On the Security of MPC-in-the-Head Signatures with Correlated GGM Trees

Thibauld Feneuil and Matthieu Rivain

CryptoExperts, Paris, France

Abstract. Recent MPC-in-the-Head techniques enable the construction of signature schemes with compact signature sizes from various hardness assumptions. These techniques rely on commitments based on GGM trees, which have been optimized to further reduce the signature size with the so-called *one-tree* or *correlated tree* optimizations. While the one-tree technique has no incidence on the security of the scheme, this is not obvious for the correlated tree technique, and a formal security analysis of this technique has been missing in the literature.

In this work, we fill this gap and provide the first formal security analysis of MPC-in-the-Head signature schemes based on correlated trees. We first exhibit a potential security flaw of this technique which rules out any hope for a security reduction to the underlying hardness assumption. In particular, we show that recovering the first λ bits of the secret witness is sufficient to achieve a full key recovery (where λ is the security level). The underlying assumption should hence be such that recovering these λ bits is as hard as recovering the full witness. Some state-of-the-art schemes do not satisfy this condition, which prevents a direct application of the correlated tree technique.

We then provide a formal security proof for signature schemes based on the correlated tree technique under this degraded hardness assumption. Our proof comes in several variants, in the random oracle mode or in the ideal cipher model, depending on the specific correlated tree construction. We also introduce a tweak for the instantiation of the leaf seed expansion in the ideal cipher model, which allows us to achieve a tighter security reduction. Our result shows that MPC-in-the-Head signatures based on correlated trees can achieve strong security guarantees and provides the first formal security proof for the MQOM v2 signature scheme, the on-going candidate in the NIST post-quantum standardization process with the shortest signature size in the MPC-in-the-Head family.

Keywords: Post-Quantum Signatures · Security Proofs · MPC-in-the-Head · Correlated GGM Trees · H-Coefficient Technique

1 Introduction

The MPC-in-the-Head (MPCitH) paradigm [IKOS07] provides a flexible approach for designing signature schemes based on a wide range of security assumptions. This is illustrated by the presence of six MPCitH schemes among the fourteen Round-2 candidates in the NIST process for standardizing additional post-quantum signature schemes [ABC+24]. Recent frameworks specializing this paradigm [BBD+23, FR25] achieve signature sizes below 5 kilobytes while relying on conservative security assumptions, such as the security of AES [BBB+24], the hardness of solving unstructured multivariate quadratic systems [BBFR24], or the hardness of solving unstructured syndrome decoding problems [ABB+24b].

These constructions heavily rely on GGM seed trees [GGM86, KKW18]. A significant portion of the signature size is devoted to sibling paths in these trees. To reduce this overhead, two optimization techniques have been introduced. The *one-tree technique* [BBM+24] combines several GGM trees into a single tree in order to merge and hence shorten the revealed sibling paths. This optimization is purely combinatorial and does not affect the security of the scheme. It can therefore be applied to essentially all recent MPCitH constructions. The *correlated tree* technique consists in keeping an XOR-invariant across the tree [GYW+23, HJ24]. More precisely, the tree is constructed such that at each layer, the XOR of all the nodes equals the first λ bits of the secret witness. By tweaking the definition of the expanded tapes from the leaf seeds, this structure allows to reduce the impact of the GGM tree in the signature size of λ bits for each of the τ

repetitions, which yields a valuable reduction. Compared to the one-tree technique, correlated trees further offer certain practical advantages from an implementation perspective [BF26].

However, the correlated tree technique has never been formally proven secure in the context of MPC-in-the-Head signatures. Although a recent work [KLS25] introduces a scheme based on this technique and provides a security proof, a subsequent paper [KX25] identifies a flaw in that proof arising precisely from the use of correlated trees. Consequently, the authors of [KLS25] later released a revised version of their scheme that no longer relies on this technique. At present, no concrete attack against the correlated tree optimization is known, and no complete security proof has been established either. As a result, the security guarantees of the NIST candidate MQOM (version 2) [BBFR24], which relies on this technique, remain unclear.

Our contributions. In this work, we provide a thorough security analysis of the correlated tree technique for MPC-in-the-Head signatures. We emphasize that this technique does not affect existential unforgeability under key-only attacks (EUF-KO). However, it does impact existential unforgeability under chosen-message attacks (EUF-CMA). Indeed, the technique introduces additional structure in the scheme's randomness, making signatures potentially more prone to leaking information about the secret key.

We first exhibit a potential security flaw of this technique which rules out any hope for a security reduction to the underlying hardness assumption. Indeed, if an adversary can recover the λ first bits of the witness, then they learn the root of the tree and can reconstruct the entire tree, ultimately recovering the full witness (i.e., the full secret key). It results that the security of signatures based on correlated trees can only rely on the degraded assumption that recovering the first λ bits of the witness is as hard as recovering the full witness. We formalize this property as the *partial guessing one-wayness* (PGOW) of the underlying problem. While this assumption appears to hold for many MPCitH schemes, it is not always satisfied. For instance, the SDitH v2 signature scheme [ABB+24b], the min-entropy of these λ bits is strictly smaller than λ. Consequently, using correlated trees in such a setting would significantly reduce the security below λ bits.

We then provide formal security proofs that correlated tree commitments and openings do not leak information about the secret key, under suitable assumptions. Our analysis formalize the *correlated tree oracle* capturing the leakage provided by signature queries and show how to simulate this oracle without access to the secret key using the H-Coefficient technique [Pat09, CS14]. Our analysis considers several instantiations of the symmetric primitives used to define the tree. We begin with the random oracle model, assuming that seed derivation, seed commitment, and seed expansion primitives are instantiated using hash functions or extendable-output functions. In this setting, we obtain a clean and conceptually simple proof under the PGOW assumption. However, hash-based instantiations (*e.g.*, SHA3/SHAKE) are computationally expensive, and recent MPCitH schemes typically instead rely on AES, which enjoys hardware acceleration on modern processors. We therefore analyze security in the ideal cipher model (the ideal assumption capturing a block cipher like AES). First, we consider an instantiation based on the Davies-Meyer mode. We show that security holds only if the underlying one-way function satisfies an additional ad-hoc property which we call *relation guessing one-wayness* (RGOW), generalizing the PGOW assumption in a way that depends on the underlying signature scheme. Next, we study the use of a block cipher in counter mode. In this case, due to the way domain separation is implemented, the resulting proof is non-tight. Finally, we tweak the Davies-Meyer-based construction to obtain a tight security proof that does not rely on the additional ad-hoc assumption (though still requiring the hardness of recovering the first λ bits of the witness). Altogether, these four variants of our security proof demonstrate that correlated trees can be proven secure but only with careful instantiations and a degraded hardness assumption.

Finally, we show how the simulation security of the introduced correlated tree oracle can be leveraged to establish the existential unforgeability under chosen-message attacks for MPC-in-the-Head signatures based on correlated trees. We apply this result to the NIST candidate MQOM v2 [BBFR24], an MPC-in-the-Head signature scheme based on the hardness of solving unstructured multivariate quadratic systems (MQ). In its current version, this scheme uses correlated trees with the Davies-Meyer instantiation, and its EUF-CMA security therefore relies on the PGOW and RGOW assumptions for MQ. Our analysis shows that while PGOW seems essentially as hard as the standard MQ problem, RGOW opens the door to some attack degrading the security by a few bits. Using our proposed enhanced Davies-Meyer instantiation, we can avoid

the RGOW assumption and hence rule out this attack path, albeit at the cost of a small increase in the number of AES calls.

Concurrent work. [KX26] studies the security of correlated trees in MPCitH schemes through tweaked variants of MQOM v2. Excluding their QROM results, our analysis applies to a broader range of settings, does not rely on substantial design modifications, and achieves tighter security bounds. A more detailed comparison is provided in Appendix A.

2 MPC-in-the-Head Signatures from Correlated Trees

A GGM tree [GGM86] consists in pseudorandomly expanding a root seed rseed into N leaf seeds $\mathtt{lseed[0]}$, $\ldots$, $\mathtt{lseed}[N-1]$ using a binary tree:

$$\begin{cases} \mathsf{node}[1] = \mathsf{rseed} \\ (\mathsf{node}[2i], \mathsf{node}[2i+1]) = G(\mathsf{node}[i]) \quad \text{for } 1 \leq i \leq N-1 \end{cases} \tag{1}$$

where G is a seed derivation function. The leaf seeds are then defined as the last N nodes, namely $\mathtt{lseed}[i] := \mathsf{node}[N+i]$ for all $i \in [0, N-1]$, assuming that N is a power of two for the sake of simplicity. We denote $\ell := \log_2(N)$ the height of the tree.

The GGM tree structure enables to reveal all-but-one leaf seeds from ℓ nodes of the tree. Let $i^* \in [0, N-1]$ be the index of the leaf seed that should remain hidden. This means that all its ancestors, whose indexes are in the set $\mathcal{H} = \{\lfloor (N+i^*)/2^j \rfloor \mid 0 \leq j \leq \ell - 1\}$, should remain hidden. On the other hand, the *sibling path*, which is made of the siblings of the hidden nodes, can be revealed without giving any information on $\mathtt{lseed}[i^*]$. From the binary tree structure, these are the nodes in $\{\mathsf{node}[i \oplus 1] \mid i \in \mathcal{H}\}$. By running the tree derivation of Equation (1) from all the seeds in the sibling path, one reconstructs a partial tree composed of all the nodes but the hidden path. In particular, this partial tree includes all the leaves but the hidden leaf $\mathtt{lseed}[i^*]$.

GGM seed trees have been introduced in the context of MPC-in-the-Head to commit additive secret sharings with efficient opening of all-but-one shares [KKW18]. From a random root seed rseed, one expands a GGM seed tree to obtain the leaf seeds $\mathtt{lseed[0]}$, $\ldots$, $\mathtt{lseed}[N-1]$. Then, each leaf seed is expanded into a share: $[\![x]\!]_i = \mathsf{PRG}(\mathtt{lseed}[i]) \in \mathbb{F}^n$ and a auxiliary value Δ_x is defined as:

$$\Delta_x = x - \sum_{i=0}^{N-1} [\![x]\!]_i \ . \tag{2}$$

By definition, $([\![x]\!]_0 + \Delta_x)$, $[\![x]\!]_1$, $\ldots$, $[\![x]\!]_{N-1}$ forms an additive secret sharing of x. This additive sharing is committed by deriving a commitment for each seed: $\mathtt{ls_com}[i] = \mathsf{SeedCommit}(\mathtt{lseed}[i])$, and sending a global commitment: $\mathsf{com} = \mathsf{Commit}(\mathtt{ls_com[0]}, \ldots, \mathtt{ls_com}[N-1], \Delta_x)$ to the verifier. Later on, the verifier can challenge the prover to open all the additive shares of x but one, say the share of index i^*. The prover then reveals the sibling path of $\mathtt{lseed}[i^*]$, the auxiliary value Δ_x and the commitment $\mathtt{ls_com}[i^*]$. From all the leaf seeds (but $\mathtt{lseed}[i^*]$), the verifier can expand all the shares $[\![x]\!]_i$ (but $[\![x]\!]_{i^*}$) and correct $[\![x]\!]_0$ with Δ_x. The verifier can also recompute the commitments $\mathtt{ls_com}[i]$, for all $i \neq i^*$, and together with $\mathtt{ls_com}[i^*]$ and Δ_x, recompute and check the global commitment.

Correlated GGM Trees. The *correlated tree* technique [GYW+23] consists in keeping an invariant across all the layers of the tree: the XOR of all the nodes of a given depth is equal to a fixed value δ. This is ensured by defining:

$$\mathsf{node}[2] \leftarrow \{0,1\}^\lambda \quad \text{and} \quad \mathsf{node}[3] = \mathsf{node}[2] \oplus \delta \ ,$$

and then deriving the rest of the tree as in Equation (1) with a special instantiation of the seed derivation function G of the form:

$$G : x \mapsto (H(x), H(x) \oplus x) \ ,$$

where H is a hash function or a pseudorandom function. With this definition of G, the two child nodes $\mathsf{node}[2k]$ and $\mathsf{node}[2k+1]$ of $\mathsf{node}[k]$ with $k \geq 2$ satisfy the relation:

$$\mathsf{node}[k] = \mathsf{node}[2k] \oplus \mathsf{node}[2k+1] \ .$$

This way, for all $j \in [1, \ell]$, the XOR of all the nodes of depth j is equal to the XOR of all the nodes of depth $j-1$ (their parents) which, by induction, is equal to $\mathsf{node}[2] \oplus \mathsf{node}[3] = \delta$.

In the context of MPC-in-the-Head signatures, the correlated tree technique can be used to reduce the signature size. The principle of this optimization proposed in [HJ24] is to define the XOR-invariant δ as the λ first bits of the secret witness x. Each leaf seed $\mathtt{lseed}[i]$ is then expanded as

$$[\![x]\!]_i = \mathtt{lseed}[i] \parallel \mathrm{PRG}(\mathtt{lseed}[i])$$

namely by prepending the seed itself to the expanded share $[\![x]\!]_i$. Doing so, the first λ bits of the auxiliary value Δ_x defined in Equation (2) satisfy:

$$\mathsf{First}_\lambda(\Delta_x) = \mathsf{First}_\lambda(x) - \mathsf{First}_\lambda\Big(\sum_{i=0}^{N-1} \mathtt{lseed}[i]\Big) = 0 \ .$$

As a result, one can discard the first λ bits of Δ_x in the commitment, which saves λ bits per GGM tree in the signature. Since the signature typically includes τ GGM trees, with τ ranging between 12 and 16, this optimization saves up to 192–256 bytes in the signature for $\lambda = 128$.

On domain separation. While we did not explicitly mention it in the above description, the seed derivation function H is typically tweaked with a domain separator which depends on a salt value salt randomly generated for each signature. In addition, H is further tweaked using the depth of the node in the tree. The principle is the following: we should avoid having different hidden nodes $\{h\}$ (over the same or different signatures) entering the same instance of the function H with revealed outputs $\{r\}$. This is because in that case, the adversary could exhaustively search for h such that $H(h) \in \{r\}$ and recover a solution with an average of $2^\lambda/|\{r\}|$ trials. Since revealing any hidden node enable a full key recovery, such an attack effectively lowers the security of the scheme by $\log_2(|\{r\}|)$ bits. If the same function H is involved in the seed commitment and/or expansion (which is common), these different uses should also be domain-separated for the same reason.

Modern MPC-in-the-Head Signatures. Recent MPC-in-the-Head signature schemes (see e.g. [FR25, BBD+23,HJ24,KLS25]), including those selected for Round 2 of the ongoing NIST process, rely on VOLE-in-the-Head [BBD+23] or its variant Threshold-Computation-in-the-Head [FR25]. In both frameworks, GGM trees are involved to build a *line commitment scheme* (LCS), in which the prover can commit to a degree-one polynomial P_x and later open an evaluation $P_x(\omega)$ without revealing further information. The line commitment scheme is then used as a building block in a specific protocol named a *Polynomial Interactive Oracle Proof* (PIOP) in which the prover sends a polynomial oracle to the verifier and later opens some evaluations of this oracle. We provide a general overview of these constructions hereafter.

Let $\mathbb{F}$ be the finite field in which the secret witness x is defined. We denote $\mathbb{K}$ an extension of $\mathbb{F}$ of size $|\mathbb{K}| \geq N$ (where N is the size of the GGM tree) and $\mathbb{K}[X]$ the set of univariate polynomials over $\mathbb{K}$. For a polynomial $P \in \mathbb{K}[X]$, we further denote $P(\infty)$ its leading coefficient. In the following, we assume that $\mathbb{F}$ and $\mathbb{K}$ are binary fields. We usually denote the XOR operation on bitstrings by $\oplus$, and the addition in $\mathbb{F}$ and $\mathbb{K}$ by $+$. However, these notations may occasionally be used interchangeably in contexts where sequences of field elements are interpreted as bitstrings, or conversely, where bitstrings are viewed as sequences of field elements.

Witness-checking function. The considered MPCitH signature schemes rely on (families of) one-way functions. Consider a scheme based on a family $\mathcal{F}$ of functions $f : \bar{\mathcal{X}} \to \bar{\mathcal{Y}}$. Generating a key pair consists in sampling a random function f from $\mathcal{F}$, a random input $\bar{x} \in \bar{\mathcal{X}}$ and computing the output $\bar{y} = f(\bar{x})$. The secret key is then defined as $\bar{x}$ while the public key is $(f, \bar{y})$. The signature is a zero-knowledge proof of knowledge of $\bar{x}$ such that $f(\bar{x}) = \bar{y}$ made non-interactive (and tweaked with input message) using the Fiat-Shamir transform. For this purpose, one relies on an *arithmetization* of the verification of $f(\bar{x}) = \bar{y}$. This arithmetization is typically defined as a system of multivariate polynomial equations:

$$F_y(x) = 0 \quad \Leftrightarrow \quad f(\bar{x}) = \bar{y}$$

where F_y is a vector of m polynomial functions over $\mathbb{F}$ and x is a *secret witness* derived from $\bar{x}$. Here we denote y the *public statement* encoding the function f and the output $\bar{y}$. The pair (x, y) is called a *witness-statement instance* of the family $\mathcal{F}$ and we denote $\mathcal{X} \times \mathcal{Y}$ its definition domain, with $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. The signature is then defined as a proof of knowledge of a solution $x \in \mathcal{X}$ of $F_y(x) = 0$. We call F_y the *witness checking* function and stress that finding a preimage of 0 for F_y is as hard as inverting f on $\bar{y}$, which is the underlying hardness assumption of the scheme.

In the following, we denote $\mathcal{D}_{\mathcal{F}}$ the distribution of witness-statement instances of $\mathcal{F}$. In other words, $(x, y) \leftarrow \mathcal{D}_{\mathcal{F}}$ denotes the process of sampling a random function f from $\mathcal{F}$, a random input $\bar{x}$, computing $\bar{y} = f(\bar{x})$, deriving $y \in \mathcal{Y}$ from $(f, \bar{y})$ and deriving $x \in \mathcal{X}$ from $\bar{x}$, which satisfy $F_y(x) = 0$.

Example 1. In FAEST [BBB+24], the family $\mathcal{F}$ is composed of functions mapping $\bar{x}$ to the AES encryption of a fixed plaintext under the key $\bar{x}$. The witness x extends $\bar{x}$ with some auxiliary values derived from the underlying AES encryption. In MQOM [BBFR24], the family $\mathcal{F}$ is composed of random multivariate quadratic functions. The witness x is simply $\bar{x}$, which is a solution of the MQ system defined by $f(\bar{x}) = \bar{y}$ and the function F_y is simply defined as $x \mapsto f(x) - \bar{y}$. In both these examples, the λ first bits of x are the same as the λ first bits of $\bar{x}$, which are uniformly distributed over $\{0,1\}^\lambda$. This is a crucial property for the security of the correlated tree technique as we will see.

Along the paper, we will implicitly consider that a family of one-way functions $\mathcal{F}$ comes associated with a witness checking and underlying distribution $\mathcal{D}_{\mathcal{F}}$ as defined above.

Line commitment scheme. The committed degree-1 polynomial (or line) is defined from the expanded leaf seeds of the GGM tree as:

$$P_x(X) = \Delta_x \cdot P_0(X) + \sum_{i=0}^{N-1} [\![x]\!]_i \cdot P_i(X) \ \in \mathbb{K}[X] \tag{3}$$

where $P_0, \ldots, P_{N-1} \in \mathbb{K}[X]$ are the unique degree-1 polynomials satisfying $P_i(\omega_i) = 0$ and $P_i(\infty) = 1$ for a fixed evaluation set $\Omega = \{\omega_0, \ldots, \omega_{N-1}\} \subseteq \mathbb{K}$. From this definition, we have that P_x encodes the secret x by:

$$P_x(\infty) = \Delta_x + \sum_{i=0}^{N-1} [\![x]\!]_i = x \ .$$

Moreover, the evaluation of P_x in a point $\omega_{i^*} \in \Omega$ can be derived from all the shares $[\![x]\!]_i$ but $[\![x]\!]_{i^*}$ by:

$$P_x(\omega_{i^*}) = \Delta_x \cdot P_0(\omega_{i^*}) + \sum_{i=0}^{N-1} [\![x]\!]_i \cdot P_i(\omega_{i^*}) = \Delta_x \cdot P_0(\omega_{i^*}) + \sum_{i \neq i^*} [\![x]\!]_i \cdot P_i(\omega_{i^*})$$

where the above equality holds because $P_{i^*}(\omega_{i^*}) = 0$. Since any other evaluation of P_x would require the knowledge of $[\![x]\!]_{i^*}$, the verifier only learns $P_x(\omega_{i^*})$ from the all-but-one opening.

Polynomial IOP. Besides the line commitment scheme, MPC-in-the-Head signatures also rely on a specific protocol named a *Polynomial Interactive Oracle Proof* (PIOP). A PIOP is an interactive proof in which the prover can send a *polynomial oracle* $[P_1, \ldots, P_n]$ to the verifier for polynomials $P_1, \ldots, P_n \in \mathbb{K}[X]$ of prescribed degree d (with $d = 1$ in our setting). From such a polynomial oracle, the verifier can then query some evaluations. In practice, the polynomial oracle are replaced by LCS commitments and the evaluation queries are implemented by LCS openings for the queried evaluations.

The PIOP aims at proving knowledge of the secret witness $x \in \mathbb{F}^n$ satisfying $F_y(x) = (0, \ldots, 0) \in \mathbb{F}^m$, where F_y is the witness-checking function introduced above, which is a vector of m degree-2 functions over $\mathbb{F}$.[1] A simple PIOP proving knowledge of x satisfying $F_y(x) = 0$ is depicted in Figure 1. This protocol is the PIOP equivalent of the QuickSilver protocol [YSWW21] within the VOLE-in-the-Head framework [BBD+23] or the Π_{PC} MPC protocol within the TCitH framework [FR25]. Most of the recent MPC-in-the-Head signatures, including all six on-going MPCitH candidates in the NIST process, rely on this specific PIOP.

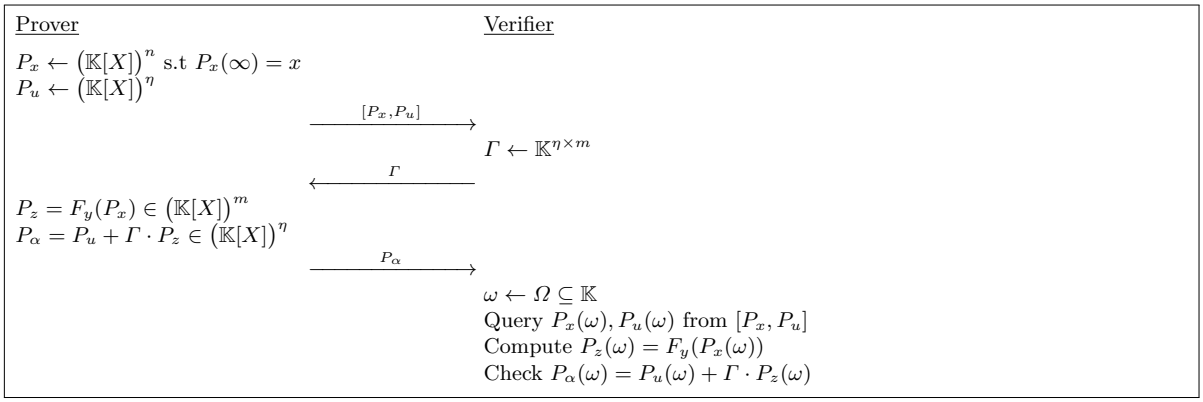

Fig. 1: Polynomial IOP.

In Figure 1, $P_x(\infty)$ denotes the degree-1 coefficient of P_x. The computation of $P_z := F_y(P_x)$ is done by homogenizing F_y. Specifically, for $F_y = (f_1, \ldots, f_m)$ with $f_j(v) := v^T \cdot A_j \cdot v + b_j^T \cdot v + c_j$ for some A_j, b_j and c_j, we define $f_j(P_x(X)) := P_x(X)^T \cdot A_j \cdot P_x(X) + b_j^T \cdot P_x(X) \cdot X + c_j \cdot X^2$, so that the degree-2 coefficient of $f_j(P_x(X))$ equals $f_j(x)$ by definition. We thus get that P_z is of degree 1 if and only if $F_y(P_x(\infty)) = (0, \ldots, 0) \in \mathbb{K}^m$, meaning that $x = P_x(\infty)$ is a valid solution of $F_y(x) = 0$. The soundness of this protocol follows from the Schwartz-Zippel lemma. Assume that $P_\alpha(\omega) = P_u(\omega) + \Gamma \cdot F_y(P_x(\omega))$ holds for 3 different points $r \in \Omega$, then because P_x, P_u, P_α are guaranteed to be of degree 1, we must have $P_\alpha = P_u + \Gamma \cdot F_y(P_x)$ and hence $\Gamma \cdot F_y(x) = 0$. Then we have $F_y(x) = 0$ with overwhelming probability over the random choice of Γ. Conversely, in the presence of a malicious prover (who does not know a correct solution x), we can only have $P_\alpha(\omega) = P_u(\omega) + F_y(P_x(\omega))$ for at most two values ω in Ω. The soundness error of the PIOP is hence of $2/|\Omega|$. In order to amplify this soundness error, the PIOP is typically repeated τ times in parallel, which results in a soundness error of $(2/|\Omega|)^\tau$.[2]

The zero-knowledge property of this protocol holds for two reasons. First, thanks to the addition of the random vector polynomial P_u, the vector polynomial P_α is further uniformly random. Then, any evaluations $P_x(\omega), P_u(\omega)$ are independent of x thanks to the randomness involved in these polynomials.

[1] We note that the PIOP can be generalized to support witness-checking functions of degree greater than 2. However, since degree 2 is the most commonly used in practice, we restrict our exposition to this case for simplicity. Extending the analysis to higher degrees would not affect our results.

[2] While Threshold-Computation-in-the-Head [FR25] uses such parallel repetition approach, VOLE-in-the-Head [BBD+23] relies on a slightly different approach: τ line commitments on a small field $\mathbb{K}$ are combined into a single line commitment on a larger field $\mathbb{K}'$ (of size $\approx |\mathbb{K}|^\tau$), then the PIOP is executed only once with the larger field $\mathbb{K}'$. The obtained soundness is of $2/|\Omega|^\tau$.

Remark 1. In this PIOP, the purpose of the random challenge Γ is to batch the different equations of F_y embedded in P_z into a smaller vector polynomial P_α to save communication. Namely, P_α has η coordinates instead of m, where $\eta < m$ is a security parameter. We note that this challenge can be avoided which amounts to define $\Gamma = \mathrm{Id}_m$ in the PIOP and saves two rounds of interaction. This option is actually implemented in one of the variants of the MQOM v2 signature scheme [BBFR24].

3 Security Issues in MPC-in-the-Head Signatures with Correlated GGM Trees

The security of an MPCitH scheme typically relies on the hardness of finding a witness that satisfies the public constraints encoded in the public key, i.e., finding $x \in \mathbb{F}^n$ such that $F_y(x) = 0$, using the notation of the previous section, which is as hard as solving the underlying hard problem. More precisely, two main security notions are considered for signature schemes:

- The *existential unforgeability under key-only attacks* (EUF-KO) requires that it is computationally infeasible for an adversary to forge a valid signature given only the public key.
- The *existential unforgeability under chosen-message attacks* (EUF-CMA) requires that, given access to a signing oracle replying queries from the adversary, it remains computationally infeasible to forge a valid signature on a new message.

In the MPCitH literature, EUF-KO security is typically established by a reduction to the underlying hard problem. EUF-CMA security is then obtained by completing this reduction with an argument showing that the signatures returned by the signing oracle do not leak useful information about the secret key and can typically be simulated without knowledge of the secret key. This typically relies on the zero-knowledge property of the underlying MPCitH proof system.

While EUF-KO security can still be established using standard techniques in the presence of correlated trees, the situation is more subtle for EUF-CMA security. The correlation introduced by correlated trees partially leaks information about the secret key. In particular, the scheme is susceptible to the attack we introduce below.

3.1 The λ-bit Attack and PGOW Assumption

An adversary may attempt to guess the first λ bits of the witness, denote $\mathsf{First}_\lambda(x)$ hereafter. If the guess is correct, the adversary can recover a hidden node at any level of the tree. For instance, the two siblings at the first level satisfy $\mathtt{node[3]} \oplus \mathtt{node[2]} = \mathsf{First}_\lambda(x)$. Since exactly one of these siblings is known and the other hidden, $\mathsf{First}_\lambda(x)$ directly yields the hidden node. This in turn reveals the full hidden path until the hidden leaf seed $\mathtt{lseed}[i^*]$. From this leaf seed, the adversary can compute the corresponding share as $[\![x]\!]_{i^*} = \mathtt{lseed}[i^*] \parallel \mathrm{PRG}(\mathtt{lseed}[i^*])$ and subsequently recover the full witness as $x = \sum_0^{N-1} [\![x]\!]_i + \Delta_x$.

Partial-Guessing One-Way Security. To prevent such an attack, it must be computationally infeasible to recover the first λ bits of the witness x. We formalize this requirement through a notion that we call *partial-guessing one-way security*, which captures the additional property that the one-way function family (and associated witness-checking function) must satisfy in order for the scheme to remain secure.

Definition 1 (Partial-Guessing One-Way Security). *A one-way function family $\mathcal{F}$ is said to be $(\lambda, t, Q, \varepsilon_{\mathrm{PGOW}}(t,Q))$-PGOW-secure if any t-time adversary $\mathcal{A}$ making Q queries to the guessing oracle has at most $\varepsilon_{\mathrm{PGOW}}(t,Q)$ probability of winning the $(\lambda,\mathcal{F})$-PGOW game described in Figure 2 (i.e. the game returns 1).*

$(\lambda, \mathcal{F})$-PGOW Game:

1. $(x, y) \leftarrow \mathcal{D}_{\mathcal{F}}$
2. Run $\mathcal{A}^{\mathcal{G}(\cdot)}(y)$
3. The game returns 0 ($\mathcal{A}$ loses)

Guessing oracle $\mathcal{G}(\hat{x}_0)$:

1. **if** $\hat{x}_0 = \mathsf{First}_\lambda(x)$ **then** the game returns 1 ($\mathcal{A}$ wins)

Fig. 2: Partial-guessing one-way (PGOW) security game.

PGOW (In)security of Certain OWFs. Currently, the only NIST candidate relying on correlated trees is the Round-2 version of MQOM. One can observe that the MQ problem, with its natural arithmetization as used in MQOM (see Example 1), satisfies the partial-guessing one-way property defined above. Indeed, the hardness of solving the corresponding PGOW problem for MQ can be reduced to the hardness of solving MQ itself. Suppose there exists an efficient adversary capable of guessing the first λ bits of a solution to an MQ instance. Then, by fixing these variables accordingly, one obtains a new MQ system with fewer unknowns and a large number of remaining equations. The resulting system is highly overdetermined and can therefore be solved efficiently, allowing the recovery of the full solution.

A similar security reduction can be applied to the scheme from [HJ24]. In their setting, the hardness of solving the corresponding PGOW problem for their *subfield bilinear collision* (SBC) assumption can be reduced to the hardness of SBC itself, since knowing the first λ bits of the witness yields linear relations with the remaining bits.

However, such security reductions do not hold for all problems. For example, when the witness is structured or sparse, the PGOW security might be weaker than the underlying one-wayness. This is for instance the case in SDitH v2 [ABB+24b] where the witness is composed of several small vectors of Hamming weight at most 1. For the L1 parameter set, the first 128 bits of the witness have a min-entropy of approximately 88 bits. Consequently, applying correlated trees to SDitH v2 would result in a serious security degradation. This illustrates that signature designers must exercise caution when adopting correlated trees, as the required partial-guessing one-wayness property may not be satisfied in all constructions.

3.2 Additional Issues with Davies-Meyer and the RGOW Assumption

While the above vulnerability is inherent to the correlated tree technique itself, it may not be the only source of security degradation. We exhibit hereafter additional issues arising when the seed expansion PRG is instantiated with a Davies-Meyer construction, as in MQOM [BBFR24]. In this setting, a λ-bit block is computed as $E_{\mathsf{sep}}(\mathsf{seed}) \oplus \sigma(\mathsf{seed})$ where E is a block cipher, σ is a linear orthomorphism, and sep is a public domain-separation string. An attacker may attempt to invert a cipher call involved in the expansion in order to recover the corresponding seed. Since the cipher key sep is public, this inversion only requires knowledge of the cipher output.

Consider an adversary attempting to guess the cipher output involved in the $(k-1)$-th λ-bit block of the PRG applied to $\texttt{lseed}[i^*]$ to derive the k-th block of $[\![x]\!]_{i^*}$, for $k > 1$. This block is used to mask the k-th λ-bit block of the witness x in the computation of the auxiliary value Δ_x. Denoting by $\mathsf{Block}_\lambda^{(k)}(\cdot)$ the

function that outputs the k-th λ-bit chunk of its input, we obtain:

$$\begin{aligned}
&\mathsf{Block}_\lambda^{(k)}(\Delta_x) \oplus \mathsf{Block}_\lambda^{(k)}(\sum_{i\neq i^*} [\![x]\!]_i) \\
&\quad = \mathsf{Block}_\lambda^{(k)}(x) \oplus \mathsf{Block}_\lambda^{(k)}([\![x]\!]_{i^*}) \\
&\quad = \mathsf{Block}_\lambda^{(k)}(x) \oplus E_{\mathsf{sep}}(\texttt{lseed[}i^*\texttt{]}) \oplus \sigma(\texttt{lseed[}i^*\texttt{]}) \\
&\quad = \mathsf{Block}_\lambda^{(k)}(x) \oplus E_{\mathsf{sep}}(\texttt{lseed[}i^*\texttt{]}) \oplus \sigma(\mathsf{First}_\lambda(x) \oplus \sum_{i\neq i^*} \texttt{lseed[}i\texttt{]}) \\
&\quad = \mathsf{Block}_\lambda^{(k)}(x) \oplus \sigma(\mathsf{First}_\lambda(x)) \oplus E_{\mathsf{sep}}(\texttt{lseed[}i^*\texttt{]}) \oplus \sigma(\sum_{i\neq i^*} \texttt{lseed[}i\texttt{]}) \ .
\end{aligned}$$

Assume now that $\mathsf{Block}_\lambda^{(k)}(x) \oplus \sigma(\mathsf{First}_\lambda(x))$ can be recovered efficiently (or at least more efficiently than with exhaustive search). From the above equality, the adversary can then recover $E_{\mathsf{sep}}(\texttt{lseed[}i^*\texttt{]})$, since all other quantities are public. As the cipher key sep is known, the adversary can invert the block cipher and recover the leaf seed $\texttt{lseed[}i^*\texttt{]}$. As for the λ-bit attack, this yields the full secret. We thus get that, in addition to a successful guess of $\mathsf{First}_\lambda(x)$, a successful guess of $\mathsf{Block}_\lambda^{(k)}(x) \oplus \sigma(\mathsf{First}_\lambda(x))$, which is a linear function of x, for any block index j, also leads to a full key recovery.

Considering the PRG blocks used to generate $[\![u]\!]_{i^*}$, a similar yet more subtle attack can be mounted. In the general signature scheme depicted in Section 2, the secrecy of $[\![u]\!]_{i^*}$ ensures the masking of sensitive information in the polynomial P_α (which is part of the signature). The information contained in P_α can be fully captured from the pair $(P_\alpha(\omega), \alpha_1)$, where $P_\alpha(\omega)$ is the open evaluation which can be derived from the open leaf seeds and α_1 is the degree-1 coefficient of P_α. The latter can be expressed as follows:

$$\alpha_1 = u_1 + \Gamma \cdot z_1 = \Big([\![u]\!]_{i^*} + \sum_{i\neq i^*} [\![u]\!]_i \Big) + \Gamma \cdot z_1 \ , \tag{4}$$

where z_1 and u_1 denote the degree-1 coefficient of P_z and P_u, respectively. The coordinates of $z_1 = (z_{1,1}, \ldots, z_{1,m})$ can then be expressed as

$$\begin{aligned}
z_{1,j} &= P_x(0) \cdot (A_j^T + A_j) \cdot x + b_j^T \cdot P_x(0) \\
&= (P_x(\omega) - \omega x)^T \cdot (A_j^T + A_j) \cdot x + b_j^T \cdot (P_x(\omega) - \omega x) \\
&= (-2w) \cdot x^T \cdot A_j \cdot x + (P_x(\omega)^T \cdot (A_j^T + A_j) - w \cdot b_j^T) \cdot x + b_j^T \cdot P_x(\omega) \ ,
\end{aligned} \tag{5}$$

for every $j \in \{1, \ldots, m\}$, where A_j and b_j are the matrices and vectors defining the j-th component of F_y. For fields $\mathbb{F}, \mathbb{K}$ of characteristic 2, which we assume here and is the case for MQOM, the above expression simplifies to $z_{1,j} = (P_x(\omega)^T \cdot (A_j^T + A_j) - w \cdot b_j^T) \cdot x + b_j^T \cdot P_x(\omega)$, which is a linear function of x. We thus have that $\Gamma \cdot z_1$ is a public linear function of x whose coefficients depend on the randomness of Γ and $P_x(\omega)$. Then the k-th λ-bit block of Equation (4) yields the following equality:

$$\begin{aligned}
E_{\mathsf{sep}}(\texttt{lseed[}i^*\texttt{]}) = &\underbrace{\sigma(\mathsf{First}_\lambda(x)) \oplus \mathsf{Block}_\lambda^{(k)}(\Gamma \cdot z_1)}_{\text{depends on } x} \\
&\oplus \underbrace{\mathsf{Block}_\lambda^{(k)}(\alpha_1) \oplus \mathsf{Block}_\lambda^{(k)}\big(\textstyle\sum_{i\neq i^*} [\![u]\!]_i\big) \oplus \sigma\big(\textstyle\sum_{i\neq i^*} \texttt{lseed[}i\texttt{]}\big)}_{\text{public value}} \ ,
\end{aligned}$$

given that $\mathsf{Block}_\lambda^{(k)}([\![u]\!]_{i^*}) = E_{\mathsf{sep}}(\texttt{lseed[}i^*\texttt{]}) \oplus \sigma(\texttt{lseed[}i^*\texttt{]})$ for some domain-separation string sep. We thus get that a successful guess of $\sigma(\mathsf{First}_\lambda(x)) \oplus \mathsf{Block}_\lambda^{(k)}(\Gamma \cdot z_1)$, which is a linear function of x, also leads to a full key recovery.

To summarize, each repetition of each signature provides the adversary with functions $g : \mathbb{F}^n \to \{0,1\}^\lambda$ such that a successful guess of $g(x)$ leads to a full key recovery. Let $n_x = \frac{n \log |\mathbb{F}|}{\lambda}$ and $n_u = \frac{\eta \log |\mathbb{K}|}{\lambda}$ denote the

number of λ-bit blocks in x and u, respectively. Then, for each signature, the adversary obtains $n_x + n_u - 1$ such functions $g_1, \ldots, g_{n_x+n_u-1}$ defined as follows:

$$g_k : x \mapsto \begin{cases} \sigma(\mathsf{First}_\lambda(x)) \oplus \mathsf{Block}_\lambda^{(k+1)}(x) & \text{if } k \in \{1, \ldots, n_x - 1\} \\ \sigma(\mathsf{First}_\lambda(x)) \oplus \mathsf{Block}_\lambda^{(k+1-n_x)}(\Gamma \cdot z_1) & \text{if } k \in \{n_x, \ldots, n_x + n_u - 1\} \end{cases} \tag{6}$$

for some random Γ, ω and $P_x(\omega)$. We note that the $n_x - 1$ first functions are the same for all the signatures, while the n_u last functions depend on the signature randomness. Indeed, as shown above, the term $\Gamma \cdot z_1$ is a public linear function of x depending on the randomness of Γ, ω and $P_x(\omega)$.

Remark 2. For the sake of simplicity, we assume here that both x and u fit into complete λ-bit blocks. In practice, a λ-bit block may contain the last bits of x together with the first bits of u. The described attack behaves similarly in such a setting.

Relation-Guessing One-Way Security. To prevent the additional vulnerabilities described above, it must be computationally infeasible to recover $g(x)$ from the public statement y for any of the functions g defined in Equation (6) obtained from the revealed signatures. We formalize this requirement through a notion that we call *relation-guessing one-way security* (RGOW), which captures the additional property that the underlying one-way function must satisfy in order for the scheme to remain secure. In this definition, we consider a general distribution $\mathcal{R}$ for a set of relations $R = \{g : \mathbb{F}^n \to \{0,1\}^\lambda\}$ for which the adversary can try to guess $g(x)$. The adversary is granted access to a relation-sampling oracle $\mathcal{S}$, which outputs fresh relation sets drawn from $\mathcal{R}$. The adversary queries $\mathcal{S}$ to increase the set of admissible functions g and can then query a guessing oracle $\mathcal{G}$ to attempt to guess $g(x)$ for any of the functions g obtained from the relation-sampling oracle.

Definition 2 (Relation-Guessing One-Way Security). *Let $\mathcal{R}$ be a distribution for a set of relations. A one-way function family $\mathcal{F}$ is said to be $(\mathcal{R}, t, Q, \varepsilon_{\text{RGOW}}(t,Q))$-RGOW-secure if any t-time adversary $\mathcal{A}$ making Q queries to the guessing oracle has at most $\varepsilon_{\text{RGOW}}(t,Q)$ probability of winning the $(\mathcal{F}, \mathcal{R})$-RGOW game described in Figure 3 (i.e. the game returns 1).*

$(\mathcal{R}, \mathcal{F})$-RGOW Game:

1. $(x, y) \leftarrow \mathcal{D}_\mathcal{F}$
2. $S \leftarrow \emptyset$
3. Run $\mathcal{A}^{\mathcal{G}(\cdot), \mathcal{S}()}(y)$
4. The game returns 0 ($\mathcal{A}$ loses)

Guessing oracle $\mathcal{G}(g, \hat{x}_0)$:

1. **if** $g \in S$ and $g(x) = \hat{x}_0$ **then** the game returns 1 ($\mathcal{A}$ wins)

Relation sampling oracle $\mathcal{S}()$:

1. $R \leftarrow \mathcal{R}$
2. $S \leftarrow S \cup R$
3. **return** R

Fig. 3: Relation-guessing one-way (RGOW) security game.

4 Simulation Security of Correlated GGM Tree Openings

This section is devoted to prove the simulation security of the correlated GGM tree technique. We first introduce the *correlated tree oracle* capturing the leakage provided by signature queries and recall the H-coefficient technique, which are the main tools used in our proof. Then, we prove the simulation security of this oracle under the PGOW or RGOW assumptions, for different instantiations of the seed derivation, seed commitment and seed expansion functions. Namely, we first consider a simple instantiation based on hash functions in the random oracle model, and then we show how to instantiate the technique in the ideal cipher model, using either a block cipher in Davies-Meyer mode or in counter mode. We finally introduce a tweak of the Davies-Meyer instantiation to obtain a tight security avoiding the RGOW assumption.

4.1 The Correlated Tree Oracle

We start by formally defining the correlated tree oracle, which captures the distribution of the correlated GGM tree opening in an MPC-in-the-Head signature.

A *cryptographic oracle* is an idealized primitive $\mathcal{C}$ modelled by a stateful probabilistic algorithm with access to a fresh source of randomness. In this work, we consider two types of cryptographic oracles: *random oracles* and *ideal ciphers*. A random oracle models an ideal random function. On input a message $m \in \{0,1\}^*$, it replies with a uniformly random string $h \in \{0,1\}^{|\mathsf{out}|}$ of length $|\mathsf{out}|$, and answers consistently to repeated queries. An ideal cipher, takes as input a message $m \in \{0,1\}^\lambda$ and a key $k \in \{0,1\}^\lambda$, and replies with a ciphertext $c \in \{0,1\}^\lambda$. For any fixed key k, the ideal cipher behaves as a random permutation over $\{0,1\}^\lambda$, and answers consistently to repeated queries. We may rely on several cryptographic oracles, denoted by $\mathcal{C} = (\mathcal{C}_1, \ldots, \mathcal{C}_t)$, which may be of different types. We also consider *oracle functions*, which are functions making calls to somes cryptographic oracles as sub-routines.

Definition 3 (Correlated Tree Oracle). *Let $\lambda, \kappa, \ell \in \mathbb{N}$. Let $\mathcal{F}$ be a one-way function family and $(x, y) \in \mathcal{X} \times \mathcal{Y}$ a witness-statement instance of $\mathcal{F}$. Given a salt length $|\mathsf{salt}| \in \mathbb{N}$, let $H^{\mathrm{tree}}, H^{\mathrm{com}}, H^{\mathrm{rnd}}$ be oracle functions satisfying:*

$$\begin{aligned} H^{\mathrm{tree}} &: \{0,1\}^{|\mathsf{salt}|} \times [\ell] \times \{0,1\}^\lambda \to \{0,1\}^\lambda \ , \\ H^{\mathrm{com}} &: \{0,1\}^{|\mathsf{salt}|} \times \{0,1\}^\lambda \to \{0,1\}^{2\lambda} \ , \\ H^{\mathrm{rnd}} &: \{0,1\}^{|\mathsf{salt}|} \times \{0,1\}^\lambda \to \{0,1\}^\kappa \ . \end{aligned}$$

Let $f : \{0,1\}^{|\mathsf{salt}|} \times \{0,1\}^\kappa \times \mathcal{X} \times \mathcal{Y} \times \{0,1\}^\ell \times (\{0,1\}^\lambda)^\ell \times \mathbb{K}^{\eta \times m} \to \{0,1\}^\kappa$ be a deterministic function which for every $(\mathsf{salt}, x, y, b, r_1, \ldots, r_\ell, \Gamma)$ is such that

$$r_{\mathrm{rnd}} \in \{0,1\}^\kappa \ \mapsto \ f(\mathsf{salt}, r_{\mathrm{rnd}}, x, y, b, r_1, \ldots, r_\ell, \Gamma) \in \{0,1\}^\kappa$$

is a bijection. The correlated tree oracle *defined with respect to x, y, H^{tree}, H^{com}, H^{rnd} and f is the oracle $\mathcal{O}$ that, on input a salt salt, outputs a tuple $(b, r_1, \ldots, r_\ell, r_{\mathrm{com}}, \Gamma, \beta)$ as described in Figure 4.*

It is not hard to see that the output distribution of $\mathcal{O}(\mathsf{salt})$ corresponds to the distribution of the correlated GGM tree opening in an MPCitH signature for a well-defined function f where $h_1, \ldots, h_\ell$ are the hidden nodes, $r_1, \ldots, r_\ell$ are their revealed siblings, r_{com} is the revealed seed commitment digest of the hidden leaf, and b corresponds to the bit-decomposition of the hidden index i^*. Moreover, β corresponds to the extra information which cannot be derived from the sibling nodes. Specifically, β is the pair (Δ_x, α_1) serialized as a κ-bit string.[3] The associated function f satisfies the bijection property from Definition 3, where Δ_x is masked with $[\![x]\!]_{i^*}$ and α_1 is masked with $[\![u]\!]_{i^*}$, both composing r_{rnd}.

[3] We recall that in addition to the revealed nodes, the signature includes Δ_x and P_α where P_α is fully determined by $P_\alpha(r)$ (which is computable from the revealed nodes) and α_1, its degree-1 coefficient.

$\mathcal{O}(\mathsf{salt})$:

1. $h_0 \leftarrow \{0,1\}^\lambda$
2. $b := (b_0, \ldots, b_{\ell-1}) \leftarrow \{0,1\}^\ell$
3. **for** $j = 0$ **to** $\ell - 1$ **do**
4. $\quad (u_{j,0}, u_{j,1}) = G_{\text{tree}}(\mathsf{salt}, j, h_j)$
5. $\quad (r_{j+1}, h_{j+1}) = (u_{j,b_j}, u_{j,1-b_j})$
6. **end for**
7. $r_{\text{com}} = H^{\text{com}}(\mathsf{salt}, h_\ell)$
8. $r_{\text{rnd}} = H^{\text{rnd}}(\mathsf{salt}, h_\ell)$
9. $\Gamma \leftarrow \mathbb{K}^{\eta \times m}$
10. $\beta = f(\mathsf{salt}, r_{\text{rnd}}, x, y, b, r_1, \ldots, r_\ell, \Gamma)$
11. **return** $(b, r_1, \ldots, r_\ell, r_{\text{com}}, \Gamma, \beta)$

$G_{\text{tree}}(\mathsf{salt}, j, h_j)$:

1. **if** $j = 0$ **then**
2. $\quad u_{j,0} = h_0$
3. $\quad u_{j,1} = h_0 \oplus \mathsf{First}_\lambda(x)$
4. **else**
5. $\quad u_{j,0} = H^{\text{tree}}(\mathsf{salt}, j, h_j)$
6. $\quad u_{j,1} = u_{j,0} \oplus h_j$
7. **return** $(u_{j,0}, u_{j,1})$

Fig. 4: Correlated tree oracle.

Real world vs. ideal world. The zero-knowledge property of an MPC-in-the-Head scheme based on correlated GGM trees relies on the ability to simulate the oracle $\mathcal{O}$ without knowledge of the secret input x. We will show in the following that we can achieve such a simulation under an idealized assumption (in the ROM or in the ICM) such that a distinguisher $\mathcal{D}$ has a negligible advantage in distinguishing the *real world*, where $\mathcal{D}$ has oracle access to $\mathcal{O}$, from the *ideal world*, where $\mathcal{D}$ has oracle access to the simulator Sim. In both cases, the distinguisher has oracle access to the cryptographic oracles $\mathcal{C}$. In our proofs, the ideal-world simulator Sim shall always work the same way, by simulating the responses to $\mathcal{O}$'s queries with fresh uniform randomness.

We give a formal definition of the ideal-world simulator and of the distinguishing advantage in the following.

Definition 4 (Ideal World Simulator). *Let $\mathcal{O}$ be a correlated tree oracle under Definition 3. The* ideal-world simulator *for $\mathcal{O}$ is a probabilistic algorithm* Sim *which on input* salt *responds by uniformly sampling*

$$(b, r_1, \ldots, r_\ell, r_{\text{com}}, \Gamma, \beta) \leftarrow \{0,1\}^\ell \times (\{0,1\}^\lambda)^\ell \times \{0,1\}^{2\lambda} \times \mathbb{K}^{\eta \times m} \times \{0,1\}^\kappa$$

(independently of the input) and returning this tuple.

Our security proofs rely on bounding the advantage of any distinguisher $\mathcal{D}$ in distinguishing the real world, when $\mathcal{D}$ interacts with $(\mathcal{C}, \mathcal{O})$, from the ideal world, when $\mathcal{D}$ interacts with $(\mathcal{C}, \mathsf{Sim})$. We give a formal definition of this advantage hereafter.

Definition 5 (Distinguishing Advantage). *Let $\mathcal{O}$ be a correlated tree oracle under Definition 3 with y as underlying one-way function statement and $\mathcal{C}$ as underlying cryptographic oracles. Let* Sim *be the ideal-world simulator for $\mathcal{O}$ from Definition 4. Let $\mathcal{D}$ be a distinguishing algorithm having oracle access to either $(\mathcal{C}, \mathcal{O})$ or $(\mathcal{C}, \mathsf{Sim})$, taking y as input and returning a bit. The* distinguishing advantage *of $\mathcal{D}$ is defined as*

$$\varepsilon_{\text{CTO}}(\mathcal{D}) := \left| \Pr[\mathcal{D}^{\mathcal{C},\mathcal{O}}(y) = 1] - \Pr[\mathcal{D}^{\mathcal{C},\mathsf{Sim}}(y) = 1] \right|.$$

Let $\mathsf{D}(Q_{\mathcal{C}}, Q_{\mathcal{O}}, \mu)$ *be the set of all distinguishers making at most at most* $Q_{\mathcal{C}}$ *queries to* $\mathcal{C}$ *and* $Q_{\mathcal{O}}$ *queries to* $\mathcal{O}$*, with* μ *the maximum number of queries to* $\mathcal{O}$ *with the same input salt. The* maximum distinguishing advantage against $\mathcal{O}$ *with query bounds* $(Q_{\mathcal{C}}, Q_{\mathcal{O}}, \mu)$ *is defined as*

$$\varepsilon_{\mathrm{CTO}}(Q_{\mathcal{C}}, Q_{\mathcal{O}}, \mu) := \max_{\mathcal{D} \in \mathsf{D}(Q_{\mathcal{C}}, Q_{\mathcal{O}}, \mu)} \varepsilon_{\mathrm{CTO}}(\mathcal{D}) \ .$$

Remark 3. In practice, a master salt is randomly picked at the beginning of the signing process and a signature yields τ correlated tree openings with different salts which are derived from the master salt. By querying several signatures one may obtain (multi)collisions on the (derived) salts but the maximum number of openings with the same salt, captured by the parameter μ, should remain small with overwhelming probability. Bounding the multicollisions on the salt is deferred to the security analysis of the signature scheme in Section 5.

Query sets and transcripts. In the following, we shall denote by $\mathcal{Q}_{\mathcal{O}}$ and $\mathcal{Q}_{\mathcal{C}}$ the sets of queries to $\mathcal{O}$ and $\mathcal{C}$ respectively made by $\mathcal{D}$. By query, we mean an input-output pair to the corresponding oracle. A query $q \in \mathcal{Q}_{\mathcal{O}}$ is of the form

$$q = (\mathsf{salt}^{(q)}, (b^{(q)}, \boldsymbol{r}^{(q)}, r_{\mathrm{com}}^{(q)}, \Gamma^{(q)}, \beta^{(q)}))$$

where $\mathsf{salt}^{(q)}$ is the input to $\mathcal{O}$ and $(b^{(q)}, \boldsymbol{r}^{(q)}, r_{\mathrm{com}}^{(q)}, \Gamma^{(q)}, \beta^{(q)})$ is the corresponding output, with $\boldsymbol{r}^{(q)} := (r_1^{(q)}, \ldots, r_\ell^{(q)})$. The pair $(\mathcal{Q}_{\mathcal{C}}, \mathcal{Q}_{\mathcal{O}})$ is called the *transcript* of $\mathcal{D}$'s interaction.

We say that a transcript $(\mathcal{Q}_{\mathcal{C}}, \mathcal{Q}_{\mathcal{O}})$ is *attainable* if there exist random tapes for $\mathcal{D}$, $\mathcal{C}$, $\mathcal{O}$ and Sim such that $\mathcal{D}$ produces this transcript either when interacting with $(\mathcal{C}, \mathcal{O})$ or when interacting with $(\mathcal{C}, \mathsf{Sim})$. For a given oracle $\mathcal{C}$ and query list $\mathcal{Q}_{\mathcal{C}}$, we denote by $\mathcal{C} \vdash \mathcal{Q}_{\mathcal{C}}$ the event that all the queries in $\mathcal{Q}_{\mathcal{C}}$ are consistent with the oracle $\mathcal{C}$.

By definition of the correlated tree oracle $\mathcal{O}$, for every query $q \in \mathcal{Q}_{\mathcal{O}}$ (in the real world), the values $h_1^{(q)}$, $\ldots$, $h_\ell^{(q)}$ internally computed by $\mathcal{O}$ (which compose the hidden path of the GGM tree) can be expressed as a function of x and the open values $r_1^{(q)}$, $\ldots$, $r_\ell^{(q)}$ as:

$$h_j^{(q)} = \mathsf{First}_\lambda(x) \oplus \sum_{k=1}^{j} r_k^{(q)} \ . \tag{7}$$

We extend the notation h_j in the setting of the simulator Sim (in the ideal world), where it is defined according to the above relation.

4.2 Application of the H-Coefficient Technique

We recall the principle of the H-coefficient technique from [Pat09, CS14], applied to our context. We consider the interaction transcript $(\mathcal{Q}_{\mathcal{C}}, \mathcal{Q}_{\mathcal{O}})$ produced by the distinguisher in either the real world or the ideal world. The H-coefficient technique consists in partitioning the set of attainable transcripts into two disjoint classes: Good and Bad. Intuitively, Bad contains the "unlikely" or "problematic" transcripts where the analysis fails (e.g., collisions, inconsistencies, etc.), while Good contains the remaining transcripts where we can compare real and ideal probabilities cleanly.

Let $\Pr^{(\mathsf{re})}$ denote the probability distribution of transcripts in the real world, and $\Pr^{(\mathsf{id})}$ the probability distribution of transcripts in the ideal world. The H-coefficient technique bounds the distinguishing advantage by combining: (i) an upper bound on the probability of landing in Bad in the ideal world, and (ii) a lower bound on the likelihood ratio $\Pr^{(\mathsf{re})}[(\mathcal{Q}_{\mathcal{C}}, \mathcal{Q}_{\mathcal{O}})]/\Pr^{(\mathsf{id})}[(\mathcal{Q}_{\mathcal{C}}, \mathcal{Q}_{\mathcal{O}})]$ for all $(\mathcal{Q}_{\mathcal{C}}, \mathcal{Q}_{\mathcal{O}}) \in \mathsf{Good}$. Concretely, and for any partition $\mathsf{Good} \cup \mathsf{Bad}$ of the attainable transcripts, we have the following bound on the distinguishing advantage:

$$\varepsilon_{\mathrm{CTO}}(\mathcal{D}) \leq \Pr^{(\mathsf{id})}\big[(\mathcal{Q}_{\mathcal{C}}, \mathcal{Q}_{\mathcal{O}}) \in \mathsf{Bad}\big] + \left(1 - \frac{\Pr^{(\mathsf{re})}\big[(\mathcal{Q}_{\mathcal{C}}, \mathcal{Q}_{\mathcal{O}}) \in \mathsf{Good}\big]}{\Pr^{(\mathsf{id})}\big[(\mathcal{Q}_{\mathcal{C}}, \mathcal{Q}_{\mathcal{O}}) \in \mathsf{Good}\big]}\right) \ . \tag{8}$$

The interested reader is referred to [CS14] for a comprehensive introduction to this technique and a detailed demonstration of the above bound.

4.3 Security in the Random Oracle Model

We first bound the distinguishing advantage in the random oracle model. Concretely, we consider that H^{tree}, H^{com}, H^{rnd} are three random oracles. In this context, the distinguishing advantage against the correlated tree oracle only relies on the PGOW security assumption.

Theorem 1. *Let $\mathcal{F}$ be a $(\lambda, t, Q, \varepsilon_{\text{PGOW}}(t, Q))$-PGOW-secure family of one-way functions and let $(x, y) \in \mathcal{X} \times \mathcal{Y}$ be a random witness-statement instance of $\mathcal{F}$. Let H^{tree}, H^{com}, H^{rnd} be random oracles. Let $\mathcal{O}$ be a correlated tree oracle with respect to x, y, H^{tree}, H^{com} and H^{rnd}. Then, the maximum distinguishing advantage against $\mathcal{O}$ with query bounds $(Q_{\mathcal{H}}, Q_{\mathcal{O}}, \mu)$ is upper-bound by:*

$$\varepsilon_{\text{CTO}}(Q_{\mathcal{H}}, Q_{\mathcal{O}}, \mu) \leq \varepsilon_{\text{PGOW}}(t, Q) + \frac{(\mu - 1) \cdot \ell \cdot Q_{\mathcal{O}}}{2^{\lambda+1}}$$

with $t = (\mu + 1) \cdot Q_{\mathcal{H}} + Q_{\mathcal{O}}$ and $Q = \mu \cdot Q_{\mathcal{H}}$.

Informally, there are two situations in which an adversary may succeed in distinguishing the two settings (the real world and the ideal word), *i.e.* where the two settings behave differently. The first situation occurs when an input collision happens between two calls to the random oracle made in two different $\mathcal{O}$ queries. In the real world, such calls produce identical outputs, whereas in the ideal world they differ with overwhelming probability, since each simulator query relies on fresh randomness. This event is referred to as $\mathsf{InputCollision}$, and its probability can be easily upper-bounded using standard probabilistic arguments. The second situation arises when the adversary successfully guesses the input of a random-oracle call involved in an $\mathcal{O}$ query. This input is a hidden node h_j which is meant to remain secret. Knowing such a hidden node or correctly guessing it allows the adversary to distinguish the two settings by checking whether the random-oracle output is consistent with the corresponding $\mathcal{O}$ response. Since by Equation (7), such a hidden node consists of the sum of the long-term secret $x^L = \mathsf{First}_\lambda(x)$ and a public value, this distinguishing advantage can be reduced to solving the corresponding PGOW problem. This event is referred to as $\mathsf{InputGuessing}$.

Proof. The following proof relies on the H-Coefficient technique as recalled in Section 4.2.

As explained earlier, a query $q \in \mathcal{Q}_{\mathcal{O}}$ to the correlated tree oracle $\mathcal{O}$ is of the form $q = (\mathsf{salt}^{(q)}, (b^{(q)}, \boldsymbol{r}^{(q)}, r_{\text{com}}^{(q)}, \Gamma^{(q)}, \beta^{(q)}))$ with $\boldsymbol{r}^{(q)} = (r_1^{(q)}, \ldots, r_\ell^{(q)})$. On the other hand, a query $q \in \mathcal{Q}_{\mathcal{H}}$ to the random oracle $\mathcal{H}^{(q)} \in \{H^{\text{tree}}, H^{\text{com}}, H^{\text{rnd}}\}$ is of the form $q = (\mathsf{in}^{(q)}, \mathsf{out}^{(q)})$ such that $\mathsf{out}^{(q)} = \mathcal{H}^{(q)}(\mathsf{in}^{(q)})$. Without loss of generality, we can assume that $\mathsf{in}^{(q)}$ is of the form:

$$\mathsf{in}^{(q)} = \begin{cases} (\mathsf{salt}^{(q)}, j^{(q)}, m^{(q)}) & \text{if } \mathcal{H}^{(q)} = H^{\text{tree}} \\ (\mathsf{salt}^{(q)}, m^{(q)}) & \text{if } \mathcal{H}^{(q)} \in \{H^{\text{com}}, H^{\text{rnd}}\} \end{cases}$$

We extend the notation $j^{(q)}$ when $\mathcal{H}^{(q)} \in \{H^{\text{com}}, H^{\text{rnd}}\}$ by defining $j^{(q)} = \ell$.

We define the set of bad transcripts Bad as follows. A transcript $(\mathcal{Q}_{\mathcal{H}}, \mathcal{Q}_{\mathcal{O}}) \in \mathsf{Bad}$ is a transcript for which one of the following events holds:

- $\mathsf{InputCollision}$: There are distinct queries $q, q' \in \mathcal{Q}_{\mathcal{O}}$ for the same salt (*i.e.* $\mathsf{salt}^{(q)} = \mathsf{salt}^{(q')}$) such that $h_j^{(q)} = h_j^{(q')}$ for some $j \in \{1, \ldots, \ell\}$.
- $\mathsf{InputGuessing}$: There are queries $q \in \mathcal{Q}_{\mathcal{H}}$ and $q' \in \mathcal{Q}_{\mathcal{O}}$ such that $\mathsf{salt}^{(q)} = \mathsf{salt}^{(q')}$ and $m^{(q)} = h_j^{(q')}$, with $j := j^{(q)}$.

Analysis of the bad transcripts in the ideal world. We bound the probability of the above events in the ideal world (i.e., when the transcript is generated by the ideal-world simulator).

We first bound the probability of the $\mathsf{InputCollision}$ event in the ideal world. For fixed $\mathsf{salt} \in \{0,1\}^{|\mathsf{salt}|}$, consider a pair of $q, q' \in \mathcal{Q}_{\mathcal{O}}$ such that $\mathsf{salt} = \mathsf{salt}^{(q)} = \mathsf{salt}^{(q')}$, and an index $j \in [\ell]$. Using Equation (7), we have:

$$\begin{aligned}\Pr^{(\mathsf{id})}\big[h_j^{(q)} = h_j^{(q')}\big] &= \Pr^{(\mathsf{id})}\Big[\mathsf{First}_\lambda(x) \oplus \sum_{k\le j} r_k^{(q)} = \mathsf{First}_\lambda(x) \oplus \sum_{k\le j} r_k^{(q')}\Big] \\ &= \Pr^{(\mathsf{id})}\Big[r_j^{(q)} = r_j^{(q')} \oplus \sum_{k<j}\Big(r_k^{(q)} \oplus r_k^{(q')}\Big)\Big] = \frac{1}{2^\lambda}\end{aligned}$$

using the fact that $r_j^{(q)}$ and $r_j^{(q')}$ are uniform and independent in the ideal world. Thus, for every j and any pair of queries in $\mathcal{Q}_{\mathcal{O}}$, the probability that $\mathsf{InputCollision}$ holds is at most $1/2^\lambda$. If we let $C_{\mathsf{salt}} \le \mu$ denote the number of queries in $\mathcal{Q}_{\mathcal{O}}$ using the salt salt, then

$$\begin{aligned}\Pr^{(\mathsf{id})}[\mathsf{InputCollision}] &\le \sum_{\mathsf{salt}\in\{0,1\}^{|\mathsf{salt}|}} \sum_{j\in\{1,\ldots,\ell\}} \binom{C_{\mathsf{salt}}}{2} \cdot \frac{1}{2^\lambda} \\ &\le (\mu-1)\sum_j \sum_{\mathsf{salt}} \frac{C_{\mathsf{salt}}}{2^{\lambda+1}} \le (\mu-1)\cdot\ell\cdot\frac{Q_{\mathcal{O}}}{2^{\lambda+1}}\end{aligned}$$

Let us now bound the probability of the $\mathsf{InputGuessing}$ event in the ideal world. We will show that if this event occurs, then we can build an adversary $\mathcal{A}$ breaking the partial-guessing one-wayness of $\mathcal{F}$ with high probability. This adversary $\mathcal{A}$ takes as input y and queries the partial guessing oracle $\mathcal{G}$ of the $(\lambda,\mathcal{F})$-PGOW security game (see Figure 2). Whenever $\mathcal{A}$ inputs $\hat{x}_0 = \mathsf{First}_\lambda(x)$ to $\mathcal{G}$, it wins the game. This adversary $\mathcal{A}$ simply calls the distinguisher $\mathcal{D}$ and simulates the responses to $\mathcal{D}$'s queries to $\mathcal{H}$ and $\mathcal{O}$ using the ideal-world simulator. Then $\mathcal{A}$ tests several candidates $\hat{x}_0$ for $\mathsf{First}_\lambda(x)$ based on $\mathcal{D}$'s transcript. We show below that if the event $\mathsf{InputGuessing}$ occurs in $\mathcal{D}$'s transcript, then one of $\mathcal{A}$'s candidates is the right $\mathsf{First}_\lambda(x)$, and hence $\mathcal{A}$ wins the PGOW security game.

The tested candidates are defined as follows. For each query $q \in \mathcal{Q}_{\mathcal{H}}$, $\mathcal{A}$ tests the candidate:

$$\hat{x}_0 = m^{(q)} \oplus \sum_{k\le j^{(q)}} r_k^{(q')}$$

for every query $q' \in \mathcal{Q}_{\mathcal{O}}$ satisfying $\mathsf{salt}^{(q')} = \mathsf{salt}^{(q)}$. By definition of the $\mathsf{InputGuessing}$ event, there is at least one pair of queries $(q,q') \in \mathcal{Q}_{\mathcal{H}} \times \mathcal{Q}_{\mathcal{O}}$ such that $\mathsf{salt}^{(q)} = \mathsf{salt}^{(q')}$ and $m^{(q)} = h_j^{(q')}$, with $j := j^{(q)}$. By Equation (7), for this pair of queries, we have

$$\hat{x}_0 = m^{(q)} \oplus \sum_{k\le j} r_k^{(q')} = h_j^{(q')} \oplus \sum_{k\le j} r_k^{(q')} = \mathsf{First}_\lambda(x)\ .$$

The adversary $\mathcal{A}$ makes $\mu\cdot Q_{\mathcal{H}}$ queries to the guessing oracle and its execution time is roughly of $Q_{\mathcal{H}}+Q_{\mathcal{O}}$ to simulate $\mathcal{D}$'s execution in the ideal world, plus $\mu\cdot Q_{\mathcal{H}}$ to test the candidates $\hat{x}_0$. Therefore,

$$\Pr^{(\mathsf{id})}[\mathsf{InputGuessing}] = \Pr[\mathcal{A}\text{ succeeds}] \le \varepsilon_{\mathrm{PGOW}}\big(t, Q\big)\ ,$$

with $t = (\mu+1)\cdot Q_{\mathcal{H}} + Q_{\mathcal{O}}$ and $Q = \mu\cdot Q_{\mathcal{H}}$ as in the statement of the theorem. Summarizing, the probability of a bad transcript in the ideal world is upper-bounded by:

$$\Pr^{(\mathsf{id})}\big[(\mathcal{Q}_{\mathcal{H}},\mathcal{Q}_{\mathcal{O}}) \in \mathsf{Bad}\big] \le \varepsilon_{\mathrm{PGOW}}\big((\mu+1)\cdot Q_{\mathcal{H}} + Q_{\mathcal{O}}, \mu\cdot Q_{\mathcal{H}}\big) + \frac{(\mu-1)\cdot\ell\cdot Q_{\mathcal{O}}}{2^{\lambda+1}}\ .$$

Analysis of the good transcripts. Fix a good transcript $\mathcal{Q} = (\mathcal{Q}_{\mathcal{H}},\mathcal{Q}_{\mathcal{O}})$. The probability to obtain this transcript in the ideal world is given by

$$\Pr^{(\mathsf{id})}[\mathcal{H}\vdash\mathcal{Q}_{\mathcal{H}}]\cdot\prod_{q\in\mathcal{Q}_{\mathcal{O}}}\Pr^{(\mathsf{id})}[(b^{(q)},\boldsymbol{r}^{(q)},r_{\mathrm{com}}^{(q)},\Gamma^{(q)},\beta^{(q)})] = \frac{\Pr^{(\mathsf{id})}[\mathcal{H}\vdash\mathcal{Q}_{\mathcal{H}}]}{\big(2^\ell\cdot(2^\lambda)^\ell\cdot 2^{2\lambda}\cdot|\mathbb{K}|^{\eta\cdot n}\cdot 2^\kappa\big)^{Q_{\mathcal{O}}}}\ .$$

The probability to obtain this transcript in the real world is

$$\Pr^{(\mathsf{re})}[\forall q \in \mathcal{Q}_{\mathcal{O}} : (b^{(q)}, \boldsymbol{r}^{(q)}, r_{\mathrm{com}}^{(q)}, \Gamma^{(q)}, \beta^{(q)}) \mid \mathcal{H} \vdash \mathcal{Q}_{\mathcal{H}}] \cdot \Pr^{(\mathsf{re})}[\mathcal{H} \vdash \mathcal{Q}_{\mathcal{H}}] \ .$$

We can express the left term in the above product as:

$$\prod_{j=1}^{Q_{\mathcal{O}}} \Pr^{(\mathsf{re})}\left[(b^{(q_j)}, \boldsymbol{r}^{(q_j)}, r_{\mathrm{com}}^{(q_j)}, \Gamma^{(q_j)}, \beta^{(q_j)}) \mid \mathcal{H} \vdash \mathcal{Q}_{\mathcal{H}} \wedge \forall k < j : (b^{(q_k)}, \boldsymbol{r}^{(q_k)}, r_{\mathrm{com}}^{(q_k)}, \Gamma^{(q_k)}, \beta^{(q_k)})\right] \ ,$$

labelling $q_1, \ldots, q_{Q_{\mathcal{O}}}$ the queries in $\mathcal{Q}_{\mathcal{O}}$. Since the transcript is good, there is no query in $\mathcal{Q}_{\mathcal{H}}$ with a hash input that would be involved in a query to $\mathcal{O}$ (since the event $\mathsf{InputGuessing}$ does not occur). Moreover, there is no previous opening with a hash input that would also appear in this opening (since the event $\mathsf{InputCollision}$ does not occur). For all j, we thus have:

$$\Pr^{(\mathsf{re})}\left[(b^{(q_j)}, \boldsymbol{r}^{(q_j)}, r_{\mathrm{com}}^{(q_j)}, \Gamma^{(q_j)}, \beta^{(q_j)}) \mid (\mathcal{H} \vdash \mathcal{Q}_{\mathcal{H}}) \wedge (\forall k < j : (b^{(q_k)}, \boldsymbol{r}^{(q_k)}, r_{\mathrm{com}}^{(q_k)}, \Gamma^{(q_k)}, \beta^{(q_k)}))\right] = \frac{1}{2^{\ell} \cdot (2^{\lambda})^{\ell} \cdot 2^{2\lambda} \cdot |\mathbb{K}|^{\eta \cdot n} \cdot 2^{\kappa}} \ .$$

It follows that

$$\Pr^{(\mathsf{re})}[\forall q \in \mathcal{Q}_{\mathcal{O}} : (b^{(q)}, \boldsymbol{r}^{(q)}, r_{\mathrm{com}}^{(q)}, \Gamma^{(q)}, \beta^{(q)}) \mid H \vdash \mathcal{Q}_H] = \left(\frac{1}{2^{\ell} \cdot (2^{\lambda})^{\ell} \cdot 2^{2\lambda} \cdot |\mathbb{K}|^{\eta \cdot n} \cdot 2^{\kappa}}\right)^{Q_{\mathcal{O}}} \ .$$

We deduce that, for any good transcript $(\mathcal{Q}_{\mathcal{H}}, \mathcal{Q}_{\mathcal{O}}) \in \mathsf{Good}$, the probability to obtain it in the real world is equal to the probability to obtain it in the ideal world. This implies:

$$\Pr^{(\mathsf{re})}\left[(\mathcal{Q}_{\mathcal{H}}, \mathcal{Q}_{\mathcal{O}}) \in \mathsf{Good}\right] = \Pr^{(\mathsf{id})}\left[(\mathcal{Q}_{\mathcal{H}}, \mathcal{Q}_{\mathcal{O}}) \in \mathsf{Good}\right]$$

which, by Equation (8), concludes the proof. □

4.4 Security in the Ideal Cipher Model with Davies-Meyer Instantiation

Although the random oracle model yields a simple and tight proof, hash-based instantiations significantly impact the running time of the scheme. For this reason, most modern MPCitH-based constructions rely instead on a block cipher such as AES, leading to substantially more efficient implementations.

In this section, we instantiate the primitives H^{tree}, H^{com} and H^{rnd} using a block cipher E in Davies-Meyer mode. Concretely, the oracle functions H^{tree}, H^{com} and H^{rnd} are defined as:

$$\begin{aligned}
H^{\mathrm{tree}}(\mathsf{salt}, j, m) &= H_{(\mathsf{salt} \| j)}(m) \\
H^{\mathrm{com}}(\mathsf{salt}, m) &= H_{(\mathsf{salt} \| \ell)}(m) \parallel H_{(\mathsf{salt} \| \ell+1)}(m) \\
H^{\mathrm{rnd}}(\mathsf{salt}, m) &= H_{(\mathsf{salt} \| \ell+2)}(m) \parallel \ldots \parallel H_{(\mathsf{salt} \| \ell+1+n_{\mathrm{rnd}})}(m)
\end{aligned}$$

where

$$H_{\mathsf{sep}}(m) = E_{\mathsf{sep}}(m) \oplus \sigma(m) \ .$$

We assume that $\kappa = n_{\mathrm{rnd}} \cdot \lambda$ with $n_{\mathrm{rnd}} \in \mathbb{N}$. The case where κ is not a multiple of λ can be handled as explained in Remark 4. The domain separator string sep is of the form $\mathsf{salt} \parallel \mathsf{ind}$ where the salt salt is encoded on $|\mathsf{salt}|$ bits and the index $\mathsf{ind} \in \{1, \ldots, \ell + 1 + n_{\mathrm{rnd}}\}$ is encoded on $|\mathsf{ind}|$ bits where:

$$|\mathsf{ind}| = \lceil \log_2(\ell + 1 + n_{\mathrm{rnd}}) \rceil \quad \text{and} \quad |\mathsf{salt}| = \lambda - |\mathsf{ind}| \ .$$

Such an instantiation is for example used in the NIST candidate MQOM v2 [BBFR24].

For the following security statement, we need to specify the function f underlying the expression of β in the correlated tree oracle $\mathcal{O}$. Based on the general signature scheme depicted in Section 4.1, the value β returned by the oracle is the pair (Δ_x, α_1) serialized as a κ-bit string. The associated function f is formally defined as:

$$f : (\mathsf{salt}, r_{\mathrm{rnd}}, x, y, b, \boldsymbol{r}) \mapsto \begin{bmatrix} \Delta_x \\ \alpha_1 \end{bmatrix} := \begin{bmatrix} \mathsf{NextBits}_\lambda(x) \oplus \sum_{i=0}^{N-1} [\![x]\!]_i \\ z_1 \oplus \sum_{i=0}^{N-1} [\![u]\!]_i \end{bmatrix} , \tag{9}$$

where i^* is the integer whose binary decomposition is b, the shares $\{([\![x]\!]_i, [\![u]\!]_i)\}_{i \neq i^*}$ are derived from the open siblings $\boldsymbol{r} := (r_1, \ldots, r_\ell)$, the value r_{rnd} is interpreted as $([\![x]\!]_{i^*}, [\![u]\!]_{i^*})$, and z_1 is the degree-1 coefficient of the polynomial $P_z = F_y(P_x)$. We note that P_x can be computed from x (its degree-1 coefficient) and the evaluation $P_x(\omega_{i^*})$ derived from the open siblings $\boldsymbol{r}$ and Δ_x.

As explained in Section 3, the security of the Davies-Meyer relies on the relation-guessing one-wayness (RGOW) assumption for $\mathcal{F}$, in addition to the partial-guessing one-wayness (PGOW) assumption. The definition of the functions $\{g\}$ we consider under Equations (6) and (5) depends on the ideal cipher; formally, they are oracle functions. Indeed, the term $P_x(\omega)$ in the expression of z_1 is derived from the public siblings $r_1, \ldots, r_\ell$ by making calls to the ideal cipher. For this reason, we need to slightly adapt the RGOW game as follows. First we consider that the RGOW adversary is given oracle access to the ideal cipher E, in addition to the relation-sampling oracle $\mathcal{S}_{\mathrm{RGOW}}$ and the guessing oracle $\mathcal{G}_{\mathrm{RGOW}}$. Then, we consider that the distribution $\mathcal{R}$ outputs a tuple of uniform random values $(i^*, \boldsymbol{r}, \Gamma)$, with $\boldsymbol{r} = (r_1, \ldots, r_\ell)$, for which the associated set $R_{(i^*, \boldsymbol{r}, \Gamma)}$ contains the oracle functions g defined in Equation (6). This set $R_{(i^*, \boldsymbol{r}, \Gamma)}$ also contains the relation First_λ, meaning that the adversary wins whenever it finds $\hat{x}_0$ such that $\hat{x}_0 = \mathsf{First}_\lambda(x)$. Consequently, the RGOW problem over the distribution $\mathcal{R}$ can only be easier to solve than the PGOW problem, since any PGOW adversary immediately yields an RGOW adversary.

The following theorem states the simulation security of the correlated tree oracle in the ideal cipher model with Davies-Meyer instantiation.

Theorem 2. *Let $\mathcal{F}$ be a family of one-way functions and let $(x, y) \in \mathcal{X} \times \mathcal{Y}$ be a random witness-statement instance of $\mathcal{F}$. Assume that $\mathcal{F}$ is $(\mathcal{R}, t, Q, \varepsilon_{\mathrm{RGOW}}(t, Q))$-RGOW-secure, where $\mathcal{R}$ is the relation distribution defined above. Let E be an ideal cipher and let H^{tree}, H^{com}, H^{rnd} and f be oracle functions defined with respect to E as described above. Let $\mathcal{O}$ be a correlated tree oracle with respect to x, y, H^{tree}, H^{com} and H^{rnd}. Then, the maximum distinguishing advantage against $\mathcal{O}$ with query bounds $(Q_E, Q_{\mathcal{O}}, \mu)$ is upper-bound by:*

$$\varepsilon_{\mathrm{CTO}}(Q_E, Q_{\mathcal{O}}, \mu) \leq \varepsilon_{\mathrm{RGOW}}(t, Q) + \frac{(\mu - 1) \cdot (2\ell + 1 + n_{\mathrm{rnd}}) \cdot Q_{\mathcal{O}}}{2^{\lambda+1}}$$

with $t = (2\mu + 1) \cdot Q_E + Q_{\mathcal{O}}$ and $Q = 2\mu \cdot Q_E$.

The following proof is similar to that in the random oracle model, considering similar bad events as well as extra ones stemming from the use of an ideal cipher instead of a random oracle. The first bad event occurs when a collision happens on the plaintext-key pair in two calls to the ideal cipher made in two different $\mathcal{O}$ queries. This event is referred to as $\mathsf{KeyInputCollision}$ and is analyzed in exactly the same way as $\mathsf{InputCollision}$ in the ROM proof. The second bad event arises when the adversary successfully guesses the input of an ideal-cipher call involved in an $\mathcal{O}$ query. This event is referred to as $\mathsf{KeyInputGuessing}$ and similarly follows the same analysis as $\mathsf{InputGuessing}$ in the ROM proof. The two remaining events stem from the fact that, for a fixed domain separator sep, the outputs of the oracle function H_{sep} are correlated, since they correspond to different evaluations of the same permutation. In contrast, the simulation relies on fresh, independent randomness, and this discrepancy can potentially be exploited by an adversary. Thus, the third situation occurs when an output collision happens between two calls to the ideal cipher made in two different $\mathcal{O}$ queries. In the real world, such a collision never occurs, whereas in the ideal world it may occur with negligible probability. This event is referred to as $\mathsf{KeyOutputCollision}$, and its probability can be upper-bounded using standard probabilistic arguments. Finally, the fourth situation arises when the adversary successfully guesses the output of an ideal-cipher call involved in an $\mathcal{O}$ query. This output is intended to remain secret, since knowledge of it would allow the adversary to invert the ideal-cipher call and recover the

long-term secret (as the cipher key is public). Because such a secret output can be expressed as a public relation involving the long-term secret $x^L = \mathsf{First}_\lambda(x)$ and public values, this distinguishing advantage can be reduced to solving instance of the RGOW problem over $\mathcal{R}$. This event is referred to as KeyOutputGuessing.

Proof. We denote the transcript of $\mathcal{D}$'s interaction by $\mathcal{Q} = (\mathcal{Q}_E, \mathcal{Q}_\mathcal{O})$. A E-query $(\mathsf{key}, \mathsf{in}, \mathsf{out}) \in \mathcal{Q}_E$ satisfies $\mathsf{out} = E_{\mathsf{key}}(\mathsf{in})$. We also denote $s_j^{(q)}$ the output of the block cipher with index j for a query $q \in \mathcal{Q}_\mathcal{O}$, which *in the real world* is defined as:

$$s_j^{(q)} = \begin{cases} E_{(\mathsf{salt}^{(q)} \| j)}(h_j^{(q)}) \text{ when } 1 \le j < \ell \text{ (in } H^{\mathrm{tree}}) \\ E_{(\mathsf{salt}^{(q)} \| j)}(h_\ell^{(q)}) \text{ when } \ell \le j \le \ell + 1 + n_{\mathrm{rnd}} \text{ (in } H^{\mathrm{com}} \text{ or } H^{\mathrm{rnd}}) \end{cases}$$

By structure of the construction, we have the following relations, for all $j \in \{1, \ldots, \ell+1+n_{\mathrm{rnd}}\}$ and $q \in \mathcal{Q}_\mathcal{O}$:

$$\begin{cases} u_{j,0}^{(q)} = s_j^{(q)} \oplus \sigma(h_j^{(q)}), & \text{when } 1 \le j < \ell \text{ (in } H^{\mathrm{tree}}) \\ r_{\mathrm{com},k}^{(q)} = s_j^{(q)} \oplus \sigma(h_\ell^{(q)}), & \text{when } j = \ell + k \text{ with } k \in \{0,1\} \text{ (in } H^{\mathrm{com}}), \\ r_{\mathrm{rnd},k}^{(q)} = s_j^{(q)} \oplus \sigma(h_\ell^{(q)}), & \text{when } j = \ell + 2 + k \text{ with } k \in \{0, \ldots, n_{\mathrm{rnd}} - 1\} \text{ (in } H^{\mathrm{rnd}}), \end{cases}$$

where $r_{\mathrm{com},k}^{(q)}$ (resp. $r_{\mathrm{rnd},k}^{(q)}$) is the kth λ-bit block of $r_{\mathrm{com}}^{(q)}$ (resp. $r_{\mathrm{rnd}}^{(q)}$) and where

$$u_{j,0}^{(q)} = \begin{cases} r_{j+1}^{(q)} & \text{if } b_j^{(q)} = 0 \\ h_{j+1}^{(q)} & \text{if } b_j^{(q)} = 1 \end{cases}$$

In the ideal world, the hidden values $\{h_j^{(q)}\}$ are defined from the simulated revealed values $\{r_j^{(q)}\}$ and x from Equation (7) and the cipher outputs $\{s_j^{(q)}\}$ are defined from those values using the above relations.

We define the set of bad transcripts Bad as follows. A transcript $(\mathcal{Q}_E, \mathcal{Q}_\mathcal{O}) \in \mathsf{Bad}$ is a transcript for which one of the following events holds:

- KeyInputCollision: There are distinct queries $q, q' \in \mathcal{Q}_\mathcal{O}$ for the same salt (*i.e.* $\mathsf{salt}^{(q)} = \mathsf{salt}^{(q')}$) such that $h_j^{(q)} = h_j^{(q')}$ for some $j \in \{1, \ldots, \ell\}$.
- KeyOutputCollision: There are distinct queries $q, q' \in \mathcal{Q}_\mathcal{O}$ for the same salt salt (*i.e.* $\mathsf{salt}^{(q)} = \mathsf{salt}^{(q')}$) such that $s_j^{(q)} = s_j^{(q')}$ for some $j \in \{1, \ldots, \ell + 1 + n_{\mathrm{rnd}}\}$.
- KeyInputGuessing: There is a query $q \in \mathcal{Q}_\mathcal{O}$ and a query $((\mathsf{salt} \parallel j), \mathsf{in}, \mathsf{out}) \in \mathcal{Q}_E$ such that $\mathsf{salt}^{(q)} = \mathsf{salt}$ and $\mathsf{in} = h_j^{(q)}$.
- KeyOutputGuessing: There is a query $q \in \mathcal{Q}_\mathcal{O}$ and a query $((\mathsf{salt} \parallel j), \mathsf{in}, \mathsf{out}) \in \mathcal{Q}_E$ such that $\mathsf{salt}^{(q)} = \mathsf{salt}$ and $\mathsf{out} = s_j^{(q)}$.

Analysis of the bad transcripts in ideal world. The analysis of the KeyInputCollision event is strictly identical to the case with the random oracle (see event InputCollision). We thus have

$$\Pr^{(\mathsf{id})}[\mathsf{KeyInputCollision}] \le \frac{(\mu - 1) \cdot \ell \cdot Q_\mathcal{O}}{2^{\lambda+1}}.$$

We bound the probability of the KeyOutputCollision event in the ideal world. For fixed $\mathsf{salt} \in \{0,1\}^{|\mathsf{salt}|}$ and index $j \in \{1, \ldots, \ell + 1 + n_{\mathrm{rnd}}\}$, consider a pair of queries $q, q' \in \mathcal{Q}_\mathcal{O}$ such that $\mathsf{salt} = \mathsf{salt}^{(q)} = \mathsf{salt}^{(q')}$. For $j < \ell$, we have

$$\begin{aligned} \Pr^{(\mathsf{id})}[s_j^{(q)} = s_j^{(q')}] &= \Pr^{(\mathsf{id})}[u_{j,0}^{(q)} \oplus \sigma(h_j^{(q)}) = u_{j,0}^{(q')} \oplus \sigma(h_j^{(q')})] \\ &= \Pr^{(\mathsf{id})}[r_{j+1}^{(q)} \oplus (b_j^{(q)} \cdot x^L) \oplus \sigma(h_j^{(q)}) = r_{j+1}^{(q')} \oplus (b_j^{(q')} \cdot x^L) \oplus \sigma(h_j^{(q')})] \\ &= \frac{1}{2^\lambda} \end{aligned}$$

using the fact that $r_{j+1}^{(q)}$ and $r_{j+1}^{(q')}$ are uniform and independent in the ideal world. Similarly, when $j = \ell + k$, with $k \in \{0, 1\}$, we have

$$\Pr^{(\mathsf{id})}[s_j^{(q)} = s_j^{(q')}] = \Pr^{(\mathsf{id})}[r_{\mathrm{com},k}^{(q)} \oplus \sigma(h_\ell^{(q)}) = r_{\mathrm{com},k}^{(q')} \oplus \sigma(h_\ell^{(q')})] = \frac{1}{2^\lambda}$$

using the fact that $r_{\mathrm{com},k}^{(q)}$ and $r_{\mathrm{com},k}^{(q')}$ are uniform and independent in the ideal world. Finally, for $j = \ell+2+k$, with $k \in \{0, \ldots, n_{\mathrm{rnd}} - 1\}$, we have

$$\Pr^{(\mathsf{id})}[s_j^{(q)} = s_j^{(q')}] = \Pr^{(\mathsf{id})}[r_{\mathrm{rnd},k}^{(q)} \oplus \sigma(h_\ell^{(q)}) = r_{\mathrm{rnd},k}^{(q')} \oplus \sigma(h_\ell^{(q')})] = \frac{1}{2^\lambda}$$

using the fact that $r_{\mathrm{rnd},k}^{(q)}$ and $r_{\mathrm{rnd},k}^{(q')}$ are uniform and independent, which itself follows from the fact that $\beta^{(q)}$ and $\beta^{(q')}$ are uniform and independent in the ideal world, together with the bijectivity property of the function f from Definition 3. Thus, given a salt and an index, for any pair of queries in $\mathcal{Q}_\mathcal{O}$, the probability that $\mathsf{KeyOutputCollision}$ holds is $1/2^\lambda$. If we let $C_{\mathsf{salt}} \leq \mu$ denote the number of queries in $\mathcal{Q}_\mathcal{O}$ using the salt salt, then

$$\begin{aligned}\Pr^{(\mathsf{id})}[\mathsf{KeyOutputCollision}] &\leq \sum_{\mathsf{salt} \in \{0,1\}^{|\mathsf{salt}|}} \sum_{j \in \{1,\ldots,\ell+1+n_{\mathrm{rnd}}\}} \binom{C_{\mathsf{salt}}}{2} \cdot \frac{1}{2^\lambda} \\ &\leq (\mu - 1) \cdot \sum_j \sum_{\mathsf{salt}} \frac{C_{\mathsf{salt}}}{2^{\lambda+1}} = \frac{(\mu-1) \cdot (\ell + 1 + n_{\mathrm{rnd}}) \cdot Q_\mathcal{O}}{2^{\lambda+1}}\end{aligned}$$

Let us now bound the probability of the *guessing event* defined as

$$\mathsf{Guessing} = \mathsf{KeyInputGuessing} \cup \mathsf{KeyOutputGuessing}$$

in the ideal world. We will show that if this event occurs, then we can build an adversary $\mathcal{A}$ breaking the RGOW security of $\mathcal{F}$ over $\mathcal{R}$. This adversary $\mathcal{A}$ takes as input the statement y and queries the guessing oracle $\mathcal{G}_{\mathrm{RGOW}}$ and the relation-sampling oracle $\mathcal{S}_{\mathrm{RGOW}}$. Whenever $\mathcal{A}$ inputs $\hat{x}_0 = \mathsf{First}_\lambda(x)$ or $\hat{x}_0 = g(x)$ to $\mathcal{G}_{\mathrm{RGOW}}$ for some previously sampled relation g defined in Equation (6), it wins the corresponding game. This adversary $\mathcal{A}$ simply calls the distinguisher $\mathcal{D}$ and replies $\mathcal{D}$'s queries as follows. On one hand, $\mathcal{A}$ forwards $\mathcal{D}$'s queries to the ideal cipher E. On the other hand, for each query q to $\mathcal{O}$, $\mathcal{A}$ queries the RGOW relation sampling oracle $\mathcal{S}_{\mathrm{RGOW}}$ to obtain a random tuple $(i^*, \boldsymbol{r}, \Gamma)$. $\mathcal{A}$ then replies to q by sampling random values $r_{\mathrm{com}}^{(q)}$ and $\beta^{(q)}$ and returning $(b^{(q)}, \boldsymbol{r}^{(q)}, r_{\mathrm{com}}^{(q)}, \Gamma^{(q)}, \beta^{(q)})$ where $b^{(q)}$ denotes the binary decomposition of i^*, and $\boldsymbol{r}^{(q)}, \Gamma^{(q)}$ are simply set to $\boldsymbol{r}, \Gamma$ returned by $\mathcal{S}_{\mathrm{RGOW}}$. We denote $\{g_k^{(q)}\}$ the set of functions defined from $\boldsymbol{r}^{(q)}$ and $\Gamma^{(q)}$ as in Equation (6) for which $\mathcal{A}$ can then try to guess $\hat{x}_0 = g_k^{(q)}(x)$. Because all the elements are uniformly random in this response, the distribution of $\mathcal{D}$'s view is perfectly consistent with the ideal world distribution, hence $\mathcal{D}$'s transcript is distributed according to the ideal world distribution. Then $\mathcal{A}$ tests several candidates $\hat{x}_0$ for $\mathsf{First}_\lambda(x)$ and several candidates $\hat{x}_0$ for $g(x)$ with $\mathcal{G}_{\mathrm{RGOW}}$, with g among the accumulated set of relations from $\mathcal{S}_{\mathrm{RGOW}}$. We show below that if the event $\mathsf{Guessing}$ occurs in $\mathcal{D}$'s transcript, then one of $\mathcal{A}$'s candidates is correct and hence $\mathcal{A}$ wins the RGOW security game.

The tested candidates are defined as follows. For each query $((\mathsf{salt} \parallel j), \mathsf{in}, \mathsf{out}) \in \mathcal{Q}_E$ and each query $q \in \mathcal{Q}_\mathcal{O}$ satisfying $\mathsf{salt}^{(q)} = \mathsf{salt}$,

- $\mathcal{A}$ tests the candidate $\hat{x}_0 := \mathsf{in} \oplus \sum_{k \leq \min(j,\ell)} r_k^{(q)}$ with $\mathcal{G}_{\mathrm{RGOW}}$. If $\mathsf{KeyInputGuessing}$ holds for q and the current cipher query, we have

$$\hat{x}_0 := \mathsf{in} \oplus \sum_{i \leq \min(j,\ell)} r_i^{(q)} = h_{\min(j,\ell)}^{(q)} \oplus \sum_{i \leq \min(j,\ell)} r_i^{(q)} = \mathsf{First}_\lambda(x) \ .$$

– for $j < \ell$, $\mathcal{A}$ tests the candidate $\hat{x}_0 := \Phi^{-1}(r_{j+1}^{(q)} \oplus \mathsf{out} \oplus \sigma(\sum_{i \le j} r_i^{(q)}))$ with $\mathcal{G}_{\text{RGOW}}$, with $\Phi(\cdot) := \sigma(\cdot) \oplus b_j^{(q)} \cdot \mathrm{Id}(\cdot)$. If KeyOutputGuessing holds for q and the current cipher query with $j < \ell$, we have

$$\begin{aligned}\Phi(\hat{x}_0) &:= r_{j+1}^{(q)} \oplus \mathsf{out} \oplus \sigma(\sum_{i \le j} r_i^{(q)}) \\ &= r_{j+1}^{(q)} \oplus s_j^{(q)} \oplus \sigma(\sum_{i \le j} r_i^{(q)}) = (r_{j+1}^{(q)} \oplus u_{j,0}^{(q)}) \oplus \sigma(h_j^{(q)} \oplus \sum_{i \le j} r_i^{(q)}) \\ &= (b_j^{(q)} \cdot \mathsf{First}_\lambda(x)) \oplus \sigma(\mathsf{First}_\lambda(x)) = \Phi(\mathsf{First}_\lambda(x)) \ .\end{aligned}$$

– for $j = \ell + k$ with $k \in \{0, 1\}$, $\mathcal{A}$ tests $\hat{x}_0 := \sigma^{-1}(r_{\text{com},k}^{(q)} \oplus \mathsf{out} \oplus \sigma(\sum_{i \le \ell} r_i^{(q)}))$ with $\mathcal{G}_{\text{RGOW}}$. If KeyOutputGuessing holds for q and the current cipher query with $j = \ell + k$, we have

$$\begin{aligned}\sigma(\hat{x}_0) &:= r_{\text{com},k}^{(q)} \oplus \mathsf{out} \oplus \sigma(\sum_{i \le \ell} r_i^{(q)}) = r_{\text{com},k}^{(q)} \oplus s_j^{(q)} \oplus \sigma(\sum_{i \le \ell} r_i^{(q)}) \\ &= \sigma(h_\ell^{(q)} \oplus \sum_{i \le \ell} r_i^{(q)}) = \sigma(\mathsf{First}_\lambda(x)) \ .\end{aligned}$$

– for $j = \ell + 2 + k$ with $k \in \{0, \ldots, n_{\text{rnd}} - 1\}$, $\mathcal{A}$ tests the candidate

$$\hat{x}_0 := \mathsf{out} \oplus \beta_k^{(q)} \oplus f'_k(b^{(q)}, r_1^{(q)}, \ldots, r_\ell^{(q)})$$

for $g_k^{(q)}$ where $\beta_k^{(q)}$ is the kth λ-bit block of $\beta^{(q)}$ and where f'_k is defined as:

$$f'_k : (b, r_1, \ldots, r_\ell) \mapsto \mathsf{Block}_\lambda^{(k)}\big(\sum\nolimits_{i \neq i^*} [\![x \parallel u]\!]_i\big) \oplus \sigma\big(\sum\nolimits_{i \neq i^*} \mathtt{lseed[}i\mathtt{]}\big)$$

with i^* the index with binary decomposition $b^{(q)}$ and $[\![x \parallel u]\!]_i$ the concatenation of $[\![x]\!]_i$ and $[\![u]\!]_i$, where $[\![x \parallel u]\!]_i$ and $\mathtt{lseed[}i\mathtt{]}$ are derived from the siblings $r_1^{(q)}, \ldots, r_\ell^{(q)}$ for all $i \neq i^*$. If KeyOutputGuessing holds for q and the current cipher query with $j = \ell + 2 + k$, the above relation holds by definition of the $\{g_k^{(q)}\}$ functions (see Section 3).

The execution time of $\mathcal{A}$ is roughly of $Q_E + Q_{\mathcal{O}}$ to simulate $\mathcal{D}$'s execution in the ideal world, plus $2\mu \cdot Q_E$ to test the candidates $\hat{x}_0$ for RGOW. Therefore,

$$\Pr^{(\mathsf{id})}[\mathsf{Guessing}] = \Pr[\mathcal{A} \text{ succeeds}] \le \varepsilon_{\text{RGOW}}(t, Q) \ .$$

with $t = (2\mu + 1) \cdot Q_E + Q_{\mathcal{O}}$ and $Q = 2\mu \cdot Q_E$.

Summarizing, the probability of a bad transcript in the ideal world is upper-bounded by:

$$\Pr^{(\mathsf{id})}\big[(\mathcal{Q}_E, \mathcal{Q}_{\mathcal{O}}) \in \mathsf{Bad}\big] \le \varepsilon_{\text{RGOW}}(t, Q) + \frac{(\mu - 1) \cdot (2\ell + n_{\text{rnd}} + 1) \cdot Q_{\mathcal{O}}}{2^{\lambda+1}} \ .$$

Analysis of the good transcripts. Fix a good transcript $\mathcal{Q} = (\mathcal{Q}_E, \mathcal{Q}_{\mathcal{O}})$. The probability to obtain this transcript in the ideal world is given by

$$\Pr^{(\mathsf{id})}[E \vdash \mathcal{Q}_E] \cdot \prod_{q \in \mathcal{Q}_{\mathcal{O}}} \Pr[(b^{(q)}, \boldsymbol{r}^{(q)}, r_{\text{com}}^{(q)}, \Gamma^{(q)}, \beta^{(q)})] = \frac{\Pr^{(\mathsf{id})}[E \vdash \mathcal{Q}_E]}{\big(2^\ell \cdot (2^\lambda)^\ell \cdot 2^{2\lambda} \cdot |\mathbb{K}|^{\eta \cdot n} \cdot 2^\kappa\big)^{Q_{\mathcal{O}}}} \ .$$

The probability to obtain this transcript in the real world is

$$\Pr^{(\mathsf{re})}[\forall q \in \mathcal{Q}_{\mathcal{O}} : (b^{(q)}, \boldsymbol{r}^{(q)}, r_{\text{com}}^{(q)}, \Gamma^{(q)}, \beta^{(q)}) \mid E \vdash \mathcal{Q}_E] \cdot \Pr^{(\mathsf{re})}[E \vdash \mathcal{Q}_E] \ .$$

We can express the left term of the above product as

$$\prod_{i=1}^{Q_{\mathcal{O}}} \Pr^{(\mathsf{re})}\left[(b^{(q_i)}, \boldsymbol{r}^{(q_i)}, r_{\mathrm{com}}^{(q_i)}, \Gamma^{(q_i)}, \beta^{(q_i)}) \mid E \vdash \mathcal{Q}_E \wedge \forall k < i : (b^{(q_k)}, \boldsymbol{r}^{(q_k)}, r_{\mathrm{com}}^{(q_k)}, \Gamma^{(q_k)}, \beta^{(q_k)})\right].$$

Note that the realization "$(b^{(q_i)}, \boldsymbol{r}^{(q_i)}, r_{\mathrm{com}}^{(q_i)}, \Gamma^{(q_i)}, \beta^{(q_i)})$" occurs if and only if, for all $j \in \{1, \ldots, \ell+1+n_{\mathrm{rnd}}\}$, we have $s_j^{(q_i)} = E_{(\mathsf{salt}\|j)}(h_j^{(q_i)})$. Since the transcript is good, there is no query in Q_E with the same cipher input that one that would be in the transcript (since KeyInputGuessing does not occur). Similarly, there is no query in Q_E with the same cipher output that one that would be in the transcript (since KeyOutputGuessing does not occur). Moreover, there is no previous transcripts with the same cipher input that one that would be in this transcript (since KeyInputCollision does not occur), while there is no previous transcripts with the same cipher output that one that would be in this transcript (since KeyOutputCollision does not occur). For all i, we thus have

$$\Pr^{(\mathsf{re})}[(b^{(q_i)}, \boldsymbol{r}^{(q_i)}, r_{\mathrm{com}}^{(q_i)}, \Gamma^{(q_i)}, \beta^{(q_i)}) \mid E \vdash \mathcal{Q}_E \wedge \forall k < i : (b^{(q_k)}, \boldsymbol{r}^{(q_k)}, r_{\mathrm{com}}^{(q_k)}, \Gamma^{(q_k)}, \beta^{(q_k)})] \geq \frac{1}{2^{\ell} \cdot (2^{\lambda})^{\ell} \cdot 2^{2\lambda} \cdot |\mathbb{K}|^{\eta \cdot n} \cdot 2^{\kappa}} .$$

It follows that

$$\Pr^{(\mathsf{re})}[\forall q \in \mathcal{Q}_{\mathcal{O}} : (b^{(q)}, \boldsymbol{r}^{(q)}, r_{\mathrm{com}}^{(q)}, \Gamma^{(q)}, \beta^{(q)}) \mid E \vdash \mathcal{Q}_E] \geq \left(\frac{1}{2^{\ell} \cdot (2^{\lambda})^{\ell} \cdot 2^{2\lambda} \cdot |\mathbb{K}|^{\eta \cdot n} \cdot 2^{\kappa}}\right)^{Q_{\mathcal{O}}} ,$$

and so the probability to obtain the transcript in the real world is at least the probability to obtain it in the ideal world. The proof finally holds from the above results applied to Equation (8). □

Remark 4. In this section and in the following ones, for the sake of simplicity, we assume that κ is a multiple of λ, namely $\kappa := n_{\mathrm{rnd}} \cdot \lambda$ with $n_{\mathrm{rnd}} \in \mathbb{N}$. If this is not the case, H^{rnd} can simply be constructed by truncating a few bits from the last cipher output. This truncation does not affect our security proofs. Indeed, in the ideal world, the truncated bits of $s_{n+1+n_{\mathrm{rnd}}}^{(q)}$ are sampled uniformly at random. The KeyOutputCollision event can therefore be analyzed using the same probabilistic argument. The KeyOutputGuessing event becomes even less likely, since the adversary would additionally need to correctly guess the truncated bits. In other words, the truncation transforms the problem of recovering these bits from a computational problem into a purely statistical guessing problem. On the other hand, the events KeyInputCollision and KeyInputGuessing are independent of the cipher outputs and are therefore unaffected by the truncation.

4.5 Security in the Ideal Cipher Model with Counter Mode Instantiation

The Davies-Meyer instantiation analyzed in the previous section is attractive for its computational performance when the underlying block cipher is instantiated with AES, which is much faster than hashing on modern CPU. In addition, the Davies-Meyer mode enables to factor most of the key schedules involved in the computation. Unfortunately, the security of the obtained construction relies on the ad hoc RGOW security assumption on the one-way function family $\mathcal{F}$. In this section, we still consider using a block cipher in the ideal cipher model, but in counter mode instead of Davies-Meyer, namely putting the secret as the cipher key and the domain separation as the cipher input. Concretely, the oracle functions H^{tree}, H^{com} and H^{rnd} are defined as:

$$\begin{aligned}
H^{\mathrm{tree}}(\mathsf{salt}, j, m) &= E_m(\mathsf{salt}, j) \\
H^{\mathrm{com}}(\mathsf{salt}, m) &= (E_m(\mathsf{salt}, \ell) \parallel E_m(\mathsf{salt}, \ell+1)) \\
H^{\mathrm{rnd}}(\mathsf{salt}, m) &= E_m(\mathsf{salt}, \ell+2) \parallel \ldots \parallel E_m(\mathsf{salt}, \ell+1+n_{\mathrm{rnd}})
\end{aligned}$$

where $\kappa = n_{\mathrm{rnd}} \cdot \lambda$ with $n_{\mathrm{rnd}} \in \mathbb{N}$ (*c.f.* Remark 4). Such a cipher mode instantiation is used by all the ongoing MPCitH-based round-2 NIST candidates except MQOM, and is further used in [KLS25]. We stress that the security analysis below is for the specific case of correlated GGM trees, which are not involved in these schemes.

Theorem 3. *Let $\mathcal{F}$ be a $(\lambda, t, Q, \varepsilon_{\mathrm{PGOW}}(t,Q))$-PGOW-secure family of one-way functions and let $(x,y) \in \mathcal{X} \times \mathcal{Y}$ be a random witness-statement instance of $\mathcal{F}$. Let E be an ideal cipher and let H^{tree}, H^{com}, H^{rnd} be oracle functions defined with respect to E as described above. Let $\mathcal{O}$ be a correlated tree oracle with respect to x, y, H^{tree}, H^{com} and H^{rnd}. Then, the maximum distinguishing advantage against $\mathcal{O}$ with query bounds $(Q_E, Q_{\mathcal{O}}, \mu)$ is upper-bound by:*

$$\varepsilon_{\mathrm{CTO}}(Q_E, Q_{\mathcal{O}}, \mu) \leq \varepsilon_{\mathrm{PGOW}}(t,Q) + \frac{(\mu-1)\cdot \ell \cdot Q_{\mathcal{O}}}{2^{\lambda+1}} + \frac{\left((\ell+1+n_{\mathrm{rnd}})\cdot Q_{\mathcal{O}}\right)^2}{2^{2\lambda+1}} \ ,$$

with $t := Q_{\mathcal{O}} \cdot Q_E \cdot (\ell+1+n_{\mathrm{rnd}}) + (Q_{\mathcal{O}} + Q_E)$ and $Q := Q_{\mathcal{O}} \cdot Q_E \cdot (\ell+1+n_{\mathrm{rnd}})$.

The events that allow an adversary to distinguish between the real world and the ideal world with the counter mode instantiation are exactly the same as for the proof in the ICM with Davies-Meyer mode. In particular, we again consider the two events KeyInputCollision and KeyOutputCollision, which occur when an input collision or an output collision happens between two calls to the ideal cipher made in the $\mathcal{O}$ queries. We also consider the events KeyInputGuessing and KeyOutputGuessing, which occur when the adversary successfully guesses a key-input pair or a key-output pair of an ideal-cipher call involved in an $\mathcal{O}$ query. The security analysis of KeyInputCollision and KeyInputGuessing is strictly identical to that of the proof with Davies-Meyer mode, since swapping the cipher key and the cipher input does not affect these events. The differences in the analysis arise for KeyOutputCollision and KeyOutputGuessing. The main difficulty for the analysis of the counter mode instantiation is that a key-output pair of a cipher call does not contain any domain separation, which is now encoded in the cipher input. While the probability of KeyOutputCollision can still be upper-bounded using standard probabilistic arguments, the reduction to PGOW arising for KeyOutputGuessing suffers from looseness. Indeed, without domain separation, the extractor can no longer precisely determine which $\mathcal{O}$ queries are compatible with a given E query. In the Davies-Meyer instantiation, domain separation enables to filter the $\mathcal{O}$ queries that could be combined with each E query, a property that is no longer available with counter mode.

Proof. We denote the transcript of $\mathcal{D}$'s interaction by $\mathcal{Q} = (\mathcal{Q}_E, \mathcal{Q}_{\mathcal{O}})$. A E-query $(\mathsf{key}, \mathsf{in}, \mathsf{out}) \in \mathcal{Q}_E$ satisfies $\mathsf{out} = E_{\mathsf{key}}(\mathsf{in})$. We also denote $s_j^{(q)}$ the output of the block cipher with $(\mathsf{salt}^{(q)}, j)$ as key in the query $q \in \mathcal{Q}_{\mathcal{O}}$. By structure of the construction, we have the following relations: for all j,

$$\begin{cases} u_{j,0}^{(q)} & = s_j^{(q)}, \text{ when } 1 \leq j < \ell \text{ (\textit{i.e.} } H^{\mathrm{tree}}) \\ r_{\mathrm{com},k}^{(q)} & = s_j^{(q)}, \text{when } j = \ell + k \text{ with } k \in \{0,1\} \text{ (in } H^{\mathrm{com}}), \\ r_{\mathrm{rnd},k}^{(q)} & = s_j^{(q)}, \text{when } j = \ell + 2 + k \text{ with } k \in \{0, \ldots, n_{\mathrm{rnd}} - 1\} \text{ (in } H^{\mathrm{rnd}}), \end{cases}$$

where $r_{\mathrm{com},k}^{(q)}$ (resp. $r_{\mathrm{rnd},k}^{(q)}$) is the kth λ-bit block of $r_{\mathrm{com}}^{(q)}$ (resp. $r_{\mathrm{rnd}}^{(q)}$). We only consider attainable transcripts.

We define the set of bad transcripts Bad as follows. A transcript $(\mathcal{Q}_E, \mathcal{Q}_{\mathcal{O}}) \in \mathsf{Bad}$ is a transcript for which one of the following events holds:

- KeyInputCollision: There are distinct queries $q, q' \in \mathcal{Q}_{\mathcal{O}}$ for the same salt (*i.e.* $\mathsf{salt}^{(q)} = \mathsf{salt}^{(q')}$) such that $h_j^{(q)} = h_j^{(q')}$ for some $j \in \{1, \ldots, \ell\}$.
- KeyOutputCollision: There are queries $q, q' \in \mathcal{Q}_{\mathcal{O}}$ such that $h_j^{(q)} = h_{j'}^{(q')}$ and $s_j^{(q)} = s_{j'}^{(q')}$ for some $j, j' \in \{1, \ldots, \ell + 1 + r_{\mathrm{rnd}}\}$ with $(q,j) \neq (q', j')$, where $h_j^{(q)} := h_\ell^{(q)}$ when $j > \ell$ (same for $h_j^{(q')}$).

– KeyInputGuessing: There is a query $q \in \mathcal{Q}_{\mathcal{O}}$ and a query $(\mathsf{key}, (\mathsf{salt}, j), \mathsf{out}) \in \mathcal{Q}_E$ such that $\mathsf{salt}^{(q)} = \mathsf{salt}$ and $\mathsf{key} = h_j^{(q)}$.
– KeyOutputGuessing: There is a query $q \in \mathcal{Q}_{\mathcal{O}}$ and a query $(\mathsf{key}, \mathsf{in}, \mathsf{out}) \in \mathcal{Q}_E$ such that $\mathsf{key} = h_j^{(q)}$ and $\mathsf{out} = s_j^{(q)}$.

Analysis of the bad transcripts in ideal world. The analysis of the KeyInputCollision event is strictly identical to that in the proofs of Theorems 1 and 2. We thus have

$$\Pr^{(\mathsf{id})}[\mathsf{KeyInputCollision}] \leq \frac{(\mu-1)\cdot\ell\cdot Q_{\mathcal{O}}}{2^{\lambda+1}}.$$

We bound the probability of the KeyOutputCollision event in the ideal world. Consider a pair of queries $q, q' \in \mathcal{Q}_{\mathcal{O}}$ and a pair of indexes $j, j' \in \{1, \ldots, \ell+1+r_{\mathrm{rnd}}\}$. By Equation (7), we have:

$$h_j^{(q)} = r^{(q)}_{\min(j,\ell)} \oplus \Big(\mathsf{First}_\lambda(x) \oplus \sum_{k<\min(j,\ell)} r_k^{(q)}\Big)$$

And by definition, we have:

$$s_j^{(q)} = \begin{cases} r^{(q)}_{j+1} \oplus (\ldots) & \text{when } 1 \leq j < \ell \\ r^{(q)}_{\mathrm{com},k} \oplus (\ldots) & \text{when } j = \ell + k \text{ with } k \in \{0,1\} \\ r^{(q)}_{\mathrm{rnd},k} \oplus (\ldots) & \text{when } j = \ell + 2 + k \text{ with } k \in \{0, \ldots, n_{\mathrm{rnd}} - 1\} \end{cases}$$

where the terms $(\ldots)$ depend on the other elements in q. Since $r_j^{(q)}, r_{j+1}^{(q)}, r^{(q)}_{\mathrm{com},k}$ and $r^{(q)}_{\mathrm{rnd},k}$ are uniformly random and independent of each other in the ideal world, we deduce that $(h_j^{(q)}, s_j^{(q)})$ is uniformly distributed over $\{0,1\}^\lambda \times \{0,1\}^\lambda$. As a result:

$$\Pr^{(\mathsf{id})}[h_j^{(q)} = h_{j'}^{(q')} \wedge s_j^{(q)} = s_{j'}^{(q')}] = \frac{1}{2^{2\lambda}}\ .$$

Summing over all the possible pairs of pairs $(q, j), (q', j')$, we get

$$\Pr^{(\mathsf{id})}[\mathsf{KeyOutputCollision}] \leq \frac{\big((\ell+1+n_{\mathrm{rnd}})\cdot Q_{\mathcal{O}}\big)^2}{2^{2\lambda+1}}$$

Let us now bound the probability of the Guessing event in the ideal world, where this event is the union event of KeyInputGuessing and KeyOutputGuessing. We will show that if this event occurs, then we can build an adversary $\mathcal{A}$ breaking the partial-guessing one-wayness of $\mathcal{F}$ with high probability. This adversary $\mathcal{A}$ takes as input y and queries the guessing oracle $\mathcal{G}$. Whenever $\mathcal{A}$ inputs $\hat{x}_0 = \mathsf{First}_\lambda(x)$ to $\mathcal{G}$, it wins the game. This adversary $\mathcal{A}$ simply calls the distinguisher $\mathcal{D}$ and simulates the responses to $\mathcal{D}$'s queries to E and $\mathcal{O}$ using the ideal-world simulator. Then $\mathcal{A}$ tests several candidates $\hat{x}_0$ based on $\mathcal{D}$'s transcript. We show below that if the event Guessing occurs in $\mathcal{D}$'s transcript, then one of $\mathcal{A}$'s candidates is the right $\mathsf{First}_\lambda(x)$, and hence $\mathcal{A}$ wins the PGOW security game.

The tested candidates are defined as follows. For each query $(\mathsf{key}, \mathsf{in}, \mathsf{out}) \in \mathcal{Q}_E$, each query $q \in \mathcal{Q}_{\mathcal{O}}$ and each $j \in \{1, \ldots, \ell+1+n_{\mathrm{rnd}}\}$, $\mathcal{A}$ tests the candidate $\hat{x}_0 := \mathsf{key} \oplus \sum_{k\leq\min(j,\ell)} r_k^{(q)}$. Indeed, if KeyInputGuessing or KeyOutputGuessing holds for such pair of queries, we have $\mathtt{key} = h_j^{(q)}$ for some index j, which implies

$$\hat{x}_0 := \mathsf{key} \oplus \sum_{k\leq\min(j,\ell)} r_k^{(q)} = h_j^{(q)} \oplus \sum_{k\leq\min(j,\ell)} r_k^{(q)} = \mathsf{First}_\lambda(x)\ .$$

The execution time of $\mathcal{A}$ is roughly of $Q_E + Q_{\mathcal{O}}$ to simulate $\mathcal{D}$'s execution in the ideal world, plus at most $Q_{\mathcal{O}} \cdot Q_E \cdot (\ell+1+n_{\mathrm{rnd}})$ to test the candidates $\hat{x}_0$. Therefore,

$$\Pr^{(\mathsf{id})}[\mathsf{Guessing}] = \Pr[\mathcal{A} \text{ succeeds}] \leq \varepsilon_{\mathrm{PGOW}}\big(t, Q\big)\ .$$

with $t := Q_{\mathcal{O}} \cdot Q_E \cdot (\ell+1+n_{\mathrm{rnd}}) + (Q_{\mathcal{O}} + Q_E)$ and $Q := Q_{\mathcal{O}} \cdot Q_E \cdot (\ell+1+n_{\mathrm{rnd}})$.

Analysis of the good transcripts. The analysis of the good transcripts is identical to the one in the proof of Theorem 2. □

4.6 Security in the Ideal Cipher Model with Enhanced Davies-Meyer Instantiation

As explained in Section 3, using the Davies-Meyer mode increases the attack surface, and requires the ad-hoc RGOW assumption which is easier to solve than the PGOW assumption to obtain security guarantees. On the other hand, while the counter mode does not require additional assumptions, it leads to a non-tight security reduction which is not satisfactory, although we did not identify any concrete attacks. In this section, we introduce a third construction to avoid relying on the RGOW assumption while still achieving a tight security reduction. This construction is inspired by the Davies-Meyer mode, but includes an additional cipher call to mitigate the attack paths identified in Section 3.

The issue depicted in Section 3 arises because the output blocks of H^{rnd} are used to mask values that (linearly) depend on the secret x. In Davies-Meyer mode, the feedback term $\sigma(m)$ also linearly depends on x, and thus the cipher outputs in H^{rnd} are in linear relation with x. We mitigate this issue by introducing an additional cipher call: instead of adding $\sigma(m)$ to prevent inversion of the cipher call $E_{(\mathsf{salt}\|j)}(m)$, we add the value $E_{(\mathsf{salt}\|\ell+n_{\mathrm{rnd}}+2)}(m)$.

Concretely, the oracle functions H^{tree}, H^{com}, and H^{rnd} are defined as follows:

$$
\begin{aligned}
H^{\mathrm{tree}}(\mathsf{salt}, j, m) &= E_{(\mathsf{salt}\|j)}(m) \oplus \sigma(m) \\
H^{\mathrm{com}}(\mathsf{salt}, m) &= \big(E_{(\mathsf{salt}\|\ell)}(m) \oplus \sigma(m)\big) \parallel \big(E_{(\mathsf{salt}\|\ell+1)}(m) \oplus \sigma(m)\big) \\
H^{\mathrm{rnd}}(\mathsf{salt}, m) &= E_{(\mathsf{salt}\|\ell+2)}(m) \oplus E_{(\mathsf{salt}\|\ell+n_{\mathrm{rnd}}+2)}(m) \parallel \ldots \parallel E_{(\mathsf{salt}\|\ell+n_{\mathrm{rnd}}+1)}(m) \oplus E_{(\mathsf{salt}\|\ell+n_{\mathrm{rnd}}+2)}(m)
\end{aligned}
$$

where $\kappa = n_{\mathrm{rnd}} \cdot \lambda$ with $n_{\mathrm{rnd}} \in \mathbb{N}$ (*c.f.* Remark 4). Compared to the standard Davies-Meyer instantiation, the index ind in the domain-separator string ranges in $\{1, \ldots, \ell + 2 + n_{\mathrm{rnd}}\}$ (*i.e.* with the extra "$\ell + n_{\mathrm{rnd}} + 2$" value) and is hence encoded on $|\mathsf{ind}| = \lceil \log_2(\ell + 2 + n_{\mathrm{rnd}}) \rceil$ bits.

The following theorem states the simulation security of the correlated tree oracle in the ideal cipher model with our enhanced Davies-Meyer instantiation.

Theorem 4. *Let $\mathcal{F}$ be a family of one-way functions and let $(x, y) \in \mathcal{X} \times \mathcal{Y}$ be a random witness-statement instance of $\mathcal{F}$. Assume that $\mathcal{F}$ is $(\lambda, t, Q, \varepsilon_{\mathrm{PGOW}}(t, Q))$-PGOW-secure. Let E be an ideal cipher and let H^{tree}, H^{com}, H^{rnd} be oracle functions defined with respect to E as described above. Let $\mathcal{O}$ be a correlated tree oracle with respect to x, y, H^{tree}, H^{com} and H^{rnd}. Then, the maximum distinguishing advantage against $\mathcal{O}$ with query bounds $(Q_E, Q_{\mathcal{O}}, \mu)$ is upper-bound by:*

$$
\varepsilon_{\mathrm{CTO}}(Q_E, Q_{\mathcal{O}}, \mu) \leq \varepsilon_{\mathrm{PGOW}}(t, Q) + \frac{(\mu - 1) \cdot (2\ell + n_{\mathrm{rnd}} + 2) \cdot Q_{\mathcal{O}}}{2^{\lambda+1}} + \frac{\mu \cdot Q_E}{2^{\lambda}} \ ,
$$

with $t := (2\mu + 1) \cdot Q_E + Q_{\mathcal{O}}$ and $Q := 2\mu \cdot Q_E$.

The security proof closely follows that of Theorem 2. Apart from extending the collision analysis to account for the additional cipher call, the only substantive difference lies in the treatment of $\mathsf{KeyOutputGuessing}$ for H^{rnd}. In this case, we rely on a probabilistic argument instead of a reduction to the RGOW security of $\mathcal{F}$.

Proof. We denote the transcript of $\mathcal{D}$'s interaction by $\mathcal{Q} = (\mathcal{Q}_E, \mathcal{Q}_{\mathcal{O}})$. A E-query $(\mathsf{key}, \mathsf{in}, \mathsf{out}) \in \mathcal{Q}_E$ satisfies $\mathsf{out} = E_{\mathsf{key}}(\mathsf{in})$. We also denote $s_j^{(q)}$ the output of the block cipher with index j for a query $q \in \mathcal{Q}_{\mathcal{O}}$, which *in the real world* is defined as:

$$
s_j^{(q)} = \begin{cases} E_{(\mathsf{salt}^{(q)}\|j)}(h_j^{(q)}) & \text{when } 1 \leq j < \ell \text{ (in } H^{\mathrm{tree}}) \\ E_{(\mathsf{salt}^{(q)}\|j)}(h_\ell^{(q)}) & \text{when } \ell \leq j \leq \ell + 2 + n_{\mathrm{rnd}} \text{ (in } H^{\mathrm{com}} \text{ or } H^{\mathrm{rnd}}) \end{cases}
$$

By structure of the construction, we have the following relations, for all $j \in \{1, \ldots, \ell+1+n_{\mathrm{rnd}}\}$ and $q \in \mathcal{Q}_{\mathcal{O}}$:

$$\begin{cases} u_{j,0}^{(q)} = s_j^{(q)} \oplus \sigma(h_j^{(q)}), & \text{when } 1 \leq j < \ell \text{ (in } H^{\mathrm{tree}}) \\ r_{\mathrm{com},k}^{(q)} = s_j^{(q)} \oplus \sigma(h_\ell^{(q)}), & \text{when } j = \ell + k \text{ with } k \in \{0,1\} \text{ (in } H^{\mathrm{com}}), \\ r_{\mathrm{rnd},k}^{(q)} = s_j^{(q)} \oplus s_{\ell+n_{\mathrm{rnd}}+2}^{(q)}, & \text{when } j = \ell + 2 + k \text{ with } k \in \{0, \ldots, n_{\mathrm{rnd}} - 1\} \text{ (in } H^{\mathrm{rnd}}), \end{cases}$$

where $r_{\mathrm{com},k}^{(q)}$ (resp. $r_{\mathrm{rnd},k}^{(q)}$) is the kth λ-bit block of $r_{\mathrm{com}}^{(q)}$ (resp. $r_{\mathrm{rnd}}^{(q)}$) and where

$$u_{j,0}^{(q)} = \begin{cases} r_{j+1}^{(q)} & \text{if } b_j^{(q)} = 0 \\ h_{j+1}^{(q)} & \text{if } b_j^{(q)} = 1 \end{cases}$$

In the ideal world, the hidden values $\{h_j^{(q)}\}$ are defined from the simulated revealed values $\{r_j^{(q)}\}$ and x from Equation (7). Regarding the cipher outputs, $s_{\ell+n_{\mathrm{rnd}}+2}^{(q)}$ is sampled uniformly at random, while the $\{s_j^{(q)}\}$, for $j < \ell + n_{\mathrm{rnd}} + 2$, are defined from $\{h_j^{(q)}\}$, $\{r_j^{(q)}\}$ and $s_{\ell+n_{\mathrm{rnd}}+2}^{(q)}$ using the above relations.

We define the set of bad transcripts Bad as in the proof of Theorem 2. A transcript $(\mathcal{Q}_E, \mathcal{Q}_{\mathcal{O}}) \in \mathsf{Bad}$ is a transcript for which one of the following events holds:

- $\mathsf{KeyInputCollision}$: There are distinct queries $q, q' \in \mathcal{Q}_{\mathcal{O}}$ for the same salt (*i.e.* $\mathsf{salt}^{(q)} = \mathsf{salt}^{(q')}$) such that $h_j^{(q)} = h_j^{(q')}$ for some $j \in \{1, \ldots, \ell\}$.
- $\mathsf{KeyOutputCollision}$: There are distinct queries $q, q' \in \mathcal{Q}_{\mathcal{O}}$ for the same salt salt (*i.e.* $\mathsf{salt}^{(q)} = \mathsf{salt}^{(q')}$) such that $s_j^{(q)} = s_j^{(q')}$ for some $j \in \{1, \ldots, \ell + 2 + n_{\mathrm{rnd}}\}$.
- $\mathsf{KeyInputGuessing}$: There is a query $q \in \mathcal{Q}_{\mathcal{O}}$ and a query $((\mathsf{salt} \parallel j), \mathsf{in}, \mathsf{out}) \in \mathcal{Q}_E$ such that $\mathsf{salt}^{(q)} = \mathsf{salt}$ and $\mathsf{in} = h_j^{(q)}$.
- $\mathsf{KeyOutputGuessing}$: There is a query $q \in \mathcal{Q}_{\mathcal{O}}$ and a query $((\mathsf{salt} \parallel j), \mathsf{in}, \mathsf{out}) \in \mathcal{Q}_E$ such that $\mathsf{salt}^{(q)} = \mathsf{salt}$ and $\mathsf{out} = s_j^{(q)}$.

Analysis of the bad transcripts in ideal world. The analysis of the $\mathsf{KeyInputCollision}$ event is strictly identical to the one in the previous proofs. We thus have

$$\Pr^{(\mathsf{id})}[\mathsf{KeyInputCollision}] \leq \frac{(\mu-1) \cdot \ell \cdot Q_{\mathcal{O}}}{2^{\lambda+1}}.$$

Regarding the analysis of the $\mathsf{KeyOutputCollision}$ event, this is identical to the analysis of $\mathsf{KeyOutputCollision}$ in the proof of Theorem 2, except that there is now $\ell + n_{\mathrm{rnd}} + 2$ cipher calls per query instead of $\ell + n_{\mathrm{rnd}} + 1$. We then have

$$\Pr^{(\mathsf{id})}[\mathsf{KeyOutputCollision}] \leq \frac{(\mu-1) \cdot (\ell + n_{\mathrm{rnd}} + 2) \cdot Q_{\mathcal{O}}}{2^{\lambda+1}}.$$

We bound the probability of the $\mathsf{KeyOutputGuessing}$ event in the ideal world for $j \geq \ell + 2$. For any given cipher query set $\mathcal{Q}_E$ and any assignment of the $\{r_{\mathrm{rnd},k}^{(q)}\}_{q \in \mathcal{Q}_{\mathcal{O}}, k}$ values, we have the two following probabilities that hold by the uniform randomness of $s_{\ell+n_{\mathrm{rnd}}+2}^{(q)}$ over $\{0,1\}^\lambda$. First, for any $\mathsf{salt} \in \{0,1\}^{|\mathsf{salt}|}$, we have

$$\begin{aligned} &\Pr^{(\mathsf{id})}[\exists q \in \mathcal{Q}_{\mathcal{O}} \text{ s.t. } \mathsf{salt}^{(q)} = \mathsf{salt} \text{ and } ((\mathsf{salt}, \ell + n_{\mathrm{rnd}} + 2), \star, s_{\ell+n_{\mathrm{rnd}}+2}^{(q)}) \in \mathcal{Q}_E] \\ &\quad \leq \sum_{q \in \mathcal{Q}_{\mathcal{O}} \,:\, \mathsf{salt}^{(q)} = \mathsf{salt}} \Pr^{(\mathsf{id})}[((\mathsf{salt}, \ell + n_{\mathrm{rnd}} + 2), \star, s_{\ell+n_{\mathrm{rnd}}+2}^{(q)}) \in \mathcal{Q}_E] \\ &\quad \leq \mu \cdot \frac{|\mathcal{Q}_E[(\mathsf{salt}, \ell + n_{\mathrm{rnd}} + 2)]|}{2^\lambda} \end{aligned}$$

where $\mathcal{Q}_E[\mathsf{key}]$ contains all the E queries where the cipher key is key. Moreover, we have for every $j = \ell+2+k$, with $k \in \{0, \ldots, n_{\mathrm{rnd}} - 1\}$,

$$\begin{aligned}
&\Pr^{(\mathsf{id})}[\exists q \in \mathcal{Q}_{\mathcal{O}} \text{ s.t. } \mathsf{salt}^{(q)} = \mathsf{salt} \text{ and } ((\mathsf{salt}, j), \star, s_j^{(q)}) \in \mathcal{Q}_E] \\
&\quad = \Pr^{(\mathsf{id})}[\exists q \in \mathcal{Q}_{\mathcal{O}} \text{ s.t. } \mathsf{salt}^{(q)} = \mathsf{salt} \text{ and } ((\mathsf{salt}, j), \star, s_{\ell+n_{\mathrm{rnd}}+2}^{(q)} \oplus r_{\mathrm{rnd},k}^{(q)}) \in \mathcal{Q}_E] \\
&\quad \leq \mu \cdot \frac{|\mathcal{Q}_E[(\mathsf{salt}, j)]|}{2^\lambda} \ .
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\Pr^{(\mathsf{id})}[\mathsf{KeyOutputGuessing} \text{ with } j \geq \ell + 2] \\
&\qquad \leq \sum_{\mathsf{salt}, j} \frac{\mu \cdot |\mathcal{Q}_E[(\mathsf{salt}, j)]|}{2^\lambda} \leq \frac{\mu \cdot Q_E}{2^\lambda}.
\end{aligned}$$

It remains to bound the probability of the union of the events $\mathsf{KeyInputGuessing}$ (for all j) and $\mathsf{KeyOutputGuessing}$ with $j \leq \ell + 1$. We rely on the same reduction to the PGOW security of $\mathcal{F}$ as in the proof of Theorem 2. We obtain

$$\Pr^{(\mathsf{id})}[\mathsf{KeyInputGuessing} \vee \mathsf{KeyOutputGuessing} \text{ with } j \leq \ell + 1] \leq \varepsilon_{\mathrm{PGOW}}\big(t, Q\big)$$

where $t := (2\mu + 1) \cdot Q_E + Q_{\mathcal{O}}$ and $Q := 2\mu \cdot Q_E$.

Summarizing, the probability of a bad transcript in the ideal world is upper-bounded by:

$$\Pr^{(\mathsf{id})}\big[(\mathcal{Q}_E, \mathcal{Q}_{\mathcal{O}}) \in \mathsf{Bad}\big] \leq \varepsilon_{\mathrm{PGOW}}(t, Q) + \frac{(\mu - 1) \cdot (2\ell + n_{\mathrm{rnd}} + 2) \cdot Q_{\mathcal{O}}}{2^{\lambda+1}} + \frac{\mu \cdot Q_E}{2^\lambda} \ .$$

Analysis of the good transcripts. The analysis of the good transcripts is identical to that in the proof of Theorem 2. We thus have that the probability to obtain a good transcript in the real world is at least the probability to obtain it in the ideal world. The proof then holds by the H-coefficient inequality (see (8)). □

Remark 5. An alternative construction would be to introduce an additional cipher call also for both H^{tree} and H^{com}. Although this modification would simplify the proof –since the entire $\mathsf{KeyOutputGuessing}$ event could then be handled via a purely probabilistic analysis– it would result in less efficient schemes, as it would require approximately $2N$ additional cipher encryptions, where N denotes the size of the underlying seed tree.

5 Unforgeability of MPC-in-the-Head Signatures with Correlated GGM Trees

In the previous section, we showed that the output of the correlated tree oracle is indistinguishable from a simulation based on fresh randomness, assuming that the underlying one-way function is PGOW secure. We now show that based on this indistinguishability, the underlying MPCitH signature scheme (under the general description of Section 2) is existentially unforgeable under chosen-message attacks (EUF-CMA) assuming that it is existentially unforgeable under key-only attacks (EUF-KO). In other words, we show that if an adversary cannot forge a valid signature given only the public key, then they also cannot forge a signature when additionally granted access to a signing oracle.

We stress that the EUF-KO security of such signature scheme does not depend on the internal structure of the GGM trees. It solely relies on the extractable binding property of the hash commitment H^{com} (in the ROM or the ICM) and on the soundness of the underlying PIOP. These are well-established security arguments which we do not restate here. Instead, we focus on the reduction from EUF-KO to EUF-CMA security, which relies on the correlated tree oracle analysis and constitutes the main technical contribution of our work.

$\underline{\mathsf{Sign}(\mathsf{sk} := (x, y), \mathsf{msg})}$:

1. $\mathsf{msalt} \leftarrow \{0,1\}^{|\mathsf{msalt}|}$
2. $\forall e \in [\tau] \ : \ \mathsf{salt}^{(e)} = (\mathsf{msalt}, e)$
3. $\forall e \in [\tau] \ : \ (\mathtt{ls_com}^{(e)}, \Delta_x^{(e)}, \mathtt{key}^{(e)}, P_x^{(e)}, P_u^{(e)}) = \mathsf{LCS.Commit}(\mathsf{salt}^{(e)}, x)$
4. $\mathsf{com}_1 = \mathrm{Hash}_1(\{\mathtt{ls_com}^{(e)}, \Delta_x^{(e)}\}_e)$
5. $\{\Gamma^{(e)}\}_e = \mathsf{XOF}_1(\mathsf{com}_1)$
6. $\forall e \in [\tau] \ : \ P_\alpha^{(e)} = P_u^{(e)} + \Gamma^{(e)} \cdot F_y(P_x^{(e)})$
7. $\mathsf{com}_2 = \mathrm{Hash}_2(\{P_\alpha^{(e)}\}_e, \mathsf{com}_1, \mathsf{msg})$
8. $\{i^{*(e)}\}_e = \mathsf{XOF}_2(\mathsf{com}_2)$
9. $\forall e \in [\tau] \ : \ (r_1^{(e)}, \ldots, r_\ell^{(e)}, r_{\mathrm{com}}^{(e)}) = \mathsf{LCS.Open}(\mathtt{key}^{(e)}, i^{*(e)})$
10. **return** $\sigma := (\mathsf{msalt}, \mathsf{com}_1, \mathsf{com}_2, \{P_\alpha^{(e)}, r_1^{(e)}, \ldots, r_\ell^{(e)}, r_{\mathrm{com}}^{(e)}, \Delta_x^{(e)}\}_e)$

Fig. 5: Signing algorithm.

5.1 Description of the Signature Scheme

The signature scheme whose security we establish follows the general MPC-in-the-Head paradigm described in Section 2. It is based on a one-way function family $\mathcal{F}$ (with associated arithmetization). The secret key is a witness-statement pair (x, y) for a random instance of $\mathcal{F}$, and the public key is y.

The signing algorithm is described in Figure 5. It relies on the line commitment scheme (LCS) and PIOP protocol introduced in Section 2. The protocol is repeated τ times and made non-interactive using the Fiat-Shamir transformation. Concretely, a commitment com_1 is first computed by hashing the τ LCS commitments. The batching challenges $\{\Gamma^{(e)}\}$ are then derived from com_1 using an XOF. The polynomials $\{P_\alpha^{(e)}\}$ are then computed and hashed together with com_1 and the message msg to obtain a second hash commitment com_2. The evaluation opening challenges $\{i^{*(e)}\}$ are then derived from com_2 using another XOF.

The LCS is composed of three routines:

- $\mathsf{LCS.Commit}$ takes as input a salt and the secret witness x, and outputs a leaf seeds' commitment $\mathtt{ls_com}$ (concatenation of all the leaf seed commitments, or a hash of it), an auxiliary value Δ_x, an opening $\mathtt{key}$ and the polynomials P_x and P_u. The opening key contains sufficient information to recompute the whole tree (namely at least the two master nodes).
- $\mathsf{LCS.Open}$ takes as input the opening $\mathtt{key}$ and an index i^*, and outputs the sibling path $(r_1, \ldots, r_\ell)$ and the commitment r_{com} to the hidden seed. These values, together with Δ_x, enable the verifier to partially reconstruct the tree (all the leaves except that of index i^*) to evaluate the polynomials P_x and P_u at ω_{i^*}, and to verify that they are consistent with the commitment $\mathtt{ls_com}$.
- $\mathsf{LCS.Eval}$ takes as input the salt, the opening index i^*, the auxiliary value Δ_x, and the opening $(r_1, \ldots, r_\ell, r_{\mathrm{com}})$, and outputs the evaluation of the committed polynomials at ω_{i^*} together with the recomputed commitment $\mathtt{ls_com}$.

The two first routines are used during signing, while the last one is used during verification. As discussed later, the MQOM v2 signature scheme [BBFR24] closely follow the above description.

5.2 EUF-CMA Security

The EUF-CMA security proof relies on the correlated tree oracle introduced in Section 4.1. The value β returned by the oracle is the pair (Δ_x, α_1) serialized as a κ-bit string. The associated function f is formally defined as:

$$f : (\mathsf{salt}, r_{\mathrm{rnd}}, x, y, b, \boldsymbol{r}) \mapsto \begin{bmatrix} \Delta_x \\ \alpha_1 \end{bmatrix} := \begin{bmatrix} \mathsf{NextBits}_\lambda(x) \oplus \sum_{i=0}^{N-1} [\![x]\!]_i \\ z_1 \oplus \sum_{i=0}^{N-1} [\![u]\!]_i \end{bmatrix} , \tag{10}$$

where i^* is the integer whose binary decomposition is b, the shares $\{([\![x]\!]_i, [\![u]\!]_i)\}_{i \neq i^*}$ are derived from $\boldsymbol{r} := (r_1, \ldots, r_\ell)$ in the same way as in $\mathsf{LCS.Eval}$, the value r_{rnd} is interpreted as $([\![x]\!]_{i^*}, [\![u]\!]_{i^*})$, and z_1 is the degree-1 coefficient of the polynomial $P_z = F_y(P_x)$. Here, P_x is the degree-1 polynomial whose leading coefficient is x and such that $P_x(\omega_{i^*})$ is obtained from $\mathsf{LCS.Eval}(\mathsf{salt}, i^*, r_1, \ldots, r_\ell, r_{\mathrm{com}}, \Delta_x)$.

The EUF-CMA security of the signature scheme is stated in the following theorem, whose proof is given below.

Theorem 5. *Let* Hash_1, Hash_2, XOF_1, XOF_2 *be modeled as random oracles. Let* $\mathcal{C}$ *be a cryptographic oracle (random oracle or ideal cipher) and let* H^{tree}, H^{com}, H^{rnd} *be oracle functions defined from* $\mathcal{C}$*. Let* $\mathcal{F}$ *be a one-way function family. Let* $\mathcal{O}$ *be a correlated tree oracle associated with* H^{tree}, H^{com}, H^{rnd}*, a random witness-statement instance of* $\mathcal{F}$*, and the function* f *defined by Equation* (10)*. Consider the signature scheme defined in Figure 5, with LCS based on the oracle functions* H^{tree}, H^{com}, H^{rnd}*. Let* $\varepsilon_{\text{EUF-KO}}$ *be the maximum advantage of winning the EUF-KO game against this signature scheme in time* t*. Then, for every* $\mu \in \mathbb{N}^*$*, the advantage* $\varepsilon_{\text{EUF-CMA}}$ *of any adversary running in time* t *against the EUF-CMA security of the same scheme is upper-bounded by:*

$$\varepsilon_{\text{EUF-CMA}}(t, Q_{\mathcal{C}}, Q_{\mathcal{H}}, Q_{sig}) \leq \varepsilon_{\text{EUF-KO}}(t, Q_{\mathcal{C}}, Q_{\mathcal{H}}) + \frac{2\, Q_{sig} \cdot Q_{\mathcal{H}}}{2^{2\lambda}} + \varepsilon_{\mathrm{MCOL}}(2^{|\mathsf{msalt}|}, \mu + 1, Q_{sig}) + \varepsilon_{\mathrm{CTO}}(Q_{\mathcal{C}}, Q_{\mathcal{O}}, \mu)\,, \quad (11)$$

with $Q_{\mathcal{O}} = \tau \cdot Q_{sig}$*, where* $Q_{\mathcal{H}}$*,* $Q_{\mathcal{C}}$*,* Q_{sig} *denotes the number of queries made by the adversary to the random oracles (including* Hash_1*,* Hash_2*,* XOF_1 *and* XOF_2*), the cryptographic oracle* $\mathcal{C}$ *and the signing oracle, respectively,* $\varepsilon_{\mathrm{CTO}}(Q_{\mathcal{C}}, Q_{\mathcal{O}}, \mu)$ *denotes the maximum distinguishing advantage against* $\mathcal{O}$ *with query bounds* $(Q_{\mathcal{C}}, Q_{\mathcal{O}}, \mu)$ *and* $\varepsilon_{\mathrm{MCOL}}(2^s, \mu, Q)$ *denotes the probability of obtaining a* μ*-collision when sampling* Q *random elements uniformly from a set of size* 2^s*.*

Proof. Assume that there exists an EUF-CMA adversary $\mathcal{A}$ running in time t and outputting a valid forgery with probability $\varepsilon_{\text{EUF-CMA}}$. We construct an adversary $\mathcal{B}$ against EUF-KO security that runs in time t and succeeds with probability $\varepsilon_{\text{EUF-KO}}$ verifying (11).

The EUF-KO adversary $\mathcal{B}$ is given the public key y and aims to produce a valid forgery under this key. It interacts with $\mathcal{A}$ by simulating the EUF-CMA game with respect to the same public key. In particular, $\mathcal{B}$ answers all hash queries made by $\mathcal{A}$ using lazy sampling, while recording each query in a list $\mathcal{Q}_H$. When $\mathcal{A}$ outputs a purported forgery σ, adversary $\mathcal{B}$ forwards the same signature σ as its own forgery in the EUF-KO game.

It remains to explain how adversary $\mathcal{B}$ simulates the signing oracle. To this end, we consider the following sequence of games.

- $\mathsf{Game\ 0}$. In this game, adversary $\mathcal{B}$ is additionally given the secret key and answers all signing queries of $\mathcal{A}$ using the genuine signing algorithm. Hence, the simulation is perfect. The probability that $\mathcal{B}$ outputs a valid forgery is therefore exactly the success probability of $\mathcal{A}$, namely $\varepsilon_0 = \varepsilon_{\text{EUF-CMA}}$.
- $\mathsf{Game\ 1}$. In this game, instead of running the original signing algorithm verbatim, $\mathcal{B}$ uses the modified procedure described in Figure 6. This procedure is functionally equivalent to the genuine signing algorithm, with the following difference: the Fiat-Shamir challenges $\Gamma^{(e)}$ and $i^{*(e)}$ are sampled at the beginning of the execution, and the random oracles XOF_1 and XOF_2 are later programmed to output the preselected challenges on the appropriate input. Apart from this reordering, the signing process remains unchanged. Some steps are reorganized and isolated into a subroutine, and the computation of P_α involves partial recomputation: the degree-1 coefficient is first determined, and the full polynomial P_α is then recovered using the evaluation-opening procedure, as in the verification algorithm. While programming the random oracles, the adversary $\mathcal{B}$ aborts whenever the programmed input has already been queried by $\mathcal{A}$, which would cause an inconsistency in the simulation. Whenever $\mathcal{B}$ does not abort, the distribution of the challenges simulated signatures is identical to that of the original signing algorithm. Hence, the success probability in this game satisfies:

$$\varepsilon_1 \geq \varepsilon_0 - \varepsilon_{\mathrm{ABORT}}$$

$\underline{\mathsf{Sign}(\mathsf{sk} := (x, y), \mathsf{msg})}$:

1. $\mathsf{msalt} \leftarrow \{0,1\}^{|\mathsf{msalt}|}$
2. $\forall e \in [\tau] \,:\, \mathsf{salt}^{(e)} = (\mathsf{msalt}, e)$
3. $\forall e \in [\tau] \,:\, (i^{*(e)}, r_1^{(e)}, \ldots, r_\ell^{(e)}, r_{\mathrm{com}}^{(e)}, \Delta_x^{(e)}, \alpha_1^{(e)}, \Gamma^{(e)}) \leftarrow \mathsf{Routine}(\mathsf{salt}^{(e)}, x)$
4. $\forall e \in [\tau] \,:\, (\mathtt{ls_com}^{(e)}, p_x^{(e)}, p_u^{(e)}) = \mathsf{LCS.Eval}((\mathsf{msalt}, e), i^{*(e)}, r_1^{(e)}, \ldots, r_\ell^{(e)}, r_{\mathrm{com}}^{(e)}, \Delta_x^{(e)})$
5. $\forall e \in [\tau] \,:\, p_\alpha^{(e)} = p_u^{(e)} + \Gamma^{(e)} \cdot F_y(p_x^{(e)})$
6. $\forall e \in [\tau] \,:\, \textbf{interpolate}\ P_\alpha^{(e)}$ from $\alpha_1^{(e)}$ and $P_\alpha^{(e)}(\omega_{i^{*(e)}}) := p_\alpha^{(e)}$.
7. $\mathsf{com}_1 = \mathrm{Hash}_1(\{\mathtt{ls_com}^{(e)}, \Delta_x^{(e)}\}_e)$
8. $\textbf{program}\ \mathsf{XOF}_1(\mathsf{com}_1) := \{\Gamma^{(e)}\}_e$
9. $\mathsf{com}_2 = \mathrm{Hash}_2(\{P_\alpha^{(e)}\}_e, \mathsf{com}_1, \mathsf{msg})$
10. $\textbf{program}\ \mathsf{XOF}_2(\mathsf{com}_2) := \{i^{*(e)}\}_e$
11. $\textbf{return}\ \sigma := (\mathsf{msalt}, \mathsf{com}_1, \mathsf{com}_2, \{P_\alpha^{(e)}, r_1^{(e)}, \ldots, r_\ell^{(e)}, r_{\mathrm{com}}^{(e)}, \Delta_x^{(e)}\}_e)$

$\underline{\mathsf{Routine}(\mathsf{salt}^{(e)}, x)}$:

1. $i^{*(e)} \leftarrow [N]$
2. $\Gamma^{(e)} \leftarrow \mathbb{K}^{\eta \times m}$
3. $(\mathtt{ls_com}^{(e)}, \Delta_x^{(e)}, \mathtt{key}^{(e)}, P_x^{(e)}, P_u^{(e)}) = \mathsf{LCS.Commit}(\mathsf{salt}^{(e)}, x)$
4. $(r_1^{(e)}, \ldots, r_\ell^{(e)}, r_{\mathrm{com}}^{(e)}) = \mathsf{LCS.Open}(\mathtt{key}^{(e)}, i^{*(e)})$
5. $P_\alpha^{(e)} \leftarrow P_u^{(e)} + \Gamma^{(e)} \cdot F_y(P_x^{(e)})$
6. $\alpha_1^{(e)} = \mathsf{coeff}_1(P_\alpha^{(e)})$
7. $\textbf{return}\ (i^{*(e)}, r_1^{(e)}, \ldots, r_\ell^{(e)}, r_{\mathrm{com}}^{(e)}, \Delta_x^{(e)}, \alpha_1^{(e)}, \Gamma^{(e)})$

Fig. 6: Intermediary signing algorithm (Game 1).

where $\varepsilon_{\text{ABORT}}$ is the probability that $\mathcal{B}$ aborts due to an inconsistency when programming the random oracles. Let us now upper bound this probability.

We consider a concrete signing query made by $\mathcal{A}$, and we analyze the probability that $\mathcal{B}$ aborts during the simulation of this query. Let us denote by $\mathcal{Q}_{\mathrm{Hash}_1}$, $\mathcal{Q}_{\mathrm{Hash}_2}$, $\mathcal{Q}_{\mathsf{XOF}_1}$, $\mathcal{Q}_{\mathsf{XOF}_2}$, the sets of previous queries to the random oracles Hash_1, Hash_2, XOF_1, XOF_2, respectively (counting the indirect queries made by the signing oracle). Consider the values $\mathsf{in}_1 = \{\mathtt{ls_com}^{(e)}, \Delta_x^{(e)}\}_e$, com_1 and com_2 in the ongoing signature. We have:[4]

$$
\begin{aligned}
&\Pr\left[\mathsf{com}_1 \in \mathcal{Q}_{\mathsf{XOF}_1} \vee \mathsf{com}_2 \in \mathcal{Q}_{\mathsf{XOF}_2}\right] \\
&\quad\le \Pr\left[\mathsf{com}_1 \in \mathcal{Q}_{\mathsf{XOF}_1} \vee (*, \mathsf{com}_1, *) \in \mathcal{Q}_{\mathrm{Hash}_2}\right] \\
&\qquad + \Pr\left[\mathsf{com}_2 \in \mathcal{Q}_{\mathsf{XOF}_2} \mid (*, \mathsf{com}_1, *) \notin \mathcal{Q}_{\mathrm{Hash}_2}\right] \\
&\quad\le \underbrace{\Pr\left[\mathsf{com}_1 \in \mathcal{Q}_{\mathsf{XOF}_1} \vee (*, \mathsf{com}_1, *) \in \mathcal{Q}_{\mathrm{Hash}_2} \mid \mathsf{in}_1 \notin \mathcal{Q}_{\mathrm{Hash}_1}\right]}_{\le (|\mathcal{Q}_{\mathsf{XOF}_1}| + |\mathcal{Q}_{\mathrm{Hash}_2}|)/2^{2\lambda}} \\
&\qquad + \underbrace{\Pr\left[\mathsf{in}_1 \in \mathcal{Q}_{\mathrm{Hash}_1}\right]}_{\le |\mathcal{Q}_{\mathrm{Hash}_1}|/|\mathcal{T}|} + \underbrace{\Pr\left[\mathsf{com}_2 \in \mathcal{Q}_{\mathsf{XOF}_2} \mid (*, \mathsf{com}_1, *) \notin \mathcal{Q}_{\mathrm{Hash}_2}\right]}_{\le |\mathcal{Q}_{\mathsf{XOF}_2}|/2^{2\lambda}}
\end{aligned}
$$

where $\mathcal{T}$ denotes the space of possible inputs $\{\mathtt{ls_com}^{(e)}, \Delta_x^{(e)}\}_e$ to Hash_1. If $\mathsf{in}_1 \notin \mathcal{Q}_{\mathrm{Hash}_1}$, then com_1 is uniformly sampled in the ongoing signature query which leads to the upper bound of the first term. If $(*, \mathsf{com}_1, *) \notin \mathcal{Q}_{\mathrm{Hash}_2}$, then com_2 is uniformly sampled in the ongoing signature query which leads to the upper bound of the third term. Then the second term is upper-bounded by the probability that

[4] The first inequality holds since for any three events A, B, C, with $\Pr[\neg C] > 0$, we have $\Pr[A \vee B] \le \Pr[A \vee C] + \Pr[B \wedge \neg C]$ and $\Pr[B \wedge \neg C] \le \Pr[B \mid \neg C]$.

in_1, which is freshly sampled from $\mathcal{T}$, coincides with one of the previous inputs to Hash_1. While it is hard to precisely evaluate the size of $\mathcal{T}$, we have that $\{\mathtt{ls_com}^{(e)}, \Delta_x^{(e)}\}_e$ is obtained by expanding fresh randomness over a space of size $2^{|\mathsf{msalt}|} \cdot (2^\lambda)^\tau$ (made of the master salt and the τ master seeds). While the expansion procedure could in theory cause some collisions, we can reasonably expect that the size of $\mathcal{T}$ is close to $2^{|\mathsf{msalt}|} \cdot (2^\lambda)^\tau$ and clearly $|\mathcal{T}| \gg 2^{2\lambda}$. We deduce:

$$\Pr\left[\mathsf{com}_1 \in \mathcal{Q}_{\mathsf{XOF}_1}\right] \leq \frac{|\mathcal{Q}_{\mathrm{Hash}_1}| + |\mathcal{Q}_{\mathrm{Hash}_2}| + |\mathcal{Q}_{\mathsf{XOF}_1}| + |\mathcal{Q}_{\mathsf{XOF}_2}|}{2^{2\lambda}} \ .$$

The probability of abortion over all the signature queries is then upper-bounded by

$$\varepsilon_{\mathrm{ABORT}} \leq \frac{Q_{\mathrm{sig}} \cdot Q_{\mathcal{H}}}{2^{2\lambda}} \ .$$

- Game 2. We replace the subroutine in Figure 6 by a call to the correlated tree oracle $\mathcal{O}$ defined in Figure 4, where the function f is defined as in Equation (10). By construction of f, the behavior of $\mathcal{O}$ is identical to that of the replaced subroutine. Therefore, this modification is purely syntactic and does not change the distribution of the simulated signatures. The adversary's success probability in this game remains unchanged: $\varepsilon_2 = \varepsilon_1$.
- Game 3. The adversary $\mathcal{B}$ aborts if a master salt value is sampled strictly more than μ times, *i.e.*, if a $(\mu+1)$-collision occurs in the master salt space. The probability of such an event is bounded by $\varepsilon_{\mathrm{MCOL}}(2^{|\mathsf{msalt}|}, \mu+1, Q_{\mathrm{sig}})$. Apart from this abort condition, the game is identical to the previous one. We hence have:
$$\varepsilon_3 \geq \varepsilon_2 - \varepsilon_{\mathrm{MCOL}}(2^{|\mathsf{msalt}|}, \mu+1, Q_{\mathrm{sig}}) \ .$$
- Game 4. We replace calls to the oracle $\mathcal{O}$ by calls to its simulator. The distinguishing gap between this game and the previous one is bounded by the distinguishing advantage against $\mathcal{O}$, implying
$$\varepsilon_{\mathrm{CTO}}(Q_{\mathcal{C}}, Q_{\mathcal{O}}, \mu),$$
where $Q_{\mathcal{O}} = \tau \cdot Q_{\mathrm{sig}}$ and μ bounds the maximal number of calls using the same salt. After this replacement, the signing simulation no longer depends on the secret key and we have $\varepsilon_4 \geq \varepsilon_3 - \varepsilon_{\mathrm{CTO}}(Q_{\mathcal{C}}, Q_{\mathcal{O}}, \mu)$.
- Game 5. Finally, $\mathcal{B}$ aborts if $\mathcal{A}$ makes a query to Hash_2 that output a value com_2 contained in a previously generated signature. In case no such abortion occurs, the forged signature output by $\mathcal{A}$ must contain a value com_2 that has never been generated during the signing simulation, and the forgery is hence valid in the EUF-KO game. We have:
$$\varepsilon_{\text{EUF-KO}} = \varepsilon_5 \geq \varepsilon_4 - \frac{Q_{\mathrm{sig}} \cdot Q_{\mathcal{H}}}{2^{2\lambda}} \ .$$

Summing up the bounds incurred in each game transition, we obtain (11) which concludes the proof. □

5.3 Application to MQOM

The ongoing NIST candidate MQOM v2 is very close to the signature scheme described in Figure 5, with two main differences: (1) it uses a single Fiat-Shamir challenge Γ for all the LCS instances, and (2) it incorporates a grinding procedure in XOF_2 to boost the soundness of the evaluation opening. In our context, taking (1) into account would mean considering that the correlated tree oracle $\mathcal{O}$ generates openings by batch of τ with the same Γ. Such a change has no incidence on our security analysis. On the other hand, grinding consists in making more calls to XOF_2 in the signing algorithm (by including some small proof of work), which improves the EUF-KO security bound but has no impact on the EUF-CMA security analysis. In summary, both changes do not affect the validity of our security theorems.

The EUF-CMA security of MQOM v2 is obtained by plugging the result of Theorem 2 (simulation security of the correlated tree oracle in the ICM with Davies-Meyer instantiation) for ε_{CTO} into Theorem 5. This gives the following bound on the EUF-CMA advantage of MQOM v2:

$$\varepsilon_{\text{EUF-CMA}} \leq \varepsilon_{\text{EUF-KO}} + \frac{2\,Q_{\text{sig}} \cdot Q_{\mathcal{H}}}{2^{2\lambda}} + \varepsilon_{\text{MCOL}}(2^{|\mathsf{msalt}|}, \mu+1, Q_{\text{sig}}) \\ + \varepsilon_{\text{PGOW}}(t, Q) + \varepsilon_{\text{RGOW}}(t', Q') + \frac{(\mu-1)\cdot(2\ell + n_{\text{rnd}} + 1)\cdot Q_{\mathcal{O}}}{2^{\lambda+1}}$$

with $Q_{\mathcal{O}} = \tau \cdot Q_{\text{sig}}$, $t = (2\mu+1)\cdot Q_E + Q_{\mathcal{O}}$, $Q = 2\mu \cdot Q_E$, $t' = (2\mu+1)\cdot Q_E + Q_{\mathcal{O}}$ and $Q' = \mu \cdot Q_E$, where Q_E denotes the number of queries made by the adversary to the ideal cipher.

Tweaking MQOM v2 with the enhanced Davies-Meyer instantiation of Section 4.6 to avoid the RGOW assumption, we obtain the following EUF-CMA security bound (by plugging the result of Theorem 4 into Theorem 5):

$$\varepsilon_{\text{EUF-CMA}} \leq \varepsilon_{\text{EUF-KO}} + \frac{2\,Q_{\text{sig}} \cdot Q_{\mathcal{H}}}{2^{2\lambda}} + \varepsilon_{\text{MCOL}}(2^{|\mathsf{msalt}|}, \mu+1, Q_{\text{sig}}) \\ + \varepsilon_{\text{PGOW}}(t, Q) + \frac{(\mu-1)\cdot(2\ell + n_{\text{rnd}} + 2)\cdot Q_{\mathcal{O}}}{2^{\lambda+1}} + \frac{\mu \cdot Q_E}{2^{\lambda}}$$

Bounding Multi-Collisions. In MQOM, the master salt is $\lambda - 12$ bits long (with 12 bits dedicated to domain separation). Assuming that the maximal number Q_{sig} of queries is upper bounded by 2^{64} (as required in the NIST call), we obtain a negligible probability that the largest multi-collision in the master salt space is greater than $\mu = 4$. More precisely, using the Chernoff Bound, we have (see for example [KLS24, Section 2.2]):

$$\varepsilon_{\text{MCOL}}(2^{|\mathsf{msalt}|}, \mu+1, Q_{\text{sig}}) < \left(\frac{2eQ_{\text{sig}}}{5\cdot 2^{|\mathsf{msalt}|}}\right)^5 \cdot 2^{|\mathsf{msalt}|} \leq \frac{Q_{\text{sig}}^5}{2^{4|\mathsf{msalt}|}} < \frac{1}{2^{\lambda}}$$

for $\lambda \in \{128, 192, 256\}$ whenever $|\mathsf{msalt}| = \lambda - 12$ and $Q_{\text{sig}} \leq 2^{64}$.

PGOW Security. As briefly discussed in Section 3, the PGOW security of MQOM can be easily reduced to the hardness of solving the MQ problem underlying the secret key, namely with n variables and m equations. Let us denote t_{WMQ} the running time of solving a *weak* MQ instance with $n - \lambda'$ variables and m equations (with probability close to 1), where $\lambda' = \lambda / \log_2 |\mathbb{F}|$. In that case, if we have an adversary against the PGOW problem running in time t_{PGOW} and making Q queries to the guessing oracle, we can build an adversary against the MQ problem with n variables and m equations running in time $t_{\text{PGOW}} + Q \cdot t_{\text{WMQ}}$, by simply reducing the original MQ system into a simpler one using the output of the PGOW adversary. Concretely, the adversary $\mathcal{A}_{\text{MQ}}$ against MQ runs $\mathcal{A}_{\text{PGOW}}$ and, for each query $\hat{x}_0$ made by $\mathcal{A}_{\text{PGOW}}$ to the guessing oracle, $\mathcal{A}_{\text{MQ}}$ attempts solving the weak MQ instance obtained by fixing the λ' first variables to the value of $\hat{x}_0$ and keeping the $n - \lambda'$ last variables as unknowns. The success probability of this adversary is the same as that of $\mathcal{A}_{\text{PGOW}}$, which implies:

$$\varepsilon_{\text{PGOW}}(t_{\text{PGOW}}, Q) \leq \varepsilon_{\text{MQ}}(t_{\text{MQ}}) \quad \text{with} \quad t_{\text{MQ}} := t_{\text{PGOW}} + Q \cdot t_{\text{WMQ}}$$

for every Q and t_{PGOW}. The above inequality further implies:

$$\frac{\varepsilon_{\text{MQ}}}{t_{\text{MQ}}} \leq 2^{-\kappa} \quad \Longrightarrow \quad \frac{\varepsilon_{\text{PGOW}}}{t_{\text{PGOW}}} \;\leq\; 2^{-\kappa} \cdot \left(\frac{t_{\text{MQ}}}{t_{\text{PGOW}}}\right) \;=\; 2^{-\kappa} \cdot \left(1 + \frac{Q \cdot t_{\text{WMQ}}}{t_{\text{PGOW}}}\right) .$$

In other words, κ bits of MQ security implies at least $\kappa - \log_2\left(1 + \frac{Q \cdot t_{\text{WMQ}}}{t_{\text{PGOW}}}\right)$ bits of PGOW security.

To interpret the above inequality, let us first consider the case of a small Q so that $Q \cdot t_{\text{WMQ}} \ll t_{\text{PGOW}}$. In this case, we have that breaking the PGOW problem essentially consists in breaking the MQ problem.

If on the other hand, the running time of $\mathcal{A}_{\text{PGOW}}$ is dominated by the calls to the guessing oracle, namely $t_{\text{PGOW}} \approx Q \cdot t_{\mathcal{G}}$ for $t_{\mathcal{G}}$ denoting the running time of the guessing oracle, we get a loss of $\log_2\left(1 + \frac{t_{\text{WMQ}}}{t_{\mathcal{G}}}\right)$ bits of security (assuming $t_{\text{WMQ}} > t_{\mathcal{G}}$). This loss depends on the relative cost of solving a weak MQ instance and making a query to the guessing oracle. In a practical setting, such a query amounts evaluating one node derivation function of the GGM tree – typically one AES block – which is commonly estimated to $t_{\mathcal{G}} \approx 2^{15}$ elementary operations. On the other hand, the solving time of a weak MQ instance t_{WMQ} might be greater than this (though significantly smaller than the solving time of the original MQ instance), which might result in a small security gap.

Fortunately, this security gap seems purely inherited from the non-tightness of the above PGOW to MQ reduction, and not from an intrinsic weakness of the PGOW problem. Obvious ways to attack the PGOW problem are:

1. by directly solving the original MQ system and making a single query to the guessing oracle with the λ' first variables of the solution;
2. by exhaustively searching for the λ' first variables of the secret MQ solution and making a query to the guessing oracle for each guess.

The complexity of the first attack is similar to that of solving the original MQ system while that of the second attack is $2^{\lambda} \cdot t_{\mathcal{G}}$ with $t_{\mathcal{G}} \approx 2^{15}$ in practice, which precisely matches the desired security for the underlying MQ problem while targeting NIST levels 1, 3 or 5 (*i.e.* exhaustive search of an AES secret key, estimated to $\lambda+15$ bits of security). Moreover, it seems unlikely that a significantly more efficient attack against PGOW exists. This would implicitly imply that an efficient attack against MQ exists that consists in partially solving the MQ system to recover (a candidate for) the λ' first variables of the solution, and then solving the remaining weak MQ instance. Should such an attack exist, it would effectively be speeded up by the availability of the guessing oracle in the PGOW problem (in place of solving the weak MQ instance). Fortunately, no such attack is known, and recovering the λ' first variables of the MQ solution seems essentially as hard as solving the original MQ system.

RGOW (In)Security. Let us analyze the RGOW security of MQOM. As introduced in Section 3, for each query to the relation oracle, the adversary get functions $g_1, \ldots, g^{(i)}_{n_x+n_u-1}$ for which they can try to guess $g_k(x)$, where x is the secret MQ solution.

The $n_x - 1$ first functions are the same for all the signatures (new signatures do not provide different relations), which is why we drop the superscript index i for those functions. They are defined as

$$g_k : x \mapsto \sigma(\mathsf{First}_\lambda(x)) \oplus \mathsf{Block}_\lambda^{(k+1)}(x)$$

for $1 \leq k \leq n_x - 1$. For those functions, we have a similar situation in the PGOW case, where the adversary can try to guess some value $g_k(x)$ to obtain a weak MQ instance with $n - \lambda'$ variables and m equations, where $\lambda' = \lambda / \log_2 |\mathbb{F}|$.

Let us now deal with the signature-dependent functions $g_k(x)$ for $k \geq n_x$. These functions are defined as:

$$g_k : x \mapsto \sigma(\mathsf{First}_\lambda(x)) \oplus \mathsf{Block}_\lambda^{(k+1-n_x)}(\Gamma \cdot z_1) \ .$$

In the context of MQOM, a packing technique is used (see [BBFR24, Sect. 2.1.2]) which implies that the m coordinates of F_y are defined over the extension field $\mathbb{K}$ of $\mathbb{F}$, thus reducing the number of coordinates of z_1 to $\hat{m} = m/\mu$, where $\mu = [\mathbb{K} : \mathbb{F}]$. The matrices and vectors defining the coordinates of F_y are denoted $\hat{A}_j \in \mathbb{K}^{n \times n}$ and $\hat{b}_j \in \mathbb{K}^n$ for $1 \leq j \leq \hat{m}$. In this context, we have:

$$\forall j \in [\hat{m}],\ z_{1,j} = (P_x(\omega)^T \cdot (\hat{A}_j^T + \hat{A}_j) - w \cdot \hat{b}_j^T) \cdot x + \hat{b}_j^T \cdot P_x(\omega) \ .$$

Let us define the matrices $M \in \mathbb{K}^{\hat{m} \times n}$ and $O \in \mathbb{K}^{\hat{m} \times n}$ as

$$M := \begin{pmatrix} P_x(\omega)^T \cdot (\hat{A}_1^T + \hat{A}_1) - \omega \cdot \hat{b}_1^T \\ \vdots \\ P_x(\omega)^T \cdot (\hat{A}_{\hat{m}}^T + \hat{A}_{\hat{m}}) - \omega \cdot \hat{b}_{\hat{m}}^T \end{pmatrix} \quad \text{and} \quad O := \begin{pmatrix} -\omega \cdot \hat{b}_1^T \\ \vdots \\ -\omega \cdot \hat{b}_{\hat{m}}^T \end{pmatrix} .$$

Let us denote $\hat{\lambda}' := \lambda'/\mu = \lambda/\log_2|\mathbb{K}|$ (where $\mu = [\mathbb{K} : \mathbb{F}]$) and $\Phi^{(\hat{r})}_{\mathbb{K}\to\mathbb{F}} : \mathbb{K}^{\hat{r}\times n} \to \mathbb{F}^{(\hat{r}\cdot\mu)\times n}$ the linear application that transforms each row of the input matrix into μ rows of the output matrix in the canonical $\mathbb{F}$-basis of $\mathbb{K}$. When the dimensions are clear from the context, we omit the superscript $(\hat{r})$. Let M' be the full-rank matrix that corresponds to the linear application $x \mapsto \sigma(\mathsf{First}_\lambda(x))$ and let Γ_k the kth block of $\hat{\lambda}'$ rows of Γ. We have:

$$g_k : x \mapsto \Phi_{\mathbb{K}\to\mathbb{F}}(\Gamma_k \cdot M + M') \cdot x + \Phi_{\mathbb{K}\to\mathbb{F}}(\Gamma_k \cdot O) \ .$$

If $\Phi_{\mathbb{K}\to\mathbb{F}}(\Gamma_k \cdot M + M')$ is full rank (namely of rank λ'), then g_k is a affine transformation with an image of size $|\mathbb{F}|^{\lambda'} = 2^\lambda$. This case is equivalent to that of the $n_x - 1$ first functions and the PGOW problem (a correct guess for $g_k(x)$ gives a weak MQ instance with $n - \lambda'$ variables and m equations). On the other hand, in case of rank deficiency in $\Phi_{\mathbb{K}\to\mathbb{F}}(\Gamma_k \cdot M + M')$, the situation changes. Let us assume that the rank of $\Phi_{\mathbb{K}\to\mathbb{F}}(\Gamma_k \cdot M + M')$ is $\lambda' - \delta$. In that case, the image of g_k is $|\mathbb{F}|^\delta$ *smaller*, meaning that an exhaustive search for $g_k(x)$ only cost $2^\lambda/|\mathbb{F}|^\delta$. In practice, an adversary can get many such g_k with different Γ_k, M and M' and just needs to find one with the desired rank deficiency. For a target δ, the attack runs as follows:

- Generate many signatures such that having a relation with a rank deficiency of δ, *i.e.* $\text{rank}(\Phi_{\mathbb{K}\to\mathbb{F}}(\Gamma_k \cdot M + M')) \leq \lambda' - \delta$.
- Test the $|\mathbb{F}|^{\lambda'-\delta} = 2^\lambda/|\mathbb{F}|^\delta$ possible candidates for $g_k(x)$ using the guessing oracle (which is one AES call in practice).

The adversary needs to collect $\frac{1}{p_\delta}$ relations in average to get one with the target rank deficiency, where p_δ is the probability to obtain the desired rank deficiency. The attack cost is thus of

$$\frac{1}{p_\delta} + \frac{2^\lambda}{|\mathbb{F}|^\delta}.$$

To estimate the RGOW security of MQOM, we thus need to quantify the probability to get the desired rank deficiency δ:

$$p_\delta := \Pr[\text{rank}(\Phi_{\mathbb{K}\to\mathbb{F}}(\Gamma_k \cdot M + M')) = \lambda' - \delta] \ ,$$

over the randomness of $\{\hat{A}_j, \hat{b}_j\}$, ω and $P(\omega)$.

To get an upper bound for p_δ, let us analyze the probability distribution of $\Gamma_k \cdot M + M'$. The vector $P(\omega)$ is uniformly random from $\mathbb{K}^n$, therefore the probability that it has at most 2 non-zero coordinates is negligible (and much smaller than $2^{-\lambda}$ for the different parameters of MQOM). Assuming that $P(\omega)$ has at least 3 non-zero coordinates, $P_x(\omega)^T \cdot (\hat{A}_j^T + \hat{A}_j)$ is a vector uniformly random in $\mathbb{K}^n$, using the fact the $\hat{A}_j$ is a random triangular matrix. As consequence, M follows a uniform distribution in $\mathbb{F}^{\hat{m}\times n}$, using the fact that $\hat{A}_j$'s randomness are independent. Moreover, Γ_k is composed of some rows of Γ that are uniformly sampled, so Γ_k is uniformly random from $\mathbb{K}^{\hat{\lambda}'\times\hat{m}}$. In what follows, we assume that $P(\omega)$ has at least 3 non-zero coordinates and so both M and Γ_k are uniformly random.

Lemma 1. *The rank-deficiency probability p_δ is given by*

$$p_\delta = \sum_{\hat{\theta}=0}^{\hat{\lambda}'} p^{(\text{rank})}_{|\mathbb{F}|,\lambda'-\mu\cdot\hat{\theta},n-\mu\cdot\hat{\theta},(\lambda'-\delta)-\mu\cdot\hat{\theta}} \cdot p^{(\text{rank})}_{|\mathbb{K}|,\hat{\lambda}',\hat{m},\hat{\lambda}'-\hat{\theta}}$$

where $p^{(\text{rank})}_{q,a,b,c}$ is the probability that a uniformly-random matrix of $\mathbb{F}_q^{a\times b}$ is of rank exactly c, which is given by

$$p^{(\text{rank})}_{q,a,b,c} = \frac{S_{q,a,b,c}}{q^{a\cdot b}} \quad \textit{with} \quad S_{q,a,b,c} = \prod_{i=0}^{c-1} \frac{(q^a - q^i)(q^b - q^i)}{q^c - q^i}$$

where $S_{q,a,b,c}$ is the number of matrices of $\mathbb{F}_q^{a\times b}$ of weight exactly r.

Proof. Let us assume that $\mathrm{rank}_{\mathbb{K}}(\Gamma_k) = \hat{\lambda}' - \hat{\theta}$ for some $\hat{\theta}$. There exists $P \in \mathbb{K}^{\hat{\lambda}' \times \hat{\lambda}'}$ and $Q \in \mathbb{K}^{\hat{m} \times \hat{m}}$ invertible such that

$$\Gamma_k = P \begin{pmatrix} I_{\hat{\lambda}' - \hat{\theta}} & 0 \\ 0 & 0 \end{pmatrix} Q$$

Since Q is invertible, the distribution of QM is uniformly random over $\mathbb{K}^{\hat{m} \times n}$, as M is uniformly random. We denote $X \in \mathbb{K}^{(\hat{\lambda}' - \hat{\theta}) \times n}$ the first $\hat{\lambda}' - \hat{\theta}$ rows of QM. Denoting $P^{-1} \cdot M' := \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix}$ with $Y_1 \in \mathbb{K}^{(\hat{\lambda}' - \hat{\theta}) \times n}$ and $Y_2 \in \mathbb{K}^{\hat{\theta} \times n}$, we have

$$\Gamma_k \cdot M + M' = P \begin{pmatrix} X + Y_1 \\ Y_2 \end{pmatrix}.$$

Let $\mathrm{Im}_{\mathbb{F}^n}(C)$ be the set $\{C \cdot x \mid x \in \mathbb{F}^n\}$ for a matrix C with coefficients in $\mathbb{K}$ (or in $\mathbb{F}$). We have the following relation:

$$\begin{aligned}
\mathrm{rank}(\Phi_{\mathbb{K}\to\mathbb{F}}(\Gamma_k \cdot M + M')) &= \log |\operatorname{Im}_{\mathbb{F}^n}(\Phi_{\mathbb{K}\to\mathbb{F}}(\Gamma_k \cdot M + M'))| \,/\, \log|\mathbb{F}| \\
&= \log |\operatorname{Im}_{\mathbb{F}^n}(\Gamma_k \cdot M + M')| \,/\, \log|\mathbb{F}| \\
&= \log |\operatorname{Im}_{\mathbb{F}^n}\begin{pmatrix} X + Y_1 \\ Y_2 \end{pmatrix}| \,/\, \log|\mathbb{F}| \qquad (12) \\
&= \log |\operatorname{Im}_{\mathbb{F}^n}(\Phi_{\mathbb{K}\to\mathbb{F}}\begin{pmatrix} X + Y_1 \\ Y_2 \end{pmatrix})| \,/\, \log|\mathbb{F}| \\
&= \mathrm{rank}\begin{pmatrix} \Phi_{\mathbb{K}\to\mathbb{F}}(X) + \Phi_{\mathbb{K}\to\mathbb{F}}(Y_1) \\ \Phi_{\mathbb{K}\to\mathbb{F}}(Y_2) \end{pmatrix}
\end{aligned}$$

where (12) holds because P is invertible. The matrix $\Phi_{\mathbb{K}\to\mathbb{F}}(X)$ follows a uniform distribution over $\mathbb{F}^{(\lambda' - \mu\cdot\hat{\theta}) \times n}$. Since $P^{-1} \cdot M'$ is full rank (as the product of an invertible matrix and a full-rank matrix), it follows that $\Phi_{\mathbb{K}\to\mathbb{F}}(Y_2)$ is of rank $\mu \cdot \hat{\theta}$, *i.e.* it is full-rank. Given Y_1 and Y_2, the probability over the randomness of X that

$$\begin{pmatrix} \Phi_{\mathbb{K}\to\mathbb{F}}(X) + \Phi_{\mathbb{K}\to\mathbb{F}}(Y_1) \\ \Phi_{\mathbb{K}\to\mathbb{F}}(Y_2) \end{pmatrix}$$

is of rank r, for $r \geq \mu \cdot \hat{\theta}$, is the probability that a uniformly-random matrix of $\mathbb{F}^{(\lambda' - \mu\cdot\hat{\theta}) \times (n - \mu\cdot\hat{\theta})}$ has rank $r - \mu \cdot \hat{\theta}$, which is given by

$$p^{(\mathrm{rank})}_{|\mathbb{F}|, \lambda' - \mu\cdot\hat{\theta}, n - \mu\cdot\hat{\theta}, r - \mu\cdot\hat{\theta}}.$$

This follows from the fact that one can remove in $\Phi_{\mathbb{K}\to\mathbb{F}}(X) + \Phi_{\mathbb{K}\to\mathbb{F}}(Y_1)$ the columns corresponding to the full-rank part of $\Phi_{\mathbb{K}\to\mathbb{F}}(Y_2)$, reducing the problem to evaluating the probability that the resulting matrix has rank $r - \mu \cdot \hat{\theta}$. We thus deduce that

$$\begin{aligned}
p_\delta &:= \Pr[\mathrm{rank}(\Phi_{\mathbb{K}\to\mathbb{F}}(\Gamma_k \cdot M + M')) = \lambda' - \delta] \\
&= \sum_{\hat{\theta}=0}^{\hat{\lambda}'} \Pr[\mathrm{rank}(\Phi_{\mathbb{K}\to\mathbb{F}}(\Gamma_k \cdot M + M')) = \lambda' - \delta \mid \mathrm{rank}_{\mathbb{K}}(\Gamma_k) = \hat{\lambda}' - \hat{\theta}] \cdot \Pr[\mathrm{rank}_{\mathbb{K}}(\Gamma_k) = \hat{\lambda}' - \hat{\theta}] \\
&= \sum_{\hat{\theta}=0}^{\hat{\lambda}'} p^{(\mathrm{rank})}_{|\mathbb{F}|, \lambda' - \mu\cdot\hat{\theta}, n - \mu\cdot\hat{\theta}, (\lambda' - \delta) - \mu\cdot\hat{\theta}} \cdot p^{(\mathrm{rank})}_{|\mathbb{K}|, \hat{\lambda}', \hat{m}, \hat{\lambda}' - \hat{\theta}}.
\end{aligned}$$

□

We give in Table 1 the cost of the RGOW attack against the current version of MQOM. We observe that rank deficiency occurs with very small probability, but it is not totally negligible for small fields. For the first security level, the RGOW security are few bits below the target security of 128(+15) bits, namely 3 bits smaller when working on $\mathbb{F}_2$ and 4 bits smaller when working on $\mathbb{F}_{16}$. While it does not impact significantly the security, using the enhanced construction of Section 4.6 would formally avoid this small security degradation.

Table 1: Cost of the RGOW attack against the MQOM scheme.

Field	Security Level	$n = m$	λ'	Target security (log)	δ	p_δ	Attack cost (log)
$\mathbb{F}_2$	L1	160	128	128 (+15)	0	≈ 1	≈ 128 (+ 15)
					1	$\approx 2^{-32}$	≈ 127 (+ 15)
					2	$\approx 2^{-67}$	≈ 126 (+ 15)
					3	$\approx 2^{-103}$	$\approx$ **125** (+ 15)
					4	$\approx 2^{-142}$	≈ 142 (+ 15)
$\mathbb{F}_{16}$	L1	56	32	128 (+15)	0	≈ 1	≈ 128 (+ 15)
					1	$\approx 2^{-100}$	$\approx$ **124** (+ 15)
					2	$\approx 2^{-208}$	≈ 208 (+ 15)
$\mathbb{F}_{256}$	L1	48	16	128 (+15)	0	≈ 1	$\approx$ **128** (+ 15)
					1	$\approx 2^{-264}$	≈ 264 (+ 15)

Conclusion on MQOM Security. In summary, our analysis shows that the EUF-CMA security of MQOM v2 can be reduced to the hardness of the underlying MQ problem, with a small security degradation due to the PGOW & RGOW attacks. The PGOW security requires the hardness of retrieving the λ first bits of the MQ solution. This problem reduces to the original MQ problem with some tightness gap but its practical hardness is essentially the same as that of the original MQ problem. On the other hand, the RGOW security of MQOM is slightly below the target security (by ≈ 4 bits for NIST security level 1), due to the small probability of rank deficiency in the signature-dependent functions g_k. Tweaking MQOM with the enhanced Davies-Meyer instantiation of Section 4.6 would formally avoid this small security degradation at a moderate cost increase, which seems a valuable option for a future version of MQOM.

5.4 Application to Other Schemes

In the previous section, we show how to apply our results on MQOM with the MQ problem. We could apply them to all the existing MPCitH schemes (should they rely on the correlated tree technique). One can distinguish three categories of schemes:

- Schemes involving a sparse witness (i.e., low entropy in the first bits), which lead to insecure constructions in this setting. This is the case, for instance, of SDitH [ABB+24b] and PERK [ABB+24a].
- Schemes for which guessing the first λ bits of the witness is as hard as recovering the secret key itself. This is the case of FAEST [BBB+24], where these bits directly correspond to the secret.
- Schemes where guessing the first λ bits leads to an easier problem, which can then be solved to recover the full secret. In this case, one obtains a reduction to the underlying hard problem, potentially with some loss (i.e., a non-tight reduction). This applies to MQOM [BBFR24], Mirath [AAB+24], and RYDE [ABB+24c].

We emphasize that, among the cited schemes, only MQOM currently relies on correlated trees. The above classification should therefore be understood as analyzing how these schemes would behave if this optimization was applied to them.

6 Extension to Alternative Constructions

So far, we have focused on the setting as introduced in [HJ24] and used in MQOM v2, which mainly aims at reducing the signature size, and which can be considered as the results of two optimizations:

- **Half-tree optimization**: in a node derivation, the right child is defined as the XOR of the left child and the parent node.

– **Correlated seed expansion**: the λ first bits of the expanded tapes are set as the corresponding leaf seed.

While on their own, each of these optimizations saves one λ-bit block encryption per leaf (overall resulting in computational saving), combining them allows to also reduce the signature size. Alternative constructions, combining different subsets of optimizations, are possible. In this section, we overview some of them.

Another optimization that could be considered is the one-tree technique [BBM+24], which consists of interleaving τ trees of N leaves into a single one with $\tau \cdot N$ leaves. The latter saves signature size as the opened paths of the trees are merged. However, this saving is not compatible with the previous one. Indeed, setting the tree root as a long-term secret does not help to make some specific subsets of leaves summing to this secret value. While the half-tree optimization and the correlated seed expansion do not reduce further the communication of the one-tree technique, they might still be useful to reduce the computational cost. Therefore, one can wonder how secure is each combination of those three optimizations (one-tree, half-tree, and correlated expansion).

We provide a complete overview in Table 2. Some combinations have been already studied in the literature, and we analyzed the others hereafter. We show in Section 6.1 that the usage of the correlated seed expansion *without* the half-tree technique makes the construction vulnerable to the attacks described in Section 3, as so it can be proved secure only under the PGOW (and RGOW) assumptions. This observation can be extended when one also relies on the one-tree technique (still without the half-tree optimization). Finally, we show in Section 6.2 that combining the three optimizations lead to an insecure construction.

One-Tree	Half-tree	Corr. Seed Expansion	Analyzed in	Secure	Comments
✗	✗	✗	-	Yes	Baseline MPCitH, with no optimization
✓	✗	✗	[BBM+24]	Yes	-
✗	✓	✗	[BCD25]	Yes	-
✗	✗	✓	This work	PGOW (RGOW)	see Section 6.1
✓	✓	✗	[WKK+25]	Yes	-
✗	✓	✓	This work	PGOW (RGOW)	Main results of this work
✓	✗	✓	This work	PGOW (RGOW)	see Section 6.1
✓	✓	✓	This work	**No**	see Section 6.2

Table 2: Overview of the security of each possible combination of the optimizations.

6.1 No Correlation inside the Tree

Let us assume that we do not use the half-tree technique but still use the correlated seed expansion. Although this variant does not reduce the communication cost, it remains vulnerable to the λ-bit attack. Indeed, the following relation still holds in that setting:

$$\mathsf{First}_\lambda(\Delta_x) = \mathsf{First}_\lambda(x) - \sum_{i=0}^{N-1} \texttt{lseed}[i] \ .$$

The only difference is that this value is no longer zero. As the auxiliary value Δ_x is public and all the leaf seeds are public except one, guessing $\mathsf{First}_\lambda(x)$ yields the hidden leaf seed $\texttt{lseed}[i^*]$. From this seed, the

adversary can compute the corresponding share $[\![x]\!]_{i^*} = \texttt{lseed}[i^*] \parallel \mathrm{PRG}(\texttt{lseed}[i^*])$ and subsequently recover the full witness as $x = \sum_{i=0}^{N-1} [\![x]\!]_i + \Delta_x$.

This setting can be formally proven secure (under the PGOW and RGOW assumptions) using a simplified version of the proofs of the previous sections. Indeed, these proofs establish the security of a more structured construction, which is thus more prone to attacks. Hence, simplifying the argument about the opened tree path yields a proof for the setting without correlation inside the tree. Concretely, since the tree derivation does not depend on a long-term secret and the node derivation is performed using a double-length pseudorandom generator (PRG), one can use a game hop to replace the revealed nodes by fresh random nodes relying solely on the PRG security, making the rest of the proof independent of how the revealed sibling paths have been generated. After this game hop, the sibling path $(r_1, \ldots, r_\ell)$ does not help to distinguish between a call to the correlated tree oracle[5] and a call to the simulator, as those quantities are sampled uniformly at random in both cases.

This approach also extends the analysis to the one-tree technique. Indeed, as argued above, a PRG security argument allows us to replace the revealed sibling paths with random nodes in a game hop, and this remains true with the one-tree technique. This makes the security arguments after this game hop independent of how the revealed leaves have been generated, and one can thus rely on similar proofs to establish the security of a construction with the one-tree optimization and the correlated seed expansion, but without the half-tree optimization. Using the one-tree technique just requires proving the indistinguishability between the correlated tree oracle and its simulator for the τ parallel repetitions simultaneously.

6.2 Insecurity of the Correlated One-Tree Technique

Let us now assume that we use both the half-tree optimization and the correlated seed expansion in the context of the one-tree technique [BBM+24]. We show that the secret witness can then be recovered from a single signature.

For the eth repetition, we denote $\texttt{lseed}[e][i]$ the ith leaf seed and $\Delta_x^{(e)}$ the auxiliary value. When relying on the correlated seed expansion, as explained in Section 2, we have

$$\mathsf{First}_\lambda(\Delta_x^{(e)}) = \mathsf{First}_\lambda(x) - \sum_{i=0}^{N-1} \texttt{lseed}[e][i] \ .$$

The one-tree technique uses a GGM tree whose $\tau \cdot N$ leaves are $\{\texttt{lseed}[e][i]\}_{e,i}$. For each repetition, one leaf remains hidden, implying that τ leaves remain hidden in total.

Let us assume that there exists a subtree T in which the number of hidden leaves is *even*, but not zero. We denote E the set of repetitions whose hidden leaf lies in this subtree. By assumption, $|E|$ is even. For the eth repetition, we further denote S_e the set of indices of its leaves that belong to the subtree:

$$S_e = \{i \in [0, N-1] : \texttt{lseed}[e][i] \text{ is a leaf of } T\} \ .$$

Denoting δ_T the root of the subtree T, we have

$$\delta_T = \sum_{e=1}^{\tau} \sum_{i \in S_e} \texttt{lseed}[e][i]$$

[5] We still call the oracle producing tree openings the *correlated tree oracle*, even though there is technically no correlation inside the tree itself in this case, but only in the seed expansion.

since the half-tree technique ensures that the nodes of each layer of a subtree sum to its root. Using the above relation on $\mathsf{First}_\lambda(\Delta_x^{(e)})$ coming from the correlated seed expansion, we thus obtain

$$\begin{aligned}
\delta_T &= \sum_{e=1}^{\tau} \sum_{i \in S_e} \texttt{lseed}[e][i] \\
&= \left(\sum_{e \in E} \sum_{i \in S_e} \texttt{lseed}[e][i] \right) + \left(\sum_{e \notin E} \sum_{i \in S_e} \texttt{lseed}[e][i] \right) \\
&= \sum_{e \in E} \left(\mathsf{First}_\lambda(x) - \mathsf{First}_\lambda(\Delta_x^{(e)}) - \sum_{i \notin S_e} \texttt{lseed}[e][i] \right) + \left(\sum_{e \notin E} \sum_{i \in S_e} \texttt{lseed}[e][i] \right) \\
&= \underbrace{\left(\sum_{e \in E} \mathsf{First}_\lambda(x) \right)}_{=0 \text{ since } |E| \text{ is even}} + \underbrace{\left(\sum_{e \notin E} \sum_{i \in S_e} \texttt{lseed}[e][i] - \sum_{e \in E} \sum_{i \notin S_e} \texttt{lseed}[e][i] - \sum_{e \in E} \mathsf{First}_\lambda(\Delta_x^{(e)}) \right)}_{\text{public value}} .
\end{aligned}$$

Therefore, the root δ_T of this subtree can be publicly recovered. An attacker can thus expand this subtree, as all its nodes can be deterministically computed from its root, and thereby recover the hidden leaves it contains, which leak the full witness.

To conclude, when using the one-tree technique combined with the half-tree technique and the correlated seed expansion, any subtree with an even non-zero number of hidden leaves leaks the secret, and such a subtree always exists when $\tau \geq 2$.

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`.

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`.

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`.

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`.

ABC+24. Gorjan Alagic, Maxime Bros, Pierre Ciadoux, David Cooper, 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 first round of the additional digital signature schemes for the nist post-quantum cryptography standardization process. NIST Internal Report, NIST IR 8528, 2024.

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`.

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.

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`.

BBM+24. Carsten Baum, Ward Beullens, Shibam Mukherjee, Emmanuela Orsini, Sebastian Ramacher, Christian Rechberger, Lawrence Roy, and Peter Scholl. One tree to rule them all: Optimizing GGM trees and OWFs for post-quantum signatures. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part I*, volume 15484 of *LNCS*, pages 463–493. Springer, Singapore, December 2024.

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.

BF26. Ryad Benadjila and Thibauld Feneuil. Breaking the myth of MPCitH inefficiency: Optimizing MQOM for embedded platforms. Cryptology ePrint Archive, Paper 2026/078, 2026.

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.

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.

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

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.

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.

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.

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.

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

KLS25. 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.

KX25. Haruhisa Kosuge and Keita Xagawa. New security proofs of MPC-in-the-head signatures in the quantum random oracle model. Cryptology ePrint Archive, Report 2025/1999, 2025.

KX26. Haruhisa Kosuge and Keita Xagawa. Towards formal security proofs of MQOM. Cryptology ePrint Archive, Paper 2026/629, 2026.

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.

WKK+25. Yalan Wang, Bryan Kumara, Harsh Kasyap, Liqun Chen, Sumanta Sarkar, Christopher J.P. Newton, Carsten Maple, and Ugur Ilker Atmaca. BACON: An improved vector commitment construction with applications to signatures. Cryptology ePrint Archive, Report 2025/1411, 2025.

YSWW21. Kang Yang, Pratik Sarkar, Chenkai Weng, and Xiao Wang. QuickSilver: Efficient and affordable zero-knowledge proofs for circuits and polynomials over any field. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 2986–3001. ACM Press, November 2021.

A Comparison with [KX26]

Concurrently to our work, [KX26] proposes a security proof for the usage of correlated trees in MPCitH schemes. [KX26] focuses on proving the security of MQOM v2, while our work studies the security of correlated trees in a more general setting. As a consequence, some terms appear only in [KX26], as it is closer to the concrete design of MQOM, while they are absent in our work since they are orthogonal to the analysis of correlated trees. In particular, in MQOM v2:

- the root seeds for the tau executions of BLC/AVC are derived from a master seed, which introduces a PRG distinguishing advantage in [KX26], whereas we assume these seeds to be uniformly random in our proof;
- the message is pre-hashed before being included in the Fiat–Shamir transformation, which introduces a collision probability on the message digest in [KX26], while we do not consider such a pre-hash step.

These differences are independent of the security analysis of correlated trees.

Excluding the QROM results of [KX26], our analysis covers a broader range of settings than [KX26], without relying on heavy design tweaks (such as matching the seed size to the witness size), while also achieving tighter security guarantees. We describe below the differences more in details.

Salt Handling and Modeling Choices. Both [KX26] and our work assume that salts are tweaked via concatenation with domain separation. While this is explicit in [KX26], it is only implicitly stated in our article. This choice significantly simplifies the analysis. Additionally, [KX26] assumes that the salt is included in the Fiat–Shamir hash from which the opening challenge is derived. It also proposes a variant where the seed size matches the witness size.

Proof Techniques. The techniques developed in the second proof of [KX26] are essentially the same as those used in our work. Inspired by existing TCCR security proofs, both works reduce TCCR security to a computational assumption through an extraction argument based on the H-technique. The first proof of [KX26] differs from this approach, as it relies solely on game hops.

Practical Relevance of the Seed-Size Tweak. [KX26] proposes a variant where the seed size matches the witness size. While this leads to a provable security result (up to a conjecture), it significantly degrades performance (including signature size), to the point that it becomes less efficient than MQOM variants that do not use correlated trees. Thus, this approach is mainly of theoretical interest and does not lead to a practical design.

Tightness of the Security Proofs. The main difference between the two works lies in the tightness of the reductions.

- we target tight reductions.
- [KX26] introduces several non-tight terms (e.g., $Q^2/2^\lambda$).

Three main sources of non-tightness can be identified in [KX26]:

1. Salt collision handling. [KX26] assumes that no salt collision occurs, leading to a term $Q_{\text{sign}}^2/2^{\lambda-12}$. In contrast, we allow multi-collisions (up to multiplicity μ), which requires explicitly handling collision events (InputCollision, KeyInputCollision, KeyOutputCollision), but avoids this loss.
2. Challenge reprogramming. [KX26] analyzes reprogramming using entropy arguments on the λ-bit salt, leading to a loose bound. We provide a tighter analysis (see Game 1).
3. Reduction to computational assumptions. [KX26] reduces to PDOW, while we introduce the PGOW assumption, which is strictly stronger: any adversary against PDOW immediately yields an adversary against PGOW with the same performance. However, the converse reduction is not tight. The underlying reason is that, in the PDOW-based reduction, the adversary must guess among all queries to the TCCR

hash function, introducing a loss proportional to the number of such queries. By contrast, when working under the PGOW assumption, this guessing step is avoided, as the reduction can exhaustively apply the guessing oracle to all candidates. For this reason, PGOW appears to be the most appropriate notion to capture the security of the scheme, even though a (non-tight) reduction to PDOW remains possible.

Block Cipher Instantiations While the first proof of [KX26] considers only the random oracle model, the second proof aims to establish security when instantiating the construction with a block cipher in ICM mode, under the additional tweak that the seed size matches the witness size. The resulting proof appears consistent with our Davies–Meyer-based analysis in Section 4.4.

In this setting, where the seed size matches the witness size, PGOW security can be reduced to standard one-way security. The RGOW assumption introduced in our work does not explicitly appear in the theorem of [KX26]; however, it corresponds to Conjecture 6.3 in [KX26], which is not further elaborated. While somewhat ad hoc, the RGOW assumption allows one to formally capture the argument underlying this conjecture. Moreover, we provide a concrete analysis of RGOW in the context of MQOM, showing that it induces only a small security loss (on the order of a few bits).

Up to this point -and aside from the fact that [KX26] operates in a no-collision setting- the analysis of the second proof in [KX26] is essentially identical to our Davies–Meyer proof, under the seed-size matching tweak.

Minor Technical Difference. One (minor) technical difference between the proofs of [KX26] and our work lies in the choice of the value output by the security games: [KX26] outputs α_0 (sampled at random), while we output α_1 (*c.f.* Game 2.1 in [KX26]). Although these two choices are generally equivalent -since α_0 and α_1 are related- this distinction leads to a subtle issue in [KX26] when the evaluation point is zero. In that case, α_1 cannot be derived from α_0, as this would require computing the inverse of zero.

Quantum security proof. The final key difference between our work and [KX26] is that [KX26] provides security arguments in the quantum setting, whereas our work specifically focuses on the classical (*i.e.* non-quantum) setting. Moreover, the quantum security arguments of [KX26] only apply to their first proof strategy based on game hopping (and not to the H-coefficient technique), and only in the case where the primitives are instantiated with hash functions rather than block ciphers. Consequently, obtaining a quantum security proof in the ideal-cipher model remains an open problem.