Towards Formal Security Proofs of MQOM

小菅 悠久 (Haruhisa Kosuge)[1] and 草川 恵太 (Keita Xagawa)[2]

[1] NTT Social Informatics Laboratories, Japan
hrhs.kosuge@ntt.com
[2] Technology Innovation Institute, UAE
keita.xagawa@tii.ae

October 27, 2026

Abstract. Recent MPC-in-the-Head (MPCitH) signatures increasingly rely on aggressive GGM-tree optimizations to reduce signature size and computational cost, culminating in *secret-key-root correlated GGM trees* as used in SBC (Huth and Joux, CRYPTO 2024), MQOM (NIST PQC Standardization for Additional Signature Round-2, 2024), and rBN++ (Kim, Lee, and Son, EUROCRYPT 2025). While this technique yields substantial compression, it introduces a dependency loop in the proof (Kosuge and Xagawa, CiC 2026(3)).

We analyze MQOM and resolve this circularity by providing EUF-CMA security proofs for a slightly modified variant. The modifications consist of incorporating a salt into the input of a hash function and modifying the evaluation domain to ensure that all evaluation points are nonzero.

For the resulting variant of MQOM, we prove EUF-CMA security for the GF(2) parameter sets in the random-oracle plus ideal-cipher model, assuming EUF-NMA security, standard one-wayness, and a heuristic conjecture on matrices. Our proof combines the H-coefficient technique with one-wayness, which may be of independent interest.

We also prove EUF-CMA security in the (quantum) random-oracle model, where block-cipher-based hash functions are modeled as random oracles. The proof relies on EUF-NMA security and partial-guessing one-wayness (Feneuil and Rivain, ASIACRYPT 2026), which can be reduced classically to partial-domain one-wayness.

Keywords: MPC-in-the-Head signature, secret-key-root correlated GGM tree, H-coefficient technique, quantum random oracle model

Table of Contents

1 Introduction 2
2 Preliminaries 7
3 Security of scGGM-based AVC of modified MQOM in the (Q)ROM 12
4 Security of scGGM-based AVC of modified MQOM in the ICM 16
5 EUF-CMA Security of MQOM 23
A MQOM 31
B Summary of Oracles and Adversaries 33
C Variations of One-wayness 33
D Review of cGGM-based AVC 35
E Proof of Theorem 3.1 35
F Examples of H-Coefficient Technique with One-wayness 40

This article is based on an earlier article: Haruhisa Kosuge and Keita Xagawa: Towards Formal Security Proofs of MQOM, ASIACRYPT 2026, © IACR 2026.

1 Introduction

MPC-in-the-head signatures: One of the standard methods for constructing digital signatures is the *Fiat–Shamir* (FS) transform [FS87], which converts identification (ID) schemes into signature schemes. A prominent paradigm for constructing such ID schemes from arbitrary hard problems is *MPC-in-the-Head* (MPCitH) [IKOS07, GMO16]. Owing to its generality and efficiency, the MPCitH paradigm has undergone rapid development in recent years and has become a central approach for constructing *post-quantum cryptographic* (PQC) signature schemes, which aim to maintain security even against quantum adversaries. The practical relevance of MPCitH-based signatures was further highlighted by the 2022 NIST call for additional digital signature schemes [NIS22], to which nine MPCitH-based proposals were submitted. Among them, six schemes advanced to Round 2, and FAEST [BBB+24], MQOM [BBFR24], and SDitH [ABB+24b] advanced to Round 3 [ABC+26], demonstrating the maturity and viability of the MPCitH paradigm in practice. From a design perspective, improving signature size and efficiency has been a major driving force behind recent progress in MPCitH-based schemes. Two prominent variants of MPCitH have emerged in this context: *VOLE-in-the-Head* (VOLEitH) [BBD+23] and *Threshold Computation in the Head* (TCitH) [FR23, FR25].

GGM tree and (secret-key-root) correlated GGM tree: MPCitH signature schemes and their variants commonly use all-but-one vector commitments (AVC) [BBD+23], which are often implemented using the Goldreich–Goldwasser–Micali (GGM) tree [GGM86] for efficient and consistent partial disclosure of secret values. Since the tree's structure directly impacts both computational cost and signature size, optimizing GGM-tree techniques is a key research focus.

One promising technique is *half tree technique* for constructing *correlated GGM (cGGM) trees*, proposed by Guo et al. [GYW+23] in the context of pseudorandom correlation generators. This technique roughly halves the computational cost of the GGM trees. Several signature schemes have already employed cGGM-based AVCs [CLY+24, BCD25, WKK+25].

A further optimization for cGGM-based AVCs is the *secret-key-root cGGM tree*, which we call an *scGGM* tree. Here, all cGGM trees share a common root, namely a prefix of the secret key. This allows a reduction in the signature size. SBC [HJ24], MQOM [BBFR24], and rBN^{++} [KLS25a] employ this technique and SBC$_{\mathsf{VOLE}}$ [HJ25] further extended this technique to the scGGM forest.

Missing security proofs: To the best of our knowledge, the SBC paper [HJ24] and the MQOM specification [BBFR25] do not include a security proof to justify their use of scGGM-based AVC. Moreover, while Kim, Lee, and Son [KLS25a] presented a security proof for their proposed signature scheme rBN^{++}, which employs the scGGM tree, Kosuge and Xagawa [KX26a] later pointed out a gap in that proof and Kim et al. later removed the "secret-key-root" optimization [KLS24, Ver.20251120:150611]. Consequently, at present, there is no available proof technique that can be directly applied to establish the security of MQOM and the original version of rBN^{++}. NIST has also explicitly recognized this issue in their report [ABC+26, Section 3.7]:

> Like the other schemes that have undergone significant recent changes, MQOM will require careful further analysis. Currently, security proofs of MQOM in the Random Oracle Model (ROM) and QROM might not be considered mature or complete. Of particular interest to NIST is continued refinement of the provable analysis of MQOM security, particularly in models that capture the power of quantum attackers [KX26a, FR26, KX26b]. [3]

This leads us to the following research question:

Can MPCitH signatures based on scGGM trees be proven secure?

1.1 Our Contribution

We analyze MQOM and develop a proof technique that resolves the circularity arising from the secret-key-root correlation for slightly modified MQOM. We incorporate a salt into the input of the hash function Hash_4 and modify the evaluation domain so that all evaluation points are nonzero; we denote the resulting variant by $\mathsf{MQOM}_{\mathtt{salt},\Omega}$. We denote the variant with only the evaluation-domain modification by MQOM_{Ω}. The purpose of adding salt to the input of Hash_4, whose output determines the hidden shares, is to eliminate the need for

[3] Reference numbers in the quotation have been adapted to match our bibliography.

a heuristic min-entropy argument. This modification simplifies the proof, while the circularity caused by the secret-key-root correlation remains and is handled separately.

We formulate a simulation notion for scGGM-based AVCs in MQOM, which can be viewed as a computational honest-verifier zero-knowledge property. This notion abstracts the portion of a signing transcript generated from an scGGM tree and characterizes when a real opening transcript can be replaced by a simulated one that is independent of the secret-key root. Observe that the simulatability of the scGGM-based AVC in MQOM_{Ω} is the same as that of $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ since it is irrelevant to the inclusion of `salt` to the hash function.

We prove the simulatability of the scGGM-based AVC in MQOM_{Ω} in the *ideal cipher model (ICM)* using the H-coefficient technique [Pat09, CS14]. The proof relies on the standard one-wayness assumption together with a heuristic matrix-rank conjecture, which we validate experimentally. Due to a technical restriction explained later, this result currently holds only for the GF(2) parameter sets.

We also prove this simulatability for the scGGM-based AVC in MQOM_{Ω} in the ROM and QROM by modeling the block-cipher-based hash function as a random oracle, but without the heuristic matrix-rank conjecture. The underlying assumption is the partial-guessing one-wayness assumption.

This is a heuristic modeling choice since the block-cipher-based hash function in MQOM is based on the Davies–Meyer construction, which is *distinguishable* from a random oracle. We do not claim that the concrete construction is indifferentiable from a random oracle, but the proof can be considered as a sanity check; the resulting proof may be viewed as heuristic evidence that generic quantum attacks do not invalidate the EUF-CMA security of MQOM. The underlying assumption is partial-guessing one-wayness (PGOW) [FR26],[4] which we further extend to the quantum setting. We note that the PGOW assumption holds for PPT and QPT adversaries with classical guessing queries if partial-domain one-wayness (PDOW) [OP01] holds.

The salt modification and the evaluation-domain restriction play different roles in our modular analysis. The former provides fresh entropy for the adaptive reprogramming of Hash_4, avoiding a heuristic min-entropy assumption. The latter ensures that all evaluation points are invertible, as required both by the simulated procedures in the EUF-NMA-to-EUF-CMA reduction and by the ICM or (Q)ROM simulatability proofs. Accordingly, those simulatability results are established for MQOM_{Ω}. Combining the corresponding simulatability result with the reduction yields an end-to-end security theorem for $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ in each setting.

We note that the above proof for $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ can be carried over to MQOM_{Ω} if one assumes heuristically that the input to Hash_4 has sufficiently high min-entropy. While we regard this assumption as plausible, as discussed in [FR26], we do not establish it formally. In particular, the latter EUF-CMA security result in the QROM provides heuristic evidence for the security of MQOM_{Ω} against quantum adversaries.

To explain the challenges and our solutions, we first review the GGM tree in AVC and its optimizations.

Review of GGM-tree techniques: We briefly review the GGM tree and its variants. Suppose that a signer generates τ batches of $N = 2^d$ seeds.

In a standard GGM tree, the signer chooses a root $\mathtt{node}[1] \leftarrow \{0,1\}^\lambda$ at first. For each internal node index $k \in [1 : N]$, the signer derives two children as $(\mathtt{node}[2k], \mathtt{node}[2k+1]) := \mathsf{PRG}(\mathtt{node}[k])$, where $\mathsf{PRG} : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$. It then obtains the leaf nodes $\mathtt{node}[N], \ldots, \mathtt{node}[2N-1]$ and defines $\mathtt{seed}[i] := \mathtt{node}[N+i]$ for all $i \in [N]$. The signer further computes commitments $\mathtt{com}[i] := \mathsf{H}(\mathtt{seed}[i])$ and random tapes $\mathtt{tape}[i] := \mathsf{XOF}(\mathtt{seed}[i])$. See Figure 1 for graphical explanations. For a hidden index i^*, the signer reveals the sibling nodes along the path from the leaf $\mathtt{node}[N + i^*] = \mathtt{seed}[i^*]$ to the root, together with $\mathtt{com}[i^*]$. From this opening data, the verifier reconstructs all seeds $\mathtt{seed}[i]$ for $i \neq i^*$, recomputes their commitments $\mathtt{com}[i]$ and tapes $\mathtt{tape}[i]$ for $i \neq i^*$. In typical MPCitH signatures, tapes `tape` are used to form secret shares, say, an N-out-of-N sharing of the secret key $x \in \{0,1\}^{|x|}$, which is implemented by publishing an offset $\Delta_x := x \oplus \bigoplus_{i\in[N]} \mathtt{tape}[i][0 : |x|]$, where $X[i : j]$ is the substring of X containing bits with indices $i, i+1, \ldots, j-1$.

We next review correlated GGM (cGGM) tree [GYW+23]: To reduce the computational cost of the GGM tree, Guo et al. [GYW+23] introduced the use of *circular correlation robust (CCR)* hash functions [CKKZ12] $\mathsf{H}_e : \{0,1\}^\lambda \to \{0,1\}^\lambda$ for $e \in [\tau]$ within the GGM tree. Very roughly speaking, they replace PRG with $\mathsf{PRG}' : s \mapsto (\mathsf{H}_e(s), \mathsf{H}_e(s) \oplus s)$ in the e-th GGM tree. By the construction, we have

$$\delta = \mathtt{node}[2] \oplus \mathtt{node}[3] = \cdots = \bigoplus_{i\in[2^d:2^{d+1}]} \mathtt{node}[i] = \bigoplus_{i\in[N]} \mathtt{seed}[i]. \tag{1}$$

Due to this correlation, this GGM tree is dubbed as *correlated* GGM tree. Note that the cGGM trees can share the *same* root $\delta \leftarrow \{0,1\}^\lambda$. In particular, for every $e \in [\tau]$ the leaves satisfy $\bigoplus_{i\in[N]} \mathtt{seed}[i] = \delta$, which induces a correlation across all the τ trees.

[4] PGOW captures the hardness of finding the secret-key prefix δ, while giving the adversary access to a guessing oracle that, on input δ', returns whether $\delta' = \delta$.

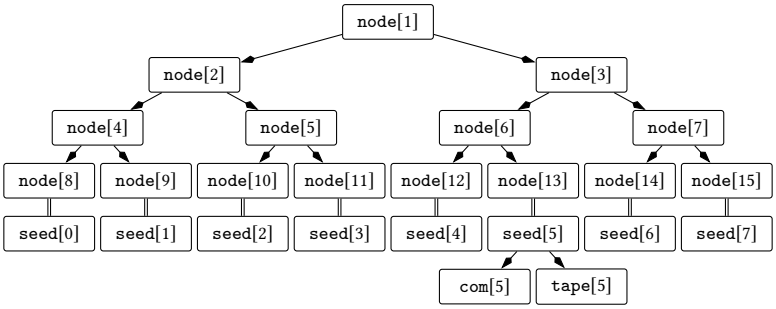


Fig. 1. Example of all-but-one commitment scheme based on the (c)GGM tree with $N = 8$. For $i = 0, \ldots, 7$, we let $\mathtt{seed}[i] \coloneqq \mathtt{node}[N + i]$. If we consider cGGM-based AVC, we do not have $\mathtt{node}[1]$.

In *secret-key-root correlated GGM (scGGM) tree* [HJ24, KLS25a], the signer sets a prefix of its secret key as the root, that is, $\delta \coloneqq x[0 : \lambda]$. In addition, it sets $\Delta_x \coloneqq x \oplus \bigoplus_{i\in[N]}(\mathtt{seed}[i] \,\|\, \mathtt{tape}[i])$, where $\mathtt{tape}[i] \coloneqq \mathsf{XOF}(\mathtt{seed}[i])$. Since $\bigoplus_{i\in[N]} \mathtt{seed}[i] = \delta$, the first λ bits of Δ_x always cancel and need not be included in the signature, yielding a reduction of $\tau\lambda$ bits in the signature size.

Role of scGGM-based AVC simulatability in signature proofs: Our main technical contribution is a simulatability theorem for the scGGM-based AVC used in MQOM. This theorem is motivated by the following standard simulation-based proof strategy for MPCitH signatures [KX26a].

- Game 0 is the original EUF-CMA game.
- Game 1 reprograms the Fiat–Shamir hash so that the signing oracle can sample the challenge value in advance.
- Game 2 computes the hidden indices i^* at the beginning of each signing query.
- Game 3 replaces the real scGGM-based AVC opening data associated with i^* by simulated AVC data. This step changes the opening data of AVC, the commitment $\mathtt{com}[i^*]$, and the hidden seed/tape $\mathtt{seed}[i^*], \mathtt{tape}[i^*]$.
- Game 4 replaces the remaining secret-key-dependent masked values, such as the offset Δ_x derived using $\mathtt{tape}[i^*]$, by random values. After this replacement, the signing oracle no longer needs the secret key.

The simulation step after Game 2 is precisely where scGGM-based AVC simulatability is needed.

For scGGM-based AVCs, the main difficulty is that the usual two-step simulation argument for GGM-based or cGGM-based AVCs becomes circular, because the correlation root is a prefix of the secret key [KX26a]. For ordinary GGM-based AVCs, or cGGM-based AVCs with an independently sampled correlation root δ, one can first randomize the opening data, commitments, and $(\mathtt{seed}[i^*], \mathtt{tape}[i^*])$ as in Game 3, and then randomize the values masked by $\mathtt{tape}[i^*]$ as in Game 4. In an scGGM tree, however, $\mathtt{tape}[i^*]$ is derived from the secret-key prefix δ and is also used to mask secret-key-dependent values such as Δ_x. Thus, randomizing $\mathtt{tape}[i^*]$ requires arguing that δ is hidden, while hiding the remaining secret-key-dependent values relies on the randomness of $\mathtt{tape}[i^*]$.

Our scGGM-based AVC simulatability theorem resolves this circularity by replacing the opening data, commitments, and the masked secret-key-dependent values simultaneously. Equivalently, it justifies the combined transition from Game 2 directly to Game 4, rather than treating it as two independent hops.

Simulatability of scGGM-based AVC in MQOM_Ω in the ICM: MQOM uses a block-cipher-based hash function[5] in its scGGM tree. We prove its security for MQOM_Ω in the ICM. This modification is required for a technical reason: every evaluation point $\omega \in \Omega$ must be nonzero.

Our approach is to apply the well-known H-coefficient technique [Pat09, CS14] to establish simulation security. To this end, we define good and bad transcripts consisting of oracle queries and their corresponding answers. Among other requirements for applying the H-coefficient technique, we must upper-bound the probability that the transcript in Game 4 is bad. If δ were perfectly hidden from the adversary, this analysis would closely follow the information-theoretic analysis of (tweakable) CCR hashes [GKWY20, GKW+20]. In our setting, however, the adversary *knows* the public key, which may reveal δ to an unbounded adversary.

[5] MQOM adopts the Davies–Meyer-like (DM-like) tweakable hash $\mathsf{H}(s, t) = \mathsf{EncFF}(s, t) = E(t, s) \oplus \psi(s)$, where $E : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ is a block cipher and ψ is a linear orthomorphism.

Our key observation is that the bad-transcript probability can instead be bounded by the *computational* advantage associated with one-wayness. More concretely, we construct a polynomial-time reduction that simulates Game 4 and recovers $\delta \in \{0,1\}^\lambda$ or the secret witness $x \in \mathbb{F}^n$ whenever the adversary produces a bad transcript, where $\mathbb{F} = \mathrm{GF}(2^k)$ for some k.[6] In one case, the reduction algorithm obtains from a bad transcript a relation $z = M \cdot x + c \in \mathbb{F}^{\lambda/k}$ for the secret witness $x \in \mathbb{F}^n$, where $kn > \lambda$ and $M \in \mathbb{F}^{(\lambda/k)\times n}$ is determined by the adversary's queries and the public key of MQOM.

For this reduction to be useful, we need $kn - \lambda$ to be sufficiently small. We also conjecture that the rank of M is $\lambda/k - O(1)$; experimentally, we always observe $\mathrm{rank}(M) = \lambda/k$. Unfortunately, the former requirement is satisfied only by the parameter sets with $\mathbb{F} = \mathrm{GF}(2)$.[7]

To the best of our knowledge, this is the first application of the H-coefficient technique in which the bad-transcript probability is bounded via computational hardness rather than purely information-theoretic arguments. We believe that this observation may be of independent interest.

Simulatability of scGGM-based AVC in MQOM_Ω in the (Q)ROM: In the ROM, the real and simulated experiments differ only at random-oracle inputs used to derive the opening path associated with i^*, the commitment $\mathtt{com}[i^*]$, and the hidden seed/tape $\mathtt{seed}[i^*], \mathtt{tape}[i^*]$; we bound this difference by reducing it to PGOW. Replacing the oracle outputs at these inputs allows the simulator to sample these AVC opening objects independently of the secret-key root δ. Since $\mathtt{tape}[i^*]$ is the hidden randomness that masks the offset Δ_x and the other secret-key-dependent values in the signature, these masked values can then be replaced by uniform values. Therefore, in the ROM, the distinguishing advantage is bounded by the probability that the adversary queries one of these hidden random-oracle inputs. This probability is in turn tightly bounded by the success probability of a PGOW adversary that runs the simulated experiment.

In the QROM, we use the semi-classical one-way-to-hiding (SC-O2H) lemma [AHU19]. The O2H technique turns a distinguishing advantage between two oracle worlds into the ability to find an input on which the two random oracles differ. We define two random oracles, where the two oracles may differ only on inputs in a set $\mathcal{S}$. We define f by summing the input node and the related node values as in Equation 1. Then, if the input belongs to $\mathcal{S}$, the output of f is δ. Using the guessing oracle of the PGOW game, the reduction can check whether a candidate input belongs to $\mathcal{S}$. This membership test allows the reduction to implement the punctured execution in the SC-O2H lemma, answering with the simulated oracle outside $\mathcal{S}$ and stopping at the first detected query in $\mathcal{S}$. The SC-O2H lemma gives a contribution $2\sqrt{Q_{\mathsf{H}_{\mathsf{avc}}}\mathsf{Adv}^{\mathrm{pgow}}}$, in addition to the internal-collision loss.

Summary: Wrapping up, we formally prove the following theorems:

- $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ is EUF-CMA-secure in the ROM+ICM for the GF(2) parameter sets, assuming EUF-NMA security, one-wayness, and the matrix-rank conjecture (in addition to standard assumptions on hash functions and PRGs).
- $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ is EUF-CMA-secure in the (Q)ROM when the scheme is EUF-NMA secure, the block-cipher-based hash function is modeled as a random oracle, and the PGOW assumption holds (again, in addition to standard assumptions on hash functions and PRGs). We do not need to adopt the matrix-rank conjecture.

Open problems: Several interesting problems remain: The first challenge is to show the security of MQOM (or its variants) in the ROM+ICM under the standard one-wayness assumption, without our heuristic matrix-rank conjectures. The second one is to extend the current proof to the QROM plus the *quantum ideal-cipher model (QICM)*. Currently, we have fewer tools in the QICM than in the QROM, and the extension will require developing such tools for the QICM. Another important direction is to show the security of other signature schemes, rBN^{++} with block-cipher-based hashes [KLS25a] and $\mathsf{SBC}_{\mathsf{VOLE}}$ [HJ25].

1.2 Related Works

Study of GGM trees in the MPCitH signatures: Katz, Kolesnikov, and Wang [KKW18] used a binary seed tree, motivated by the NNL stateless-revocation scheme [NNL01], to reduce the communication needed to reveal all

[6] MQOM has three categories, $\mathbb{F} = \mathrm{GF}(2)$, GF(16), and GF(256) as in Table 1.

[7] See parameter sets summarized in Table 1. As in [FR26], a reduction for all parameter sets can be obtained by employing the RGOW assumption of [FR26].

but one of the parties' seeds. This seed-tree optimization was subsequently adopted in MPCitH signature schemes such as Picnic (ver. 2.0) [ZCD+19, Section 7.3.1] and BBQ [DDOS19], where hash functions are used as doubling PRGs. Bui et al. [BCC+24] proposed a construction based on a doubling PRG based on fixed-key block ciphers, $x \mapsto (\pi_0(x) \oplus x, \pi_1(x) \oplus x)$, to compute two child nodes.

On the cGGM tree, Guo et al. [GYW+23] proposed the half-tree technique and cGGM tree using $x \mapsto (\mathsf{H}_e(x), \mathsf{H}_e(x) \oplus x)$. Several signature schemes already employed the cGGM tree with the DM-like hash $\mathsf{H}_e(x) := \pi_e(x) \oplus \psi(x)$, where ψ is a special linear function: ReSolveD [CLY+24], Bui et al. [BCD25], BACON [WKK+25], and rVOLEitH [KLS25b]. We note that the DM-like hash in the random permutation model or ICM is *distinguishable* from the random oracle [CDMP05, KM07].

The scGGM tree is adopted in SBC [HJ24], rBN^{++} [KLS25a], and MQOM [BBFR24], where MQOM used the DM-like hash H_e. In $\mathsf{SBC}_{\mathsf{VOLE}}$, Huth and Joux [HJ25] proposed the scGGM forest. They employed $\mathsf{H}_e(x) := \pi_{e,0}(x) \oplus \pi_{e,1}(x)$ with two random permutations $\pi_{e,0}$ and $\pi_{e,1}$, which is indifferentiable from the random oracle (see, e.g., [GBJ+23]). Thus, our techniques for $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ could formally prove the security of $\mathsf{SBC}_{\mathsf{VOLE}}$ in the ROM.

Security proofs for MPCitH signatures based on 3*-round protocols:* The Fiat–Shamir transform [FS87] converts a 3-round protocol into a signature scheme. Don et al. [DFMS22b] established a tight online-extractability for commit-and-open Σ-protocols by leveraging the notion of extractable QROM, a refinement built upon Zhandry's compressed QRO technique [Zha19]. In a companion work [DFMS22a], they proved *tight* security in the QROM for both a NIZK proof of knowledge and a signature scheme obtained via the FS transform applied to commit-and-open Σ-protocols (e.g., Picnic [CDG+17]); their argument proceeds through the framework of Chung et al. [CFHL21], which itself relies on compressed QRO [Zha19]. For 5/7-round versions, we refer the reader to [AHJ+23, HJMN24, BBB+25, KX26a].

Concurrent work: Feneuil and Rivain [FR26] independently and concurrently provide security proofs for MQOM (ver. 2.0), which is essentially equivalent to the version 2.1 considered in this work.

They study MQOM and three variants of MQOM depending on how Hash_e in the GGM tree, Commit, and XOF for the tape computation are instantiated: (1) random oracles, (2) Davies–Meyer–like hash functions, which are the original, (3) counter-mode constructions, and (4) Davies–Meyer–like hash functions combined with XOR-of-Permutations (XoP) constructions [GBJ+23, Din24].

They introduce a notion of simulation security for scGGM-based AVCs in MQOM and prove it using the H-coefficient technique, similarly to our proof for MQOM_{Ω}. To establish simulation security, they introduce two new assumptions, PGOW and relation-guessing one-wayness (RGOW). In the RGOW security game, an adversary is given access to relations $g : \mathbb{F}^n \to \{0,1\}^\lambda$ and succeeds if it queries a generated relation g together with a value $\hat{x}_0$ such that $g(x) = \hat{x}_0$. Their security results rely on different assumptions depending on the instantiation: the first, third, and fourth variants are proven secure under PGOW for the key-generation algorithm, while the second, i.e., the original one, additionally requires RGOW. Finally, they analyze the hardness of PGOW and RGOW and provide theoretical evidence supporting these assumptions. In particular, a part of their analysis for RGOW is closely related to the conjecture used in our work.

Based on this simulability, they proved MQOM's EUF-CMA security in the ROM. Due to differences in the handling of salts in Hash_4, their proof additionally relies on a heuristic argument on the min-entropy of the hash input.

1.3 Organization

Section 2 reviews basic notions and notation, including (s)cGGM-based AVC, and MQOM. Section 3 presents the security of scGGM-based AVC for MQOM in the (Q)ROM, with the full proof deferred to Appendix E. Section 4 establishes the security of scGGM-based AVC for MQOM_{Ω} in the ideal cipher model (ICM).

Building on these AVC security results, Section 5 proves the EUF-CMA security of $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ in the (Q)ROM, and derives corollaries for $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ in the ROM+ICM and those for $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ in the ROM and in the QROM. Appendix A reviews the parameter sets and the verification algorithm of MQOM. Appendix B summarizes the oracles and adversaries used in our main security proofs. Appendix C introduces several variants of one-wayness and discusses their relationships. Appendix D reviews the security proof of the cGGM-based AVC. Appendix E provides the full proof of the security of scGGM-based AVC for MQOM in the (Q)ROM.

1.4 Version Notes

In what follows, $\mathsf{MQOM}[F]$ and $\mathsf{MQOM}_{\mathtt{salt}}[F]$ denote the variants of MQOM and $\mathsf{MQOM}_{\mathtt{salt}}$, respectively, in which $\mathsf{H}_{\mathsf{avc}}$ is instantiated by F.

Version of 2026-03-31. The version dated 2026-03-31 considered two variants. The first was $\mathsf{MQOM}_1 := \mathsf{MQOM}_{\mathtt{salt}}[\mathsf{RO}]$, with the original security parameter. For MQOM_1, we proved EUF-CMA security in the (Q)ROM under EUF-NMA security and PDOW in the classical setting, or domain-extension onewayness (DEOW) together with PRF security in the quantum setting.

The second was $\mathsf{MQOM}_2 := \mathsf{MQOM}_{\mathtt{salt}}[\mathsf{EncFF}^E]$, with an enlarged security parameter. For MQOM_2, we proved EUF-CMA security in the ROM+ICM under EUF-NMA security, standard one-wayness, and a special conjecture on certain matrices. This result applied only to the GF(2) parameter sets and to the three-round setting.

That version also contained detailed proofs of the EUF-NMA security of MQOM_1 and MQOM_2.

Version of 2026-05-27. (Submitted to Asiacrypt 2026) The version dated 2026-05-27 reorganizes the proof by first proving a hiding/simulatability property of the underlying scGGM-based AVC and then deriving the EUF-CMA security of $\mathsf{MQOM}_{\mathtt{salt}}$ from it. For $\mathsf{MQOM}_{\mathtt{salt}}[\mathsf{EncFF}^E]$, we prove EUF-CMA security in the ROM+ICM under EUF-NMA security, standard one-wayness, and the same special conjecture. This result applies to the GF(2) parameter sets, now covers the five-round setting, and no longer requires increasing the security parameter.

For $\mathsf{MQOM}_{\mathtt{salt}}[\mathsf{RO}]$, we prove EUF-CMA security in the (Q)ROM under EUF-NMA security and PGOW. Classically, PGOW follows from PDOW, while in the quantum setting, it follows from PRF security and DEOW.

Version of 2026-09-18. (Based on the camera-ready version for Asiacrypt 2026) The version dated 2026-09-18 replaced $\mathsf{MQOM}_{\mathtt{salt}}[\mathsf{EncFF}^E]$ with $\mathsf{MQOM}_{\mathtt{salt},\Omega}[\mathsf{EncFF}^E]$. This change was made because a reviewer at ASIACRYPT 2026 pointed out that ω_{i^*} must be nonzero for the ROM+ICM proof. We also added a discussion of the security implications of our theorems in Section 5. This version is a major revision of the camera-ready version.

Version of 2026-09-21. The version dated 2026-09-21 further replaced $\mathsf{MQOM}_{\mathtt{salt}}[\mathsf{RO}]$ with $\mathsf{MQOM}_{\mathtt{salt},\Omega}[\mathsf{RO}]$, since the same issue also arises in the EUF-CMA-to-EUF-NMA reduction.

Version of 2026-10-27. This revision corrects the computation of x_0 in MQOM_{Ω}. The original MQOM specification uses an optimization that assumes $\omega_0 = 0$. For the modified evaluation domain, we restore the term $-\omega_0 \Delta_x$ and use $x_0 := -\omega_0 \Delta_x - \sum_{i \in [N]} \omega_i \tilde{x}_i$, with the corresponding correction to verification.

We also revise the QROM proof of AVC simulatability by replacing the double-sided O2H argument with the semi-classical O2H lemma of Ambainis, Hamburg, and Unruh [AHU19]. The previous argument did not justify transferring the double-sided extractor between the real-transcript and simulated-transcript experiments. The revised reduction uses a punctured oracle and incurs an additional square-root factor in the number of random-oracle queries. We update the corresponding theorem and EUF-CMA corollary accordingly.

Furthermore, we explicitly account for collisions between hidden oracle inputs and inputs used during reconstruction, including collisions caused by zero hidden internal nodes. We clarify the field embeddings and binary representations used in the algebraic analysis and correct the reconstruction-query counts.

Finally, we withdraw the previously claimed implication from DEOW and PRF security to quantum PGOW, since the proposed reduction did not simulate the exact guessing oracle. The QROM results remain stated under the PGOW assumption. We also correct the classical PGOW-to-PDOW reduction to account for the adversary's final output, giving a loss of $Q_{\mathsf{guess}} + 1$.

2 Preliminaries

The security parameter is denoted by $\lambda \in \mathbb{Z}^+$. We use the standard O-notation. DPT, PPT, and QPT stand for deterministic, probabilistic, and quantum polynomial time, respectively.

For $n \in \mathbb{Z}^{\geq 0}$, we let $[n] := \{0, \ldots, n-1\}$, which is $\emptyset$ if $n = 0$. For $n_1, n_2 \in \mathbb{Z}^{\geq 0}$, we use the half-open interval notation $[n_1 : n_2] := \{n_1, n_1 + 1, \ldots, n_2 - 1\}$ following Dijkstra and the notation in Python. (If $n_1 \geq n_2$, then $[n_1 : n_2] = \emptyset$.) For two finite sets $\mathcal{X}$ and $\mathcal{Y}$, $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$ denotes a set of all functions whose domain is $\mathcal{X}$ and codomain is $\mathcal{Y}$.

For a distribution D, we often write "$x \leftarrow D$," which indicates that we take a sample x according to D. For a finite set $\mathcal{S}$, $U(\mathcal{S})$ denotes the uniform distribution over $\mathcal{S}$. We often write "$x \leftarrow \mathcal{S}$" instead of "$x \leftarrow U(\mathcal{S})$."

Game $\mathsf{Expt}^{\text{euf-nma}}_{\mathsf{DS},\mathcal{A}}(1^\lambda)$ / $\mathsf{Expt}^{\text{euf-cma}}_{\mathsf{DS},\mathcal{A}}(1^\lambda)$	SIGN(msg)
1 : $\mathcal{Q} := \emptyset$	1 : $\sigma \leftarrow \mathsf{Sign}(\mathtt{sk}, \mathtt{msg})$
2 : $(\mathtt{vk}, \mathtt{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$	2 : $\mathcal{Q} := \mathcal{Q} \cup \{\mathtt{msg}\}$
3 : $(\mathtt{msg}^*, \sigma^*) \leftarrow \mathcal{A}(\mathtt{vk})$ // euf-nma	3 : **return** σ
4 : $(\mathtt{msg}^*, \sigma^*) \leftarrow \mathcal{A}^{\text{SIGN}}(\mathtt{vk})$ // euf-cma	
5 : **if** $\mathtt{msg}^* \in \mathcal{Q}$ **then return** 0 // euf-cma	
6 : **return** $\mathsf{Vrfy}(\mathtt{vk}, \mathtt{msg}^*, \sigma^*)$	

Fig. 2. Games for EUF-NMA and EUF-CMA security (Definition 2.2).

If inp is a string, then "$\mathsf{out} \leftarrow \mathsf{A}^O(\mathsf{inp})$" denotes the output of algorithm A running on input inp with access to a set of oracles O. If A and oracles are deterministic, then out is a fixed value and we write "$\mathsf{out} := \mathsf{A}^O(\mathsf{inp})$." We also use the notation "$\mathsf{out} := \mathsf{A}(\mathsf{inp}; r)$" to make the randomness r of A explicit.

For a function $f : \{0,1\}^n \to \{0,1\}^m$, *quantum access* to f is modeled as an oracle access to unitary $O_f : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$. By convention, we will use the notation $\mathsf{A}^{|f\rangle,g}$ to stress A's *quantum* and classical access to f and g, respectively. For a function $f : \mathcal{X} \to \mathcal{Y}$, we denote the procedure reprogramming $f(x)$ with h by $f := f[x \mapsto h]$.

For a non-negative integer $i \in [2^x]$, $\mathsf{bin}_x(i)$ denotes the x-bit representation of the integer i.

2.1 Digital Signature

The model for digital signature schemes is summarized as follows:

Definition 2.1. *A digital signature scheme* DS *consists of the following triple of PPT algorithms* (Gen, Sign, Vrfy)*:*

- $\mathsf{Gen}(1^\lambda) \to (\mathtt{vk}, \mathtt{sk})$*: a key-generation algorithm that, on input* 1^λ*, outputs a pair of keys* $(\mathtt{vk}, \mathtt{sk})$*.* vk *and* sk *are verification and signing keys, respectively.*
- $\mathsf{Sign}(\mathtt{sk}, \mathtt{msg}) \to \sigma$*: a signing algorithm that takes as input signing key* sk *and message* $\mathtt{msg} \in \mathcal{M}$ *and outputs signature* $\sigma \in \mathcal{S}$*.*
- $\mathsf{Vrfy}(\mathtt{vk}, \mathtt{msg}, \sigma) \to 0/1$*: a verification algorithm that takes as input verification key* vk*, message* $\mathtt{msg} \in \mathcal{M}$*, and signature* σ *and outputs its decision* 1 *for acceptance or* 0 *for rejection.*

We require perfect correctness; that is, for any message $\mathtt{msg} \in \mathcal{M}$*, we have*

$$\Pr[(\mathtt{vk}, \mathtt{sk}) \leftarrow \mathsf{Gen}(1^\lambda), \sigma \leftarrow \mathsf{Sign}(\mathtt{sk}, \mathtt{msg}) : \mathsf{Vrfy}(\mathtt{vk}, \mathtt{msg}, \sigma) = 1] = 1.$$

Security notions: We review the standard security notions of existential unforgeability under chosen-message attacks (EUF-CMA) and under no-message attacks (EUF-NMA).

Definition 2.2 (EUF-CMA and EUF-NMA security). *Let* $\mathsf{DS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *be a digital signature scheme. For any* $\mathcal{A}$*, we define its* euf-cma/euf-nma *advantages against* DS *as* $\mathsf{Adv}^{\text{euf-cma}}_{\mathsf{DS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Expt}^{\text{euf-cma}}_{\mathsf{DS},\mathcal{A}}(1^\lambda) = 1]$ *and* $\mathsf{Adv}^{\text{euf-nma}}_{\mathsf{DS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Expt}^{\text{euf-nma}}_{\mathsf{DS},\mathcal{A}}(1^\lambda) = 1]$*, where* $\mathsf{Expt}^{\text{euf-cma}}_{\mathsf{DS},\mathcal{A}}(1^\lambda)$ *and* $\mathsf{Expt}^{\text{euf-nma}}_{\mathsf{DS},\mathcal{A}}(1^\lambda)$ *are the experiments described in Figure 2. We say that* DS *is EUF-CMA-secure (resp., EUF-NMA-secure) if* $\mathsf{Adv}^{\text{euf-cma}}_{\mathsf{DS},\mathcal{A}}(\lambda)$ *(resp.,* $\mathsf{Adv}^{\text{euf-nma}}_{\mathsf{DS},\mathcal{A}}(\lambda)$*) is negligible in* λ *for any QPT adversary* $\mathcal{A}$*.*

2.2 3/5-Round Identification

We consider 3-round and 5-round identification (ID) schemes. We only treat *public-coin* ID schemes; that is, the verifier chooses i-th challenge uniformly at random from the challenge set $\mathcal{C}_i$. The syntax follows:

Definition 2.3 ($(2n+1)$-round identification). *A* $(2n+1)$*-round ID scheme* ID *consists of the following tuple of PPT algorithms* $(\mathsf{Gen}, \mathsf{P}_1, \mathcal{C}_1, \ldots, \mathsf{P}_n, \mathcal{C}_n, \mathsf{P}_{n+1}, \mathsf{V})$*:*

- $\mathsf{Gen}(1^\lambda) \to (\mathtt{vk}, \mathtt{sk})$*: a key-generation algorithm that takes* 1^λ *as input, where* λ *is the security parameter, and outputs a pair of keys* $(\mathtt{vk}, \mathtt{sk})$*.* vk *and* sk *are public verification and signing keys, respectively.*
- $\mathsf{P}_1(\mathtt{sk}; \rho) \to (a_1, \mathtt{state}_1)$*: a first prover algorithm that takes signing key* sk *and randomness* ρ *as input, and outputs the first message* a_1 *and state* $\mathtt{state}_1$*.*

- $\mathsf{P}_i(c_{i-1}, \mathtt{state}_{i-1}) \to (a_i, \mathtt{state}_i)$*: an i-th prover algorithm for $i = 2, \ldots, n$ that takes the $(i-1)$-th challenge $c_{i-1} \in \mathcal{C}_{i-1}$ and state $\mathtt{state}_{i-1}$ as input, and outputs the i-th message a_i and state $\mathtt{state}_i$.*
- $\mathsf{P}_{n+1}(c_n, \mathtt{state}_n) \to z/\perp$*: the last prover algorithm that takes the n-th challenge $c_n \in \mathcal{C}_n$ and state $\mathtt{state}_n$ as input, and outputs the last message z or $\perp$.*
- $\mathsf{V}(\mathtt{vk}, a_1, c_1, \ldots, a_n, c_n, z) \to 0/1$*: a verification algorithm that takes verification key $\mathtt{vk}$ and the transcript $a_1, c_1, \ldots, a_n, c_n, z$ as input and outputs its decision, 1 (true) or 0 (false).*

We assume perfect correctness; a verifier always outputs 1 for an arbitrary honestly-generated key and transcript. If $n = 1$, then we will drop some subscripts and write a scheme as $(\mathsf{Gen}, \mathsf{P}_1, C, \mathsf{P}_2, \mathsf{V})$ and a transcript as (a, c, z).

2.3 Other Security Notions

We consider the following versions of PRF, PRG, and collision-resistant security.

Definition 2.4 (PRF). *Let $\mathsf{PRF} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a function. We define its advantage as*

$$\mathsf{Adv}^{\mathrm{prf}}_{\mathsf{PRF},\mathcal{A}}(\lambda) := \left| \Pr_{k \leftarrow \mathcal{K}} \left[1 \leftarrow \mathcal{A}^{\mathsf{F}_0}()\right] - \Pr_{\mathsf{F}_1 \leftarrow \mathsf{Func}(\mathcal{X},\mathcal{Y})} \left[1 \leftarrow \mathcal{A}^{\mathsf{F}_1}()\right] \right|,$$

where $\mathsf{F}_0(x)$ returns $\mathsf{PRF}(k, x)$. We say that PRF is pseudorandom if $\mathsf{Adv}^{\mathrm{prf}}_{\mathsf{PRF},\mathcal{A}}(\lambda)$ is negligible for any PPT/QPT adversary $\mathcal{A}$.

Definition 2.5 (PRG). *Let $\mathsf{PRG} : \mathcal{K} \to \mathcal{R}$ be a function. We define its advantage as*

$$\mathsf{Adv}^{\mathrm{prg}}_{\mathsf{PRG},\mathcal{A}}(\lambda) := \left| \Pr_{k \leftarrow \mathcal{K}} [1 \leftarrow \mathcal{A}(\mathsf{PRG}(k))] - \Pr_{r \leftarrow \mathcal{R}} [1 \leftarrow \mathcal{A}(r)] \right|.$$

We say that PRG *is* pseudorandom *if $\mathsf{Adv}^{\mathrm{prg}}_{\mathsf{PRG},\mathcal{A}}(\lambda)$ is negligible for any PPT/QPT adversary $\mathcal{A}$.*

Definition 2.6 (Collision-Resistant Hash). *Let $\mathsf{Hash} : \mathcal{X} \to \mathcal{Y}$ be a function. We define its collision-resistant advantage as*

$$\mathsf{Adv}^{\mathrm{cr}}_{\mathsf{Hash},\mathcal{A}}(\lambda) := \Pr\left[(x, x') \leftarrow \mathcal{A}(\mathsf{Hash}) \,:\, x \neq x' \wedge \mathsf{Hash}(x) = \mathsf{Hash}(x')\right].$$

We say that Hash *is* collision-resistant *if $\mathsf{Adv}^{\mathrm{cr}}_{\mathsf{Hash},\mathcal{A}}(\lambda)$ is negligible for any PPT/QPT adversary $\mathcal{A}$.*

2.4 Adaptive Reprogramming of Quantum Random Oracle

It is hard to distinguish whether the random oracle is reprogrammed or not if the min-entropy of the reprogrammed point is sufficiently high.

Lemma 2.1 ([GHHM21, Theorem 1], adapted). *Let $\mathcal{X}_1$, $\mathcal{X}_2$,and $\mathcal{Y}$ be finite sets, and $\mathrm{O}_0 = \mathrm{O}_1 \leftarrow \mathsf{Func}(\mathcal{X}_1 \times \mathcal{X}_2, \mathcal{Y})$. We define an oracle* Repro *as follows: on input a distribution D and $x_2 \in \mathcal{X}_2$, it samples $(x_1, \mathsf{side}) \leftarrow D$ and $y \leftarrow \mathcal{Y}$, reprograms only O_1 as $\mathrm{O}_1[(x_1, x_2) \mapsto y]$, and returns (x_1, side). We let D_i be a distribution that the adversary $\mathcal{A}$ queries to* Repro *in the i-th query. Suppose that $\mathcal{A}$ makes R queries to* Repro *and q quantum queries to $|\mathrm{O}_b\rangle$. Then, the distinguishing advantage of $\mathcal{A}$ is bounded by*

$$\left| \Pr\left[1 \leftarrow \mathcal{A}^{|\mathrm{O}_0\rangle, \mathrm{Repro}}()\right] - \Pr\left[1 \leftarrow \mathcal{A}^{|\mathrm{O}_1\rangle, \mathrm{Repro}}()\right] \right| \leq \frac{3}{2} \sum_{i \in [R]} \sqrt{q p^{(i)}_{\mathsf{max}}},$$

where we define $p^{(i)}_{\mathsf{max}} := \mathbb{E} \max_{\hat{x}_1} \Pr_{(x_1, \mathsf{side}) \leftarrow D_i}[x_1 = \hat{x}_1]$ with the expectation taken over $\mathcal{A}$'s behavior up to the i-th query.

2.5 (Secret-key-root) Correlated GGM-based All-but-one Vector Commitment

We formulate a correlated GGM-based all-but-one vector commitment (cGGM-based AVC) scheme as a tuple of three algorithms, $(\mathsf{Commit}, \mathsf{Open}, \mathsf{Recon})$, where we assume that $N = 2^d$ and let $i^* \in [N]$ be a hidden index. We further let n_b denote the tape length in λ-bit blocks. In this section, the functions used for tree derivation, seed commitment, and tape expansion are modeled as independent random oracles $\mathsf{H}_{\mathsf{tree}}$, $\mathsf{H}_{\mathsf{com}}$, and $\mathsf{H}_{\mathsf{tape}}$, respectively.

AVC.Commit($\mathtt{salt}, \mathtt{rseed}, \delta, e$)

1 : $(\mathtt{node}[2], \mathtt{node}[3]) := (\mathtt{rseed}, \mathtt{rseed} \oplus \delta)$
2 : **for** $j \in [1 : d]$ **do**
3 : **for** $k \in [2^j : 2^{j+1}]$ **do**
4 : $\mathtt{node}[2k] := \mathsf{H}_{\mathsf{tree}}(\mathtt{salt}, e, j, \mathtt{node}[k])$
5 : $\mathtt{node}[2k+1] := \mathtt{node}[2k] \oplus \mathtt{node}[k]$
6 : **for** $i \in [N]$ **do**
7 : $\mathtt{seed}[i] := \mathtt{node}[N+i]$
8 : $\mathtt{com}[i] := \mathsf{H}_{\mathsf{com}}(\mathtt{salt}, e, \mathtt{seed}[i])$
9 : $\mathtt{tape}[i] := \mathsf{H}_{\mathsf{tape}}(\mathtt{salt}, e, \mathtt{seed}[i])$
10 : $\mathtt{decom} := (\mathtt{node}, \mathtt{com})$
11 : **return** $\mathtt{com}, \mathtt{decom}, \mathtt{seed}, \mathtt{tape}$

AVC.Open($\mathtt{decom}, e, i^*$)

1 : Parse $\mathtt{decom} = (\mathtt{node}, \mathtt{com})$
2 : **for** $j \in [1 : d+1]$ **do**
3 : $\mathtt{path}[j] := \mathtt{node}[\mathsf{copath}_j(i^*)]$
4 : **return** $\mathtt{pdecom} := (\mathtt{path}, \mathtt{com}[i^*])$

AVC.Recon($\mathtt{salt}, \mathtt{pdecom}, e, i^*$)

1 : Parse $\mathtt{pdecom} = (\mathtt{path}, \mathtt{com}[i^*])$
2 : Initialize all nodes by $\perp$
3 : **for** $j \in [1 : d+1]$ **do**
4 : $\mathtt{node}[\mathsf{copath}_j(i^*)] := \mathtt{path}[j]$
5 : **for** $j \in [1 : d]$ **do**
6 : **for** $k \in [2^j : 2^{j+1}]$ **do**
7 : **if** $\mathtt{node}[k] \neq \perp$ **then**
8 : $\mathtt{node}[2k] := \mathsf{H}_{\mathsf{tree}}(\mathtt{salt}, e, j, \mathtt{node}[k])$
9 : $\mathtt{node}[2k+1] := \mathtt{node}[2k] \oplus \mathtt{node}[k]$
10 : **for** $i \in [N] \setminus \{i^*\}$ **do**
11 : $\mathtt{seed}[i] := \mathtt{node}[N+i]$
12 : $\mathtt{com}[i] := \mathsf{H}_{\mathsf{com}}(\mathtt{salt}, e, \mathtt{seed}[i])$
13 : $\mathtt{tape}[i] := \mathsf{H}_{\mathsf{tape}}(\mathtt{salt}, e, \mathtt{seed}[i])$
14 : **return** $\mathtt{com}, (\mathtt{seed}[i])_{i \neq i^*}, (\mathtt{tape}[i])_{i \neq i^*}$

Fig. 3. Algorithms for the abstract (s)cGGM-based AVC, where $N = 2^d$.

Syntax: We first specify the input-output behavior of the cGGM-based AVC algorithms. The algorithms below describe the AVC procedure for a single repetition. The repetition index e is given explicitly and is included in the inputs.

- $\mathsf{Commit}(\mathtt{salt}, \mathtt{rseed}, \delta, e) \to (\mathtt{com}, \mathtt{decom}, \mathtt{seed}, \mathtt{tape})$: The commitment algorithm takes a salt $\mathtt{salt}$, a root seed $\mathtt{rseed} \in \{0,1\}^\lambda$, a correlation value $\delta \in \{0,1\}^\lambda$, a repetition index e as input. It outputs a commitment $\mathtt{com}$, decommitment information $\mathtt{decom}$, seeds $\mathtt{seed} \in (\{0,1\}^\lambda)^N$, and tapes $\mathtt{tape} \in (\{0,1\}^{n_b \lambda})^N$.
- $\mathsf{Open}(\mathtt{decom}, e, i^*) \to \mathtt{pdecom}$: The opening algorithm takes decommitment information $\mathtt{decom}$, the repetition index e, and a hidden leaf index $i^* \in [N]$ as input. It outputs partial decommitment information $\mathtt{pdecom}$.
- $\mathsf{Recon}(\mathtt{salt}, \mathtt{pdecom}, e, i^*) \to (\mathtt{com}, (\mathtt{seed}[i])_{i \neq i^*}, (\mathtt{tape}[i])_{i \neq i^*})$: The reconstruction algorithm takes $\mathtt{salt}$, partial decommitment information $\mathtt{pdecom}$, the repetition index e, and a hidden leaf index $i^* \in [N]$ as input. It outputs the reconstructed commitment $\mathtt{com}$, the reconstructed opened seeds $(\mathtt{seed}[i])_{i \neq i^*}$, and the reconstructed opened tapes $(\mathtt{tape}[i])_{i \neq i^*}$.

We use heap-style indexing for the correlated tree. The two children of the correlated root are $\mathtt{node}[2]$ and $\mathtt{node}[3]$, and the leaves are $\mathtt{node}[N], \ldots, \mathtt{node}[2N-1]$. Thus, the leaf corresponding to index $i^* \in [N]$ is $\mathtt{node}[N+i^*]$. For each layer $j \in \{1, \ldots, d\}$, define

- $\mathsf{path}_j(i^*) := \lfloor (N+i^*)/2^{d-j} \rfloor$, which is the index of the hidden-path node at layer j.
- $\mathsf{copath}_j(i^*) := \lfloor (N+i^*)/2^{d-j} \rfloor \oplus 1$, which is the index of the sibling of the hidden-path node at layer j.

Construction: We describe the abstract cGGM-based AVC scheme in Figure 3. The construction uses three functions, $\mathsf{H}_{\mathsf{tree}}$, $\mathsf{H}_{\mathsf{com}}$, and $\mathsf{H}_{\mathsf{tape}}$, for tree derivation, seed commitment, and tape expansion, respectively. In the ROM/QROM, they are modeled as independent random oracles.

The tree is initialized by setting $(\mathtt{node}[2], \mathtt{node}[3]) := (\mathtt{rseed}, \mathtt{rseed} \oplus \delta)$. Then, for each internal node $\mathtt{node}[k]$ in layer j, the two children are derived as $(\mathtt{node}[2k], \mathtt{node}[2k+1]) := (\mathsf{H}_{\mathsf{tree}}(\mathtt{salt}, e, j, \mathtt{node}[k]), \mathsf{H}_{\mathsf{tree}}(\mathtt{salt}, e, j, \mathtt{node}[k]) \oplus \mathtt{node}[k])$. Hence, the xor of all nodes in every layer is equal to δ as in eq.(1), and in particular $\bigoplus_{i \in [N]} \mathtt{seed}[i] = \delta$.

When the correlation value, δ, is set to the first λ bits of a witness and each leaf tape begins with the corresponding leaf seed, we call the resulting construction a secret-key-root cGGM-based AVC, or scGGM-based AVC. In an scGGM-based AVC, the invariant $\bigoplus_{i \in [N]} \mathtt{seed}[i] = \delta$ implies that the first λ bits of the witness-dependent correction value are always 0^λ. Hence, these bits can be omitted from the opening information.

2.6 MQOM

Specification: We recall the specification of MQOM [BBFR25] in the notation used throughout this paper. For parameters, please see Table 1 in Appendix A. Let $\mathbb{F} = \mathrm{GF}(2^k)$ be a finite field and let $x \in \mathbb{F}^n$ be the secret witness. The public key is generated from a seed $\mathtt{mseed}_{\mathtt{eq}}$ that determines a system of quadratic equations and from the corresponding syndrome $\hat{y}$. Let $\mathbb{K}$ be an extension field of $\mathbb{F}$ with $[\mathbb{K} : \mathbb{F}] = \mu$. Let $\hat{m} := m/\mu$. More precisely, $(\{\hat{A}_i\}, \{\hat{b}_i\})_{i\in[\hat{m}]} := \mathsf{ExpandEquations}(\mathtt{mseed}_{\mathtt{eq}})$, where $\hat{A}_i \in \mathbb{K}^{n\times n}$ and $\hat{b}_i \in \mathbb{K}^n$, and the public value is $\hat{y} = (\hat{y}_0, \dots, \hat{y}_{\hat{m}-1}) \in \mathbb{K}^{\hat{m}}$. We define $\hat{A}(s) := (s^\top \cdot \hat{A}_0 \cdot s, \dots, s^\top \cdot \hat{A}_{\hat{m}-1} \cdot s) \in \mathbb{K}^{\hat{m}}$ for $s \in \mathbb{K}^n$, $\hat{B} := [\hat{b}_0^\top; \dots; \hat{b}_{\hat{m}-1}^\top] \in \mathbb{K}^{\hat{m}\times n}$, whose i-th row is $\hat{b}_i^\top$. These values satisfy the relation $\hat{F}_{\mathtt{mseed}_{\mathtt{eq}},\hat{y}}(x) = 0^{\hat{m}}$ for $x \in \mathbb{F}^n$ over $\mathbb{K}^{\hat{m}}$, where

$$\hat{F}_{\mathtt{mseed}_{\mathtt{eq}},\hat{y}}(x) := \hat{A}(x) + \hat{B} \cdot x - \hat{y}.$$

The signer proves knowledge of $x \in \mathbb{F}^n$ such that $\hat{F}_{\mathtt{mseed}_{\mathtt{eq}},\hat{y}}(x) = 0^{\hat{m}}$. The verification key and signing key are $\mathtt{vk} := (\mathtt{mseed}_{\mathtt{eq}}, \hat{y})$ and $\mathtt{sk} := (\mathtt{vk}, x)$, respectively.

To show the relation, MQOM uses a 3-round TCitH protocol [FR23, FR25], where $\eta = \hat{m}$:

- The prover in the first round commits $P_x(X) = x_0 + xX \in (\mathbb{K}[X])^n$ and $P_u(X) = u_0 + u_1 X \in (\mathbb{K}[X])^\eta$ into $\mathtt{com}_1$, where $\eta = \hat{m}$ and commitments are implemented by the line commitment using (sc)GGM tree [BBFR25]. It also commits polynomial $P_\alpha(X) = \alpha_0 + \alpha_1 X \in (\mathbb{K}[X]^\eta)$ into $\mathtt{com}_2$ by using the hash function, where $P_z(X) := X^2 \cdot \hat{F}_{\mathtt{mseed}_{\mathtt{eq}},\hat{y}}(P_x(X)/X)$ and $P_\alpha(X) := P_u(X) + P_z(X)$.
- The verifier sends a random challenge $\omega_{i^*} \leftarrow \Omega = \{\omega_0, \dots, \omega_{N-1}\} \subseteq \mathbb{K}$.
- The prover in the third round responds with the opening information of commitments and values $x_{\mathtt{eval}} := P_x(\omega_{i^*})$ and $u_{\mathtt{eval}} := P_u(\omega_{i^*})$.
- The verifier checks the commitments and the relation by $P_\alpha(\omega_{i^*}) = P_u(\omega_{i^*}) + P_z(\omega_{i^*})$, where the left hand side can be computed from α_0, α_1, and ω_{i^*} and the right hand side can be computed from $\hat{F}$, $u_{\mathtt{eval}}$, $x_{\mathtt{eval}}$, and ω_{i^*}.

The line commitments of P_x and P_u are implemented as follows: Fix $\Omega = \{\omega_0, \dots, \omega_{N-1}\} \subseteq \mathbb{K}$. Let $P_i(X) = X - \omega_i \in \mathbb{K}[X]$ for $i \in [N]$. Suppose that we have N random pairs $(\bar{x}_i, \bar{u}_i)_{i\in[N]} \in (\mathbb{F}^n \times \mathbb{K}^\eta)^N$ via the (sc)GGM tree. Let $\Delta_x := x - \sum_{i\in[N]} \bar{x}_i \in \mathbb{F}^n$. We then define

$$P_x(X) := \Delta_x P_0(X) + \sum_{i\in[N]} \bar{x}_i P_i(X) = \underbrace{\left(-\omega_0 \Delta_x - \sum_{i\in[N]} \omega_i \bar{x}_i\right)}_{=: x_0} + \underbrace{\left(\Delta_x + \sum_{i\in[N]} \bar{x}_i\right)}_{x} X \in \mathbb{K}[X]^n,$$

$$P_u(X) := \sum_{i\in[N]} \bar{u}_i P_i(X) = \underbrace{-\sum_{i\in[N]} \omega_i \bar{u}_i}_{=: u_0} + \underbrace{\sum_{i\in[N]} \bar{u}_i}_{=: u_1} X \in \mathbb{K}[X]^\eta.$$

This definition allows the verifier to compute $x_{\mathtt{eval}}$ and $u_{\mathtt{eval}}$ from $(\bar{x}_i, \bar{u}_i)_{i\neq i^*}$ as $P_x(\omega_{i^*}) = \Delta_x P_0(\omega_{i^*}) + \sum_{i\in[N]} \bar{x}_i P_i(\omega_{i^*}) = \Delta_x P_0(\omega_{i^*}) + \sum_{i\neq i^*} \bar{x}_i P_i(\omega_{i^*})$ and $P_u(\omega_{i^*}) = \sum_{i\in[N]} \bar{u}_i P_i(\omega_{i^*}) = \sum_{i\neq i^*} \bar{u}_i P_i(\omega_{i^*})$.

Applying the Fiat–Shamir transform (with grinding), we obtain the signature scheme described in Figure 4, Figure 5, and Figure 11. Here, each Hash_i outputs 2λ bits, i.e., has range $\{0,1\}^{2\lambda}$.

MQOM also specifies the 5-round version: The authors take $\eta < \hat{m}$, the verifier chooses a random challenge $\Gamma \leftarrow \mathbb{K}^{\eta\times\hat{m}}$, and the prover responds with $P_\alpha(X) := P_u(X) + \Gamma \cdot P_z(X) \in (\mathbb{K}[X])^\eta$.

In this paper, we first modify MQOM so that Hash_4 for the Fiat–Shamir transform incorporates `salt` as Line 29 of Figure 5 and Line 5 of Figure 11.

This modification avoids the heuristic argument about the min-entropy of the input to Hash_4. We note that inclusion of the salt in the input of Hash_4 is often employed in the MPCitH signature schemes [AAB+24, ABB+24a, ABB+24c].

We also modify the evaluation set $\Omega \subseteq \mathbb{K}$ to exclude 0. We denote the variant with both modifications by $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ and the variant with only the evaluation-domain modification by MQOM_{Ω}. See Section 4 for the details of the parameter settings.

The scGGM-based AVC instantiation in MQOM*:* The signing and verification algorithms use the syntax of Section 2.5 with the concrete salted correlated GGM tree described in Figure 6. For each tree index $e \in [\tau]$, the signer samples $\mathtt{rseed}[e] \in \{0,1\}^\lambda$ and sets the correlation value to $\delta := \mathsf{FirstBits}_\lambda(x)$. The construction uses tweaked salts $t_{\mathtt{salt},i,e,j}$

MQOM.Gen()

$\mathtt{seed}_{\mathtt{key}} \leftarrow \{0,1\}^{2\lambda}; (x, \mathtt{mseed}_{\mathtt{eq}}) := \mathsf{XOF}_0(\mathtt{seed}_{\mathtt{key}})$ // $x \in \mathbb{F}^n$

$(\{\hat{A}_i\}, \{\hat{b}_i\}) := \mathsf{ExpandEquations}(\mathtt{mseed}_{\mathtt{eq}})$ // $\hat{A}_i \in \mathbb{K}^{n \times n}$, $\hat{b}_i \in \mathbb{K}^n$

for $i \in [\hat{m}]$ **do**

  $\hat{y}_i := x^\top \hat{A}_i x + \hat{b}_i^\top x$

$\hat{y} = (\hat{y}_0, \ldots, \hat{y}_{\hat{m}-1})$ // $\hat{y} \in \mathbb{K}^{\hat{m}}$

$\mathtt{vk} := (\mathtt{mseed}_{\mathtt{eq}}, \hat{y})$; $\mathtt{sk} := (\mathtt{vk}, x)$

return $(\mathtt{vk}, \mathtt{sk})$

Fig. 4. Key-generation algorithm of MQOM (ver.2.1).

for domain separation and a Davies–Meyer-like tweakable hash function $\mathsf{H}_{\mathsf{avc}}(s,t) = \mathsf{EncFF}(s,t) := E(t,s) \oplus \psi(s)$[8] to derive tree nodes, seed commitments, and tapes.

Associated one-wayness assumptions: For a key pair generated by MQOM.Gen, write $\mathtt{vk} = (\mathtt{mseed}_{\mathtt{eq}}, \hat{y})$ and let x denote the secret witness contained in the signing key. The standard one-wayness (OW) assumption for MQOM key generation requires that, given vk, it is hard to find any x' satisfying $\hat{F}_{\mathtt{mseed}_{\mathtt{eq}},\hat{y}}(x') = 0^{\hat{m}}$. The partial-guessing one-wayness (PGOW) assumption instead requires that it is hard to recover $\delta = \mathsf{FirstBits}_\lambda(x)$, even with access to an oracle that tests whether a candidate equals δ [FR26]. We denote the resulting MQOM function family by $\mathsf{F}_{\mathsf{MQOM}}$. Formal definitions of OW and PGOW are given in Appendix C.

3 Security of scGGM-based AVC of modified MQOM in the (Q)ROM

This section defines the simulation experiment for the scGGM-based AVC openings that appear in the signing oracle of MQOM_Ω and proves its security in the ROM and QROM. For each signing query, MQOM_Ω generates τ scGGM trees with the common secret-key root δ and opens, in each repetition $e \in [\tau]$, all leaves except the challenge-dependent hidden leaf $\mathbf{i}^*[e]$. The data to be simulated are not only the partial decommitment $\mathtt{pdecom}[e]$ and the hidden commitment $\mathtt{com}[e][\mathbf{i}^*[e]]$, but also the MQOM-specific exposed values that are masked by the hidden tape, such as $\Delta_x^{(1)}[e]$ and $\alpha_0[e]$. This definition fixes the transcript interface of the simulation problem; the next section proves the corresponding statement for the original ideal-cipher-based scGGM instantiation of MQOM_Ω.

We define the simulation experiment for the scGGM-based AVC openings that arise in the signing procedure of MQOM_Ω. After key generation, the first-stage adversary $\mathcal{A}_0$ chooses the signing-query contexts, consisting of salts and hidden-index vectors. For each $q \in [Q]$, the experiment generates the τ opening-related transcripts either honestly from the scGGM trees or by the simulator. The second-stage adversary $\mathcal{A}_1$ then receives these transcripts and tries to distinguish the real experiment from the simulated experiment. Thus, the experiment captures exactly the part of the MQOM_Ω signing oracle in the intermediate EUF-CMA game, in which the openings and the associated exposed values are generated after the hidden indices have been fixed.

We define the auxiliary map in the same style as Feneuil and Rivain [FR26].

For each repetition e, the first auxiliary map $f_{\mathrm{aux},0}$ computes the MQOM_Ω-specific value $\Delta_x^{(1)}[e]$ from the secret key x and the tree-derived seeds and tapes. After the full vector $\Delta_x^{(1)}$ has been computed, the public value Γ is determined as $\Gamma := I_\eta$ or $\mathsf{XOF}_8(\mathsf{Hash}_7(h_{\mathtt{com}}, \Delta_x^{(1)}))$. The second auxiliary map $f_{\mathrm{aux},1}$ then computes $\alpha_0[e]$ from the corresponding tree-derived seeds and tapes together with the resulting public value Γ.

Fix a signing-query context, a repetition e, and a hidden index $i^* = \mathbf{i}^*[e]$. We further fix all public values and all seeds and tapes reconstructed from $\mathtt{pdecom}[e]$, and regard only the hidden tape value $\mathtt{tape}[e][i^*]$ as variable. We write

$$f_{\mathrm{aux}} := \left(f_{\mathrm{aux},0}, f_{\mathrm{aux},1}\right)$$

for the resulting combined auxiliary map.

[8] $E : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ is a blockcipher and $\psi : (s_l, s_r) \in \{0,1\}^{\lambda/2} \times \{0,1\}^{\lambda/2} \mapsto (s_l \oplus s_r, s_l)$ is a linear orthomorphism [GKWY20].

MQOM.Sign($\mathtt{sk}, \mathtt{msg}$) / $\mathsf{MQOM}_{\mathtt{salt},\Omega}$.Sign($\mathtt{sk}, \mathtt{msg}$)

$(\mathtt{vk}, x) \coloneqq \mathsf{Parse}(\mathtt{sk}); (\mathtt{mseed}_{\mathtt{eq}}, \hat{y}) \coloneqq \mathsf{Parse}(\mathtt{vk}); (\{\hat{A}_i\}, \{\hat{b}_i\}) \coloneqq \mathsf{ExpandEquations}(\mathtt{mseed}_{\mathtt{eq}})$
$\mathtt{mseed} \leftarrow \{0,1\}^\lambda; \mathtt{salt} \leftarrow \{0,1\}^\lambda$
$\mathtt{msg_hash} \coloneqq \mathsf{Hash}_2(\mathtt{msg})$
$(\mathtt{rseed}[0], \ldots, \mathtt{rseed}[\tau-1]) \coloneqq \mathsf{PRG}(\mathtt{mseed})$
$\delta \coloneqq \mathsf{FirstBits}_\lambda(x)$
for $e \in [\tau]$ **do** // Commit P_x and P_u
  $(\mathtt{com}[e], \mathtt{decom}[e], \mathtt{seed}[e], \mathtt{tape}[e]) \coloneqq \mathsf{MQOM.Commit}(\mathtt{salt}, \mathtt{rseed}[e], \delta, e)$
  $h_{\mathtt{com}}[e] \coloneqq \mathsf{Hash}_6(\mathtt{com}[e])$
  for $i \in [N]$ **do**
    $(\bar{x}[e][i], \bar{u}[e][i]) \coloneqq \mathsf{Parse}(\mathtt{seed}[e][i], \mathtt{tape}[e][i])$ // $\bar{x}[e][i] \in \mathbb{F}^n$ and $\bar{u}[e][i] \in \mathbb{K}^\eta$
  $\Delta_x[e] \coloneqq x - \sum_{i\in[N]} \bar{x}[e][i]$ // The first λ bits of $\Delta_x[e]$ are 0^λ
  $\Delta_x^{(1)}[e] \coloneqq (\Delta_x[e])[\lambda : kn]$
  $x_0[e] \coloneqq -\omega_0 \Delta_x[e] - \sum_{i\in[N]} \omega_i \cdot \bar{x}[e][i] \in \mathbb{K}^n$ // $\omega_0 = 0$ in MQOM
  $u_0[e] \coloneqq -\sum_{i\in[N]} \omega_i \cdot \bar{u}[e][i]$
  $u_1[e] \coloneqq \sum_{i\in[N]} \bar{u}[e][i]$
$\mathtt{com}_1 \coloneqq \mathsf{Hash}_7(h_{\mathtt{com}}, \Delta_x^{(1)})$
$\Gamma \coloneqq \boldsymbol{I}_\eta \in \mathbb{K}^{\eta\times\eta}$ // 3-round version, where $\hat{m} = \eta$
$\Gamma \coloneqq \mathsf{XOF}_8(\mathtt{com}_1) \in \mathbb{K}^{\eta\times\hat{m}}$ // 5-round version, where $\hat{m} > \eta$
for $e \in [\tau]$ **do** // Compute P_z and P_α
  for $i \in [\hat{m}]$ **do**
    $z_{0,i} \coloneqq x_0[e]^\top \cdot \hat{A}_i^\top \cdot x_0[e];$
    $z_{1,i} \coloneqq x_0[e]^\top \cdot \hat{A}_i^\top \cdot x + (x^\top \cdot \hat{A}_i^\top + \hat{b}_i^\top) \cdot x_0[e]$
  $z_0 \coloneqq (z_{0,0}, \ldots, z_{0,\hat{m}-1})$
  $z_1 \coloneqq (z_{1,0}, \ldots, z_{1,\hat{m}-1})$
  $\alpha_0[e] \coloneqq u_0[e] + \Gamma \cdot z_0$
  $\alpha_1[e] \coloneqq u_1[e] + \Gamma \cdot z_1$
$\mathtt{com}_2 \coloneqq \mathsf{Hash}_3(\alpha_0, \alpha_1)$
$h_2 \coloneqq \mathsf{Hash}_4(\mathtt{vk}, \mathtt{com}_1, \mathtt{com}_2, \mathtt{msg_hash})$ // MQOM
$h_2 \coloneqq \mathsf{Hash}_4(\mathtt{vk}, \mathtt{salt}, \mathtt{com}_1, \mathtt{com}_2, \mathtt{msg_hash})$ // $\mathsf{MQOM}_{\mathtt{salt},\Omega}$
$\mathtt{nonce} \coloneqq 0$
$(\mathbf{i}^*, \mathtt{val}) \coloneqq \mathsf{XOF}_5(h_2, \mathtt{nonce})$
while $\mathtt{val} \neq 0^w$ **do**
  $\mathtt{nonce} \coloneqq \mathtt{nonce} + 1$
  $(\mathbf{i}^*, \mathtt{val}) \coloneqq \mathsf{XOF}_5(h_2, \mathtt{nonce})$
for $e \in [\tau]$ **do**
  $\mathtt{pdecom}[e] \coloneqq \mathsf{MQOM.Open}(\mathtt{decom}[e], e, \mathbf{i}^*[e])$
return $\sigma \coloneqq \mathsf{Serialize}(\mathtt{salt}, \mathtt{com}_1, \mathtt{com}_2, \alpha_1, \mathtt{pdecom}, \Delta_x^{(1)}, \mathtt{nonce})$

Fig. 5. Expanded signing procedures for MQOM (ver. 2.1) and $\mathsf{MQOM}_{\mathtt{salt},\Omega}$. The latter replaces the original evaluation domain $\Omega = \{\omega_0, \ldots, \omega_{N-1}\} \subseteq \mathbb{K}$ with one containing only nonzero evaluation points and includes $\mathtt{salt}$ in the input to Hash_4.

If $\mathsf{H}_{\mathsf{avc}}$ inside MQOM_Ω is modeled as a random oracle, our proof uses only the following local property: Viewed as a function of the hidden tape value,

$$f_{\mathrm{aux}} : \mathtt{tape}[e][i^*][0 : kn - \lambda + k\mu\eta] \longmapsto \left(\Delta_x^{(1)}[e], \alpha_0[e]\right),$$

the map f_{aux} is a bijection onto $\mathbb{F}^{n-\lambda/k} \times \mathbb{K}^\eta$ since ω_{i^*} is always non-zero.

Remark 3.1. If we consider another auxiliary map

$$f'_{\mathrm{aux}} : \mathtt{tape}[e][i^*][0 : kn - \lambda + k\mu\eta] \longmapsto \left(\Delta_x^{(1)}[e], \alpha_1[e]\right),$$

then the map is a bijection even if ω_{i^*} is zero as in [FR26].

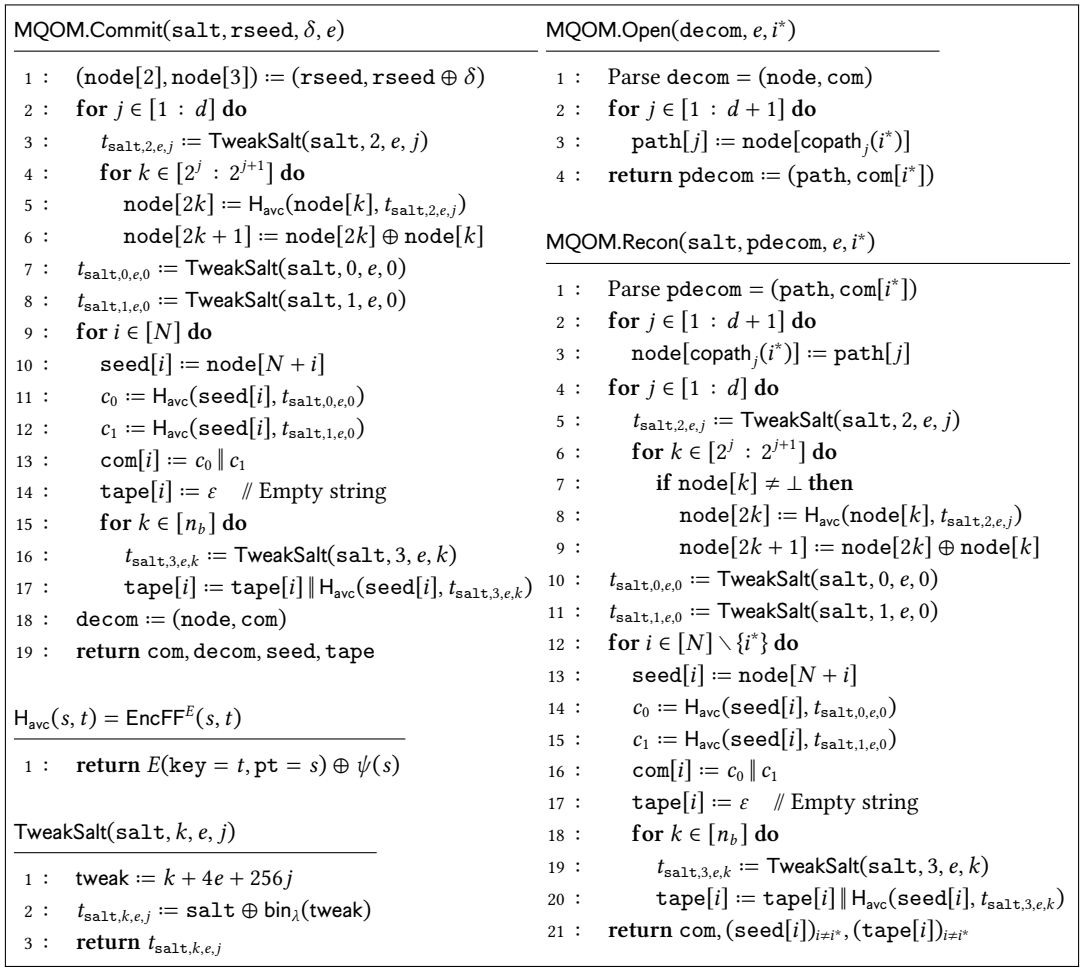


Fig. 6. The scGGM-based AVC instantiation used in MQOM.

Remark 3.2. Our definition is based on the multi-instance hiding formulation of Kosuge and Xagawa [KX26a] and the simulation security by Feneuil and Rivain [FR26]. The local bijectivity requirement for the (Q)ROM case is inspired by Feneuil and Rivain [FR26]. We also make explicit how the public auxiliary value Γ is generated for 5-round version. In our abstraction, Γ is not a local value associated with a single repetition, but a batching value from $\mathtt{com}_1 = \mathsf{Hash}_7(h_{\mathtt{com}}, \Delta_x^{(1)})$. The second-stage value $\alpha_0[e]$ of each repetition is then computed using this same value Γ. This formulation reflects the random-linear-combination or batching mechanism used in MQOM and related schemes, and makes the definition applicable to a broader class of MPC-in-the-head signatures.

Definition 3.1 (AVC simulation advantage of MQOM_Ω). *For an adversary $\mathcal{A}$, we define its AVC simulation advantage against MQOM_Ω as*

$$\mathsf{Adv}^{\mathrm{avc}}_{\mathsf{MQOM}_\Omega,\mathcal{A}}(\lambda) := \left|\Pr[\mathsf{Expt}^{\mathrm{avc},0}_{\mathsf{MQOM}_\Omega,\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Expt}^{\mathrm{avc},1}_{\mathsf{MQOM}_\Omega,\mathcal{A}}(\lambda) = 1]\right|,$$

where the experiments are described in Figure 7.

Remark 3.3. We exclude tweak collisions in Line 5 of the experiment in Figure 7, whereas Feneuil and Rivain [FR26] allow up to μ occurrences of the same salt/tweak.

We prove the AVC simulation theorem for MQOM_Ω in both the ROM and the QROM.

Theorem 3.1 (Simulatability of MQOM_Ω's AVC in the ROM and QROM). *Let $\mathsf{H}_{\mathsf{avc}}$ be a random oracle. Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ be a QPT adversary for the simulation experiment in Figure 7. Assume that $\mathcal{A}_1$ makes at most $Q_{\mathsf{H}_{\mathsf{avc}}}$*

$\mathsf{Expt}^{\mathrm{avc},b}_{\mathsf{MQOM}_{\Omega},\mathcal{A}}(\lambda)$

$(\mathtt{vk},\mathtt{sk}) \leftarrow \mathsf{MQOM.Gen}(1^\lambda)$
$(\mathtt{vk},x) \coloneqq \mathsf{Parse}(\mathtt{sk})$
$\delta \coloneqq \mathsf{FirstBits}_\lambda(x)$
$(\mathtt{state},\{(\mathtt{salt}^{(q)},\mathbf{i}^{*(q)})\}_{q\in[Q]}) \leftarrow \mathcal{A}_0(\mathtt{vk})$
assert no collision in prefixes of salts $\mathsf{FirstBits}_{\lambda-12}(\mathtt{salt}^{(q)})$
for $q \in [Q]$ **do**
  if $b = 0$ **then**
    $\mathsf{out}^{(q)} \coloneqq \mathcal{O}^{\mathrm{O}}_{\mathrm{real}}(\mathtt{vk},x,\delta,\mathtt{salt}^{(q)},\mathbf{i}^{*(q)})$
  else
    $\mathsf{out}^{(q)} \coloneqq \mathcal{O}^{\mathrm{O}}_{\mathrm{ideal}}(\mathtt{vk},\mathtt{salt}^{(q)},\mathbf{i}^{*(q)})$
$b' \leftarrow \mathcal{A}_1^{\mathrm{O}}(\mathtt{state},\{\mathsf{out}^{(q)}\}_{q\in[Q]})$
return b'

$\mathcal{O}^{\mathrm{O}}_{\mathrm{ideal}}(\mathtt{vk},\mathtt{salt},\mathbf{i}^*)$

for $e \in [\tau]$ **do**
  $i^* \coloneqq \mathbf{i}^*[e]$
  $\mathtt{path}[e] \leftarrow (\{0,1\}^\lambda)^d$
  $\mathtt{com}[e][i^*] \leftarrow \{0,1\}^{2\lambda}$
  $\mathtt{pdecom}[e] \coloneqq (\mathtt{path}[e],\mathtt{com}[e][i^*])$
  $(\mathtt{com}[e],(\mathtt{seed}[e][i])_{i\neq i^*},(\mathtt{tape}[e][i])_{i\neq i^*}) \coloneqq \mathsf{MQOM.Recon}^{\mathrm{O}}(\mathtt{salt},\mathtt{pdecom}[e],e,i^*)$
  $\Delta_x^{(1)}[e] \leftarrow \mathbb{F}^{n-\lambda/k}$
  $\alpha_0[e] \leftarrow \mathbb{K}^\eta$
return $(\mathtt{pdecom},\Delta_x^{(1)},\alpha_0)$

$\mathcal{O}^{\mathrm{O}}_{\mathrm{real}}(\mathtt{vk},x,\delta,\mathtt{salt},\mathbf{i}^*)$

for $e \in [\tau]$ **do**
  $i^* \coloneqq \mathbf{i}^*[e]$
  $\mathtt{rseed}[e] \leftarrow \{0,1\}^\lambda$
  $(\mathtt{com}[e],\mathtt{decom}[e],\mathtt{seed}[e],\mathtt{tape}[e]) \coloneqq \mathsf{MQOM.Commit}^{\mathrm{O}}(\mathtt{salt},\mathtt{rseed}[e],\delta,e)$
  $h_{\mathtt{com}}[e] \coloneqq \mathsf{Hash}_6(\mathtt{com}[e])$
  $\Delta_x^{(1)}[e] \coloneqq f_{\mathrm{aux},0}(\mathtt{vk},x,\mathtt{seed}[e],\mathtt{tape}[e])$
$\mathtt{com}_1 \coloneqq \mathsf{Hash}_7(h_{\mathtt{com}},\Delta_x^{(1)})$
$\Gamma \coloneqq \boldsymbol{I}_\eta$ // 3-round version
$\Gamma \coloneqq \mathsf{XOF}_8(\mathtt{com}_1)$ // 5-round version
for $e \in [\tau]$ **do**
  $i^* \coloneqq \mathbf{i}^*[e]$
  $\mathtt{pdecom}[e] \coloneqq \mathsf{MQOM.Open}^{\mathrm{O}}(\mathtt{decom}[e],e,i^*)$
  $\alpha_0[e] \coloneqq f_{\mathrm{aux},1}(\mathtt{vk},x,\mathtt{seed}[e],\mathtt{tape}[e],\Gamma)$
return $(\mathtt{pdecom},\Delta_x^{(1)},\alpha_0)$

$f_{\mathrm{aux},0}(\mathtt{vk},x,\mathtt{seed},\mathtt{tape})$

for $i \in [N]$ **do**
  $(\bar{x}[i],\bar{u}[i]) \coloneqq \mathsf{Parse}(\mathtt{seed}[i],\mathtt{tape}[i])$
$\Delta_x \coloneqq x - \sum_{i\in[N]} \bar{x}[i]$
$\Delta_x^{(1)} \coloneqq \Delta_x[\lambda : kn]$
return $\Delta_x^{(1)}$

$f_{\mathrm{aux},1}(\mathtt{vk},x,\mathtt{seed},\mathtt{tape},\Gamma)$

$(\mathtt{mseed}_{\mathtt{eq}},\hat{y}) \coloneqq \mathsf{Parse}(\mathtt{vk})$
$(\{\hat{A}_i\},\{\hat{b}_i\})_{i\in[\hat{m}]} \coloneqq \mathsf{ExpandEquations}(\mathtt{mseed}_{\mathtt{eq}})$
for $i \in [N]$ **do**
  $(\bar{x}[i],\bar{u}[i]) \coloneqq \mathsf{Parse}(\mathtt{seed}[i],\mathtt{tape}[i])$
$\Delta_x \coloneqq x - \sum_{i\in[N]} \bar{x}[i]$
$x_0 \coloneqq -\omega_0\Delta_x - \sum_{i\in[N]} \omega_i \cdot \bar{x}[i]$
$u_0 \coloneqq -\sum_{i\in[N]} \omega_i \cdot \bar{u}[i]$
for $i \in [\hat{m}]$ **do**
  $z_{0,i} \coloneqq x_0^\top \cdot \hat{A}_i^\top \cdot x_0$
$z_0 \coloneqq (z_{0,0},\ldots,z_{0,\hat{m}-1})$
$\alpha_0 \coloneqq u_0 + \Gamma \cdot z_0$
return α_0

Fig. 7. Simulation experiment for the scGGM-based AVC openings of MQOM. Here O denotes the oracle environment used in the experiment: it contains the ideal cipher $E^{\pm}$ in the ICM, the random oracle $\mathsf{H}_{\mathsf{avc}}$ in the (Q)ROM, and the remaining random oracles used to compute the auxiliary values.The auxiliary procedures compute the MQOM-specific values $\Delta_x^{(1)}$ and α_0 from the full tree-derived seeds and tapes in the real branch, while the ideal branch samples them directly.

(classical/quantum) queries to $\mathsf{H}_{\mathsf{avc}}$. *Then there exists a QPT adversary* $\mathcal{A}_{\mathrm{pgow}}$ *such that*

$$\mathsf{Adv}^{\mathrm{avc}}_{\mathsf{MQOM}_{\Omega},\mathcal{A}}(\lambda) \le \begin{cases} \mathsf{Adv}^{\mathrm{pgow}}_{\mathsf{F}_{\mathsf{MQOM}},\mathcal{A}_{\mathrm{pgow}}}(\lambda) + 2\Pr[\delta = 0^{\lambda}] + \frac{2Q\tau(2N-d-3)}{2^{\lambda}} & \text{in the ROM,} \\ 2\sqrt{Q_{\mathsf{H}_{\mathsf{avc}}}\mathsf{Adv}^{\mathrm{pgow}}_{\mathsf{F}_{\mathsf{MQOM}},\mathcal{A}_{\mathrm{pgow}}}(\lambda)} + 2\Pr[\delta = 0^{\lambda}] + \frac{2Q\tau(2N-d-3)}{2^{\lambda}} & \text{in the QROM.} \end{cases}$$

The running time of $\mathcal{A}_{\mathrm{pgow}}$ *is approximately that of* $\mathcal{A}$. *The adversary* $\mathcal{A}_{\mathrm{pgow}}$ *makes at most* $2Q_{\mathsf{H}_{\mathsf{avc}}}$ *queries to its guessing oracle.*

Proof sketch. Our key step is to resolve the secret-key-root circularity by switching to an oracle that is independently random on the hidden tree, commitment, and tape inputs, thereby randomizing the scGGM opening and the secret-key-dependent values simultaneously. Under the switched oracle, after excluding the internal-collision event, resulting in the loss of $2\Pr[\delta = 0^{\lambda}] + 2Q\tau(2N-d-3)/2^{\lambda}$, the real and simulated transcripts are identically distributed by the bijectivity of the auxiliary map, while any ROM adversary distinguishing the two oracle worlds must query a hidden input from which the secret-key prefix δ can be reconstructed, yielding a reduction to PGOW. In the QROM, the SC-O2H lemma extracts such a hidden input with square-root loss; see the full proof in Appendix E.

4 Security of scGGM-based AVC of modified MQOM in the ICM

This section proves the simulation security of the scGGM-based AVC used in MQOM_{Ω}, the modified MQOM, in the ICM. We first recall the H-coefficient technique. If the reader is unfamiliar with the security proof of the cGGM-tree-based AVC, we recommend seeing our brief review in Appendix D. We then discuss the parameter settings, our modification on Ω, and give the proof of our main theorem (Theorem 4.2).

***H*-coefficient technique**: We employ the H-coefficient technique [Pat09, CS14, JN18] to upper-bound the distinguisher's advantage between the real and ideal experiments with oracle sets $\mathcal{O}_{\mathrm{real}}$ and $\mathcal{O}_{\mathrm{ideal}}$. An interaction transcript, denoted by θ, consists of the ordered list of query–answer pairs. We consider deterministic distinguishers. Let Θ_{real} (resp. Θ_{ideal}) denote the transcript random variable in the real (resp. ideal) world. A transcript θ is *attainable* if $\Pr[\Theta_{\mathrm{ideal}} = \theta] > 0$.

Theorem 4.1 (H-coefficient Technique). *We consider the two sets of oracles* $\mathcal{O}_{\mathrm{real}}$ *and* $\mathcal{O}_{\mathrm{ideal}}$. *Let* $\mathcal{D}$ *be a deterministic distinguisher. Let* $\mathcal{T}$ *be a set of all attainable transcripts made by the distinguisher* $\mathcal{D}$. *We then consider a partition* Bad *and* Good *of* $\mathcal{T}$, *that is,* $\mathcal{T} = \mathsf{Bad} \sqcup \mathsf{Good}$. *Suppose that there exist non-negative* ϵ_{good} *and* ϵ_{bad} *such that for all* $\theta \in \mathsf{Good}$, *it holds that* $\frac{\Pr[\Theta_{\mathrm{real}}=\theta]}{\Pr[\Theta_{\mathrm{ideal}}=\theta]} \ge 1-\epsilon_{\mathrm{good}}$ *and* $\Pr[\Theta_{\mathrm{ideal}} \in \mathsf{Bad}] \le \epsilon_{\mathrm{bad}}$. *Then, we have* $|\Pr[\mathcal{D}^{\mathcal{O}_{\mathrm{real}}} = 1] - \Pr[\mathcal{D}^{\mathcal{O}_{\mathrm{ideal}}} = 1]| \le \epsilon_{\mathrm{good}} + \epsilon_{\mathrm{bad}}$.

Appendix F gives warm-up application of the H-coefficient technique with one-wayness, where we show the tweaked CCR (TCCR) property of EncFF with hidden shift x^* where the distinguisher is given $(f, f(x^*))$.

Parameter settings, packing, embedding, and conversions: We first review the definitions of fields: They define $\mathrm{GF}(2^4)$ by $\mathrm{GF}(2)[\rho]/\langle\rho^4+\rho+1\rangle$, $\mathrm{GF}(2^8)$ by $\mathrm{GF}(2)[\xi]/\langle\xi^8+\xi^4+\xi^3+\xi+1\rangle$, $\mathrm{GF}(2^{16})$ by $\mathrm{GF}(2^8)[\nu]/\langle\nu^2+\nu+\xi^5\rangle$, We explicitly denote the field lifting $\mathbb{F} \hookrightarrow \mathbb{K}$ by ι [BBFR25, Section 3.1]. They used natural liftings $\mathrm{GF}(2) \hookrightarrow \mathrm{GF}(2^8), \mathrm{GF}(2^{16})$ and $\mathrm{GF}(2^8) \hookrightarrow \mathrm{GF}(2^{16})$. From $\mathrm{GF}(2^4)$ to $\mathrm{GF}(2^8)$ (and $\mathrm{GF}(2^{16})$), they used the field morphism, which maps $\rho \in \mathrm{GF}(2^4)$ to $\xi^7+\xi^6+\xi^5 \in \mathrm{GF}(2^8)$. Note that they are $\mathbb{F}$-linear. We often omit this ι if it is obvious from the context, e.g., for $a \in \mathbb{F}$ and $z \in \mathbb{K}$, we write $a \cdot z$ instead of $\iota(a)z$

We explicitly consider linear conversions $\mathsf{B2F} : \{0,1\}^k \to \mathbb{F}$ and $\mathsf{B2K} : \{0,1\}^{k\mu} \to \mathbb{K}$, which are linear isomorphisms by our assumption and adopted from those in [BBFR25, Section 2.3]. We denote those linear inverses as $\mathsf{F2B}$ and $\mathsf{K2B}$.

Due to linearity, we have $\mathsf{F2B}(x_0) \oplus \mathsf{F2B}(x_1) = \mathsf{F2B}(x_0 + x_1)$. We extend their domain and range as vector and matrix spaces: For example, if $x \in \mathbb{F}^n$, we have $\mathsf{F2B}(x) \in \{0,1\}^{kn}$.

Fixing an $\mathbb{F}$-basis $(\beta_0, \dots, \beta_{\mu-1})$ of $\mathbb{K}$ as in [BBFR25], we also define the $\mathbb{F}$-linear isomorphism

$$\mathsf{F2K} : \mathbb{F}^{\mu} \to \mathbb{K}, \qquad (a_0, \dots, a_{\mu-1}) \mapsto \sum_{j=0}^{\mu-1} a_j \beta_j,$$

which is known as ϕ in [BBFR25], and denote its inverse by $\mathsf{K2F}$.

MQOM employs a linear orthomorphism $\psi : \{0,1\}^\lambda \to \{0,1\}^\lambda$ in their $\mathsf{H}_{\mathsf{avc}} = \mathsf{EncFF}$, which is defined as $\psi(s_l \| s_r) := (s_l \oplus s_r) \| s_l$ for $s_l, s_r \in \{0,1\}^{\lambda/2}$. We will use $\bar{\psi} : s \mapsto \psi(s) \oplus s$, which is $\bar{\psi}(s_l \| s_r) = ((s_l \oplus s_r) \| s_l) \oplus (s_l \| s_r) = s_r \| (s_l \oplus s_r)$ for $s_l, s_r \in \{0,1\}^{\lambda/2}$. We note that ψ and $\bar{\psi}$ are efficiently invertible.

We consider GF(2) parameter sets of MQOM because of the following reasons: Let $\mathbb{F} = \mathrm{GF}(2^k)$ and $\mathbb{K} = \mathrm{GF}(2^{k\mu})$. In all parameter sets, we have $m = n$ and $\hat{m} = m/\mu = n/\mu$. By the way, we have $|x| - \lambda = kn - \lambda = \lambda/4$ for all GF(2) parameter sets, but it is at least $(3/4)\lambda$ for other parameter sets. Since our reduction requires $2^{|x|-\lambda}$ is small, we only consider GF(2) parameter sets. If we adopt the RGOW assumption [FR26], there is no problem and reduction does not require $2^{|x|-\lambda}$ factor. Since we only consider GF(2) parameter sets, we have $|\Delta_x^{(1)}[e]| + |\alpha_0[e]| < 2\lambda$ and $n_b = 2$.

We recall that, Line 10 of $\mathsf{MQOM.Sign}$ (Figure 5), "$(\bar{x}[e][i], \bar{u}[e][i]) := \mathsf{Parse}(\mathtt{seed}[e][i], \mathtt{tape}[e][i])$". The bit length of the LHS is $kn + k\mu\eta$, while $|\mathtt{seed}[e][i]| + |\mathtt{tape}[e][i]| = (1 + n_b)\lambda$, where $kn + k\mu\eta < (1 + n_b)\lambda$ for all parameter sets. For technical reasons, we do not want to truncate $\mathtt{tape}[e][i]$ as in Section 3. Hence, we consider $\bar{u}'[e][i] \in \mathbb{K}^{\eta'}$ with $\eta' := ((1 + n_b)\lambda - kn)/(k\mu)$ and let

$$(\bar{x}[e][i], \bar{u}'[e][i]) := \mathsf{Parse}(\mathtt{seed}[e][i], \mathtt{tape}[e][i]).$$

We expand the length of $\alpha_0[e]$ also and we will use $\Gamma' = [\Gamma; O_{(\eta'-\eta)\times\hat{m}}] \in \mathbb{K}^{\eta'\times\hat{m}}$.

We finally note that the implementation of E in MQOM [BBFR25].

- For the L1 parameter sets, E is instantiated by the AES-128 encryption algorithm.
- For the L3 parameter sets, E is defined as follows: $E : (k, p) \in \{0,1\}^{192} \times \{0,1\}^{192} \mapsto \mathsf{Enc}(k \| 0^{64}, p \| 0^{64})[0 : 192]$, where Enc is the encryption algorithm of Rijndael-256-256.
- For the L5 parameter sets, E is instantiated by the Rijndael-256-256 encryption algorithm.

Remark 4.1. Thus, formally speaking, the function E used for the L3 parameter sets is not itself an ideal cipher on $\{0,1\}^{192}$, since it is obtained by truncating the output of a 256-bit block cipher. Nevertheless, this does not cause any issue in our proof.

For the L3 parameter sets, we formulate the proof using the underlying 256-bit ideal cipher Enc, with the prescribed zero-padding of keys and inputs. We consider an augmented experiment that retains the full 256-bit cipher outputs and correspondingly augments $\mathtt{com}$, $\Delta_x^{(1)}$, and α_0, rather than discarding the additional coordinates. Both the challenger and the adversary operate on these augmented values, and cipher-query transcripts and output-side collision events are defined using full 256-bit blocks. Thus, for each fixed key, distinct cipher inputs have distinct outputs, and inverse queries are well defined. The augmentation is chosen so that the original real and ideal transcripts are recovered by the prescribed blockwise truncation and coordinate projection. Consequently, any distinguisher for the original experiment yields a distinguisher for the augmented experiment with the same advantage by applying this projection before proceeding.

Modification of MQOM: In the proof, we assume that $\omega_{i^*} \neq 0$, while the MQOM specification sets $\omega_0 = 0 \in \mathbb{K}$. To resolve this issue, we need to modify the specification so that the evaluation set excludes 0. Let $\Omega = \{\omega_0, \omega_1, ...\}$ denote the sequence of ω_i's generated by the Gray code in $\mathbb{K}$, where $\omega_0 = 0$. Depending on N and $\mathbb{K}$ in Table 1, we modify the set of the evaluation points as follows:

- For the parameter set $N = 256$ over $\mathbb{K} = \mathrm{GF}(256)$, we remove the element ω_0 and use $\Omega' = \{\omega'_0, ..., \omega'_{254}\} := \{\omega_1, ..., \omega_{255}\}$. Equivalently, the evaluation domain size becomes $N = 255$. Due to this modification, we remove the last two leaves $\mathtt{node}[e][510]$ and $\mathtt{node}[e][511]$ from each tree and set $\mathtt{seed}[e][0] := \mathtt{node}[e][255], \ldots, \mathtt{seed}[e][254] := \mathtt{node}[e][509]$ in Line 7 of $\mathsf{MQOM.Commit}$ in Figure 6. To keep the format of $\mathtt{pdecom}$ unchanged, we use a dummy node when $\mathtt{i}^*[e] = 0$.
- For the parameter set $N = 2048$ over $\mathbb{K} = \mathrm{GF}(2^{16})$, we similarly remove ω_0 and use $\Omega' = \{\omega'_0, ..., \omega'_{2047}\} := \{\omega_1, ..., \omega_{2048}\}$. With this modification, every evaluation point is nonzero.

An alternative solution, without changing Ω, would be to reformulate the proof using α_1 as the primary variable instead of α_0 as in the concurrent work [FR26].

4.1 Main Theorem

We want to show the computational indistinguishability between the two games in Figure 7. To do so, we first note that, by the correctness of MQOM's AVC (Figure 6), the following two procedures are identical for any $e \in [\tau]$ and $i^* \in [N]$:

– Procedure 1: Compute $(\mathtt{com}, \mathtt{decom}, \mathtt{seed}, \mathtt{tape}) := \mathsf{MQOM.Commit}(\mathtt{salt}, \mathtt{rseed}, \delta, e)$. Compute $\mathtt{pdecom} := \mathsf{MQOM.Open}(\mathtt{decom}, e, i^*)$. Parse $\mathtt{pdecom} = (\mathtt{path}, \mathtt{com}[i^*])$. Output $(\mathtt{com}, \mathtt{seed}, \mathtt{tape}, \mathtt{path})$.
– Procedure 2: Compute $\mathtt{pdecom} = (\mathtt{path}, \mathtt{com}[i^*])$ correctly from $\mathtt{salt}, \mathtt{rseed}, \delta$, and e. Reconstruct commitments, seeds, and tapes by $(\mathtt{com}, (\mathtt{seed}[i])_{i\neq i^*}, (\mathtt{tape}[i])_{i\neq i^*}) := \mathsf{MQOM.Recon}(\mathtt{salt}, \mathtt{pdecom}, e, i^*)$. Compute $\mathtt{seed}[i^*] := \delta \oplus \bigoplus_{i\in[1:d+1]} \mathtt{path}[i]$ and $\mathtt{tape}[i^*] := \mathsf{H}_{\mathsf{avc}}(\mathtt{seed}[i^*], t_{\mathtt{salt},3,e,0}) \,\|\, \mathsf{H}_{\mathsf{avc}}(\mathtt{seed}[i^*], t_{\mathtt{salt},3,e,1})$. Output $(\mathtt{com}, \mathtt{seed}, \mathtt{tape}, \mathtt{path})$.

Using this equivalence and expanding the algorithms, the games $\mathsf{Expt}^{\mathrm{avc},b}_{\mathsf{MQOM}_\Omega,\mathcal{A}}(\lambda)$ for $b = 0, 1$ can be rewritten as Figure 8. We note that if $N = 255$, then when $i^* = 0$, we need to use S_{d-1} instead of S_d.

We note that $\mathtt{node}[\mathsf{path}_{j-1}(i^*)] = S_{j-1} \oplus \delta$, where $S_{j-1} := \oplus_{j'\in[1:j]}\mathtt{path}[j']$ due to the relation in eq.(1). Finally, although the computation of $\mathsf{MQOM.Recon}^E$ in Line 14 of $\mathcal{O}_{\mathrm{ideal}}$ may appear redundant, it is included to ensure that the queries to E are aligned with those made by $\mathcal{O}_{\mathrm{real}}$.

Theorem 4.2. *Let $\mathbb{F} = \mathrm{GF}(2)$ and $n_b = 2$. Let $\mathcal{A}$ be a distinguisher that makes at most Q_E queries to $E^{\pm}$. Suppose that the conjecture (Definition 4.3) holds. There exists an adversary $\mathcal{A}_{\mathrm{ow}}$ satisfying*

$$\mathsf{Adv}^{\mathrm{avc}}_{\mathsf{MQOM}_\Omega,\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\mathrm{ow}}_{\mathsf{F}_{\mathsf{MQOM}},\mathcal{A}_{\mathrm{ow}}}(\lambda).$$

The running time of $\mathcal{A}_{\mathrm{ow}}$ is that of $\mathcal{A}$ plus $2^{kn-\lambda} \cdot \mathsf{poly}(Q', \lambda)$, where $Q' := Q_E + Q \cdot \tau \cdot (5N - d - 4)$.

Proof of Theorem 4.2: We first introduce notation for query-answer transcripts. We maintain the following query-answer lists:

– $\mathcal{Q}_{\mathrm{node}}$ stores information related to $\mathtt{path}[e][j]$. For each opening transcript, and for all $e \in [\tau]$ and $j = 2, \ldots, d$, we add the tuple $(t_{\mathtt{salt},2,e,j-1}, b, S_{j-1}, \mathtt{path}[e][j])$ to $\mathcal{Q}_{\mathrm{node}}$.
– $\mathcal{Q}_{\mathrm{com}}$ stores information related to $\mathtt{com}[e][i^*]$. For each query, and for all $e \in [\tau]$, we add the tuples $(t_{\mathtt{salt},0,e,0}, S_d, c_0)$ and $(t_{\mathtt{salt},1,e,0}, S_d, c_1)$ to $\mathcal{Q}_{\mathrm{com}}$.
– $\mathcal{Q}_{\mathrm{aux}}$ stores information related to $\Delta^{(1)}_x[e]$ and $\alpha'_0[e]$. For each query, and for all $e \in [\tau]$, we add the tuple $(t_{\mathtt{salt},3,e,0}, t_{\mathtt{salt},3,e,1}, S_d, \Delta^{(1)}_x[e], \alpha'_0[e])$ to $\mathcal{Q}_{\mathrm{aux}}$.
– $\mathcal{Q}_E$ stores all queries to the ideal cipher E. Specifically, it contains tuples (t, s, y) such that either $E^+(t, s) = y$ (an encryption query) or $E^-(t, y) = s$ (a decryption query). We note that the computations performed by $\mathsf{MQOM.Recon}$ in Figure 8 are recorded in $\mathcal{Q}_E$ rather than in $\mathcal{Q}_{\mathrm{node}}$, $\mathcal{Q}_{\mathrm{com}}$, or $\mathcal{Q}_{\mathrm{aux}}$. For each tree, $\mathsf{MQOM.Recon}$ makes at most $N - d$ node queries, $2(N - 1)$ commitment queries, and $2(N - 1)$ tape queries to E^+.

The transcript θ for the H-coefficient technique consists of $(\mathtt{vk}, \mathtt{sk})$ together with $\mathcal{Q}_{\mathrm{node}}$, $\mathcal{Q}_{\mathrm{com}}$, $\mathcal{Q}_{\mathrm{aux}}$, and $\mathcal{Q}_E$. If there is any inconsistency among the recorded query-answer pairs, an adversary may distinguish the real and ideal games. We therefore define *bad transcripts* to capture such events.

Bad transcripts with respect to nodes and commitments: We first consider bad events related to the computation of tree nodes and commitments. Intuitively, each node or commitment computation corresponds to an ideal-cipher evaluation on a particular internal input and output pair. A bad event occurs when a pair recorded in $\mathcal{Q}_E$ collides with one of these internal values.

We distinguish two types of collisions:

– An *input collision* occurs when a value used as an input to the ideal cipher in a node or commitment computation also appears as a cipher input in $\mathcal{Q}_E$. For example, in the case of commitments, this occurs if there exist $(t, s, *) \in \mathcal{Q}_{\mathrm{com}}$ and $(t, s \oplus \delta, *) \in \mathcal{Q}_E$.
– An *output collision* occurs when a value used as an output of the ideal cipher in a node or commitment computation also appears as a cipher output in $\mathcal{Q}_E$. For example, in the case of commitments, this occurs if there exist $(t, s, z) \in \mathcal{Q}_{\mathrm{com}}$ and $(t, *, \psi(s \oplus \delta) \oplus z) \in \mathcal{Q}_E$.

For brevity, we denote these two classes by the subscripts F and B, respectively.

Formally, the bad transcripts with respect to nodes and commitments are defined as follows:

Definition 4.1. *We define the following four bad events:*

– $\mathsf{Bad}_{\mathrm{node},F}$*: There exist* $(t, b, s, z) \in \mathcal{Q}_{\mathrm{node}}$ *and* $(t, s \oplus \delta, *) \in \mathcal{Q}_E$.
– $\mathsf{Bad}_{\mathrm{node},B}$*: There exist* $(t, b, s, z) \in \mathcal{Q}_{\mathrm{node}}$ *and* $(t, *, \psi(s \oplus \delta) \oplus b \cdot (s \oplus \delta) \oplus z) \in \mathcal{Q}_E$.
– $\mathsf{Bad}_{\mathrm{com},F}$*: There exist* $(t, s, z) \in \mathcal{Q}_{\mathrm{com}}$ *and* $(t, s \oplus \delta, *) \in \mathcal{Q}_E$.
– $\mathsf{Bad}_{\mathrm{com},B}$*: There exist* $(t, s, z) \in \mathcal{Q}_{\mathrm{com}}$ *and* $(t, *, \psi(s \oplus \delta) \oplus z) \in \mathcal{Q}_E$.

$\mathsf{Expt}^{\mathrm{avc},b}_{\mathsf{MQOM}_{\Omega},\mathcal{A}}(\lambda)$

1 : $(\mathtt{vk},\mathtt{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$
2 : $(\mathtt{vk},x) := \mathsf{Parse}(\mathtt{sk})$
3 : $\delta := \mathsf{F2B}(x)[0:\lambda]$
4 : $(\mathtt{mseed}_{\mathtt{eq}},\{\hat{y}_i\}) := \mathsf{Parse}(\mathtt{vk})$
5 : $(\{\hat{A}_i\},\{\hat{b}_i\}) := \mathsf{ExpandEquations}(\mathtt{mseed}_{\mathtt{eq}})$
6 : $(\mathtt{state},\{\mathtt{salt}^{(q)},\mathbf{i}^{*(q)},\}_{q\in[Q]}) \leftarrow \mathcal{A}_0(\mathtt{vk})$
7 : **assert** no collision in prefixes of salts $\mathsf{FirstBits}_{\lambda-12}(\mathtt{salt}^{(q)})$
8 : **for** $q \in [Q]$ **do**
9 : **if** $b = 0$ **then**
10 : $\mathsf{out}^{(q)} := \mathcal{O}_{\mathrm{real}}(\mathtt{salt}^{(q)},\mathbf{i}^{*(q)})$
11 : **else** $\mathsf{out}^{(q)} := \mathcal{O}_{\mathrm{ideal}}(\mathtt{salt}^{(q)},\mathbf{i}^{*(q)})$
12 : $b' \leftarrow \mathcal{A}_1^{E^{\pm}}(\mathtt{state},\{\mathsf{out}^{(q)}\}_{q\in[Q]})$
13 : **return** b'

$\mathcal{O}_{\mathrm{ideal}}(\mathtt{salt},\mathbf{i}^*)$

1 : **for** $e \in [\tau]$ **do**
2 : $i^* := \mathbf{i}^*[e]$
3 : $\mathtt{path}[e][1] \leftarrow \{0,1\}^\lambda$
4 : **for** $j = 2,\dots,d$ **do**
5 : $b := \mathsf{copath}_j(i^*) \bmod 2$
6 : $S_{j-1} := \bigoplus_{j'\in[1:j]} \mathtt{path}[e][j']$
7 : $\mathtt{path}[e][j] \leftarrow \{0,1\}^\lambda$
8 : $S_d := \bigoplus_{j'\in[1:d+1]} \mathtt{path}[e][j']$
9 : $c_0 \leftarrow \{0,1\}^\lambda$
10 : $c_1 \leftarrow \{0,1\}^\lambda$
11 : $\mathtt{com}[e][i^*] := c_0 \,\|\, c_1$
12 : $\mathtt{pdecom}[e] := (\mathtt{path}[e],\mathtt{com}[e][i^*])$
13 : $(\mathtt{com}[e],\mathtt{seed}[e],\mathtt{tape}[e]) := \mathsf{MQOM.Recon}^E(\mathtt{salt},\mathtt{pdecom}[e],e,i^*)$
14 : **for** $e \in [\tau]$ **do**
15 : $\Delta_x^{(1)}[e] \leftarrow \mathbb{F}^{n-\lambda/k}$
16 : $\alpha_0'[e] \leftarrow \mathbb{K}^{\eta'}$
17 : Compute $\Gamma' := [\Gamma; O_{(\eta'-\eta)\times\hat{m}}] \in \mathbb{K}^{\eta'\times\hat{m}}$
18 : **return** $(\mathtt{pdecom},\Delta_x^{(1)},\alpha_0')$

$\mathcal{O}_{\mathrm{real}}(\mathtt{salt},\mathbf{i}^*)$

1 : **for** $e \in [\tau]$ **do**
2 : $i^* := \mathbf{i}^*[e]$
3 : $\mathtt{path}[e][1] \leftarrow \{0,1\}^\lambda$
4 : **for** $j = 2,\dots,d$ **do**
5 : $b := \mathsf{copath}_j(i^*) \bmod 2$
6 : $S_{j-1} := \bigoplus_{j'\in[1:j]} \mathtt{path}[e][j']$
7 : $\mathtt{path}[e][j] := E(t_{\mathtt{salt},2,e,j-1}, S_{j-1}\oplus\delta) \oplus \psi(S_{j-1}\oplus\delta) \oplus b\cdot(S_{j-1}\oplus\delta)$
8 : $S_d := \bigoplus_{j'\in[1:d+1]} \mathtt{path}[e][j']$
9 : $\mathtt{seed}[e][i^*] := S_d \oplus \delta$
10 : $c_0 := E(t_{\mathtt{salt},0,e,0}, S_d\oplus\delta) \oplus \psi(S_d\oplus\delta)$
11 : $c_1 := E(t_{\mathtt{salt},1,e,0}, S_d\oplus\delta) \oplus \psi(S_d\oplus\delta)$
12 : $\mathtt{com}[e][i^*] := c_0 \,\|\, c_1$
13 : $\zeta_0 := E(t_{\mathtt{salt},3,e,0}, S_d\oplus\delta) \oplus \psi(S_d\oplus\delta)$
14 : $\zeta_1 := E(t_{\mathtt{salt},3,e,1}, S_d\oplus\delta) \oplus \psi(S_d\oplus\delta)$
15 : $\mathtt{tape}[e][i^*] := \zeta_0 \,\|\, \zeta_1$
16 : $\mathtt{pdecom}[e] := (\mathtt{path}[e],\mathtt{com}[e][i^*])$
17 : $(\mathtt{com}[e],\mathtt{seed}[e],\mathtt{tape}[e]) := \mathsf{MQOM.Recon}^E(\mathtt{salt},\mathtt{pdecom}[e],e,i^*)$
18 : **for** $i \in [N]$ **do**
19 : $m := \mathtt{seed}[e][i] \,\|\, \mathtt{tape}[e][i] \in \{0,1\}^{3\lambda}$
20 : $\bar{x}[e][i] := \mathsf{B2F}(m[0:kn]) \in \mathbb{F}^n$
21 : $\bar{u}'[e][i] := \mathsf{B2K}(m[kn:3\lambda]) \in \mathbb{K}^{\eta'}$
22 : $\Delta_x[e] := x - \sum_{i\in[N]} \bar{x}[e][i] \in \mathbb{F}^n$
23 : $\Delta_x^{(1)}[e] := \Delta_x[e][\lambda/k : n] \in \mathbb{F}^{n-\lambda/k}$
24 : $x_0[e] := -\omega_0\cdot\Delta_x[e] - \sum_{i\in[N]} \omega_i\cdot\bar{x}[e][i] \in \mathbb{K}^n$
25 : $u_0'[e] := -\sum_{i\in[N]} \omega_i\cdot\bar{u}'[e][i] \in \mathbb{K}^{\eta'}$
26 : Compute $\Gamma' := [\Gamma; O_{(\eta'-\eta)\times\hat{m}}] \in \mathbb{K}^{\eta'\times\hat{m}}$
27 : **for** $e \in [\tau]$ **do**
28 : **for** $i \in [\hat{m}]$ **do**
29 : $z_{0,i} := x_0[e]^\top \cdot \hat{A}_i \cdot x_0[e] \in \mathbb{K}$
30 : $z_0 := (z_{0,0},\dots,z_{0,\hat{m}-1}) \in \mathbb{K}^{\hat{m}}$
31 : $\alpha_0'[e] := u_0'[e] + \Gamma'\cdot z_0 \in \mathbb{K}^{\eta'}$
32 : **return** $(\mathtt{pdecom},\Delta_x^{(1)},\alpha_0')$

Fig. 8. Expanded $\mathsf{Expt}^{\mathrm{avc},b}_{\mathsf{MQOM}_{\Omega},\mathcal{A}}(\lambda)$. $\Gamma := I_{\eta\times\eta}$ in the 3-round version and $\Gamma := \mathsf{XOF}_8(\mathtt{com}_1) \in \mathbb{K}^{\eta\times\hat{m}}$ in the 5-round version, where $\mathtt{com}_1 := \mathsf{Hash}_7(h_{\mathtt{com}},\Delta_x^{(1)})$ and $h_{\mathtt{com}}[e] := \mathsf{Hash}_6(\mathtt{com}[e])$.

Bad transcripts with respect to $\Delta_x^{(1)}[e]$ and $\alpha_0'[e]$ in $\mathcal{Q}_{\mathrm{aux}}$: Intuitively, we define bad transcripts capturing when queries to the ideal cipher E collide with those used to compute $\Delta_x^{(1)}[e]$ and $\alpha_0'[e]$.

The first type of bad event corresponds to a collision on the input side of E. This occurs if there exist $(t_0, t_1, s, *, *) \in \mathcal{Q}_{\mathrm{aux}}$ and $(t_b, s \oplus \delta, *) \in \mathcal{Q}_E$ for some $b \in \{0,1\}$.

The second type of bad event corresponds to a collision on the output side of E. If we consider the real world, this captures queries to outputs of the form $\xi_b \oplus \psi(S_d \oplus \delta)$ for $b \in \{0,1\}$. Hereafter, we explain how to formalize this second event in the ideal world.

Fix q and e. Let $i^* := \mathrm{i}^*[e]$, and define $t_0 := t_{\mathtt{salt},3,e,0}$ and $t_1 := t_{\mathtt{salt},3,e,1}$ for ease of notation.

Recall that $\alpha_0'[e] := u_0'[e] + \Gamma' \cdot z_0 \in \mathbb{K}^{\eta'}$, where $x_0[e] := -\omega_0 \Delta_x - \sum_i \omega_i \cdot \bar{x}[e][i] \in \mathbb{K}^n$, $u_0'[e] := -\sum_i \omega_i \cdot \bar{u}'[e][i] \in \mathbb{K}^{\eta'}$, and $z_0 = (z_{0,0}, \ldots, z_{0,\hat{m}-1})$ with $z_{0,i} := \hat{A}_i(x_0[e]) \in \mathbb{K}$. We also recall that $\bar{x}[e][i] := \mathsf{B2F}(\mathtt{seed}[e][i] \,\|\, \mathtt{tape}[e][i][0 : kn - \lambda]) \in \mathbb{F}^n$, $\bar{u}'[e][i] := \mathsf{B2K}(\mathtt{tape}[e][i][kn - \lambda : 2\lambda]) \in \mathbb{K}^{\eta'}$, and $x = \sum_{i \in [N]} \bar{x}[e][i] + \Delta_x[e]$, where $\Delta_x[e] = 0^{\lambda/k} \,\|\, \Delta_x^{(1)}[e] \in \mathbb{F}^n$.

For convenience, we define

$$\tilde{x}_1 := \textstyle\sum_{i \neq i^*} \bar{x}[e][i] = x - \Delta_x[e] - \bar{x}[e][i^*] \in \mathbb{F}^n, \tag{2}$$

$$\tilde{u}_0' := -\textstyle\sum_{i \neq i^*} \omega_i \cdot \bar{u}'[e][i] = u_0'[e] + \omega_{i^*} \cdot \bar{u}'[e][i^*] \in \mathbb{K}^{\eta'}. \tag{3}$$

Note that the values $\tilde{x}_1$ and $\tilde{u}_0'$ are publicly computable.

Furthermore, we want to have $x_0[e] = \omega_{i^*}(\dot{x} - x)$ for some publicly computable $\dot{x}$. To do so, we let

$$\begin{aligned}
\dot{x} &:= \Delta_x[e] + \tilde{x}_1 - \omega_{i^*}^{-1} \cdot (\omega_0 \Delta_x + \textstyle\sum_{i \neq i^*} \omega_i \bar{x}[e][i]) \in \mathbb{K}^n \\
&= x - \bar{x}[e][i^*] - \omega_{i^*}^{-1} \cdot (\omega_0 \Delta_x[e] + \textstyle\sum_{i \neq i^*} \omega_i \bar{x}[e][i]) \\
&= x - \omega_{i^*}^{-1} \cdot (\omega_0 \Delta_x[e] + \textstyle\sum_{i \in [N]} \omega_i \bar{x}[e][i]) \\
&= x + \omega_{i^*}^{-1} \cdot x_0[e],
\end{aligned} \tag{4}$$

which is publicly computable as we wanted.

For notational convenience, define

$$\delta_\psi := \psi(\delta) \,\|\, \psi(\delta), \quad \tilde{\delta}_\psi := \psi(S_d) \,\|\, \psi(S_d), \quad \text{and } \xi := E(t_0, \delta \oplus S_d) \,\|\, E(t_1, \delta \oplus S_d). \tag{5}$$

Using these, we have

$$\mathtt{tape}[e][i^*] = \zeta_0 \,\|\, \zeta_1 = \delta_\psi \oplus \tilde{\delta}_\psi \oplus \xi. \tag{6}$$

From this, we obtain

$$\begin{aligned}
\Delta_x^{(1)}[e] &= (x - \textstyle\sum_{i \in [N]} \bar{x}[e][i])[\lambda/k : n] = x[\lambda/k : n] - \tilde{x}_1[\lambda/k : n] - \bar{x}[e][i^*][\lambda/k : n] \\
&= x[\lambda/k : n] - \tilde{x}_1[\lambda/k : n] - \mathsf{B2F}(\mathtt{tape}[e][i^*][0 : kn - \lambda]) \\
&= x[\lambda/k : n] - \tilde{x}_1[\lambda/k : n] - \mathsf{B2F}((\delta_\psi \oplus \tilde{\delta}_\psi \oplus \xi)[0 : kn - \lambda]),
\end{aligned} \tag{7}$$

$$\begin{aligned}
\alpha_0'[e] &= u_0'[e] + \Gamma' \cdot \hat{A}(x_0[e]) = \tilde{u}_0' - \omega_{i^*} \cdot \bar{u}'[e][i^*] + \Gamma' \cdot \hat{A}\big(\omega_{i^*} \cdot (x - \dot{x})\big) \\
&= \tilde{u}_0' - \omega_{i^*} \cdot \mathsf{B2K}((\delta_\psi \oplus \tilde{\delta}_\psi \oplus \xi)[kn - \lambda : 2\lambda]) + \omega_{i^*}^2 \Gamma' \cdot \hat{A}(x - \dot{x}),
\end{aligned} \tag{8}$$

where we use $\hat{A}(ws) = w^2 \cdot \hat{A}(-s)$ for $w \in \mathbb{K}$ and $s \in \mathbb{K}^n$ and $x_0[e] = \omega_{i^*} \cdot (\dot{x} - x)$ in eq.(4).

Rearranging eq.(7) and eq.(8), we express ξ as a function of public and recorded values. Specifically, we define functions F_0 and F_1 with range $\{0,1\}^\lambda$ by

$$\begin{aligned}
\xi &= \delta_\psi \oplus \tilde{\delta}_\psi \\
&\quad \oplus \mathsf{F2B}(x[\lambda/k : n] - \tilde{x}_1[\lambda/k : n] - \Delta_x^{(1)}[e]) \,\|\, \mathsf{K2B}\big(\omega_{i^*}^{-1} \cdot (\tilde{u}_0' - \alpha_0'[e]) + \omega_{i^*} \Gamma' \cdot \hat{A}(x - \dot{x})\big) \\
&=: F_0(x, i^*, \tilde{x}_1, S_d, \dot{x}, \Gamma', \tilde{u}_0', \Delta_x^{(1)}, \alpha_0') \,\|\, F_1(x, i^*, \tilde{x}_1, S_d, \dot{x}, \Gamma', \tilde{u}_0', \Delta_x^{(1)}, \alpha_0').
\end{aligned} \tag{9}$$

Finally, we consider an augmented query-answer list $\mathcal{Q}'_{\mathrm{aux}}$ that stores $(t_0, t_1, S_d, \Delta_x^{(1)}[e], \alpha_0'[e]; i^*, \tilde{x}_1, \dot{x}, \tilde{u}_0', \Gamma')$. We define the corresponding sets of bad transcripts as follows:

Definition 4.2. *We define the following bad events:*

- $\mathsf{Bad}_{\mathrm{aux},F}$*: There exist* $(t_0, t_1, S_d, *, *; \ldots) \in \mathcal{Q}'_{\mathrm{aux}}$ *and* $(t_b, \delta \oplus S_d, *) \in \mathcal{Q}_E$ *for some* $b \in \{0,1\}$.
- $\mathsf{Bad}_{\mathrm{aux},B,b}$ *for* $b = 0, 1$*: There exist* $(t_0, t_1, S_d, \Delta_x^{(1)}[e], \alpha_0'[e]; i^*, \tilde{x}_1, \dot{x}, \tilde{u}_0', \Gamma') \in \mathcal{Q}'_{\mathrm{aux}}$ *and* $(t_b, *, \xi_b) \in \mathcal{Q}_E$ *such that* $\xi_b = F_b(x, i^*, \tilde{x}_1, S_d, \dot{x}, \Gamma', \tilde{u}_0', \Delta_x^{(1)}, \alpha_0')$.

Bad transcripts: impossible collisions in $E(t,\cdot)$: We consider potential collisions in $E(t,\cdot)$ arising from pairs of entries in $\mathcal{Q}_{\mathrm{node}}$, $\mathcal{Q}_{\mathrm{com}}$, and $\mathcal{Q}_{\mathrm{aux}}$, produced by $\mathcal{O}_{\mathrm{ideal}}$ (and similarly by $\mathcal{O}_{\mathrm{real}}$). For example, consider two (possibly distinct) entries $(t,b,s,z),(t',b',s',z') \in \mathcal{Q}_{\mathrm{node}}$ such that $t = t'$ and $\psi(s\oplus\delta)\oplus b\cdot(s\oplus\delta)\oplus z = \psi(s'\oplus\delta)\oplus b'\cdot(s'\oplus\delta)\oplus z'$. Such collisions would require two entries with the same tweak t. However, since the game aborts upon any tweak collision, these events cannot occur.

Bounding $\Pr[\Theta_{\mathrm{ideal}} \in \mathsf{Bad}]$*:* We define $\mathsf{Bad} \coloneqq \mathsf{Bad}_{\mathrm{node},F} \cup \mathsf{Bad}_{\mathrm{node},B} \cup \mathsf{Bad}_{\mathrm{com},F} \cup \mathsf{Bad}_{\mathrm{com},B} \cup \mathsf{Bad}_{\mathrm{aux},F} \cup \mathsf{Bad}_{\mathrm{aux},B,0} \cup \mathsf{Bad}_{\mathrm{aux},B,1}$. We now upper-bound $\Pr[\Theta_{\mathrm{ideal}} \in \mathsf{Bad}]$. To this end, we construct a reduction algorithm $\mathcal{A}_{\mathrm{ow}}$ that simulates the ideal experiment as follows:

1. Receive $\mathtt{vk}$ from its challenger and compute $\hat{F}_{\mathtt{mseed}_{\mathrm{eq}},\hat{y}}$.
2. Fix a random tape of $\mathcal{A}$ and run it as a deterministic distinguisher. Simulate the ideal experiment while maintaining $\mathcal{Q}_{\mathrm{node}}$, $\mathcal{Q}_{\mathrm{com}}$, $\mathcal{Q}'_{\mathrm{aux}}$, and $\mathcal{Q}_E$.
3. If the game does not abort, attempt to find x' such that $\hat{F}_{\mathtt{mseed}_{\mathrm{eq}},\hat{y}}(x') = 0$ as follows:
   - $\mathsf{Bad}_{\mathrm{node},F}$: For each $t = t_{\mathtt{salt},2,e,j-1}$, find $(t,b,s,z) \in \mathcal{Q}_{\mathrm{node}}$ and $(t,s',*) \in \mathcal{Q}_E$. Let $x_0 \coloneqq \mathsf{B2F}(s\oplus s') \in \mathbb{F}^{\lambda/k}$. For all $x_1 \in \mathbb{F}^{n-\lambda/k}$, check whether $x' \coloneqq (x_0,x_1) \in \mathbb{F}^n$ satisfies $\hat{F}_{\mathtt{mseed}_{\mathrm{eq}},\hat{y}}(x') = 0$, and output x' if so.
   - $\mathsf{Bad}_{\mathrm{node},B}$: For each $t = t_{\mathtt{salt},2,e,j-1}$, find $(t,b,s,z) \in \mathcal{Q}_{\mathrm{node}}$ and $(t,*,y) \in \mathcal{Q}_E$. Let $x_0 \coloneqq \mathsf{B2F}(\psi^{-1}(y\oplus z)\oplus s)$ if $b = 0$, and $x_0 \coloneqq \mathsf{B2F}(\tilde{\psi}^{-1}(y\oplus z)\oplus s)$ otherwise. For all $x_1 \in \mathbb{F}^{n-\lambda/k}$, check whether $x' \coloneqq (x_0,x_1)$ satisfies $\hat{F}_{\mathtt{mseed}_{\mathrm{eq}},\hat{y}}(x') = 0$, and output x' if so.
   - $\mathsf{Bad}_{\mathrm{com},F}$: For each $t = t_{\mathtt{salt},i,e,0}$, find $(t,s,z) \in \mathcal{Q}_{\mathrm{com}}$ and $(t,s',*) \in \mathcal{Q}_E$. Let $x_0 \coloneqq s\oplus s'$. For all $x_1 \in \mathbb{F}^{n-\lambda/k}$, check whether $x' \coloneqq (x_0,x_1)$ satisfies $\hat{F}_{\mathtt{mseed}_{\mathrm{eq}},\hat{y}}(x') = 0$, and output x' if so.
   - $\mathsf{Bad}_{\mathrm{com},B}$: For each $t = t_{\mathtt{salt},i,e,0}$, find $(t,s,z) \in \mathcal{Q}_{\mathrm{com}}$ and $(t,*,y) \in \mathcal{Q}_E$. Let $x_0 \coloneqq \psi^{-1}(y\oplus z)\oplus s$. For all $x_1 \in \mathbb{F}^{n-\lambda/k}$, check whether $x' \coloneqq (x_0,x_1)$ satisfies $\hat{F}_{\mathtt{mseed}_{\mathrm{eq}},\hat{y}}(x') = 0$, and output x' if so.
   - $\mathsf{Bad}_{\mathrm{aux},F}$: For each $t_0 = t_{\mathtt{salt},3,e,0}$ and $t_1 = t_{\mathtt{salt},3,e,1}$, find $(t_0,t_1,s,*,*;\ldots) \in \mathcal{Q}'_{\mathrm{aux}}$ and $(t_b,s',*) \in \mathcal{Q}_E$ for some $b \in \{0,1\}$. Let $x_0 \coloneqq \mathsf{B2F}(s'\oplus s)$. For all $x_1 \in \mathbb{F}^{n-\lambda/k}$, check whether $x' \coloneqq (x_0,x_1)$ satisfies $\hat{F}_{\mathtt{mseed}_{\mathrm{eq}},\hat{y}}(x') = 0$, and output x' if so.
   - $\mathsf{Bad}_{\mathrm{aux},B,b}$ for $b = 0,1$: For each $t_b = t_{\mathtt{salt},3,e,b}$, find $(t_0,t_1,S_d,\Delta_x^{(1)}[e],\alpha'_0[e];i^*,\tilde{x}_1,\dot{x},\tilde{u}'_0,\Gamma') \in \mathcal{Q}'_{\mathrm{aux}}$ and $(t_b,*,\xi_b) \in \mathcal{Q}_E$.

     We then recover x' by solving a system of linear equations. By Lemma 4.1, we obtain

$$(M_\psi \oplus M'_{\mathtt{vk},\dot{x},\Gamma',b})\cdot x_{\mathrm{bin}} = c'_b \oplus \xi_b \oplus \psi(S_d) \in \mathrm{GF}(2)^\lambda.$$

     We solve this system and output a solution $x' \coloneqq \mathsf{B2F}(x_{\mathrm{bin}})$ such that $\hat{F}_{\mathtt{mseed}_{\mathrm{eq}},\hat{y}}(x') = 0$. The running time depends on the rank of the matrix $M_\psi \oplus M'_{\mathtt{vk},\dot{x},\Gamma',b} \in \mathrm{GF}(2)^{\lambda\times kn}$. Under our matrix-rank conjecture (Definition 4.3) below, this matrix has rank $\lambda - O(1)$, yielding running time $2^{kn-\lambda}\cdot\mathsf{poly}(\lambda)$.

Therefore, assuming that our conjecture (Definition 4.3) holds, we obtain

$$\Pr[\Theta_{\mathrm{ideal}} \in \mathsf{Bad}] \le \mathsf{Adv}^{\mathrm{ow}}_{\mathsf{F}_{\mathrm{MQOM}},\mathcal{A}_{\mathrm{ow}}}(\lambda).$$

How to solve $\xi_b = F_b(x,\ldots)$ for $\mathsf{Bad}_{\mathrm{aux},B,b}$. It is easy to see, when we fix x_1, $\xi_b = F_b(x,\ldots)$ yields the system of *quadratic* equations on $x_0 \in \mathbb{F}^{\lambda/k}$ because of $\hat{A}(x-\dot{x})$ in eq.(9). Our key observation is that subtracting $\hat{y}$ with an appropriate factor yields the system of *linear* equations on $x_0 \in \mathbb{F}^{\lambda/k}$.

Lemma 4.1. *Suppose that $\xi_b = F_b(x,\ldots)$ holds, then we can construct a system of linear equations on $x_{\mathrm{bin}} = \mathsf{F2B}(x) \in \mathrm{GF}(2)^{kn}$ defined as*

$$(M_\psi \oplus M'_{\mathtt{vk},\dot{x},\Gamma',b})\cdot x_{\mathrm{bin}} = c'_b \oplus \xi_b \oplus \psi(S_d).$$

To show this lemma, we first consider the quadratic function of x, $\omega_{i^*}^{-1}\cdot(\tilde{u}'_0 - \alpha'_0[e]) + \omega_{i^*}\Gamma'\cdot\hat{A}(x-\dot{x})$ inside of $\mathsf{K2B}(\ldots)$ in eq.(9). Recall that we have $\hat{y} = \hat{A}(x) + \hat{B}\cdot x$. For each $i \in [\hat{m}]$, we have $(\hat{A}(x-\dot{x})-\hat{y})[i] = (x-\dot{x})^\top\cdot\hat{A}_i\cdot(x-\dot{x}) - (x^\top\cdot\hat{A}_i\cdot x + \hat{b}_i^\top\cdot x) = (-\dot{x}^\top\hat{A}_i^\top - \dot{x}^\top\hat{A}_i - \hat{b}_i^\top)\cdot x + \dot{x}^\top\hat{A}_i\dot{x} = M_i\cdot x + \zeta_i$, where $M_i \coloneqq -(\dot{x}^\top\hat{A}_i^\top + \dot{x}^\top\hat{A}_i + \hat{b}_i^\top)$ and $\zeta_i \coloneqq \dot{x}^\top\hat{A}_i\dot{x}$ are publicly computable. Thus, we can write

$$\begin{aligned}\omega_{i^*}^{-1}\cdot(\tilde{u}'_0 - \alpha'_0[e]) + \omega_{i^*}\Gamma'\cdot\hat{A}(x-\dot{x}) &= \omega_{i^*}\Gamma'\cdot\hat{A}(x-\dot{x}) - \omega_{i^*}\Gamma'\cdot\hat{y} \\ &\quad + \omega_{i^*}^{-1}\cdot(\tilde{u}'_0 - \alpha'_0[e]) + \omega_{i^*}\Gamma'\cdot\hat{y} \in \mathbb{K}^{\eta'} \\ &= M_{\mathtt{vk},\dot{x},\Gamma'}\cdot\iota(x) + c, \end{aligned} \tag{10}$$

where we explicitly write the lifting $\iota : \mathbb{F} \hookrightarrow \mathbb{K}$, $M_{\mathsf{vk},\dot{x},\Gamma'} := \omega_{i^*}\Gamma'[M_0;\ldots;M_{\hat{m}-1}] \in \mathbb{K}^{\eta'\times n}$, and $c := \omega_{i^*}\Gamma' \cdot (\zeta_0,\ldots,\zeta_{\hat{m}-1}) + \omega_{i^*}^{-1} \cdot (\tilde{u}'_0 - \alpha'_0[e]) + \omega_{i^*}\Gamma' \cdot \hat{y} \in \mathbb{K}^{\eta'}$.

Letting $x_{\mathrm{bin}} = \mathsf{F2B}(x) \in \mathrm{GF}(2)^{kn}$ and rewriting eq.(9), we have the following equation over GF(2):

$$\begin{aligned}
\xi \oplus \delta_\psi \oplus \tilde{\delta}_\psi &= \mathsf{F2B}(x[\lambda/k : n] - \tilde{x}_1[\lambda/k : n] - \Delta_x^{(1)}[e]) \,\|\, \mathsf{K2B}(M_{\mathsf{vk},\dot{x},\Gamma'} \cdot \iota(x) + c) \\
&= \left(\mathsf{F2B}(x[\lambda/k : n]) - \mathsf{F2B}(\tilde{x}_1[\lambda/k : n] + \Delta_x^{(1)}[e])\right) \,\|\, \left(\mathsf{K2B}(M_{\mathsf{vk},\dot{x},\Gamma'} \cdot \iota(x)) + \mathsf{K2B}(c)\right) \\
&= (M'_{\mathsf{vk},\dot{x},\Gamma',0} \cdot x_{\mathrm{bin}} \oplus c'_0) \,\|\, (M'_{\mathsf{vk},\dot{x},\Gamma',1} \cdot x_{\mathrm{bin}} \oplus c'_1), \qquad (11)
\end{aligned}$$

where we use appropriate $M'_{\mathsf{vk},\dot{x},\Gamma',b} \in \mathrm{GF}(2)^{\lambda\times kn}$ and $c'_b \in \mathrm{GF}(2)^\lambda$ for $b = 0,1$ since $x_{\mathrm{bin}} \mapsto \mathsf{K2B}(M \cdot \iota(\mathsf{B2F}(x_{\mathrm{bin}})))$ is GF(2)-linear.

We then define a linear map $M_\psi : \mathrm{GF}(2)^{kn} \to \mathrm{GF}(2)^\lambda$ as $x_{\mathrm{bin}} \mapsto \psi(x_{\mathrm{bin}}[0 : \lambda])$. Using this definition and the definition δ_ψ and $\tilde{\delta}_\psi$ in eq.(5), we can write the LHS of eq.(11) as

$$\begin{aligned}
\xi \oplus \delta_\psi \oplus \tilde{\delta}_\psi &= (\psi(\delta) \oplus \psi(S_d) \oplus \xi_0) \,\|\, (\psi(\delta) \oplus \psi(S_d) \oplus \xi_1) \\
&= (M_\psi \cdot x_{\mathrm{bin}} \oplus \xi_0 \oplus \psi(S_d)) \,\|\, (M_\psi \cdot x_{\mathrm{bin}} \oplus \xi_1 \oplus \psi(S_d))
\end{aligned}$$

Finally, we have two systems of *linear* equations on x as

$$(M_\psi \oplus M'_{\mathsf{vk},\dot{x},\Gamma',b}) \cdot x_{\mathrm{bin}} = c'_b \oplus \xi_b \oplus \psi(S_d)$$

for $b = 0,1$. To obtain enough information of x from the system, we need to conjecture that the following holds:

Definition 4.3 (Conjecture). *We conjecture that* $\mathrm{rank}_{\mathrm{GF}(2)}(M_\psi - M'_{\mathsf{vk},\dot{x},\Gamma',b}) = \lambda - O(1)$ *for all queries.*

Remark 4.2. We verify our conjecture experimentally as follows: For each 12 GF(2) parameter sets, we generate random 100 verification keys $(A_i, b_i)_{i\in[m]} \leftarrow (\mathrm{GF}(2)^{n\times n} \times \mathrm{GF}(2)^n)^m$. On each key, we generate 100 random $\dot{x} \leftarrow \mathrm{GF}(2)^n$ for 3-round version (and 100 random pair $(\dot{x},\Gamma) \leftarrow \mathrm{GF}(2)^n \times \mathrm{GF}(2)^{\mu\eta'\times m}$ for 5-round version for $\omega_{i^*}\Gamma$), construct $M_b := M_\psi - M'_{\mathsf{vk},\dot{x},\Gamma',b}$ for $b = 0$ and 1, and compute the rank of M_b. We totally compute the matrices M_b $240,000 = 12 \cdot 100 \cdot 100 \cdot 2$ times, and all of them are full rank. We put "$-O(1)$" to allow a constant rank deficiency.

Likelihood ratio on good transcripts: We then show that, for any $\theta \in \mathsf{Good} := \mathcal{T} \setminus \mathsf{Bad}$, $\frac{\Pr[\Theta_{\mathrm{real}}=\theta]}{\Pr[\Theta_{\mathrm{ideal}}=\theta]} \geq 1$ holds.

Let us fix $\theta \in \mathsf{Good}$. In the ideal experiment, we choose $\mathtt{pdecom}$ and α_0 uniformly at random, which are independent of E. Thus, we have

$$\begin{aligned}
\Pr[\Theta_{\mathrm{ideal}} = \theta] &= \Pr_{\mathsf{Ideal}}[(\mathtt{vk},\mathtt{sk}), \mathcal{Q}_{\mathrm{node}}, \mathcal{Q}_{\mathrm{com}}, \mathcal{Q}_{\mathrm{aux}}, \mathcal{Q}_E] = \Pr_{\mathsf{Ideal}}[\mathcal{Q}_{\mathrm{node}}, \mathcal{Q}_{\mathrm{com}}, \mathcal{Q}_{\mathrm{aux}} \mid (\mathtt{vk},\mathtt{sk}), \mathcal{Q}_E] \cdot \Pr_{\mathsf{Ideal}}[(\mathtt{vk},\mathtt{sk}), \mathcal{Q}_E] \\
&= \Pr_{\mathsf{Gen},E}[(\mathtt{vk},\mathtt{sk}), \mathcal{Q}_E] \cdot \left(2^{-d\lambda} \cdot 2^{-2\lambda} \cdot 2^{-2\lambda}\right)^{\tau\cdot Q}.
\end{aligned}$$

On the other hand, since $\mathtt{pdecom}$ and α_0 are bound by ideal ciphers in the real experiment, we have

$$\begin{aligned}
\Pr[\Theta_{\mathrm{real}} = \theta] &= \Pr_{\mathsf{Real}}[(\mathtt{vk},\mathtt{sk}), \mathcal{Q}_{\mathrm{node}}, \mathcal{Q}_{\mathrm{com}}, \mathcal{Q}_{\mathrm{aux}}, \mathcal{Q}_E] = \Pr_{\mathsf{Real}}[\mathcal{Q}_{\mathrm{node}}, \mathcal{Q}_{\mathrm{com}}, \mathcal{Q}_{\mathrm{aux}} \mid (\mathtt{vk},\mathtt{sk}), \mathcal{Q}_E] \cdot \Pr_{\mathsf{Ideal}}[(\mathtt{vk},\mathtt{sk}), \mathcal{Q}_E] \\
&\geq \Pr_{\mathsf{Gen},E}[(\mathtt{vk},\mathtt{sk}), \mathcal{Q}_E] \cdot \left(2^{-d\lambda} \cdot 2^{-2\lambda} \cdot 2^{-2\lambda}\right)^{\tau\cdot Q} = \Pr[\Theta_{\mathrm{ideal}} = \theta].
\end{aligned}$$

Thus, for any $\theta \in \mathsf{Good}$, we obtain $\frac{\Pr[\Theta_{\mathrm{real}}=\theta]}{\Pr[\Theta_{\mathrm{ideal}}=\theta]} \geq 1$. This implies $\epsilon_{\mathrm{good}} = 0$.

Conclusion via the H-coefficient technique: Applying the H-coefficient technique (Theorem 4.1), we obtain that there exists a machine $\mathcal{A}_{\mathrm{ow}}$ satisfying

$$\mathsf{Adv}^{\mathrm{avc}}_{\mathsf{MQOM}_\Omega,\mathcal{A}}(\lambda) \leq \epsilon_{\mathrm{good}} + \epsilon_{\mathrm{bad}} \leq 0 + \mathsf{Adv}^{\mathrm{ow}}_{\mathsf{F}_{\mathsf{MQOM}},\mathcal{A}_{\mathrm{ow}}}(\lambda)$$

as we wanted, where $\mathcal{A}_{\mathrm{ow}}$'s running time is that of $\mathcal{A}$ plus $2^{kn-\lambda} \cdot \mathsf{poly}(Q',\lambda)$. □

5 EUF-CMA Security of MQOM

Let $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ be the variant of MQOM that includes the signing salt $\mathtt{salt}$ in the input to Hash_4 when deriving the Fiat–Shamir challenge and uses an evaluation domain $\Omega \subseteq \mathbb{K} \setminus \{0\}$. By using AVC's security, we can show the EUF-CMA security of $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ in the ROM and QROM as follows:

Theorem 5.1 (EUF-NMA to EUF-CMA of $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ in the ROM and QROM). *Let Hash_4 be a random oracle. For a QPT adversary $\mathcal{A}$ that queries the oracles X at most Q_X times, there exist QPT adversaries $\mathcal{A}_{\mathrm{nma}}$, $\mathcal{A}_{\mathrm{hash}}$, $\mathcal{A}_{\mathrm{prg}}$, and $\mathcal{A}_{\mathrm{avc}}$ such that the following bounds hold. We have*

$$\begin{aligned}\mathsf{Adv}^{\text{euf-cma}}_{\mathsf{MQOM}_{\mathtt{salt},\Omega},\mathcal{A}}(\lambda) \le{}& \mathsf{Adv}^{\text{euf-nma}}_{\mathsf{MQOM}_{\mathtt{salt},\Omega},\mathcal{A}_{\mathrm{nma}}}(\lambda) + \mathsf{Adv}^{\mathrm{cr}}_{\mathsf{Hash}_2,\mathcal{A}_{\mathrm{hash}}}(\lambda) + Q_{\mathsf{Sign}} \cdot \mathsf{Adv}^{\mathrm{prg}}_{\mathsf{PRG},\mathcal{A}_{\mathrm{prg}}}(\lambda) \\ &+ \mathsf{Adv}^{\mathrm{avc}}_{\mathsf{MQOM}_{\Omega},\mathcal{A}_{\mathrm{avc}}}(\lambda) + Q_{\mathsf{Sign}}^2/2^{\lambda-12} + (\epsilon_{\mathrm{ROM}} \text{ or } \epsilon_{\mathrm{QROM}}),\end{aligned}$$

where

$$\epsilon_{\mathrm{ROM}} := Q_{\mathsf{Sign}}(Q_{\mathsf{Sign}} + Q_{\mathsf{Hash}_4})/2^{\lambda} \text{ and } \epsilon_{\mathrm{QROM}} := \frac{3}{2} Q_{\mathsf{Sign}} \sqrt{(Q_{\mathsf{Sign}} + Q_{\mathsf{Hash}_4})/2^{\lambda}}.$$

The running times of $\mathcal{A}_{\mathrm{nma}}$, $\mathcal{A}_{\mathrm{hash}}$, $\mathcal{A}_{\mathrm{prg}}$, and $\mathcal{A}_{\mathrm{avc}}$ are approximately that of $\mathcal{A}$.

Remark 5.1. The tweak-collision term $Q_{\mathsf{Sign}}^2/2^{\lambda-12}$ is not inherent to our proof, but is introduced to simplify the AVC simulation. For very large Q_{Sign}, improving this bound would require handling multi-collisions of salts or using longer salts.

Remark 5.2. If we adopt a heuristic assumption that $\mathtt{com}_1$ has sufficiently high min-entropy, then we can show the same result for MQOM_{Ω}.

As a direct corollary, we obtain the following.

Corollary 5.1 (EUF-NMA to EUF-CMA of $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ in the ROM+ICM). *Let Hash_4 be a random oracle. Let $\mathsf{H}_{\mathsf{avc}}(s,t) := E(t,s) \oplus \psi(s)$ with an ideal cipher E. Let $\mathbb{F} = \mathrm{GF}(2)$. Suppose that the conjecture (Definition 4.3) holds.*

For a PPT adversary $\mathcal{A}$ that queries the oracles X at most Q_X times, there exist PPT adversaries $\mathcal{A}_{\mathrm{hash}}$, $\mathcal{A}_{\mathrm{prg}}$, and $\mathcal{A}_{\mathrm{nma}}$, and an adversary $\mathcal{A}_{\mathrm{ow}}$ such that

$$\begin{aligned}\mathsf{Adv}^{\text{euf-cma}}_{\mathsf{MQOM}_{\mathtt{salt},\Omega},\mathcal{A}}(\lambda) \le{}& \mathsf{Adv}^{\text{euf-nma}}_{\mathsf{MQOM}_{\mathtt{salt},\Omega},\mathcal{A}_{\mathrm{nma}}}(\lambda) + \mathsf{Adv}^{\mathrm{cr}}_{\mathsf{Hash}_2,\mathcal{A}_{\mathrm{hash}}}(\lambda) + Q_{\mathsf{Sign}} \cdot \mathsf{Adv}^{\mathrm{prg}}_{\mathsf{PRG},\mathcal{A}_{\mathrm{prg}}}(\lambda) \\ &+ \mathsf{Adv}^{\mathrm{ow}}_{\mathsf{F}_{\mathsf{MQOM}},\mathcal{A}_{\mathrm{ow}}}(\lambda) + Q_{\mathsf{Sign}}^2/2^{\lambda-12} + Q_{\mathsf{Sign}}(Q_{\mathsf{Sign}} + Q_{\mathsf{Hash}_4})/2^{\lambda}.\end{aligned}$$

The running times of $\mathcal{A}_{\mathrm{nma}}$, $\mathcal{A}_{\mathrm{hash}}$, and $\mathcal{A}_{\mathrm{prg}}$ are that of $\mathcal{A}$ plus $\mathsf{poly}(Q,\lambda)$ with $Q = Q_{\mathsf{Sign}} + Q_{\mathsf{Hash}_4}$. The running time of $\mathcal{A}_{\mathrm{ow}}$ is that of $\mathcal{A}$ plus $2^{kn-\lambda} \cdot \mathsf{poly}(Q',\lambda)$ with $Q' = Q_E + Q_{\mathsf{Sign}} \cdot \tau \cdot (5N - d - 4)$.

Proof. The corollary follows from Theorem 5.1 and Theorem 4.2. □

Remark 5.3. Since the reduction using the H-coefficient technique requires manipulating the adversary's random tape, we can consider only PPT adversaries.

Remark 5.4. The main security loss is "$+2^{kn-\lambda} \cdot \mathsf{poly}(Q',\lambda)$" in $\mathcal{A}_{\mathrm{ow}}$'s running time in Corollary 5.1, which can be large since $Q' = Q_E + O(Q_{\mathsf{Sign}}\tau N)$.

Combining Theorem 5.1 and Theorem 3.1, we obtain the following corollary for $\mathsf{MQOM}_{\mathtt{salt},\Omega}$.

Corollary 5.2 (EUF-NMA to EUF-CMA of $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ in the (Q)ROM). *Let $\mathsf{H}_{\mathsf{avc}}$ and Hash_4 be random oracles. For a QPT adversary $\mathcal{A}$ that queries the oracles X at most Q_X times, there exist QPT adversaries $\mathcal{A}_{\mathrm{nma}}$, $\mathcal{A}_{\mathrm{hash}}$, $\mathcal{A}_{\mathrm{prg}}$, and $\mathcal{A}_{\mathrm{pgow}}$ such that the following bounds hold. We have*

$$\begin{aligned}\mathsf{Adv}^{\text{euf-cma}}_{\mathsf{MQOM}_{\mathtt{salt},\Omega},\mathcal{A}}(\lambda) \le{}& \mathsf{Adv}^{\text{euf-nma}}_{\mathsf{MQOM}_{\mathtt{salt},\Omega},\mathcal{A}_{\mathrm{nma}}}(\lambda) + \mathsf{Adv}^{\mathrm{cr}}_{\mathsf{Hash}_2,\mathcal{A}_{\mathrm{hash}}}(\lambda) + Q_{\mathsf{Sign}} \cdot \mathsf{Adv}^{\mathrm{prg}}_{\mathsf{PRG},\mathcal{A}_{\mathrm{prg}}}(\lambda) + Q_{\mathsf{Sign}}^2/2^{\lambda-12} \\ &+ 2\Pr[\delta = 0^{\lambda}] + 2Q_{\mathsf{Sign}}\tau \cdot (2N - d - 3)/2^{\lambda} \\ &+ \left(\mathsf{Adv}^{\mathrm{pgow}}_{\mathsf{F}_{\mathsf{MQOM}},\mathcal{A}_{\mathrm{pgow}}}(\lambda) + \epsilon_{\mathrm{ROM}} \text{ or } 2\sqrt{Q_{\mathsf{H}_{\mathsf{avc}}} \mathsf{Adv}^{\mathrm{pgow}}_{\mathsf{F}_{\mathsf{MQOM}},\mathcal{A}_{\mathrm{pgow}}}(\lambda)} + \epsilon_{\mathrm{QROM}}\right),\end{aligned}$$

where ϵ_{ROM} and ϵ_{QROM} are taken from Theorem 5.1. The running times of $\mathcal{A}_{\mathrm{nma}}$, $\mathcal{A}_{\mathrm{hash}}$, $\mathcal{A}_{\mathrm{prg}}$, and $\mathcal{A}_{\mathrm{pgow}}$ are approximately that of $\mathcal{A}$. The number of oracle queries of $\mathcal{A}_{\mathrm{nma}}$ is the same as that of $\mathcal{A}$. Moreover, $\mathcal{A}_{\mathrm{pgow}}$ makes at most $2Q_{\mathsf{H}_{\mathsf{avc}}}$ queries to its guessing oracle.

5.1 On the security loss

Loss from AVC simulation security in the ICM: By Theorem 4.2, we have

$$\mathsf{Adv}^{\mathrm{avc}}_{\mathsf{MQOM}_\Omega,\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\mathrm{ow}}_{\mathsf{F}_{\mathsf{MQOM}},\mathcal{A}_{\mathrm{ow}}}(\lambda) \;\text{ and }\; \mathsf{Time}(\mathcal{A}_{\mathrm{ow}}) = \mathsf{Time}(\mathcal{A}) + 2^{kn-\lambda}\cdot\mathsf{poly}(Q_E,\lambda),$$

where we assume that $Q_E \gg Q_{\mathsf{Sign}}\cdot\tau\cdot(5N-d-4)$. Recall that $\mathcal{A}_{\mathrm{ow}}$ works as follows: (1) it runs $\mathcal{A}$ while simulating the signing oracle, Hash_4, and E, maintaining the query lists for Hash_4 and E in sorted order; and (2) it finds $(t_0,t_1,S_d,\ldots;\ldots)\in\mathcal{Q}'_{\mathrm{aux}}$ and $(t_b,*,\xi_b)\in\mathcal{Q}_E$ and recovers x' by exhaustive search over the remaining $n-\lambda/k$ undetermined coordinates. More precisely, the term $2^{kn-\lambda}\cdot\mathsf{poly}(Q_E,\lambda)$ can be written as $2^{kn-\lambda}\cdot Q_E\cdot\mathsf{poly}(\lambda)$. Indeed, the sorted query lists allow $\mathcal{A}_{\mathrm{ow}}$ to efficiently identify the relevant matching entries in $\mathcal{Q}'_{\mathrm{aux}}$ and $\mathcal{Q}_E$. For each of the at most Q_E relevant pairs, it solves a system of linear equations while enumerating $2^{kn-\lambda}$ candidates.

Suppose, for $\kappa>0$, that

$$\frac{\mathsf{Adv}^{\mathrm{ow}}_{\mathsf{F}_{\mathsf{MQOM}},\mathcal{A}_{\mathrm{ow}}}(\lambda)}{\mathsf{Time}(\mathcal{A}_{\mathrm{ow}})} \le 2^{-\kappa}.$$

Then,

$$\frac{\mathsf{Adv}^{\mathrm{avc}}_{\mathsf{MQOM}_\Omega,\mathcal{A}}(\lambda)}{\mathsf{Time}(\mathcal{A})} \le \frac{\mathsf{Adv}^{\mathrm{ow}}_{\mathsf{F}_{\mathsf{MQOM}},\mathcal{A}_{\mathrm{ow}}}(\lambda)}{\mathsf{Time}(\mathcal{A})} \le \frac{2^{-\kappa}\cdot\mathsf{Time}(\mathcal{A}_{\mathrm{ow}})}{\mathsf{Time}(\mathcal{A})} = 2^{-\kappa}\cdot\left(\frac{\mathsf{Time}(\mathcal{A})+2^{kn-\lambda}Q_E\mathsf{poly}(\lambda)}{\mathsf{Time}(\mathcal{A})}\right). \tag{12}$$

In the concrete setting, $\mathcal{A}$ can implement circuits for both E^+ and E^- and may therefore perform a number of such evaluations proportional to its running time. Accordingly, it is natural to consider the regime $Q_E=\Theta(\mathsf{Time}(\mathcal{A}))$.

Thus, ignoring relatively small factor $\mathsf{poly}(\lambda)$, the security loss $\log_2\left(\frac{\mathsf{Time}(\mathcal{A})+2^{kn-\lambda}Q_E\mathsf{poly}(\lambda)}{\mathsf{Time}(\mathcal{A})}\right)$ in Equation 12 is approximately $kn-\lambda$.

Other losses:

- The term $Q_{\mathsf{Sign}}\cdot\mathsf{Adv}^{\mathrm{prg}}_{\mathsf{PRG},\mathcal{A}_{\mathrm{prg}}}(\lambda)$ arises from replacing $(\mathtt{rseed}[0],\ldots,\mathtt{rseed}[\tau-1]) := \mathsf{PRG}(\mathtt{mseed})$ with independently sampled random values.
- The term $Q^2_{\mathsf{Sign}}/2^{\lambda-12}$ arises from excluding collisions among the salt prefixes used to derive tweaks in Corollary 5.1 and Corollary 5.2. We impose this collision-free condition for two reasons:
  1. It considerably simplifies the proof of AVC simulatability by allowing each ideal-cipher or random-oracle query to be associated with at most one signing transcript. Allowing multiple signing transcripts to share the same salt prefix would require a more involved analysis in both the ICM and the (Q)ROM.
  2. This term is relatively small compared with the adaptive-reprogramming loss below when $Q_{\mathsf{Sign}}\ll Q_{\mathsf{Hash}_4}$. For example, when $Q_{\mathsf{Sign}}=2^{64}$ and $Q_{\mathsf{Hash}_4}=2^{96}$, we have $\frac{Q^2_{\mathsf{Sign}}}{2^{\lambda-12}} < \frac{Q_{\mathsf{Sign}}(Q_{\mathsf{Sign}}+Q_{\mathsf{Hash}_4})}{2^\lambda}$.
- The term $\epsilon_{\mathrm{ROM}} = Q_{\mathsf{Sign}}(Q_{\mathsf{Sign}}+Q_{\mathsf{Hash}_4})/2^\lambda$ or $\epsilon_{\mathrm{QROM}} = \frac{3}{2}Q_{\mathsf{Sign}}\sqrt{(Q_{\mathsf{Sign}}+Q_{\mathsf{Hash}_4})/2^\lambda}$ arises from adaptive reprogramming. When $Q_{\mathsf{Sign}}\ll Q_{\mathsf{Hash}_4}$, the dominant contribution is approximately $Q_{\mathsf{Sign}}Q_{\mathsf{Hash}_4}/2^\lambda$ in the ROM and $Q_{\mathsf{Sign}}\sqrt{Q_{\mathsf{Hash}_4}/2^\lambda}$ in the QROM.

For sufficiently large numbers of queries, these bounds may cease to provide a meaningful concrete-security guarantee. In particular, when $Q_{\mathsf{Sign}}Q_{\mathsf{Hash}_4}\approx 2^\lambda$, the ROM adaptive-reprogramming bound becomes trivial. For example, when $\lambda=128$ and $Q_{\mathsf{Sign}}=2^{64}$, the maximal number of signing queries considered in the NIST call, some of the above bounds already become trivial.

Delgado [Del26] further observed that a collision in $\mathtt{mseed}$ can be detected through collisions of $(\mathtt{rseed}[0],\ldots,\mathtt{rseed}[\tau-1])$. Moreover, a collision in $\mathtt{rseed}[e]$ allows recovery of δ with sufficiently high probability, and hence recovery of the entire secret key x. This observation shows that the term $Q_{\mathsf{Sign}}\cdot\mathsf{Adv}^{\mathrm{prg}}_{\mathsf{PRG},\mathcal{A}_{\mathrm{prg}}}(\lambda)$ cannot in general be expected to remain nontrivial when $\lambda=128$ and $Q_{\mathsf{Sign}}=2^{64}$.

One possible way to mitigate the tweak-collision and adaptive-reprogramming losses is to choose a sufficiently large salt length, for example 2λ or 4λ bits. This modification is straightforward for a random-oracle-based variant, at the cost of increasing the signature size, whereas the current block-cipher-based instantiation would require a redesign of its symmetric component.

G_0–G_3

$Q := \emptyset$
$Q_{\mathrm{salt}} := \emptyset$ // $\mathsf{G}_{0.2}$–G_3
$(\mathtt{vk}, \mathtt{sk}) \leftarrow \mathsf{MQOM.Gen}(1^\lambda)$
$O := \{\mathsf{H}_{\mathsf{avc}}, \mathsf{Hash}_2, \mathsf{Hash}_4\}$
$(\mathtt{msg}^*, \sigma^*) \leftarrow \mathcal{A}^{\textsc{Sign},|O\rangle}(\mathtt{vk})$
if $\mathtt{msg}^* \in Q$ **then return** 0
if $\exists q : \mathsf{Hash}_2(\mathtt{msg}^*) = \mathsf{Hash}_2(\mathtt{msg}^{(q)})$ **then abort** // $\mathsf{G}_{0.1}$–G_3
return $\mathsf{MQOM}_{\mathtt{salt},\Omega}.\mathsf{Vrfy}(\mathtt{vk}, \mathtt{msg}^*, \sigma^*)$

Fig. 9. Games for proving EUF-NMA ⇒ EUF-CMA of $\mathsf{MQOM}_{\mathtt{salt},\Omega}$. SIGN is defined in Figure 10.

5.2 Proof of Theorem 5.1

Proof. We use the sequence of games. Let W_i be the winning event in G_i. The game descriptions and the signing oracle are given in Figure 9 and Figure 10.

- Game 0: This is the original EUF-CMA security game for $\mathsf{MQOM}_{\mathtt{salt},\Omega}$. Thus, we have $\Pr[\mathsf{G}_0 : W_0] = \mathsf{Adv}^{\text{euf-cma}}_{\mathsf{MQOM}_{\mathtt{salt},\Omega},\mathcal{A}}(\lambda)$.
- Game 0.1: In this game, we introduce an abort condition tied to the forgery message $\mathtt{msg}^*$. The game aborts if there exists a signing-query message $\mathtt{msg}^{(q)}$ such that $\mathsf{Hash}_2(\mathtt{msg}^*) = \mathsf{Hash}_2(\mathtt{msg}^{(q)})$. Thus, the abort probability is bounded by the probability of finding a collision in Hash_2. It is easy to construct an adversary $\mathcal{A}_{\mathrm{hash}}$ such that
$$|\Pr[\mathsf{G}_0 : W_0] - \Pr[\mathsf{G}_{0.1} : W_{0.1}]| \le \mathsf{Adv}^{\mathrm{cr}}_{\mathsf{Hash}_2,\mathcal{A}_{\mathrm{hash}}}(\lambda).$$
- Game 0.2: The game aborts if there exist two distinct signing queries whose salts have the same first $\lambda - 12$ bits. This is the relevant collision event for TweakSalt, because the last 12 bits of the tweaked salt are reserved for domain separation.[9] By the birthday bound, in both the ROM and the QROM,
$$|\Pr[\mathsf{G}_{0.1} : W_{0.1}] - \Pr[\mathsf{G}_{0.2} : W_{0.2}]| \le Q_{\mathsf{Sign}}^2 / 2^{\lambda-12}.$$
- Game 0.3: In this game, we replace the values $(\mathtt{rseed}^{(q)}[0], \ldots, \mathtt{rseed}^{(q)}[\tau-1])$ output by $\mathsf{PRG}(\mathtt{mseed}^{(q)})$ with uniformly random strings. Using a standard hybrid argument over the Q_{Sign} signing queries, this step reduces to the pseudorandomness of PRG. Hence, in both the ROM and the QROM, there exists an adversary $\mathcal{A}_{\mathrm{prg}}$ such that
$$|\Pr[\mathsf{G}_{0.2} : W_{0.2}] - \Pr[\mathsf{G}_{0.3} : W_{0.3}]| \le Q_{\mathsf{Sign}} \cdot \mathsf{Adv}^{\mathrm{prg}}_{\mathsf{PRG},\mathcal{A}_{\mathrm{prg}}}(\lambda).$$
- Game 1: In this game, the signing oracle begins by choosing h_2 uniformly at random and reprogramming it as the output of Hash_4.

  ROM. The adversary notices this reprogramming only if it queries the programmed point before the signing oracle performs the programming. Since the programmed input contains the fresh salt $\mathtt{salt}^{(q)}$, and since the salt-prefix collision event has already been excluded in Game 0.2, we have
$$|\Pr[\mathsf{G}_{0.3} : W_{0.3}] - \Pr[\mathsf{G}_1 : W_1]| \le Q_{\mathsf{Sign}}(Q_{\mathsf{Sign}} + Q_{\mathsf{Hash}_4})/2^\lambda.$$
  QROM. We apply the adaptive reprogramming lemma Lemma 2.1 with $x_1 := (\mathtt{vk}, \mathtt{salt}^{(q)}, \mathtt{com}_1, \mathtt{com}_2)$ and $x_2 := h_{\mathtt{msg}}$. The distribution D_q samples the fresh salt $\mathtt{salt}^{(q)}$, generates the signing transcript, and outputs x_1 together with the signature as side information side. Since $\mathtt{salt}^{(q)}$ is freshly sampled inside the signing oracle, for every reprogramming query q we have $p^{(q)}_{\max} \le 2^{-\lambda}$.

  Applying the adaptive reprogramming lemma gives
$$|\Pr[\mathsf{G}_{0.3} : W_{0.3}] - \Pr[\mathsf{G}_1 : W_1]| \le \frac{3}{2} Q_{\mathsf{Sign}} \sqrt{(Q_{\mathsf{Sign}} + Q_{\mathsf{Hash}_4})/2^\lambda}.$$

[9] We have $k + 4e + 256j < 2^{12}$ for all parameter sets.

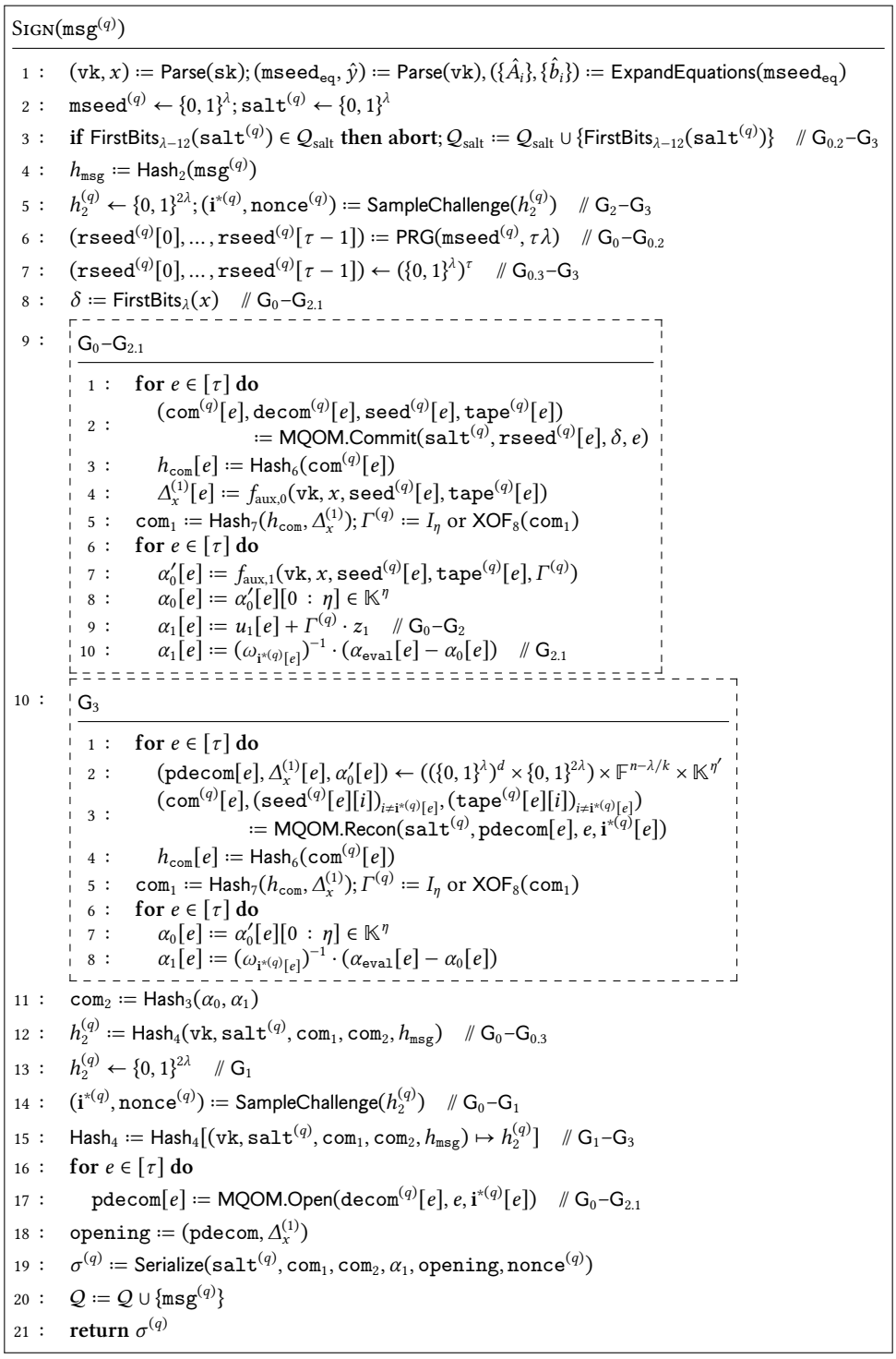


Fig. 10. Signing oracle of Figure 9. We omit the detail of the computation of $u_1[e]$, z_1, and $\alpha_{\mathtt{eval}}[e]$. Note that $\alpha_{\mathtt{eval}}[e]$ is computable from $\mathtt{vk}$, $\Gamma^{(q)}$, $\mathbf{i}^{*(q)}$, $\Delta_x^{(1)}$, $(\mathtt{seed}[e][i])_{i \neq \mathbf{i}^{*(q)}[e]}$, and $(\mathtt{tape}[e][i])_{i \neq \mathbf{i}^{*(q)}[e]}$ as in Figure 11.

– Game 2: In this game, the signing oracle computes $(\mathbf{i}^{*(q)}, \mathtt{nonce}^{(q)})$ at the beginning of the signing oracle. Since $h_2^{(q)}$ is uniformly random and $(\mathbf{i}^{*(q)}, \mathtt{nonce}^{(q)})$ is deterministically derived from $h_2^{(q)}$, this is a purely conceptual change. Thus, in both the ROM and the QROM, $\Pr[\mathsf{G}_1 : W_1] = \Pr[\mathsf{G}_2 : W_2]$.
– Game 2.1: In this game, the signing oracle computes $\alpha_1[e]$ from the opened value while keeping $\alpha_0[e]$ as the honestly generated constant coefficient. After computing $\alpha_{\mathtt{eval}}[e] := u_{\mathtt{eval}}[e] + \Gamma \cdot z_{\mathtt{eval}}[e]$, the signing oracle sets $\alpha_1[e] := (\omega_{\mathbf{i}^{*(q)}[e]})^{-1} \cdot \big(\alpha_{\mathtt{eval}}[e] - \alpha_0[e]\big)$. This only rewrites the line relation $P_\alpha^{(e)}(\omega_{\mathbf{i}^{*(q)}[e]}) = \alpha_{\mathtt{eval}}[e]$, and therefore does not change the distribution. Consequently, in both the ROM and the QROM, we have

$$\Pr[\mathsf{G}_2 : W_2] = \Pr[\mathsf{G}_{2.1} : W_{2.1}].$$

– Game 3: In this game, for each signing query q and repetition e, we choose the real opening transcript and the signature-exposed auxiliary values uniformly at random. The commitment and the opened shares/tapes are then reconstructed from the simulated opening transcript by the standard reconstruction and parsing procedures, as described in the signing oracle in Figure 10. The rest of the signing computation is unchanged.

  Since we already eliminate the collision of tweaks in Game 0.2, it is obvious that there exists an adversary $\mathcal{A}_{\mathrm{avc}}$ satisfying

$$|\Pr[\mathsf{G}_{2.1} : W_{2.1}] - \Pr[\mathsf{G}_3 : W_3]| \le \mathsf{Adv}^{\mathrm{avc}}_{\mathsf{MQOM}_\Omega, \mathcal{A}_{\mathrm{avc}}}(\lambda).$$

It remains to reduce G_3 to EUF-NMA. In G_3, the signing oracle is secret-key-free. Hence, an EUF-NMA adversary $\mathcal{A}_{\mathrm{nma}}$ can simulate the signing oracle internally. The adversary $\mathcal{A}_{\mathrm{nma}}$ gives $\mathtt{vk}$ to $\mathcal{A}$, answers signing queries using the G_3 signing oracle, and outputs the final forgery $(\mathtt{msg}^*, \sigma^*)$ produced by $\mathcal{A}$.

The programming of Hash_4 performed during simulated signing does not affect the verification of the final forgery. Indeed, the verification of $(\mathtt{msg}^*, \sigma^*)$ queries Hash_4 only on the input determined by $(\mathtt{vk}, \mathtt{salt}^*, \mathtt{com}_1^*, \mathtt{com}_2^*, \mathsf{Hash}_2(\mathtt{msg}^*))$, while any previously signed message is rejected, and Game 0.1 excludes the case $\mathsf{Hash}_2(\mathtt{msg}^*) = \mathsf{Hash}_2(\mathtt{msg}^{(q)})$ for a signing query $\mathtt{msg}^{(q)}$. Therefore, in both the ROM and the QROM,

$$\Pr[\mathsf{G}_3 : W_3] \le \mathsf{Adv}^{\text{euf-nma}}_{\mathsf{MQOM}_{\mathtt{salt},\Omega}, \mathcal{A}_{\mathrm{nma}}}(\lambda).$$

Combining the corresponding game-hop bounds gives the claimed ROM and QROM bounds. This completes the proof. □

Acknowledgment

Parts of the manuscript were edited with the assistance of generative AI tools (ChatGPT, OpenAI; model: GPT-5.5 Thinking/GPT-5.6 Think Deeper/GPT-6 Astra) for language polishing (e.g., phrasing, grammar, and clarity) and another tool (Gemini 3, Google) for making basic figures. All AI-assisted suggestions were reviewed and revised by the authors, who take full responsibility for the correctness and integrity of the manuscript.

References

AAB+24. Gor Adj, Nicolas Aragon, Stefano Barbero, Magali Bardet, Emmanuele Bellini, Loïc Bidoux, Jesús-Javier Chi-Domínguez, Victor Dyseryn, Andre Esser, Thibauld Feneuil, Philippe Gaborit, Romaric Neveu, Matthieu Rivain, Luis Rivera-Zamarripa, Carlo Sanna, Jean-Pierre Tillich, Javier Verbel, and Floyd Zweydinger. Mirath (merger of MIRA/MiRitH). Technical report, National Institute of Standards and Technology, 2024. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures. 11

ABB+24a. Najwa Aaraj, Slim Bettaieb, Loïc Bidoux, Alessandro Budroni, Victor Dyseryn, Andre Esser, Thibauld Feneuil, Philippe Gaborit, Mukul Kulkarni, Victor Mateu, Marco Palumbi, Lucas Perin, Matthieu Rivain, Jean-Pierre Tillich, and Keita Xagawa. PERK. Technical report, National Institute of Standards and Technology, 2024. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures. 11

ABB+24b. Carlos Aguilar Melchor, Slim Bettaieb, Loïc Bidoux, Thibauld Feneuil, Philippe Gaborit, Nicolas Gama, Shay Gueron, James Howe, Andreas Hülsing, David Joseph, Antoine Joux, Mukul Kulkarni, Edoardo Persichetti, Tovohery H. Randrianarisoa, Matthieu Rivain, and Dongze Yue. SDitH — Syndrome Decoding in the Head. Technical report, National Institute of Standards and Technology, 2024. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures. 2

ABB+24c. Nicolas Aragon, Magali Bardet, Loïc Bidoux, Jesús-Javier Chi-Domínguez, Victor Dyseryn, Thibauld Feneuil, Philippe Gaborit, Antoine Joux, Romaric Neveu, Matthieu Rivain, Jean-Pierre Tillich, and Adrien Vinçotte. RYDE. Technical report, National Institute of Standards and Technology, 2024. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures. 11

ABC+26. Gorjan Alagic, Maxime Bros, Pierre Ciadoux, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Hamilton Silberg, Daniel Smith-Tone, and Noah Waller. Status report on the second round of the additional digital signature schemes for the NIST post-quantum cryptography standardization process. Technical Report NIST IR 8610, National Institute of Standards and Technology, 2026. 2

AHJ+23. Carlos Aguilar Melchor, Andreas Hülsing, David Joseph, Christian Majenz, Eyal Ronen, and Dongze Yue. SDitH in the QROM. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part VII*, volume 14444 of *LNCS*, pages 317–350. Springer, Singapore, December 2023. 6

AHU19. Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 269–295. Springer, Cham, August 2019. 5, 7, 36

BBB+24. Carsten Baum, Lennart Braun, Ward Beullens, Cyprien Delpech de Saint Guilhem, Michael Klooß, Christian Majenz, Shibam Mukherjee, Emmanuela Orsini, Sebastian Ramacher, Christian Rechberger, Lawrence Roy, and Peter Scholl. FAEST. Technical report, National Institute of Standards and Technology, 2024. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures. 2

BBB+25. Carsten Baum, Ward Beullens, Lennart Braun, Cyprien Delpech de Saint Guilhem, Michael Klooß, Christian Majenz, Shibam Mukherjee, Emmanuela Orsini, Sebastian Ramacher, Christian Rechberger, Lawrence Roy, and Peter Scholl. Shorter, tighter, FAESTer: Optimizations and improved (QROM) analysis for VOLE-in-the-head signatures. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VI*, volume 16005 of *LNCS*, pages 124–156. Springer, Cham, August 2025. 6

BBD+23. Carsten Baum, Lennart Braun, Cyprien Delpech de Saint Guilhem, Michael Klooß, Emmanuela Orsini, Lawrence Roy, and Peter Scholl. Publicly verifiable zero-knowledge and post-quantum signatures from VOLE-in-the-head. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 581–615. Springer, Cham, August 2023. 2

BBFR24. Ryad Benadjila, Charles Bouillaguet, Thibauld Feneuil, and Matthieu Rivain. MQOM — MQ on my Mind. Technical report, National Institute of Standards and Technology, 2024. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures. 2, 6, 40

BBFR25. Ryad Benadjila, Charles Bouillaguet, Thibauld Feneuil, and Matthieu Rivain. The mqom specification version 2.1 (released on 2025/09/22). Website, September 2025. 2, 11, 16, 17, 31

BCC+24. Dung Bui, Eliana Carozza, Geoffroy Couteau, Dahmun Goudarzi, and Antoine Joux. Faster signatures from MPC-in-the-head. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part I*, volume 15484 of *LNCS*, pages 396–428. Springer, Singapore, December 2024. 6

BCD25. Dung Bui, Kelong Cong, and Cyprien Delpech de Saint Guilhem. Faster VOLEitH signatures from all-but-one vector commitment and half-tree. In Willy Susilo and Josef Pieprzyk, editors, *ACISP 25, Part I*, volume 15658 of *LNCS*, pages 205–223. Springer, Singapore, July 2025. 2, 6, 35, 36

BHH+19. Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 61–90. Springer, Cham, December 2019. 36, 39

CDG+17. Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 1825–1842. ACM Press, October / November 2017. 6

CDMP05. Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård revisited: How to construct a hash function. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 430–448. Springer, Berlin, Heidelberg, August 2005. 6

CFHL21. Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 598–629. Springer, Cham, October 2021. 6

CKKZ12. Seung Geol Choi, Jonathan Katz, Ranjit Kumaresan, and Hong-Sheng Zhou. On the security of the "free-XOR" technique. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 39–53. Springer, Berlin, Heidelberg, March 2012. 3

CLY+24. Hongrui Cui, Hanlin Liu, Di Yan, Kang Yang, Yu Yu, and Kaiyi Zhang. ReSolveD: Shorter signatures from regular syndrome decoding and VOLE-in-the-head. In Qiang Tang and Vanessa Teague, editors, *PKC 2024, Part I*, volume 14601 of *LNCS*, pages 229–258. Springer, Cham, April 2024. 2, 6, 35, 36

CS14. Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 327–350. Springer, Berlin, Heidelberg, May 2014. 3, 4, 16

DDOS19. Cyprien Delpech de Saint Guilhem, Lauren De Meyer, Emmanuela Orsini, and Nigel P. Smart. BBQ: Using AES in picnic signatures. In Kenneth G. Paterson and Douglas Stebila, editors, *SAC 2019*, volume 11959 of *LNCS*, pages 669–692. Springer, Cham, August 2019. 6

Del26. José Luis Delgado. Signing-key recovery from unsalted root expansion and salt-binding repair for MQOM v2. Cryptology ePrint Archive, Paper 2026/1542, 2026. 24

DFMS22a. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Efficient NIZKs and signatures from commit-and-open protocols in the QROM. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 729–757. Springer, Cham, August 2022. 6

DFMS22b. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Online-extractability in the quantum random-oracle model. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 677–706. Springer, Cham, May / June 2022. 6

Din24. Itai Dinur. Tight indistinguishability bounds for the XOR of independent random permutations by Fourier analysis. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part I*, volume 14651 of *LNCS*, pages 33–62. Springer, Cham, May 2024. 6

FR23. Thibauld Feneuil and Matthieu Rivain. Threshold linear secret sharing to the rescue of MPC-in-the-head. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part I*, volume 14438 of *LNCS*, pages 441–473. Springer, Singapore, December 2023. 2, 11

FR25. Thibauld Feneuil and Matthieu Rivain. Threshold computation in the head: Improved framework for post-quantum signatures and zero-knowledge arguments. *Journal of Cryptology*, 38(3):28, July 2025. 2, 11

FR26. Thibauld Feneuil and Matthieu Rivain. On the security of MPC-in-the-head signatures with correlated GGM trees. Cryptology ePrint Archive, Paper 2025/1411, 2026. To appear ASIACRYPT 2026. 2, 3, 5, 6, 12, 13, 14, 17, 33, 34

FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Berlin, Heidelberg, August 1987. 2, 6

GBJ+23. Aldo Gunsing, Ritam Bhaumik, Ashwin Jha, Bart Mennink, and Yaobin Shen. Revisiting the indifferentiability of the sum of permutations. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 628–660. Springer, Cham, August 2023. 6

GGM86. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, October 1986. 2

GHHM21. Alex B. Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight adaptive reprogramming in the QROM. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 637–667. Springer, Cham, December 2021. 9

GKW+20. Chun Guo, Jonathan Katz, Xiao Wang, Chenkai Weng, and Yu Yu. Better concrete security for half-gates garbling (in the multi-instance setting). In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 793–822. Springer, Cham, August 2020. 4, 40, 41, 42

GKWY20. Chun Guo, Jonathan Katz, Xiao Wang, and Yu Yu. Efficient and secure multiparty computation from fixed-key block ciphers. In *2020 IEEE Symposium on Security and Privacy*, pages 825–841. IEEE Computer Society Press, May 2020. 4, 12, 40, 41, 42

GMO16. Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. ZKBoo: Faster zero-knowledge for Boolean circuits. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016*, pages 1069–1083. USENIX Association, August 2016. 2

GYW+23. Xiaojie Guo, Kang Yang, Xiao Wang, Wenhao Zhang, Xiang Xie, Jiang Zhang, and Zheli Liu. Half-tree: Halving the cost of tree expansion in COT and DPF. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part I*, volume 14004 of *LNCS*, pages 330–362. Springer, Cham, April 2023. 2, 3, 6, 35, 36

HJ24. Janik Huth and Antoine Joux. MPC in the head using the subfield bilinear collision problem. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part I*, volume 14920 of *LNCS*, pages 39–70. Springer, Cham, August 2024. 2, 4, 6

HJ25. Janik Huth and Antoine Joux. VOLE-in-the-head signatures from subfield bilinear collisions. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part IV*, volume 16248 of *LNCS*, pages 3–34. Springer, Singapore, December 2025. 2, 5, 6

HJMN24. Andreas Hülsing, David Joseph, Christian Majenz, and Anand Kumar Narayanan. On round elimination for special-sound multi-round identification and the generality of the hypercube for MPCitH. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part I*, volume 14920 of *LNCS*, pages 373–408. Springer, Cham, August 2024. 6

IKOS07. Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 21–30. ACM Press, June 2007. 2

JN18. Ashwin Jha and Mridul Nandi. A survey on applications of H-technique: Revisiting security analysis of PRP and PRF. Cryptology ePrint Archive, Report 2018/1130, 2018. 16

KKW18. Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 525–537. ACM Press, October 2018. 5

KLS24. Seongkwang Kim, Byeonghak Lee, and Mincheol Son. Relaxed vector commitment for shorter signatures. Cryptology ePrint Archive, Report 2024/1004, 2024. 2

KLS25a. Seongkwang Kim, Byeonghak Lee, and Mincheol Son. Relaxed vector commitment for shorter signatures. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part IV*, volume 15604 of *LNCS*, pages 427–455. Springer, Cham, May 2025. 2, 4, 5, 6

KLS25b. Seongkwang Kim, Byeonghak Lee, and Mincheol Son. Shorter VOLE-in-the-head-based signatures from vector semi-commitment. Cryptology ePrint Archive, Report 2025/1077, 2025. 6

KM07. Hidenori Kuwakado and Masakatu Morii. Indifferentiability of single-block-length and rate-1 compression functions. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E90-A(10):2301–2308, 2007. 6

KX26a. Haruhisa Kosuge and Keita Xagawa. New security proofs of TC-in-the-Head signatures in the quantum random oracle model. *IACR Communications in Cryptology*, 3(3), 2026. 2, 4, 6, 14

KX26b. Haruhisa Kosuge and Keita Xagawa. Towards formal security proofs of mqom. Cryptology ePrint Archive, Report 2026/629, 2026. 2

NIS22. NIST. Call for additional digital signature schemes for the post-quantum cryptography standardization process, October 2022. 2

NNL01. Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 41–62. Springer, Berlin, Heidelberg, August 2001. 5

OP01. Tatsuaki Okamoto and David Pointcheval. The gap-problems: A new class of problems for the security of cryptographic schemes. In Kwangjo Kim, editor, *PKC 2001*, volume 1992 of *LNCS*, pages 104–118. Springer, Berlin, Heidelberg, February 2001. 3, 34

Pat09. Jacques Patarin. The "coefficients H" technique (invited talk). In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *SAC 2008*, volume 5381 of *LNCS*, pages 328–345. Springer, Berlin, Heidelberg, August 2009. 3, 4, 16

WKK+25. Yalan Wang, Bryan Kumara, Harsh Kasyap, Liqun Chen, Sumanta Sarkar, Christopher J.P. Newton, Carsten Maple, and Ugur Ilker Atmaca. An improved vector commitment construction with applications to signatures. In *TrustCom 2025*, pages 1634–1639, 2025. See also https://eprint.iacr.org/2025/1411. 2, 6

ZCD+19. Greg Zaverucha, Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Jonathan Katz, Xiao Wang, and Vladmir Kolesnikov. Picnic. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions. 6

Zha19. Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 239–268. Springer, Cham, August 2019. 6

Supporting Materials

A MQOM

We review the parameter sets in Table 1 and the verification algorithm in Figure 11.

Table 1. Parameter Sets of the original MQOM (v.2.1) [BBFR25] with $|x|$ and $|\Delta_x^{(1)}[e]| + |\alpha_0[e]|$.

Parameter Sets	MQ Param.		Proof System Param.					Lengths (bits)	
	$\lvert\mathbb{F}\rvert$	$m = n$	τ	N	μ	η	w	$\lvert x\rvert$	$\lvert\Delta_x^{(1)}[e]\rvert + \lvert\alpha_0[e]\rvert$
MQOM2-L1-gf2-short-3r/5r	2	160	12	2048	16	10 / 8	8	160	192 / 160
MQOM2-L1-gf2-fast-3r/5r	2	160	17	256	8	20 / 16	9	160	192 / 160
MQOM2-L1-gf16-short-3r/5r	16	56	12	2048	4	14 / 8	8	224	320 / 224
MQOM2-L1-gf16-fast-3r/5r	16	56	17	256	2	28 / 16	9	224	320 / 224
MQOM2-L1-gf256-short-3r/5r	256	48	12	2048	2	24 / 8	8	384	640 / 384
MQOM2-L1-gf256-fast-3r/5r	256	48	17	256	1	48 / 16	9	384	640 / 384
MQOM2-L3-gf2-short-3r/5r	2	240	18	2048	16	15 / 12	12	240	288 / 240
MQOM2-L3-gf2-fast-3r/5r	2	240	27	256	8	30 / 24	3	240	288 / 240
MQOM2-L3-gf16-short-3r/5r	16	84	18	2048	4	21 / 12	12	336	480 / 336
MQOM2-L3-gf16-fast-3r/5r	16	84	27	256	2	42 / 24	3	336	480 / 336
MQOM2-L3-gf256-short-3r/5r	256	72	18	2048	2	36 / 12	12	576	960 / 576
MQOM2-L3-gf256-fast-3r/5r	256	72	27	256	1	72 / 24	3	576	960 / 576
MQOM2-L5-gf2-short-3r/5r	2	320	25	2048	16	20 / 16	6	320	384 / 320
MQOM2-L5-gf2-fast-3r/5r	2	320	36	256	8	40 / 32	4	320	384 / 320
MQOM2-L5-gf16-short-3r/5r	16	116	25	2048	4	29 / 16	6	464	672 / 464
MQOM2-L5-gf16-fast-3r/5r	16	116	36	256	2	58 / 32	4	464	672 / 464
MQOM2-L5-gf256-short-3r/5r	256	96	25	2048	2	48 / 16	6	768	1280 /768
MQOM2-L5-gf256-fast-3r/5r	256	96	36	256	1	96 / 32	4	768	1280 /768

- $|\Delta_x^{(1)}[e]| = n \cdot \log_2(|\mathbb{F}|) - \lambda$
- $|\alpha_0[e]| = \eta \cdot \log_2(|\mathbb{K}|) = \mu\eta \cdot \log_2(|\mathbb{F}|)$
- $\lambda = 128, 192, 256$ (for Cat. I, III, V respectively).

MQOM.Vrfy($\mathtt{vk}, \mathtt{msg}, \sigma$) / $\mathsf{MQOM}_{\mathtt{salt},\Omega}$.Vrfy($\mathtt{vk}, \mathtt{msg}, \sigma$)

1 : $(\mathtt{mseed}_{\mathtt{eq}}, \hat{y}) \coloneqq \mathsf{Parse}(\mathtt{vk}); (\{\hat{A}_i\}, \{\hat{b}_i\}) \coloneqq \mathsf{ExpandEquations}(\mathtt{mseed}_{\mathtt{eq}})$

2 : $(\mathtt{salt}, \mathtt{com}_1, \mathtt{com}_2, \alpha_1, \mathtt{pdecom}, \Delta_x^{(1)}, \mathtt{nonce}) \coloneqq \mathsf{Parse}(\sigma)$

3 : $\mathtt{msg_hash} \coloneqq \mathsf{Hash}_2(\mathtt{msg})$

4 : $h_2 \coloneqq \mathsf{Hash}_4(\mathtt{vk}, \mathtt{com}_1, \mathtt{com}_2, \mathtt{msg_hash})$ // MQOM

5 : $h_2 \coloneqq \mathsf{Hash}_4(\mathtt{vk}, \mathtt{salt}, \mathtt{com}_1, \mathtt{com}_2, \mathtt{msg_hash})$ // $\mathsf{MQOM}_{\mathtt{salt},\Omega}$

6 : $(\mathbf{i}^*, \mathtt{val}) \coloneqq \mathsf{XOF}_5(h_2, \mathtt{nonce})$

7 : **if** $\mathtt{val} \neq 0^w$ **then return** 0

8 : **for** $e \in [\tau]$ **do**

9 : $(\mathtt{com}[e], (\mathtt{seed}[e][i])_{i \neq \mathbf{i}^*[e]}, (\mathtt{tape}[e][i])_{i \neq \mathbf{i}^*[e]}) \coloneqq \mathsf{MQOM.Recon}(\mathtt{salt}, \mathtt{pdecom}[e], e, \mathbf{i}^*[e])$

10 : $h_{\mathtt{com}}[e] \coloneqq \mathsf{Hash}_6(\mathtt{com}[e])$

11 : **for** $i \in [N] \setminus \{\mathbf{i}^*[e]\}$ **do** $(\bar{x}[e][i], \bar{u}[e][i]) \coloneqq \mathsf{Parse}(\mathtt{seed}[e][i], \mathtt{tape}[e][i])$

12 : $\Delta_x[e] \coloneqq 0^\lambda \,\|\, \Delta_x^{(1)}[e]$

13 : // Compute $x_{\mathtt{eval}} = P_x(\omega)$

14 : $x_{\mathtt{eval}}[e] \coloneqq (\omega_{\mathbf{i}^*[e]} - \omega_0) \cdot \Delta_x[e] + \sum_{i \neq \mathbf{i}^*[e]} (\omega_{\mathbf{i}^*[e]} - \omega_i) \cdot \bar{x}[e][i]$ // $\omega_0 = 0$ in MQOM

15 : $u_{\mathtt{eval}}[e] \coloneqq \sum_{i \neq \mathbf{i}^*[e]} (\omega_{\mathbf{i}^*[e]} - \omega_i) \cdot \bar{u}[e][i]$ // Compute $u_{\mathtt{eval}} = P_u(\omega)$

16 : $\mathtt{com}_1' \coloneqq \mathsf{Hash}_7(h_{\mathtt{com}}, \Delta_x^{(1)})$

17 : **if** $\mathtt{com}_1' \neq \mathtt{com}_1$ **then return** 0

18 : $\Gamma \coloneqq \boldsymbol{I}_\eta$ // 3-round version

19 : $\Gamma \coloneqq \mathsf{XOF}_8(\mathtt{com}_1)$ // 5-round version

20 : **for** $e \in [\tau]$ **do**

21 : **for** $i \in [\hat{m}]$ **do**

22 : $t_{\mathtt{eval}} \coloneqq \hat{A}_i \cdot x_{\mathtt{eval}}[e] + \hat{b}_i \cdot \omega_{\mathbf{i}^*[e]}$

23 : $z_{\mathtt{eval},i} \coloneqq t_{\mathtt{eval}}^\top \cdot x_{\mathtt{eval}}[e] - \hat{y}_i \cdot \omega_{\mathbf{i}^*[e]}^2$

24 : $z_{\mathtt{eval}} \coloneqq (z_{\mathtt{eval},0}, \ldots, z_{\mathtt{eval},\hat{m}-1})$ // $= P_z(\omega_{\mathbf{i}^*[e]})$

25 : $\alpha_{\mathtt{eval}} \coloneqq u_{\mathtt{eval}}[e] + \Gamma \cdot z_{\mathtt{eval}}$

26 : $\alpha_0[e] \coloneqq \alpha_{\mathtt{eval}} - \alpha_1[e] \cdot \omega_{\mathbf{i}^*[e]}$

27 : $\mathtt{com}_2' \coloneqq \mathsf{Hash}_3(\alpha_0, \alpha_1)$

28 : **if** $\mathtt{com}_2' \neq \mathtt{com}_2$ **then return** 0

29 : **return** 1

Fig. 11. Expanded verification algorithm of MQOM (ver.2.1) and $\mathsf{MQOM}_{\mathtt{salt},\Omega}$.

B Summary of Oracles and Adversaries

Table 2. Oracles, Adversaries, and Assumptions Used in the Main Security Proofs

Result	Symbol	Role
Corollary 5.1	Hash_2	Hash function used for message hashing; its collision resistance is used through $\mathcal{A}_{\text{hash}}$.
	Hash_4	Random oracle used to derive the Fiat–Shamir challenge and reprogrammed at signing queries.
	E	Ideal cipher used to instantiate the Davies–Meyer-like function $\mathsf{H}_{\mathsf{avc}}(s,t) = E(t,s) \oplus \psi(s)$.
	$\mathcal{A}$	EUF-CMA adversary against $\mathsf{MQOM}_{\mathtt{salt},\Omega}$.
	$\mathcal{A}_{\text{nma}}$	EUF-NMA adversary constructed from $\mathcal{A}$.
	$\mathcal{A}_{\text{hash}}$	Collision-resistance adversary against Hash_2.
	$\mathcal{A}_{\text{prg}}$	Pseudorandomness adversary against PRG.
	$\mathcal{A}_{\text{ow}}$	One-wayness adversary against MQOM.Gen.
Corollary 5.2	$\mathsf{H}_{\mathsf{avc}}$	Random oracle replacing the Davies–Meyer-like function in the scGGM-based AVC of MQOM.
	Hash_2	Hash function used for message hashing; its collision resistance is used through $\mathcal{A}_{\text{hash}}$.
	Hash_4	Random oracle used to derive the Fiat–Shamir challenge and reprogrammed at signing queries.
	$\mathcal{A}$	EUF-CMA adversary against $\mathsf{MQOM}_{\mathtt{salt},\Omega}$ when $\mathsf{H}_{\mathsf{avc}}$ is modeled as a random oracle.
	$\mathcal{A}_{\text{nma}}$	EUF-NMA adversary constructed from $\mathcal{A}$.
	$\mathcal{A}_{\text{hash}}$	Collision-resistance adversary against Hash_2.
	$\mathcal{A}_{\text{prg}}$	Pseudorandomness adversary against PRG.
	$\mathcal{A}_{\text{pgow}}$	Partial-guessing one-wayness adversary extracting $\delta = \mathsf{FirstBits}_\lambda(x)$ from a bad $\mathsf{H}_{\mathsf{avc}}$ query.

C Variations of One-wayness

We define the one-wayness notions used in the security proofs. Let $\mathsf{F} = \{\mathsf{F}_\theta : \mathcal{X} \to \mathcal{Y}\}_{\theta\in\Theta}$ be an efficiently computable family of functions indexed by public information θ. Assume that $\mathcal{X} \subseteq \{0,1\}^\ell$ for some $\ell \geq \lambda$. Let $\mathcal{D}_\mathsf{F}$ be a distribution that samples (θ, x, y) such that $x \in \mathcal{X}$ and $y = \mathsf{F}_\theta(x)$. In the applications considered in this paper, θ is part of the public key and x is the secret witness.

Standard one-wayness asks the adversary to recover a full preimage of y. Partial-guessing one-wayness, following Feneuil and Rivain [FR26], asks the adversary to recover only the distinguished prefix $\delta := \mathsf{FirstBits}_\lambda(x)$ while it can access to a checking oracle. This notion matches our reductions, where a bad oracle query reveals precisely this prefix, rather than the entire witness.

Definition C.1 (One-wayness). *We say that* F *is* one-way *with respect to* $\mathcal{D}_\mathsf{F}$ *if, for any PPT/QPT adversary* $\mathcal{A}$*, its advantage* $\mathsf{Adv}^{\text{ow}}_{\mathsf{F},\mathcal{A}}(\lambda)$ *is negligible in* λ*, where*

$$\mathsf{Adv}^{\text{ow}}_{\mathsf{F},\mathcal{A}}(\lambda) := \Pr\left[(\theta,x,y) \leftarrow \mathcal{D}_\mathsf{F}, x' \leftarrow \mathcal{A}(\theta,y) : \mathsf{F}_\theta(x') = y\right].$$

Definition C.2 (Partial-guessing one-wayness (PGOW), adapted [FR26]). *For* $x \in \mathcal{X}$*, let* $\delta := \mathsf{FirstBits}_\lambda(x)$ *and define the guessing predicate*

$$\textsc{Guess}^{\text{pgow}}_x(\delta') := \mathsf{boole}(\delta' = \delta).$$

In the quantum setting[10]*, access to this predicate is given as the quantum oracle*

$$|\textsc{Guess}^{\text{pgow}}_x\rangle : |\delta'\rangle|b\rangle \longmapsto |\delta'\rangle|b \oplus \textsc{Guess}^{\text{pgow}}_x(\delta')\rangle.$$

[10] We allow quantum access because the QROM reduction invokes this predicate coherently while answering quantum random-oracle queries.

We say that F *is* partial-guessing one-way *with respect to* $\mathcal{D}_{\mathsf{F}}$ *if, for any PPT/QPT adversary* $\mathcal{A}$ *with access to* $\textsc{Guess}_x^{\mathsf{pgow}}$*, its advantage* $\mathsf{Adv}^{\mathsf{pgow}}_{\mathsf{F},\mathcal{A}}(\lambda)$ *is negligible in* λ*, where*

$$\mathsf{Adv}^{\mathsf{pgow}}_{\mathsf{F},\mathcal{A}}(\lambda) := \Pr\left[(\theta, x, y) \leftarrow \mathcal{D}_{\mathsf{F}}, \delta' \leftarrow \mathcal{A}^{\textsc{Guess}_x^{\mathsf{pgow}}}(\theta, y) \,:\, \delta' = \delta\right].$$

The guessing predicate in Definition C.2 abstracts the candidate test used in the reduction. Such a test can be justified in several ways. The most direct approach is to use a scheme-specific argument that validates candidates extracted from primitive queries. For example, Feneuil and Rivain [FR26, Section 5.3] use the concrete structure of MQOM to validate such candidates. When no such candidate test is available, one can still fall back to partial-domain one-wayness. This fallback is inherently non-tight: the reduction must guess which guessing query is the first correct one. This loss is formalized in Lemma C.1.

Definition C.3 (Partial-domain one-wayness (PDOW), adapted [OP01]). *Let* $\mathsf{F} = \{\mathsf{F}_\theta : \mathcal{X} \to \mathcal{Y}\}_{\theta\in\Theta}$ *be a function family. Let* $\mathcal{D}_{\mathsf{F}}$ *be a distribution that samples* (θ, x, y) *with* $\mathsf{F}_\theta(x) = y$*. For an adversary* $\mathcal{A}$*, we define its partial-domain one-wayness advantage as*

$$\mathsf{Adv}^{\mathsf{pdow}}_{\mathsf{F},\mathcal{A}}(\lambda) := \Pr\left[(\theta, x, y) \leftarrow \mathcal{D}_{\mathsf{F}}, \delta := \mathsf{FirstBits}_\lambda(x), \delta' \leftarrow \mathcal{A}(\theta, y) \,:\, \delta' = \delta\right].$$

We say that F *is partial-domain one-way if* $\mathsf{Adv}^{\mathsf{pdow}}_{\mathsf{F},\mathcal{A}}(\lambda)$ *is negligible for every PPT adversary* $\mathcal{A}$*.*

The following lemma shows that if F is PDOW, then F is also PGOW in the *classical* setting.

Lemma C.1 (PGOW to PDOW). *Let* $\mathcal{A}_{\mathrm{pgow}}$ *be an adversary against partial-guessing one-wayness that makes at most* Q_{guess} *classical guessing-oracle queries. Then there exists an adversary* $\mathcal{A}_{\mathrm{pdow}}$ *against partial-domain one-wayness such that*

$$\mathsf{Adv}^{\mathsf{pgow}}_{\mathsf{F},\mathcal{A}_{\mathrm{pgow}}}(\lambda) \le (Q_{\mathsf{guess}} + 1) \cdot \mathsf{Adv}^{\mathsf{pdow}}_{\mathsf{F},\mathcal{A}_{\mathrm{pdow}}}(\lambda).$$

Moreover, $\mathcal{A}_{\mathrm{pdow}}$ *runs in essentially the same time as* $\mathcal{A}_{\mathrm{pgow}}$*.*

Proof. The adversary $\mathcal{A}_{\mathrm{pdow}}$ is given (θ, y). The definition of $\mathcal{A}_{\mathrm{pdow}}$ follows:

- It samples an index $\hat{j} \leftarrow [Q_{\mathsf{guess}} + 1]$ and runs $\mathcal{A}_{\mathrm{pgow}}(\theta, y)$.
  - If $\hat{j} < Q_{\mathsf{guess}}$: Whenever $\mathcal{A}_{\mathrm{pgow}}$ submits its j-th guessing query δ_j, the adversary $\mathcal{A}_{\mathrm{pdow}}$ answers as follows.
    - If $j < \hat{j}$, it returns 0.
    - If $j = \hat{j}$, it outputs δ_j as its PDOW solution and stops.
    - If $\mathcal{A}_{\mathrm{pgow}}$ terminates before submitting its $\hat{j}$-th guessing query, then $\mathcal{A}_{\mathrm{pdow}}$ outputs an arbitrary λ-bit string, e.g., 0^λ, and stops.
  - If $\hat{j} = Q_{\mathsf{guess}}$: On every j-th guessing query, $\mathcal{A}_{\mathrm{pdow}}$ always answers 0. When $\mathcal{A}_{\mathrm{pgow}}$ halts with output δ', then $\mathcal{A}_{\mathrm{pdow}}$ outputs δ' and stops.

Consider a successful execution of $\mathcal{A}_{\mathrm{pgow}}$ in the PGOW experiment, numbering its guessing queries from 0. If a guessing query equals δ, let $j^\star$ be the index of the first such query. Otherwise, set $j^\star := Q_{\mathsf{guess}}$.

If $j^\star < Q_{\mathsf{guess}}$ and $\hat{j} = j^\star$, all earlier guessing queries are incorrect, so answering them with 0 perfectly simulates the real guessing oracle. The reduction therefore reaches query $j^\star$ and outputs $\delta_{j^\star} = \delta$.

If $j^\star = Q_{\mathsf{guess}} = \hat{j}$, every guessing query is incorrect. Thus, answering every query with 0 perfectly simulates the entire execution. Since this execution is successful, the final output δ' equals δ, and the reduction outputs δ'.

For every successful PGOW execution, the independent uniform choice of $\hat{j}$ equals $j^\star$ with probability $1/(Q_{\mathsf{guess}} + 1)$. Hence,

$$\mathsf{Adv}^{\mathsf{pdow}}_{\mathsf{F},\mathcal{A}_{\mathrm{pdow}}}(\lambda) \ge \frac{1}{Q_{\mathsf{guess}} + 1} \mathsf{Adv}^{\mathsf{pgow}}_{\mathsf{F},\mathcal{A}_{\mathrm{pgow}}}(\lambda).$$

Rearranging gives the claim. □

D Review of cGGM-based AVC

The definition of cGGM-based AVC is given in Section 2.5. An example of the computation for $N = 8$ is summarized in Figure 1, where we use $\mathsf{H}_{\mathsf{tree}} : \{0,1\}^\lambda \to \{0,1\}^\lambda$, $\mathsf{H}_{\mathsf{com}} : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$, and $\mathsf{H}_{\mathsf{tape}} : \{0,1\}^\lambda \to \{0,1\}^{n_b\lambda}$.[11]

We say that cGGM-based AVC is secure if, for any $\mathtt{salt}$, e, i^*, and any PPT adversary, the following two distributions are indistinguishable:

- Real: Sample $\mathtt{rseed}, \delta \leftarrow \{0,1\}^\lambda$ and compute $(\mathtt{com}, \mathtt{decom}, \mathtt{seed}, \mathtt{tape}) := \mathsf{AVC.Commit}(\mathtt{salt}, \mathtt{rseed}, \delta, e)$. Compute $\mathtt{pdecom} := \mathsf{AVC.Open}(\mathtt{decom}, e, i^*)$. Output $(\mathtt{pdecom}, \mathtt{tape}[i^*])$.
- Ideal: Sample $\mathtt{pdecom} \leftarrow (\{0,1\}^\lambda)^d \times \{0,1\}^{2\lambda}$ and $\mathtt{tape}[i^*] \leftarrow \{0,1\}^{n_b\lambda}$. Output $(\mathtt{pdecom}, \mathtt{tape}[i^*])$.

We briefly review the existing proofs of security for cGGM-based AVC [GYW+23, CLY+24, BCD25]. We follow the notation of Bui et al. [BCD25], slightly adapted to our setting, and explain how to simulate the real and ideal games using TCCR/TCR hashes or random functions, respectively, where we only use $\mathtt{salt}$ as a tweak for simplicity.

Let $\delta \in \{0,1\}^\lambda$ be the secret correlation value. The reduction algorithm has access to three oracles $(O_{\mathsf{tree}}, O_{\mathsf{com}}, O_{\mathsf{tape}})$ defined as follows:

- In the real world:

$$\begin{aligned}\mathcal{O}^{\mathrm{TCCR}}_{\mathsf{tree},\delta}(s,t,b) &:= \mathsf{H}_{\mathsf{tree}}(s \oplus \delta, t) \oplus b \cdot \delta,\\ \mathcal{O}^{\mathrm{TCR}}_{\mathsf{com},\delta}(s,t) &:= \mathsf{H}_{\mathsf{com}}(s \oplus \delta, t),\\ \mathcal{O}^{\mathrm{TCR}}_{\mathsf{tape},\delta}(s,t) &:= \mathsf{H}_{\mathsf{tape}}(s \oplus \delta, t);\end{aligned}$$

- In the ideal world: independent random functions $f_{\mathsf{tree}}(s,t,b)$, $f_{\mathsf{com}}(s,t)$, and $f_{\mathsf{tape}}(s,t)$.

Using these oracles, the reduction computes the co-paths, commitments, and tapes as follows:

1. Sample $\mathtt{path}[1] \leftarrow \{0,1\}^\lambda$.
2. For $j = 2, \ldots, d$, define:

$$S_{j-1} := \bigoplus_{j' \in [1:j]} \mathtt{path}[j'],$$

$$\mathtt{path}[j] := \begin{cases} O_{\mathsf{tree}}(S_{j-1}, \mathtt{salt}, 0) & \text{if } \mathsf{copath}_j(i^*) \text{ is a left node,}\\ O_{\mathsf{tree}}(S_{j-1}, \mathtt{salt}, 1) \oplus S_{j-1} & \text{otherwise.}\end{cases}$$

3. Finally, set $S_d := \bigoplus_{j' \in [1:d+1]} \mathtt{path}[j']$ and compute $\mathtt{com}[i^*] := O_{\mathsf{com}}(S_d, \mathtt{salt})$ and $\mathtt{tape}[i^*] := O_{\mathsf{tape}}(S_d, \mathtt{salt})$.

This is a perfect simulation because, in the real world, the oracles computes values consistently with $\mathtt{node}[\mathsf{path}_{j-1}(i^*)] = S_{j-1} \oplus \delta$. A graphical illustration for the case $N = 8$ and $i^* = 5$ is given in Figure 12.

This formulation isolates the dependency on the secret correlation value δ and enables a reduction to the security of the underlying hash primitives.

E Proof of Theorem 3.1

We recap the theorem here.

Theorem E.1 (Theorem 3.1). *Let $\mathsf{H}_{\mathsf{avc}}$ be a random oracle. Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ be a QPT adversary for the simulation experiment in Figure 7. Assume that $\mathcal{A}_1$ makes at most $Q_{\mathsf{H}_{\mathsf{avc}}}$ queries to $\mathsf{H}_{\mathsf{avc}}$. Then there exists a QPT adversary $\mathcal{A}_{\mathrm{pgow}}$ such that*

$$\mathsf{Adv}^{\mathrm{avc}}_{\mathsf{MQOM}_{\Omega},\mathcal{A}}(\lambda) \le \begin{cases} \mathsf{Adv}^{\mathrm{pgow}}_{\mathsf{F}_{\mathsf{MQOM}},\mathcal{A}_{\mathrm{pgow}}}(\lambda) + 2\Pr[\delta = 0^\lambda] + \frac{2Q\tau(2N-d-3)}{2^\lambda} & \textit{in the ROM,}\\ 2\sqrt{Q_{\mathsf{H}_{\mathsf{avc}}}\mathsf{Adv}^{\mathrm{pgow}}_{\mathsf{F}_{\mathsf{MQOM}},\mathcal{A}_{\mathrm{pgow}}}(\lambda)} + 2\Pr[\delta = 0^\lambda] + \frac{2Q\tau(2N-d-3)}{2^\lambda} & \textit{in the QROM.}\end{cases}$$

The running time of $\mathcal{A}_{\mathrm{pgow}}$ is approximately that of $\mathcal{A}$. The adversary $\mathcal{A}_{\mathrm{pgow}}$ makes at most $2Q_{\mathsf{H}_{\mathsf{avc}}}$ queries to its guessing oracle.

Before giving the proof, we review the semi-classical oneway-to-hiding (SC-O2H) lemma. We then prove our theorem, Theorem 3.1.

[11] n_b denotes "the number of blocks."

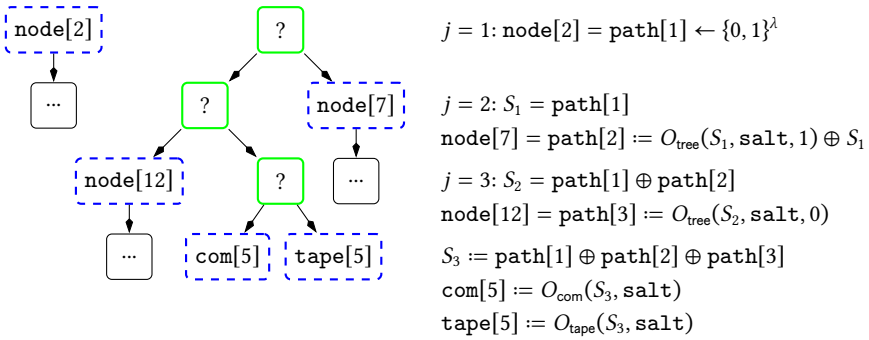


Fig. 12. How to use oracles for $N = 8$, $d = 3$, and $i^* = 5$ in [GYW+23, CLY+24, BCD25], where we adopt the notations. Green nodes $\mathtt{node}[3]$, $\mathtt{node}[6]$, and $\mathtt{node}[13] = \mathtt{seed}[5]$ are hidden.

Semi-classical O2H Technique [AHU19]: We use the following form of the semi-classical O2H lemma.

Lemma E.1 (Semi-classical O2H Technique [BHH+19, Definition 1 and Lemma 3]). *Let $S \subseteq \mathcal{X}$ be random. Let $\mathsf{H}, \mathsf{H}' : \mathcal{X} \to \mathcal{Y}$ be random functions satisfying $\mathsf{H}(x) = \mathsf{H}'(x)$ for every $x \notin S$. Let z be a random bitstring. The tuple $(\mathsf{H}, \mathsf{H}', S, z)$ may have an arbitrary joint distribution. Let $\mathcal{A}$ be a quantum oracle algorithm making at most q oracle queries, and let E be an arbitrary classical event.*

Define the punctured oracle $\mathsf{H}'\backslash S$ as follows. Before answering each quantum query, it computes the membership bit $\mathsf{boole}(x \in S)$ into an auxiliary qubit initially set to $|0\rangle$ and measures that qubit. It then answers the query using H'. Let Find be the event that at least one of these measurements returns 1. Then

$$\left|\Pr\left[\mathcal{A}^{|\mathsf{H}\rangle}(z) : \mathsf{E}\right] - \Pr\left[\mathcal{A}^{|\mathsf{H}'\rangle}(z) : \mathsf{E}\right]\right| \le 2\sqrt{q \Pr\left[\mathcal{A}^{|\mathsf{H}'\backslash S\rangle}(z) : \mathsf{Find}\right]}.$$

To bound the probability of Find, the simulator may stop at the first membership measurement returning 1 and measure the corresponding query input. Thus, it only needs to implement the membership test and the oracle values outside S.

Proof. We use a common sequence of games for the ROM and the QROM. Let W_i be the event that $\mathcal{A}_1$ outputs 1 in G_i.

For clarity, we present the proof for a full binary tree with $N = 2^d$ leaves. The same bound also holds for the pruned $N = 255$ tree defined above, with $d = 8$ denoting its maximum depth. In that case, all hidden-path formulas and index ranges use the actual depth of the hidden leaf, and the dummy copath entry is ignored.

- Game 0: This is the real experiment $\mathsf{Expt}^{\mathrm{avc},0}_{\mathsf{MQOM}_\Omega,\mathcal{A}}(\lambda)$ in Figure 7. Roughly speaking, the experiment is summarized as follows:
  1. The first adversary $\mathcal{A}_0$ is given vk and outputs $\{(\mathtt{salt}^{(q)}, \mathbf{i}^{*(q)})\}_{q\in[Q]}$.
  2. The real oracle computes the transcripts $\mathsf{out}^{(q)}$ correctly by using δ, x, and $\mathsf{H}_{\mathsf{avc}}$.
  3. The second adversary $\mathcal{A}_1$ receives the transcripts and has access to $\mathsf{H}_{\mathsf{avc}}$ and outputs b'.
- Game 1: We first define $\mathsf{H}_{\mathsf{avc}}{}'$, which disagrees on the hidden nodes. To do so, we define the set of hidden nodes from δ and public information, $\mathtt{path}$. Recall that in any subtree, the sum of leaves equals the subtree's root. For each $q \in [Q]$, $e \in [\tau]$, and layer $j \in [1 : d+1]$, define the hidden node at layer j and the hidden leaf seed by

$$\mathtt{node}^{(q,e,j)}_{\mathsf{hid}} := \delta \oplus \bigoplus_{j'=1}^{j} \mathtt{path}^{(q)}[e][j'],$$
$$\mathtt{seed}^{(q,e)}_{\mathsf{hid}} := \mathtt{node}^{(q,e,d)}_{\mathsf{hid}}.$$

We set

$$\mathcal{S}_{\mathsf{tree}} := \left\{\left(\mathtt{node}^{(q,e,j)}_{\mathsf{hid}}, \mathsf{TweakSalt}(\mathtt{salt}^{(q)}, 2, e, j)\right) : q \in [Q],\ e \in [\tau],\ j \in [1 : d]\right\},$$
$$\mathcal{S}_{\mathsf{com}} := \left\{\left(\mathtt{seed}^{(q,e)}_{\mathsf{hid}}, \mathsf{TweakSalt}(\mathtt{salt}^{(q)}, k, e, 0)\right) : q \in [Q],\ e \in [\tau],\ k \in \{0,1\}\right\},$$
$$\mathcal{S}_{\mathsf{tape}} := \left\{\left(\mathtt{seed}^{(q,e)}_{\mathsf{hid}}, \mathsf{TweakSalt}(\mathtt{salt}^{(q)}, 3, e, j)\right) : q \in [Q],\ e \in [\tau],\ j \in [n_b]\right\},$$

and

$$\mathcal{S} := \mathcal{S}_{\mathsf{tree}} \cup \mathcal{S}_{\mathsf{com}} \cup \mathcal{S}_{\mathsf{tape}}. \tag{13}$$

The modified oracle $\mathsf{H}_{\mathsf{avc}}'$ agrees with $\mathsf{H}_{\mathsf{avc}}$ outside $\mathcal{S}$ and returns independently uniform values in $\{0,1\}^\lambda$ on $\mathcal{S}$. Since $\mathcal{A}_0$ fixes the salts and hidden indices before $\mathcal{A}_1$ starts, and since the challenger aborts when the first $\lambda - 12$ bits of two salts collide, the set $\mathcal{S}$ is fixed before $\mathcal{A}_1$ makes any random-oracle query.

Game 1 is defined by using $\mathsf{H}_{\mathsf{avc}}$ *and* $\mathsf{H}_{\mathsf{avc}}'$ as follows:

1. The first adversary $\mathcal{A}_0$ is given $\mathtt{vk}$ and outputs $\{(\mathtt{salt}^{(q)}, \mathbf{i}^{*(q)})\}_{q\in[Q]}$.
2. The real oracle computes the transcripts $\mathsf{out}^{(q)}$ correctly by using δ, x, and $\mathsf{H}_{\mathsf{avc}}$.
3. The second adversary $\mathcal{A}_1$ receives the transcripts and has access to $\mathsf{H}_{\mathsf{avc}}'$ rather than $\mathsf{H}_{\mathsf{avc}}$ and outputs b'.

– Game 2: This is the simulated experiment $\mathsf{Expt}^{\mathrm{avc},1}_{\mathsf{MQOM}_\Omega,\mathcal{A}}(\lambda)$.

Equivalently, Game 2 is rewritten as follows:

1. The first adversary $\mathcal{A}_0$ is given $\mathtt{vk}$ and outputs $\{(\mathtt{salt}^{(q)}, \mathbf{i}^{*(q)})\}_{q\in[Q]}$.
2. The ideal oracle computes the transcripts $\mathsf{out}^{(q)}$, which are chosen uniformly at random.
3. The second adversary $\mathcal{A}_1$ receives the transcripts and has access to $\mathsf{H}_{\mathsf{avc}}'$ rather than $\mathsf{H}_{\mathsf{avc}}$ and outputs b'.

To identify the cases in which Game 1 and Game 2 may differ, let $\mathcal{S}_{\mathrm{rec}}$ denote the set of oracle inputs used by MQOM.Recon to reconstruct the opened transcripts, and define $\mathsf{Coll}_{\mathrm{int}} := \{\mathcal{S} \cap \mathcal{S}_{\mathrm{rec}} \neq \emptyset\}$. Once salt-prefix collisions have been excluded, TweakSalt produces distinct tweaked salts for different signing queries, repetitions, layers, and purposes; hence, only values used with the same tweaked salt need to be compared.

Fix a tree (q, e) and suppose that $\delta \neq 0^\lambda$. We expose the tree layer by layer. In addition to collisions between hidden and opened nodes that are evaluated with the same tweaked salt, we declare a bad event whenever a hidden internal node equals 0^λ. Each layer is analyzed conditioned on the absence of all bad events declared at earlier layers.

– At the first layer, the two node values are $\mathtt{rseed}$ and $\mathtt{rseed} \oplus \delta$. They cannot coincide because $\delta \neq 0^\lambda$. The hidden node is uniform, so the probability that it equals 0^λ is $1/2^\lambda$. We declare this event bad because a zero parent has two identical children.
– At each internal layer $j \in \{2, \ldots, d-1\}$, the hidden node cannot coincide with its sibling, since their parent is nonzero. Moreover, the hidden parent differs from every opened parent at the preceding layer. Its layer-specific random-oracle output is therefore fresh and uniform, even after exposing the outputs used to generate the children of the opened parents. Consequently, the hidden node coincides with any fixed non-sibling opened node with probability $1/2^\lambda$.
  There are $2^j - 2$ such non-sibling opened nodes. A union bound gives a contribution of at most $(2^j - 2)/2^\lambda$ for these collisions. We also declare the event that the hidden node equals 0^λ bad, contributing at most $1/2^\lambda$. These events are bounded together, without conditioning on individual comparisons within the layer.
– At the leaf level, the hidden leaf cannot coincide with its sibling because their parent is nonzero. By the same fresh-output argument, its equality with any of the $N - 2$ non-sibling opened leaves has probability $1/2^\lambda$. Thus, the leaf-level contribution is at most $(N-2)/2^\lambda$. We do not need to exclude a zero hidden leaf, since it has no children.

Every internal collision is covered by the enlarged bad event. Summing the bounds on the probability of the first bad layer gives

$$\Pr[\mathsf{Coll}_{\mathrm{int}}^{(q,e)} \mid \delta \neq 0^\lambda] \leq \sum_{j=2}^{d} \frac{2^j - 2}{2^\lambda} + \sum_{j=1}^{d-1} \frac{1}{2^\lambda} = \frac{2N - d - 3}{2^\lambda},$$

where $\mathsf{Coll}_{\mathrm{int}}^{(q,e)}$ denotes an internal collision in the fixed tree (q, e).

Therefore, in Game 1, we have

$$\begin{aligned}
\Pr[\mathsf{Coll}_{\mathrm{int}}] &\leq \Pr[\mathsf{Coll}_{\mathrm{int}} \mid \delta = 0^\lambda] \cdot \Pr[\delta = 0^\lambda] + \Pr[\mathsf{Coll}_{\mathrm{int}} \mid \delta \neq 0^\lambda] \cdot \Pr[\delta \neq 0^\lambda] \\
&\leq \Pr[\delta = 0^\lambda] + \Pr[\mathsf{Coll}_{\mathrm{int}} \mid \delta \neq 0^\lambda] \\
&\leq \Pr[\delta = 0^\lambda] + \frac{Q\tau(2N - d - 3)}{2^\lambda},
\end{aligned}$$

where we take the union bound for (q, e).

An analogous layer-wise argument gives the same bound in the ideal experiment, G_2.

In the ROM and QROM analyses below, we first consider executions that abort when $\mathsf{Coll}_{\mathrm{int}}$ occurs. Rather than introducing separate game notation, we write their probabilities as joint events with $\neg\mathsf{Coll}_{\mathrm{int}}$. On this event, the hidden oracle inputs are disjoint from the reconstruction inputs. The bijectivity of the combined auxiliary map for MQOM_{Ω}, which outputs $\Delta_x^{(1)}$ and α_0, and the fact that $\mathcal{A}_1$ cannot access $\mathsf{H}_{\mathsf{avc}}$ therefore imply that, for every event E considered below,

$$\Pr[\mathsf{G}_1 : \mathsf{E} \wedge \neg\mathsf{Coll}_{\mathrm{int}}] = \Pr[\mathsf{G}_2 : \mathsf{E} \wedge \neg\mathsf{Coll}_{\mathrm{int}}].$$

The restriction to $\neg\mathsf{Coll}_{\mathrm{int}}$ will be removed at the end of each analysis.

ROM analysis. We now assume that $\mathcal{A}_1$ has classical access to the random oracle $\mathsf{H}_{\mathsf{avc}}$. Let Bad be the event that $\mathcal{A}_1$ queries $\mathsf{H}_{\mathsf{avc}}$ on an input in $\mathcal{S}$. Since $\mathsf{H}_{\mathsf{avc}}$ and $\mathsf{H}_{\mathsf{avc}}{}'$ agree outside $\mathcal{S}$, the fundamental lemma of game playing, applied to the collision-aborting executions, gives

$$|\Pr[\mathsf{G}_0 : W_0 \wedge \neg\mathsf{Coll}_{\mathrm{int}}] - \Pr[\mathsf{G}_1 : W_1 \wedge \neg\mathsf{Coll}_{\mathrm{int}}]| \le \Pr[\mathsf{G}_1 : \mathsf{Bad} \wedge \neg\mathsf{Coll}_{\mathrm{int}}].$$

The identity between the collision-free branches of Game 1 and Game 2 therefore gives

$$|\Pr[\mathsf{G}_0 : W_0 \wedge \neg\mathsf{Coll}_{\mathrm{int}}] - \Pr[\mathsf{G}_2 : W_2 \wedge \neg\mathsf{Coll}_{\mathrm{int}}]| \le \Pr[\mathsf{G}_2 : \mathsf{Bad} \wedge \neg\mathsf{Coll}_{\mathrm{int}}] \le \Pr[\mathsf{G}_2 : \mathsf{Bad}].$$

We construct a PGOW adversary $\mathcal{A}_{\mathrm{pgow}}$ that bounds $\Pr[\mathsf{G}_2 : \mathsf{Bad}]$. Given (θ, y), the adversary $\mathcal{A}_{\mathrm{pgow}}$ runs $\mathcal{A}_0(\theta, y)$ to obtain

$$\left(\mathtt{state}, \{(\mathtt{salt}^{(q)}, \mathbf{i}^{*(q)})\}_{q\in[Q]}\right).$$

It aborts if the first $\lambda - 12$ bits of two distinct salts $\mathtt{salt}^{(q)}$ and $\mathtt{salt}^{(q')}$ collide. Otherwise, it samples the simulated transcripts exactly as in Game 2 and runs $\mathcal{A}_1$ on these transcripts.

To recover δ from random-oracle queries, $\mathcal{A}_{\mathrm{pgow}}$ associates each query with the recorded AVC transcript including the opening path $\mathtt{path}^{(q)}[e]$. A query to $\mathsf{H}_{\mathsf{avc}}$ has the form (s, t), where t is a tweaked salt. By the non-collision assumption, the first $\lambda - 12$ bits of t identify the recorded transcript labelled by q. Once this transcript is identified, the raw salt $\mathtt{salt}^{(q)}$ is also known, and the tweak value is recovered from $t \oplus \mathtt{salt}^{(q)}$. Indeed, this value is the binary encoding of $k + 4e + 256j$ for some auxiliary index j. Thus, decoding it recovers the domain index k, the repetition index e, and the auxiliary index j.

- If $k \in \{0, 1\}$, the query is interpreted as a commitment query and is treated as a candidate for $\mathcal{S}_{\mathsf{com}}$; in this case, s is treated as a candidate hidden leaf seed.
- If $k = 2$, the query is interpreted as a tree-derivation query and is treated as a candidate for $\mathcal{S}_{\mathsf{tree}}$; in this case, s is treated as a candidate hidden tree node and j is the layer index.
- If $k = 3$, the query is interpreted as a tape-expansion query and is treated as a candidate for $\mathcal{S}_{\mathsf{tape}}$; in this case, s is treated as a candidate hidden leaf seed.

Consequently, the pair (q, e) identifies at most one recorded AVC transcript. The domain index k determines whether the query is treated as a candidate for $\mathcal{S}_{\mathsf{com}}$, $\mathcal{S}_{\mathsf{tree}}$, or $\mathcal{S}_{\mathsf{tape}}$.

After identifying the corresponding transcript and indices, $\mathcal{A}_{\mathrm{pgow}}$ extracts a candidate secret-key root from the recorded query. For each recorded query, $\mathcal{A}_{\mathrm{pgow}}$ first uses the above decoding procedure to determine whether the query is associated with some recorded transcript and repetition. If no such pair (q, e) is found, it skips the query. For a tree-derivation query associated with (q, e, j) and node input $\mathtt{node}$, it computes

$$\delta' := \mathtt{node} \oplus \bigoplus_{j'=1}^{j} \mathtt{path}^{(q)}[e][j'].$$

For a commitment or tape-expansion query associated with (q, e) and leaf-seed input $\mathtt{seed}$, it computes

$$\delta' := \mathtt{seed} \oplus \bigoplus_{j'=1}^{d} \mathtt{path}^{(q)}[e][j'].$$

It submits each such δ' to its guessing oracle. If the guessing oracle accepts some candidate, $\mathcal{A}_{\mathrm{pgow}}$ outputs that candidate; otherwise, it outputs an arbitrary value.

If Bad occurs in Game 2, then one recorded query is a hidden input for some transcript. For that query, the reconstruction above gives

$$\delta' = \delta = \mathsf{FirstBits}_\lambda(x).$$

Thus, the guessing oracle accepts. Therefore,

$$\Pr[\mathsf{G}_2 : \mathsf{Bad}] \le \mathsf{Adv}^{\mathrm{pgow}}_{\mathsf{F}_{\mathrm{MQOM}},\mathcal{A}_{\mathrm{pgow}}}(\lambda).$$

Combining the above bounds gives

$$\mathsf{Adv}^{\mathrm{avc}}_{\mathsf{MQOM}_\Omega,\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\mathrm{pgow}}_{\mathsf{F}_{\mathrm{MQOM}},\mathcal{A}_{\mathrm{pgow}}}(\lambda) + 2\Pr[\delta = 0^\lambda] + \frac{2Q\tau(2N-d-3)}{2^\lambda}$$

in the ROM.

QROM analysis. We now assume that $\mathcal{A}_1$ has quantum access to the random oracle $\mathsf{H}_{\mathsf{avc}}$. The classical bad-query argument used in the ROM does not apply directly. Instead, we apply the semi-classical O2H lemma to the reprogramming set $\mathcal{S}$ defined in Equation 13.

We define a transcript-dependent reconstruction map f. For a tree-derivation input $\xi = (\mathtt{node}, t_{\mathtt{salt}^{(q)},2,e,j})$ associated with a recorded AVC transcript, define

$$f(\xi) := \mathtt{node} \oplus \bigoplus_{j'=1}^{j} \mathtt{path}^{(q)}[e][j']. \tag{14}$$

For a commitment or tape-expansion input $\xi = (\mathtt{seed}, t_{\mathtt{salt}^{(q)},k,e,j})$ associated with a recorded AVC transcript, define

$$f(\xi) := \mathtt{seed} \oplus \bigoplus_{j'=1}^{d} \mathtt{path}^{(q)}[e][j']. \tag{15}$$

For inputs whose salt, purpose, repetition, and layer or block indices do not occur in the definition of $\mathcal{S}$, define $f(\xi) := \perp$. By construction, $\xi \in \mathcal{S}$ if and only if $f(\xi) = \delta$.

Applying Lemma E.1 to the collision-aborting executions, there exists a quantum oracle algorithm $\mathcal{B}$ running in approximately the same time as $\mathcal{A}$ such that

$$|\Pr[\mathsf{G}_0 : W_0 \wedge \neg\mathsf{Coll}_{\mathrm{int}}] - \Pr[\mathsf{G}_1 : W_1 \wedge \neg\mathsf{Coll}_{\mathrm{int}}]| \le 2\sqrt{Q_{\mathsf{H}_{\mathsf{avc}}}\Pr[\mathsf{G}_1 : \mathsf{Find} \wedge \neg\mathsf{Coll}_{\mathrm{int}}]}.$$

Here, Find denotes the event that, in the punctured execution of Game 1, at least one membership measurement returns 1. When this occurs, measuring the corresponding query input yields $\xi \in \mathcal{S}$, from which the reconstruction map recovers $f(\xi) = \delta$. On $\neg\mathsf{Coll}_{\mathrm{int}}$, the punctured executions of Game 1 and Game 2 have the same distribution until Find occurs. Hence, [12]

$$\begin{aligned}\Pr[\mathsf{G}_1 : \mathsf{Find} \wedge \neg\mathsf{Coll}_{\mathrm{int}}] &= \Pr[\mathsf{G}_2 : \mathsf{Find} \wedge \neg\mathsf{Coll}_{\mathrm{int}}] \\ &\le \Pr[\mathsf{G}_2 : \mathsf{Find}].\end{aligned}$$

Thus, it remains to bound the extraction probability in Game 2.

We construct a PGOW adversary $\mathcal{A}_{\mathrm{pgow}}$ by simulating Game 2 for $\mathcal{A}$:

1. It first runs $\mathcal{A}_0(\theta, y)$, aborts if the first $\lambda - 12$ bits of two distinct salts collide.
2. It generates the simulated transcripts as in Game 2
3. It runs the punctured execution of Game 2 with these transcripts. Before answering each oracle query, it tests membership in $\mathcal{S}$ as in Lemma E.1, using f and the PGOW guessing oracle.
4. When Find occurs, it measures the corresponding query input ξ, outputs $f(\xi)$, and stops without answering that query. If Find never occurs, it outputs $\perp$.

[12] In previous versions, we incorrectly assumed that this equality also held for the extractor supplied by the double-sided O2H lemma [BHH+19].

Whenever Find occurs, the measured input ξ lies in S. Since $f(S) = \{\delta\}$, the adversary outputs $f(\xi) = \delta = \mathsf{FirstBits}_\lambda(x)$ and wins the PGOW game. Therefore,

$$\Pr[\mathsf{G}_2 : \mathsf{Find}] \le \mathsf{Adv}^{\mathrm{pgow}}_{\mathsf{F}_{\mathrm{MQOM}},\mathcal{A}_{\mathrm{pgow}}}(\lambda).$$

It follows that

$$|\Pr[\mathsf{G}_0 : W_0] - \Pr[\mathsf{G}_2 : W_2]| \le 2\sqrt{Q_{\mathsf{H}_{\mathrm{avc}}}\,\mathsf{Adv}^{\mathrm{pgow}}_{\mathsf{F}_{\mathrm{MQOM}},\mathcal{A}_{\mathrm{pgow}}}(\lambda)} + 2\Pr[\delta = 0^\lambda] + \frac{2Q\tau(2N-d-3)}{2^\lambda},$$

where $\mathcal{A}_{\mathrm{pgow}}$ makes at most $2Q_{\mathsf{H}_{\mathrm{avc}}}$ quantum queries to its guessing oracle: two per membership test, one to compute the guessing predicate and one to uncompute it. This proves the QROM bound and completes the proof. □

F Examples of H-Coefficient Technique with One-wayness

Let $E : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ be an ideal cipher. For a function $\psi : \{0,1\}^\lambda \to \{0,1\}^\lambda$, we define $\bar{\psi} : \{0,1\}^\lambda \to \{0,1\}^\lambda$ by setting $\bar{\psi}(s) := \psi(s) \oplus s$. We say that ψ is an *orthomorphism* if both ψ and $\bar{\psi}$ are permutations. In what follows, we assume that both ψ and $\bar{\psi}$ are linear and efficiently invertible.

Let us consider the TCCR hash candidate $\mathsf{H} = \mathsf{EncFF}$ defined by

$$\mathsf{H}(s,t) = \mathsf{EncFF}(s,t) := E(t,s) \oplus \psi(s)$$

as in [BBFR24], which is a variant of the Davies-Meyer transform. [13]

For $x \in \{0,1\}^\lambda$, we define an oracle $\mathcal{O}^{\mathrm{TCCR}}_x$ as

$$\mathcal{O}^{\mathrm{TCCR}}_x(s,t,b) := \mathsf{EncFF}(s \oplus x, t) \oplus b \cdot x = E(t, s \oplus x) \oplus \psi(s \oplus x) \oplus b \cdot x.$$

We want to show that the following games are computationally indistinguishable:

$$\begin{aligned}
\mathsf{Real} :& (f, f(x), x) \leftarrow \mathsf{Gen}(1^\lambda), E \leftarrow \mathsf{IC}_\lambda;\\
& b \leftarrow \mathcal{A}^{E^\pm, \mathcal{O}^{\mathrm{TCCR}}_x}(f, f(x)),\ \text{output } b\\
\mathsf{Ideal} :& (f, f(x), x) \leftarrow \mathsf{Gen}(1^\lambda), E \leftarrow \mathsf{IC}_\lambda, \rho \leftarrow \mathsf{Func}(\{0,1\}^{2\lambda+1}, \{0,1\}^\lambda);\\
& b \leftarrow \mathcal{A}^{E^\pm, \rho}(f, f(x)),\ \text{output } b,
\end{aligned}$$

where the distinguisher is required *not* to query both $(s,t,0)$ and $(s,t,1)$ to its oracle for any $(s,t) \in \{0,1\}^{2\lambda}$.

We modify the security proof of TCCR hash for $\widehat{\mathsf{MMO}}^E(s,t) := E(t, \psi(s)) \oplus \psi(s)$ in Guo et al. [GKWY20]. While Guo et al. [GKW+20] showed the multi-instance TCCR (miTCCR) property, where an adversary is given to multiple oracles for $x_0, \ldots, x_{u-1}$, we only need TCCR because the secret x is shared among all signatures. In other words, when we consider the multi-use EUF-CMA security, we need to consider the miTCCR property of EncFF.

Theorem F.1 (TCCR with one-wayness). *Let $\mathcal{A}$ be a PPT distinguisher that queries to the oracle $\mathcal{O}$ and $E^\pm$ at most $Q_{\mathcal{O}}$-times and Q_E-times. There exists an adversary $\mathcal{A}_{\mathrm{ow}}$ satisfying*

$$\left| \Pr_{E,\mathsf{Gen},\mathcal{A}}[\mathsf{Real} = 1] - \Pr_{E,\rho,\mathsf{Gen},\mathcal{A}}[\mathsf{Ideal} = 1] \right| \le \mathsf{Adv}^{\mathrm{ow}}_{\mathsf{Gen},\mathcal{A}_{\mathrm{ow}}}(\lambda) + Q_{\mathcal{O}}^2/2^\lambda.$$

The running time of $\mathcal{A}_{\mathrm{ow}}$ is that of $\mathcal{A}$ plus $Q_{\mathcal{O}}Q_E \cdot \mathsf{poly}(\lambda)$.

Proof (proof of theorem). By fixing the random tape of $\mathcal{A}$ and averaging over it, it suffices to prove the claim for deterministic distinguishers. Hence, fix an arbitrary deterministic distinguisher $\mathcal{A}$ and show that

$$\left| \Pr_{E,\mathsf{Gen}}[\mathsf{Real} = 1] - \Pr_{E,\rho,\mathsf{Gen}}[\mathsf{Ideal} = 1] \right| \le \mathsf{Adv}^{\mathrm{ow}}_{\mathsf{Gen},\mathcal{A}_{\mathrm{ow}}}(\lambda) + Q_{\mathcal{O}}^2/2^\lambda.$$

[13] The original Davies-Meyer transform converts $E : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ to $\mathsf{DM}^E(s,t) = E(t,s) \oplus s$.

Transcripts: In the real experiment, the oracle $\mathcal{O}_x^{\mathrm{TCCR}}$ is defined by

$$\mathcal{O}_x^{\mathrm{TCCR}}(s,t,b) \coloneqq \mathsf{EncFF}(s \oplus x, t) \oplus b \cdot x = E(t, s \oplus x) \oplus \psi(s \oplus x) \oplus b \cdot x.$$

In the ideal experiment, $\mathcal{A}$ instead has access to a random function $\rho \leftarrow \mathsf{Func}(\{0,1\}^{2\lambda+1}, \{0,1\}^{\lambda})$.

We record the interaction of $\mathcal{A}$ with its oracles by a transcript $Q \coloneqq (Q_{\mathcal{O}}, Q_E, x)$, where x is the secret sampled by Gen. Here, $Q_{\mathcal{O}}$ stores all query–answer tuples (s,t,b,z) made to the $\mathcal{O}$-oracle, and Q_E stores all ideal-cipher query–answer tuples under $E^{\pm}$. For convenience, we represent Q_E as a set of triples (t,u,v) meaning that $E(t,u) = v$. We only consider attainable transcripts induced by $\mathcal{A}$.

For each $t \in \{0,1\}^{\lambda}$, let $Q_E[t] \coloneqq \{(u,v) : (t,u,v) \in Q_E\}$ denote the set of cipher relations under key t. We also define $Q_{\mathcal{O}}[t] \coloneqq \{(s,b,z) : (s,t,b,z) \in Q_{\mathcal{O}}\}$.

Bad transcripts: We define the following subsets of bad transcripts:

- Bad_1: There exists a query $(s,t,b,z) \in Q_{\mathcal{O}}$ and a query $(t, s \oplus x, *)$ or $(t, *, \psi(s \oplus x) \oplus b \cdot x \oplus z)$ in Q_E.
- Bad_2: There exist two distinct tuples $(s_i,t_i,b_i,z_i) \neq (s_j,t_j,b_j,z_j)$ in $Q_{\mathcal{O}}$ such that

$$t_i = t_j \quad \text{and} \quad \psi(s_i) \oplus b_i \cdot x \oplus z_i = \psi(s_j) \oplus b_j \cdot x \oplus z_j.$$

Note that Bad_2 forces a collision among the (same-key) outputs of the permutation $E_{t_i}(\cdot)$ on the internal inputs $s_i \oplus x$ and $s_j \oplus x$.

Following [GKWY20, GKW+20], let us bound ϵ_{bad}, the probabilities that a transcript in the ideal world is in those two bad sets.

Bounding $\Pr[\Theta_{\mathrm{ideal}} \in \mathsf{Bad}_1]$ *(one-wayness reduction):* In [GKWY20, GKW+20], Guo et al. fixed some $(s,t,b,z) \in Q_{\mathcal{O}}$ and showed that $\Pr[(t, s \oplus x, *) \in Q_E] \leq |Q_E[t]|/2^{\rho}$, where ρ is the min-entropy of x. They conclude that $\Pr[\Theta_{\mathrm{ideal}} \in \mathsf{Bad}_1] \leq 2\mu p/2^{\rho}$ in [GKW+20, Theorem 2] or $2qp/2^{\rho}$ in [GKWY20, Theorem 5] (we skip the definition of p, q, μ, etc). Unfortunately, our setting degrades the min-entropy of x to 0 in the worst case since the distinguisher is given $(f, f(x))$. Instead, we show that if this event occurs, then one can find x' violating one-wayness.

Concretely, we can write a reduction algorithm $\mathcal{A}_{\mathrm{ow}}$ as follows:

1. Receive $(f, f(x))$. Initialize two tables $Q_{\mathcal{O}}$ and Q_E.
2. Run $\mathcal{A}$ and simulate the oracles as follows:
   - $\rho(s,t,b)$: This is implemented by lazy sampling with a table $Q_{\mathcal{O}}$. If $(s,t,b,z) \in Q_{\mathcal{O}}$, then answer z. If $(s,t,1-b,z) \in Q_{\mathcal{O}}$, then abort. Otherwise, sample $z \leftarrow \{0,1\}^{\lambda}$, store (s,t,b,z) in $Q_{\mathcal{O}}$. Then perform the following checks for Bad_1:
     - For every $(u,v) \in Q_E[t]$, compute the candidate $x' \coloneqq s \oplus u$. If $f(x') = f(x)$, then output x' and halt.
     - For every $(u,v) \in Q_E[t]$, compute $w \coloneqq v \oplus z \oplus \psi(s)$. If $b = 0$ and $f(\psi^{-1}(w)) = f(x)$, then output $x' \coloneqq \psi^{-1}(w)$ and halt. If $b = 1$ and $f(\bar{\psi}^{-1}(w)) = f(x)$, then output $x' \coloneqq \bar{\psi}^{-1}(w)$ and halt.

     If not, return z to $\mathcal{A}$.
   - $E^{+}(t,u)$: This is implemented by lazy sampling with table Q_E. If $(t,u,v) \in Q_E$, then answer v. Otherwise, sample v uniformly from the outputs not yet used in $Q_E[t]$ and store (t,u,v) in Q_E. Then pervform the following checks for Bad_1:
     - For every $(s,b,z) \in Q_{\mathcal{O}}[t]$, compute $x' \coloneqq s \oplus u$. If $f(x') = f(x)$, then output x' and halt.
     - For every $(s,b,z) \in Q_{\mathcal{O}}[t]$, compute $w \coloneqq v \oplus z \oplus \psi(s)$. If $b = 0$ and $f(\psi^{-1}(w)) = f(x)$, then output $x' \coloneqq \psi^{-1}(w)$ and halt. If $b = 1$ and $f(\bar{\psi}^{-1}(w)) = f(x)$, then output $x' \coloneqq \bar{\psi}^{-1}(w)$ and halt.

     If not, return v to $\mathcal{A}$.
   - $E^{-}(t,v)$: This is implemented by lazy sampling with table Q_E. If $(t,u,v) \in Q_E$, then answer u. Otherwise, sample u uniformly from the inputs not yet used in $Q_E[t]$ and store (t,u,v) in Q_E. Then perform the same two checks as in the $E^{+}(t,u)$ case. If either check succeeds, output the corresponding candidate x' and halt. Otherwise, return u to $\mathcal{A}$.

If a transcript belongs to Bad_1, then the reduction $\mathcal{A}_{\mathrm{ow}}$ outputs a preimage x' of $f(x)$. To see this, consider the following two cases. Suppose that there exist $(s,t,b,z) \in Q_{\mathcal{O}}$ and $(t,u,v) \in Q_E$ satisfying $u = s \oplus x$. By our definition, the first check correctly detects this event and outputs $x' = s \oplus u$ after checking $f(x') = f(x)$. Next, suppose that there exist $(s,t,b,z) \in Q_{\mathcal{O}}$ and $(t,u,v) \in Q_E$ satisfying $v = \psi(s \oplus x) \oplus b \cdot x \oplus z$, which implies $\psi(x) \oplus b \cdot x = v \oplus z \oplus \psi(s) =: w$. Then, the second check correctly detects this event and outputs $x' = \psi^{-1}(w)$ or $\bar{\psi}^{-1}(w)$, depending on the value of b, after checking $f(x') = f(x)$. Hence, we have

$$\Pr[\Theta_{\mathrm{ideal}} \in \mathsf{Bad}_1] \leq \mathsf{Adv}^{\mathrm{ow}}_{\mathsf{Gen},\mathcal{A}_{\mathrm{ow}}}(1^{\lambda}).$$

Bounding $\Pr[\Theta_{\mathrm{ideal}} \in \mathsf{Bad}_2]$ *(collision probability):* This part is the same as that in [GKWY20]. Let us fix $t \in \{0,1\}^\lambda$. We consider two distinct queries (s,t,b,z) and $(s',t,b',z') \in \mathcal{Q}_{\mathcal{O}}$. We first claim that $\psi(s \oplus x) \neq \psi(s' \oplus x)$. To show it, we assume that $\psi(s \oplus x) = \psi(s' \oplus x)$. If so, then $s = s'$ since ψ is a linear orthomorphism. However, if $s = s'$, then the two queries are (s,t,b,z) and (s,t,b',z'). Since we restrict the distinguisher's query, b should be equal to b'. But, this means $z = z'$, and this contradicts our assumption, two *distinct* queries. We then have

$$\begin{aligned}&\Pr[\psi(s \oplus x) \oplus b \cdot x \oplus z = \psi(s' \oplus x) \oplus b' \cdot x \oplus z'] \\ &= \Pr[\psi(s) \oplus \psi(s') \oplus (b \oplus b')x = z \oplus z'] \leq 2^{-\lambda},\end{aligned}$$

since z and z' are chosen uniformly at random from $\{0,1\}^\lambda$. Thus, fixing t, for any pair of distinct queries in $\mathcal{Q}_{\mathcal{O}}$, the probability that a transcript is in Bad_2 is at most $1/2^\lambda$. Considering all queries, we have

$$\Pr[\Theta_{\mathrm{ideal}} \in \mathsf{Bad}_2] \leq \sum_t \binom{|\mathcal{Q}_{\mathcal{O}}[t]|}{2} \cdot 2^{-\lambda} \leq Q_{\mathcal{O}}^2/2^\lambda.$$

Likelihood ratio on good transcripts: We then fix good transcripts $\mathsf{Good} := \mathcal{T} \setminus (\mathsf{Bad}_1 \cup \mathsf{Bad}_2)$. Following the arguments in [GKWY20, GKW+20], we have that $\epsilon_{\mathrm{good}} = 0$.

Conclusion via the H-coefficient technique: Applying the H-coefficient technique (Theorem 4.1), we obtain that there exists a PPT machine $\mathcal{A}_{\mathrm{ow}}$ satisfying

$$\begin{aligned}|\Pr[\mathsf{Real} = 1] - \Pr[\mathsf{Ideal} = 1]| &\leq \epsilon_{\mathrm{good}} + \epsilon_{\mathrm{bad}} \\ &\leq 0 + \mathsf{Adv}^{\mathrm{ow}}_{\mathsf{Gen},\mathcal{A}_{\mathrm{ow}}}(\lambda) + Q_{\mathcal{O}}^2/2^\lambda\end{aligned}$$

as we wanted. □