Compact, Efficient and Non-Separable Hybrid Signatures

Julien Devevey[1], Morgane Guerreau[2], and Maxime Roméas[1]

[1] ANSSI, Paris, France
julien.devevey@ssi.gouv.fr
maxime.romeas@ssi.gouv.fr
[2] PQShield
morgane.guerreau@pqshield.com

Abstract. The transition to post-quantum cryptography involves balancing the long-term threat of quantum adversaries with the need for post-quantum algorithms and their implementations to gain maturity safely. Hybridization, i.e. combining classical and post-quantum schemes, offers a practical and safe solution.
We introduce a new security notion for hybrid signatures, Hybrid EU-CMA, which captures cross-protocol, separability, and recombination attacks that may occur during the post-quantum transition, while encompassing standard unforgeability guarantees. Using this framework, we adapt the Fiat-Shamir (with or without aborts) transform to build hybrid signature schemes that satisfy our notion from two identification schemes. Compared to simple concatenation of signatures, our construction (i) has no separability issues, (ii) reduces signature size, (iii) runs faster, and (iv) remains easily implementable.
As a concrete application, we propose Silithium, a hybrid signature combining the identification schemes underlying EC-Schnorr and ML-DSA. Implementing Silithium requires only an ML-DSA implementation supporting the "external μ" option during verification and an elliptic curve library. In the security analysis, we show that our scheme can be safely used along with ML-DSA and either EC-Schnorr or ECDSA. A proof-of-concept OpenSSL implementation demonstrates its practicality, simplicity, and performance.

Keywords: Post-Quantum Transition, Hybrid Signatures, Fiat-Shamir

1 Introduction

The ongoing transition to post-quantum cryptography presents one of the most delicate challenges in modern security engineering. While post-quantum algorithms are progressively being standardized, their deployment in real-world infrastructures requires careful planning to mitigate security risks. Although these newly standardized algorithms have undergone at least a decade of analysis, the confidence and maturity gained through cryptanalysis, implementation, and deployment cannot yet compare to that accumulated for classical factorization-

and discrete logarithm-based algorithms. This creates a tension between the long-term threat posed by cryptographically relevant quantum computers and the need for post-quantum algorithms to gain maturity through large-scale deployment, all while maintaining the robust security assurances of established classical schemes. For these reasons, several European security agencies—such as ANSSI [1], BSI [9], and NLNCSA [26]—strongly recommend an intermediate state known as *hybridization*: combining classical and post-quantum algorithms in such a way that overall security is preserved as long as at least one algorithm remains secure.

In the case of digital signatures, the most straightforward approach to hybridization is *concatenation*—that is, forming a hybrid verification key, signing key, and signature by concatenating the respective values of at least two schemes, with verification checking both signatures independently and accepting only if both succeed. This method guarantees unforgeability ($\mathsf{EU\text{-}CMA}$ security) of the hybrid as long as one component remains unforgeable. However, it fails to ensure strong unforgeability ($\mathsf{sEU\text{-}CMA}$) whenever one component is probabilistic, due to attacks that exploit the independence of the component signatures. This distinction is critical in practice, as $\mathsf{sEU\text{-}CMA}$ security is required in applications such as authenticated key exchange [2,18], SSH [4], and cryptocurrencies [21].

As Table 1 illustrates, different transition strategies require distinct patterns of key updates. A purely post-quantum transition involves a single key update, whereas a "hygienic" hybridization approach requires two updates to ensure strict separation between key lifecycles. In practice, however, users and systems are more likely to adopt a minimal-overhead strategy, adding a post-quantum key to an existing pre-quantum key and later phasing out the pre-quantum component. In this scenario, achieving only EU-CMA security for the hybrid signature is insufficient. Indeed, a concatenated signature can be stripped to yield a valid component signature or recombined from independent component signatures to form a valid hybrid. Such separability makes the scheme vulnerable to upgrade, downgrade, and cross-protocol attacks, undermining the security guarantees of the hybrid approach.

Strategy	Key Updates
Post-Quantum only	$\mathsf{vk}_1 \xrightarrow{\qquad\qquad} \mathsf{vk}_2$
Hygienic Hybridization	$\mathsf{vk}_1 \to (\mathsf{vk}_1', \mathsf{vk}_2') \to \mathsf{vk}_2$
Expected Hybridization	$\mathsf{vk}_1 \to (\mathsf{vk}_1, \mathsf{vk}_2) \to \mathsf{vk}_2$

Table 1: Key usage during the post-quantum transition.

Non-separability. Previous works introduced the notion of *non-separability* to capture the idea that access to a hybrid signature oracle should not facilitate forgery against any of its components [7,5]. More concretely, a non-separable hybrid scheme prevents an adversary from extracting valid component signatures—or generating them for related messages—based on hybrid signatures obtained for messages of their choice. As definitions evolved, it became clear that non-separability is not a single property but rather a spectrum: from separability, to *weak non-separability* (WNS), which ensures that producing a component signature from hybrid ones leaves recognizable artifacts, and *strong non-separability* (SNS), which unconditionally prohibits the reconstruction of any valid component signature from hybrid ones.

These refinements, along with the broader design and security goals for hybrid signatures, are discussed in two recent IETF PQC drafts [6] and [28]. The latter also introduces *simultaneous verification* (SV), an informal property which ensures that an honest but sloppy verifier cannot prematurely terminate the verification process after validating only one component. Together, SNS and SV represent the most stringent design objectives for hybrid signatures, ensuring that the hybrid behaves as a single, indivisible algorithm rather than two independent ones. As discussed above, simple concatenation of signatures does not satisfy these requirements.

Following the guidance of [6,28], an ideal hybrid signature scheme should achieve the following goals:

Hybrid security. Its EU-CMA security should be at least as hard to break as the strongest security assumption it relies on.

Transition security. SNS, SV, and resistance to cross-protocol attacks in the transition scenario of Figure 1.

Advanced security. sEU-CMA security and Beyond UnForgeability Features (BUFF) [11,13,14] to ensure usability across a wide range of real-world applications.

Efficiency. Computational and space requirements no worse than concatenation, minimizing the operational cost of hybridization.

Implementation simplicity. Reusing existing standardized implementations whenever possible to facilitate adoption and increase trust.

Generality. Instantiable with all schemes within a broad family, such as Fiat-Shamir-based signatures, supporting cryptographic agility.

Proof composability. Leveraging the existing security proofs and cryptanalysis of standardized components wherever possible.

Existing constructions. Several generic approaches to constructing hybrid signature schemes have been explored in the literature. The most straightforward one is concatenation, in which component signatures are computed independently and then concatenated. While this approach ensures unforgeability of the hybrid as long as at least one component remains secure, it fails to preserve strong unforgeability whenever any of the component is probabilistic. Moreover, it does not satisfy SNS or SV. A concrete example is the Composite ML-DSA construction defined in the IETF draft [28] which specifies concatenation between ML-DSA and pre-quantum signature algorithms. In this design, a fixed prefix is prepended to each message before both components sign it, a modification that allows the scheme to achieve WNS, since the prefix serves as a recognizable artifact of hybridization. However, SNS remains out of reach for this design.

To overcome these limitations, several works have proposed more sophisticated hybrid constructions. A first line of research [7,16] investigates *nested* hybrids, where a message is first signed by one component, and the resulting pair consisting of the message and its signature is then signed by the second component. This design preserves the strong unforgeability of the outer (second) component. However, if the outer scheme is broken, the security of the construction collapses to that of simple concatenation. Furthermore, nested hybrids do not achieve SNS—a limitation that is typical for hybrid schemes making black-box use of one of their components. Indeed, the hybrid verification algorithm can be executed by anyone and it needs to verify the signature of the black-box component at some point.

Bindel and Hale [5] propose a family of generic and semi-generic hybrid constructions based on the Fiat-Shamir (FS) paradigm. Their FS-FS hybrid combines two FS signatures using a single hash computation, thereby improving efficiency compared to concatenation while maintaining comparable signature sizes. They also present semi-generic hybrids involving RSA [29], DSA, or Falcon [15]. All of their constructions are claimed EU-CMA secure in the Random Oracle Model (ROM), but proofs are not available. Their FS-FS approach cannot be instantiated with ML-DSA as it uses the Fiat-Shamir *with aborts* (FSwA) framework, and it is unclear how to make sure that the two identification schemes have the same challenge space, required by the construction. Moreover, they observe that hybridization of signatures whose verification ends with a digest comparison allows for means of mixing the two components to ensure SV.

A recent eprint by Janneck [19] introduces three new hybrids that use their post-quantum component in black-box—aiming for FIPS compliance—and preserve strong unforgeability if it holds for at least one component. However, as discussed previously, this black-box constraint makes their hybrids separable with respect to the post-quantum component, thus preventing them from achieving SNS. In addition, [19] formalizes a new security property called Random-Message Validity, which serves to analyze the BUFF security properties of their schemes.

Despite these advances, the literature still exhibits gaps. Most hybrid signature constructions either fail to ensure strong non-separability for all components, lack simultaneous verification, or do not provide proof composability and generality across different signature families. Furthermore, existing works do not consider adversaries in the Quantum Random Oracle Model (QROM), limiting the relevance of their security guarantees in post-quantum settings. To date, no existing construction simultaneously achieves strong non-separability for both components, simultaneous verification, proof composability, and generality across signature families—while maintaining efficiency and ease of implementation comparable to concatenation. In addition, the ongoing evolution of formal definitions for non-separability, the lack of formal proofs for schemes targeting these notions, and the insufficient consideration of cross-protocol attacks highlight the need for a more rigorous and unified framework for hybrid signature security. Such a framework should accurately capture the operational reality of hybrid deployments during the post-quantum transition and facilitate the writing of rigorous security proofs.

1.1 Contributions and Technical Overview

We take the following steps to address the challenges identified in the literature.

Formalization. To accurately capture cross-protocol, separability, and recombination attacks that may arise during the post-quantum transition, we introduce in Section 3 a new hybrid unforgeability notion, denoted $\mathsf{H\text{-}EU\text{-}CMA}^{\mathsf{X}}$, together with its *strong* variant. Given signature oracles for a hybrid signature scheme Σ_{H} as well as for its two component schemes Σ_1 and Σ_2, the adversary must forge a signature for Σ_{X}. We illustrate this new notion in Figure 1.

If the target scheme is one of the two components, security under this notion ensures that no adversary can separate hybrid signatures to create a forgery against that component. When the target scheme is the hybrid one, the notion guarantees that no adversary can recombine component signatures into a valid hybrid signature. Our notion goes even further, allowing the adversary to mix and match multiple signatures across all oracles.

Finally, we observe that our notion subsumes the standard $\mathsf{(s)EU\text{-}CMA}$ notions: disabling the components oracles yields the traditional unforgeability game for the hybrid scheme, while disabling the hybrid oracle and one component oracle recovers the standard unforgeability notions for the remaining component. As such, the $\mathsf{H\text{-}EU\text{-}CMA}$ security notion appears as the *de facto* definition for hybrid signature schemes that simultaneously achieve goals Hybrid security and Transition security, except for SV. This unified definition provides the foundation for the formal security proofs of our hybrid construction presented in Section 4.

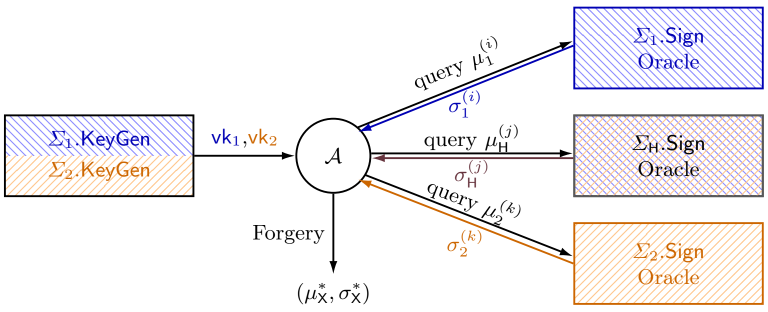

Fig. 1: $\mathsf{H\text{-}EU\text{-}CMA}^{\mathsf{X}}$ security game. Adversary $\mathcal{A}$ wins if Σ_{X} verifies σ^* correctly for message μ^*, and μ^* was never queried to $\Sigma_{\mathsf{X}}.\mathsf{Sign}$.

Construction. Throughout this section, the reader may keep in mind the examples of the $\mathsf{Schnorr}$ [31] identification scheme[3] and that underlying $\mathsf{ML\text{-}DSA}$. Our

[3] An identification scheme is a three-message interactive proof, where a prover first *commits* to a value, then *responds* to a *challenge* issued by the verifier, with the goal of convincing the verifier of knowledge of a secret.

construction, presented in Section 4, begins with two identification schemes that share the same challenge space, where the second one may involve aborts. We start with the following idea: concatenate the identification schemes, by merging all prover messages while using a single challenge from the verifier, and then convert the resulting protocol into a signature scheme using the FSwA framework—an approach reminiscent of the hybrid construction of [5]. However, we account for potential aborts and adopt the BUFF transform, in which the message is first hashed together with a hash of the verification key to achieve the related security notions.

Since only one protocol may abort, modifying the commitment of the other one seems unnecessary. Therefore, we take this commitment out of the while loop and generate it only once. We put it as a prefix to the message before the BUFF hashing step, thereby minimizing the size of the hash input inside the loop and improving efficiency. The resulting scheme is formally described in Figure 5 of Section 4 and satisfies the following properties.

Hybrid security and Transition security. We study the security of our hybrid scheme with respect to the BUFF variant of the component signature schemes, as these are the versions typically deployed in practice, particularly for post-quantum schemes. We prove that our hybrid scheme is $\mathsf{H\text{-}EU\text{-}CMA}^{\mathsf{H}}$-secure in the ROM as long as one of the two underlying assumptions remains hard. Similarly, it is $\mathsf{H\text{-}EU\text{-}CMA}^{1}$- and $\mathsf{H\text{-}EU\text{-}CMA}^{2}$-secure in the ROM as long as the corresponding underlying assumption holds. Simultaneous verification is achieved because the verification of the second component inherently requires recovering the commitment of the first scheme. Since the second scheme recomputes the challenge, part of the verification for the first scheme is also performed in this step. A lazy verifier may, at most, omit the final portion of the first scheme's verification if one exists, but simultaneous verification is otherwise ensured.

Advanced security: sEU-CMA*.* The aforementioned proofs can be extended to the sH-EU-CMA setting. Moreover, since we can extract valid signatures for the non-BUFF version of the second component, our scheme is sEU-CMA-secure in the QROM as long as its second component is. However, the sEU-CMA security of the scheme is as weak as the weakest out of the following three: breaking the EU-CMA security of the hybrid scheme or breaking the sEU-CMA security of any of the two components without breaking its EU-CMA security, *i.e.* being able to produce new signatures for already signed messages but not for new messages.

Advanced security: BUFF*.* We prove that our construction satisfies the three usual BUFF notions: exclusive ownership and message-bound signature result from the collision resistance of the hash function used. Non-resignability is trickier to prove, but the proof follows from standard arguments [13].

Efficiency. Our scheme is efficient: compared to the concatenation of the two BUFF FS(wA) signature schemes, it saves one hash computation without increasing the input sizes for the remaining hash computations. Other operations remain unchanged.

Implementation simplicity. As noted earlier, our scheme essentially makes calls to the non-BUFF FS(wA) component signature scheme. In practice, this allows us to use, for example, an "external μ" implementation of ML-DSA as a black-box.

Generality. Our construction can hybridize any post-quantum FS(wA) signature with EC-Schnorr. This includes the NIST standard ML-DSA [27], nine of the fourteen candidates from the NIST round 2 additional digital signature schemes standardization process [24] as well as the South Korean standard Haetae [10], demonstrating the flexibility and broad applicability of our construction. Sharing a common challenge space is the case in practice: implementations convert the output of a cryptographic hash function into the challenge with post-processing (*e.g.*, SampleInBall in ML-DSA). In Appendices A and B, we demonstrate how to adapt the underlying protocols of ML-DSA and Schnorr to use the output set of the hash function as their challenge space. While this adaptation must be done for each scheme, it represents the only part that needs modification when applying our hybridization technique.

Proof composability. Our proof strategy for H-EU-CMA security closely follows existing proofs of EU-CMA security for FS(wA) signature schemes. Our QROM proof is a direct reduction from the hybrid signature to the non-BUFF component signature scheme, providing strong assurance that our hybrid scheme is at least as secure as its components.

Comparison with previous schemes. While similar to [5], our scheme incorporates the BUFF transform and accounts for potential aborts, resulting in stronger security guarantees and greater flexibility in instantiation. Compared to the BoP2 construction from [19], our approach slightly opens the black-box of the second signature component, achieving SNS for both components rather than just one. Additionally, our design computes one less hash for the second signature, which could be costly depending on the instantiation. We give a comparison between existing hybrid signatures constructions in Table 2.

Hybrid construction	EU-CMA	sEU-CMA	SNS w.r.t.	H-EU-CMA	BUFF	SV
Concatenation [28]	✓	✗	none	✗	?	✗
Nested [7,16]	✓	✓*	none	✗	✗	✗
FS-FS [5]	✓†	✓*†	all†	?	✗	✓
BoP2 [19]	✓	✓	only one	✗	✓	✓
This work	✓	✓*	all	✓	✓	✓

†: no proof given.

Table 2: Comparison of existing generic hybrid signatures constructions. For (s)EU-CMA, a mark indicates that the property holds as long as one component remains secure, while an asterisk indicates that the property holds only if the post-quantum component remains secure. Our stronger H-EU-CMA notion captures attacks linked to the transition scenarios of Fig. 1.

Instantiation. In Section 5, we instantiate our construction with EC-Schnorr and ML-DSA, yielding the Silithium signature scheme. The different challenge space issue is solved in Appendices A and B, with a negligible security loss. Our construction satisfies all security notions introduced above, though these guarantees are stated with respect to the BUFF EC-Schnorr signature scheme, which is not standard. We outline how our generic proofs can be adapted to the case where Silithium is defined as a hybrid between ML-DSA and ECDSA. Since ECDSA employs a different hash function than ML-DSA and Silithium, access to an ECDSA oracle does not interfere with the random oracle management in our proofs, yielding comparable security guarantees.

To demonstrate the simplicity of our approach, we implemented Silithium within a fork of the popular OpenSSL open-source library. This integration leverages OpenSSL's support for both elliptic curve operations and the "external μ" variant of ML-DSA. Our source code is available at `https://github.com/mguerrea/openssl-silithium`.

Finally, to circumvent the limitations of ECDSA, Appendix C presents a second instantiation hybridizing ML-DSA with EdDSA.

2 Preliminaries

Notations. Given a set X, we denote by $U(\mathsf{X})$ the uniform distribution over X. For any integer η, we let S_η denote the set $\{-\eta, \ldots, 0, \ldots \eta\}$. The statistical distance between two distributions A and B is denoted by $\Delta(A, B)$. For two random variables X and Y, let $\mathsf{Supp}(X) = \{x | \Pr(X = x) > 0\}$. We define the conditional min-entropy of X given Y as $H_\infty(X|Y) = -\log(\max_{x,y} \Pr(X = x|Y = y))$.

We let $\mathcal{R}$ (resp. $\mathcal{R}_q$) denote the ring $\mathbb{Z}[x]/(x^n + 1)$ (resp. $\mathbb{Z}_q[x]/(x^n + 1)$) for n a power of two. For any integers r and α, we define $r \bmod {}^+\alpha$ (resp. $r \bmod {}^\pm\alpha$) as the unique integer r' in $[0, \alpha - 1]$ (resp. $(-\alpha/2, \alpha/2]$) such that $r = r' \bmod \alpha$.

An algorithm $\mathcal{A}$ with oracle access to $\mathcal{O}$ is denoted $\mathcal{A}^{\mathcal{O}}$. In the ROM (resp. QROM), a hash function $H : \{0,1\}^* \to \{0,1\}^n$ is modeled as a random oracle if its outputs are independently sampled from $U(\{0,1\}^n)$ and an adversary $\mathcal{A}$ can query (resp. in superposition) an oracle to have access to H. For a probabilistic algorithm f, we write $f(\cdot; x)$ to explicitly denote its random coins x.

2.1 Probabilities

Lemma 1. *Let $q > 0$ be an integer. Let $X_1, \ldots, X_q$ be* i.i.d. *discrete random variables. For any probabilistic function $f : \mathsf{Supp}(X_1)^q \to [0, q-1] \cap \mathbb{Z}$ with random coins R, let $Y = X_{f(X_1,\ldots,X_q;R)}$. It holds:*

$$H_\infty(Y|R) \geq H_\infty(X_1) - \log(q).$$

Proof. For any $k \in \mathsf{Supp}(X_1)$ and $r \in \mathsf{Supp}(R)$, we have

$$\Pr(Y = k|R = r) \leq \Pr(\exists i, X_i = k) \leq \sum_{i=1}^{q} \Pr(X_i = k) = q \Pr(X_1 = k).$$

We recall the general forking lemma from [3] in a way such that the reprogrammed value may not be chosen by the forking algorithm, which does not change the probability computations from [3, Lemma 1]. We also note that it can be applied in presence of oracles that are unrelated to the random oracle.

Lemma 2 (General Forking Lemma [3, Lemma 1]). *For any algorithm* A *making at most* q *queries to a random oracle* $H : \{0,1\}^* \to \mathcal{C}$ *and outputting a pair* (i, σ)*, where* $0 \le i \le q$*, we define the algorithm pair* $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2)$ *in Figure 2. Then, for any distribution* D*, the following holds:*

$$\Pr_{x \hookleftarrow D}(\mathsf{A}(x) \to (i,\sigma) \wedge i > 0) \le \frac{q}{|\mathcal{C}|} + \sqrt{q \cdot \Pr_{x \hookleftarrow D}(\mathsf{F}(x) \to 1)} \ .$$

If A *also needs an access to an oracle that is independent of its random oracle, the above still holds by assuming that* $\mathcal{F}$ *also has access to this oracle.*

```
Game F($x$):
  1: $st = (i, \sigma, \mathcal{S}) \leftarrow \mathcal{F}_1(x)$
  2: $(\star, c) \leftarrow \mathcal{S}[i-1]$
  3: $c' \hookleftarrow U(\mathcal{C} \setminus \{c\})$
  4: $(b, \star, \star) \leftarrow \mathcal{F}_2(x, c', st)$
  5: return $b$

$\mathcal{F}_1(x)$:
  1: Pick random coins $\rho$ for $\mathsf{A}$
  2: $\mathcal{S} \leftarrow []$
  3: $(i, \sigma) \leftarrow \mathsf{A}^H(x; \rho)$
  4: return $(i, \sigma, \mathcal{S})$

$\mathcal{F}_2(x, c', (i, \sigma, \mathcal{S}))$:
  1: if $i = 0$ then
  2:     return $(0, \varepsilon, \varepsilon)$
  3: $(y, c) \leftarrow \mathcal{S}[i-1]$
  4: $\mathcal{S}[i] \leftarrow (y, c')$
  5: $\mathcal{S} \leftarrow \mathcal{S}[: i]$
  6: $(i', \sigma') \leftarrow \mathsf{A}^H(x; \rho)$
  7: if $i = i'$ then
  8:     return $(1, \sigma, \sigma')$
  9: return $(0, \varepsilon, \varepsilon)$

$H(y)$:
  1: if $\exists k, \mathcal{S}[k] = (y, c')$ then
  2:     $c \leftarrow c'$
  3: else
  4:     $c \hookleftarrow U(\mathcal{C})$
  5: $\mathcal{S}$.append($(y, c)$)
  6: return $c$
```

Fig. 2: Forking algorithm $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2)$ and its random oracle. We use Python syntax for list manipulations.

2.2 Security Assumptions

We recall the definitions of the collision game for hash functions [30], the Discrete Logarithm problem (DL) [12], and the Module-Short Integer Solution (MSIS) and the Module-Learning with Errors (MLWE) problems [23].

Definition 1 (Collision Game). *Let* $H : \{0,1\}^* \to \{0,1\}^n$*. For any adversary* $\mathcal{A}$*, we define* $\mathsf{Adv}_H^{\mathsf{col}}(\mathcal{A}) = \Pr(\mathcal{A}() \to (x, x') | H(x) = H(x') \wedge x \ne x')$*.*

Definition 2 (DL). *Let* $\mathbb{G}$ *be a group with generator* g *and order* p*. For any adversary* $\mathcal{A}$*, its advantage in the* DL *problem is defined as:*

$$\mathsf{Adv}_{\mathbb{G},g}^{\mathsf{DL}}(\mathcal{A}) = \Pr_{k \hookleftarrow U(\mathbb{Z}_p)}(k \leftarrow \mathcal{A}(g^k)).$$

Definition 3 (MSIS). *Let q, k, ℓ, B be four integers. For any adversary $\mathcal{A}$, its advantage in the* $\mathsf{MSIS}_{n,q,k,\ell,B}$ *problem is defined as:*

$$\mathsf{Adv}^{\mathsf{MSIS}}_{n,q,k,\ell,B}(\mathcal{A}) = \Pr_{\mathbf{A} \hookleftarrow U(\mathcal{R}_q^{k\times\ell})} (\mathbf{A}\mathbf{x} = 0 \bmod q \wedge \|\mathbf{x}\|_\infty \leq B | \mathbf{x} \leftarrow \mathcal{A}(\mathbf{A})).$$

Definition 4 (MLWE). *Let q, k, ℓ be three integers and let χ be a distribution over $\mathbb{Z}$, extended over $\mathcal{R}$ where each coefficient is* i.i.d. *following χ. For any distinguisher $\mathcal{A}$, its advantage in the* $\mathsf{MLWE}_{n,q,k,\ell,\chi}$ *problem is defined as:*

$$\mathsf{Adv}^{\mathsf{MLWE}}_{n,q,k,\ell,\chi}(\mathcal{A}) = \left| \Pr_{\substack{\mathbf{A} \hookleftarrow U(\mathcal{R}_q^{k\times\ell}) \\ \mathbf{b} \hookleftarrow U(\mathcal{R}_q^{k})}} (1 \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{b})) - \Pr_{\substack{\mathbf{A} \hookleftarrow U(\mathcal{R}_q^{k\times\ell}) \\ \mathbf{s},\mathbf{e} \hookleftarrow \chi^\ell \otimes \chi^k}} (1 \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})) \right|.$$

2.3 Signature Schemes

We now briefly recall the formalism of digital signatures.

Definition 5. *A signature scheme Σ is a tuple* (Σ.KeyGen, Σ.Sign, Σ.Verify) *of probabilistic polynomial time (PPT) algorithms with the following specifications:*

- Σ.KeyGen : $1^\lambda \to$ (vk, sk) *takes as input a security parameter λ and outputs a verification key* vk *and a signing key* sk.
- Σ.Sign : (sk, μ) $\to \sigma$ *takes as inputs a signing key* sk *and a message μ and outputs a signature σ.*
- Σ.Verify : (vk, μ, σ) $\to b \in \{0, 1\}$ *takes as inputs a verification key* vk, *a message μ and a signature σ and accepts ($b = 1$) or rejects ($b = 0$).*

Σ is correct if for any μ and (vk, sk) *in the range of* Σ.KeyGen *it always holds:*

$$\Sigma.\mathsf{Verify}(\mathsf{vk}, \mu, \Sigma.\mathsf{Sign}(\mathsf{sk}, \mu)) = 1.$$

Definition 6 (EU-CMA). *Let Σ be a signature scheme. For any adversary $\mathcal{A}$ making at most Q_s queries to a* Σ.Sign(sk, ·) *oracle $\mathcal{O}$, its advantage against the* EU-CMA *game is defined as:*

$$\mathsf{Adv}^{\mathsf{EU\text{-}CMA}}(\mathcal{A}) = \Pr_{\substack{(\mathsf{vk},\mathsf{sk}) \leftarrow \Sigma.\mathsf{KeyGen}(1^\lambda) \\ (\sigma^*, \mu^*) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{vk})}} \begin{pmatrix} \Sigma.\mathsf{Verify}(\mathsf{vk}, \sigma^*, \mu^*) = 1 \wedge \\ \mu^* \text{ was not queried to } \mathcal{O} \end{pmatrix}.$$

If $Q_s = 0$, the game is called EU-NMA *instead. If the second condition is replaced with "σ^* is not a reply from $\mathcal{O}$ to a query for μ^*", we call it* sEU-CMA *instead.*

"BUFF-ing" signature schemes ensures security properties that go beyond unforgeability. We recall the Fiat-Shamir version of this transform.

Definition 7 (BUFF **transform**). *Let Σ be a signature scheme and $\mathcal{H}$ be a collision-resistant hash function. We define* BUFF(Σ) *as:*

- BUFF(Σ).KeyGen *is exactly* Σ.KeyGen *(*sk *may optionally include* $\mathcal{H}$(vk)*).*

- $\mathsf{BUFF}(\Sigma).\mathsf{Sign}$, *on input a signing key* sk *and a message* M, *first computes* $\mu = \mathcal{H}(\mathcal{H}(\mathsf{vk}), M)$ *and returns* $\Sigma.\mathsf{Sign}(\mathsf{sk}, \mu)$.
- $\mathsf{BUFF}(\Sigma).\mathsf{KeyGen}$, *on input a verification key* vk, *a signature* σ *and a message* M, *first computes* $\mu = \mathcal{H}(\mathcal{H}(\mathsf{vk}), M)$ *and returns* $\Sigma.\mathsf{Verify}(\mathsf{vk}, \sigma, \mu)$.

We recall the properties targeted by this transform: *exclusive ownership* prevents reusing a signature under a different key; *message binding* ensures a signature cannot be valid for two messages; and *non-resignability* prevents producing a valid key and signature pair for an unknown signed message.

Definition 8 (Exclusive Ownership [8]). *The advantage of an adversary* $\mathcal{A}$ *against the* malicious strong universal exclusive ownership *property of a signature scheme* Σ *is defined as:*

$$\mathsf{Adv}^{\mathsf{EO}}_{\Sigma}(\mathcal{A}) = \Pr_{(\mathsf{vk},\mathsf{vk}',m,m',\sigma)\leftarrow\mathcal{A}}(\Sigma.\mathsf{Verify}(\mathsf{vk}, m, \sigma) = \Sigma.\mathsf{Verify}(\mathsf{vk}', m', \sigma) = 1 \wedge \mathsf{vk} \neq \mathsf{vk}').$$

Definition 9 (Message Bound Signatures [8]). *The advantage of an adversary* $\mathcal{A}$ *against the* message bound signatures *property of a signature scheme* Σ *is defined as:*

$$\mathsf{Adv}^{\mathsf{MBS}}_{\Sigma}(\mathcal{A}) = \Pr_{(\mathsf{vk},m,m',\sigma)\leftarrow\mathcal{A}}(\Sigma.\mathsf{Verify}(\mathsf{vk}, m, \sigma) = \Sigma.\mathsf{Verify}(\mathsf{vk}, m', \sigma) = 1 \wedge m \neq m').$$

Definition 10 (Non-Resignability [13]). *The advantage of adversaries* $\mathcal{A}$, $\mathcal{B}$ *against the* non-resignability *property of a signature scheme* Σ *and a randomized algorithm* aux *that, on input a message and a signing key, returns a so-called hint is defined as:*

$$\mathsf{Adv}^{\mathsf{NR}}_{\Sigma,\mathsf{aux}}(\mathcal{A}, \mathcal{B}) = \Pr_{\substack{(\mathsf{vk},\mathsf{sk})\leftarrow\Sigma.\mathsf{KeyGen}(1^\lambda)\\ m\leftarrow\mathcal{B}(\mathsf{sk})\\ \sigma\leftarrow\Sigma.\mathsf{Sign}(\mathsf{sk},m)\\ (\mathsf{vk}^*,\sigma^*)\leftarrow\mathcal{A}(\mathsf{sk},\sigma,\mathsf{aux}(m,\mathsf{sk}))}}(\Sigma.\mathsf{Verify}(\mathsf{vk}^*, \sigma^*, m) = 1 \wedge \mathsf{vk}^* \neq \mathsf{vk}).$$

2.4 Identification Schemes

We recall the definition of an identification scheme as they lie at the core of the Fiat-Shamir transform, which we recall in the next section.

Definition 11 (Identification Scheme). *An identification scheme is a tuple of PPT algorithms* $\mathsf{ID} = (\mathsf{Igen}, \mathsf{P}, \mathsf{V})$ *such that:*

- Igen*: On input the security parameter* 1^λ, *algorithm* Igen *outputs a verification key* vk *and a secret key* sk. *We assume that* vk *defines the challenge space* $\mathcal{C}$.
- P*: The prover* $\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2)$ *is split into two algorithms: given* sk, *algorithm* P_1 *produces a* commitment w *(first message sent to the verifier) and a state* st*; algorithm* P_2, *on input* $(\mathsf{sk}, w, \mathsf{st})$ *and a uniformly random* challenge $c \in \mathcal{C}$ *sent by the verifier in response to commitment* w, *outputs an* answer z.

- V*: On input* (vk, w, c, z)*, the deterministic verifier* V *outputs* 1 *or* 0*.*

We let $\mathsf{P}(\mathsf{sk}, \mathsf{vk}) \leftrightarrow \mathsf{V}(\mathsf{vk})$ *denote the transcript* (w, c, z) *of an interaction between the prover and the verifier, as illustrated in Figure 3.*

P_2 *may output* $z = \perp$*, in which case we say that the identification scheme* aborts*. The probability this happens is called the aborting probability of* ID*.*

We say that ID *has* unique response *if for any* (w, c)*, there is at most one value for* z *such that* (w, c, z) *is a valid transcript.*

$$\begin{array}{|l c r|}\hline \mathsf{P}(\mathsf{sk}, \mathsf{vk}) & & \mathsf{V}(\mathsf{vk}) \\ \hline & \xrightarrow{w} & \\ & \xleftarrow{c} & c \hookleftarrow U(\mathcal{C}) \\ & \xrightarrow{z} & \text{Accept or Reject} \\ \hline \end{array}$$

Fig. 3: Interaction between P and V

We start by recalling *completeness* and *commitment-recoverability*, two properties which allow to define and prove the correctness of $\mathsf{FS}[\mathsf{ID}, H]$.

Definition 12 (Completeness and commitment-recoverability). *An identification scheme* $\mathsf{ID} = (\mathsf{Igen}, \mathsf{P}, \mathsf{V})$ *is* complete *if for any* $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Igen}(1^\lambda)$*, for any challenge* $c \in \mathcal{C}$*, we have:*

$$\Pr\Big[\mathsf{V}(\mathsf{vk}, (w, c, z)) = 1 \mid (w, c, z) \leftarrow (\mathsf{P}(\mathsf{sk}, \mathsf{vk}) \leftrightarrow \mathsf{V}(\mathsf{vk})) \land z \neq \perp\Big] = 1,$$

where the randomness is taken over the random coins of P*.*

In addition, ID *satisfies* commitment-recoverability *if for any public key* vk*, challenge* $c \in \mathcal{C}$*, and answer* z*, there is at most one commitment* w *such that the transcript* (w, c, z) *is valid, and there exists a PPT algorithm* $\mathsf{ID.Rec}$ *such that* $w = \mathsf{ID.Rec}(\mathsf{vk}, c, z)$*.*

Unless otherwise specified, we only consider identification schemes satisfying commitment-recoverability. The notions of *honest-verifier zero-knowledge* and *commitment min-entropy* allow to reduce the $\mathsf{EU\text{-}CMA}$ game of $\mathsf{FS}[\mathsf{ID}, H]$ to its $\mathsf{EU\text{-}NMA}$ one.

Definition 13 (HVZK and commitment min-entropy). *An identification scheme* $\mathsf{ID} = (\mathsf{Igen}, \mathsf{P}, \mathsf{V})$ *is* Honest-Verifier Zero-Knowledge *(HVZK) if there exists a PPT simulator* Sim *such that, conditioned on* $z \neq \perp$*:*

$$\Delta\Big((w, c, z) \leftarrow (\mathsf{P}(\mathsf{sk}, \mathsf{vk}) \leftrightarrow \mathsf{V}(\mathsf{vk}))\ ,\ \mathsf{Sim}(c, \mathsf{vk})\Big) = 0.$$

ID *has* α bits of commitment min-entropy *if for any* $(\mathsf{vk}, \mathsf{sk})$ *in the range of* IGen*:*

$$H_\infty\Big(w | (w, c, z) \leftarrow (\mathsf{P}(\mathsf{sk}, \mathsf{vk}) \leftrightarrow \mathsf{V}(\mathsf{vk}))\Big) \geq \alpha.$$

The $\mathsf{EU\text{-}NMA}$ problem is linked via the Forking Lemma to the 2-special-soundness of the identification scheme, which asks an adversary to find two valid transcripts starting with the same commitment. In the context of lattice-based schemes, it is useful to introduce a lossy key generation algorithm, giving rise to the lossy-2-special-soundness property. In this work, we actually use a weaker notion: the adversary does not choose the second challenge. Looking ahead, this relaxation allows our modified identification schemes in Appendices A and B to satisfy this notion, while still allowing reductions to use the Forking Lemma.

Definition 14 (Soundness). *Let* ID *be an identification scheme with challenge space* $\mathcal{C}$ *and let* $\mathsf{LossyIGen}$ *be an algorithm that on input* 1^λ *returns* vk.

- *For any distinguisher* $\mathcal{B}$*, its advantage in the* key-indistinguishability with respect to $\mathsf{LossyIGen}$ *game is defined as:*

$$\mathsf{Adv}^{\mathsf{key\text{-}ind}}_{\mathsf{ID}}(\mathcal{B}) = \left| \Pr_{\mathsf{vk} \leftarrow \mathsf{LossyIGen}(1^\lambda)} (\mathcal{B}(\mathsf{vk}) \to 1) - \Pr_{(\mathsf{vk},\mathsf{sk}) \leftarrow \mathsf{ID.IGen}(1^\lambda)} (\mathcal{B}(\mathsf{vk}) \to 1) \right| .$$

- *For any adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, its advantage in the* lossy-2-special-soundness with respect to $\mathsf{LossyIGen}$ *game is defined as:*

$$\mathsf{Adv}^{\mathsf{lossy\text{-}2\text{-}ss}}_{\mathsf{ID}}(\mathcal{A}) = \Pr_{\substack{\mathsf{vk} \leftarrow \mathsf{LossyIGen}(1^\lambda) \\ (w,c_1,z_1,st) \leftarrow \mathcal{A}_1(\mathsf{vk}) \\ c_2 \hookleftarrow U(\mathcal{C} \setminus \{c_1\}) \\ z_2 \leftarrow \mathcal{A}_2(st,z_2)}} (\mathsf{V}(\mathsf{vk}, w, c_1, z_1) = \mathsf{V}(\mathsf{vk}, w, c_2, z_2) = 1).$$

If $\mathsf{LossyIGen}$ *calls* $\mathsf{ID.IGen}$ *and returns* vk*, we call this game* 2-special-soundness*.*

Finally, the notion of computational unique response (CUR) is necessary to achieve strong unforgeability.

Definition 15 (CUR). *Let* ID *be an identification scheme. For any adversary* $\mathcal{A}$*, its advantage in the* CUR *game is defined as:*

$$\mathsf{Adv}^{\mathsf{CUR}}_{\mathsf{ID}}(\mathcal{A}) = \Pr_{\substack{(\mathsf{sk},\mathsf{vk}) \leftarrow \mathsf{ID.IGen}(1^\lambda) \\ (w,c,z,z') \leftarrow \mathcal{A}(\mathsf{vk})}} (\mathsf{V}(\mathsf{vk}, w, c, z) = \mathsf{V}(\mathsf{vk}, w, c, z') = 1 \ \wedge \ z \neq z').$$

2.5 Fiat-Shamir with Aborts Transform

We now recall the Fiat-Shamir with Aborts (FSwA) transform in Figure 4, which allows to transform a commitment-recoverable identification scheme into a digital signature, even when the identification scheme has a nonzero aborting probability. Essentially, it asks the signer to run as many times as necessary a non-interactive version of the identification scheme, where the challenge is sampled as the hash of the commitment w and the message to sign μ. The signature is the first pair (c, z) such that $z \neq \bot$, which is verified by first recovering $w = \mathsf{ID.Rec}(\mathsf{vk}, c, z)$ and then by checking that $c = H(w, \mu)$. Finally, the validity of the transcript (w, c, z) is verified.

The resulting signature is correct as long as ID is complete, and its HVZK property allows to reduce the $\mathsf{EU\text{-}CMA}$ game to the $\mathsf{EU\text{-}NMA}$ game, as long as ID has sufficiently high commitment min-entropy.

```
KeyGen(1^λ):
1: $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{ID.IGen}(1^\lambda)$
2: return $(\mathsf{vk}, \mathsf{sk})$
```

```
Sign(sk, μ):
1: $z \leftarrow \perp$
2: while $z = \perp$ do
3:     $(w, \mathsf{st}) \leftarrow \mathsf{ID.P}_1(\mathsf{sk})$
4:     $c \leftarrow H(w, \mu)$
5:     $z \leftarrow \mathsf{ID.P}_2(\mathsf{sk}, \mathsf{st}, w, c)$
6: return $(c, z)$
```

```
Verify(vk, (c, z), μ):
1: $w \leftarrow \mathsf{ID.Rec}(\mathsf{vk}, c, z)$
2: if $c \neq H(w, \mu)$ then
3:     return 0
4: $b \leftarrow \mathsf{ID.V}(\mathsf{vk}, (w, c, z))$
5: return $b$
```

Fig. 4: Fiat-Shamir with Aborts Signature $\mathsf{FSwA}[\mathsf{ID}, H]$.

3 Hybrid Security for Signature Schemes

In practice, the usual EU-CMA notion or its strong variant are not sufficient for hybrid signatures in the wild, as highlighted by existing non-separability notions. We therefore introduce a new security notion, called *Hybrid Unforgeability*, for hybrid signature schemes where an adversary is given access to signature oracles for the hybrid signature and its two components, and must forge a signature for a target scheme out of the three. The goal of this new notion is to capture the various attack vectors enabled by the post-quantum transition scenarios illustrated in Figure 1 where a hybrid signature scheme coexists with its components.

Definition 16 (Hybrid Unforgeability). *Let $\Sigma_\mathsf{H}, \Sigma_1, \Sigma_2$ be three signature schemes such that $\Sigma_\mathsf{H}.\mathsf{KeyGen}(1^\lambda)$ calls $(\mathsf{vk}_i, \mathsf{sk}_i) \leftarrow \Sigma_i.\mathsf{KeyGen}(1^\lambda)$ for $i \in \{1, 2\}$ and then returns $\mathsf{vk}_\mathsf{H} = (\mathsf{vk}_1, \mathsf{vk}_2)$ and $\mathsf{sk}_\mathsf{H} = (\mathsf{sk}_1, \mathsf{sk}_2)$ (up to bijection).*

Let $\mathsf{X} \in \{\mathsf{H}, 1, 2\}$. For any $\mathsf{Y} \in \{\mathsf{H}, 1, 2\}$, let $\mathcal{O}_\mathsf{Y}(\mathsf{sk}_\mathsf{Y})$ denote the oracle that, on input μ, returns $\Sigma_\mathsf{Y}.\mathsf{Sign}(\mathsf{sk}_\mathsf{Y}, \mu)$. We define the $\mathsf{H\text{-}EU\text{-}CMA}^\mathsf{X}_{Q_1,Q_2,Q_\mathsf{H}}$ advantage of an adversary $\mathcal{A}$ making Q_Y queries to $\mathcal{O}_\mathsf{Y}(\mathsf{sk}_\mathsf{Y})$, for any $\mathsf{Y} \in \{\mathsf{H}, 1, 2\}$ as:

$$\mathsf{Adv}^{\mathsf{H\text{-}EU\text{-}CMA}^\mathsf{X}}_{Q_1,Q_2,Q_\mathsf{H}}(\mathcal{A}) = \Pr_{\substack{(\mathsf{vk}_1,\mathsf{sk}_1)\leftarrow\Sigma_1.\mathsf{KeyGen}(1^\lambda)\\ (\mathsf{vk}_2,\mathsf{sk}_2)\leftarrow\Sigma_2.\mathsf{KeyGen}(1^\lambda)\\ (\sigma^*,\mu^*)\leftarrow\mathcal{A}^{\mathcal{O}_\mathsf{H}(\mathsf{sk}_\mathsf{H}),\mathcal{O}_1(\mathsf{sk}_1),\mathcal{O}_2(\mathsf{sk}_2)}(\mathsf{vk}_1,\mathsf{vk}_2)}} \begin{pmatrix} \Sigma_\mathsf{X}.\mathsf{Verify}(\mathsf{vk}_\mathsf{X}, \sigma^*, \mu^*) = 1 \\ \wedge \\ \mu^* \text{ was not queried to } \mathcal{O}_\mathsf{X} \end{pmatrix}.$$

We omit Q_1, Q_2, Q_H from the name of the game whenever clear from context. If we replace the condition on μ^* with "σ^* was not $\mathcal{O}_\mathsf{X}$'s answer to a signature query for μ^*", we instead denote the game with $\mathsf{sH\text{-}EU\text{-}CMA}^\mathsf{X}$, standing for strong hybrid unforgeability.

Remark 1. Conveniently, our new hybrid unforgeability notion subsumes several existing ones. In particular, the $\mathsf{H\text{-}EU\text{-}CMA}^\mathsf{H}$ (resp. $\mathsf{sH\text{-}EU\text{-}CMA}^\mathsf{H}$) game coincides with the standard EU-CMA (resp. sEU-CMA) game of Σ_H when $Q_1 = Q_2 = 0$.

Similarly, when $Q_1 = Q_2 = Q_\mathsf{H} = 0$, the $\mathsf{H\text{-}EU\text{-}CMA}^\mathsf{H}$ game coincides with the EU-NMA one for Σ_H.

4 Fiat-Shamir with *Partial* Aborts and Identification Schemes Concatenation

Assuming that we have two identification schemes, of which at most one may have aborts, we exhibit a variant of the BUFF Fiat-Shamir transform to turn them into a hybrid signature scheme satisfying all three $\mathsf{H\text{-}EU\text{-}CMA}^{\mathsf{X}}$ security notions, $\mathsf{X} \in \{1, 2, \mathsf{H}\}$. It can be seen as an application of the Fiat-Shamir transform over the concatenation of the two identification schemes, with the following tweaks. First, the commitment of the non-aborting protocol is not resampled during the rejection phase of the aborting protocol. Second, it is included in the BUFF hash of the message and not in the rejection loop to save time during signing. Last, the two protocols share the raw output of the hash function, and it is up to them to use it to deduce their respective challenge. Note that during key generation, we restart if the verification keys are identical, which should never happen in practice as different identification schemes usually have different verification key sizes and verification keys have large entropy.

4.1 Description and Correctness of the Signature Scheme

Our main construction, denoted $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H]$, is presented in Figure 5. The two signing algorithms shown are functionally identical and differ only in the way they are written, see Figure 4 for further details. The signature scheme relies on two hash functions $H : \{0,1\}^* \to \{0,1\}^\ell$ and $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\alpha$. The former is modeled as a random oracle in the proofs, while the security of the scheme relies on the collision resistance of the latter.

Theorem 1 (Correctness). *Let $\ell > 0$. Let ID_1 and ID_2 be two complete identification schemes with ID_1 (resp. ID_2) having aborting probability 0 (resp. β), and challenge space $\{0,1\}^\ell$ for both. The scheme $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H]$ defined in Figure 5 is a correct signature scheme. Moreover, the runtime of Sign is at most the sum of those of $\mathsf{FS}[\mathsf{ID}_1, H].\mathsf{Sign}$ and $\mathsf{BUFF}(\mathsf{FSwA}[\mathsf{ID}_2, H]).\mathsf{Sign}$.*

Proof. For any $i \in \{1, 2\}$ and genuine transcripts (w_i, c, z_i) such that $z_2 \neq \bot$ and $w_i = \mathsf{ID}_i.\mathsf{Rec}(\mathsf{vk}_i, c, z_i)$, it holds that $\mathsf{ID}_i.\mathsf{V}$ accepts this transcript, by completeness. In particular, the distribution of c does not matter. The runtime of Sign is most easily analyzed using the second description of the algorithm, which makes calls to $\mathsf{FSwA}[\mathsf{ID}_2, H].\mathsf{Sign}$. All remaining operations either correspond to a call to $\mathcal{H}$, included in BUFF, or are already present in $\mathsf{FS}[\mathsf{ID}_1, H].\mathsf{Sign}$. □

4.2 (s)$\mathsf{H\text{-}EU\text{-}CMA}^{\mathsf{H}}$ Security in the QROM

As a warm-up, we show that $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H]$ is at least as secure as the signature scheme $\mathsf{FSwA}[\mathsf{ID}_2, H]$ in the QROM, even in the presence of additional oracles for its components, whatever choice for Σ_1 we make.

```
KeyGen($1^\lambda$):
1: repeat
2:     $(\mathsf{vk}_1, \mathsf{sk}_1) \leftarrow \mathsf{ID}_1.\mathsf{IGen}(1^\lambda)$
3:     $(\mathsf{vk}_2, \mathsf{sk}_2) \leftarrow \mathsf{ID}_2.\mathsf{IGen}(1^\lambda)$
4: until $\mathsf{vk}_1 \neq \mathsf{vk}_2$
5: $tr \leftarrow \mathcal{H}(\mathsf{vk}_1 \| \mathsf{vk}_2)$
6: return $((\mathsf{vk}_1, \mathsf{vk}_2), (tr, \mathsf{sk}_1, \mathsf{sk}_2))$

Sign($(tr, \mathsf{sk}_1, \mathsf{sk}_2), M$):
1: $(w_1, st_1) \leftarrow \mathsf{ID}_1.\mathsf{P}_1(\mathsf{sk}_1)$
2: $\mu \leftarrow \mathcal{H}(tr \| w_1 \| M)$
3: $z_2 \leftarrow \perp$
4: while $z_2 = \perp$ do
5:     $(w_2, st_2) \leftarrow \mathsf{ID}_2.\mathsf{P}_1(\mathsf{sk}_2)$
6:     $c = H(w_2 \| \mu)$
7:     $z_2 \leftarrow \mathsf{ID}_2.\mathsf{P}_2(\mathsf{sk}_2, c, st_2)$
8: $z_1 \leftarrow \mathsf{ID}_1.\mathsf{P}_2(\mathsf{sk}_1, c, st_1)$
9: return $(z_1, z_2, c)$
```

```
Verify($(\mathsf{vk}_1, \mathsf{vk}_2), (z_1, z_2, c), M$):
1: $w_1 \leftarrow \mathsf{ID}_1.\mathsf{Rec}(\mathsf{vk}_1, c, z_1)$
2: $\mu \leftarrow \mathcal{H}(\mathcal{H}(\mathsf{vk}_1 \| \mathsf{vk}_2) \| w_1 \| M)$
3: $w_2 \leftarrow \mathsf{ID}_2.\mathsf{Rec}(\mathsf{vk}_2, c, z_2)$
4: if $H(w_2 \| \mu) \neq c$ then
5:     return 0
6: else if not $\mathsf{ID}_1.\mathsf{V}(w_1, c, z_1)$ then
7:     return 0
8: else if not $\mathsf{ID}_2.\mathsf{V}(w_2, c, z_2)$ then
9:     return 0
10: return 1

Sign($(tr, \mathsf{sk}_1, \mathsf{sk}_2), M$):
1: $(w_1, st_1) \leftarrow \mathsf{ID}_1.\mathsf{P}_1(\mathsf{sk}_1)$
2: $\mu \leftarrow \mathcal{H}(tr \| w_1 \| M)$
3: $(z_2, c) \leftarrow \mathsf{FSwA}[\mathsf{ID}_2, H].\mathsf{Sign}(\mathsf{sk}_2, \mu)$
4: $z_1 \leftarrow \mathsf{ID}_1.\mathsf{P}_2(\mathsf{sk}_1, c, st_1)$
5: return $(z_1, z_2, c)$
```

Fig. 5: Hybrid Fiat-Shamir Transform $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H]$. The two formulations of Sign are functionally equivalent.

Theorem 2 (QROM Security). *Let $\ell > 0$. Let ID_1 (resp. ID_2) be an identification scheme satisfying HVZK, with aborting probability 0 (resp. β), α_1 (resp. α_2) bits of commitment min-entropy and challenge space $\{0,1\}^\ell$. Let Σ_1 be a signature scheme such that $\Sigma_1.\mathsf{KeyGen} = \mathsf{ID}_1.\mathsf{IGen}$, up to bijection. Let $\Sigma_\mathsf{H} = \mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H]$ as defined in Figure 5 and $\Sigma_2 = \mathsf{BUFF}(\mathsf{FSwA}[\mathsf{ID}_2, H])$.*

In the QROM, by modeling H as a random oracle, there exist adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$, explicitly given in the proof of this theorem, such that for any hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\alpha$, and adversary $\mathcal{A}$, we have that:

$$\mathsf{Adv}^{\mathsf{H\text{-}EU\text{-}CMA}^\mathsf{H}}_{Q_2,Q_2,Q_\mathsf{H}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{EU\text{-}CMA}}_{\mathsf{FSwA}[\mathsf{ID}_2,H]}(\mathcal{B}_1) + \mathsf{Adv}^{\mathsf{col}}_{\mathcal{H}}(\mathcal{B}_2) \ .$$

where $\mathcal{B}_1$ and $\mathcal{B}_2$ make the same amount of quantum random oracle queries as $\mathcal{A}$, make $Q_2 + Q_\mathsf{H}$ classical signature queries and have essentially the same runtime as $\mathcal{A}$. Assuming that ID_1 has unique response, we moreover have

$$\mathsf{Adv}^{\mathsf{sH\text{-}EU\text{-}CMA}^\mathsf{H}}_{Q_1,Q_2,Q_\mathsf{H}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{sEU\text{-}CMA}}_{\mathsf{FSwA}[\mathsf{ID}_2,H]}(\mathcal{B}_1) + \mathsf{Adv}^{\mathsf{col}}_{\mathcal{H}}(\mathcal{B}_2) \ .$$

Proof. The adversary $\mathcal{B}_1$, on input vk_2, runs $\mathsf{ID}_1.\mathsf{IGen}$ to get $\mathsf{sk}_1, \mathsf{vk}_1$. It is then able to run the rest of $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H].\mathsf{KeyGen}(1^\lambda)$ and calls $\mathcal{A}$ on $(\mathsf{vk}_1, \mathsf{vk}_2)$. It answers $\mathcal{A}$'s queries in the following manner.

Hash queries. $\mathcal{B}_1$ forwards the query to its own random oracle and forwards back the answer to $\mathcal{A}$.

Σ_1 **signature.** On input M, $\mathcal{B}_1$ is able to run $\Sigma_1.\mathsf{Sign}(\mathsf{sk}_1, \mathcal{H}(\mathcal{H}(\mathsf{vk}_1) \| M))$ and returns the answer to $\mathcal{A}$.

Σ_2 **signature.** On input M, $\mathcal{B}_1$ computes $\mu = \mathcal{H}(\mathcal{H}(\mathsf{vk}_2)\|M)$ and calls its signature oracle on μ. It returns the answer of its oracle to $\mathcal{A}$.

Σ_{H} **signature.** On input M, $\mathcal{B}_1$ first runs $(w_1, st_1) \leftarrow \mathsf{ID}_1.\mathsf{P}_1(\mathsf{sk}_1)$, then computes $\mu = \mathcal{H}(tr\|w_1\|M)$ and calls its signature oracle on μ to get (z_2, c). It then computes $z_1 \leftarrow \mathsf{ID}_1.\mathsf{P}_2(\mathsf{sk}_1, c, st_1)$ and returns (z_1, z_2, c) to $\mathcal{A}$.

When $\mathcal{A}$ outputs a forgery (z_1^*, z_2^*, c^*) for message M^*, $\mathcal{B}_1$ outputs (z_2^*, c^*) as its forgery, for the message $\mu^* = \mathcal{H}(tr\|\mathsf{ID}_1.\mathsf{Rec}(\mathsf{vk}_1, z_1^*, c^*)\|M^*)$.

We study the implications of the existence of a signature query for μ_i made by $\mathcal{B}_1$ such that $\mu^* = \mu_i$. This signature query was made in response to a signature query made by $\mathcal{A}$, for some message M. Let (z_2, c) be the answer to the query made by $\mathcal{B}_1$ and $w_1^* = \mathsf{ID}_1.\mathsf{Rec}(\mathsf{vk}_1, z_1^*, c^*)$. Two cases arise.

(i) $\mu_i = \mathcal{H}(tr\|w_1\|M)$, i.e. $\mathcal{B}_1$ was responding to a Σ_{H} signature query. There are two subcases.
   (a) $w_1\|M = w_1^*\|M^*$. In particular, $w_1 = w_1^*$. Assuming that (z_1^*, z_2^*, c^*) is accepted by $\Sigma_{\mathsf{H}}.\mathsf{Verify}$, due to the unique response of ID_1, either $c \neq c^*$ or (z_1^*, z_2, c^*) was the signature returned to $\mathcal{A}$ after its query for M.
   (b) $w_1\|M \neq w_1^*\|M^*$. As $\mu^* = \mu_i$, we have two preimages for μ^* with different suffixes for $\mathcal{H}$.

(ii) $\mu_i = \mathcal{H}(\mathcal{H}(\mathsf{vk}_2)\|M)$, i.e. $\mathcal{B}_1$ was responding to a Σ_2 signature query. There are again two subcases.
   (a) $tr = \mathcal{H}(\mathsf{vk}_2)$. As $\mathsf{vk}_2 \neq \mathsf{vk}_1\|\mathsf{vk}_2$, we have two preimages for tr for $\mathcal{H}$.
   (b) $tr \neq \mathcal{H}(\mathsf{vk}_2)$. As $\mu^* = \mu_i$, we have two preimages, with different prefixes, for μ^* for $\mathcal{H}$.

In the $\mathsf{H\text{-}EU\text{-}CMA}^{\mathsf{H}}$ contexts, $\mathcal{B}_1$ wins only if (z_2^*, c^*) is a valid signature for μ^* for $\mathsf{FSwA}[\mathsf{ID}_2, H]$ and μ^* was not queried by $\mathcal{B}_1$. If $\mathcal{A}$ wins, then (z_1^*, z_2^*, c^*) is a valid signature for M^* for Σ_{H} and M^* was not queried to its Σ_{H} oracle. Then, if $\mathcal{A}$ wins, (z_2^*, c^*) is a valid signature for μ^* for $\mathsf{FSwA}[\mathsf{ID}_2, H]$, and $\mathcal{B}_1$ wins if it did not query μ^*. Moreover, if $\mathcal{A}$ wins and μ^* was queried, then we are in one of the above cases, with the exception of case (i)a, as $\mathcal{A}$ did not make a query for M^* to its Σ_{H} oracle. In all of those cases, we found a collision for $\mathcal{H}$, and $\mathcal{B}_2$ is defined as running $\mathcal{B}_1$ and its challenger, and checking whether one of those cases is realized. Then, if $\mathcal{A}$ wins, either $\mathcal{B}_1$ wins too or $\mathcal{B}_2$ finds a collision for $\mathcal{H}$.

The $\mathsf{sH\text{-}EU\text{-}CMA}^{\mathsf{H}}$ context is identical with the exception of the case (i)a. If $\mathcal{A}$ wins and case (i)a happens, we must have $(z_1^*, z_2^*, c^*) \neq (z_1, z_2, c)$. It is then impossible to have $(z_2^*, c^*) = (z_2, c)$ as it implies that $z_1 = z_1^*$, as shown above. Hence, if $\mathcal{A}$ wins the $\mathsf{sH\text{-}EU\text{-}CMA}^{\mathsf{H}}$ game and we are in case (i)a, then $\mathcal{B}_1$ wins the $\mathsf{sEU\text{-}CMA}$ game too, yielding the same conclusion.

□

4.3 (s)H-EU-CMA Security in the ROM

We now show that our construction satisfies all H-EU-CMA security flavors in the ROM. Our proof strategy follows from standard Fiat-Shamir security arguments: we simulate the three signature oracles using the HVZK property of ID_1 and ID_2, while reprogramming the random oracle as needed. The main technical challenge lies in bounding the probability that an input is reprogrammed more than once, since all three signature oracles share the same random oracle—making this step more intricate than in the standard $\mathsf{EU\text{-}CMA}$ setting. Once this issue is handled, we can get rid of the signature oracles, and reduce the $\mathsf{H\text{-}EU\text{-}CMA}^{\mathsf{X}}$ security to the $\mathsf{EU\text{-}NMA}$ security of Σ_{X}.

Extending the proof to the strong variant adds an adversary against the CUR property of ID_1 and ID_2, as is also the case for standard Fiat-Shamir signatures.

Lemma 3. *Let $\ell > 0$. Let ID_1 (resp. ID_2) be an identification scheme satisfying HVZK, with aborting probability 0 (resp. β), α_1 (resp. α_2) bits of commitment min-entropy and challenge space $\{0,1\}^\ell$.*

Let $\Sigma_{\mathsf{H}} = \mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H]$ as in Figure 5. Let $\Sigma_1 = \mathsf{BUFF}(\mathsf{FS}[\mathsf{ID}_1, H])$ and $\Sigma_2 = \mathsf{BUFF}(\mathsf{FSwA}[\mathsf{ID}_2, H])$. Let $\mathcal{O}_i(\mathsf{sk}_i)$ be the oracle that on input M answers with $\Sigma_i.\mathsf{Sign}(\mathsf{sk}_i, M)$ for $i \in \{\mathsf{H}, 1, 2\}$.

In the ROM, by modeling H as a random oracle, there exist PPT classical adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$, explicitly given in the proof of this theorem, such that for any $\mathsf{X} \in \{\mathsf{H}, 1, 2\}$ and hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\alpha$, and PPT classical adversary $\mathcal{A}$ making Q_i queries to oracle $\mathcal{O}_i(\mathsf{sk}_i)$ for $i \in \{\mathsf{H}, 1, 2\}$ and Q_h queries to the random oracle, we have that

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H\text{-}EU\text{-}CMA}^{\mathsf{X}}}_{Q_1,Q_2,Q_{\mathsf{H}}}(\mathcal{A}) &\leq \frac{\beta}{(1-\beta)^2} \cdot \left(Q_{\mathsf{H}} \cdot 2^{-\alpha_1-\alpha_2} + Q_2 \cdot 2^{-\alpha_2}\right) \\ &+ \left(Q_h + Q_1 + \frac{Q_2 + Q_{\mathsf{H}}}{1-\beta}\right)\left(\frac{Q_{\mathsf{H}} \cdot 2^{-\alpha_1-\alpha_2}}{1-\beta} + Q_1 \cdot 2^{-\alpha_1} + \frac{Q_2 \cdot 2^{-\alpha_2}}{1-\beta}\right) \\ &+ \mathsf{Adv}^{\mathsf{EU\text{-}NMA}}_{\Sigma_{\mathsf{X}}}(\mathcal{B}_0) + \mathsf{Adv}^{\mathsf{col}}_{\mathcal{H}}(\mathcal{B}_1)\ . \qquad (1)\end{aligned}$$

If $\mathcal{A}$ plays the $\mathsf{sH\text{-}EU\text{-}CMA}^{\mathsf{X}}$ game, there exist two more adversaries $\mathcal{B}_2$ and $\mathcal{B}_3$ explicitly given in the proof such that Equation 1 holds by adding $\mathsf{Adv}^{\mathsf{CUR}}_{\mathsf{ID}_{\mathsf{X}}}(\mathcal{B}_{1+\mathsf{X}})$ if $\mathsf{X} \neq \mathsf{H}$, otherwise $\mathsf{Adv}^{\mathsf{CUR}}_{\mathsf{ID}_1}(\mathcal{B}_2) + \mathsf{Adv}^{\mathsf{CUR}}_{\mathsf{ID}_2}(\mathcal{B}_3)$ to the right-hand side.

Proof. First, we introduce intermediate oracles in Figure 6, where oracles $\mathcal{O}^u_{\mathsf{X}}$ for $\mathsf{X} \in \{1, 2, \mathsf{H}\}$ replace the generation of the challenge c with a uniformly sampled one. They patch accordingly the random oracle afterwards. The oracle $\mathcal{O}^{s_2}_{\mathsf{H}}$ moreover simulates the ID_2 transcript using its HVZK simulator.

We consider the difference when $\mathcal{A}$ is given access to $\mathcal{O}_{\mathsf{H}}, \mathcal{O}_1, \mathcal{O}_2, H$, *i.e.* the $\mathsf{H\text{-}EU\text{-}CMA}^{\mathsf{X}}$ game and when it is given access to $\mathcal{O}^u_{\mathsf{H}}, \mathcal{O}^u_1, \mathcal{O}^u_2, H$, which we dub the $\mathsf{Game}^{\mathsf{X}}_1$ game. Let $\{s_i\}_i$ be the set of all bit strings of the form $w_1\|\mu$ or $w_2\|\mu$ generated during oracle calls to $\mathcal{O}_1, \mathcal{O}_2$ or $\mathcal{O}_{\mathsf{H}}$ (resp. $\mathcal{O}^u_1, \mathcal{O}^u_2$ or $\mathcal{O}^u_{\mathsf{H}}$), even the ones that are rejected, as well as bit strings sent by $\mathcal{A}$ to the random oracle H. We consider the following event Bad: "$\exists i < j, s_i = s_j$ and s_j was not a query from $\mathcal{A}$ to the random oracle". Conditioned on this event not happening, the oracles are strictly identical in both cases: no random oracle output is overwritten, all of them are chosen uniformly, and values that are not programmed during rejected iterations are not accessed afterwards. As such, we have

$$\mathsf{Adv}^{\mathsf{H\text{-}EU\text{-}CMA}^{\mathsf{X}}}_{Q_1,Q_2,Q_{\mathsf{H}}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{Game}^{\mathsf{X}}_1}_{Q_1,Q_2,Q_{\mathsf{H}}}(\mathcal{A}) + \Pr(\mathsf{Bad}).$$

```
$\mathcal{O}^u_{\mathsf{H}}(tr, \mathsf{sk}_1, \mathsf{sk}_2)(M)$:
 1: $(w_1, st_1) \leftarrow \mathsf{ID}_1.\mathsf{P}_1(\mathsf{sk}_1)$
 2: $\mu \leftarrow \mathcal{H}(tr \| w_1 \| M)$
 3: $z_2 \leftarrow \perp$
 4: while $z_2 = \perp$ do
 5:     $(w_2, st_2) \leftarrow \mathsf{ID}_2.\mathsf{P}_1(\mathsf{sk}_2)$
 6:     $c \hookleftarrow U(\{0,1\}^\ell)$
 7:     $z_2 \leftarrow \mathsf{ID}_2.\mathsf{P}_2(\mathsf{sk}_2, c, st_2)$
 8: $z_1 \leftarrow \mathsf{ID}_1.\mathsf{P}_2(\mathsf{sk}_1, c, st_1)$
 9: Set $H(w_2 \| \mu) \leftarrow c$
10: return $(z_1, z_2, c)$
```

```
$\mathcal{O}^u_1(\mathsf{sk}_1)(M)$:
 1: $\mu \leftarrow \mathcal{H}(\mathcal{H}(\mathsf{vk}_1) \| M)$
 2: $c \hookleftarrow U(\{0,1\}^\ell)$
 3: $(w_1, st_1) \leftarrow \mathsf{ID}_1.\mathsf{P}_1(\mathsf{sk}_1)$
 4: $z_1 \leftarrow \mathsf{ID}_1.\mathsf{P}_2(\mathsf{sk}_1, c, st_1)$
 5: Set $H(w_1 \| \mu) \leftarrow c$
 6: return $(z_1, c)$
```

```
$\mathcal{O}^{s2}_{\mathsf{H}}(tr, \mathsf{sk}_1, \mathsf{vk}_2)(M)$:
 1: $c \hookleftarrow U(\{0,1\}^\ell)$
 2: $(w_1, st_1) \leftarrow \mathsf{ID}_1.\mathsf{P}_1(\mathsf{sk}_1)$
 3: $\mu \leftarrow \mathcal{H}(tr \| w_1 \| M)$
 4: $(w_2, z_2) \leftarrow \mathsf{Sim}_2(\mathsf{vk}_2, c)$
 5: $z_1 \leftarrow \mathsf{ID}_1.\mathsf{P}_2(\mathsf{sk}_1, c, st_1)$
 6: Set $H(w_2 \| \mu) \leftarrow c$
 7: return $(z_1, z_2, c)$
```

```
$\mathcal{O}^u_2(\mathsf{sk}_2)(M)$:
 1: $\mu \leftarrow \mathcal{H}(\mathcal{H}(\mathsf{vk}_2) \| M)$
 2: $z_2 \leftarrow \perp$
 3: while $z_2 = \perp$ do
 4:     $(w_2, st_2) \leftarrow \mathsf{ID}_2.\mathsf{P}_1(\mathsf{sk}_2)$
 5:     $c \hookleftarrow U(\{0,1\}^\ell)$
 6:     $z_2 \leftarrow \mathsf{ID}_2.\mathsf{P}_2(\mathsf{sk}_2, c, st_2)$
 7: Set $H(w_2 \| \mu) \leftarrow c$
 8: return $(z_2, c)$
```

Fig. 6: Intermediary signature oracles.

We assess the probability of Bad happening in $\mathsf{Game}^{\mathsf{X}}_1$ by letting $B^{(\mathsf{H})}_i$ (resp. $B^{(2)}_i$) denote the number of iterations necessary to answer the i-th $\mathcal{O}^u_{\mathsf{H}}$ (resp. $\mathcal{O}^u_2$) query for $i \leq Q_{\mathsf{H}}$ (resp. Q_2). Note that $B^{(\mathsf{H})}_i$ and $B^{(2)}_i$ are identically and independently distributed and follow the geometrical law with parameter $1 - \beta$.

As oracle $\mathcal{O}^u_1$ (resp. $\mathcal{O}^u_2$ and $\mathcal{O}^u_{\mathsf{H}}$) sample a commitment for ID_1 (resp. ID_2) and as this commitment has to match the begin of any recorded bit string, using the union bound and the commitment min-entropy of ID_1 and ID_2, we end up with the following upper bound:

$$\Pr\left(\mathsf{Bad} \,\middle|\, \begin{matrix} B^{(\mathsf{H})}_1, \ldots, B^{(\mathsf{H})}_{Q_{\mathsf{H}}} \\ B^{(2)}_1, \ldots, B^{(2)}_{Q_2} \end{matrix}\right) \leq Q \cdot \left(Q_1 \cdot 2^{-\alpha_1} + \sum_{i=1}^{Q_2} B^{(2)}_i \cdot 2^{-\alpha_2} + \sum_{i=1}^{Q_{\mathsf{H}}} B^{(\mathsf{H})}_i \cdot 2^{-\alpha_2}\right),$$

where we let $Q = Q_1 + \sum_{i=1}^{Q_2} B^{(2)}_i + \sum_{i=1}^{Q_{\mathsf{H}}} B^{(\mathsf{H})}_i + Q_h$ be the number of elements in $\{s_i\}_i$. The law of total probabilities with this bound gives:

$$\begin{aligned}
\Pr(\mathsf{Bad}) \leq\ & \mathbb{E}\left(\sum_{i=1}^{Q_{\mathsf{H}}} B^{(\mathsf{H})}_i\right) \cdot \left(Q_h + Q_1 + \mathbb{E}\left(\sum_{j=1}^{Q_2} B^{(2)}_j\right)\right) \cdot 2^{-\alpha_2} \\
& + Q_1 \cdot \left(Q_h + \mathbb{E}\left(\sum_{j=1}^{Q_{\mathsf{H}}} B^{(\mathsf{H})}_j\right) + Q_1 + \mathbb{E}\left(\sum_{j=1}^{Q_2} B^{(2)}_j\right)\right) \cdot 2^{-\alpha_1} \\
& + \mathbb{E}\left(\sum_{i=1}^{Q_2} B^{(2)}_i\right) \cdot \left(Q_h + \mathbb{E}\left(\sum_{j=1}^{Q_{\mathsf{H}}} B^{(\mathsf{H})}_j\right) + Q_1\right) \cdot 2^{-\alpha_2} \\
& + \mathbb{E}\left(\left(\sum_{i=1}^{Q_{\mathsf{H}}} B^{(\mathsf{H})}_i\right)^2\right) \cdot 2^{-\alpha_2} + \mathbb{E}\left(\left(\sum_{i=1}^{Q_2} B^{(2)}_i\right)^2\right) \cdot 2^{-\alpha_2}.
\end{aligned}$$

We identify the mean and 2nd order moment of a sum of geometrical laws, thus $\mathbb{E}(\sum_{i=1}^{Q_{\mathsf{H}}} B_i^{(\mathsf{H})}) = Q_{\mathsf{H}}/(1-\beta)$, $\mathbb{E}(\sum_{i=1}^{Q_2} B_i^{(2)}) = Q_2/(1-\beta)$, and

$$\mathbb{E}\left(\left(\sum_{i=1}^{Q_{\mathsf{Y}}} B_i^{(\mathsf{Y})}\right)^2\right) = \frac{(Q_{\mathsf{Y}})^2}{(1-\beta)^2} + \frac{\beta \cdot Q_{\mathsf{Y}}}{(1-\beta)^2}, \forall \mathsf{Y} \in \{2, \mathsf{H}\}.$$

This gives the bound:

$$\begin{aligned}
\Pr(\mathsf{Bad}) \leq\ & \frac{Q_{\mathsf{H}}}{1-\beta} \cdot \left(Q_h + Q_1 + \frac{Q_2}{1-\beta}\right) \cdot 2^{-\alpha_2} \\
&+ Q_1 \left(Q_h + \frac{Q_{\mathsf{H}}}{1-\beta} + Q_1 + \frac{Q_2}{1-\beta}\right) 2^{-\alpha_1} + \frac{Q_2}{1-\beta}\left(Q_h + Q_1 + \frac{Q_{\mathsf{H}}}{1-\beta}\right) 2^{-\alpha_2} \\
&+ \left(\frac{(Q_{\mathsf{H}})^2}{(1-\beta)^2} + \frac{\beta \cdot Q_{\mathsf{H}}}{(1-\beta)^2}\right) \cdot 2^{-\alpha_2} + \left(\frac{(Q_2)^2}{(1-\beta)^2} + \frac{\beta \cdot Q_2}{(1-\beta)^2}\right) \cdot 2^{-\alpha_2} \\
=\ & \left(Q_h + Q_1 + \frac{Q_2 + Q_{\mathsf{H}}}{1-\beta}\right)\left(\frac{Q_{\mathsf{H}} \cdot 2^{-\alpha_2}}{1-\beta} + Q_1 \cdot 2^{-\alpha_1} + \frac{Q_2 \cdot 2^{-\alpha_2}}{1-\beta}\right) \\
&+ \frac{\beta}{(1-\beta)^2} \cdot \left(Q_{\mathsf{H}} \cdot 2^{-\alpha_2} + Q_2 \cdot 2^{-\alpha_2}\right).
\end{aligned}$$

```
$\mathcal{O}_{\mathsf{H}}^{s}(\mathsf{vk}_1, \mathsf{vk}_2)(M)$:
    $c \hookleftarrow U(\{0,1\}^\ell)$
    $(w_1, z_1) \leftarrow \mathsf{Sim}_1(\mathsf{vk}_1, c)$
    $\mu \leftarrow \mathcal{H}(\mathcal{H}(\mathsf{vk}_1 \| \mathsf{vk}_2) \| w_1 \| M)$
    $(w_2, z_2) \leftarrow \mathsf{Sim}_2(\mathsf{vk}_2, c)$
    Set $H(w_2 \| \mu) \leftarrow c$
    return $(z_1, z_2, c)$

$\mathcal{O}_1^{s}(\mathsf{vk}_1)(M)$:
    $\mu \leftarrow \mathcal{H}(\mathcal{H}(\mathsf{vk}_1) \| M)$
    $c \hookleftarrow U(\{0,1\}^\ell)$
    $(w_1, z_1) \leftarrow \mathsf{Sim}_1(\mathsf{vk}_1, c)$
    Set $H(w_1 \| \mu) \leftarrow c$
    return $(z_1, c)$

$\mathcal{O}_2^{s}(\mathsf{vk}_2)(M)$:
    $\mu \leftarrow \mathcal{H}(\mathcal{H}(\mathsf{vk}_2) \| M)$
    $c \hookleftarrow U(\{0,1\}^\ell)$
    $(w_2, z_2) \leftarrow \mathsf{Sim}_2(\mathsf{vk}_2, c)$
    Set $H(w_2 \| \mu) \leftarrow c$
    return $(z_2, c)$
```

Fig. 7: Simulation oracles.

Next, we replace the $\mathcal{O}_{\mathsf{H}}^{u}$ oracle with $\mathcal{O}_{\mathsf{H}}^{s_2}$, and the $\mathcal{O}_2^{u}$ oracle with $\mathcal{O}_2^{s}$ from Figure 7. By the HVZK property of ID_2, the resulting game $\mathsf{Game}_2^{\mathsf{X}}$ is such that

$$\mathsf{Adv}_{Q_1,Q_2,Q_{\mathsf{H}}}^{\mathsf{Game}_1^{\mathsf{X}}}(\mathcal{A}) = \mathsf{Adv}_{Q_1,Q_2,Q_{\mathsf{H}}}^{\mathsf{Game}_2^{\mathsf{X}}}(\mathcal{A}).$$

Similarly, we use the HVZK property of ID_1, allowing us to replace $\mathcal{O}_{\mathsf{H}}^{s_2}$ with $\mathcal{O}_{\mathsf{H}}^{s}$, and $\mathcal{O}_1^{u}$ with $\mathcal{O}_1^{s}$. The resulting game $\mathsf{Game}_3^{\mathsf{X}}$ is such that

$$\mathsf{Adv}_{Q_1,Q_2,Q_{\mathsf{H}}}^{\mathsf{Game}_2^{\mathsf{X}}}(\mathcal{A}) = \mathsf{Adv}_{Q_1,Q_2,Q_{\mathsf{H}}}^{\mathsf{Game}_3^{\mathsf{X}}}(\mathcal{A}).$$

Finally, we note that the three simulated oracles do not use the signing key. As such, we define an adversary $\mathcal{B}_0$ that only queries the random oracle H, and that calls $\mathcal{A}$ by simulating the three oracles $\mathcal{O}^s_{\mathsf{Y}}$ and random oracle H' which is H except on inputs that were reprogrammed by the other oracles. When $\mathcal{A}$ outputs a forgery σ^*, M^*, the reduction $\mathcal{B}_0$ outputs it as its own. It wins if $\mathcal{A}$ wins, and it did not patch the random oracle on any input used during the verification process. Reprogramming only happens whenever a call to an oracle is made. Since $\mathcal{A}$ must output a message M^* for which it did not query a signature for Σ_{X}, its forgery was either on an input not reprogrammed, or it found a collision for the hash function $\mathcal{H}$, as it implies that the associated μ^* is identical to some μ_i that appeared during some query to oracle $\mathcal{O}_{\mathsf{Y}}$ with message M_i. We have:

$\mathsf{Y} = \mathsf{X}$. As $M_i \neq M^*$, we have two preimages for μ^* by $\mathcal{H}$.

$\mathsf{Y} \neq \mathsf{X}$. There are two subcases.

  $tr_{\mathsf{X}} = tr_{\mathsf{Y}}$. As $\mathsf{vk}_{\mathsf{X}} \neq \mathsf{vk}_{\mathsf{Y}}$, where $\mathsf{vk}_{\mathsf{H}} = \mathsf{vk}_1 \| \mathsf{vk}_2$, we have two preimages for tr_{X} by $\mathcal{H}$.

  $tr_{\mathsf{X}} \neq tr_{\mathsf{Y}}$. As $\mu^* = \mu_i$, we have two preimages for μ^* by $\mathcal{H}$.

The adversary $\mathcal{B}_1$ is defined as running $\mathcal{B}_0$ and its challenger, and returning such a collision, if one is found. This gives the final bound.

In the strong unforgeability setting, we furthermore look into what happens when $M^* = M_i$. If the forgery is not on an input that was reprogrammed, then this forgery is valid for the reduction $\mathcal{B}$. Otherwise, there is a signature σ_i generated by $\mathcal{O}_{\mathsf{X}}$ for M_i such that the challenges contained in σ^* and σ_i are identical, as well as the inputs of the random oracle recovered during verification. We then have:

- If $\mathsf{X} = 1$ or 2, then we extract an adversary $\mathcal{B}_{\mathsf{X}+1}$ against the CUR property of ID_{X}, as we have two transcripts having common commitment and challenge, but different answers, as the forgery differs from the signature.
- If $\mathsf{X} = \mathsf{H}$, we may have found a collision for $\mathcal{H}$ in the case where $\mu^* = \mu_i$ but the ID_1 commitments are different for the signature and the forgery. Otherwise, we extract an adversary $\mathcal{B}_2$ or $\mathcal{B}_3$ against the CUR property of either ID_1 or ID_2, as both the signature and forgery contain a valid transcript for both ID_1 and ID_2, where those transcripts have common commitments and challenges, and either the ID_1 or ID_2 transcripts have different answers.

This concludes the extension of the proof to the $\mathsf{sH\text{-}EU\text{-}CMA}$ setting. □

We show that one can swap Σ_1 with any signature scheme not using H.

Corollary 1. *Let everything as in Lemma 3, except that Σ_1 is any signature scheme such that $\Sigma_1.\mathsf{KeyGen} = \mathsf{ID}_1.\mathsf{IGen}$ and Σ_1 does not use H.*

In the ROM, by modeling H as a random oracle, there exist PPT classical adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$, explicitly given in the proof of this theorem, such that for any $\mathsf{X} \in \{\mathsf{H}, 1, 2\}$ and hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\alpha$, and PPT classical adversary $\mathcal{A}$ making Q_i queries to oracle $\mathcal{O}_i(\mathsf{sk}_i)$ for $i \in \{\mathsf{H}, 1, 2\}$ and Q_h queries *to the random oracle, we have that*

$$\mathsf{Adv}^{\mathsf{H\text{-}EU\text{-}CMA}^{\mathsf{X}}}_{Q_1,Q_2,Q_{\mathsf{H}}}(\mathcal{A}) \le \mathsf{Adv}^{\mathsf{game}}_{\Sigma_{\mathsf{X}}}(\mathcal{B}_0) + \frac{\beta}{(1-\beta)^2} \cdot \left(Q_{\mathsf{H}} \cdot 2^{-\alpha_1-\alpha_2} + Q_2 \cdot 2^{-\alpha_2}\right)$$
$$+ \left(Q_h + \frac{Q_2 + Q_{\mathsf{H}}}{1-\beta}\right)\left(\frac{Q_{\mathsf{H}} \cdot 2^{-\alpha_1-\alpha_2}}{1-\beta} + \frac{Q_2 \cdot 2^{-\alpha_2}}{1-\beta}\right) + \mathsf{Adv}^{\mathsf{col}}_{\mathcal{H}}(\mathcal{B}_1) \ , \qquad (2)$$

where game *is* $\mathsf{EU\text{-}NMA}$ *for* $\mathsf{X} = 2$*,* $\mathsf{EU\text{-}CMA}$ *for* $\mathsf{X} = 1$ *or* $\mathsf{H\text{-}EU\text{-}CMA}^{\mathsf{H}}_{Q_1,0,0}$ *for* $\mathsf{X} = \mathsf{H}$*. If* $\mathcal{A}$ *plays the* $\mathsf{sH\text{-}EU\text{-}CMA}^{\mathsf{X}}$ *game, there exist two more adversaries* $\mathcal{B}_2$ *and* $\mathcal{B}_3$ *explicitly given in the proof such that Equation 2 holds by changing* game *to* $\mathsf{sEU\text{-}CMA}$ *if* $\mathsf{X} = 1$*, or adding* $\mathsf{Adv}^{\mathsf{CUR}}_{\mathsf{ID}_2}(\mathcal{B}_3)$ *if* $\mathsf{X} = 2$*, otherwise adding* $\mathsf{Adv}^{\mathsf{CUR}}_{\mathsf{ID}_1}(\mathcal{B}_2) + \mathsf{Adv}^{\mathsf{CUR}}_{\mathsf{ID}_2}(\mathcal{B}_3)$ *to the right-hand side.*

Proof. Consider the same sequence of hybrid games as in the proof of Lemma 3, with the difference that $\mathcal{O}_1$ is never replaced and we do not consider $\mathcal{O}^u_1$ nor $\mathcal{O}^s_1$. The bounds computed in the above proof are still valid, with $Q_1 = 0$ as Σ_1 does not use H. This reduces the $\mathsf{H\text{-}EU\text{-}CMA}^{\mathsf{X}}_{Q_1,Q_2,Q_{\mathsf{H}}}$ game to the $\mathsf{H\text{-}EU\text{-}CMA}^{\mathsf{X}}_{Q_1,0,0}$ game. If $\mathsf{X} = 1$, this is the $\mathsf{EU\text{-}CMA}$ game of Σ_1. If $\mathsf{X} = 2$, this game reduces to the $\mathsf{EU\text{-}NMA}$ one of Σ_2, by sampling $(\mathsf{vk}_1, \mathsf{sk}_2) \leftarrow \mathsf{ID}_1.\mathsf{IGen}(1^\lambda)$ and managing the Σ_1 signature oracle. □

We conclude by showing that the $\mathsf{EU\text{-}NMA}$ security of $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H]$ reduces to the special soundness of ID_1 and ID_2, which are also the base assumptions for $\mathsf{FS}[\mathsf{ID}_1, H]$ and $\mathsf{FSwA}[\mathsf{ID}_2, H]$. In the proof, we show that the identification scheme ID, that concatenates ID_1 and ID_2, has a Fiat-Shamir transform whose $\mathsf{EU\text{-}NMA}$ game is equivalent to that of $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H]$, allowing us to conclude using standard arguments, in particular Lemma 2.

Theorem 3 **($\mathsf{EU\text{-}NMA}$ Security of $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H]$).** *Let everything as in Lemma 3. Let two algorithms* $\mathsf{LossyIGen}_i$*,* $i \in \{1, 2\}$*. In the ROM, there exist three adversaries explicitly given in the proof,* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *against the lossy-2-special-soundness of* ID_1 *and* ID_2 *and* $\mathcal{B}_3$ *against the collision game of* $\mathcal{H}$*, and two distinguishers* $\mathcal{D}_1$ *and* $\mathcal{D}_2$ *against the key-indistinguishability of* ID_1 *and* ID_2 *such that for any adversary* $\mathcal{A}$ *against the* $\mathsf{EU\text{-}NMA}$ *security of* $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H]$ *making at most* Q_h *random oracle queries:*

$$\mathsf{Adv}^{\mathsf{EU\text{-}NMA}}_{\mathsf{H\text{-}FSwA}[\mathsf{ID}_1\|\mathsf{ID}_2,H]}(\mathcal{A}) \le \min_{i\in\{1,2\}} \left(\mathsf{Adv}^{\mathsf{key\text{-}ind}}_{\mathsf{ID}_i}(\mathcal{D}_i) + \frac{Q_h + 1}{2^\ell} + \sqrt{(Q_h + 1)\left(\mathsf{Adv}^{\mathsf{lossy\text{-}2\text{-}ss}}_{\mathsf{ID}_i}(\mathcal{B}_i) + (2 - i) \cdot \mathsf{Adv}^{\mathsf{col}}_{\mathcal{H}}(\mathcal{B}_3)\right)}\right).$$

Proof. We first consider the following three games.

Game_0. This is the standard $\mathsf{EU\text{-}NMA}$ game.

$\mathsf{Game}_i, i \in \{1, 2\}$. This is Game_0 except that $\mathsf{LossyIGen}_i$ replaces $\mathsf{ID}_i.\mathsf{IGen}$. $\mathcal{D}_i$ is defined as taking vk_i as input, running $\mathsf{vk}_{3-i} \leftarrow \mathsf{ID}_{3-i}.\mathsf{IGen}(1^\lambda)$ and calling $\mathcal{A}$

on $\mathsf{vk} = (\mathsf{vk}_1, \mathsf{vk}_2)$. If $\mathcal{A}$ wins, $\mathcal{D}_i$ returns 1, else it returns 0. We have

$$\forall i \in \{1,2\}, \mathsf{Adv}^{\mathsf{key\text{-}ind}}_{\mathsf{ID}_i}(\mathcal{D}_i) = \left| \mathsf{Adv}^{\mathsf{EU\text{-}NMA}}(\mathcal{A}) - \mathsf{Adv}^{\mathsf{Game}_i}(\mathcal{A}) \right| .$$

Before defining the reductions to the lossy-2-special-soundness, we define the algorithm A, that on input $\mathsf{vk} = (\mathsf{vk}_1, \mathsf{vk}_2)$ and access to the random oracle H runs $\mathcal{A}^H(\mathsf{vk})$ and records all random oracle queries made by it. When $\mathcal{A}$ outputs a forgery $(\sigma^* = (z_1^*, z_2^*, c^*), M^*)$, let $w_i^* = \mathsf{ID}_1.\mathsf{Rec}(\mathsf{vk}_i, z_i^*, c^*), i \in \{1,2\}$ as well as $\mu^* = \mathcal{H}(\mathcal{H}(\mathsf{vk}), w_1^*, M^*)$. Algorithm A checks whether $\mathcal{A}$ wins by computing $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H].\mathsf{Verify}(\mathsf{vk}, \sigma^*, M^*)$, which entails querying once more the random oracle. Algorithm A finally outputs $(i, (\sigma^*, M^*))$, where i is the rank of the first random oracle query it made for $w_2^* \| \mu^*$, if $\mathcal{A}$ wins, otherwise $(0, \varepsilon)$.

Applying Lemma 2 on A, we get an algorithm $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2)$ that on input vk may return $(1, ((z_1^*, z_2^*, c^*), M^*), ((z_1' z_2', c'), M'))$ with $c' \neq c^*$ chosen externally. Letting $w_i' = \mathsf{ID}_i.\mathsf{Rec}(\mathsf{vk}_i, z_i', c'), i \in \{1,2\}$ and $\mu' = \mathcal{H}(\mathcal{H}(\mathsf{vk}), w_1', M')$, we have $w_2^* = w_2'$ and $\mu^* = \mu'$.

Adversary $\mathcal{B}_i, i \in \{1,2\}$, playing the lossy-2-special-soudness of ID_i, on input vk_i, runs $(\mathsf{vk}_{3-i}, \mathsf{sk}_{3-i}) \leftarrow \mathsf{ID}_{3-i}.\mathsf{IGen}(1^\lambda)$ and runs $\mathcal{F}_1$ on $\mathsf{vk} = (\mathsf{vk}_1, \mathsf{vk}_2)$. When it returns $(i^*, (z_1^*, z_2^*, c^*, M^*))$, $\mathcal{B}_i$ returns (w_i^*, c^*, z_i^*) and gets $c' \neq c^*$. It then runs $\mathcal{F}_2(c', st)$. Assuming it returns $(1, (z_1^*, z_2^*, c^*, M^*), (z_1', z_2', c', M'))$, if $i = 1$ and $w_1' = w_1^*$ or if $i = 2$, $\mathcal{B}_i$ returns z_i'. Otherwise, it aborts.

Adversary $\mathcal{B}_i$ then wins if and only if it did not abort, which is equivalent to $\mathcal{F}_2$ returning something starting with 1 for $\mathcal{B}_2$. It is equivalent to $\mathcal{F}_2$ returning something starting with 1 and $w_1' = w_1^*$ for $\mathcal{B}_1$. However, if $w_1' \neq w_1^*$ we have found a collision for $\mathcal{H}$ as we have two preimages for $\mu^* = \mu'$, and we define $\mathcal{B}_3$ as running everything, and returning this collision, if found. We have the bound:

$$\mathsf{Adv}^{\mathsf{Game}_i}(\mathcal{A}) \leq \frac{Q_h + 1}{2^\ell} + \sqrt{(Q_h + 1) \cdot \left(\mathsf{Adv}^{\mathsf{lossy\text{-}2\text{-}ss}}_{\mathsf{ID}_i}(\mathcal{B}_i) + (2 - i) \cdot \mathsf{Adv}^{\mathsf{col}}_{\mathcal{H}}(\mathcal{B}_3) \right)} .$$

□

Remark 2. If Σ_1 is instead a signature scheme that does not use H, by considering $\mathsf{LossyIGen}_1 = \mathsf{ID}_1.\mathsf{IGen}$, the same proof allows to reduce the $\mathsf{H\text{-}EU\text{-}CMA}^{\mathsf{H}}_{Q_1,0,0}$ game to a "2-special-soundness for ID_1 with a Σ_1 signature oracle" game. As the signature oracle is independent from H, the forking lemma still applies.

4.4 BUFF Properties

We now turn to proving the BUFF properties of our construction.

Theorem 4 (EO Security of $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H]$). *In the standard model, there exist adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$, explicitly given in the proof of this theorem, such that for any hash functions $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\alpha$ and $H : \{0,1\}^* \to \{0,1\}^\ell$, and any adversary $\mathcal{A}$ against the EO security of $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H]$, we have*

$$\mathsf{Adv}^{\mathsf{EO}}_{\mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H]}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{Col}}_{\mathcal{H}}(\mathcal{B}_0) + \mathsf{Adv}^{\mathsf{Col}}_{H}(\mathcal{B}_1).$$

Proof. We use the notations of Fig. 5. Let $(\mathsf{vk}, \mathsf{vk}', m, m', \sigma)$ be $\mathcal{A}$'s output. If $\mathcal{A}$ wins the EO game, then σ is a valid signature for m (resp. m') under vk (resp. vk'). From line 4 of the verification algorithm in Fig. 5, it follows that $c = H(w_2\|\mu) = H(w_2'\|\mu')$ where μ and μ' are both α-bit strings. If $\mu \neq \mu'$, this directly yields a collision for H. Otherwise, by line 2, we have

$$\mathcal{H}(\mathcal{H}(\mathsf{vk}_1\|\mathsf{vk}_2)\|w_1\|m) = \mathcal{H}(\mathcal{H}(\mathsf{vk}_1'\|\mathsf{vk}_2')\|w_1'\|m'). \tag{3}$$

By the EO definition, $\mathcal{A}$ can only win if $(\mathsf{vk}_1, \mathsf{vk}_2) \neq (\mathsf{vk}_1', \mathsf{vk}_2')$. We directly obtain a collision for $\mathcal{H}$ if $\mathcal{H}(\mathsf{vk}_1\|\mathsf{vk}_2) = \mathcal{H}(\mathsf{vk}_1'\|\mathsf{vk}_2')$. Else, Eq. 3 yields a collision for $\mathcal{H}$ since both inputs have distinct α-bit prefixes. Finally, adversary $\mathcal{B}_0$ (resp. $\mathcal{B}_1$) runs $\mathcal{A}$ and outputs the corresponding collision on H (resp. $\mathcal{H}$), if any. □

Theorem 5 **(MBS Security of $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1\|\mathsf{ID}_2, H]$).** *In the standard model, there exist adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$, explicitly given in the proof of this theorem, such that for any hash functions $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\alpha$ and $H : \{0,1\}^* \to \{0,1\}^\ell$, and any adversary $\mathcal{A}$ against the MBS security of $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1\|\mathsf{ID}_2, H]$, we have*

$$\mathsf{Adv}^{\mathsf{MBS}}_{\mathsf{H\text{-}FSwA}[\mathsf{ID}_1\|\mathsf{ID}_2,H]}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{Col}}_{\mathcal{H}}(\mathcal{B}_0) + \mathsf{Adv}^{\mathsf{Col}}_{H}(\mathcal{B}_1).$$

Proof. We use the notations of Fig. 5. Let $(\mathsf{vk}, m, m', \sigma)$ be $\mathcal{A}$'s output. If $\mathcal{A}$ wins the MBS game, then σ is a valid signature for both m and m' under vk. The proof proceeds identically to that of Theorem 4 except that Eq. 3 now becomes

$$\mathcal{H}(\mathcal{H}(\mathsf{vk}_1\|\mathsf{vk}_2)\|w_1\|m) = \mathcal{H}(\mathcal{H}(\mathsf{vk}_1\|\mathsf{vk}_2)\|w_1\|m'). \tag{4}$$

Since $\mathcal{A}$ can only win if $m \neq m'$, we immediately obtain a collision for $\mathcal{H}$ if one was not already found for H. Finally, both adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$ run $\mathcal{A}$ and output their respective collisions for H or $\mathcal{H}$, if any. □

Analyzing the non-resignability of our construction requires introducing the Hide-and-Seek game from [13].

Definition 17 (Hide-and-Seek). *Let H be a random oracle. For any adversaries $\mathcal{A}$ and $\mathcal{D}$, we define* $\mathsf{Adv}^{\mathsf{HnS}}_{H}(\mathcal{A}, \mathcal{D}) := \Pr_{\substack{(x,z)\leftarrow\mathcal{D}^H \\ x^*\leftarrow\mathcal{A}^H(H(x),z)}}(x = x^*)$.

Theorem 6 **(NR Security of $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1\|\mathsf{ID}_2, H]$).** *In the ROM, where $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\alpha$ and $H : \{0,1\}^* \to \{0,1\}^\ell$ are modeled as random oracles, for any adversaries $\mathcal{A}$ and $\mathcal{D}$ against the NR security of $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1\|\mathsf{ID}_2, H]$, there exist adversaries $\bar{\mathcal{A}}$, $\bar{\mathcal{D}}$, such that*

$$\mathsf{Adv}^{\mathsf{NR}}_{\mathsf{H\text{-}FSwA}[\mathsf{ID}_1\|\mathsf{ID}_2,H],\mathsf{aux}}(\mathcal{A}, \mathcal{D}) \leq Q_A \cdot \mathsf{Adv}^{\mathsf{HnS}}_{\mathcal{H}}(\bar{\mathcal{A}}, \bar{\mathcal{D}}) + \frac{1}{2^\ell} + Q_h \cdot Q_D \cdot 2^{-\alpha},$$

where $\mathcal{A}$ and $\bar{\mathcal{A}}$ make at most Q_A queries to $\mathcal{H}$, $\mathcal{A}$ makes at most Q_h queries to H and $\mathcal{D}$ makes at most Q_D queries to $\mathcal{H}$.

```
Games Game_0 to Game_1:
1: $(\mathsf{vk}, tr, \mathsf{sk}) \leftarrow \Sigma.\mathsf{KeyGen}(1^\lambda)$
2: $m \leftarrow \mathcal{D}^{H,\mathcal{H}}(\mathsf{sk})$
3: $\sigma \leftarrow \Sigma.\mathsf{Sign}(\mathsf{sk}, m)$
4: $(z_1, z_2, c) \leftarrow \sigma$
5: $(\mathsf{vk}^*, \sigma^*) \leftarrow \mathcal{A}^{H,\bar{\mathcal{H}}}(\mathsf{sk}, \sigma, \mathsf{aux}(\mathsf{sk}, m))$
6: if $\mathsf{vk}^* = \mathsf{vk}$ then return $\perp$
7: if $\Sigma.\mathsf{Verify}(\mathsf{vk}^*, \sigma^*, m) = 1$ then
8:     return $1$
9: return $\perp$

Oracle $H(y)$:
1: return $H(y)$

Oracle $\bar{\mathcal{H}}(y)$:
1: if $y = \cdot \| m$ then   // Game_1
2:     abort                  // Game_1
3: return $\mathcal{H}(y)$
```

Fig. 8: Games Game_0 to Game_1 for the proof of Theorem 6.

Proof. We follow the first game hop of [13,19] as defined in Fig. 8.

Game Game_0 is the NR game for $\Sigma := \mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H]$. We define Game_1 as Game_0, except that $\mathcal{H}$ aborts whenever queried by $\mathcal{A}$ on an input with suffix m (the message produced by $\mathcal{D}$). Let $\bar{\mathcal{H}}$ denote this new oracle. There exist adversaries $\bar{\mathcal{A}}$, $\mathcal{B}$, and $\bar{\mathcal{D}}$, described in Fig. 9, against the HnS property of $\mathcal{H}$, such that

$$\begin{cases} \underset{\substack{(\mathsf{vk},tr,\mathsf{sk})\leftarrow\Sigma.\mathsf{KeyGen} \\ m\leftarrow\mathcal{D}^{H,\mathcal{H}}(\mathsf{sk})}}{H_\infty}(m \mid \mathcal{H}, \mathsf{sk}, \mathsf{aux}(\mathsf{sk}, m)) \leq \underset{(x,z)\leftarrow\bar{\mathcal{D}}^{H,\mathcal{H}}}{H_\infty}(x \mid \mathcal{H}, z) \\ |\Pr(1 \leftarrow \mathsf{Game}_0^{\mathcal{A}}) - \Pr(1 \leftarrow \mathsf{Game}_1^{\mathcal{A}})| \quad \leq Q_A \cdot \mathsf{Adv}_{\mathcal{H}}^{\mathsf{HnS}}(\bar{\mathcal{A}}, \bar{\mathcal{D}}) \end{cases}$$

Specifically, $\bar{\mathcal{D}}$ returns $x = tr\|w_1\|m$ and $z = (\mathsf{sk}, \mathsf{vk}, \mathsf{st}_1, w_1, \mathsf{aux}(\mathsf{sk}, m))$.
On input $\nu = \mathcal{H}(x) = \mathcal{H}(tr\|w_1\|m)$ and z, $\bar{\mathcal{A}}$ samples a random index $i \in [1, Q_A]$, runs $\mathcal{B}^H(\nu, z)$ internally, inspects its i-th query to obtain $(tr^*\|w_1^*\|m_i^*)$, and outputs $(tr\|w_1\|m_i^*)$. Moreover,

$$\begin{aligned} \underset{(x,z)\leftarrow\bar{\mathcal{D}}^{\mathcal{H}}}{H_\infty}(x \mid \mathcal{H}, z) &= \underset{\substack{(\mathsf{vk},tr,\mathsf{sk}_1\|\mathsf{sk}_2)\leftarrow\Sigma.\mathsf{KeyGen} \\ m\leftarrow\mathcal{D}^{H,\mathcal{H}}(\mathsf{sk}) \\ (w_1,\mathsf{st}_1)\leftarrow\mathsf{ID}_1.\mathsf{P1}(\mathsf{sk}_1)}}{H_\infty}(\mathsf{vk}, m, w_1 \mid \mathcal{H}, \mathsf{sk}, \mathsf{st}_1, w_1, \mathsf{aux}(\mathsf{sk}, m)) \\ &\geq \underset{[\ldots]}{H_\infty}(m \mid \mathcal{H}, \mathsf{sk}, \mathsf{st}_1, w_1, \mathsf{aux}(\mathsf{sk}, m)) \\ &\geq \underset{\substack{(\mathsf{vk},tr,\mathsf{sk})\leftarrow\Sigma.\mathsf{KeyGen} \\ m\leftarrow\mathcal{D}^{H,\mathcal{H}}(\mathsf{sk})}}{H_\infty}(m \mid \mathcal{H}, \mathsf{sk}, \mathsf{aux}(\mathsf{sk}, m)) \end{aligned}$$

where the first inequality holds because min-entropy can only decrease when fewer random variables are considered, and the second follows from independence of m and (st_1, w_1). By construction, both $\bar{\mathcal{D}}$ and $\bar{\mathcal{A}}$ preserve the efficiency of $\mathcal{D}$ and $\mathcal{A}$. Furthermore, $\bar{\mathcal{A}}$ makes at most Q_A queries to $\mathcal{H}$. Since $\mathcal{A}$ makes only classical queries, the two games differ only if $\mathcal{B}$ queries an input on which $\mathcal{H}$ and $\bar{\mathcal{H}}$ disagree. Hence,

$$\begin{aligned} \left|\Pr(1 \leftarrow \mathsf{Game}_0^{\mathcal{A}}) - \Pr(1 \leftarrow \mathsf{Game}_1^{\mathcal{A}})\right| &\leq \Pr(\exists i \in [1, Q_A] : m_i^* = m) \\ &\leq Q_A \cdot \mathsf{Adv}_{\mathcal{H}}^{\mathsf{HnS}}(\bar{\mathcal{A}}, \bar{\mathcal{D}}) \end{aligned}$$

```
$\bar{\mathcal{D}}^{H,\mathcal{H}}$:
  1: $(\mathsf{vk}, tr, \mathsf{sk}) \leftarrow \Sigma.\mathsf{KeyGen}(1^\lambda)$
  2: $m \leftarrow \mathcal{D}^{H,\mathcal{H}}(\mathsf{sk})$
  3: $\mathsf{sk}_1 \| \mathsf{sk}_2 \leftarrow \mathsf{sk}$
  4: $(w_1, st_1) \leftarrow \mathsf{ID}_1.\mathsf{P}_1(\mathsf{sk}_1)$
  5: $x \leftarrow tr \| w_1 \| m$
  6: $z \leftarrow (\mathsf{sk}, \mathsf{vk}, \mathsf{st}_1, w_1, \mathsf{aux}(\mathsf{sk}, m))$
  7: return $(x, z)$
```

```
$\mathcal{B}^H(\nu, z)$:
  1: $(\mathsf{sk}, \mathsf{vk}, \mathsf{st}_1, w_1, h) \leftarrow z$
  2: $\mathsf{sk}_1 \| \mathsf{sk}_2, \mathsf{vk}_1 \| \mathsf{vk}_2 \leftarrow \mathsf{sk}, \mathsf{vk}$
  3: $(z_2, c) \leftarrow \mathsf{FSwA}[\mathsf{ID}_2, H].\mathsf{Sign}(\mathsf{sk}_2, \nu)$
  4: $z_1 \leftarrow \mathsf{ID}_1.\mathsf{P}_2(\mathsf{sk}_1, c, st_1)$
  5: $\sigma \leftarrow (z_1, z_2, c)$
  6: $(\mathsf{vk}^*, \sigma^*) \leftarrow \mathcal{A}^{H,\bar{\mathcal{H}}}(\mathsf{sk}, \sigma, h)$
  7: return $\bot$
```

Fig. 9: Reductions $\mathcal{B}$ and $\bar{\mathcal{D}}$ against the Hide-and-Seek property of $\mathcal{H}$.

In Game_1, we have the following inequality:

$$\begin{aligned}
\Pr(\mathcal{A} \text{ wins}) &= \Pr\begin{pmatrix} \mathcal{A} \text{ wins} \wedge \\ \mathcal{A} \text{ queried } H \text{ on } w_2^* \| \mu^* \end{pmatrix} + \Pr\begin{pmatrix} \mathcal{A} \text{ wins } \wedge \mathcal{A} \text{ did not} \\ \text{query } H \text{ on } w_2^* \| \mu^* \end{pmatrix} \\
&\leq \Pr(\mathcal{A} \text{ queried } H \text{ on } w_2^* \| \mu^*) + \frac{1}{2^\ell} \\
&\leq Q_h \cdot 2^{-H_\infty(\mu^* | \mathsf{view}(\mathcal{A}))} + \frac{1}{2^\ell} \ ,
\end{aligned}$$

as $H(w_2^* \| \mu^*) \hookleftarrow (\{0,1\}^\ell)$ and where $\mathsf{view}(\mathcal{A}) = (H, \bar{\mathcal{H}}, \mathsf{sk}, \sigma, \mathsf{aux}(\mathsf{sk}, m), \mathsf{vk}^*, \sigma^*)$. Conditioned on $\mathcal{D}$ not querying $\mathcal{H}$ on $\mathcal{H}(\mathsf{vk}^*) \| w_2^* \| m$, the value μ^* is independent from $\mathsf{view}(\mathcal{A})$ and its min-entropy is α. Conditioned on $\mathcal{D}$ querying $\mathcal{H}$ on $\mathcal{H}(\mathsf{vk}^*) \| w_2^* \| m$, we apply Lemma 1, where the X_i are the outputs of the query of $\mathcal{D}$ to $\mathcal{H}$ and f runs the first five lines of Game_1, using X_i as the answer to $\mathcal{D}$'s i-th query to $\mathcal{H}$. It returns the index i such that the i-th query of $\mathcal{D}$ to $\mathcal{H}$ was on $\mathsf{vk}^* \| w_2^* \| m$. Lemma 1 gives the min-entropy on μ^* even with knowledge of all random coins of the procedure, which includes $\mathsf{view}(\mathcal{A})$. □

Remark 3. While adapting this proof in the QROM is out of scope of this work, [13] already explains how to adapt the first game hop.

5 Silithium: Instantiation with EC-Schnorr and ML-DSA

We now propose our main instantiation of the $\mathsf{H\text{-}FSwA}[\mathsf{ID}_1 \| \mathsf{ID}_2, H]$ construction: the Silithium signature scheme. We use $\mathsf{ID}_1 = \mathsf{ID}_{\mathsf{Schnorr}}$ as defined in Figure 11, in Appendix A and $\mathsf{ID}_2 = \mathsf{ID}_{\mathsf{ML\text{-}DSA}}$ as defined in Figure 14, in Appendix B.

5.1 Specification

In our specification, we consider a cyclic subgroup $\mathbb{G}$ of an elliptic curve with generator point G of order n. We assume access to a function $\mathsf{ECCLib.RandomPoint}$ that outputs a pair (k, R), with an integer $0 < k < n$ and curve point R such that $k.G = R$ and $k \hookleftarrow U(\mathbb{Z}_n)$. We also assume access to an ML-DSA implementation supporting the "external μ" variant described in FIPS 204 [27]. Finally, following the specification of ML-DSA, $\mathcal{H}$ is instantiated as $\mathsf{SHAKE256}(\cdot, 512)$ and H as $\mathsf{SHAKE256}(\cdot, 2\lambda)$, and we also require an implementation of SHAKE256, where λ is the security parameter. The resulting scheme is described in Figure 10.

These aforementioned ingredients are enough to implement Silithium: they allow to generate key pairs, and to update the hash of the verification key contained in the signing key. During signing, the Schnorr commitment R is embedded into the μ computation and then ML-DSA.Sign_mu is used as a black-box. The verification procedure is performed in a similar way.

Note that it is also possible to embed R in the ML-DSA context string during the signature, thus removing the need to compute μ externally. However, this cannot be applied to verification due to our modification of tr.

On simultaneous verification. Verifying the $\mathsf{ID}_{\mathsf{ML\text{-}DSA}}$ part of the signature implies first recovering the $\mathsf{ID}_{\mathsf{Schnorr}}$ commitment. As the hash verification is common to the two schemes, this implies verifying the $\mathsf{ID}_{\mathsf{Schnorr}}$ component. Moreover, to verify the $\mathsf{ID}_{\mathsf{Schnorr}}$ part, one must first recover the $\mathsf{ID}_{\mathsf{ML\text{-}DSA}}$ commitment. One could however ignore the norm bound check for $\mathsf{ID}_{\mathsf{ML\text{-}DSA}}$. As this implies rewriting an incomplete implementation of ML-DSA.Verify, we believe that a lazy developer would not choose to go down this path.

```
KeyGen():
1: $(\mathsf{vk}_1, \mathsf{sk}_1) \leftarrow \mathsf{ECCLib.RandomPoint}()$
2: $(\mathsf{vk}_2, \mathsf{sk}_2) \leftarrow \mathsf{ML\text{-}DSA.KeyGen}()$
3: $(\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0) \leftarrow \mathsf{sk}_2$
4: $tr \leftarrow \mathcal{H}(\mathsf{vk}_1 \| \mathsf{vk}_2)$
5: $\mathsf{sk}_2 \leftarrow (\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0)$
6: return $((\mathsf{vk}_1, \mathsf{vk}_2), (\mathsf{sk}_1, \mathsf{sk}_2))$

Verify$((\mathsf{vk}_1, \mathsf{vk}_2), (\tilde{c}, \mathbf{z}, \mathbf{h}, x), M)$:
1: if $x < 0$ or $x \geq n$ then
2:     return $0$
3: $R \leftarrow x.G - \tilde{c}.\mathsf{vk}_1$
4: $\mu \leftarrow \mathcal{H}(\mathcal{H}(\mathsf{vk}_1 \| \mathsf{vk}_2) \| R \| M)$
5: $b \leftarrow \mathsf{ML\text{-}DSA.Verify\_mu}(\mathsf{vk}_2, \mu, (\mathbf{z}, \tilde{c}, \mathbf{h}))$
6: return $b$

Sign$((\mathsf{sk}_1, \mathsf{sk}_2), M)$: (external $\mu$)
1: $(r, R) \leftarrow \mathsf{ECCLib.RandomPoint}()$
2: $\mu \leftarrow \mathcal{H}(tr \| R \| M)$
3: $(\mathbf{z}, \tilde{c}, \mathbf{h}) \leftarrow \mathsf{ML\text{-}DSA.Sign\_mu}(\mathsf{sk}_2, \mu)$
4: $x \leftarrow r + \mathsf{sk}_1 \cdot \tilde{c} \bmod n$
5: return $(\tilde{c}, \mathbf{z}, \mathbf{h}, x)$

Sign$((\mathsf{sk}_1, \mathsf{sk}_2), M)$: (context)
1: $(r, R) \leftarrow \mathsf{ECCLib.RandomPoint}()$
2: $(\mathbf{z}, \tilde{c}, \mathbf{h}) \leftarrow \mathsf{ML\text{-}DSA.Sign}(\mathsf{sk}_2, M, R)$
3: $x \leftarrow r + \mathsf{sk}_1 \cdot \tilde{c} \bmod n$
4: return $(\tilde{c}, \mathbf{z}, \mathbf{h}, x)$
```

Fig. 10: Silithium specification. Key generation updates the hash of vk_2 in sk_2.

5.2 Theoretical Analysis

Section 4, when instantiated with $\mathsf{ID}_1 = \mathsf{ID}_{\mathsf{Schnorr}}$ and $\mathsf{ID}_2 = \mathsf{ID}_{\mathsf{ML\text{-}DSA}}$, gives the following results, that see Silithium as a hybrid between $\mathsf{BUFF}(\mathsf{FS}[\mathsf{ID}_{\mathsf{Schnorr}}, H])$ and ML-DSA.

Corollary 2 (EC-Schnorr+ML-DSA **hybrid**). *Let* $\Sigma_1 = \mathsf{BUFF}(\mathsf{FS}[\mathsf{ID}_{\mathsf{Schnorr}}, H])$ *and* $\Sigma_2 = \mathsf{BUFF}(\mathsf{FSwA}[\mathsf{ID}_{\mathsf{ML\text{-}DSA}}, H]) = \mathsf{ML\text{-}DSA}$. *The scheme* $\Sigma_{\mathsf{H}} = \mathsf{Silithium}$ *defined in Figure 10 is correct and:*

- H-EU-CMA$^{\mathsf{H}}$ *(resp.* sH-EU-CMA$^{\mathsf{H}}$*) secure in the QROM as long as* ML-DSA *is* EU-CMA *(resp.* sEU-CMA*) secure in the QROM,*
- sH-EU-CMA^1 *secure in the ROM under the* DL *assumption in* $\mathbb{G}$,
- H-EU-CMA^2 *(resp.* sH-EU-CMA^2*) secure in the ROM under both (resp. all three)* $\mathsf{MLWE}_{n,q,k,\ell,U(S_\eta)}$ *and* $\mathsf{MSIS}_{n,q,k,\ell+1,2\zeta}$ *(resp. and* $\mathsf{MSIS}_{n,q,k,\ell,2\zeta'}$*) assumptions,*
- H-EU-CMA$^{\mathsf{H}}$ *secure in the ROM under the* DL *assumption in* $\mathbb{G}$ *or both the* $\mathsf{MLWE}_{n,q,k,\ell,U(S_\eta)}$ *and* $\mathsf{MSIS}_{n,q,k,\ell+1,2\zeta}$ *assumptions,*
- sH-EU-CMA$^{\mathsf{H}}$ *secure under both the same assumptions as its* H-EU-CMA$^{\mathsf{H}}$ *security and the* $\mathsf{MSIS}_{n,q,k,\ell,2\zeta'}$ *assumption,*

where ζ *and* ζ' *are defined in Lemma 7 and Lemma 6, respectively.*

The above result seems artificial on the classical signature side. Indeed, while EC-Schnorr is standard [17], it does not use SHAKE256 as its hash function. We are interested in the case where Σ_1 is chosen as ECDSA [25], as it is likely to be the classical signature used in tandem with Silithium and ML-DSA.

Theorem 7 (ECDSA+ML-DSA **hybrid**). *Let* $\Sigma_1 = \mathsf{ECDSA}$ *and* $\Sigma_2 = \mathsf{ML\text{-}DSA}$. *The signature scheme* $\Sigma_{\mathsf{H}} = \mathsf{Silithium}$ *defined in Figure 10 is correct and:*

- H-EU-CMA$^{\mathsf{H}}$ *(resp.* sH-EU-CMA$^{\mathsf{H}}$*) secure in the QROM as long as* ML-DSA *is* EU-CMA *(resp.* sEU-CMA*) secure in the QROM,*
- (s)H-EU-CMA^1 *secure in the ROM as long as* ECDSA *is* EU-CMA *secure,*
- H-EU-CMA^2 *(resp.* sH-EU-CMA^2*) secure in the ROM under both (resp. all three)* $\mathsf{MLWE}_{n,q,k,\ell,U(S_\eta)}$ *and* $\mathsf{MSIS}_{n,q,k,\ell+1,2\zeta}$ *(resp. and* $\mathsf{MSIS}_{n,q,k,\ell,2\zeta'}$*) assumptions,*
- H-EU-CMA$^{\mathsf{H}}$ *secure in the ROM under the hardness of* ECDSA *key-recovery or both the* $\mathsf{MLWE}_{n,q,k,\ell,U(S_\eta)}$ *and* $\mathsf{MSIS}_{n,q,k,\ell+1,2\zeta}$ *assumptions,*
- sH-EU-CMA$^{\mathsf{H}}$ *secure under both the same assumptions as its* H-EU-CMA$^{\mathsf{H}}$ *security and the* $\mathsf{MSIS}_{n,q,k,\ell,2\zeta'}$ *assumption,*

where ζ *and* ζ' *are defined in Lemma 7 and Lemma 6, respectively.*

Proof. The correctness follows from Theorem 1. The first point is a direct application of Theorem 2. The rest is an application of Corollary 1 and Remark 2. Note that breaking the "2-special-soundness for ID_1 with a Σ_1 signature oracle" implies recovering the signing key used by ECDSA.

5.3 Implementation Considerations

One of our claims is the relative simplicity of implementing Silithium, providing that the following requirements are met:

- availability of an elliptic curve library;

- availability of an ML-DSA implementation supporting the "external μ" feature described in FIPS 204 [27];
- familiarity with elliptic curve operations.

Note that familiarity with ML-DSA is not required, as the ML-DSA functions are used in a black-box approach. Thus, our scheme is intended to be easy to implement for software engineers with prior experience in implementing classical (elliptic curve) cryptography.

Implementation approach. To meet the above requirements, we implemented Silithium within (a fork of) OpenSSL's default provider. The OpenSSL library already includes both an ML-DSA implementation supporting "external μ" computation and built-in elliptic curve operations. Silithium was implemented directly inside OpenSSL codebase, as access to internal elliptic curve functions was required. While it could have been implemented as a third party provider, embedding it within the default provider significantly sped up development. We provide three Silithium instantiations, described in Table 3.

Variant	ML-DSA	Curve	Signature Size
Silithium-44	ML-DSA-44	P-256	2420 + 32
Silithium-65	ML-DSA-65	P-384	3309 + 48
Silithium-87	ML-DSA-87	P-521	4627 + 66

Table 3: Silithium instantiations.

Implementation effort. The implementation was carried out by a developer familiar with OpenSSL internals and its elliptic curve API. The majority of the effort was divided between implementing elliptic curve operations and integrating the resulting code into OpenSSL. The Silithium implementation itself boils down to less than 500 lines of codes (excluding OpenSSL-related logic). The total workload to produce a functional proof-of-concept amounted to roughly one week. We emphasize that, although functionally correct, the provided code is not production-ready and should not be used as-is. The workload estimate is given only as an indication of the expected development cost of Silithium for an average cryptography software engineer.

Performances. When comparing a single execution of Silithium to the successive execution of ECDSA (or EC-S-DSA) and ML-DSA, it can be observed that Silithium performs one fewer call to the underlying hash function, which is SHA-256 in our instanciation. This optimization yields only a marginal performance gain for short messages, as the cost of SHA-256 (especially when accelerated via the sha-ni instruction set) is negligible compared to that of ML-DSA. However, the benefit becomes more noticeable when signing larger messages. Our experiments confirmed that Silithium is as fast as the hybrid concatenation scheme, although we do not report detailed benchmark results here, as precise hash benchmarking is out of scope of this work. Since our implementation is fully integrated within OpenSSL, one can easily measure performance under various conditions. Compared to the BoP-2 construction from [19], Silithium saves two hash computations, and is therefore theoretically faster—though the lack of published code does not allow for direct experimental comparisons. From a code-size perspective, Silithium employs a single hash function and is thus more compact than the hybrid concatenation scheme.

A note on masking. Although a masked implementation is beyond the scope of this work, we note that masking Silithium is neither harder nor easier than masking its components, namely ML-DSA and the relevant elliptic curve operations. In other words, if a developer has access to a masked implementation of ML-DSA and a masked elliptic curve library, then masking Silithium is straightforward. In particular, we emphasize that the computation of μ relies solely on public data.

References

1. ANSSI. ANSSI views on the post-quantum cryptography transition (2023 follow up).
2. Christoph Bader, Dennis Hofheinz, Tibor Jager, Eike Kiltz, and Yong Li. Tightly-secure authenticated key exchange. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part I*, volume 9014 of *LNCS*, pages 629–658. Springer, Berlin, Heidelberg, March 2015.
3. Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 390–399. ACM Press, October / November 2006.
4. Florian Bergsma, Benjamin Dowling, Florian Kohlar, Jörg Schwenk, and Douglas Stebila. Multi-ciphersuite security of the Secure Shell (SSH) protocol. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *ACM CCS 2014*, pages 369–381. ACM Press, November 2014.
5. Nina Bindel and Britta Hale. A note on hybrid signature schemes. Cryptology ePrint Archive, Report 2023/423, 2023.
6. Nina Bindel, Britta Hale, Deirdre Connolly, and Flo D. Hybrid signature spectrums. Internet-Draft draft-ietf-pquip-hybrid-signature-spectrums-07, IETF, 2025.
7. Nina Bindel, Udyani Herath, Matthew McKague, and Douglas Stebila. Transitioning to a quantum-resistant public key infrastructure. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017*, pages 384–405. Springer, Cham, 2017.
8. Jacqueline Brendel, Cas Cremers, Dennis Jackson, and Mang Zhao. The provable security of Ed25519: Theory and practice. In *2021 IEEE Symposium on Security and Privacy*, pages 1659–1676. IEEE Computer Society Press, May 2021.
9. BSI. Cryptographic mechanisms: Recommendations and key lengths - BSI tr-02102-1, 2024.
10. Jung Hee Cheon, Hyeongmin Choe, Julien Devevey, Tim Güneysu, Dongyeon Hong, Markus Krausz, Georg Land, Marc Möller, Damien Stehlé, and MinJune Yi. HAETAE: Shorter lattice-based fiat-shamir signatures. *IACR TCHES*, 2024(3):25–75, 2024.

11. Cas Cremers, Samed Düzlü, Rune Fiedler, Marc Fischlin, and Christian Janson. BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures. In *2021 IEEE Symposium on Security and Privacy*, pages 1696–1714. IEEE Computer Society Press, May 2021.
12. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.
13. Jelle Don, Serge Fehr, Yu-Hsuan Huang, Jyun-Jie Liao, and Patrick Struck. Hide-and-seek and the non-resignability of the BUFF transform. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024, Part III*, volume 15366 of *LNCS*, pages 347–370. Springer, Cham, December 2024.
14. Jelle Don, Serge Fehr, Yu-Hsuan Huang, and Patrick Struck. On the (in)security of the BUFF transform. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part I*, volume 14920 of *LNCS*, pages 246–275. Springer, Cham, August 2024.
15. Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU. Submission to the NIST Post-Quantum Cryptography Standardization Project, 2020.
16. Diana Ghinea, Fabian Kaczmarczyck, Jennifer Pullman, Julien Cretin, Stefan Kölbl, Rafael Misoczki, Jean-Michel Picod, Luca Invernizzi, and Elie Bursztein. Hybrid post-quantum signatures in hardware security keys. In *Applied Cryptography and Network Security Workshops: ACNS 2023 Satellite Workshops, ADSC, AIBlock, AIHWS, AIoTS, CIMSS, Cloud S&P, SCI, SecMT, SiMLA, Kyoto, Japan, June 19–22, 2023, Proceedings*, page 480–499. Springer-Verlag, 2023.
17. International Organization for Standardization. It security techniques — digital signatures with appendix — part 3: Discrete logarithm based mechanisms, November 2018. Standard number 76382 on ISO website.
18. Tibor Jager, Eike Kiltz, Doreen Riepel, and Sven Schäge. Tightly-secure authenticated key exchange, revisited. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 117–146. Springer, Cham, October 2021.
19. Jonas Janneck. Bird of prey: Practical signature combiners preserving strong unforgeability. Cryptology ePrint Archive, Paper 2025/1844, 2025.
20. Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 552–586. Springer, Cham, April / May 2018.
21. Evan Klitzke. Bitcoin transaction malleability, 2017.
22. D. Knuth. *The Art of Computer Programming.* Addison-Wesley, 1997.
23. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *DCC*, 75(3):565–599, 2015.
24. National Institute of Standards and Technology. Post-Quantum Cryptography: Additional Digital Signature Schemes — Round 2. `https://csrc.nist.gov/projects/pqc-dig-sig/round-2-additional-signatures`, 2022.
25. National Institute of Standards and Technology. Digital signature standard (dss): Federal information processing standards publication 186-5. Technical Report FIPS 186-5, U.S. Department of Commerce, National Institute of Standards and Technology, Gaithersburg, MD, USA, February 2023.
26. NLNCSA. Bereid je voor op de dreiging van quantumcomputers, 2021.

27. National Institute of Standards and Technology. Module-lattice-based digital signature standard. Technical Report Federal Information Processing Standards Publications (FIPS) 204, 2024, U.S. Department of Commerce, Washington, D.C., 2024.
28. Mike Ounsworth, John Gray, Massimiliano Pala, Jan Klaußner, and Scott Fluhrer. Composite ML-DSA for use in X.509 Public Key Infrastructure. Internet-Draft draft-ietf-lamps-pq-composite-sigs-12, IETF, 2025.
29. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.
30. Phillip Rogaway. Formalizing human ignorance. In Phong Q. Nguyen, editor, *Progress in Cryptology - VIETCRYPT 06*, volume 4341 of *LNCS*, pages 211–228. Springer, Berlin, Heidelberg, September 2006.
31. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, New York, August 1990.

A Schnorr Identification Scheme with tweaked Challenge

Let $\mathbb{G}$ be a cyclic group with generator g and prime order p. We describe a modification of Schnorr's identification scheme, where the challenge is a bit-string $c \in \{0,1\}^\ell$. It is then up to the prover and the verifier to interpret it as an integer in $[0, 2^\ell - 1]$ and reduce it mod p to get the "real" challenge.

```
IGen(1^λ):
  1: $\mathsf{sk} \hookleftarrow U(\mathbb{Z}_p)$
  2: $\mathsf{vk} \leftarrow g^{\mathsf{sk}}$
  3: return $(\mathsf{vk}, \mathsf{sk})$

Rec(vk, z, c):
  1: $c' \leftarrow \sum_{i=0}^{\ell-1} c_i 2^i \bmod p$
  2: return $g^z \mathsf{vk}^{-c'}$

P_1(sk):
  1: $y \hookleftarrow U(\mathbb{Z}_p)$
  2: $w \leftarrow g^y$
  3: return $(w, st = y)$

P_2(sk, w, c, st):
  1: $c' \leftarrow \sum_{i=0}^{\ell-1} c_i 2^i \bmod p$
  2: $z = y + c'\mathsf{sk}$
  3: return $z$

V(vk, w, c, z):
  1: $c' \leftarrow \sum_{i=0}^{\ell-1} c_i 2^i \bmod p$
  2: if $w \neq g^z \mathsf{vk}^{-c'}$ then
  3:     return 0
  4: return 1
```

Fig. 11: Schnorr identification scheme with ℓ bits challenge $\mathsf{ID}_{\mathsf{Schnorr}}$.

Lemma 4 (Adapted from [31]). *Let ℓ be an integer such that $2^\ell > p$. The scheme from Figure 11 is complete, commitment-recoverable, HVZK, has unique response and for any adversary $\mathcal{A}$ against its 2-special-soundness, there exists an adversary $\mathcal{B}$ against the discrete logarithm in $\mathbb{G}$ such that:*

$$\mathsf{Adv}^{2-\mathsf{ss}}_{\mathsf{ID}_{\mathsf{Schnorr}}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{DL}}_{\mathbb{G},g}(\mathcal{B}) + \frac{1}{p}.$$

Proof. The completeness, commitment-recoverability and HVZK proofs are identical to the original description of the Schnorr identification scheme. Concerning the 2-special-soundness property, we recall that given (w, c_1, c_2, z_1, z_2) such that $g^{z_1}\mathsf{vk}^{-c_1'} = w = g^{z_2}\mathsf{vk}^{-c_2'}$, the discrete log of vk is $(z_1 - z_2)\cdot(c_1' - c_2')^{-1} \bmod p$ as long as $(c_1' - c_2')$ is nonzero. For any $c_1 \in [0, 2^\ell - 1]$ and $c_2 \hookleftarrow U([0, 2^\ell - 1]$, the probability that $c_1 = c_2 \bmod p$ while $c_1 \neq c_2$ is such that:

$$\begin{aligned}\Pr_{c_1 \neq c_2}(c_1 = c_2 \bmod p) &= \Pr(c_1 = c_2 \bmod p) - \Pr(c_1 = c_2) \\ &\leq \frac{\lfloor (2^\ell - 1)/p \rfloor + 1}{2^\ell} - \frac{1}{2^\ell} \leq \frac{1}{p} \ ,\end{aligned}$$

as at most $\lfloor (2^\ell - 1)/p \rfloor + 1$ values in $[0, 2^\ell - 1]$ are sent to k when reduced mod p.

Finally, given four elements (w, c, z, z') such that $g^z\mathsf{vk}^{-c} = w = g^{z'}\mathsf{vk}^{-c}$, it implies $g^z = g^{z'}$ thus $z = z' \bmod p$, showing the scheme has unique response. □

B ML-DSA Identification Scheme with tweaked Challenge

We recall supporting algorithms for ML-DSA in Fig. 12. These algorithms are extended on vectors by applying them coefficient-wise.

```
Power2Round_q(r, d):
  1: $r = r \bmod {}^{+}q$
  2: $r_0 = r \bmod {}^{\pm}2^d$
  3: return $((r - r_0)/2^d, 2)$

MakeHint_q(z, r, α):
  1: $r_1 \leftarrow \mathsf{HighBits}_q(r, \alpha)$
  2: $v_1 \leftarrow \mathsf{HighBits}(r + z, \alpha)$
  3: return $|r_1 - v_1|$

Decompose_q(r, α):
  1: $r \leftarrow r \bmod {}^{+}q$
  2: $r_0 \leftarrow r \bmod {}^{\pm}\alpha$
  3: if $r - r_0 = q - 1$ then
  4:     $(r_1, r_0) \leftarrow (0, r_0 - 1)$
  5: else
  6:     $r_1 \leftarrow (r - r_0)/\alpha$
  7: return $(r_1, r_0)$

UseHint_q(h, r, α):
  1: $m \leftarrow (q - 1)/\alpha$
  2: $(r_1, r_0) \leftarrow \mathsf{Decompose}_q(r, \alpha)$
  3: return $r_1 + h \cdot \mathsf{sgn}(r_0) \mod {}^{+}m$

HighBits_q(r, α):
  1: $(r_1, r_0) \leftarrow \mathsf{Decompose}_q(r, \alpha)$
  2: return $r_1$

LowBits_q(r, α):
  1: $(r_1, r_0) \leftarrow \mathsf{Decompose}_q(r, \alpha)$
  2: return $r_0$
```

Fig. 12: Supporting Algorithms for ML-DSA.

The inside-out Fisher-Yates shuffling algorithm is used in ML-DSA to sample the challenge polynomial. We recall it in Fig. 13.

```
SampleInBall($\tau, (j_i, s_i)_{i=n-\tau}^{n-1}$):
  1: $\mathbf{c} \leftarrow 0^n$
  2: for $i = n - \tau$ to $n - 1$ do
  3:     $c_i \leftarrow c_{j_i}$
  4:     $c_{j_i} \leftarrow (-1)^{s_i}$
  5: return $\sum_{i=0}^{n-1} c_i x^i$
```

Inputs: $\tau \leq n$, $s_i \in \{0, 1\}, j_i \leq i, \forall i \leq n - 1$
Outputs: ternary polynomial c with τ nonzero elements.

Fig. 13: SampleInBall description.

Lemma 5 (Adapted from [22]). *Let $0 < \tau \leq n$ be two integers. For all integers $i \in [n-\tau, n-1]$, let $s_i \hookleftarrow U(\{0,1\})$ and $j_i \hookleftarrow U(\{0,1,\ldots,i\})$, then the output of* $\mathsf{SampleInBall}(\tau, (j_i, s_i)_{i=n-\tau}^{n-1}$ *follows $U(\{x \in \mathcal{R} | \|x\|_\infty = 1 \wedge \|x\|_1 = \tau\})$.*

We propose the following tweak to the ML-DSA identification scheme, as described in [20, Section 4]. Instead of using a challenge c that is a ternary polynomial with fixed Hamming weight, we use a bitstring of length ℓ, the one that is used to derive the inputs of SampleInBall in the ML-DSA signature scheme. Turning the bitstring into a polynomial is left to both P_2 and V.

```
IGen(1^λ):
 1: $\mathbf{A} \hookleftarrow U(\mathcal{R}_q^{k\times\ell})$
 2: $\mathbf{s}_1, \mathbf{s}_2 \hookleftarrow S_\eta^k \times S_\eta^\ell$
 3: $\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 \bmod q$
 4: $(\mathbf{t}_1, \mathbf{t}_0) = \mathsf{Power2Round}_q(\mathbf{t}, d)$
 5: return $\mathsf{vk} = (\mathbf{A}, \mathbf{t}_1), \mathsf{sk} = (\mathbf{A}, \mathbf{s}_1, \mathbf{s}_2)$

V(vk, w_1, c, z, h):
 1: $c' \leftarrow \mathsf{SampleInBall}(\tau, H(c))$
 2: if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ then
 3:     return 0
 4: if $\mathbf{w}_1 \neq \mathsf{UseHint}(\mathbf{h}, \mathbf{A}\mathbf{z} - c'\mathbf{t}_1 \cdot 2^d, 2\gamma_2)$
    then
 5:     return 0
 6: return 1

P_1(sk):
 1: $\mathbf{y} \hookleftarrow \tilde{S}_{\gamma_1}^\ell$
 2: $\mathbf{w} = \mathbf{A}\mathbf{y} \bmod q$
 3: $\mathbf{w}_1 = \mathsf{HighBits}(\mathbf{w}, 2\gamma_2)$
 4: return $w = \mathbf{w}_1, st = \mathbf{y}$

P_2(vk, c, y):
 1: $c' \leftarrow \mathsf{SampleInBall}(\tau, H(c))$
 2: $\mathbf{z} \leftarrow \mathbf{y} + c\mathbf{s}_1$
 3: $\mathbf{r}_0 \leftarrow \mathsf{LowBits}(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2)$
 4: if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ or $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$
    then
 5:     return $\perp$
 6: $\mathbf{h} \leftarrow \mathsf{MakeHint}(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$
 7: if $\|c\mathbf{t}_0\|_\infty \geq \gamma_2$ or $\|\mathbf{h}\|_1 \geq \omega$ then
 8:     return $\perp$
 9: return $(\mathbf{z}, \mathbf{h})$
```

Fig. 14: ML-DSA identification scheme with $\{0,1\}^\ell$ challenge space $\mathsf{ID}_{\mathsf{ML\text{-}DSA}}$.

This change does not modify the following properties of the scheme.

Lemma 6 (Adapted from [20, Section 4.3]). *The scheme from Figure 14 is complete, HVZK and commitment recoverable. Let $\zeta' = \max(2(\gamma_1 - \beta), 4\gamma_2 + 2)$. For any adversary playing the key-indistinguishability game, there exists an adversary against the $\mathsf{MLWE}_{n,q,k,\ell,U(S_\eta)}$ problem with the same advantage, with respect to the* LossyIGen *algorithm that samples $(\mathbf{A}, \mathbf{t}) \hookleftarrow U(\mathcal{R}_q^{k\times\ell} \times \mathcal{R}_q^k)$ and outputs $\mathsf{vk} = (\mathbf{A}, \mathbf{t}_1)$ with $(\mathbf{t}_1, \mathbf{t}_0) = \mathsf{Power2Round}(\mathbf{t}, d)$.*
For any adversary playing the CUR *game, there exists an adversary against the $\mathsf{MSIS}_{n,q,k,\ell+1,\zeta'}$ problem with the same advantage.*

However, the change in the challenge space means that we have to adapt the proof of the folklore lossy-2-special-soundness property.

Lemma 7 (Lossy-2-Special-Soundness). *Let $\mathsf{ID}_{\mathsf{ML\text{-}DSA}}$ be the identification scheme from Figure 14. Let $H : \{0,1\}^\ell \to \mathcal{X}$ be a hash function modeled as a* *random oracle and where $\mathcal{X}$ is the input set of* $\mathsf{SampleInBall}(\tau, \cdot)$*. Let $\mathcal{A}$ be an adversary against the lossy-2-special-soundness of the scheme from Figure 14. Let $\zeta = \max(\gamma_1 - \beta, 2\gamma_2 + 1 + \tau 2^{d-1})$. There exists an adversary $\mathcal{B}$ against the* $\mathsf{MSIS}_{n,q,k,\ell+1,2\zeta}$ *problem such that:*

$$\mathsf{Adv}^{\mathsf{lossy\text{-}2\text{-}ss}}_{\mathsf{ID}_{\mathsf{ML\text{-}DSA}}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{MSIS}}_{n,q,k,\ell+1,2\zeta}(\mathcal{B}) + \frac{1}{2^\tau \binom{n}{\tau}} \ .$$

Proof. Let $(\mathbf{w}_1, c_1, c_2, \mathbf{z}_1, \mathbf{z}_2, \mathbf{h}_1, \mathbf{h}_2)$ with $c_1 \neq c_2$ such that:

- $\max(\|\mathbf{z}_1\|_\infty, \|\mathbf{z}_2\|_\infty) < \gamma_1 - \beta$,
- $\mathsf{UseHint}_q(\mathbf{h}_1, \mathbf{A}\mathbf{z}_1 - c_1'\mathbf{t}_1 \cdot 2^d, 2\gamma_2) = \mathbf{w}_1 = \mathsf{UseHint}_q(\mathbf{h}_2, \mathbf{A}\mathbf{z}_2 - c_2'\mathbf{t}_1 \cdot 2^d, 2\gamma_2)$,

where $c_b' = \mathsf{SampleInBall}(\tau, H(c_b)), b \in \{1,2\}$. For $b \in \{1,2\}$, there exists $\mathbf{u}_b \in \mathcal{R}_q^k$ such that $2\gamma_2 \cdot \mathsf{UseHint}_q(\mathbf{h}_2, \mathbf{A}\mathbf{z}_b - c_b'\mathbf{t}_1 \cdot 2^d, 2\gamma_2) = \mathbf{A}\mathbf{z}_b - c_b'\mathbf{t}_1 2^d + \mathbf{u}_b$ and $\|\mathbf{u}_b\|_\infty \leq 2\gamma_2 + 1$. We can thus rewrite $\mathbf{A}\mathbf{z}_1 - c_1'\mathbf{t} + \mathbf{u}_1 + c_1'\mathbf{t}_0 = \mathbf{A}\mathbf{z}_2 - c_2'\mathbf{t} + \mathbf{u}_2 + c_2'\mathbf{t}_0$. Letting $\mathbf{u}_b' = \mathbf{u}_b + c_b'\mathbf{t}_0$, we get:

$$(\mathbf{A}|\mathbf{t}|\mathbf{I}) \cdot \begin{pmatrix} \mathbf{z}_1 - \mathbf{z}_2 \\ c_2' - c_1' \\ \mathbf{u}_1' - \mathbf{u}_2' \end{pmatrix} = 0 \bmod q \quad \text{and} \quad \|\mathbf{u}_b'\|_\infty \leq \tau 2^{d-1} + 2\gamma_2 + 1 \ .$$

Thus, we have a candidate solution for a $\mathsf{MSIS}_{n,q,k,\ell+1,2\zeta}$ instance. This is indeed a solution if it is nonzero, which is the case if $c_1' \neq c_2'$. As H is a random oracle, we use Lemma 5 to conclude that for fixed $c_1 \in \{0,1\}^\ell$ and $c_2 \hookleftarrow U(\{0,1\}^\ell)$, this happens with probability at least $1 - 1/2^\tau\binom{n}{\tau}$. □

C Edilithium

This variant uses the curve edwards448. Nicknamed Edilithium, this is a hybrid variant of both ML-DSA65 and EdDSA448, and is described in Figure 15.

Direct application of Corollary 1 and Remark 2 give the following result.

Theorem 8. *Let $\Sigma_1 =$* EdDSA448*, and $\Sigma_2 =$* ML-DSA-65*. The signature scheme $\Sigma_\mathsf{H} =$* Edilithium *defined in Figure 15 is correct and:*

- H-EU-CMA$^\mathsf{H}$ *(resp.* sH-EU-CMA$^\mathsf{H}$*) secure in the QROM as long as* ML-DSA *is* EU-CMA *(resp.* sEU-CMA*) secure in the QROM,*
- (s)H-EU-CMA^1 *secure in the ROM as long as* EdDSA *is* EU-CMA *secure,*
- H-EU-CMA^2 *(resp.* sH-EU-CMA^2*) secure in the ROM under both (resp. all three) the* $\mathsf{MLWE}_{n,q,k,\ell,U(S_\eta)}$ *and* $\mathsf{MSIS}_{n,q,k,\ell+1,2\zeta}$ *(resp. and* $\mathsf{MSIS}_{n,q,k,\ell,2\zeta'}$*) assumptions,*
- H-EU-CMA$^\mathsf{H}$ *secure in the ROM under the hardness of* EdDSA *key-recovery or both the* $\mathsf{MLWE}_{n,q,k,\ell,U(S_\eta)}$ *and* $\mathsf{MSIS}_{n,q,k,\ell+1,2\zeta}$ *assumptions,*
- sH-EU-CMA$^\mathsf{H}$ *secure under both the same assumptions as its* H-EU-CMA$^\mathsf{H}$ *security and the* $\mathsf{MSIS}_{n,q,k,\ell,2\zeta'}$ *assumption,*

where ζ and ζ' are defined in Lemma 7 and Lemma 6, respectively.

```
$\underline{\mathsf{KeyGen}():}$
 1: $(\mathsf{vk}_1, \mathsf{sk}_1) \leftarrow \mathsf{EdDSA448.KeyGen}()$
 2: $(\mathsf{vk}_2, \mathsf{sk}_2) \leftarrow \mathsf{ML\text{-}DSA\text{-}65.KeyGen}()$
 3: $(\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0) \leftarrow \mathsf{sk}_2$
 4: $tr \leftarrow \mathcal{H}(\mathsf{vk}_1 \| \mathsf{vk}_2)$
 5: $\mathsf{sk}_2 \leftarrow (\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0)$
 6: return $((\mathsf{vk}_1, \mathsf{vk}_2), (\mathsf{sk}_1, \mathsf{sk}_2))$

$\underline{\mathsf{Verify}((\mathsf{vk}_1, \mathsf{vk}_2), (\tilde{c}, \mathbf{z}, \mathbf{h}, x), M):}$
 1: if $x < 0$ or $x \geq n$ then
 2:     return $0$
 3: $R \leftarrow x.G - \tilde{c}.\mathsf{vk}_1$
 4: $\mu \leftarrow \mathcal{H}(\mathcal{H}(\mathsf{vk}_1 \| \mathsf{vk}_2) \| R \| M)$
 5: $\sigma \leftarrow (\mathbf{z}, \tilde{c}, \mathbf{h})$
 6: $b \leftarrow \mathsf{ML\text{-}DSA\text{-}65.Verify\_mu}(\mathsf{vk}_2, \mu, \sigma)$
 7: return $b$

$\underline{\mathsf{Sign}(\mathsf{sk} = (\mathsf{sk}_1, \mathsf{sk}_2), M):}$ (external $\mu$)
 1: $(h_0, \ldots, h_{911}) = \mathsf{SHAKE256}(\mathsf{sk}, 912)$
 2: $r = \mathsf{SHAKE256}(h_{456} \| \ldots \| h_{911} \| M, 912)$
 3: $R = r.G$
 4: $\mu \leftarrow \mathcal{H}(tr \| R \| M)$
 5: $(\mathbf{z}, \tilde{c}, \mathbf{h}) \leftarrow \mathsf{ML\text{-}DSA\text{-}65.Sign\_mu}(\mathsf{sk}_2, \mu)$
 6: $x \leftarrow r + \mathsf{sk}_1 \cdot \tilde{c} \bmod n$
 7: return $(\tilde{c}, \mathbf{z}, \mathbf{h}, x)$

$\underline{\mathsf{Sign}(\mathsf{sk} = (\mathsf{sk}_1, \mathsf{sk}_2), M):}$ (context)
 1: $(h_0, \ldots, h_{911}) = \mathsf{SHAKE256}(\mathsf{sk}, 912)$
 2: $r = \mathsf{SHAKE256}(h_{456} \| \ldots \| h_{911} \| M, 912)$
 3: $R = r.G$
 4: $(\mathbf{z}, \tilde{c}, \mathbf{h}) \leftarrow \mathsf{ML\text{-}DSA\text{-}65.Sign}(\mathsf{sk}_2, M, R)$
 5: $x \leftarrow r + \mathsf{sk}_1 \cdot \tilde{c} \bmod n$
 6: return $(\tilde{c}, \mathbf{z}, \mathbf{h}, x)$
```

Fig. 15: Edilithium specification. Key generation updates the hash of vk_2 in sk_2.