS (Dec 2025). https://doi.org/10.1007/978-981-96-9095-4_11
22. 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
23. 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 (May 2014). https://doi.org/10.1007/978-3-642-55220-5_19
24. 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 (Dec 2021). https://doi.org/10.1007/978-3-030-92075-3_10
25. Cui, H., Wang, X., Yang, K., Yu, Y.: Actively secure half-gates with minimum overhead under duplex networks. In: EUROCRYPT 2023, Part II. pp. 35–67. LNCS (Jun 2023). https://doi.org/10.1007/978-3-031-30617-4_2
26. Daemen, J., Rijmen, V.: The block cipher rijndael. In: International Conference on Smart Card Research and Advanced Applications. pp. 277–284. Springer (1998)
27. 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
28. Dittmer, S., Ishai, Y., Lu, S., Ostrovsky, R.: Authenticated garbling from simple correlations. In: CRYPTO 2022, Part IV. pp. 57–87. LNCS (Aug 2022). https://doi.org/10.1007/978-3-031-15985-5_3
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

30. 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
31. Doerner, J., shelat, a.: Scaling ORAM for secure computation. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017. pp. 523–535. ACM Press (Oct / Nov 2017). https://doi.org/10.1145/3133956.3133967
32. FAEST Team: FAEST architecture-optimized implementation. https://github.com/faest-sign/faest-arch-opt
33. Franzese, N., Katz, J., Lu, S., Ostrovsky, R., Wang, X., Weng, C.: Constant-overhead 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
34. Gilboa, N., Ishai, Y.: Distributed point functions and their applications. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 640–658 (May 2014). https://doi.org/10.1007/978-3-642-55220-5_35
35. 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
36. Guo, C., Katz, J., Wang, X., Weng, C., Yu, Y.: Better concrete security for half-gates garbling (in the multi-instance setting). In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part II. LNCS, vol. 12171, pp. 793–822 (Aug 2020). https://doi.org/10.1007/978-3-030-56880-1_28
37. 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
38. 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
39. 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 (Jun 2023). https://doi.org/10.1007/978-3-031-30545-0_12
40. Gupta, K., Kumaraswamy, D., Chandran, N., Gupta, D.: LLAMA: A low latency math library for secure inference. PoPETs **2022**(4), 274–294 (Oct 2022). https://doi.org/10.56553/popets-2022-0109
41. 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
42. Ishai, Y., Kilian, J., Nissim, K., Petrank, E.: Extending oblivious transfers efficiently. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 145–161 (Aug 2003). https://doi.org/10.1007/978-3-540-45146-4_9
43. 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
44. Katz, J., Ranellucci, S., Rosulek, M., Wang, X.: Optimizing authenticated garbling for faster secure two-party computation. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part III. LNCS, vol. 10993, pp. 365–391 (Aug 2018). https://doi.org/10.1007/978-3-319-96878-0_13

45. Keller, M., Orsini, E., Scholl, P.: MASCOT: Faster malicious arithmetic secure computation with oblivious transfer. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) ACM CCS 2016. pp. 830–842. ACM Press (Oct 2016). https://doi.org/10.1145/2976749.2978357
46. Kim, S., Lee, B., Son, M.: Relaxed vector commitment for shorter signatures. pp. 427–455. LNCS (Jun 2025). https://doi.org/10.1007/978-3-031-91134-7_15
47. 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
48. Lin, F., Xing, C., Yao, Y.: More efficient zero-knowledge protocols over $\mathbb{Z}_{2^k}$ via galois rings. Cryptology ePrint Archive, Paper 2023/150, To appear at CRYPTO 2024 (2023), https://eprint.iacr.org/2023/150
49. 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 (Aug 2009). https://doi.org/10.1007/978-3-642-04159-4_21
50. Rambus: AES-IP-36 AES ECB/CBC/CTR accelerators, https://www.rambus.com/security/crypto-accelerator-cores/aes-ip-36/
51. Reuse, D..: AES-DPA-FIA SW library - Advanced DPA- and FIA-resistant FortiCrypt AES SW library, https://www.design-reuse.com/sip/advanced-dpa-and-fia-resistant-forticrypt-aes-sw-library-ip-52392/
52. Reuse, D..: CryptoCORE 400G - 400G AES Encryption Core, https://www.design-reuse.com/sip/400g-aes-encryption-core-ip-46356/
53. Rosulek, M., Roy, L.: Three halves make a whole? Beating the half-gates lower bound for garbled circuits. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part I. LNCS, vol. 12825, pp. 94–124. Virtual Event (Aug 2021). https://doi.org/10.1007/978-3-030-84242-0_5
54. Roy, L.: SoftSpokenOT: Quieter OT extension from small-field silent VOLE in the minicrypt model. In: CRYPTO 2022, Part I. pp. 657–687. LNCS (Aug 2022). https://doi.org/10.1007/978-3-031-15802-5_23
55. Wang, X., Malozemoff, A.J., Katz, J.: EMP-toolkit: Efficient MultiParty computation toolkit. https://github.com/emp-toolkit (2016)
56. Wang, X., Ranellucci, S., Katz, J.: Authenticated garbling and efficient maliciously secure two-party computation. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017. pp. 21–37. ACM Press (Oct / Nov 2017). https://doi.org/10.1145/3133956.3134053
57. Wang, X., Ranellucci, S., Katz, J.: Global-scale secure multiparty computation. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017. pp. 39–56. ACM Press (Oct / Nov 2017). https://doi.org/10.1145/3133956.3133979
58. 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
59. 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), https://www.usenix.org/conference/usenixsecurity21/presentation/weng
60. Weng, C., Yang, K., Yang, Z., Xie, X., Wang, X.: AntMan: Interactive zero-knowledge proofs with sublinear communication. pp. 2901–2914. ACM Press (2022). https://doi.org/10.1145/3548606.3560667

61. 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
62. Yang, K., Wang, X., Zhang, J.: More efficient MPC from improved triple generation and authenticated garbling. In: Ligatti, J., Ou, X., Katz, J., Vigna, G. (eds.) ACM CCS 2020. pp. 1627–1646. ACM Press (Nov 2020). https://doi.org/10.1145/3372297.3417285
63. 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
64. 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
65. 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
66. Zahur, S., Rosulek, M., Evans, D.: Two halves make a whole - reducing data transfer in garbled circuits using half gates. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 220–250 (Apr 2015). https://doi.org/10.1007/978-3-662-46803-6_8
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
68. Zhang, W., Guo, X., Yang, K., Zhu, R., Yu, Y., Wang, X.: Efficient actively secure DPF and RAM-based 2PC with one-bit leakage. pp. 561–577. IEEE Computer Society Press (2024). https://doi.org/10.1109/SP54263.2024.00205

A The Half-Gates Garbling Scheme

We first recall the garbling schemes abstraction in [36] which is a simplified version of [15] sufficient for the presentation of half-gates. A Boolean circuit consists of AND and XOR gates with fan-in 2. We represent any such circuit by a tuple $f = (n_{\mathsf{in}}, n_{\mathsf{out}}, n_{\mathsf{gates}}, \mathsf{Gates})$, where $n_{\mathsf{in}} \geq 2$ denotes the number of input wires, $n_{\mathsf{out}} \geq 1$ is the number of output wires, and n_{gates} is the number of gates. Such a circuit has exactly $n_{\mathsf{in}} + n_{\mathsf{gates}}$ wires that we number starting from 1; we let $\mathsf{In} = \{1, ..., n_{\mathsf{in}}\}$ and $\mathsf{Out} = n_{\mathsf{in}} + n_{\mathsf{gates}} - n_{\mathsf{out}} + 1, ..., n_{\mathsf{in}} + n_{\mathsf{gates}}$. The set $\mathsf{Gates} = (a, b, c, T)$, containing n_{gates} tuples, specifies the wiring of the circuit; a tuple $(a, b, c, T) \in \mathsf{Gates}$ with $a, b, c \in \{1, ..., n_{\mathsf{in}} + n_{\mathsf{gates}}\}$ represents a gate of type $T \in \{\mathsf{XOR}, \mathsf{AND}\}$ with input wires a, b and output wire c. For a circuit f we let $|f| = C$ denote the number of AND gates in f. With slight abuse of notation, we let f also denote the function $f : \{0,1\}^{n_{\mathsf{in}}} \to \{0,1\}^{n_{\mathsf{out}}}$ computed by the circuit. We consider a restricted class of garbling schemes in which garbling involves assigning two λ-bit labels to each wire of the circuit, and evaluation involves computing one label for each output wire.

Definition 10. *A garbling scheme is defined by three algorithms* $(\mathsf{Garble}, \mathsf{Eval}, \mathsf{Decode})$*:*

- Garble*: On input* 1^λ *and a boolean circuit* f*, outputs* $(\mathcal{GC}, \{\mathsf{L}_i^0, \mathsf{L}_i^1\}_{i\in\mathsf{In}}, d)$*, where* $\mathcal{GC}$ *is the garbled truth tables,* $\mathsf{L}_i^0, \mathsf{L}_i^1$ *are the wire labels for the* i*-th input wire, and and* d *is decoding information.*

- Eval*: On input* $(\mathcal{GC}, \{\mathsf{L}_i\}_{i\in\mathsf{In}})$ *as above, outputs the output wire labels* $\{\mathsf{L}_j\}_{j\in\mathsf{Out}}$*.*

- Decode*: On input* $(d, \{\mathsf{L}_j\}_{j\in\mathsf{Out}})$ *as above, outputs either a symbol* $\perp$ *indicating invalid output wire values, or the plain output* y*.*

We recall the correctness property of the garbling scheme as follows.

Definition 11. *A garbling scheme* $(\mathsf{Garble}, \mathsf{Eval}, \mathsf{Decode})$ *in Definition 10 is* correct *if for any circuit* f *and any input* $\boldsymbol{x} \in \{0,1\}^{n_{\mathsf{in}}}$*,* $\mathsf{Decode}(d, \mathsf{Eval}(\mathcal{GC}, \{\mathsf{L}_i^{x_i}\}_{i\in\mathsf{In}})) = f(\boldsymbol{x})$*, where* $(\mathcal{GC}, \{\mathsf{L}_i^0, \mathsf{L}_i^1\}_{i\in\mathsf{In}}, d) \leftarrow \mathsf{Garble}(1^\lambda, f)$*.*

We specify the half-gate garbled circuit scheme $\mathsf{HalfGates}^{\mathsf{H}}$ in Fig. 7. Note that we directly incorporate the authenticity check in the Decode algorithm.

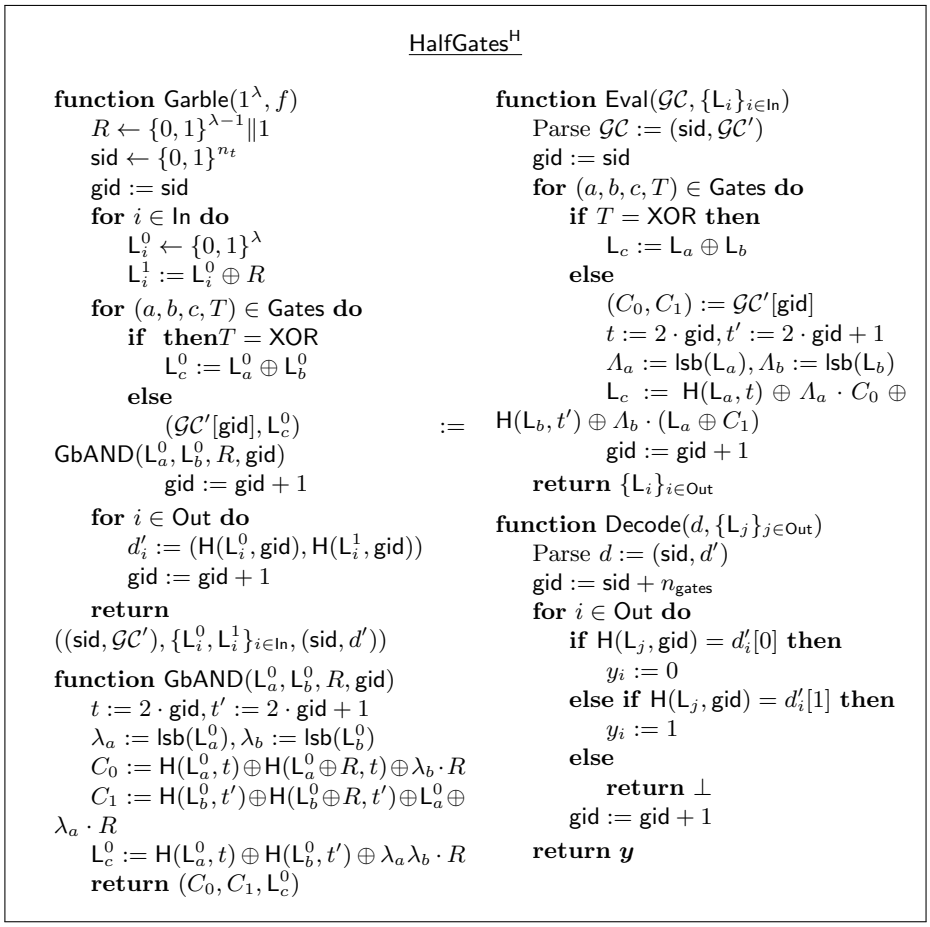

Fig. 7: The half-gates garbled circuit scheme. $\mathsf{H} : \{0,1\}^\lambda \times \{0,1\}^{n_t} \to \{0,1\}^\lambda$ is a tweakable hash function with tweak space $\{0,1\}^{n_t}$.

We also recall the quantified, multi-instance security notion of garbling schemes in [36]. In the following, we focus on constructions in the ideal cipher model since it encompasses all the efficient block-cipher-based garbled circuit constructions in the literature [36,53,37].

Definition 12. *A garbling scheme* (Garble, Eval, Decode) *defined in Definition 10 is* (q_E, u, C, ϵ)-private *if there is a simulator* $\mathcal{S}$ *so that for any distinguisher* D *making* q_E *queries to* E *and any* $(f^i, \boldsymbol{x}^i)_{i\in[u]}$ *with* $\sum_i |f^i| = C$*, we have*

$$\Big| \Pr_{\{(\mathcal{GC}^i,\{\mathsf{L}_j^{i,0},\mathsf{L}_j^{i,1}\},d^i)\leftarrow\mathsf{Garble}^E(1^\lambda,f^i)\}_{i\in[u]}} \Big[D^E(\{\mathcal{GC}^i,\mathsf{L}_j^{i,x_j^i},d^i\}_{i\in[u]})=1\Big] -$$
$$\Pr_{\{(\mathcal{GC}^i,\{\mathsf{L}_j^{i}\},d^i)\leftarrow\mathcal{S}^E(1^\lambda,f^i,f^i(\boldsymbol{x}^i))\}_{i\in[u]}} \big[D^E(\{\mathcal{GC}^i,\mathsf{L}_j^{i},d^i\}_{i\in[u]})=1\big]\Big| < \epsilon$$

Definition 13. *A garbling scheme* (Garble, Eval, Decode) *defined in Definition 10 is* (q_E, u, C, ϵ)-oblivious *if there is a simulator* $\mathcal{S}$ *so that for any distinguisher* D *making* q_E *queries to* E *and any* $(f^i, \boldsymbol{x}^i)_{i\in[u]}$ *with* $\sum_i |f^i| = C$*, we have*

$$\Big| \Pr_{\{(\mathcal{GC}^i,\{\mathsf{L}_j^{i,0},\mathsf{L}_j^{i,1}\})\leftarrow\mathsf{Garble}^E(1^\lambda,f^i)\}_{i\in[u]}} \Big[D^E(\{\mathcal{GC}^i,\mathsf{L}_j^{i,x_j^i}\}_{i\in[u]})=1\Big] -$$
$$\Pr_{\{(\mathcal{GC}^i,\{\mathsf{L}_j^{i}\})\leftarrow\mathcal{S}^E(1^\lambda,f^i)\}_{i\in[u]}} \big[D^E(\{\mathcal{GC}^i,\mathsf{L}_j^{i}\}_{i\in[u]})=1\big]\Big| < \epsilon$$

Definition 14. *A garbling scheme* (Garble, Eval, Decode) *defined in Definition 10 is* $(q_E, u, C, n_{\mathsf{out}}, \epsilon)$-authentic *if for any distinguisher* D *making* q_E *queries to* E *and any* $(f^i, \boldsymbol{x}^i)_{i\in[u]}$ *with* $\sum_i |f^i| = C$ *and* $\sum_i m_i = n_{\mathsf{out}}$*, we have*

$$\Pr_{\substack{\{(\mathcal{GC}^i,\{\mathsf{L}_j^{i,0},\mathsf{L}_j^{i,1}\})\\ \leftarrow\mathsf{Garble}^E(1^\lambda,f^i)\}_{i\in[u]}}} \left[\begin{array}{r} \mathcal{A}^E(\{\mathcal{GC}^i,\mathsf{L}_j^{i,x_j^i}\}_{i\in[u]}) = (i',\{\mathsf{L}'_j\}),\\ \mathsf{Eval}(\mathcal{GC}^{i'},\{\mathsf{L}_j^{i,x_j^i}\}) \neq \{\mathsf{L}'_j\} \wedge \mathsf{Decode}(d^{i'},\{\mathsf{L}'_j\}) \neq \perp \end{array}\right] < \epsilon$$

We recall the security reductions from the garbled circuit security to the miTCCR security of the hash function in [36]. Together with our results in Section 4.1, we obtain a garbled circuit scheme with a security level of up to 256-bit from standard AES in Corollary 1.

Lemma 11 (Theorem 3 in [36]). *For any integer* $\mu \geq 1$*, if* H *is* $(q_E, 2C + n_{\mathsf{out}}, u, \mu, \lambda-1, \epsilon)$*-miTCCR, then the garbling scheme* $\mathsf{HalfGates}^{\mathsf{H}}$ *is* (q_E, u, C, ϵ')*-private where*

$$\epsilon' \leq \epsilon + \frac{(2C + n_{\mathsf{out}})^{\mu+1}}{(\mu+1)! \cdot 2^{\mu n_t}}$$

Lemma 12 (Theorem 4 in [36]). *For any integer* $\mu \geq 1$*, if* H *be* $(q_E, 2C + n_{\mathsf{out}}, u, \mu, \lambda-1, \epsilon)$*-miTCCR, then the garbling scheme* $\mathsf{HalfGates}^{\mathsf{H}}$ *is* (q_E, u, C, ϵ')*-oblivious where*

$$\epsilon' \leq \epsilon + \frac{(2C + n_{\mathsf{out}})^{\mu+1}}{(\mu+1)! \cdot 2^{\mu n_t}}$$

Lemma 13 (Theorem 5 in [36]). *For any integer* $\mu \geq 1$*, if* H *be* $(q_E, 2C + n_{\mathsf{out}}, u, \mu, \lambda-1, \epsilon)$*-miTCCR, then the garbling scheme* $\mathsf{HalfGates}^{\mathsf{H}}$ *is* $(q_E, u, C, n_{\mathsf{out}}, \epsilon')$*-authentic where*

$$\epsilon' \leq \epsilon + \frac{(2C + n_{\mathsf{out}})^{\mu+1}}{(\mu+1)! \cdot 2^{\mu n_t}} + 2^{-\lambda}$$

B Proof of Lemma 1

Proof. Fix a deterministic distinguisher D making queries to the two types of oracles. The first type is the ideal cipher oracle and its inverse, where the key size is λ-bit and the block space is n-bit. In the real world, the second type is the construction oracles, namely $\mathcal{O}^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}}}(w,t,b) = \mathsf{H}_{\mathsf{miTCCR}}(\Gamma^{\mathsf{idx}} \oplus w, t) \oplus b \cdot \Gamma$ for $\mathsf{idx} \in [u]$, $\Gamma^{\mathsf{idx}} \leftarrow \mathcal{R}$; Whereas in the ideal world the second type is u independent random function from $\{0,1\}^{\lambda+n-7}$ to $\{0,1\}^{\lambda}$.

For the sake of notational clarity, we define $t_0 := [\mathsf{C}_0]_8 \| t$, $t_1 := [\mathsf{C}_1]_8 \| t$. Recall from Fig. 3 that $n \le \lambda \le 2n$, $r_L = \mathsf{left}_n(r)$ and $r_R = \mathsf{right}_{\lambda-n}(r)$. We employ a special treatment for the case where $n < \lambda < 2n$ (which means $n \nmid \lambda$). Concretely, in this case, instead of defining $z_L \leftarrow E\big(r_R \| t_0, \sigma(r_L)\big) \oplus \sigma(r_L)$ and $z_R \leftarrow \mathsf{left}_{\lambda-n}\big(E\big(r_R \| t_1, \sigma(r_L)\big) \oplus \sigma(r_L)\big)$ as in Fig. 3, we define $z_L \leftarrow E\big(r_R \| t_0, \sigma(r_L)\big) \oplus \sigma(r_L)$ and $z_R \leftarrow E\big(r_R \| t_1, \sigma(r_L)\big) \oplus \sigma(r_L)$, i.e., the truncation is removed; and further for $\mathsf{idx} \in [u]$, we define $(\mathcal{O}')^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}}}(x,t,b) = (z_L \| z_R) \oplus b \cdot (\Gamma^{\mathsf{idx}} \| [0]_{2n-\lambda})$, i.e., the output of $(\mathcal{O}')^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}}}$ is of $2n$ bits. We focus on indistinguishability of $(\mathcal{O}')^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}}}$ and an expanding random function $f' : \{0,1\}^{\lambda} \times \{0,1\}^{n-8} \times \{0,1\} \to \{0,1\}^{2n}$: note that this implies indistinguishability of the original $\mathcal{O}^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}}}$ and $f : \{0,1\}^{\lambda} \times \{0,1\}^{n-8} \times \{0,1\} \to \{0,1\}^{\lambda}$.

With the above in mind, let $\mathcal{Q}_E$ and $\mathcal{Q}_{\mathcal{O}}$ be the transcripts of ideal cipher and construction queries and responses respectively. $\mathcal{Q}_E$ consists of entries like (k,x,y), which records a query/answer pair to/from E or E^{-1} with $E(k,x) = y$. $\mathcal{Q}_{\mathcal{O}}$ consists of entries like $(\mathsf{idx}, w, t, b, z)$, which records a query/answer pair to/from the idx-th construction oracle $\mathcal{O}^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}}}$ with $\mathcal{O}^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}}}(w,t,b) = z$. Let $\mathbf{\Gamma} := (\Gamma^1, ..., \Gamma^u)$ be the vector of all global keys.

Denote the transcript of D's interaction between the oracles by $\mathcal{Q} = (\mathcal{Q}_E, \mathcal{Q}_{\mathcal{O}}, \mathbf{\Gamma})$, with $\mathcal{Q}_E$ and $\mathcal{Q}_{\mathcal{O}}$ defined above. Following [23,37], the keys $\mathbf{\Gamma}$ 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 keys used by the construction oracles, whereas in the ideal world, it is simply u independent keys each 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, for $\mathsf{idx} \in [u]$, we define $\Gamma^{\mathsf{idx}}_L := \mathsf{left}_n(\Gamma^{\mathsf{idx}})$ and $\Gamma^{\mathsf{idx}}_R := \mathsf{right}_{\lambda-n}(\Gamma^{\mathsf{idx}})$. We assume that the key size of E always equals the security parameter λ. We also recall that for each $(\mathsf{idx}, w, t, b, z) \in \mathcal{Q}_{\mathcal{O}}$ in the real world we have

$$z_L = E(w_R \oplus \Gamma^{\mathsf{idx}}_R \| t_0, \sigma(w_L \oplus \Gamma^{\mathsf{idx}}_L)) \oplus \sigma(w_L \oplus \Gamma^{\mathsf{idx}}_L) \oplus b \cdot \Gamma^{\mathsf{idx}}_L$$
$$z_R = E(w_R \oplus \Gamma^{\mathsf{idx}}_R \| t_1, \sigma(w_L \oplus \Gamma^{\mathsf{idx}}_L)) \oplus \sigma(w_L \oplus \Gamma^{\mathsf{idx}}_L) \oplus b \cdot (\Gamma^{\mathsf{idx}}_R \| [0]_{2n-\lambda}) \ .$$

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

– (B-1) There is a query $(\mathsf{idx}, w, t, b, z) \in \mathcal{Q}_{\mathcal{O}}$ and a query of the following form in $\mathcal{Q}_E$. (Recall that we defined $t_0 := [\mathsf{C}_0]_8 \| t$ and $t_1 := [\mathsf{C}_1]_8 \| t$.)

- $(w_R \oplus \Gamma_R^{\mathsf{idx}} \| t_0, \sigma(w_L \oplus \Gamma_L^{\mathsf{idx}}), \star)$
- $(w_R \oplus \Gamma_R^{\mathsf{idx}} \| t_1, \sigma(w_L \oplus \Gamma_L^{\mathsf{idx}}), \star)$, if $n < \lambda \leq 2n$
- $(w_R \oplus \Gamma_R^{\mathsf{idx}} \| t_0, \star, z_L \oplus \sigma(w_L \oplus \Gamma_L^{\mathsf{idx}}) \oplus b \cdot \Gamma_L^{\mathsf{idx}})$
- $(w_R \oplus \Gamma_R^{\mathsf{idx}} \| t_1, \star, z_R \oplus \sigma(w_L \oplus \Gamma_L^{\mathsf{idx}}) \oplus b \cdot (\Gamma_R^{\mathsf{idx}} \| [0]_{2n-\lambda}))$, if $n < \lambda \leq 2n$

– (B-2) There are distinct $(\mathsf{idx}, w, t, b, z) \in \mathcal{Q}_{\mathcal{O}}$, $(\mathsf{idx}', w', t', b', z') \in \mathcal{Q}_{\mathcal{O}}$, subject to $w_R = w'_R$ and $t = t'$ where either of the following two cases occurs.
- $\sigma(w_L) \oplus b \cdot \Gamma_L^{\mathsf{idx}} \oplus z_L = \sigma(w'_L) \oplus b' \cdot \Gamma_L^{\mathsf{idx}'} \oplus z'_L$
- $\sigma(w_L) \oplus b \cdot (\Gamma_R^{\mathsf{idx}} \| [0]_{2n-\lambda}) \oplus z_R = \sigma(w'_L) \oplus b' \cdot (\Gamma_R^{\mathsf{idx}'} \| [0]_{2n-\lambda}) \oplus z'_R$ in the case of $n < \lambda \leq 2n$. (Again, since we have removed the truncation, it always holds that $|z_R| = |z'_R| = n$.)

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

Case 1. The bad event is $\big(w_R \oplus \Gamma_R^{\mathsf{idx}} \| t_0 = k\big) \wedge \big(\sigma(w_L \oplus \Gamma_L^{\mathsf{idx}}) = x\big)$. Using the linear property of σ and rearranging the terms, this is equivalent to

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

Since σ is a permutation, we can recover Γ^{idx} from this bad query. Since $\Gamma \leftarrow \mathcal{R}$ and $\mathcal{R}$ has min-entropy at least ρ, we have

$$\Pr\left[\big(\Gamma_R^{\mathsf{idx}} = \mathsf{left}_{\lambda-n}(k) \oplus w_R\big) \wedge \big(\sigma(\Gamma_L^{\mathsf{idx}}) = x \oplus \sigma(w_L)\big)\right] \leq 2^{-\rho}$$

Case 2. The bad event is $\big(w_R \oplus \Gamma_R^{\mathsf{idx}} \| t_1 = k\big) \wedge \big(\sigma(w_L \oplus \Gamma_L^{\mathsf{idx}}) = x\big)$. Using the same argument as the first case, we conclude that the probability of this event is bounded by $2^{-\rho}$.

Case 3. The bad event is $\big(w_R \oplus \Gamma_R^{\mathsf{idx}} \| t_0 = k\big) \wedge \big(z_L \oplus \sigma(w_L \oplus \Gamma_L^{\mathsf{idx}}) \oplus b \cdot \Gamma_L^{\mathsf{idx}} = y\big)$. Consider the following two sub-cases. If $b = 0$ then the event is equivalent to $\big(\Gamma_R^{\mathsf{idx}} = w_R \oplus \mathsf{left}_{\lambda-n}(k)\big) \wedge \big(\sigma(\Gamma_L^{\mathsf{idx}}) = z_L \oplus \sigma(w_L) \oplus y\big)$, which happens with probability $2^{-\rho}$. If $b = 1$ then the event is $\big(\Gamma_R^{\mathsf{idx}} = w_R \oplus \mathsf{left}_{\lambda-n}(k)\big) \wedge \big(\sigma(\Gamma_L^{\mathsf{idx}}) \oplus \Gamma_L^{\mathsf{idx}} = z_L \oplus \sigma(w_L) \oplus y\big)$. Since σ is an orthomorphism, the map $x \mapsto \sigma(x) \oplus x$ is a permutation. We conclude that this event also happens with probability $2^{-\rho}$.

Case 4. The bad event is equivalent to $\big(w_R \oplus \Gamma_R^{\mathsf{idx}} \| t_1 = k\big) \wedge \big(z_R \oplus \sigma(w_L \oplus \Gamma_L^{\mathsf{idx}}) \oplus b \cdot (\Gamma_R^{\mathsf{idx}} \| [0]_{2n-\lambda}) = y\big)$. Rearranging the terms we have

$$\begin{aligned}
&\Pr\left[\big(w_R \oplus \Gamma_R^{\mathsf{idx}} \| t_1 = k\big) \wedge \big(z_R \oplus \sigma(w_L \oplus \Gamma_L^{\mathsf{idx}}) \oplus b \cdot \Gamma_R^{\mathsf{idx}} = y\big)\right] \\
&= \Pr\left[\Gamma_R^{\mathsf{idx}} = \mathsf{left}_{\lambda-n}(k) \oplus w_R\right] \cdot \\
&\quad \Pr\left[\sigma(\Gamma_L^{\mathsf{idx}}) = z_R \oplus \sigma(w_L) \oplus b \cdot (\Gamma_R^{\mathsf{idx}} \| [0]_{2n-\lambda}) \oplus y \mid \Gamma_R^{\mathsf{idx}} = \mathsf{left}_{\lambda-n}(k) \oplus w_R\right] \\
&\leq 2^{-\rho}
\end{aligned}$$

For $t \in \{0,1\}^{n-8}$, define $\mathcal{Q}_E[t] \subseteq \mathcal{Q}_E$ as

$$\mathcal{Q}_E[t] := \big\{(k, x, y) \in \mathcal{Q}_E \mid \mathsf{right}_{n-8}(k) = t\big\}$$

From union bound we conclude that

$$\begin{aligned}\Pr\left[\text{(B-1)}\right] &\leq \sum_{(\mathsf{idx},w,t,b,z)\in\mathcal{Q}_{\mathcal{O}}} \frac{4|\mathcal{Q}_E[t]|}{2^\rho} \\ &= \sum_{t\in\{0,1\}^{n-8}} \underbrace{\sum_{(\mathsf{idx},w,t,b,z)\in\mathcal{Q}_{\mathcal{O}}}}_{\leq\mu} \frac{4|\mathcal{Q}_E[t]|}{2^\rho} \\ &\leq \mu \sum_{t\in\{0,1\}^{n-8}} \frac{4|\mathcal{Q}_E[t]|}{2^\rho} \\ &= \frac{4\mu q_E}{2^\rho}\end{aligned}$$

For (B-2), consider distinct $(\mathsf{idx},w,t,b,z),(\mathsf{idx}',w',t',b',z')\in\mathcal{Q}_{\mathcal{O}}$. Note that even if we take condition on the values of $\mathbf{\Gamma}$, the values z,z' are uniform and independent. Thus, for $\mathsf{side}\in\{L,R\}$, we have

$$\Pr[\sigma(w_{\mathsf{side}}\oplus w'_{\mathsf{side}})\oplus b_{\mathsf{side}}\cdot\Gamma^{\mathsf{idx}}_{\mathsf{side}}\oplus b'_{\mathsf{side}}\cdot\Gamma^{\mathsf{idx}'}_{\mathsf{side}}=z_{\mathsf{side}}\oplus z'_{\mathsf{side}}]=2^{-n}.$$

Let C_t denote the number of queries in $\mathcal{Q}_{\mathcal{O}}$ with fixed tweak t. By union bound, we have

$$\begin{aligned}\Pr[\text{(B-2)}] &\leq \sum_{t\in\{0,1\}^{n-8}} \binom{C_t}{2}\frac{2}{2^n} \\ &\leq \sum_{t\in\{0,1\}^{n-8}} \frac{C_t\cdot(\mu-1)}{2}\cdot\frac{2}{2^n} \\ &= \frac{q_C(\mu-1)}{2^n}\end{aligned}$$

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

$$\begin{cases}\frac{1}{2^{\rho\cdot u}}\cdot\frac{1}{(2^n)_{q_{E,1}}\cdot\ldots\cdot(2^n)_{q_{E,m}}}\cdot\frac{1}{2^{q_C\cdot n}} & \text{if } \lambda=n \\ \frac{1}{2^{\rho\cdot u}}\cdot\frac{1}{(2^n)_{q_{E,1}}\cdot\ldots\cdot(2^n)_{q_{E,m}}}\cdot\frac{1}{2^{2\cdot q_C\cdot n}} & \text{if } n<\lambda\leq 2n\end{cases} \tag{12}$$

where $m\in[1,q_E]$ denote the number of distinct keys in $\mathcal{Q}_E$ and $q_{E,1},\ldots,q_{E,m}$ 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

$$\frac{1}{2^{\rho\cdot u}}\cdot\frac{1}{(2^n)_{q_{E,1}}\cdot\ldots\cdot(2^n)_{q_{E,m}}}\cdot p_{\mathcal{O}}$$

where

$$p_{\mathcal{O}}=\begin{cases}\Pr[\forall(\mathsf{idx},w,t,b,z)\in\mathcal{Q}_{\mathcal{O}}:\mathcal{O}^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}}}(w,t,b)=z\mid E\vdash\mathcal{Q}_E] & \text{if } n\mid\lambda \\ \Pr[\forall(\mathsf{idx},w,t,b,z)\in\mathcal{Q}_{\mathcal{O}}:(\mathcal{O}')^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}}}(w,t,b)=z\mid E\vdash\mathcal{Q}_E] & \text{if } n\nmid\lambda\end{cases}$$

Recall that for $\mathsf{idx} \in [u]$, $(\mathcal{O}')^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}}}$ is the construction oracle obtained by removing the truncation.

Fix an order in $\mathcal{Q}_{\mathcal{O}}$. For $i \in q_C$, let $(\mathsf{idx}^i, w^i, t^i, b^i, z^i)$ be the i-th entry in $\mathcal{Q}_{\mathcal{O}}$. We can expand $p_{\mathcal{O}}$ as

$$\begin{cases} \prod_{i=1}^{q_C} \Pr\big[\mathcal{O}^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}^i}}(w^i,t^i,b^i) = z^i \mid E \vdash \mathcal{Q}_E \wedge \\ \qquad \forall j < i : \mathcal{O}^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}^j}}(w^j,t^j,b^j) = z^j\big] & \text{if } n \mid \lambda \\ \prod_{i=1}^{q_C} \Pr[(\mathcal{O}')^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}^i}}(w^i,t^i,b^i) = z^i \mid E \vdash \mathcal{Q}_E \wedge \\ \qquad \forall j < i : (\mathcal{O}')^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}^j}}(w^j,t^j,b^j) = z^j] & \text{if } n \nmid \lambda \end{cases}$$

Consider the $n \mid \lambda$ case first. Note that $\mathcal{O}^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}^i}}(w^i,t^i,b^i) = z^i$ iff $\mathsf{H}_{\mathsf{miTCCR}}(\Gamma^{\mathsf{idx}^i} \oplus w^i, t^i) \oplus b^i \cdot \Gamma = z^i$, i.e.,

$$\begin{aligned} z^i_L &= E(w^i_R \oplus \Gamma^{\mathsf{idx}^i}_R \| t_0, \sigma(w^i_L \oplus \Gamma^{\mathsf{idx}^i}_L)) \oplus \sigma(w^i_L \oplus \Gamma^{\mathsf{idx}^i}_L) \oplus b^i \cdot \Gamma^{\mathsf{idx}^i}_L \\ z^i_R &= E(w^i_R \oplus \Gamma^{\mathsf{idx}^i}_R \| t_1, \sigma(w^i_L \oplus \Gamma^{\mathsf{idx}^i}_L)) \oplus \sigma(w^i_L \oplus \Gamma^{\mathsf{idx}^i}_L) \oplus b^i \cdot \Gamma_R \ (\text{if } \lambda = 2n) \end{aligned}$$

Since the transcript is good there is no query of the form $(w^i_R \oplus \Gamma^{\mathsf{idx}^i}_R \| t_0, \sigma(\Gamma^{\mathsf{idx}^i}_L \oplus w^i_L), \star)$ or $(w^i_R \oplus \Gamma^{\mathsf{idx}^i}_R \| t_1, \sigma(\Gamma^{\mathsf{idx}^i}_L \oplus w^i_L), \star)$ in $\mathcal{Q}_E$ (since (B-1) does not occur), nor is $E(w^i_R \oplus \Gamma^{\mathsf{idx}^i}_R \| t_0, \sigma(w^i_L \oplus \Gamma^{\mathsf{idx}^i}_L))$ or $E(w^i_R \oplus \Gamma^{\mathsf{idx}^i}_R \| t_1, \sigma(w^i_L \oplus \Gamma^{\mathsf{idx}^i}_L))$ determined by the fact that $\mathcal{O}^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}^j}}(w^j,t^j,b^j) = z^j$ for all $j < i$ (since D does not make two queries to $\mathcal{O}^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}^i}}$ with the same (w^i,t^i)).

Similarly, there is no query of the form $(w^i_R \oplus \Gamma^{\mathsf{idx}^i}_R \| t_0, \star, \sigma(w^i_L \oplus \Gamma^{\mathsf{idx}^i}_L) \oplus b^i \cdot \Gamma^{\mathsf{idx}^i}_L \oplus z^i)$ or $(w^i_R \oplus \Gamma^{\mathsf{idx}^i}_R \| t_1, \star, z_R \oplus \sigma(w^i_L \oplus \Gamma^{\mathsf{idx}^i}_L) \oplus b^i \cdot \Gamma^{\mathsf{idx}^i}_R)$ in $\mathcal{Q}_E$ (since (B-1) does not occur), nor is $E^{-1}(w^i_R \oplus \Gamma^{\mathsf{idx}^i}_R \| t_0, \sigma(w^i_L \oplus \Gamma^{\mathsf{idx}^i}_L) \oplus b^i \cdot \Gamma^{\mathsf{idx}^i}_L \oplus z^i_L)$ or $E^{-1}(w^i_R \oplus \Gamma^{\mathsf{idx}^i} \| t_1, \sigma(w^i_L \oplus \Gamma_L) \oplus b^i \cdot \Gamma_R \oplus z^i_R)$ determined by the fact that $\mathcal{O}^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}^j}}(w^j,t^j,b^j) = z^j$ for all $j < i$ (since (B-2) does not occur). Thus, for all i, let

$$p^i_{\mathcal{O}} = \Pr[\mathcal{O}^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}^i}}(w^i,t^i,b^i) = z^i \mid E \vdash \mathcal{Q}_E \wedge \forall j < i : \mathcal{O}^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}^j}}(w^j,t^j,b^j) = z^j]$$

we have

$$\begin{aligned} p^i_{\mathcal{O}} &\geq \frac{1}{2^n - q_E - i + 1} \geq \frac{1}{2^\lambda} && (\lambda = n) \\ p^i_{\mathcal{O}} &\geq \frac{1}{2^n - q_E - i + 1} \cdot \frac{1}{2^n - q_E - i + 1} \geq \frac{1}{2^\lambda} && (\lambda = 2n) \end{aligned}$$

On the other hand, $(\mathcal{O}')^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}^i}}(w^i,t^i,b^i) = z^i$ iff

$$\begin{aligned} z^i_L &= E(w^i_R \oplus \Gamma^{\mathsf{idx}^i}_R \| t_0, \sigma(w^i_L \oplus \Gamma^{\mathsf{idx}^i}_L)) \oplus \sigma(w^i_L \oplus \Gamma^{\mathsf{idx}^i}_L) \oplus b^i \cdot \Gamma^{\mathsf{idx}^i}_L \\ z^i_R &= E(w^i_R \oplus \Gamma^{\mathsf{idx}^i}_R \| t_1, \sigma(w^i_L \oplus \Gamma^{\mathsf{idx}^i}_L)) \oplus \sigma(w^i_L \oplus \Gamma^{\mathsf{idx}^i}_L) \oplus b^i \cdot (\Gamma^{\mathsf{idx}^i}_R \| [0]_{2n-\lambda}). \end{aligned}$$

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

$$\Pr[(\mathcal{O}')^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}^i}}(w^i,t^i,b^i)=z^i \mid E\vdash \mathcal{Q}_E\wedge\forall j<i:(\mathcal{O}')^{\mathsf{miTCCR}}_{\Gamma^{\mathsf{idx}^j}}(w^j,t^j,b^j)=z^j] \\ \geq \frac{1}{2^n-q_E-i+1}\cdot\frac{1}{2^n-q_E-i+1}\geq\frac{1}{2^{2n}}.$$

It follows that

$$p_{\mathcal{O}}\geq\frac{1}{2^{q_C\cdot n}} \qquad (\lambda=n),$$
$$p_{\mathcal{O}}\geq\frac{1}{2^{2\cdot q_C\cdot n}} \qquad (n<\lambda\leq 2n),$$

and so the probability that the real world is consistent with the transcript is always at least (12). This completes the proof. □

C Proof of Lemma 3

Our proof follows [1, Theorem 3] (which was motivated by [47]).

Proof setup. Let $t,t'\in\{0,1\}^{n-8}$ be the tweaks submitted by $\mathcal{A}$ in $\mathbf{Exp}^{\mathrm{Coll}}(\mathcal{A})$. Define $t_0:=[\mathsf{C}_0]_8\|t$, $t_1:=[\mathsf{C}_1]_8\|t$, $t'_0:=[\mathsf{C}_0]_8\|t'$, $t'_1:=[\mathsf{C}_1]_8\|t'$. For convenience, define $\Delta_k:=[0]_{\lambda-n}\|([\mathsf{C}_0]_8\oplus[\mathsf{C}_1]_8)\|[0]_{n-8}$, $\Delta_{\mathsf{C}}:=[\mathsf{C}_0]_n\oplus[\mathsf{C}_1]_n$, and $\Delta_p:=\sigma(\Delta_{\mathsf{C}})$: note that $(r_R\|t_b)\oplus\Delta_k=(r_R\|t_{\bar{b}})$ for any $r_R\in\{0,1\}^{\lambda-n}$ and $b\in\{0,1\}$, and $\sigma(x\oplus[\mathsf{C}_b]_n)\oplus\Delta_p=\sigma(x\oplus[\mathsf{C}_{\bar{b}}]_n)$ for any $x\in\{0,1\}^n$ and $b\in\{0,1\}$ (since σ 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

$$\big\{\ (r_R\|t_0,\sigma(r_L\oplus[\mathsf{C}_0]_n),y_0^0),(r_R\|t_1,\sigma(r_L\oplus[\mathsf{C}_0]_n),y_0^1),\\ (r_R\|t_0,\sigma(r_L\oplus[\mathsf{C}_1]_n),y_1^0),(r_R\|t_1,\sigma(r_L\oplus[\mathsf{C}_1]_n),y_1^1)\big\} \tag{13}$$

is called a *pile*. We stress that it is an *orderless* set. The involved keys $r_R\|t_0\in\{0,1\}^\lambda$ and $r_R\|t_1\in\{0,1\}^\lambda$ are called *the (two) keys of the pile*. For convenience, we further define

$$\mathcal{Q}_E[k]:=\big\{(x,y):(k,x,y)\in\mathcal{Q}_E\big\} \tag{14}$$

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

Following [47], 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 [41,4,47].

Due to the above "free" queries, records in the adversary's query history $\mathcal{Q}$ can be grouped into "piles" as defined in Eq. (13). 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 $\mathcal{Q}$. Since $\mathcal{A}$ 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 $\mathcal{Q}$.

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 2×2^n inputs of $E(r_R \| t_0, \cdot), E(r_R \| t_1, \cdot)$ constitute $2^n/2$ disjoint piles (i.e., no input is shared among distinct piles). Meanwhile, each pile of Eq. (13) provides 2 possible

commitment evaluations, i.e.,

$$\begin{aligned}
\mathsf{Com}(r_L\|r_R,t) = \big(\;& \sigma(r_L \oplus [\mathsf{C}_0]_n) \oplus E(r_R\|t_0, \sigma(r_L \oplus [\mathsf{C}_0]_n)),\\
& \mathsf{left}_{\lambda-n}\big(\sigma(r_L \oplus [\mathsf{C}_0]_n) \oplus E(r_R\|t_1, \sigma(r_L \oplus [\mathsf{C}_0]_n))\big),\\
& \sigma(r_L \oplus [\mathsf{C}_1]_n) \oplus E(r_R\|t_0, \sigma(r_L \oplus [\mathsf{C}_1]_n)),\\
& \mathsf{left}_{\lambda-n}\big(\sigma(r_L \oplus [\mathsf{C}_1]_n) \oplus E(r_R\|t_1, \sigma(r_L \oplus [\mathsf{C}_1]_n))\big)\big),\\
\mathsf{Com}(r_L \oplus \Delta_{\mathsf{C}}\|r_R,t) = \big(\;& \sigma(r_L \oplus [\mathsf{C}_1]_n) \oplus E(r_R\|t_0, \sigma(r_L \oplus [\mathsf{C}_1]_n)),\\
& \mathsf{left}_{\lambda-n}\big(\sigma(r_L \oplus [\mathsf{C}_1]_n) \oplus E(r_R\|t_1, \sigma(r_L \oplus [\mathsf{C}_1]_n))\big),\\
& \sigma(r_L \oplus [\mathsf{C}_0]_n) \oplus E(r_R\|t_0, \sigma(r_L \oplus [\mathsf{C}_0]_n)),\\
& \mathsf{left}_{\lambda-n}\big(\sigma(r_L \oplus [\mathsf{C}_0]_n) \oplus E(r_R\|t_1, \sigma(r_L \oplus [\mathsf{C}_0]_n))\big)\big),
\end{aligned}$$

Collisions may occur among distinct commitment evaluations from a single pile, e.g., $\mathsf{Com}(r_L\|r_R,t) = \mathsf{Com}(r_L \oplus \Delta_{\mathsf{C}}\|r_R,t)$, which will be refereed 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

$$\begin{aligned}
\big\{ \;& (r_R\|t_0, \sigma(r_L \oplus [\mathsf{C}_0]_n), y_0^0), (r_R\|t_1, \sigma(r_L \oplus [\mathsf{C}_0]_n), y_0^1),\\
& (r_R\|t_0, \sigma(r_L \oplus [\mathsf{C}_1]_n), y_1^0), (r_R\|t_1, \sigma(r_L \oplus [\mathsf{C}_1]_n), y_1^1)\big\}
\end{aligned}$$

  that has *internal collisions*, i.e., $\mathsf{Com}(r_L\|r_R,t) = \mathsf{Com}(r_L \oplus \Delta_p\|r_R,t)$.
- (B-2) $\mathcal{A}$ obtains two distinct piles

$$\begin{aligned}
\big\{ \;& (r_R\|t_0, \sigma(r_L \oplus [\mathsf{C}_0]_n), y_0^0), (r_R\|t_1, \sigma(r_L \oplus [\mathsf{C}_0]_n), y_0^1),\\
& (r_R\|t_0, \sigma(r_L \oplus [\mathsf{C}_1]_n), y_1^0), (r_R\|t_1, \sigma(r_L \oplus [\mathsf{C}_1]_n), y_1^1)\big\}
\end{aligned}$$

  and

$$\begin{aligned}
\big\{ \;& (r'_R\|t'_0, \sigma(r'_L \oplus [\mathsf{C}_0]_n), y_0^{0'}), (r'_R\|t'_1, \sigma(r'_L \oplus [\mathsf{C}_0]_n), y_0^{1'}),\\
& (r'_R\|t'_0, \sigma(r'_L \oplus [\mathsf{C}_1]_n), y_1^{0'}), (r'_R\|t'_1, \sigma(r'_L \oplus [\mathsf{C}_1]_n), y_1^{1'})\big\}
\end{aligned}$$

  that have *external collisions*, i.e., there exists $b, b' \in \{0,1\}$ such that $\mathsf{Com}(r_L \oplus b \cdot \Delta_{\mathsf{C}}\|r_R,t) = \mathsf{Com}(r'_L \oplus b' \cdot \Delta_{\mathsf{C}}\|r'_R,t')$.

We analyze them in turn.

Event (B-1). The collision $\mathsf{Com}(r_L\|r_R,t) = \mathsf{Com}(r_L \oplus \Delta_{\mathsf{C}}\|r_R,t)$ is equivalent with 2 equalities as follows:

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

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.

(B-11). First, consider the case $\mathcal{A}$ making a forward query $E(r_R\|t_0, \sigma(r_L \oplus [\mathsf{C}_0]_n)) \to y_0^0$. Then, at most $2^n/2 - 2$ queries (counting free queries) have been previously answered with each of the two keys $r_R\|t_0$ and $r_R\|t_1$ since otherwise a super query for that key would have occurred. Thus the 2 answers $E(r_R\|t_1, \sigma(r_L \oplus [\mathsf{C}_0]_n)) \to y_0^1, E(r_R\|t_1, \sigma(r_L \oplus [\mathsf{C}_1]_n)) \to y_1^1$ are distinct values uniformly picked from a set of size at least $2^n/2 + 2$. By these,

- The probability to have $\sigma(r_L \oplus [\mathsf{C}_0]_n) \oplus y_0^0 = \sigma(r_L \oplus [\mathsf{C}_1]_n) \oplus y_1^0$ is at most $1/(2^n/2+1) \le 2/2^n$;
- Conditioned on $E(r_R\|t_1, \sigma(r_L \oplus [\mathsf{C}_0]_n)) = y_0^1$, it holds

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

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

When $\mathcal{A}$ makes a backward query $E^{-1}(r_R\|t_0, y_0^0) \to x = \sigma(r_L \oplus [\mathsf{C}_0]_n)$, the analysis is similar by symmetry: the associated forward queries $E(r_R\|t_1, \sigma(r_L \oplus [\mathsf{C}_0]_n))$ and $E(r_R\|t_1, \sigma(r_L \oplus [\mathsf{C}_1]_n))$ 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 $E(r_R\|t_1, \sigma(r_L \oplus [\mathsf{C}_0]_n)) \to y_0^1$ or $E^{-1}(r_R\|t_1, y_0^1) \to x$, the associated forward queries $E(r_R\|t_0, \sigma(r_L \oplus [\mathsf{C}_0]_n))$ and $E(r_R\|t_0, \sigma(r_L \oplus [\mathsf{C}_1]_n))$ 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

$$\Pr\big[\text{(B-11)}\big] \le \frac{4q_E}{2^\lambda}. \tag{15}$$

(B-12). Suppose a super query is about to occur on $r_R \in \{0,1\}^{\lambda-n}$, meaning that the value of $E(r_R\|t_0, \cdot)$ is already known on exactly $2^n/2$ points. Let $\sigma(r_L \oplus [\mathsf{C}_0]_n), \sigma(r_L \oplus [\mathsf{C}_1]_n)$ 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 $E(r_R\|t_0, x)$ is not yet known,

and will be queried as part of the super query; note that a point $\sigma(r_L \oplus [\mathsf{C}_0]_n) \in \{0,1\}^n$ is in the domain of the super query if and only if $\sigma(r_L \oplus [\mathsf{C}_1]_n)$ is also in the domain of the super query.) Once $y_0^0 = E(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_0]_n))$ is known, the value of $y_1^0 = E(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_1]_n))$ returned by the super query is uniformly picked from $2^n/2 - 1$ possibilities. Therefore,

- The probability to have $\sigma(r_L \oplus [\mathsf{C}_0]_n) \oplus y_0^0 = \sigma(r_L \oplus [\mathsf{C}_1]_n) \oplus y_1^0$ is at most $\frac{1}{2^n/2-1} \leq \frac{4}{2^n}$;
- Conditioned on $E(r_R \| t_1, \sigma(r_L \oplus [\mathsf{C}_0]_n)) = y_0^1$, it holds

$$\begin{aligned}
&\Pr\big[\mathsf{left}_{\lambda-n}(\sigma(r_L \oplus [\mathsf{C}_0]_n) \oplus y_0^1) = \mathsf{left}_{\lambda-n}(\sigma(r_L \oplus [\mathsf{C}_1]_n) \oplus y_1^1)\big] \\
&= \sum_{z \in \{0,1\}^{2n-\lambda}} \Pr\big[y_1^1 = \big(\mathsf{left}_{\lambda-n}(\sigma(r_L \oplus [\mathsf{C}_0]_n) \oplus y_0^1 \oplus \sigma(r_L \oplus [\mathsf{C}_1]_n))\big) \| z\big] \\
&\leq \frac{2^{2n-\lambda}}{2^n/2-1} \leq \frac{4}{2^{\lambda-n}}. \qquad \text{(since } n \geq 2 \text{ implies } 2^n - 2 \leq 2^n/2\text{)}
\end{aligned}$$

Therefore, the probability that the super query returns values such that the pile $\big\{(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_0]_n), y_0^0), ..., (r_R \| t_1, \sigma(r_L \oplus [\mathsf{C}_1]_n), y_1^1)\big\}$ has internal collisions is at most $\frac{4}{2^n} \cdot \frac{4}{2^{\lambda-n}} \leq \frac{16}{2^\lambda}$. The number of choices for a pile is $2^n/4$ for a fixed value r_R. 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

$$\frac{16}{2^\lambda} \cdot \frac{2^n}{4} \leq \frac{4 \cdot 2^n}{2^\lambda}.$$

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

$$\Pr\big[\text{(B-12)}\big] \leq \frac{4 \cdot 2^n}{2^\lambda} \cdot \frac{4q_E}{2^n} \leq \frac{16q_E}{2^\lambda}. \tag{16}$$

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

$$\Pr\big[\text{(B-1)}\big] \leq \Pr\big[\text{(B-11)}\big] + \Pr\big[\text{(B-12)}\big] \leq \frac{4q_E}{2^\lambda} + \frac{16q_E}{2^\lambda} \leq \frac{20q_E}{2^\lambda}. \tag{17}$$

Event (B-2). The two piles being distinct means that $(r'_L, r'_R) \notin \{(r_L, r_R), (r_L \oplus \Delta_{\mathsf{C}}, r_R)\}$ and they consist of 8 distinct queries. Wlog assume that the pile $\big\{(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_0]_n), y_0^0), ..., (r_R \| t_1, \sigma(r_L \oplus [\mathsf{C}_1]_n), y_1^1)\big\}$ 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.

(B-21). Consider the case where $\mathcal{A}$ makes a forward query $E\big(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_0]_n)\big) \to y_0^0$ first. As argued before, the response y_0^0 comes uniformly from at least $2^n/2+2 \geq 2^n/2$ possibilities. Furthermore, the 3 associated forward queries $E\big(r_R \| t_1, \sigma(r_L \oplus [\mathsf{C}_0]_n)\big) \to y_0^1$, $E\big(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_1]_n)\big) \to y_1^0$ and $E\big(r_R \| t_1, \sigma(r_L \oplus [\mathsf{C}_1]_n)\big) \to y_1^1$ have y_1^1, y_1^0, y_1^1 uniformly distributed in at least $2^n/2$ possibilities. By these, for any pair of bits $(b, b') \in \{0,1\}^2$ the probability to have $\mathsf{Com}(r_L \oplus b \cdot \Delta_{\mathsf{C}} \| r_R, t) = \mathsf{Com}(r'_L \oplus b' \cdot \Delta_{\mathsf{C}} \| r'_R, t)$ is at most $\frac{16}{2^{2\lambda}}$, since:

– The probability to have $\sigma(r_L \oplus [\mathsf{C}_b]_n) \oplus E\big(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_b]_n)\big) = \sigma(r'_L \oplus [\mathsf{C}_{b'}]_n) \oplus E\big(r'_R \| t'_0, \sigma(r'_L \oplus [\mathsf{C}_{b'}]_n)\big)$ and $\sigma(r_L \oplus [\mathsf{C}_{\overline{b}}]_n) \oplus E\big(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_{\overline{b}}]_n)\big) = \sigma(r'_L \oplus [\mathsf{C}_{\overline{b'}}]_n) \oplus E\big(r'_R \| t'_0, \sigma(r'_L \oplus [\mathsf{C}_{\overline{b'}}]_n)\big)$ is at most $\frac{1}{2^n/2} \times \frac{1}{2^n/2} \leq 4/2^{2n}$;
– Conditioned on the ideal cipher outputs $E\big(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_b]_n)\big)$, $E\big(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_{b'}]_n)\big)$, $E\big(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_{\overline{b}}]_n)\big)$ and $E\big(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_{\overline{b'}}]_n)\big)$, the probability to have $\mathsf{left}_{\lambda-n}\big(\sigma(r_L \oplus [\mathsf{C}_{\overline{b}}]_n) \oplus E\big(r_R \| t_1, \sigma(r_L \oplus [\mathsf{C}_b]_n)\big)\big) = \mathsf{left}_{\lambda-n}\big(\sigma(r'_L \oplus [\mathsf{C}_{b'}]_n) \oplus E\big(r'_R \| t'_1, \sigma(r'_L \oplus [\mathsf{C}_{b'}]_n)\big)\big)$ and $\mathsf{left}_{\lambda-n}\big(\sigma(r_L \oplus [\mathsf{C}_{\overline{b}}]_n) \oplus E\big(r_R \| t_1, \sigma(r_L \oplus [\mathsf{C}_{\overline{b}}]_n)\big)\big) = \mathsf{left}_{\lambda-n}\big(\sigma(r'_L \oplus [\mathsf{C}_{\overline{b'}}]_n) \oplus E\big(r'_R \| t'_1, \sigma(r'_L \oplus [\mathsf{C}_{\overline{b'}}]_n)\big)\big)$ is at most $\big(\frac{2}{2^{\lambda-n}}\big)^2$;

When $\mathcal{A}$ makes a backward query $E^{-1}(r_R \| t_0, y_0^0) \to x$, the response $x = \sigma(r_L \oplus [\mathsf{C}_0]_n)$ is uniform in a set of size at least $2^n/2 + 2 \geq 2^n/2$. Again, the 3 subsequent forward queries $E\big(r_R \| t_1, \sigma(r_L \oplus [\mathsf{C}_0]_n)\big)$, $E\big(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_1]_n)\big)$ and $E\big(r_R \| t_1, \sigma(r_L \oplus [\mathsf{C}_1]_n)\big)$ 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 $\big\{(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_0]_n), y_0^0), ..., (r_R \| t_1, \sigma(r_L \oplus [\mathsf{C}_1]_n), y_1^1)\big\}$ (piles in normal queries) is at most q_E, while the number of choices for

$$\big\{(r'_R \| t'_0, \sigma(r'_L \oplus [\mathsf{C}_0]_n), y_0^{0'}), ..., (r'_R \| t'_1, \sigma(r'_L \oplus \mathsf{C}_1), y_1^{1'})\big\}$$

(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,

$$\Pr\big[\text{(B-21)}\big] \leq 4 \cdot q_E \cdot 2q_E \times \frac{16}{2^{2\lambda}} \leq \frac{128{q_E}^2}{2^{2\lambda}}. \tag{18}$$

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

– The two queries $E(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_0]_n))$ and $E(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_1]_n))$ have answers uniformly distributed in $2^n/2$ and $2^n/2 - 2 + 1 \geq 2^n/4$ possibilities respectively;
– The two queries $E(r_R \| t_1, \sigma(r_L \oplus [\mathsf{C}_0]_n))$ and $E(r_R \| t_1, \sigma(r_L \oplus [\mathsf{C}_1]_n))$ have answers uniformly distributed in $2^n/2$ and $2^n/2 - 2 + 1 \geq 2^n/4$ 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 $\mathsf{Com}(r_L \oplus b \cdot \Delta_{\mathsf{C}} \| r_R, t) = \mathsf{Com}(r'_L \oplus b' \cdot \Delta_{\mathsf{C}} \| r'_R, t')$ is at most $(8/2^\lambda)^2$.

The number of choices for

$$\big\{(r_R \| t_0, \sigma(r_L \oplus [\mathsf{C}_0]_n), y_0^0), ..., (r_R \| t_1, \sigma(r_L \oplus [\mathsf{C}_1]_n), y_1^1)\big\}$$

(piles in super queries) is at most q_E, while the number of choices for

$$\big\{\big(r'_R \| t'_0, \sigma(r'_L \oplus [\mathsf{C}_0]_n), y_0^0\big), ..., \big(r'_R \| t'_1, \sigma(r'_L \oplus [\mathsf{C}_1]_n), y_1^1\big)\big\}$$

(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,

$$\Pr\big[\text{(B-22)}\big] \leq 4 \cdot q_E \cdot 2q_E \times \frac{64}{2^{2\lambda}} \leq \frac{512{q_E}^2}{2^{2\lambda}}. \tag{19}$$

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

$$\Pr\big[\text{(B-2)}\big] \le \Pr\big[\text{(B-21)}\big] + \Pr\big[\text{(B-22)}\big] \le \frac{128{q_E}^2}{2^{2\lambda}} + \frac{512{q_E}^2}{2^{\lambda}} \le \frac{640{q_E}^2}{2^{2\lambda}}. \quad (20)$$

Summary. Gathering Eq. (17) and (20), the probability that $\mathcal{A}$ finds a collision from its query history is bounded by

$$\frac{20q_E}{2^{\lambda}} + \frac{640{q_E}^2}{2^{2\lambda}} \quad (21)$$

as claimed.

D Proof of Lemma 6

Proof. Suppose that the adversary returns (m', t') such that $\mathsf{Com}(m', t') = c$. We assume without loss of generality that the pile related to m appears in the adversary's transcript $\mathcal{Q}_E$ (since if not we can append that pile to $\mathcal{Q}_E$). We inherit the symbols and proof strategy of Lemma 3 to grant the adversary free queries such that the entries in $\mathcal{Q}_E$ 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

$$\big\{(k_0 = x_R \| t'_0, x_0, y_0^0), (k_1 = k_0 \oplus \Delta_k, x_0, y_0^1), (k_0, x_1 = x_0 \oplus \Delta_p, y_1^0), (k_1, x_1, y_1^1)\big\}$$

and define $m_0 = \sigma^{-1}(x_0) \oplus [\mathsf{C}_0]_n \| x_R$ and $m_1 = \sigma^{-1}(x_1) \oplus [\mathsf{C}_0]_n \| x_R$. By the definition of this pile, we have $m' = m_0$ or $m' = m_1$. And since $\mathsf{Com}(m', t') = c$, we have the following two cases respectively.

$$c = x_0 \oplus y_0^0 \| \mathsf{left}_{\lambda-n}(x_0 \oplus y_0^1) \| x_1 \oplus y_1^0 \| \mathsf{left}_{\lambda-n}(x_1 \oplus y_1^1) \quad (22)$$

$$c = x_1 \oplus y_1^0 \| \mathsf{left}_{\lambda-n}(x_1 \oplus y_1^1) \| x_0 \oplus y_0^0 \| \mathsf{left}_{\lambda-n}(x_0 \oplus y_0^1) \quad (23)$$

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 $\mathcal{Q}_E$ 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

$$\Pr[Eq.\ (22)] \le \frac{1}{2^n/2+2} \cdot \frac{2^{2n-\lambda}}{2^n/2+2} \cdot \frac{1}{2^n/2+1} \cdot \frac{2^{2n-\lambda}}{2^n/2+1} \le \frac{16}{2^{2\lambda}}.$$

Similarly, we have $\Pr[Eq.\ (23)] \le \frac{16}{2^{2\lambda}}$. By a union bound over all the q_E queries, we conclude that in this case $\Pr[\mathsf{Com}(m', t') = c] \le \frac{32q_E}{2^{2\lambda}}$.

– Inversion After Super Query.
For each pile returned by the super query, we have

$$\Pr[Eq.\ (22)] \leq \frac{1}{2^n/2} \cdot \frac{2^{2n-\lambda}}{2^n/2-1} \cdot \frac{1}{2^n/2} \cdot \frac{2^{2n-\lambda}}{2^n/2-1} \leq \frac{64}{2^{2\lambda}}.$$

Notice that here we use the fact that since $n > 1$, $2^n/2 - 1 \geq 2^n/4$. We also have $\Pr[Eq.\ (23)] \leq \frac{64}{2^{2\lambda}}$. 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 $\Pr[\mathsf{Com}(m', t') = c] \leq \frac{128q_E}{2^{2\lambda}}$.

In summary, we have $\Pr[\mathsf{Com}(m', t') = c] \leq \frac{32q_E}{2^{2\lambda}} + \frac{128q_E}{2^{2\lambda}} = \frac{160q_E}{2^{2\lambda}}$.