Signatures with Memory-Tight Security in the Quantum Random Oracle Model

Keita Xagawa[1]

Technology Innovation Institute, UAE
keita.xagawa@tii.ae

Abstract. Auerbach, Cash, Fersch, and Kiltz (CRYPTO 2017) initiated the study of memory tightness of reductions in cryptography in addition to the standard tightness related to advantage and running time and showed the importance of memory tightness when the underlying problem can be solved efficiently with large memory. Diemert, Geller, Jager, and Lyu (ASIACRYPT 2021) and Ghoshal, Ghosal, Jaeger, and Tessaro (EUROCRYPT 2022) gave memory-tight proofs for the multi-challenge security of digital signatures in the random oracle model.
This paper studies the memory-tight reductions for *post-quantum* signature schemes in the *quantum* random oracle model. Concretely, we show that signature schemes from lossy identification are multi-challenge secure in the quantum random oracle model via memory-tight reductions. Moreover, we show that the signature schemes from lossy identification achieve more enhanced securities considering *quantum* signing oracles proposed by Boneh and Zhandry (CRYPTO 2013) and Alagic, Majenz, Russel, and Song (EUROCRYPT 2020). We additionally show that signature schemes from preimage-sampleable functions achieve those securities via memory-tight reductions.

Keywords: memory-tight reductions · signature · provable security · post-quantum cryptography · quantum random oracle model (QROM) · plus-one unforgeability · blinded unforgeability

Table of Contents

1 Introduction

Memory-tight reductions: Provable security in cryptography consists of reductions and assumptions; we assume the hardness of a computational problem, design a cryptographic scheme, and then make a reduction algorithm $\mathcal{R}$ to solve the underlying problem by using an adversary $\mathcal{A}$ breaking the security of the scheme. The tightness of the reduction is measured by how the resources of $\mathcal{R}$ are close to the resources of $\mathcal{A}$, where resources are success probability, running time, the number of queries, etc. The tightness of the security reduction is essential because it impacts the parameters of the cryptographic schemes and, thus, the scheme's efficiency. (See e.g., [BR96, Cor00, KM07, CMS12, CKMS16].)

Auerbach, Cash, Fersch, and Kiltz [ACFK17] put forth *memory tightness* of the reduction from the view of *memory usage*. This concept is important when the underlying computational problems are memory-sensitive; that is, the problem can be solved efficiently with large memory. Examples of such problems are factoring, lattice problems, the lerning-parity-with-noise (LPN) problem, and the multi-collision problem of the hash function.

After their proposal, memory-tight reductions have gathered much attention and become an active area of cryptography: Auerbach et al. [ACFK17] gave several techniques to make security proofs memory-tight. Using those techniques, they gave a memory-tight reduction for the standard security, the existential unforgeability under chosen-message attacks (EUF-CMA security), of RSA-FDH [RSA78, BR96] in the random oracle model (ROM) [BR93]. Diemert, Geller, Jager, and Lyu [DGJL21] studied memory-tight proofs of the strongly existential unforgeability under chosen-message attacks in the *multi-challenge* setting (msEUF-CMA security), where an adversary can submit multiple attempts of forgery and it wins if one of them is a 'new' forgery. They gave memory-tight reductions in the ROM for msEUF-CMA security of RSA-PFDH [BR96], the BLS signature [BLS01], and RFS-LID [AFLT12, FS87], where RFS denotes the Fiat-Shamir transform with random nonces and LID denotes a lossy identification. Ghoshal, Ghosal, Jaeger, and Tessaro [GGJT22] also gave a memory-tight proof of the msEUF-CMA security of RSA-PFDH in the multi-challenge setting in the ROM. Bhattacharyya [Bha20] and Jaeger and Kumar [JK22] gave memory-tight proofs for the security of key encapsulation mechanisms based on the variants of the Diffie-Hellman problem in the ROM. There are studies for symmetric-key cryptography, e.g., [Din20, GJT20, GGJT22], and the lower bound of memory usage of black-box reductions [ACFK17, WMHT18, GT20, GJT20].

Post-quantum signatures and quantum random oracle model: Post-quantum signature is an emerging area of cryptography as NIST had run the standardization of PQC and selected three post-quantum signatures (Falcon, Dilithium, and SPHINCS+) [AAC+22] and they started the standardization of additional signature schemes. The security of those post-quantum signatures is proven in the *quantum random oracle model (QROM)* [BDF+11], in which an adversary can make a *quantum* query to a random oracle. As far as we surveyed, the sEUF-CMA security proof for PSF-DFDH in Boneh et al. [BDF+11] is only one memory-tight proof for signature in the QROM, where PSF is preimage-sampleable functions [GPV08] and DFDH is FDH derandomized by a pseudo-random function (PRF). The following natural question arises:

> Can we construct memory-tight reductions for the msEUF-CMA security of post-quantum PSF-based signatures in the QROM?

In addition, the memory-tight security proof of the msEUF-CMA security of the signature scheme from LID in Diemert et al. [DGJL21] assumes that the underlying LID is perfectly correct, commitment-recoverable, and is considered in the ROM. Thus, it is natural to ask the following question:

> Can we construct memory-tight reductions for the msEUF-CMA security of post-quantum LID-based signatures in the QROM and eliminate the conditions on the underlying LID?

Quantum signing oracles: Furthermore, there are extended security models for signature schemes in the quantum setting by giving *quantum access* to the signing oracle. The first one is proposed by Boneh and Zhandry and dubbed EUF-qCMA security [BZ13b]. But, we call it plus-one unforgeability (PO security in short) following [AMRS20], because, in the security game, an adversary can access the signing oracle with q quantum queries and is required to output $q+1$ distinct valid message/signature pairs. The other is proposed by Alagic, Majenz, Russell, and Song [AMRS20] and dubbed (strong) blinded unforgeability (BU/sBU security in short). In the security game, an adversary can access the signing oracle with quantum queries, while some signatures are blinded if the corresponding messages are in a filter. The adversary is required to output a valid signature on a filtered message. Doosti, Delavar, Kashefi, and Arapinis [DDKA21] also gave parametrized security definitions using quantum signing oracles and showed that some of their definition is equivalent to the blinded unforgeability. Again, to the best of the authors' knowledge, there are no memory-tight proofs for such enhanced securities for post-quantum signatures. Our third question is:

> Can we construct memory-tight reductions for those extended securities (PO, BU, and sBU) of post-quantum signatures based on PSF and LID in the QROM?

Table 1. Summary of security proofs for LID-based signatures in the QROM. IND, CUR, and PRF in the column "Assumptions" denote the key indistinguishability of LID, the computational unique response of LID, and the pseudorandomness of PRF, respectively. The mark ✓ in the column "Adv." and "Time" indicates the multiplicative loss of advantage and time is $O(1)$, respectively. The marks ✓ and x in the column "Mem." indicate the additive loss of memory usage is $O(1)\cdot\mathrm{poly}$ and $O(q)\cdot\mathrm{poly}$, respectively.

Proof	Scheme	Security	Assumptions	Adv.	Time	Mem.
KLS18+DFPS23 [KLS18, DFPS23]	FS-LID	sEUF-CMA1	IND, CUR	✓	✓	x
KLS18+DFPS23 [KLS18, DFPS23]	DFS-LID	sEUF-CMA	IND, CUR, PRF	✓	✓	x
Section 4	FS-LID	msEUF-CMA1	IND, CUR	✓	✓	✓
Section 4	RFS-LID	msEUF-CMA	IND, CUR	✓	✓	✓
Section 4	DFS-LID	msEUF-CMA	IND, CUR, PRF	✓	✓	x
Section 4	DFS$^+$-LID	msEUF-CMA	IND, CUR	✓	✓	✓
Section 5	DFS-LID	PO	IND, CUR, PRF	✓	✓	x
Section 5	DFS$^+$-LID	PO	IND, CUR	✓	✓	✓
Section D	DFS-LID	sBU	IND, CUR, PRF	✓	✓	✓

1.1 Contributions

We affirmatively answer those three questions.

New memory-tight security proofs for LID-based signatures: We give a memory-tight msEUF-CMA security proof for RFS-LID in the QROM, where we allow LID to have imperfect correctness: Following Diemert et al. [DGJL21], we first show the msEUF-CMA1 security of FS-LID with memory-tight reduction, where CMA1 denotes chosen-message attacks in the one-siganture-per-message setting [KLS18] and FS denotes the standard Fiat-Shamir transform; we then obtain a memory-tight msEUF-CMA security proof for RFS-LID by using a lemma in [DGJL21].
Our main contribution is a new memory-tight security proof of the msEUF-CMA1 security of FS-LID. To obtain this proof, we carefully merge the memory-tight msEUF-CMA1 security proof by Diemert et al. [DGJL21] and the memory-loose sEUF-CMA1 security proof by Devevey, Fallahpour, Passelègue, and Stehlé [DFPS23], where the latter is a correction of the history-free proof in Kiltz, Lyubashevsky, and Schaffner [KLS18]. We note that the sEUF-CMA1 proof of Devevey et al. [DFPS23] has a subtle error[1] and we correct it.
We further extend the security proof into the PO and sBU securities. We give a memory-loose PO security proof for DFS-LID and give a memory-tight PO security proof for DFS$^+$-LID, where DFS and DFS$^+$ denote the Fiat-Shamir transform derandomized by PRF and random oracle, respectively. We also give a memory-tight sBU security proof for DFS-LID. As far as we know, those are the first PO and sBU security proofs for the LID-based signature schemes in the QROM.
We note that we can use isogeny-based and lattice-based LID schemes, e.g., Lossy CSI-FiSh [EKP20] and G+G [DPS23], as the underlying LID schemes.
See the summary and comparison in Table 1.

New memory-tight security proofs for PSF-based signatures: We extend the memory-tight sEUF-CMA security proof for PSF-DFDH in the QROM in Boneh et al. [BDF+11] into a memory-tight msEUF-CMA1 security proof for PSF-FDH in the QROM. We then obtain a memory-tight msEUF-CMA security proof for PSF-PFDH in the QROM by using the lemma in [DGJL21] as in the case of RFS-LID. Furthermore, we show a memory-loose PO security proof of PSF-DFDH and a memory-tight PO security proof of PSF-DFDH$^+$ in the QROM, where DFDH$^+$ denotes FDH derandomized by a random function. We also give a memory-tight sBU security proof of PSF-DFDH by modifying a memory-loose BU security proof of PSF-DFDH in the QROM by Chatterjee, Chung, Liang, and Malavolta [CCLM22]. See Section E for the details.
See the summary and comparison in Table 2.

A gap between PO *and* BU *security:* As a byproduct, we found that BU security does not imply PO security, which refutes the conjecture that BU security implies PO security of message authentication code (MAC) by Alagic et al. [AMRS20].[2] We exemplify this by constructing a BU-secure but PO-insecure MAC and signature

[1] They did not consider the computational unique response property of LID [KLS18]
[2] The conference version [AMRS20] claims that BU-secure MAC is also PO-secure. Unfortunately, the newest version, '20230420:091107' [AMRS18] reported an error in their proof and removed the claim.

Table 2. Summary of security proofs for PSF-based signatures in the QROM. CR, INV, OW, and PRF in the column "Assumptions" denote the collision resistance, non-invertibility, and one-wayness of PSF and the pseudorandomness of PRF, respectively. The marks ✓ and x in the columns "Adv." and "Time" indicate whether the multiplicative loss of advantage and time is $O(1)$ or not. The marks ✓ and x in the column "Mem." indicate whether the additive loss of memory usage is $O(1) \cdot \mathrm{poly}$ and $O(q) \cdot \mathrm{poly}$, respectively. In the signing algorithm of PSF-PFDH*, the randomness for PSF is chosen as 1) choose a random pairwise hash function Q and 2) compute a randomness by $Q(m)$.

Proof	Scheme	Security	Assumptions	Adv.	Time	Mem.
BDFLSZ11 [BDF+11]	PSF-DFDH+	sEUF-CMA	CR	✓	✓	✓
KX22, LJZ22 [KX22, LJZ22]	PSF-PFDH	EUF-CMA	INV	x	✓	x
CCLM22 [CCLM22]	PSF-DFDH	BU	CR, PRF	✓	✓	x
BZ13 [BZ13b]	PSF-PFDH*	PO	OW, CR	x	x	x
BZ13 [BZ13b]	PSF-DFDH	PO	CR, PRF	✓	✓	x
subsection E.3	PSF-FDH	msEUF-CMA1	CR	✓	✓	✓
subsection E.3	PSF-PFDH	msEUF-CMA	CR	✓	✓	✓
subsection E.3	PSF-DFDH	msEUF-CMA	CR, PRF	✓	✓	x
subsection E.3	PSF-DFDH+	msEUF-CMA	CR	✓	✓	✓
subsection E.4	PSF-DFDH	PO	CR, PRF	✓	✓	x
subsection E.4	PSF-DFDH+	PO	CR	✓	✓	✓
subsection E.5	PSF-DFDH	sBU	CR, PRF	✓	✓	✓

scheme from a BU-secure MAC and signature scheme, respectively. We observe that PO-secure scheme should be sEUF-CMA-secure, but BU-secure scheme can be sEUF-CMA-insecure. Thus, making BU-secure MAC and signature scheme sEUF-CMA-insecure, the new scheme is PO-insecure. We think the conjecture should be that sBU security implies PO security. See Section C for the details.

Open problems: We have managed the imperfect correctness of LID in the memory-tight security proofs of LID-based signature schemes by following the history-free approach [KLS18, DFPS23] instead of the adaptive reprogramming appraoch [GHHM21, DFPS23, BBD+23]. The history-free approach requires LID to have statistical honest-verifier zero-knowledge (HVZK). We leave an open problem to construct memory-tight security proofs treating *divergence* HVZK [dPPRS23, DPS23] or computational HVZK, which would require the adaptive reprogramming approach.
We currently can not give the memory-tight security proofs of the PSF-based signature scheme with *imperfectly correct* PSFs. Kosuge and Xagawa [KX22] gave memory-loose security proofs using the adaptive reprogramming technique [GHHM21]. We leave an open problem to give memory-tight security proof of signature schemes based on imperfectly-correct PSFs.
Jaeger and Kumar [JK22] gave memory-tight reductions for multi-challenge, multi-user CCA security of PKE/KEM. It is interesting to consider the multi-challenge, multi-user security of signature schemes with memory-tight reductions in the ROM and QROM.

Organization: Section 2 reviews basic notions and notations, quantum computations, and lossy identification. Section 3 reviews digital signatures, their security notions, and LID-based signature schemes. Section 4 and Section 5 show msEUF-CMA and PO security of the LID-based signature schemes. Section A contains missing definitions. Section B reviews the instantiations of lossy identifications. Section C discusses the relationship between PO security and BU security of MAC and signatures. Section D shows the sBU security of a LID-based signature scheme. Section E includes the review of PSF-based signature schemes and shows their msEUF-CMA, PO, and sBU securities.

2 Preliminaries

The security parameter is denoted by $\kappa \in \mathbb{Z}^+$. We use the standard O-notations. For $n \in \mathbb{Z}^+$, we let $[n] := \{1, \ldots, n\}$. For a statement P, $[\![P]\!]$ denotes the truth value of P.
Let $\mathcal{X}$ and $\mathcal{Y}$ be two finite sets. $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$ denotes a set of all functions whose domain is $\mathcal{X}$ and codomain is $\mathcal{Y}$. For a set of disributions over $\mathcal{Y}$ indexed by $x \in \mathcal{X}$, $D = \{D_x : x \in \mathcal{X}\}$, we define $\mathsf{Func}_{\mathcal{X},\mathcal{Y}}(D)$ as a distribution of f in $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$ such that, for each $x \in \mathcal{X}$, $f(x)$ is independently drawn from a ditribution D_x. When every D_x is the same as D' on every x, we simply write $\mathsf{Func}_{\mathcal{X},\mathcal{Y}}(D')$. For two distributions D, D' over $\mathcal{Y}$, we say that D is ϵ-close to D' if the distance $|D - D'| := \sum_{y \in \mathcal{Y}} |D(y) - D'(y)|$ is at most ϵ.

For a distribution D, we often write "$x \leftarrow D$," which indicates that we take a sample x according to D. For a finite set $\mathcal{S}$, $U(\mathcal{S})$ denotes the uniform distribution over $\mathcal{S}$. We often write "$x \leftarrow \mathcal{S}$" instead of "$x \leftarrow U(\mathcal{S})$." If inp is a string, then "$\mathsf{out} \leftarrow \mathsf{A}^{O}(\mathsf{inp})$" denotes the output of algorithm A running on input inp with an access to a set of oracles O. If A and oracles are deterministic, then out is a fixed value and we write "$\mathsf{out} := \mathsf{A}^{O}(\mathsf{inp})$." We also use the notation "$\mathsf{out} := \mathsf{A}(\mathsf{inp}; r)$" to make the randomness r of A explicit. For a probabilistic algorithm A, $\mathcal{R}_{\mathsf{A}}$ denotes the radnomness space of A.
For an algorithm or adversary A, Time(A) and Mem(A) denotes the time and memory complexity of the algorithm A, respectively. For a scheme S, Time(S) and Mem(S) denotes the maximum time and memory complexity of the algorithms in the scheme S, respectively.
For any function $f\colon \{0,1\}^n \to \{0,1\}^m$, a *quantum access* to f is modeled as oracle access to unitary $O_f\colon |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$. By convention, we will use the notation $\mathsf{A}^{|f\rangle,g}$ to stress A's *quantum* and classical access to f and g.

2.1 Lemmas on Quantum Computations

Lemma 2.1 ([BZ13b, Lemma 2.5, ePrint]). *Let $\mathcal{X}$ and $\mathcal{Y}$ be two finite sets. Let $D = \{D_x\}$ and $D' = \{D'_x\}$ be two sets of efficiently sampleable distributions over $\mathcal{Y}$ indexed by $x \in \mathcal{X}$. let $\mathcal{A}$ be a quantum adversary making q (quantum) queries to an oracle $f\colon \mathcal{X} \to \mathcal{Y}$. If for each $x \in X$, $|D_x - D'_x| \le \epsilon$ holds, then $|\Pr[f \leftarrow \mathsf{Func}_{\mathcal{X},\mathcal{Y}}(D) : \mathcal{A}^{|f\rangle} = 1] - \Pr[f \leftarrow \mathsf{Func}_{\mathcal{X},\mathcal{Y}}(D') : \mathcal{A}^{|f\rangle} = 1]| \le \sqrt{(6q)^3\epsilon}$.* [3]

Lemma 2.2 ([BZ13b, Lemma 2.6, ePrint]). *Fix two finite sets $\mathcal{X}$ and $\mathcal{Y}$. Fix a set D of distributions D_x over $\mathcal{Y}$ indexed by $x \in \mathcal{X}$. Let α be the minimum over all $x \in \mathcal{X}$ of the min-entropy of the distribution D_x. Now, let $f\colon \mathcal{X} \to \mathcal{Y}$ be a function chosen according to $\mathsf{Func}_{\mathcal{X},\mathcal{Y}}(D)$. Then, any q-query quantum algorithm can only produce $(q+1)$ input/output pairs of f with probability at most $(q+1)/\lfloor 2^{\alpha} \rfloor$.*

2.2 Adversaries with Access to Random Functions

We adopt an adversary with access to random functions by following [GGJT22, Section 3]. The reductions in this paper are adversary $\mathcal{A}$ on the left side, consisting of a set of functions $\mathcal{F}$ and algorithm $\mathcal{A}_2$. We call such an adversary *a $\mathcal{F}$-oracle adversary.*

Adversary $\mathcal{A}^{O}(\mathsf{in})$	Adversary $\mathcal{A}_F^{O}(\mathsf{in})$
1 : $f \leftarrow \mathcal{F}$	1 : $K \leftarrow \mathcal{K}$
2 : $\mathsf{out} \leftarrow \mathcal{A}_2^{O,\lvert f\rangle}(\mathsf{in})$	2 : $\mathsf{out} \leftarrow \mathcal{A}_2^{O,\lvert F_K\rangle}(\mathsf{in})$
3 : **return** out	3 : **return** out

Ghoshal et al. [GGJT22] defined the *reduced complexity* of $\mathcal{A}$ by $\mathsf{Time}^*(\mathcal{A}) := \mathsf{Time}(\mathcal{A}_2)$ and $\mathsf{Mem}^*(\mathcal{A}) := \mathsf{Mem}(\mathcal{A}_2)$. We employ $\mathcal{F}$-oracle adversaries as [GGJT22] for simplicity and clean notation. This approach is justified by pseudorandom adversary $\mathcal{A}_F$ on the right-hand side as long as the game $\mathcal{A}$ plays is efficient. See [ACFK17] and [GGJT22, Lemma 2].

2.3 Lossy Identification

Abdalla et al. [AFLT12, AFLT16] defined lossy identification as a special case of a (cryptographic) identification scheme. A lossy identification scheme involves an additional *lossy* key-generation algorithm. The syntax follows:

Definition 2.1 (Lossy identification). *A lossy identification scheme* LID *consists of the following tuple of PPT algorithms* $(\mathsf{Gen}_{\mathsf{LID}}, \mathsf{LossyGen}_{\mathsf{LID}}, \mathsf{P}_1, \mathsf{P}_2, \mathsf{V})$

- $\mathsf{Gen}_{\mathsf{LID}}(1^{\kappa}) \to (vk, sk)$*: a normal key-generation algorithm that, on input 1^{κ}, where κ is the security parameter, outputs a pair of keys (vk, sk). vk and sk are public verification and secret keys, respectively.*
- $\mathsf{LossyGen}_{\mathsf{LID}}(1^{\kappa}) \to vk$*: a lossy key-generation algorithm that on input 1^{κ} outputs a lossy verification key vk.*

[3] The value $6^3 = 27 \cdot 8$ is taken from 27 in [Zha12, Corollary 7.5, ePrint] (denoted by C_0 in [BZ13b]) and 8 in [BZ13b, Lemma 2.5].

- $\mathsf{P}_1(sk) \to (w, s)$: *a first prover algorithm that takes as input signing key sk and outputs commitment w and state s.*
- $\mathsf{P}_2(sk, w, c, s) \to z$: *a second* deterministic *prover algorithm that takes as input signing key sk, commitment w, challenge c, and state s, and outputs response z.*
- $\mathsf{Vrfy}(vk, w, c, z) \to \mathsf{true}/\mathsf{false}$: *a* deterministic *verification algorithm that takes as input verification key vk, commitment w, challenge c, and response z, and outputs its decision* true *or* false.

We assume that a verification key vk defines the challenge space C and the response space $\mathcal{Z}$.

Definition 2.2 (Completeness). *For non-negligible $\rho = \rho(\kappa)$, we call* LID ρ-complete *if*

$$\Pr\left[\begin{array}{c}(vk, sk) \leftarrow \mathsf{Gen}_{\mathsf{LID}}(1^\kappa), (w, s) \leftarrow \mathsf{P}_1(sk), \\ c \leftarrow C, z := \mathsf{P}_2(sk, w, c, s)\end{array} : \mathsf{V}(vk, w, c, z) = \mathsf{true}\right] \geq \rho(\kappa).$$

We call LID perfecly complete *if it is* 1*-complete.*

In order to analyze the completeness carefully, Devevey et al. [DFPS23] introduced another definition as follows:

Definition 2.3 (Correctness [DFPS23, Def.2], adapted). *Let $\gamma, \beta > 0$. We call* LID (γ, β)-complete *if for every (vk, sk) generated by $\mathsf{Gen}_{\mathsf{LID}}(1^\kappa)$, the following holds:*

- *The verifier accepts with probability at least γ if the response z is not $\perp$. That is,*

$$\Pr\left[(w, s) \leftarrow \mathsf{P}_1(sk), c \leftarrow C, z := \mathsf{P}_2(sk, w, c, s) : \mathsf{V}(vk, w, c, z) = \mathsf{true} \mid z \neq \perp\right] \geq \gamma.$$

- *The prover aborts with probability at most β. That is,*

$$\Pr\left[(w, s) \leftarrow \mathsf{P}_1(sk), c \leftarrow C, z := \mathsf{P}_2(sk, w, c, s) : z = \perp\right] \leq \beta.$$

We note that if LID is $(1, \beta)$-correct, then it is β-complete.
The security properties of a lossy identification scheme are defined as follows:

Definition 2.4 (Key indistinguishability [AFLT16, Def.16]). *We say that* LID *is* key indistinguishable *if for any QPT adversary $\mathcal{A}$, its advantage $\mathsf{Adv}^{\text{ind-key}}_{\mathsf{LID},\mathcal{A}}(\kappa)$ is negligible in κ, where*

$$\mathsf{Adv}^{\text{ind-key}}_{\mathsf{LID},\mathcal{A}}(\kappa) := \left|\begin{array}{l}\Pr[(vk, sk) \leftarrow \mathsf{Gen}_{\mathsf{LID}}(1^\kappa) : \mathcal{A}(vk) = 1] \\ -\Pr[vk \leftarrow \mathsf{LossyGen}_{\mathsf{LID}}(1^\kappa) : \mathcal{A}(vk) = 1]\end{array}\right|.$$

Definition 2.5 (Lossiness [AFLT16, Def.16], adapted). *We say that* LID *is ϵ_ℓ-*lossy *if for any unbounded adversary $\mathcal{A}$, its advantage $\mathsf{Adv}^{\text{imp}}_{\mathsf{LID},\mathcal{A}}(\kappa)$ is at most ϵ_ℓ, where*

$$\mathsf{Adv}^{\text{imp}}_{\mathsf{LID},\mathcal{A}}(\kappa) := \Pr\left[\begin{array}{c}vk \leftarrow \mathsf{LossyGen}_{\mathsf{LID}}(1^\kappa), (w, s) \leftarrow \mathcal{A}(vk), \\ c \leftarrow C, z \leftarrow \mathcal{A}(c, s)\end{array} : \mathsf{V}(vk, w, c, z) = \mathsf{true}\right].$$

Remark 2.1 (On Lossiness). In the original definition, adversary $\mathcal{A}$ can access to the simulated transcript oracle to produce (w, s). However, since the simulated transcript oracle has no access to sk, we do need to consider this oracle.

Definition 2.6 (Statistical honest-verifier zero knowledge [DFPS23], adapted). *Let (vk, sk) be a key pair generated by $\mathsf{Gen}_{\mathsf{LID}}(1^\kappa)$. We call* LID ϵ_{zk}-HVZK *if there exists a PPT algorithm* Sim *that takes a public verification key vk and c as input and outputs (w, z) such that the distribution of (w, c, z) where $c \leftarrow C$ and $(w, z) \leftarrow \mathsf{Sim}(vk, c)$ is ϵ_{zk}-close to the distribution of the real transcript between honest prover and verifier.*

Remark 2.2. In [AFLT16, KLS18], "the distribution of the real transcript" is defined as follows: compute (w, c, z) by using the real prover and verifier; if $z = \perp$, then return $(\perp, \perp, \perp)$; otherwise return (w, c, z). Devevey et al. [DFPS23] pointed out that this definition is one of the causes of the error in the simulation. They defined "the distribution of the real transcript" as follows: compute (w, c, z) by using the real prover and verifier and output (w, c, z) *as it is.*

Definition 2.7 (Commitment recoverability [KLS18, Defnition 2.4]). *We say that* LID *is* commitment-recoverable *if for any (vk, sk) generated by $\mathsf{Gen}_{\mathsf{LID}}(1^\kappa)$, $c \in C$, and $z \in \mathcal{Z}$, there exists a* unique *w such that $\mathsf{V}(vk, w, c, z) =$* true*. In addition, we require that this unique w can be computed by a deterministic commitment-recovery algorithm* Rec*, that is, $w = \mathsf{Rec}(vk, c, z)$.*

Definition 2.8 (Computational unique response [KLS18, Defnition 2.7]). *We also say that* LID *has the* computational unique response (CUR) *property if for any QPT adversary $\mathcal{A}$, its advantage defined below is negligible in κ:*

$$\mathsf{Adv}^{\mathrm{cur}}_{\mathsf{LID},\mathcal{A}}(\kappa) := \Pr\left[\begin{array}{c}(vk, sk) \leftarrow \mathsf{Gen}_{\mathsf{LID}}(1^\kappa), (w, c, z, z') \leftarrow \mathcal{A}(vk) : \\ z \neq z' \wedge \mathsf{V}(vk, w, c, z) \wedge \mathsf{V}(vk, w, c, z')\end{array}\right].$$

Definition 2.9 (Min-entropy of commitment [KLS18, Defnition 2.6], modified). *We say that* LID *has* (α, ϵ_m)-min-entropy *if*

$$\Pr[(vk, sk) \leftarrow \mathsf{Gen}_{\mathsf{LID}}(1^\kappa) : H_\infty(w \mid (w, s) \leftarrow \mathsf{P}_1(sk)) \geq \alpha] \geq 1 - \epsilon_m.$$

In the original definition ([KLS18, Defnition 2.6]), ϵ_m is $2^{-\alpha}$. Devevey et al. [DFPS23, Definition 5] defined the min-entropy of commitment as $(\alpha, 0)$-min entropy in the above definition.

3 Digital Signature

The model for digital signature schemes is summarized as follows:

Definition 3.1. *A digital signature scheme* DS *consists of the following triple of PPT algorithms* (Gen, Sign, Vrfy)*:*

- $\mathsf{Gen}(1^\kappa) \to (vk, sk)$*: a key-generation algorithm that, on input 1^κ, where κ is the security parameter, outputs a pair of keys (vk, sk). vk and sk are called verification and signing keys, respectively.*
- $\mathsf{Sign}(sk, \mu) \to \sigma$*: a signing algorithm that takes as input signing key sk and message $\mu \in \mathcal{M}$ and outputs signature $\sigma \in \mathcal{S}$.*
- $\mathsf{Vrfy}(vk, \mu, \sigma) \to$ true/false*: a verification algorithm that takes as input verification key vk, message $\mu \in \mathcal{M}$, and signature σ and outputs its decision* true *or* false.

We require statistical correctness; that is, for any message $\mu \in \mathcal{M}$, we have

$$\Pr[(vk, sk) \leftarrow \mathsf{Gen}(1^\kappa), \sigma \leftarrow \mathsf{Sign}(sk, \mu) : \mathsf{Vrfy}(vk, \mu, \sigma) = \mathsf{true}] \geq 1 - \delta(\kappa)$$

for some negligible function δ.

Security notions: We review the standard security notions of digital signature schemes. The standard security notion, existential unforgeability against chosen-message attack (EUF-CMA), is captured by the game $\mathsf{Expt}^{\text{euf-cma}}_{\mathsf{DS},\mathcal{A}}(1^\kappa)$ in Figure 1. The multi-challenge version allows an adversary to call Forge freely [ACFK17]. We also consider a strong version, in which the adversary wins if its forgery (m^*, σ^*) is not equal to the pairs returned by Sign. For signing oracles, we have two variants, one-signature-per-message / many-signature-per-message versions, which are denoted by CMA1 and CMA, respectively. We note that for deterministic signature schemes, CMA1 security implies CMA security. The formal definition follows:

Definition 3.2 (Security notions for digital signature schemes). *Let* DS $=$ (Gen, Sign, Vrfy) *be a digital signature scheme. For any $\mathcal{A}$,* goal $\in$ {euf, seuf, meuf, mseuf}*, and* atk $\in$ {cma, cma1}*, we define its* goal-atk *advantage against* DS *as follows:*

$$\mathsf{Adv}^{\text{goal-atk}}_{\mathsf{DS},\mathcal{A}}(\kappa) := \Pr[\mathsf{Expt}^{\text{goal-atk}}_{\mathsf{DS},\mathcal{A}}(1^\kappa) = 1],$$

where $\mathsf{Expt}^{\text{goal-atk}}_{\mathsf{DS},\mathcal{A}}(1^\kappa)$ *is an experiment described in Figure 1. For* GOAL $\in$ {EUF, sEUF, mEUF, msEUF} *and* ATK $\in$ {CMA, CMA1}*, we say that* DS *is* GOAL-ATK*-secure if* $\mathsf{Adv}^{\text{goal-atk}}_{\mathsf{DS},\mathcal{A}}(\kappa)$ *is negligible for any QPT adversary $\mathcal{A}$.*

Security with respect to quantum signing oracles: Boneh and Zhandry [BZ13b] defined a new security notion of digital signature schemes with respect to a *quantum* signing oracle and dubbed it as EUF-qCMA security. We refer to this security notion as *plus-one security (PO security)* [AMRS20] because an adversary in the security game is asked to output $q + 1$ distinct valid message/signature pairs after making q quantum queries to the signing oracle. They defined it in the same spirit as the *strong* EUF security. In the original definition, the adversary outputs $q + 1$ pairs at once and stops. We introduce the oracle Forge to the security game of the PO security since we want to consider memory tightness. The formal definition follows:

Definition 3.3 (Plus-One Security [BZ13b], adapted). *Let* DS $=$ (Gen, Sign, Vrfy) *be a digital signature scheme. For any $\mathcal{A}$, we define its* po *advantage against* DS *as follows:*

$$\mathsf{Adv}^{\mathrm{po}}_{\mathsf{DS},\mathcal{A}}(\kappa) := \Pr[\mathsf{Expt}^{\mathrm{po}}_{\mathsf{DS},\mathcal{A}}(1^\kappa) = 1],$$

where $\mathsf{Expt}^{\mathrm{po}}_{\mathsf{DS},\mathcal{A}}(1^\kappa)$ *is an experiment described in Figure 2. We say that* DS *is* PO-secure *if* $\mathsf{Adv}^{\mathrm{po}}_{\mathsf{DS},\mathcal{A}}(\kappa)$ *is negligible for any QPT adversary $\mathcal{A}$.*

$\mathrm{Expt}^{\text{goal-atk}}_{\mathrm{DS},\mathcal{A}}(1^\kappa)$ for goal = euf and seuf

$(vk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$
$\mathsf{win} \coloneqq \mathsf{false}; Q \coloneqq \emptyset$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\textsc{Sign}}(vk)$
$\textsc{Forge}(m^*, \sigma^*)$
return win

$\textsc{Sign}(m)$ for atk = cma

$\sigma \leftarrow \mathsf{Sign}(sk, m)$
$Q \coloneqq Q \cup \{(m, \sigma)\}$
return σ

$\textsc{Forge}(m^*, \sigma^*)$ for goal = euf and meuf

if $\mathsf{Vrfy}(vk, m^*, \sigma^*) = \mathsf{true}$ **then**
  if $\forall \sigma : (m^*, \sigma) \notin Q$ **then**
    $\mathsf{win} \coloneqq \mathsf{true}$

$\mathrm{Expt}^{\text{goal-atk}}_{\mathrm{DS},\mathcal{A}}(1^\kappa)$ for goal = meuf and mseuf

$(vk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$
$\mathsf{win} \coloneqq \mathsf{false}; Q \coloneqq \emptyset$
run $\mathcal{A}^{\textsc{Sign},\textsc{Forge}}(vk)$
return win

$\textsc{Sign}(m)$ for atk = cma1

if $\exists (m, \sigma) \in Q$ for some σ **then**
  return σ
else $\sigma \leftarrow \mathsf{Sign}(sk, m)$
$Q \coloneqq Q \cup \{(m, \sigma)\}$
return σ

$\textsc{Forge}(m^*, \sigma^*)$ for goal = seuf and mseuf

if $\mathsf{Vrfy}(vk, m^*, \sigma^*) = \mathsf{true}$ **then**
  if $(m^*, \sigma^*) \notin Q$ **then**
    $\mathsf{win} \coloneqq \mathsf{true}$

Fig. 1. $\mathrm{Expt}^{\text{goal-atk}}_{\mathrm{DS},\mathcal{A}}(1^\kappa)$ for goal $\in$ {euf, seuf, meuf, mseuf} and atk $\in$ {cma, cma1}.

$\mathrm{Expt}^{\mathrm{po}}_{\mathrm{DS},\mathcal{A}}(1^\kappa)$

$(vk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$
$Q \coloneqq \emptyset$
run $\mathcal{A}^{|\textsc{Sign}\rangle,\textsc{Forge}}(vk)$
return $[\![\#Q > q_S]\!]$

$\textsc{Sign}: |m\rangle|y\rangle \mapsto |m\rangle|y \oplus \sigma\rangle$

/ generate randomness r on each query
/ share r on every messages
$\sigma \leftarrow \mathsf{Sign}(sk, m; r)$
return σ

$\textsc{Forge}(m^*, \sigma^*)$

if $\mathsf{Vrfy}(vk, m^*, \sigma^*) = \mathsf{true}$ **then**
  if $(m^*, \sigma^*) \notin Q$ **then**
    $Q \coloneqq Q \cup \{(m^*, \sigma^*)\}$

Fig. 2. $\mathrm{Expt}^{\mathrm{po}}_{\mathrm{DS},\mathcal{A}}(1^\kappa)$. q_S denotes the number of the signing queries.

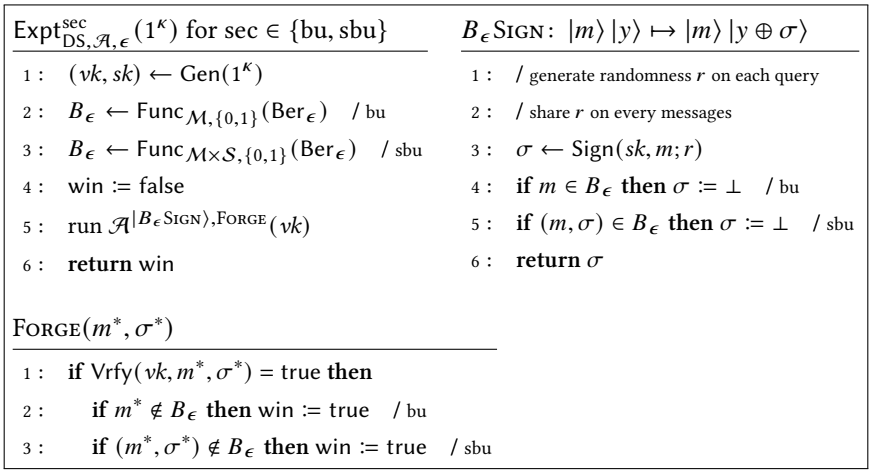

Fig. 3. $\mathrm{Expt}^{\mathrm{bu}}_{\mathsf{DS},\mathcal{A},\epsilon}(1^\kappa)$ and $\mathrm{Expt}^{\mathrm{sbu}}_{\mathsf{DS},\mathcal{A},\epsilon}(1^\kappa)$.

Alagic et al. [AMRS20] introduced another new security notion concerning a quantum signing oracle and called it *blinded unforgeability (*BU *security)*. Let $\epsilon \in \{0/2^p, 1/2^p, \ldots, (2^p-1)/2^p\}$ for some $p = p(\kappa)$ be a parameter. Let B_ϵ be a random subset of the message space $\mathcal{M}$ where each $m \in \mathcal{M}$ is independently selected with probability ϵ. Roughly speaking, an adversary is asked to output a valid signature on a message in B_ϵ while it can access a quantum signing oracle that returns a signature on a message *not* in B_ϵ. The strong version is defined by a subset of the product of the message space $\mathcal{M}$ and the signature space $\mathcal{S} \subseteq \{0,1\}^\lambda$ for some $\lambda = \lambda(\kappa)$. For $f : \mathcal{M} \to \mathcal{S}$, $B \subseteq \mathcal{M}$, and $B' \subseteq \mathcal{M} \times \mathcal{S}$, we define

$$Bf(x) := \begin{cases} \perp & x \in B \\ f(x) & \text{otherwise} \end{cases} \quad \text{and} \quad B'f(x) := \begin{cases} \perp & (x, f(x)) \in B' \\ f(x) & \text{otherwise} \end{cases}$$

Remark 3.1. We consider the oracle $|Bf\rangle$ as a mapping $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus Bf(x)\rangle$, where $y \in \{0,1\}^{\lambda+1}$, $f(x)$ is considered as $f(x)\|0 \in \{0,1\}^{\lambda+1}$, and $\perp$ is considered as $0^\lambda\|1 \in \{0,1\}^{\lambda+1}$.
In the security proofs, instead of choosing a random subset B_ϵ, we will consider $\mathsf{RF}_B : \mathcal{M} \to \mathcal{P}$, where $\mathcal{P} = \{0, 1, \ldots, 2^p - 1\}$, and we will interpret the condition $m \in B_\epsilon$ as $\mathsf{RF}_B(m) < \epsilon 2^p$. The cost of this procedure is denoted by $\mathsf{Time}(B_\epsilon)$ and $\mathsf{Mem}(B_\epsilon)$.

We again introduce the oracle FORGE to the security game and consider the multi-challenge situation. The formal definition follows:

Definition 3.4 (Blinded Unforgeability [AMRS20], adapted). *Let* $\mathsf{DS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *be a digital signature scheme. For any* $\mathcal{A}$*, any efficiently computable function* $\epsilon : \mathbb{Z}^+ \to [0,1)$*, and* $\mathrm{sec} \in \{\mathrm{bu}, \mathrm{sbu}\}$*, we define its* goal-atk *advantage against* DS *as follows:*

$$\mathsf{Adv}^{\mathrm{sec}}_{\mathsf{DS},\mathcal{A}}(\kappa) := \Pr[\mathrm{Expt}^{\mathrm{sec}}_{\mathsf{DS},\mathcal{A},\epsilon}(1^\kappa) = 1],$$

where $\mathrm{Expt}^{\mathrm{sec}}_{\mathsf{DS},\mathcal{A}}(1^\kappa)$ *is an experiment described in Figure 3. We say that* DS *is* BU-secure *(*sBU-secure*, resp.) if* $\mathsf{Adv}^{\mathrm{bu}}_{\mathsf{DS},\mathcal{A},\epsilon}(\kappa)$ *(*$\mathsf{Adv}^{\mathrm{sbu}}_{\mathsf{DS},\mathcal{A},\epsilon}(\kappa)$*, resp.) is negligible for any QPT adversary* $\mathcal{A}$ *and any efficiently computable function* $\epsilon : \mathbb{Z}^+ \to [0,1)$.

3.1 From CMA1 Security to CMA Security

Diemert et al. [DGJL21] shows that the following lemma, which is the multi-challenge version of [BPS16, Theorem 5].

Lemma 3.1 ([DGJL21, Thm.14]). *Let* DS' *be a signature scheme whose message space is* $\mathcal{M}' = \mathcal{M} \times \{0,1\}^\lambda$ *and let* DS' *be* $\mathsf{RDS}[\mathsf{DS}', \lambda]$ *in Figure 4. Let* $\mathcal{A}$ *be an adversary against the* msEUF-CMA *security of* DS *which queries to* SIGN q_S *times. Then, there exists an adversary* $\mathcal{B}$ *against the* msEUF-CMA1 *security of* DS' *such that*

$$\mathsf{Adv}^{\text{mseuf-cma}}_{\mathsf{DS},\mathcal{A}}(\kappa) \le \mathsf{Adv}^{\text{mseuf-cma1}}_{\mathsf{DS}',\mathcal{B}}(\kappa) + q_S^2 \cdot 2^{-\lambda}, \mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}), \text{ and } \mathsf{Mem}(\mathcal{B}) = \mathsf{Mem}(\mathcal{A}).$$

$\mathsf{Gen}(1^\kappa)$	$\mathsf{Sign}(sk, m)$	$\mathsf{Vrfy}(vk, m, \sigma)$ with $\sigma = (\sigma', n)$
1 : $(vk, sk) \leftarrow \mathsf{Gen}'(1^\kappa)$	1 : $n \leftarrow \{0,1\}^\lambda$	1 : **return** $\mathsf{Vrfy}'(vk, (m, n), \sigma')$
2 : **return** (vk, sk)	2 : $\sigma' \leftarrow \mathsf{Sign}'(sk, (m, n))$	
	3 : **return** $\sigma := (\sigma', n)$	

Fig. 4. Scheme $\mathsf{DS} := \mathsf{RDS}[\mathsf{DS}', \lambda]$. We require $\mathcal{M}' = \mathcal{M} \times \{0,1\}^\lambda$.

If the signature scheme is *deterministic*, then the CMA1 security implies the CMA security [KLS18]. Thus, if we derandomize a CMA1-secure signature scheme, the obtained one archives CMA security. See, e.g., [MNPV99, KM15] for the derandomization by PRF. We note that the derandomization proposed by Bellare et al. [BPS16] compute randomness as $\mathsf{RF}(sk\|m)$ instead of $\mathsf{PRF}(K, m)$ where sk is the signing key and K is a key for PRF and their proof does not work for the case of $\mathsf{H}(K, m)$ while Kiltz et al. [KLS18] credited the proof for $\mathsf{PRF}(K, m)$ to Bellare et al. [BPS16]. Unfortunately, the derandomization by PRF is sometimes annoying when we consider memory-tight reductions. In such cases, we can employ derandomization by random oracle.

3.2 Signature from Lossy Identification

We review a signature scheme constructed from a lossy identification scheme with abort [AFLT16, KLS18]. Let $\mathsf{LID} = (\mathsf{Gen}_{\mathsf{LID}}, \mathsf{LossyGen}_{\mathsf{LID}}, \mathsf{P}_1, \mathsf{P}_2, \mathsf{V})$ be a lossy idenfitication scheme. The signature scheme obtained by applying a variant of the Fiat-Shamir transform $\mathsf{FS}_{B,wz}$ is depicted in Figure 5. One might think if the underlying LID scheme is ρ-complete, then the obtained scheme is $(1-\rho)^B$-correct. Devevey et al. [DFPS23] gave a careful analysis of the correctness of the obtained signature scheme:

Lemma 3.2 ([DFPS23, Thm.8], adapted). *Let $\gamma > 0$ and $\beta \in (0,1)$. Let $B > 0$. Let $\mathsf{H} \colon \mathcal{M} \times \mathcal{W} \to \mathcal{C}$ be a hash function modeled as a random oracle. Let LID be a LID scheme that is (γ, β)-correct and has (α, ϵ_m)-commitment min-entropy. Let $\mathsf{DS} = \mathsf{FS}_{B,wz}[\mathsf{LID}, \mathsf{H}]$. Then, for any message $\mu \in \mathcal{M}$, we have*

$$\Pr_{(vk,sk)\leftarrow\mathsf{Gen}(1^\kappa)}\left[\Pr[\mathsf{V}(vk, m, \mathsf{Sign}(sk, m)) = \mathsf{true}] \geq \gamma \cdot \left(1 - \beta^B - \frac{2^{-\alpha}}{(1-\beta)^3}\right)\right] \geq 1 - \epsilon_m,$$

where the inner probalitiy is taken over the choice of H and the coins of Sign.

For simplicity, in what follows, we just say that the signature scheme is ρ'-correct with probability at least $1 - \epsilon_m$ over the choice of key, where $\rho' = \rho'(\gamma, \beta, \alpha) := \gamma \cdot \left(1 - \beta^B - \frac{2^{-\alpha}}{(1-\beta)^3}\right)$. We note that if LID is $(\gamma, 1)$-correct, then the obtained signature scheme is γ-correct.

When the underlying LID is commitment-recoverable, we can apply another variant $\mathsf{FS}_{B,cz}$ depicted in Figure 5 whose signature is of the form (c, z), which is often shorter than (w, z). If P_1 is derandomized by PRF, say, $\mathsf{P}_1(sk; \mathsf{PRF}(K, (m, k)))$, then we call this conversion as DFS instead of FS and denote $\mathsf{DFS}[\mathsf{LID}, \mathsf{H}, \mathsf{PRF}]$. If we use RF instead of PRF, then we denote it as $\mathsf{DFS}^+[\mathsf{LID}, \mathsf{H}, \mathsf{RF}]$. If we apply RDS in subsection 3.1 to the obtained scheme, then we call the conversion as RFS and denote $\mathsf{RFS}[\mathsf{LID}, \mathsf{H}, \lambda]$.

In this paper, we employ $\mathsf{FS}_{B,wz}$ to capture generic case, while [DGJL21] only consider $\mathsf{FS}_{B,cz}$. We can show the security of $\mathsf{FS}_{B,cz}$ by modifying our proofs for msEUF-CMA, sBU, and PO securities.

4 Multi-Challenge Security of Signature from Lossy Identification

Theorem 4.1 (msEUF-CMA1 security of $\mathsf{FS}_{B,wz}[\mathsf{LID}, \mathsf{H}]$). *Let $B \geq 0$. Let $\mathsf{H} \colon \mathcal{M} \times \mathcal{W} \to \mathcal{C}$ be a hash function modeled as a random oracle. Let LID be a lossy identification scheme that is ρ-complete, ϵ_{zk}-HVZK, and ϵ_ℓ-lossy, and has (α, ϵ_m)-commitment min-entropy. Let $\mathsf{DS} := \mathsf{FS}_{B,wz}[\mathsf{LID}, \mathsf{H}]$.*

Then, for a quantum adversary $\mathcal{A}$ breaking the msEUF-CMA1 security of DS that issues at most q_H quantum queries to the random oracle H, q_S classical queries to the signing oracle, and q_F classical queries to the forgery oracle, there exist quantum $\mathcal{F}$-oracle adversaries $\mathcal{A}_{\mathrm{cur}}$ against computationally unique response of LID and $\mathcal{A}_{\mathrm{ind}}$

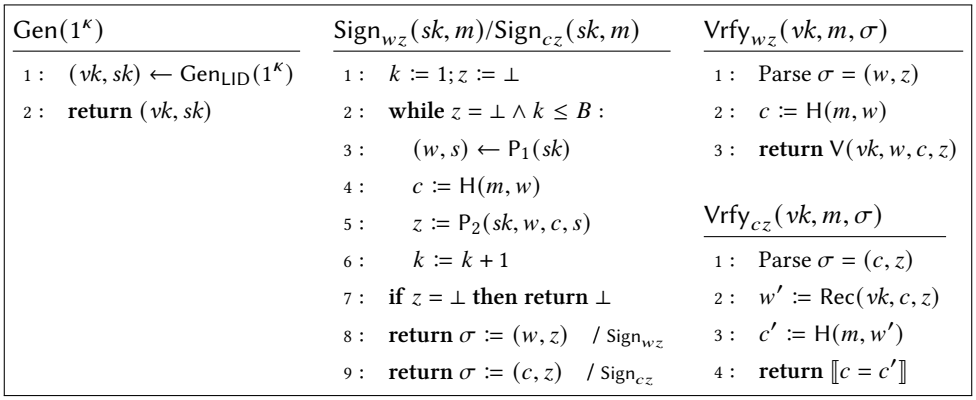

Fig. 5. Scheme $\mathsf{FS}_{B,wz}[\mathsf{LID}, \mathsf{H}] = (\mathsf{Gen}, \mathsf{Sign}_{wz}, \mathsf{Vrfy}_{wz})$ and $\mathsf{FS}_{B,cz}[\mathsf{LID}, \mathsf{H}] = (\mathsf{Gen}, \mathsf{Sign}_{cz}, \mathsf{Vrfy}_{cz})$. $\mathsf{H}\colon \mathcal{M} \times \mathcal{W} \to \mathcal{C}$ is a random oracle.

against key indistinguishability of LID *such that*

$$\begin{aligned}
\mathsf{Adv}^{\text{mseuf-cma1}}_{\mathsf{DS},\mathcal{A}}(\kappa) &\leq \mathsf{Adv}^{\text{cur}}_{\mathsf{LID},\mathcal{A}_{\text{cur}}}(\kappa) + \mathsf{Adv}^{\text{ind-key}}_{\mathsf{LID},\mathcal{A}_{\text{ind}}}(\kappa) + (q_S + q_F)(1 - \rho') \\
&\quad + 8qB2^{-\frac{-\alpha-1}{2}} + 7\epsilon_m + 4\sqrt{(6q)^3 B\epsilon_{\text{zk}}} + 2q_F 2^{-\alpha} + 8(q+1)^2\epsilon_\ell, \\
\mathsf{Time}^*(\mathcal{A}_{\text{cur}}) &= \mathsf{Time}(\mathcal{A}) + q \cdot O(B\mathsf{Time}(\mathsf{LID}) + B^2), \\
\mathsf{Mem}^*(\mathcal{A}_{\text{cur}}) &= \mathsf{Mem}(\mathcal{A}) + O(B\mathsf{Mem}(\mathsf{LID})), \\
\mathsf{Time}^*(\mathcal{A}_{\text{ind}}) &= \mathsf{Time}(\mathcal{A}) + q \cdot O(B\mathsf{Time}(\mathsf{LID}) + B^2), \\
\mathsf{Mem}^*(\mathcal{A}_{\text{ind}}) &= \mathsf{Mem}(\mathcal{A}) + O(B\mathsf{Mem}(\mathsf{LID})),
\end{aligned}$$

where $q = q_S + q_H + q_F$, ρ' *is defined as in subsection 3.2, and* $\mathcal{F} = \mathsf{Func}(\mathcal{M} \times \mathcal{W}, \mathcal{C}) \times \mathsf{Func}(\mathcal{M} \times [B], \mathcal{C}) \times \mathsf{Func}(\mathcal{M} \times [B], \mathcal{R})$.

Applying Lemma 3.1, we obtain the following corollary.

Corollary 4.1 (msEUF-CMA **security of** $\mathsf{RFS}_{B,wz}[\mathsf{LID}, \mathsf{H}, \lambda]$). *For sufficiently large* $\lambda = \omega(\kappa)$, $\mathsf{RFS}_{B,wz}[\mathsf{LID}, \mathsf{H}, \lambda]$ *has a memory-tight* msEUF-CMA *security proof.*

4.1 Proof of Theorem

Roadmap: We define thirteen games G_i for $i \in \{0, 1, \ldots, 12\}$ to show our theorem. Let W_i denote the event that G_i outputs true. Before describing games, we briefly give intuitions for games. In what follows, GetTrans(m) denotes the oracle generating at most B transcripts invoked from the signing oracle.
First, we mainly follow the proof by Devevey et al. [DFPS23]. We consider the original game (G_0), in which the signing oracle queried on m calls GetTrans(m) internally and use this real transcript as a signature. We then modify the random oracle to patch the hash value on $\mathsf{H}(m, w^{(i)})$ by $c^{(i)}$ instead of $\mathsf{RF}_{\mathsf{H}}(m, w^{(i)})$, where $(w^{(i)}, c^{(i)}, z^{(i)})$ is the i-th transcript of GetTrans(m) (G_4). We further implement GetTrans(m) by the simulator (G_7), which removes the use of sk in the following games. (We note that, because of a technical reason related to the CUR property, we modify the forge oracle to abort if the signing oracle fails to make a valid signature on input m^* (G_2). This event is ignored in Kitlz et al. [KLS18] and Devevey et al. [DFPS23].)
As Diemert et al. [DGJL21], we then modify the winning condition that $(m^*, (w^*, z^*))$ is not queried before with the condition that (w^*, z^*) is not equals to $(\tilde{w}^{(k)}, \tilde{z}^{(k)})$, a signature on m^* produced by the signing oracle (G_8). This modification allows the game to forget Q. We then continue minor but essential modifications involving the CUR property, which is ignored in Devevey et al. [DFPS23] or assumed to hold perfectly in Diemert et al. [DGJL21]. After those modifications, we arrive G_{11}, in which the winning condition is that (w^*, z^*) is not equal to $(\tilde{w}^{(k)}, \tilde{z}^{(k)})$, $w^* \neq \tilde{w}^{(k)}$, and $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$. Finally, we replace a normal verification key with a lossy verification key (G_{11}) as in [AFLT12, KLS18], and this replacement is justified key indistinguishability of LID. Finally, in G_{12}, the adversary wins with negligible probability as in Kiltz et al. [KLS18] due to ϵ_ℓ-lossiness.
The formal definitions of games follow.

Game G_0*:* This is the original game. See Figure 6 for a concrete definition of G_0, where we expand the Sign algorithm and H is defined as RF_{H}. By the definition, we have $\Pr[W_0] = \mathsf{Adv}^{\text{mseuf-cma1}}_{\mathsf{DS},\mathcal{A}}(\kappa)$.

G_i for $i \in \{0,1,2,3,4\}$

$(vk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$
$\mathsf{RF}_\mathsf{H} \leftarrow \mathsf{Func}(\mathcal{M} \times \mathcal{W}, C)$
$\mathsf{RF}_\mathsf{P} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], \mathcal{R}_{\mathsf{P}_1})$ / G_{2^-}
$Q := \emptyset$
win := false
run $\mathcal{A}^{\textsc{Sign},\textsc{Forge},|\mathsf{H}\rangle}(vk)$
return win

$\textsc{Sign}(m)$ for G_0

if $\exists (m, \sigma) \in Q$ for some σ **then return** σ
$(w, c, z) \leftarrow \textsc{GetTrans}(m)$
if $z = \perp$ **then** $\sigma := \perp$ **else** $\sigma := (w, z)$
$Q := Q \cup \{(m, \sigma)\}$
return σ

$\textsc{Sign}(m)$ for G_1–G_4

if $\exists (m, \sigma) \in Q$ for some σ **then return** σ / G_1
$\left\{\left(w^{(i)}, c^{(i)}, z^{(i)}\right)\right\}_{i\in[k]} \leftarrow \textsc{GetTrans}(m)$
if $z^{(k)} = \perp$ **then** $\sigma := \perp$ **else** $\sigma := \left(w^{(k)}, z^{(k)}\right)$
$Q := Q \cup \{(m, \sigma)\}$
return σ

H: $|m, w\rangle\,|y\rangle \mapsto |m, w\rangle\,|y \oplus c'\rangle$

$\left\{\left(w^{(i)}, c^{(i)}, z^{(i)}\right)\right\}_{i\in[k]} \leftarrow \textsc{GetTrans}(m)$ / G_4
if $\exists i : w = w^{(i)}$ **then return** $c' := c^{(i)}$ / G_4
return $c' := \mathsf{RF}_\mathsf{H}(m, w)$

$\textsc{GetTrans}(m)$

$k := 1; z^{(0)} := \perp$
while $z^{(k-1)} = \perp \wedge k \le B$:
  $(w^{(k)}, s) \leftarrow \mathsf{P}_1(sk)$ / G_1
  $(w^{(k)}, s) := \mathsf{P}_1(sk; \mathsf{RF}_\mathsf{P}(m, k))$ / G_{2^-}
  $c^{(k)} := \mathsf{RF}_\mathsf{H}(m, w^{(k)})$
  $z^{(k)} := \mathsf{P}_2(sk, w^{(k)}, c^{(k)}, s)$
  $k := k + 1$
$k := k - 1$ / cancel the last increment
return $(w^{(k)}, c^{(k)}, z^{(k)})$ / G_0
return $\left\{\left(w^{(i)}, c^{(i)}, z^{(i)}\right)\right\}_{i\in[k]}$ / G_{1^-}

$\textsc{Forge}(m^*, \sigma^*)$ where $\sigma^* = (w^*, z^*)$

$\left\{\left(\tilde{w}^{(i)}, \tilde{c}^{(i)}, \tilde{z}^{(i)}\right)\right\}_{i\in[k]} \leftarrow \textsc{GetTrans}(m^*)$ / G_{3^-}
if $\mathsf{V}(vk, \tilde{w}^{(k)}, \tilde{c}^{(k)}, \tilde{z}^{(k)}) = \mathsf{false}$ **then return** $\exists$ / G_{3^-}
$c^* := \mathsf{H}(m^*, w^*)$
if $\mathsf{V}(vk, w^*, c^*, z^*) = \mathsf{true}$ **then** / Vrfy passed
  if $(m^*, \sigma^*) \notin Q$ **then**
    win := true

Fig. 6. G_i for $i \in \{0,1,2,3,4\}$

G_i for $i \in \{4,5,6,7\}$

$(vk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$
$\mathsf{RF}_\mathsf{H} \leftarrow \mathsf{Func}(\mathcal{M} \times \mathcal{W}, C)$
$\mathsf{RF}'_\mathsf{H} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], C)$ / G_6–
$\mathsf{RF}_\mathsf{P} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], \mathcal{R}_{\mathsf{P}_1})$ / G_4–G_6
$\mathsf{RF}_\mathsf{S} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], \mathcal{R}_{\mathsf{Sim}})$ / G_7
$Q := \emptyset$
win := false
run $\mathcal{A}^{\textsc{Sign},\textsc{Forge},|\mathsf{H}\rangle}(vk)$
return win

$\textsc{Sign}(m)$

if $\textsc{GetTrans}(m) = \dashv$ **then return** $\dashv$ / G_5–
$\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i\in[k]} \leftarrow \textsc{GetTrans}(m)$
if $z^{(k)} = \bot$ **then** $\sigma := \bot$ **else** $\sigma := (w^{(k)}, z^{(k)})$
$Q := Q \cup \{(m, \sigma)\}$
return σ

$\mathsf{H}\colon |m, w\rangle |y\rangle \mapsto |m, w\rangle |y \oplus c'\rangle$

if $\textsc{GetTrans}(m) = \dashv$ **then return** $\dashv$ / G_5–
$\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i\in[k]} \leftarrow \textsc{GetTrans}(m)$
if $\exists i : w = w^{(i)}$ **then return** $c' := c^{(i)}$
return $c' := \mathsf{RF}_\mathsf{H}(m, w)$

$\textsc{GetTrans}(m)$

$k := 1; z^{(0)} := \bot$
while $z^{(k-1)} = \bot \wedge k \le B$:
  $(w^{(k)}, s) \leftarrow \mathsf{P}_1(sk; \mathsf{RF}_\mathsf{P}(m, k))$ / G_4–G_6
  $c^{(k)} := \mathsf{RF}_\mathsf{H}(m, w^{(k)})$ / G_4–G_5
  $c^{(k)} := \mathsf{RF}'_\mathsf{H}(m, k)$ / G_6–
  $z^{(k)} := \mathsf{P}_2(sk, w^{(k)}, c^{(k)}, s)$ / G_4–G_6
  $(w^{(k)}, z^{(k)}) := \mathsf{Sim}(vk, c^{(k)}; \mathsf{RF}_\mathsf{S}(m, k))$ / G_7
  $k := k + 1$
$k := k - 1$ / cancel the last increment
$\mathcal{L}_m := \{w^{(i)}\}_{i\in[k]}$ / G_5–
if $\mathsf{Coll}(\mathcal{L}_m)$ **then return** $\dashv$ / G_5–
return $\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i\in[k]}$

$\textsc{Forge}(m^*, \sigma^*)$ where $\sigma^* = (w^*, z^*)$

if $\textsc{GetTrans}(m) = \dashv$ **then return** $\dashv$ / G_5–
$\{(\tilde{w}^{(i)}, \tilde{c}^{(i)}, \tilde{z}^{(i)})\}_{i\in[k]} \leftarrow \textsc{GetTrans}(m^*)$
if $\mathsf{V}(vk, \tilde{w}^{(k)}, \tilde{c}^{(k)}, \tilde{z}^{(k)}) = \mathsf{false}$ **then return** $\dashv$
$c^* := \mathsf{H}(m^*, w^*)$
if $\mathsf{V}(vk, w^*, c^*, z^*) = \mathsf{true}$ **then** / Vrfy passed
  if $(m^*, \sigma^*) \notin Q$ **then**
    win := true

Fig. 7. G_i for $i \in \{4,5,6,7\}$

Game G_1*:* In this game, GetTrans outputs *all* transcripts instead of the last one. The signing oracle also takes the last one as a candidate for a signature. See G_1 in Figure 6 for the detail. Since this is a conceptual change, we have $\mathrm{G}_0 = \mathrm{G}_1$.

Game G_2*:* We next derandomize the prover in GetTrans as in L.4 of GetTrans in Figure 6. Since we consider *one-signature-per-one-message* situation, this derandomization by the random function RF_P changes nothing. We have $\mathrm{G}_1 = \mathrm{G}_2$.

Game G_3*:* We next modify the forge oracle as follows: On a query (m^*, σ^*), the oracle first comptues the transcripts $\{(\tilde{w}^{(i)}, \tilde{c}^{(i)}, \tilde{z}^{(i)})\}_{i\in[k]}$. If $(\tilde{w}^{(k)}, \tilde{z}^{(k)})$ is an *invalid* signature, then the oracle returns a special symbol $\dashv$ (see L.2 of Figure 6.) See G_3 in Figure 6 (and Figure 7) for the details.
We note that except ϵ_m probability of keys, this happens with probability at most $1 - \rho'$ as discussed in subsection 3.2. Notice that the adversary can obtain this information via the oracle Sign, which returns classical information. Thus, the difference between two games are upper-bounded by $(q_S + q_F)\rho'$ and we have

$$|\Pr[W_2] - \Pr[W_3]| \le (q_S + q_F)(1 - \rho') + \epsilon_m.$$

Game G_4*:* We next modify the random oracle as follows: On a query (m, w), the oracle first comptues the transcripts $\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i\in[k]}$. If the input w is equivalent to one of $w^{(i)}$, then it returns $c' := c^{(i)}$; otherwise, it returns $c' := \mathsf{RF}_\mathsf{H}(m, w)$. See G_4 in Figure 6 (and Figure 7) for the details.
Since $c^{(i)}$ is defined as $\mathsf{RF}_\mathsf{H}(m, w^{(i)})$ in GetTrans, this change nothing and we have $\mathrm{G}_3 = \mathrm{G}_4$.

Game G_5*:* We next introduce a collision check for $w^{(i)}$'s in GetTrans. If GetTrans finds a collision among $w^{(1)}, \ldots, w^{(k)}$, it outputs a special synbol $\dashv$. See G_5 in Figure 7.
For each message, the collision probability is at most $B^2 \cdot 2^{-\alpha-1}$ if $H_\infty(w)$ is α [DFPS23, Lem.11]. Using the one-sided O2H lemma [AHU19], Devevey et al. showed the following lemma, where we additionally introduce ϵ_m.

G_i for $i \in \{8, 9, 10, 11, 12\}$

$(vk, sk) \leftarrow \mathsf{Gen}_{\mathsf{LID}}(1^\kappa)$ / G_8–G_{11}
$vk \leftarrow \mathsf{LossyGen}_{\mathsf{LID}}(1^\kappa)$ / G_{12}
$\mathsf{RF}_{\mathsf{H}} \leftarrow \mathsf{Func}(\mathcal{M} \times \mathcal{W}, C)$
$\mathsf{RF}'_{\mathsf{H}} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], C)$
$\mathsf{RF}_{\mathsf{S}} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], \mathcal{R}_{\mathsf{Sim}})$
win $:=$ false
run $\mathcal{A}^{\langle \textsc{Sign}, \textsc{Forge}, |\mathsf{H}\rangle}(vk)$
return win

$\textsc{Forge}(m^*, \sigma^*)$ where $\sigma^* = (c^*, z^*)$

if $\textsc{GetTrans}(m^*) = \exists$ **then return** $\exists$
$\{(\tilde{w}^{(i)}, \tilde{c}^{(i)}, \tilde{z}^{(i)})\}_{i\in[k]} \leftarrow \textsc{GetTrans}(m^*)$
if $\exists i : w^* = \tilde{w}^{(i)}$ **then** $c^* := \tilde{c}^{(i)}$ **else** $c^* := \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$
if $\mathsf{V}(vk, \tilde{w}^{(k)}, \tilde{c}^{(k)}, \tilde{z}^{(k)}) = \mathsf{false}$ **then return** $\exists$
$\mathcal{L}_{m^*} := \{w^{(i)}\}_{i\in[k]}; \mathcal{L}'_{m^*} := \{w^{(i)}\}_{i\in[k-1]}$
if $\mathsf{V}(vk, w^*, c^*, z^*) = \mathsf{true} \wedge (w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$ **then**
  win $:=$ true / G_8
  if $(w^* \notin \mathcal{L}'_{m^*}) \vee (w^* \in \mathcal{L}'_{m^*} \wedge c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*))$ **then** / G_9
    win $:=$ true / G_9
  if $(w^* \notin \mathcal{L}_{m^*}) \vee (w^* \in \mathcal{L}'_{m^*} \wedge c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*))$ **then** / G_{10}
    win $:=$ true / G_{10}
  if $w^* \neq \tilde{w}^{(k)} \wedge c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ **then** / G_{11}–G_{12}
    win $:=$ true / G_{11}–G_{12}

Fig. 8. G_i for $i \in \{8, 9, 10, 11, 12\}$, where expand H in Forge. Sign, H, and GetTrans are those in G_7 in Figure 7.

Lemma 4.1. *Suppose that* LID *has* (α, ϵ_m)*-commitment min-entropy. Then, we have*

$$|\Pr[W_4] - \Pr[W_5]| \leq \delta_{4,5} := 2(q_S + q_H + q_F) \cdot B \cdot 2^{\frac{-\alpha-1}{2}} + \epsilon_m.$$

Game G_6*:* We next modify how to compute $c^{(k)}$ in GetTrans, in which it is computed as $\mathsf{RF}'_{\mathsf{H}}(m, k)$ instead of $\mathsf{RF}_{\mathsf{H}}(m, w^{(k)})$. We note that this does not change the adversary's view because $\mathsf{RF}'_{\mathsf{H}}$ is a random function, and if $w = w^{(i)}$ for the query (m, w), then consistent $c' = c^{(i)} = \mathsf{RF}'_{\mathsf{H}}(m, i)$ is output by H since we already exclude the collision. Thus, we have $\mathsf{G}_5 = \mathsf{G}_6$.

Game G_7*:* We next modify GetTrans to use the simulation algorithm. On a query m, the oracle comptues $c^{(i)} := \mathsf{RF}'_{\mathsf{H}}(m, i)$ and $(w^{(i)}, z^{(i)}) := \mathsf{Sim}(vk; \mathsf{RF}_{\mathsf{S}}(m))$. See G_7 in Figure 7 for the details. Since the real transcript and the simulated one is ϵ_{zk}-close, each invocation of GetTrans is $B\epsilon_{\mathrm{zk}}$-close. As Devevey et al. [DFPS23], we have the following lemma by using Lemma 2.1.

Lemma 4.2. *Suppose that* LID *is* ϵ_{zk}*-HVZK. Then, we have*

$$|\Pr[W_6] - \Pr[W_7]| \leq \delta_{6,7} := \sqrt{(6(q_S + q_H + q_F))^3 B\epsilon_{\mathrm{zk}}}.$$

Game G_8*:* We now separate from the approach by Devevey et al. [DFPS23] and remove the list Q as Diemert et al. [DGJL21]. In this game, we replace the condition $(m^*, \sigma^*) \notin Q$ with the condition $(w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$. See G_8 in Figure 8.

Lemma 4.3. *Suppose that that* LID *is* ϵ_{zk}*-HVZK and has* (α, ϵ_m)*-commit min-entropy. Then, we have*

$$|\Pr[W_7] - \Pr[W_8]| \leq \delta_{7,8} := q_F \cdot 2^{-\alpha} + \epsilon_m + \delta_{4,5} + \delta_{6,7}.$$

Proof. Suppose that the adversary queries m^* and $\sigma^* = (w^*, z^*)$ to the oracle Forge.
If m^* is queried to the signing oracle before, then there is no difference between the two games: The signing oracle returns $(\tilde{w}^{(k)}, \tilde{z}^{(k)})$ as a signature on m^* and, thus, the condition $(m^*, \sigma^*) \notin Q$ and the condition $(w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$ are equivalent.
If m^* is not queried to the signing oracle, then the two games might differ if the adversary queries m^* and $(w^*, z^*) = (\tilde{w}^{(k)}, \tilde{z}^{(k)})$, which implies $w^* = \tilde{w}^{(k)}$. This means that the adversary succeeds to guess $\tilde{w}^{(k)}$ on m^* without knowing it. Let Bad_i denote the event that m^* is not queried before but w^* equals to $\tilde{w}^{(k)}$ happens in G_i. We have

$$|\Pr[W_7] - \Pr[W_8]| \leq \Pr[\mathsf{Bad}_7] \leq |\Pr[\mathsf{Bad}_7] - \Pr[\mathsf{Bad}_3]| + \Pr[\mathsf{Bad}_3].$$

We have $\Pr[\mathsf{Bad}_3] \leq q_F \cdot 2^{-\alpha} + \epsilon_m$. This is because H does not reveal any information on $\tilde{w}^{(k)}$ and $\tilde{w}^{(k)}$ has min-entropy α with probability $1 - \epsilon_m$ over choice of (vk, sk). Since $|\Pr[\mathsf{Bad}_7] - \Pr[\mathsf{Bad}_3]|$ is upper-bounded by $\delta_{4,5} + \delta_{6,7}$ as Lemma 4.1 and Lemma 4.2, we obtain the bound. □

Game G_9*:* To ease the notation, let $\mathcal{L}_{m^*} := \{w^{(i)}\}_{i\in[k]}$ which is the w parts of the transcripts generated by GetTrans(m^*). We additionally define $\mathcal{L}'_{m^*} := \{w^{(i)}\}_{i\in[k-1]}$.
In G_8, Forge additionally checks if $w^* \in \mathcal{L}'_{m^*}$ or not; if so, we additionally checks whether $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ or not. See G_9 in Figure 8 for the details.
We have the following lemma.

Lemma 4.4. *We have* $|\Pr[W_8] - \Pr[W_9]| \le \delta_{4,5} + \delta_{6,7} + \delta_{7,8}$.

Proof. The two games differ if the adversary queries $w^* = w^{(i)}$ for $i < k$ but $c^* \neq \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$. We call this event in G_i Bad_i. We have

$$|\Pr[W_8] - \Pr[W_9]| \le \Pr[\mathrm{Bad}_8] \le |\Pr[\mathrm{Bad}_8] - \Pr[\mathrm{Bad}_3]| + \Pr[\mathrm{Bad}_3].$$

We have $\Pr[\mathrm{Bad}_3] = 0$ because $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ always holds in G_3. Since $|\Pr[\mathrm{Bad}_8] - \Pr[\mathrm{Bad}_3]|$ is upper-bounded by $\delta_{4,5} + \delta_{6,7} + \delta_{7,8}$, we obtain the bound. □

Game G_{10}*:* We next treat the case $w^* = \tilde{w}^{(k)}$ as a special case. To do so, we replace the condition $w^* \notin \mathcal{L}'_{m^*}$ with $w^* \notin \mathcal{L}_{m^*}$. See G_{10} in Figure 8 for the details.
Because of this modification, if the adversary queries $(m^*, (w^*, z^*))$ satisfying $(w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$, $w^* = \tilde{w}^{(k)}$, and $\mathsf{Vrfy}(vk, w^*, c^*, z^*) = \mathsf{true}$, then two games may differ. Fortunately, this event is easily treated by the CUR property.

Lemma 4.5. *There exists a quantum* $\mathcal{F}$*-oracle adversary* $\mathcal{A}_{\mathrm{cur}}$ *such that*

$$\begin{aligned}
&|\Pr[W_9] - \Pr[W_{10}]| \le \mathsf{Adv}^{\mathrm{cur}}_{\mathsf{LID},\mathcal{A}_{\mathrm{cur}}}(\kappa),\\
&\mathsf{Time}^*(\mathcal{A}_{\mathrm{cur}}) = \mathsf{Time}(\mathcal{A}) + (q_H + q_S + q_F)\cdot O(B\mathsf{Time}(\mathsf{LID}) + B^2),\\
&\mathsf{Mem}^*(\mathcal{A}_{\mathrm{cur}}) = \mathsf{Mem}(\mathcal{A}) + O(B\mathsf{Mem}(\mathsf{LID})),
\end{aligned}$$

where $\mathcal{F} = \mathsf{Func}(\mathcal{M}\times\mathcal{W}, C)\times\mathsf{Func}(\mathcal{M}\times[B], C)\times\mathsf{Func}(\mathcal{M}\times[B], \mathcal{R}_{\mathsf{Sim}})$.

Proof. Suppose that the queried forgery is (m^*, σ^*) with $\sigma^* = (w^*, z^*)$. Consider the computation in Forge and assume that $(w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$ and $\mathsf{V}(vk, w^*, c^*, z^*) = \mathsf{true}$ for $c^* = \mathsf{H}(m^*, w^*)$ and the flag win is set true in G_9. We have the following two cases to analyze G_{10}:

- If $w^* \neq \tilde{w}^{(k)}$, then there are no differences because of the collision checks and the flag win is set true in G_{10} also.
- If $w^* = \tilde{w}^{(k)}$, then $c^* := \tilde{c}^{(k)}$ as $\mathsf{H}(m^*, w^*)$ in L.3 of Forge. Since $w^* = \tilde{w}^{(k)}$, the flag win is unchanged in G_{10}. However, the check $(w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$ forces $z^* \neq \tilde{z}^{(k)}$, and this z^* leads to break the CUR property by outputting $(w^*, c^*, z^*, \tilde{z}^{(k)})$.

Upon the above observation, we can construct a quantum $\mathcal{F}$-oracle adversary $\mathcal{A}_{\mathrm{cur}}$ straightforwardly. The analysis of advantage, running time, and memory usage in the lemma are straightforwardly obtained. □

Remark 4.1. We note that $\tilde{z}^{(k)} \neq \perp$ holds by the check we introduced in G_3 into Forge. This check is fatal for the above proof because, if $\tilde{z}^{(k)} = \perp$, then the reduction algorithm fails to output the collision $(w^*, c^*, z^*, \tilde{z}^{(k)})$ breaking the CUR property. Kiltz et al. [KLS18] ignored the case that m^* is queried to the signing oracle, but the signing oracle fails on m^*.

Game G_{11}*:* We again modify the conditions in Forge in G_{10}: Forge checks if $\mathsf{V}(vk, w^*, c^*, z^*) = \mathsf{true}$ for $c^* = \mathsf{H}(m^*, w^*)$, $(w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$, $w^* \neq \tilde{w}^{(k)}$, and $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ or not. If so, the flag is set as true. See G_{11} in Figure 8 for the details.

Lemma 4.6. *We have* $\mathrm{G}_{10} = \mathrm{G}_{11}$.

Proof. Let us consider a forgery $(m^*, (w^*, z^*))$ satisfying $\mathsf{V}(vk, w^*, c^*, z^*) = \mathsf{true}$ for $c^* = \mathsf{H}(m^*, w^*)$ and $(w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$. Let us consider three cases: 1) If $w^* \in \mathcal{L}'_{m^*}$, then there is no diference on the condition $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ in both games. 2) If $w^* = \tilde{w}^{(k)}$, then win is kept in both games. 3) If $w^* \notin \mathcal{L}_{m^*}$, then Forge sets win as true immediately in G_{10} but sets win as true if $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ in G_{11}. We note that $c^* := \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ by L.4 of Forge. Thus, Forge in G_{11} also sets win := true and there are no differences. Thus, both games are the same. □

Game G_{12}*:* We finally replace a normal verification key with a lossy verification key. See G_{12} in Figure 8 for the details.

Lemma 4.7. *There exists a quantum* $\mathcal{F}$*-oracle adversary* $\mathcal{A}_{\mathrm{ind}}$ *such that*

$$\begin{aligned}
&|\Pr[W_{11}] - \Pr[W_{12}]| \le \mathsf{Adv}^{\mathrm{indkey}}_{\mathsf{LID},\mathcal{A}_{\mathrm{ind}}}(\kappa),\\
&\mathsf{Time}^*(\mathcal{A}_{\mathrm{ind}}) = \mathsf{Time}(\mathcal{A}) + (q_H + q_S + q_F)\cdot O(B\mathsf{Time}(\mathsf{LID}) + B^2),\\
&\mathsf{Mem}^*(\mathcal{A}_{\mathrm{ind}}) = \mathsf{Mem}(\mathcal{A}) + O(B\mathsf{Mem}(\mathsf{LID})),
\end{aligned}$$

where $\mathcal{F} = \mathsf{Func}(\mathcal{M}\times\mathcal{W}, C)\times\mathsf{Func}(\mathcal{M}\times[B], C)\times\mathsf{Func}(\mathcal{M}\times[B], \mathcal{R}_{\mathsf{Sim}})$.

Proof. We construct $\mathcal{A}_{\mathrm{ind}}$ straightforwardly. The analysis of advantage, running time, and memory usage in the lemma are straightforwardly obtained. □

Lemma 4.8. *Suppose that* LID *is* ϵ_ℓ*-lossy. Then, we have* $\Pr[W_{12}] \le 8(q_H + q_S + q_F + 1)^2\epsilon_\ell$.

Since the proof is the same as that in Kiltz et al. [KLS18], we omit it. See the proof of the case for sBU security (Lemma D.10).

5 Plus-One Unforgeability of Signature from Lossy Identification

Theorem 5.1 (PO security of $\mathsf{DFS}_{B,\mathsf{wz}}[\mathsf{LID},\mathsf{H},\mathsf{PRF}]$**).** *Let* $B \ge 0$. *Let* $\mathsf{H}\colon \mathcal{M}\times\mathcal{W}\to C$ *be a hash function modeled as a random oracle. Let* LID *be a lossy identification scheme that is* (γ,β)*-complete,* ϵ_{zk}*-HVZK, and* ϵ_ℓ*-lossy, and has* (α,ϵ_m)*-commitment min-entropy. Let* $\mathsf{DS} := \mathsf{DFS}_{B,\mathsf{wz}}[\mathsf{LID},\mathsf{H},\mathsf{PRF}]$.
Then, for a quantum adversary $\mathcal{A}$ *breaking the* PO *security of* DS *that issues at most* q_H *quantum queries to the random oracle* H, q_S *classical queries to the signing oracle, and* q_F *classical queries to the forgery oracle, there exists a quantum* $\mathcal{F}_{\mathrm{prf}}$*-oracle adversary* $\mathcal{A}_{\mathrm{prf}}$ *against pseudorandomness of* PRF *and quantum* $\mathcal{F}$*-oracle adversaries* $\mathcal{A}_{\mathrm{ind}}$ *against key indistinguishability of* LID *and* $\mathcal{A}_{\mathrm{cur}}$ *against computationally unique response of* LID *such that*

$$\begin{aligned}
\mathsf{Adv}^{\mathrm{po}}_{\mathsf{DS},\mathcal{A}}(\kappa) &\le \mathsf{Adv}^{\mathrm{pr}}_{\mathsf{PRF},\mathcal{A}_{\mathrm{prf}}}(\kappa) + \mathsf{Adv}^{\mathrm{cur}}_{\mathsf{LID},\mathcal{A}_{\mathrm{cur}}}(\kappa) + \mathsf{Adv}^{\text{ind-key}}_{\mathsf{LID},\mathcal{A}_{\mathrm{ind}}}(\kappa) + 8(q+1)^2(1-\rho')\\
&\quad + 8qB2^{-\frac{-\alpha-1}{2}} + 7\epsilon_m + 4\sqrt{(6q)^3B\epsilon_{\mathrm{zk}}} + 2(q_S+1)/\lfloor 2^\alpha\rfloor + 8(q+1)^2\epsilon_\ell,\\
\mathsf{Time}^*(\mathcal{A}_{\mathrm{prf}}) &= \mathsf{Time}(\mathcal{A}) + q_S\cdot O(B\mathsf{Time}(\mathsf{LID})) + q_F\cdot O(\mathsf{Time}(\mathsf{LID})),\\
\mathsf{Mem}^*(\mathcal{A}_{\mathrm{prf}}) &= \mathsf{Mem}(\mathcal{A}) + O(B\mathsf{Mem}(\mathsf{LID})) + q_F\cdot O(\mathsf{Mem}(\mathsf{LID})) + O(\lg(q_S)),\\
\mathsf{Time}^*(\mathcal{A}_{\mathrm{ind}}) &= \mathsf{Time}(\mathcal{A}) + q\cdot O(B\mathsf{Time}(\mathsf{LID}) + B^2),\\
\mathsf{Mem}^*(\mathcal{A}_{\mathrm{ind}}) &= \mathsf{Mem}(\mathcal{A}) + O(B\mathsf{Mem}(\mathsf{LID})),\\
\mathsf{Time}^*(\mathcal{A}_{\mathrm{cur}}) &= \mathsf{Time}(\mathcal{A}) + q\cdot O(B\mathsf{Time}(\mathsf{LID}) + B^2),\\
\mathsf{Mem}^*(\mathcal{A}_{\mathrm{cur}}) &= \mathsf{Mem}(\mathcal{A}) + O(B\mathsf{Mem}(\mathsf{LID})),
\end{aligned}$$

where $q = q_H + q_S + q_F$, ρ' *is defined as in subsection 3.2,* $\mathcal{F}_{\mathrm{prf}} = \mathsf{Func}(\mathcal{M}\times\mathcal{W}, C)$, *and* $\mathcal{F} = \mathsf{Func}(\mathcal{M}\times\mathcal{W}, C)\times\mathsf{Func}(\mathcal{M}\times[B], C)\times\mathsf{Func}(\mathcal{M}\times[B], \mathcal{R}_{\mathsf{Sim}})$.

As a corollary, when we employ a random function RF_{P} directly, the above proof can be modified into a memory-tight one.

Corollary 5.1 (PO security of $\mathsf{DFS}^+{}_{B,\mathsf{wz}}[\mathsf{LID},\mathsf{H},\mathsf{RF}_{\mathsf{P}}]$**).** $\mathsf{DFS}^+{}_{B,wz}[\mathsf{LID},\mathsf{H},\mathsf{RF}_{\mathsf{P}}]$ *has a memory-tight proof for the* PO *security.*

5.1 Proof of Theorem

We define fourteen games G_i for $i\in\{0,1,\ldots,13\}$ to show our theorem. Let W_i denote the event that the experiment outputs true in G_i.

G_i for $i \in \{0,1,2,3,4\}$

$(vk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$
$K \leftarrow \{0,1\}^\kappa$ / G_0
$\mathsf{RF}_\mathsf{P} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], \mathcal{R}_{\mathsf{P}_1})$ / G_1–
$\mathsf{RF}_\mathsf{H} \leftarrow \mathsf{Func}(\mathcal{M} \times \mathcal{W}, C)$
$Q := \emptyset$
run $\mathcal{A}^{|\textsc{Sign}\rangle, \textsc{Forge}, |\mathsf{H}\rangle}(vk)$
return $[\![\#Q > q_S]\!]$

SIGN: $|m\rangle\,|y\rangle \mapsto |m\rangle\,|m \oplus \sigma\rangle$

$(w, c, z) \leftarrow \textsc{GetTrans}(m)$ / G_0–G_1
$\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i\in[k]} \leftarrow \textsc{GetTrans}(m)$ / G_2–
$(w, z) := (w^{(i)}, z^{(i)})$ / G_2–
if $z = \bot$ **then** $\sigma := \bot$ **else** $\sigma := (w, z)$
return σ

H: $|m, w\rangle\,|y\rangle \mapsto |m, w\rangle\,|y \oplus c'\rangle$

$\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i\in[k]} \leftarrow \textsc{GetTrans}(m)$ / G_4
if $\exists i : w = w^{(i)}$ **then return** $c' := c^{(i)}$ / G_4
return $c' := \mathsf{RF}_\mathsf{H}(m, w)$

GETTRANS(m)

$k := 1; z^{(0)} := \bot$
while $z^{(k-1)} = \bot \wedge k \le B$:
  $(w^{(k)}, s) := \mathsf{P}_1(sk; \mathsf{PRF}(K, (m, k)))$ / G_0–
  $(w^{(k)}, s) := \mathsf{P}_1(sk; \mathsf{RF}_\mathsf{P}(m, k))$ / G_1–
  $c^{(k)} := \mathsf{RF}_\mathsf{H}(m, w^{(k)})$
  $z^{(k)} := \mathsf{P}_2(sk, w^{(k)}, c^{(k)}, s)$
  $k := k + 1$
$k := k - 1$ / cancel the last increment
return $(w, c, z) = (w^{(k)}, c^{(k)}, z^{(k)})$ / G_0–G_1
return $\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i\in[k]}$ / G_2–

FORGE(m^*, σ^*) where $\sigma^* = (w^*, z^*)$

$\{(\tilde{w}^{(i)}, \tilde{c}^{(i)}, \tilde{z}^{(i)})\}_{i\in[k]} \leftarrow \textsc{GetTrans}(m^*)$ / G_3–
if $\mathsf{V}(vk, \tilde{w}^{(k)}, \tilde{c}^{(k)}, \tilde{z}^{(k)}) = \mathsf{false}$ **then return** $\exists$ / G_3–
$c^* := \mathsf{H}(m^*, w^*)$
if $\mathsf{V}(vk, w^*, c^*, z^*) = \mathsf{true}$ **then** / Vrfy passed
  if $(m^*, (w^*, z^*)) \notin Q$ **then**
    $Q := Q \cup \{(m^*, (w^*, z^*))\}$

Fig. 9. G_i for $i \in \{0,1,2,3,4\}$

Roadmap: Before describing games, we briefly give intuitions for games. First, we follow the proof by Devevey et al. [DFPS23]. We consider the original game (G_0), in which the signing oracle queried on m calls $\textsc{GetTrans}(m)$ internally and use this real transcript as a signature. The oracle FORGE manages the valid forgery set Q, and the adversary wins if the cardinality of Q is larger than the number of queries to the quantum signing oracle, q_S. (See G_0 in Figure 9.) We then replace PRF with RF. This is the only non-memory-tight part that can be omitted if we derandomize the signature scheme by RF. After that, we modify the games routinely as in the msEUF-CMA1 security proof and arrive at G_7, in which $\textsc{GetTrans}(m)$ is implemented by the simulator.

We want to modify the winning condition with that $(m^*, (w^*, z^*))$ is not queried before with the condition that (w^*, z^*) is not equal to $(\tilde{w}^{(k)}, \tilde{z}^{(k)})$, a signature on m^* produced by the signing oracle since this allows us to forget Q.

To do so, we introduce the flag win (G_8) as in the proofs of msEUF-CMA1 security and sBU security, which is set true when the adversary queries a valid signature the SIGN does not compute, and the challenger outputs true if the adversary outputs at least q_S+1 distinct valid pairs of message/signature and the flag win is true. If a difference occurs between two games, then the adversary correctly guesses at least q_S+1 signatures produced by GETTRANS on *distinct* q_S+1 messages. This means that the adversary correctly q_S+1 commitments w on distinct q_S+1 messages. This chance should be negligible by the commitment min-entropy and Lemma 2.2.

Before forgetting Q, we introduce the check $c^* = \mathsf{RF}_\mathsf{H}(m^*, w^*)$ for the case $w^* = \tilde{w}^{(i)}$ for some $i < k$ (G_9) because of a thechnical reason. We then remove the condition $\#Q > q_S$ in G_{10}, and we have $\Pr[W_9] \le \Pr[W_{10}]$ since this is just *relaxation* of the winning condition and cannot be detected by the adversary. Those technical games are the core of our PO-security proof.

We then continue the modification and arrive at G_{12}, in which the winning condition is that (w^*, z^*) is not equal to $(\tilde{w}^{(k)}, \tilde{z}^{(k)})$, $w^* \neq \tilde{w}^{(k)}$, and $c^* = \mathsf{RF}_\mathsf{H}(m^*, w^*)$. Finally, we replace a normal verification key with a lossy verification key (G_{13}) as in [AFLT12, KLS18], and this replacement is justified key indistinguishability of LID. Finally, in G_{13}, the adversary wins with negligible probability as in Kiltz et al. [KLS18].

The formal definitions of games follow.

Game G_0*:* This is the original game. See Figure 9 for a concrete definition of G_0, where we expand the Sign algorithm. We have $\Pr[W_0] = \mathsf{Adv}^{\mathrm{po}}_{\mathrm{DS},\mathcal{A}}(\kappa)$.

Game G_1*:* We then replace PRF in L.3 of GetTrans with RF_P. The straightforward argument shows the following lemma. Unfortunately, this part is memory-loose because the reduction algorithm should maintain the forgery list Q.

Lemma 5.1. *There exists a quantum $\mathcal{F}$-oracle adversary $\mathcal{A}_{\mathrm{prf}}$ such that*

$$|\Pr[W_0] - \Pr[W_1]| \le \mathsf{Adv}^{\mathrm{pr}}_{\mathsf{PRF},\mathcal{A}_{\mathrm{prf}}}(\kappa),$$
$$\mathsf{Time}^*(\mathcal{A}_{\mathsf{PRF}}) = \mathsf{Time}(\mathcal{A}) + q_S \cdot O(B\mathsf{Time}(\mathsf{LID})) + q_F \cdot O(\mathsf{Time}(\mathsf{LID}))$$
$$\mathsf{Mem}^*(\mathcal{A}_{\mathsf{PRF}}) = \mathsf{Mem}(\mathcal{A}) + O(B\mathsf{Mem}(\mathsf{LID})) + q_F \cdot O(\mathsf{Mem}(\mathsf{LID})) + O(\lg(q_S)),$$

where $\mathcal{F} = \mathsf{Func}(\mathcal{M} \times \mathcal{W}, C)$.

Game G_2*:* We next make GetTrans output all transcripts instead of the last one. This modification does not change anything, and we have $\mathrm{G}_1 = \mathrm{G}_2$.

Game G_3*:* We next modify the forge oracle as follows: Before checking the validity of submitted query (m^*, σ^*), it generates its own signature $(\tilde{w}^{(k)}, \tilde{c}^{(k)}, \tilde{z}^{(k)})$ by using $\textsc{GetTrans}(m^*)$. If it fails, that is, $\mathsf{V}(vk, \tilde{w}^{(k)}, \tilde{c}^{(k)}, \tilde{z}^{(k)}) =$ false, then the forge oracle returns the special symbol $\perp$.
The adversary differentiates between the two games G_2 and G_3 if it submits such (m^*, σ^*) on which $\textsc{GetTrans}(m^*)$ fails to output a valid transcript in its B tries. We can connect this event to the generic search problem with $\lambda = 1 - \rho'$ except for the probability at most ϵ_m over keys. We here skip the proof and see the proof of sBU security (Lemma D.2) for the detail.

Lemma 5.2. *Suppose that* LID *is (γ, β)-correct and has (α, ϵ_m)-commitment min-entropy. We have*

$$|\Pr[W_2] - \Pr[W_3]| \le \Pr[\mathsf{Bad}_{m^*}] \le 8(q_S + q_F + q_H + 1)^2(1 - \rho') + \epsilon_m.$$

Game G_4*:* We next modify the random oracle as follows: On a query (m, w), the oracle first computes the transcripts $\{(w^{(i)}, c^{(i)}, z^{(i)})\}$ via GetTrans. If the input w is equivalent to one of $w^{(i)}$, then it returns $c' := c^{(i)}$; otherwise, it returns $c' := \mathsf{RF}_\mathsf{H}(m, w)$. See G_4 in Figure 9 and Figure 10 for the details. Since $c^{(i)}$ is computed as $\mathsf{RF}_\mathsf{H}(m, w^{(i)})$, this modification changes nothing and we have $\mathrm{G}_3 = \mathrm{G}_4$.

Game G_5*:* The next game introduces a collision check for $w^{(i)}$'s in GetTrans. As Lemma 4.1, we have the following lemma.

Lemma 5.3. *Suppose that* LID *has (α, ϵ_m)-commitment min-entropy. Then, we have*

$$|\Pr[W_4] - \Pr[W_5]| \le \delta_{4,5} := 2(q_S + q_H + q_F) \cdot B \cdot 2^{\frac{-\alpha-1}{2}} + \epsilon_m.$$

Game G_6*:* We next modify how to compute $c^{(k)}$ in GetTrans, in which it is computed as $\mathsf{RF}'_\mathsf{H}(m, k)$ instead of $\mathsf{RF}_\mathsf{H}(m, w^{(k)})$. We note that this does not change the adversary's view because RF'_H is a random function, and if $w = w^{(i)}$ for the query (m, w), then consistent $c' = c^{(i)} = \mathsf{RF}'_\mathsf{H}(m, i)$ is output by H. (Note that excluding the collision is crucial [DFPS23].) We have $\mathrm{G}_5 = \mathrm{G}_6$.

Game G_7*:* We next modify the signing oracle Sign to use Sim instead of P_1 and P_2. See G_7 in Figure 10 for the details. As Lemma 4.2, we have the following lemma:

Lemma 5.4. *Assume that* LID *is ϵ_{zk}-HVZK. Then, we have*

$$|\Pr[W_6] - \Pr[W_7]| \le \delta_{6,7} := \sqrt{(6(q_S + q_H + q_F))^3 B \epsilon_{\mathrm{zk}}}.$$

Game G_8*:* We replace the winning condition of $\mathcal{A}$ as follows: We introduce a flag win which is set true by Forge when the adversary queries $(w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$. The challenger outputs $[\![\#Q > \mathsf{cnt}_s]\!] \wedge \mathsf{win}$. See G_8 in Figure 11.

Lemma 5.5. *Suppose that* LID *has (α, ϵ_m)-commitment min-entropy and is ϵ_{zk}-HVZK. Then, we have*

$$|\Pr[W_7] - \Pr[W_8]| \le \delta_{7,8} := (q_S + 1)/\lfloor 2^\alpha \rfloor + \epsilon_m + \delta_{4,5} + \delta_{6,7}.$$

Proof. The two games differ if the adversary queries at least $(q_S + 1)$ valid signatures on *distinct* messages to Forge such that $(w^*, z^*) = (\tilde{w}^{(k)}, \tilde{z}^{(k)})$ on each m^*. This is because if two valid signatures share the same message, then two signatures should be equivalent.
Let Bad_i be the event that the adversary in G_i queries $(q_S + 1)$ valid signatures on *distinct* messages to Forge such that $w^* = \tilde{w}^{(k)}$ on each m^*. By routine calculation, we have

$$|\Pr[W_7] - \Pr[W_8]| \le \Pr[\mathsf{Bad}_7] \le |\Pr[\mathsf{Bad}_7] - \Pr[\mathsf{Bad}_3]| + \Pr[\mathsf{Bad}_3].$$

Since there is no leak of $\tilde{w}^{(k)}$ on m^* from the random oracle H in G_3, we have $\Pr[\mathsf{Bad}_3] \le (q_S + 1)/\lfloor 2^\alpha \rfloor + \epsilon_m$ by invoking Lemma 2.2 and the min-entropy of the commitment. Since $|\Pr[\mathsf{Bad}_7] - \Pr[\mathsf{Bad}_3]|$ is upper-bounded by $\delta_{4,5} + \delta_{6,7}$ via Lemma 5.3 and Lemma 5.4, we obtain the bound. □

G_i for $i \in \{4,5,6,7\}$

$(vk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$
$\mathsf{RF}_\mathsf{H} \leftarrow \mathsf{Func}(\mathcal{M} \times \mathcal{W}, C)$
$\mathsf{RF}'_\mathsf{H} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], C)$ / G_6–
$\mathsf{RF}_\mathsf{P} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], \mathcal{R}_{\mathsf{P}_1})$ / G_4–G_6
$\mathsf{RF}_\mathsf{S} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], \mathcal{R}_{\mathsf{Sim}})$ / G_7
$Q := \emptyset$
run $\mathcal{A}^{\langle \textsc{Sign}\rangle, \textsc{Forge}, \langle \mathsf{H}\rangle}(vk)$
return $[\![\#Q > q_S]\!]$

$\textsc{Sign}\colon |m\rangle|y\rangle \mapsto |m\rangle|m \oplus \sigma\rangle$

if $\textsc{GetTrans}(m) = \dashv$ **then return** $\dashv$ / G_5–
$\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i\in[k]} \leftarrow \textsc{GetTrans}(m)$
if $z^{(k)} = \bot$ **then** $\sigma := \bot$ **else** $\sigma := (w^{(k)}, z^{(k)})$
return σ

$\mathsf{H}\colon |m, w\rangle|y\rangle \mapsto |m, w\rangle|y \oplus c'\rangle$

if $\textsc{GetTrans}(m) = \dashv$ **then return** $\dashv$ / G_5–
$\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i\in[k]} \leftarrow \textsc{GetTrans}(m)$
if $\exists i : w = w^{(i)}$ **then return** $c' := c^{(i)}$
return $c' := \mathsf{RF}_\mathsf{H}(m, w)$

$\textsc{GetTrans}(m)$

$k := 1; z^{(0)} := \bot$
while $z^{(k-1)} = \bot \wedge k \le B$:
  $(w^{(k)}, s) \leftarrow \mathsf{P}_1(sk; \mathsf{RF}_\mathsf{P}(m, k))$ / G_4–G_6
  $c^{(k)} := \mathsf{RF}_\mathsf{H}(m, w^{(k)})$ / G_4–G_5
  $c^{(k)} := \mathsf{RF}'_\mathsf{H}(m, k)$ / G_6–
  $z^{(k)} := \mathsf{P}_2(sk, w^{(k)}, c^{(k)}, s)$ / G_4–G_6
  $(w^{(k)}, z^{(k)}) := \mathsf{Sim}(vk, c^{(k)}; \mathsf{RF}_\mathsf{S}(m, k))$ / G_7
  $k := k + 1$
$k := k - 1$ / cancel the last increment
$\mathcal{L}_m := \{w^{(i)}\}_{i\in[k]}$ / G_5–
if $\mathsf{Coll}(\mathcal{L}_m)$ **then return** $\dashv$ / G_5–
return $\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i\in[k]}$

$\textsc{Forge}(m^*, \sigma^*)$ where $\sigma^* = (w^*, z^*)$

$\{(\tilde{w}^{(i)}, \tilde{c}^{(i)}, \tilde{z}^{(i)})\}_{i\in[k]} \leftarrow \textsc{GetTrans}(m^*)$
if $\mathsf{V}(vk, \tilde{w}^{(k)}, \tilde{c}^{(k)}, \tilde{z}^{(k)}) = \mathsf{false}$ **then return** $\dashv$
$c^* := \mathsf{H}(m^*, w^*)$
if $\mathsf{V}(vk, w^*, c^*, z^*) = \mathsf{true}$ **then** / Vrfy passed
  if $(m^*, (w^*, z^*)) \notin Q$ **then**
    $Q := Q \cup \{(m^*, (w^*, z^*))\}$

Fig. 10. G_i for $i \in \{4,5,6,7\}$

G_i for $i \in \{7,8,9,10\}$

$(vk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$
$\mathsf{RF}_\mathsf{H} \leftarrow \mathsf{Func}(\mathcal{M} \times \mathcal{W}, C)$
$\mathsf{RF}'_\mathsf{H} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], C)$
$\mathsf{RF}_\mathsf{S} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], \mathcal{R}_{\mathsf{Sim}})$
$Q := \emptyset$ / G_7–G_9
win := false / G_8–
run $\mathcal{A}^{\langle \textsc{Sign}\rangle, \textsc{Forge}, \langle \mathsf{H}\rangle}(vk)$
return $[\![\#Q > q_S]\!]$ / G_7
return $[\![\#Q > q_S]\!] \wedge \mathsf{win}$ / G_8–G_9
return win / G_{10}

$\textsc{Forge}(m^*, \sigma^*)$ where $\sigma^* = (w^*, z^*)$

if $\textsc{GetTrans}(m^*) = \dashv$ **then return** $\dashv$
$\{(\tilde{w}^{(i)}, \tilde{c}^{(i)}, \tilde{z}^{(i)})\}_{i\in[k]} \leftarrow \textsc{GetTrans}(m^*)$
if $\mathsf{V}(vk, \tilde{w}^{(k)}, \tilde{c}^{(k)}, \tilde{z}^{(k)}) = \mathsf{false}$ **then return** $\dashv$
$\mathcal{L}_{m^*} := \{w^{(i)}\}_{i\in[k]}; \mathcal{L}'_{m^*} := \{w^{(i)}\}_{i\in[k-1]}$ / G_9–
if $\exists i : w^* = \tilde{w}^{(i)}$ **then** $c^* := \tilde{c}^{(i)}$ **else** $c^* := \mathsf{RF}_\mathsf{H}(m^*, w^*)$
if $\mathsf{V}(vk, w^*, c^*, z^*) = \mathsf{true}$ **then** / Vrfy passed
  if $(m^*, (w^*, z^*)) \notin Q$ **then** / G_7–G_9
    $Q := Q \cup \{(m^*, (w^*, z^*))\}$ / G_7–G_9
  if $(w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$ **then** / G_8–G_{10}
    win := true / G_8
    if $(w^* \notin \mathcal{L}'_{m^*})$ / G_9–G_{10}
     or $(w^* \in \mathcal{L}'_{m^*} \wedge c^* = \mathsf{RF}_\mathsf{H}(m^*, w^*))$ **then** / G_9–G_{10}
      win := true / G_9–G_{10}

Fig. 11. G_i for $i \in \{7,8,9,10\}$, where we expand H in $\textsc{Forge}$. $\textsc{Sign}$, H, and $\textsc{GetTrans}$ are those in G_7 in Figure 10

G_i for $i \in \{10, 11, 12, 13\}$	$\textsc{Forge}(m^*, \sigma^*)$ where $\sigma^* = (c^*, z^*)$
$(vk, sk) \leftarrow \mathsf{Gen}_{\mathsf{LID}}(1^\kappa)$ / G_{10}–G_{12}	$w^* := \mathsf{Rec}(vk, c^*, z^*)$
$vk \leftarrow \mathsf{LossyGen}_{\mathsf{LID}}(1^\kappa)$ / G_{13}	**if** $\textsc{GetTrans}(m^*) = \dashv$ **then return** $\dashv$
$\mathsf{RF}_{\mathsf{H}} \leftarrow \mathsf{Func}(\mathcal{M} \times \mathcal{W}, C)$	$\{(\tilde{w}^{(i)}, \tilde{c}^{(i)}, \tilde{z}^{(i)})\}_{i \in [k]} \leftarrow \textsc{GetTrans}(m^*)$
$\mathsf{RF}'_{\mathsf{H}} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], C)$	**if** $\mathsf{V}(vk, \tilde{w}^{(k)}, \tilde{c}^{(k)}, \tilde{z}^{(k)}) = \mathsf{false}$ **then return** $\dashv$
$\mathsf{RF}_{\mathsf{S}} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], \mathcal{R})$	$\mathcal{L}_{m^*} := \{w^{(i)}\}_{i \in [k]}; \mathcal{L}'_{m^*} := \{w^{(i)}\}_{i \in [k-1]}$
$\mathsf{win} := \mathsf{false}$	**if** $\exists i : w^* = \tilde{w}^{(i)}$ **then** $c^* := \tilde{c}^{(i)}$ **else** $c^* := \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$
run $\mathcal{A}^{\lvert\textsc{Sign}\rangle, \textsc{Forge}, \lvert\mathsf{H}\rangle}(vk)$	**if** $\mathsf{V}(vk, w^*, c^*, z^*) = \mathsf{true} \wedge (w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$ **then**
return win	**if** $(w^* \notin \mathcal{L}'_{m^*}) \vee (w^* \in \mathcal{L}'_{m^*} \wedge c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*))$ **then** / G_{10}
	$\mathsf{win} := \mathsf{true}$ / G_{10}
	if $(w^* \notin \mathcal{L}_{m^*}) \vee (w^* \in \mathcal{L}'_{m^*} \wedge c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*))$ **then** / G_{11}
	$\mathsf{win} := \mathsf{true}$ / G_{11}
	if $w^* \neq \tilde{w}^{(k)} \wedge c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ **then** / G_{12}–G_{13}
	$\mathsf{win} := \mathsf{true}$ / G_{12}–G_{13}

Fig. 12. G_i for $i \in \{10, 11, 12, 13\}$. Sign, H, and GetTrans are those in G_7 in Figure 10.

Game G_9*:* Due to technical reasons, we modify the condition in Forge before changing the final condition of the game with returning flag win.
To ease the notation, let $\mathcal{L}_{m^*} := \{w^{(i)}\}_{i \in [k]}$ which are the w parts of the transcripts generated by $\textsc{GetTrans}(m^*)$. We additionally define $\mathcal{L}'_{m^*} := \{w^{(i)}\}_{i \in [k-1]}$.
In G_9, Forge additionally checks if $w^* \in \mathcal{L}'_{m^*}$ or not; if so, we modify the condition $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$. As Lemma 4.4, we have the following lemma.

Lemma 5.6. *Suppose that* LID *has* (α, ϵ_m)*-commitment min-entropy and is* ϵ_{zk}*-HVZK. We have*

$$|\Pr[W_8] - \Pr[W_9]| \leq \delta_{4,5} + \delta_{6,7} + \delta_{7,8}.$$

Proof. The two games differ if the adversary queries $w^* = w^{(i)}$ for $i < k$ but $c^* \neq \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$. We call this event in G_i Bad_i. We have

$$|\Pr[W_8] - \Pr[W_9]| \leq \Pr[\mathsf{Bad}_8] \leq |\Pr[\mathsf{Bad}_8] - \Pr[\mathsf{Bad}_3]| + \Pr[\mathsf{Bad}_3].$$

We have $\Pr[\mathsf{Bad}_3] = 0$ because $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ always holds in G_3. Since $|\Pr[\mathsf{Bad}_8] - \Pr[\mathsf{Bad}_3]|$ is upper-bounded by $\delta_{4,5} + \delta_{6,7} + \delta_{7,8}$, we obtain the bound. □

Game G_{10}*:* We then replace the output of the game. In G_{10}, the challenger just outputs the flag win instead of $[\![\#Q > \mathsf{cnt}_s]\!] \wedge \mathsf{win}$. See G_{10} in Figure 11 (and in Figure 12). We can forget Q. Since we relax the condition and the adversary cannot detect this modification, we have

$$\Pr[W_9] \leq \Pr[W_{10}].$$

Game G_{11}*:* We then treat the case $w^* = \tilde{w}^{(k)}$ as a special case. To do so, we replace the condition $w^* \notin \mathcal{L}'_{m^*}$ with $w^* \notin \mathcal{L}_{m^*}$. See G_{11} in Figure 12 for the details. This is easily reduced to the CUR property.

Lemma 5.7. *There exists a quantum* $\mathcal{F}$*-oracle adversary* $\mathcal{A}_{\mathrm{cur}}$ *such that*

$$\begin{aligned}
&|\Pr[W_{10}] - \Pr[W_{11}]| \leq \mathsf{Adv}^{\mathrm{cur}}_{\mathsf{LID}, \mathcal{A}_{\mathrm{cur}}}(\kappa), \\
&\mathsf{Time}^*(\mathcal{A}_{\mathrm{cur}}) = O(\mathsf{Time}(\mathcal{A})) + (q_H + q_S + q_F) \cdot O(B\mathsf{Time}(\mathsf{LID}) + B^2), \\
&\mathsf{Mem}^*(\mathcal{A}_{\mathrm{cur}}) = O(\mathsf{Mem}(\mathcal{A})) + O(B\mathsf{Mem}(\mathsf{LID})),
\end{aligned}$$

where $\mathcal{F} = \mathsf{Func}(\mathcal{M} \times \mathcal{W}, C) \times \mathsf{Func}(\mathcal{M} \times [B], C) \times \mathsf{Func}(\mathcal{M} \times [B], \mathcal{R}_{\mathsf{Sim}})$.

Since the proof is the same as that of Lemma 4.5, we omit it.

Game G_{12}*:* We again modify the conditions in Forge in G_{11}: Forge checks if $(w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$, $w^* \neq \tilde{w}^{(k)}$, and $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ or not. If so, the flag is set as true. See G_{12} in Figure 12 for the details. As Lemma 4.6, this modification does not change anything and we have $G_{11} = G_{12}$.

Game G_{13}*:* We finally replace a normal verification key with a lossy verification key. See G_{13} in Figure 12 for the details.
As Lemma 4.7, we have the following lemma:

Lemma 5.8. *There exists a quantum $\mathcal{F}$-oracle adversary $\mathcal{A}_{\mathrm{ind}}$ such that*

$$
\begin{aligned}
&|\Pr[W_{12}] - \Pr[W_{13}]| \leq \mathsf{Adv}^{\mathrm{indkey}}_{\mathsf{LID}, \mathcal{A}_{\mathrm{ind}}}(\kappa),\\
&\mathsf{Time}^*(\mathcal{A}_{\mathrm{ind}}) = O(\mathsf{Time}(\mathcal{A})) + (q_H + q_S + q_F) \cdot O(B\mathsf{Time}(\mathsf{LID}) + B^2),\\
&\mathsf{Mem}^*(\mathcal{A}_{\mathrm{ind}}) = O(\mathsf{Mem}(\mathcal{A})) + O(B\mathsf{Mem}(\mathsf{LID})),
\end{aligned}
$$

where $\mathcal{F} = \mathsf{Func}(\mathcal{M} \times \mathcal{W}, C) \times \mathsf{Func}(\mathcal{M} \times [B], C) \times \mathsf{Func}(\mathcal{M} \times [B], \mathcal{R}_{\mathsf{Sim}})$.

We also have the following lemma as Kiltz et al. [KLS18]. See the proof of the case for sBU security (Lemma D.10).

Lemma 5.9. *If* LID *is ϵ_ℓ-lossy, then we have* $\Pr[W_{13}] \leq 8(q_S + q_H + q_F + 1)^2 \epsilon_\ell$.

References

AAC+22. Gorjan Alagic, Daniel Apon, David Cooper, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, and Yi-Kai Liu. Status report on the third round of the NIST post-quantum cryptography standardization process. Technical report, NIST, 2022. 2

ACFK17. Benedikt Auerbach, David Cash, Manuel Fersch, and Eike Kiltz. Memory-tight reductions. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 101–132. Springer, Heidelberg, August 2017. 2, 5, 7

ADMP20. Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 411–439. Springer, Heidelberg, December 2020. 25, 26

AFLT12. Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-secure signatures from lossy identification schemes. In Pointcheval and Johansson [PJ12], pages 572–590. 2, 5, 11, 17

AFLT16. Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly secure signatures from lossy identification schemes. *Journal of Cryptology*, 29(3):597–631, July 2016. 5, 6, 10

AHU19. Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 269–295. Springer, Heidelberg, August 2019. 13

AMRS18. Gorjan Alagic, Christian Majenz, Alexander Russell, and Fang Song. Quantum-secure message authentication via blind-unforgeability. Cryptology ePrint Archive, Report 2018/1150, 2018. https://eprint.iacr.org/2018/1150. 3, 30

AMRS20. Gorjan Alagic, Christian Majenz, Alexander Russell, and Fang Song. Quantum-access-secure message authentication via blind-unforgeability. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 788–817. Springer, Heidelberg, May 2020. 2, 3, 7, 9, 30

BBD+23. Manuel Barbosa, Gilles Barthe, Christian Doczkal, Jelle Don, Serge Fehr, Benjamin Grégoire, Yu-Hsuan Huang, Andreas Hülsing, Yi Lee, and Xiaodi Wu. Fixing and mechanizing the security proof of fiat-shamir with aborts and dilithium. In Handschuh and Lysyanskaya [HL23], pages 358–389. 4

BCD+22. Markus Bläser, Zhili Chen, Dung Hoang Duong, Antoine Joux, Ngoc Tuong Nguyen, Thomas Plantard, Youming Qiao, Willy Susilo, and Gang Tang. On digital signatures based on isomorphism problems: QROM security, ring signatures, and applications. Cryptology ePrint Archive, Report 2022/1184, 2022. https://eprint.iacr.org/2022/1184. 25, 26

BDF+11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011. 2, 3, 4, 40

Bel93. Eric David Belsley. *Rates of convergence of Markov chains related to association schemes*. PhD thesis, Harvard University, May 1993. 29

Bha20. Rishiraj Bhattacharyya. Memory-tight reductions for practical key encapsulation mechanisms. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 249–278. Springer, Heidelberg, May 2020. 2

BKV19. Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 227–247. Springer, Heidelberg, December 2019. 27

BLS01. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 514–532. Springer, Heidelberg, December 2001. 2

BPS16. Mihir Bellare, Bertram Poettering, and Douglas Stebila. From identification to signatures, tightly: A framework and generic transforms. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 435–464. Springer, Heidelberg, December 2016. 9, 10

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993. 2

BR96. Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 399–416. Springer, Heidelberg, May 1996. 2, 40

BZ13a. Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 592–608. Springer, Heidelberg, May 2013. 30

BZ13b. Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 361–379. Springer, Heidelberg, August 2013. 2, 4, 5, 7, 42, 43

CCLM22. Rohit Chatterjee, Kai-Min Chung, Xiao Liang, and Giulio Malavolta. A note on the post-quantum security of (ring) signatures. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022, Part II*, volume 13178 of *LNCS*, pages 407–436. Springer, Heidelberg, March 2022. 3, 4, 40, 44

CKMS16. Sanjit Chatterjee, Neal Koblitz, Alfred Menezes, and Palash Sarkar. Another look at tightness II: Practical issues in cryptography. In Raphael C.-W. Phan and Moti Yung, editors, *Mycrypt 2016*, volume 10311 of *LNCS*, pages 21–55. Springer, 2016. 2

CLM+18. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427. Springer, Heidelberg, December 2018. 27

CMS12. Sanjit Chatterjee, Alfred Menezes, and Palash Sarkar. Another look at tightness. In Ali Miri and Serge Vaudenay, editors, *SAC 2011*, volume 7118 of *LNCS*, pages 293–319. Springer, Heidelberg, August 2012. 2

Cor00. Jean-Sébastien Coron. On the exact security of full domain hash. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 229–235. Springer, Heidelberg, August 2000. 2

DDKA21. Mina Doosti, Mahshid Delavar, Elham Kashefi, and Myrto Arapinis. A unified framework for quantum unforgeability. *CoRR*, abs/2103.13994, 2021. 2

DFPS22. Julien Devevey, Omar Fawzi, Alain Passelègue, and Damien Stehlé. On rejection sampling in lyubashevsky's signature scheme. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part IV*, volume 13794 of *LNCS*, pages 34–64. Springer, Heidelberg, December 2022. 27

DFPS23. Julien Devevey, Pouria Fallahpour, Alain Passelègue, and Damien Stehlé. A detailed analysis of fiat-shamir with aborts. In Handschuh and Lysyanskaya [HL23], pages 327–357. 3, 4, 6, 7, 10, 11, 13, 14, 17, 18, 36

DGJL21. Denis Diemert, Kai Gellert, Tibor Jager, and Lin Lyu. Digital signatures with memory-tight security in the multi-challenge setting. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 403–433. Springer, Heidelberg, December 2021. 2, 3, 9, 10, 11, 14, 40

Din20. Itai Dinur. Tight time-space lower bounds for finding multiple collision pairs and their applications. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 405–434. Springer, Heidelberg, May 2020. 2

dPPRS23. Rafaël del Pino, Thomas Prest, Mélissa Rossi, and Markku-Juhani O. Saarinen. High-order masking of lattice signatures in quasilinear time. In *44th IEEE Symposium on Security and Privacy, SP 2023, San Francisco, CA, USA, May 21-25, 2023*, pages 1168–1185. IEEE, 2023. 4

DPS23. Julien Devevey, Alain Passelègue, and Damien Stehlé. G+G: A Fiat-Shamir lattice signature based on convolved Gaussians. In *ASIACRYPT 2023*, 2023. To appear. See https://eprint.iacr.org/2023/1477. 3, 4, 27, 28

EKP20. Ali El Kaafarani, Shuichi Katsumata, and Federico Pintore. Lossy CSI-FiSh: Efficient signature scheme with tight reduction to decisional CSIDH-512. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 157–186. Springer, Heidelberg, May 2020. 3, 25, 26, 27

FG15. Jason Fulman and Larry Goldstein. Stein's method and the rank distribution of random matrices over finite fields. *The Annals of Probability*, 43(3):1274–1314, May 2015. 29

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

GGJT22. Ashrujit Ghoshal, Riddhi Ghosal, Joseph Jaeger, and Stefano Tessaro. Hiding in plain sight: Memory-tight proofs via randomness programming. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 706–735. Springer, Heidelberg, May / June 2022. 2, 5, 40

GHHM21. Alex B. Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight adaptive reprogramming in the QROM. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 637–667. Springer, Heidelberg, December 2021. 4

GJT20. Ashrujit Ghoshal, Joseph Jaeger, and Stefano Tessaro. The memory-tightness of authenticated encryption. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 127–156. Springer, Heidelberg, August 2020. 2

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008. 2, 39, 40

GT20. Ashrujit Ghoshal and Stefano Tessaro. On the memory-tightness of hashed ElGamal. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 33–62. Springer, Heidelberg, May 2020. 2

HL23. Helena Handschuh and Anna Lysyanskaya, editors. *CRYPTO 2023, Part V*, volume 14085 of *LNCS*. Springer, Heidelberg, August 2023. 21, 22

HRS16. Andreas Hülsing, Joost Rijneveld, and Fang Song. Mitigating multi-target attacks in hash-based signatures. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016, Part I*, volume 9614 of *LNCS*, pages 387–416. Springer, Heidelberg, March 2016. 25

JK22. Joseph Jaeger and Akshaya Kumar. Memory-tight multi-challenge security of public-key encryption. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part III*, volume 13793 of *LNCS*, pages 454–484. Springer, Heidelberg, December 2022. 2, 4

KLS18. 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, Heidelberg, April / May 2018. 3, 4, 6, 7, 10, 11, 15, 16, 17, 21, 25, 39

KM07. Neal Koblitz and Alfred J. Menezes. Another look at "provable security". *Journal of Cryptology*, 20(1):3–37, January 2007. 2

KM15. Neal Koblitz and Alfred J. Menezes. The random oracle model: a twenty-year retrospective. *Des. Codes Cryptogr.*, 77:587–610, 2015. 10

KX22. Haruhisa Kosuge and Keita Xagawa. Probabilistic hash-and-sign with retry in the quantum random oracle model. Cryptology ePrint Archive, Report 2022/1359, 2022. https://eprint.iacr.org/2022/1359. 4

LJZ22. Yu Liu, Haodong Jiang, and Yunlei Zhao. Tighter post-quantum proof for plain FDH, PFDH and GPV-IBE. Cryptology ePrint Archive, Report 2022/1441, 2022. https://eprint.iacr.org/2022/1441. 4

Lyu09. Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Heidelberg, December 2009. 27

Lyu12. Vadim Lyubashevsky. Lattice signatures without trapdoors. In Pointcheval and Johansson [PJ12], pages 738–755. 27

MMO04. Theresa Migler, Kent E. Morrison, and Mitchell Ogle. Weight and rank of matrices over finite fields, 2004. 29

MNPV99. David M'Raïhi, David Naccache, David Pointcheval, and Serge Vaudenay. Computational alternatives to random number generators. In Stafford E. Tavares and Henk Meijer, editors, *SAC 1998*, volume 1556 of *LNCS*, pages 72–80. Springer, Heidelberg, August 1999. 10

PJ12. David Pointcheval and Thomas Johansson, editors. *EUROCRYPT 2012*, volume 7237 of *LNCS*. Springer, Heidelberg, April 2012. 21, 23

RSA78. Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126, February 1978. 2

WMHT18. Yuyu Wang, Takahiro Matsuda, Goichiro Hanaoka, and Keisuke Tanaka. Memory lower bounds of reductions revisited. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 61–90. Springer, Heidelberg, April / May 2018. 2

Zha12. Mark Zhandry. How to construct quantum random functions. In *53rd FOCS*, pages 679–687. IEEE Computer Society Press, October 2012. 5, 25

Supplementary Materials

A Missign Definitions

Pseudorandom functions:

Definition A.1 (Pseudorandom functions). *Let κ be the security parameter. Let $\delta = \delta(\kappa)$ and $\rho = \rho(\kappa)$ be two polynomial functions. Let* $\mathsf{PRF}\colon \{0,1\}^\kappa \times \{0,1\}^\delta \to \{0,1\}^\rho$ *be a deterministic polynomial-time algorithm. We say that* PRF *is* pseudorandom *if for any QPT adversary $\mathcal{A}$, its advantage*

$$\mathsf{Adv}^{\mathrm{pr}}_{\mathsf{PRF},\mathcal{A}}(\kappa) := \left| \begin{matrix} \Pr[K \leftarrow \{0,1\}^\kappa : \mathcal{A}^{|\mathsf{PRF}(K,\cdot)\rangle}(1^\kappa) \to 1] \\ -\Pr[\mathsf{RF} \leftarrow \mathsf{Func}(\{0,1\}^\delta, \{0,1\}^\rho) : \mathcal{A}^{|\mathsf{RF}(\cdot)\rangle}(1^\kappa) \to 1] \end{matrix} \right|$$

is negligible in κ.

Generic quantum search [Zha12, HRS16, KLS18]: Let $\mathcal{X}$ be a finite set. The generic search problem (GSP, in short) is finding $x \in \mathcal{X}$ satisfying $g(x) = 1$ given access to an oracle $g\colon \mathcal{X} \to \{0,1\}$, where for each $x \in \mathcal{X}$, $g(x)$ is drawn independently according to Ber_λ, that is, $g \leftarrow \mathsf{Func}_{\mathcal{X},\{0,1\}}(\mathsf{Ber}_\lambda)$. Kiltz et al. [KLS18] generalized this problem by modifying the global distribution Ber_λ into $\mathsf{Ber}_{\lambda(x)}$ on each x, that is, the case that $g \leftarrow \mathsf{Func}_{\mathcal{X},\{0,1\}}(\{\mathsf{Ber}_{\lambda(x)} : x \in \mathcal{X}\})$.

Lemma A.1 (Generic search problem with bounded probabilities [KLS18]). *Let $\lambda \in [0,1]$. For any quantum algorithm $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ making at most q queries to $|g\rangle$, we have*

$$\Pr[\mathrm{GSPB}_{\lambda,\mathcal{A}} = 1] \le 8(q+1)^2\lambda,$$

where game $\mathrm{GSPB}_{\lambda,\mathcal{A}}$ *is defined in Figure 13.*

$\mathrm{GSPB}_{\lambda,\mathcal{A}}$

1 : $\{\lambda(x)\}_{x\in\mathcal{X}} \leftarrow \mathcal{A}_1$
2 : **if** $\exists x \in \mathcal{X}$ s.t. $\lambda(x) > \lambda$ **then return** $\bot$
3 : **foreach** $x \in \mathcal{X} : g(x) \leftarrow \mathsf{Ber}_{\lambda(x)}$
4 : $x \leftarrow \mathcal{A}_2^{|g\rangle}$
5 : **return** $g(x)$

Fig. 13. The generic search game $\mathrm{GSPB}_{\lambda,\mathcal{A}}$.

B Instantiations of Lossy Identification

We review three instantiations of lossy identification schemes from post-quantum assumptions.

B.1 Lossy Identification Scheme based on Pseudorandom Group Action

Bläser et al. [BCD+22] gave a lossy identification scheme, which is an abstraction of $\mathsf{ID}^{\mathsf{enCh}}_{\mathsf{ls}}$ in [EKP20], based on pseudorandom group action. We briefly review cryptographic group action [ADMP20].

Definition B.1 (Group action). *Let G be a group with identity element 1_G. Let X be a set. A map $\star\colon G \times X \to X$ is* a group action *if, for all $g, h \in G$ and $x \in X$, $1_G \star x = x$ and $(gh) \star x = g \star (h \star x)$.*

In this section, we assume that G and X are finite. For the security of the LID scheme, we require the hardness of the following problem, which is an adapted version of [BCD+22, Definition 6].

Definition B.2 (S-pseudorandom problem, adapted). *Let S be a positive integer. Let $(G, X, \star)$ be a group action. The S-*pseudorandom *problem with parameter S asks to distinguish between the following two distributions:*

- $(E, g_1 \star E, g_2 \star E, \ldots, g_S \star E)$, *where $E \leftarrow X$ and $g_1, \ldots, g_S \leftarrow G$.*
- $(E, E_1, E_2, \ldots, E_S)$ *where $E, E_1, \ldots, E_S \leftarrow X$.*

For CUR property, we will use the following problem [BCD+22, Definition 7]:

Definition B.3 (Stabilizer problem). *Let $(G, X, \star)$ be a regular group action. The* stabilizer *problem is, given a random element $E \leftarrow X$, finding a non-trivial stabilizer $g \in G \setminus \{1_G\}$ satisfying $g \star E = E$. The* $\mathsf{Stab}_{(G,X,\star)}$ *assumption states that for any QPT adversary $\mathcal{A}$, its advantage*

$$\mathsf{Adv}_{\mathsf{Stab},\mathcal{A}}(\kappa) \coloneqq \Pr[E \leftarrow X, g \leftarrow \mathcal{A}(G, X, \star, E) : g \star E = E \wedge g \in G \setminus \{1_G\}]$$

is negligible in κ.

The description of the scheme follows:

Public parameter: The public parameter is a cryptographic group action $(G, X, \star)$. We have $\mathcal{W} \coloneqq X^t$ and $\mathcal{Z} \coloneqq G^t$.

Key generation: $\mathsf{Gen}_{\mathsf{LID}}$ uniformly samples $E_0 \leftarrow X$ and $g_1, \ldots, g_S \leftarrow G$, lets $g_0 \coloneqq 1_G$, and outputs

$$vk = \{E_0, E_1, \ldots, E_S\} \text{ and } sk = (vk, g_0, g_1, \ldots, g_S),$$

where $E_i \coloneqq g_i \star E_0$ for $i = 1, \ldots, S$.

Lossy key generation: $\mathsf{LossyGen}_{\mathsf{LID}}$ uniformly samples $E_0, \ldots, E_S \leftarrow X$ and outputs

$$vk = \{E_0, E_1, \ldots, E_S\}.$$

Challenge space: The challenge space is $C \coloneqq \{0, 1, \ldots, S\}^t$.

Prover: The prover's algorithms are defined as follows:

- $\mathsf{P}_1(sk)$ uniformly samples $r_1, \ldots, r_t \leftarrow G$ and returns a commitment $w = (W_1, \ldots, W_t)$, where $W_i \coloneqq r_i \star E_0$ for $i = 1, \ldots, t$, and outputs a state information $s \coloneqq (r_1, \ldots, r_t)$.
- $\mathsf{P}_2(sk, w, c, s)$, where $c = (c_1, \ldots, c_t)$ and $s = (r_1, \ldots, r_t)$, computes $z_i \coloneqq r_i \cdot g_{c_i}^{-1} \in G$ for $i = 1, \ldots, t$ and returns $z = (z_1, \ldots, z_t)$.

Reconstruction: $\mathsf{Rec}(vk, c, z)$ computes $W_i = z_i \star E_{c_i}$ for $i = 1, \ldots, t$ and returns $w = (W_1, \ldots, W_t)$.

Verifier: $\mathsf{V}(vk, w, c, z)$ checks if $w = \mathsf{Rec}(vk, c, z)$ or not.

Simulator: $\mathsf{Sim}(vk, c)$ uniformly samples $z = (z_1, \ldots, z_t) \leftarrow G^t$ and outputs $w = \mathsf{Rec}(vk, c, z)$.

The signature scheme is obtained by applying $\mathsf{DFS}_{1,cz}$ to the above lossy identification scheme.

The protocol is ϵ_ℓ-lossy with $\epsilon_\ell = \frac{1}{(S+1)^t} \cdot \prod_{i\in[S]} \frac{|X|-i|G|}{|X|} + (1 - \prod_{i\in[S]} \frac{|X|-i|G|}{|X|})$ [BCD+22, Lemma 4]. This ϵ_ℓ is negligible in κ when S is constant, $t = \omega(\lg(\kappa))$, and $|X| \gg |G|$. Key indistinguishability follows from the hardness of the S-pseudorandom problem of the underlying group action. In addition, parameters of the commitment min-entropy are $\alpha = t \cdot \lg(N)$ and $\epsilon_m = 0$. The protocol achieves perfect completeness and perfect HVZK.

We give the proof of CUR to check the memory usage of the reduction algorithm. The underlying problem is the stabilizer problem,

Lemma B.1. *The protocol has the CUR property under the* $\mathsf{Stab}_{(G,X,\star)}$ *assumption: Precisely speaking, for a quantum adversary $\mathcal{A}$ breaking the CUR property of the identification protocol, there exists a quantum adversary $\mathcal{A}_{\mathsf{stab}}$ against the* $\mathsf{Stab}_{(G,X,\star)}$ *assumption such that*

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{cur}}_{\mathsf{LID},\mathcal{A}}(\kappa) \le \mathsf{Adv}_{\mathsf{Stab},\mathcal{A}_{\mathsf{stab}}}(\kappa),\\
&\mathsf{Time}^*(\mathcal{A}_{\mathsf{stab}}) = \mathsf{Time}(\mathcal{A}) + S \cdot O(|G|, |X|),\\
&\mathsf{Mem}^*(\mathcal{A}_{\mathsf{stab}}) = \mathsf{Mem}(\mathcal{A}) + S \cdot O(|G|, |X|).
\end{aligned}$$

Proof. We construct $\mathcal{A}_{\mathsf{stab}}$ as follows: Given E chosen from X, it samples $g_1, \ldots, g_S \leftarrow G$, computes $E_i \coloneqq g_i \star E$, and runs $\mathcal{A}$ on input $(E_0 \coloneqq E, E_1, E_2, \ldots, E_S)$. The adversary $\mathcal{A}$ outputs w, c, z and ζ. Since $z \neq \zeta$, there exists $i \in [t]$ satisfying $z_i \neq \zeta_i$. $\mathcal{A}_{\mathsf{stab}}$ outputs $f \coloneqq g_{c_i}^{-1} \zeta_i^{-1} z_i g_{c_i}$ if it is not 1_G.

Let us check that f is a stabilizer. If (w, c, z) and (w, c, ζ) are valid, then we have $W_i = z_i \star E_{c_i} = \zeta_i \star E_{c_i}$. Putting $E_{c_i} = g_{c_i} \star E$, we have $(z_i g_{c_i}) \star E = (\zeta_i g_{c_i}) \star E$. Thus, $f \star E = \big((\zeta_i g_{c_i})^{-1} \cdot (z_i g_{c_i})\big) \star E = E$ and f is a stabilizer for E. It is easy to check that this f is non-trivial: Since $z_i \neq \zeta_i$, we have $z_i g_{c_i} \neq \zeta_i g_{c_i}$ and $f = (\zeta_i g_{c_i})^{-1} \cdot z_i g_{c_i} \neq 1_G$ as we wanted. Hence, the advantage of $\mathcal{A}_{\mathsf{stab}}$ is equivalent to that of $\mathcal{A}$. □

B.2 Lossy Identification Scheme based on CSIDH

We recall a lossy identification scheme $\mathsf{ID}_{\mathsf{ls}}^{\mathsf{denCh}}$ in Lossy CSI-FiSh proposed by El Kaafarani, Katsumata, and Pintore [EKP20], which is based on the hardness of the decisional Diffie-Hellman problems in the CSIDH setting.

We briefly review cryptographic group action [ADMP20] and quadratic twist.

Definition B.4 (Regulality of group action). *We say that a group action $(G, X, \star)$ is* regular *if the following two conditions hold: (transitive:) for every $x, x' \in X$, there exists $g \in G$ satisfying $x' = g \star x$. (free:) for each group element $g \in G$, $g = 1_G$ if and only if there exists some element $x \in X$ such that $x = g \star x$.*

In what follows, we assume that $G = \langle g \rangle$ of cardinality N. In the CSIDH setting, given $E = g^a \star E_0$, we can compute its quadratic twist $\mathrm{twist}(E)$, which is $g^{-a} \star E_0$ [CLM+18, BKV19, EKP20].
For the security of the LID scheme, we require the hardness of the following problem, which is an adapted version of [EKP20, Definition 4.1].

Definition B.5 (Fixed-Curve Multi-Decisional GADH problem). *Let S be a positive integer. Suppose that $G = \langle g \rangle$ of cardinarity N and X be a finite set. Let $(G, X, \star)$ be a regular group action. The* fixed-curve multi-decisional group-action Diffie-Hellman (FCMD-GADH) *problem with parameter S asks to distinguish between the following two distributions:*
- *$(E, H, g^{a_1} \star E, g^{a_1} \star H, \ldots, g^{a_S} \star E, g^{a_S} \star H)$, where $E, H \leftarrow X$ and $a_1, \ldots, a_S \leftarrow \mathbb{Z}_N$.*
- *$(E, H, E'_1, H'_1, \ldots, E'_S, H'_S)$ where $E, H, E'_1, H'_1, \ldots, E'_S, H'_S \leftarrow X$.*

If $S = 1$, the problem is said to be the decisional group-action Diffie-Hellman (D-GADH) problem.

The description of the scheme follows:
Public parameter: The public parameter is a cryptographic group action $(G, X, \star)$. Let $E_0 \in X$ be a fixed element in X. We have $\mathcal{W} := (X^2)^t$ and $\mathcal{Z} := \mathbb{Z}_N^t$.
Key generation: $\mathsf{Gen}_{\mathsf{LID}}$ uniformly samples $a_1, \ldots, a_S, b_1, b_2 \leftarrow \mathbb{Z}_N$, lets $a_0 := 0$, and outputs

$$vk = \{(E_1^{(i)}, E_2^{(i)})\}_{i \in \{0,\ldots,S\}} \text{ and } sk = (vk, a_0, a_1, \ldots, a_S, b_1, b_2),$$

where $E_1^{(0)} := g^{b_1} \star E_0$, $E_2^{(0)} := g^{b_2} \star E_0$, $E_0^{(i)} = g^{a_i} \star E_0^{(0)}$, and $E_1^{(i)} = g^{a_i} \star E_1^{(0)}$ for $i = 1, \ldots, S$. For ease of notation, for $\beta \in \{1, 2\}$ and $i \in [S]$, we define $E_\beta^{(-i)} := \mathrm{twist}(E_\beta^{(i)})$.
Lossy key generation: $\mathsf{LossyGen}_{\mathsf{LID}}$ uniformly samples $a_1, \ldots, a_S, a'_1, \ldots, a'_S, b_1, b_2 \leftarrow \mathbb{Z}_N$ and outputs

$$vk = \{(E_1^{(i)}, E_2^{(i)})\}_{i \in \{0,\ldots,S\}},$$

where $E_1^{(0)} := g^{b_1} \star E_0$, $E_2^{(0)} := g^{b_2} \star E_0$, $E_0^{(i)} = g^{a_i} \star E_0^{(0)}$, and $E_1^{(i)} = g^{a'_i} \star E_1^{(0)}$ for $i = 1, \ldots, S$.
Challenge space: The challenge space is $C := \{-S, -S+1, \ldots, S-1, S\}^t$.
Prover: The prover's algorithms are defined as follows:
- $\mathsf{P}_1(sk)$ unformly samples $r_1, \ldots, r_t \leftarrow \mathbb{Z}_N$ and returns a commitment $w = (w_1, \ldots, w_t) = \{(F_1^{(k)}, F_2^{(k)})\}_{i \in [t]}$, where $(F_1^{(k)}, F_2^{(k)}) := (g^{r_k} \star E_1^{(0)}, g^{r_k} \star E_2^{(0)})$ and outputs a state information $s := (r_1, \ldots, r_t)$.
- $\mathsf{P}_2(sk, w, c, s)$, where $c = (c_1, \ldots, c_t)$ and $s = (r_1, \ldots, r_t)$, computes, for $k \in [t]$, $z_k = r_k - a_{c_k} \in \mathbb{Z}_N$ if $c_k \geq 0$ and $z_k = r_k + b_1 + b_2 + a_{|c_k|}$ otherwise and returns $z = (z_1, \ldots, z_t)$.

Reconstruction: $\mathsf{Rec}(vk, c, z)$ computes, for $k \in [t]$, $w_k = (g^{z_k} \star E_1^{(c_k)}, g^{z_k} \star E_2^{(c_k)})$ if $c \geq 0$ and $(g^{z_k} \star E_2^{(c_k)}, g^{z_k} \star E_1^{(c_k)})$ otherwise and returns $w = (w_1, \ldots, w_t)$.
Verifier: $\mathsf{V}(vk, w, c, z)$ checks if $w = \mathsf{Rec}(vk, c, z)$ or not.
Simulator: $\mathsf{Sim}(vk, c)$ uniformly samples $z = (z_1, \ldots, z_t) \leftarrow \mathbb{Z}_N^t$ and outputs $w = \mathsf{Rec}(vk, c, z)$.
The signature scheme Lossy CSI-FiSh is obtained by applying $\mathsf{DFS}_{1,cz}$ to the above lossy identification scheme. El Kaafarani et al. showed that the protocol is ϵ_ℓ-lossy with $\epsilon_\ell = \frac{1}{(2S+1)^t} \cdot \prod_{i \in [S]} \frac{N-i}{N} + (1 - \prod_{i \in [S]} \frac{N-i}{N})$ [EKP20, Lemma 4.7]. This ϵ_ℓ is negligible in κ when S is constant and $t = \omega(\lg(\kappa))$. Key indistinguishability follows from the hardness of the FCMD-GADH problem. The protocol achieves perfect completeness, perfect HVZK, and perfect unique response. In addition, parameters of the commitment min-entropy are $\alpha = t \cdot \lg(N)$ and $\epsilon_m = 0$.

B.3 Lossy Identification Scheme based on Lattices

As an example of lossy identification based on lattices, we take a new scheme G+G proposed by Devevey, Passelègue, and Stehlé [DPS23] instead of [Lyu09, Lyu12, DFPS22]. We first define the Gaussian function with covariance parameter $\Sigma \in \mathbb{R}^{k \times k}$, which is a positive-definite symmetric matrix, and center parameter $c \in \mathbb{R}^k$ as

$$\rho_{\Sigma,c}(x) = \exp\left(-\pi(x-c)^\top \Sigma^{-1}(x-c)\right).$$

For a lattice $\Lambda \subseteq \mathrm{Span}(\Sigma)$, the Gaussian distribution over Λ with covariance parameter Σ and center parameter c is defined by a probability mass function

$$D_{\Lambda,\Sigma,c}(x) = \frac{\rho_{\Sigma,c}(x)}{\sum_{y \in \Lambda} \rho_{\Sigma,c}(y)} \text{ for } x \in \Lambda.$$

Definition B.6 (Learning With Errors (LWE), Hermite Normal Form). *Let $m, k, q \in \mathbb{Z}^+$ with $q \geq 2$. Let χ be a distribution over $\mathbb{Z}$. The $\mathsf{LWE}_{m,k,\ell,q,\chi}$ assumption states that for any QPT adversary $\mathcal{A}$, the following two distributions are computationally indistinguishable:*

$$D_1 : A \leftarrow \mathbb{Z}_q^{m\times k}; S \leftarrow \chi^{k\times\ell}; E \leftarrow \chi^{m\times\ell};\ \textit{return}\ (A, AS+E),$$
$$D_2 : A \leftarrow \mathbb{Z}_q^{m\times k}; U \leftarrow \mathbb{Z}_q^{m\times\ell};\ \textit{return}\ (A, U).$$

Definition B.7 (Short Integer Solution). *Let $m, k, q \in \mathbb{Z}^+$ with $q \geq 2$. Let $\gamma > 0$. The $\mathsf{SIS}_{m,k,q,\gamma}$ assumption states that for any QPT adversary $\mathcal{A}$, its advantage*

$$\mathsf{Adv}_{\mathsf{SIS},\mathcal{A}}(\kappa) := \Pr[A \leftarrow \mathbb{Z}_q^{m\times k}, x \leftarrow \mathcal{A}(A) : Ax \equiv 0 \pmod q \wedge x \neq 0 \wedge \|x\| \leq \gamma]$$

is negligible in κ.

The description of the LID scheme follows:

Public parameter: The public parameters are $m \geq \ell > 0$, $k > m + \ell$, and $C \subseteq \mathbb{Z}_2^\ell$, odd modulus q, a bound $\gamma \in \mathbb{R}^+$, a distribution χ over $\mathbb{Z}$, Gaussian parameters s and σ. Define $\Sigma\colon \mathbb{Z}^{k\times\ell} \to \mathbb{R}^{k\times k}$ as $S \mapsto \sigma^2 I_k - s^2 SS^\top$. Let $J := [I_m \mid 0^{m\times(k-m)}]^T \in \mathbb{Z}^{k\times m}$. Let $\mathcal{W} := \mathbb{Z}_{2q}^m$ and $\mathcal{Z} := \{z \in \mathbb{Z}^k \mid \|z\| \leq \gamma\}$.

Key generation: $\mathsf{Gen}_{\mathsf{LID}}$ computes vk and sk as follows: $A_1 \leftarrow \mathbb{Z}_q^{m\times(k-m-\ell)}$; $(S_1, S_2) \leftarrow \chi^{(k-m-\ell)\times\ell} \times \chi^{m\times\ell}$; $B := A_1 S_1 + S_2 \bmod q$; $A := [qJ - 2B \mid 2A_1 \mid 2I_m] \in \mathbb{Z}_{2q}^{m\times k}$; $S := [I_\ell \mid S_1^\top \mid S_2^\top]^\top \in \mathbb{Z}^{k\times\ell}$; $vk := A$; $sk := S$; outputs vk and sk.

Lossy key generation: $\mathsf{LossyGen}_{\mathsf{LID}}$ compute vk as follows: $A_1 \leftarrow \mathbb{Z}_q^{m\times(k-m-\ell)}$; $B \leftarrow \mathbb{Z}_q^{m\times\ell}$; $A := [qJ - 2B \mid 2A_1 \mid 2I_m] \in \mathbb{Z}_{2q}^{m\times k}$; outputs $vk := A$.

Challenge space: The challenge space is $C \subseteq \mathbb{Z}_2^\ell$. See [DPS23] for the parameter choices.

Prover: The prover's algorithms are defined as follows:

- $\mathsf{P}_1(sk)$ samples $y \leftarrow D_{\mathbb{Z}^k,\Sigma(S),0}$ and computes $w := Ay \bmod 2q$. It also samples seed for a seed of P_2. It outputs w and $\mathsf{st} := (y, \mathsf{seed})$.
- $\mathsf{P}_2(sk, w, c, \mathsf{st})$ samples $k \leftarrow D_{\mathbb{Z}^\ell, s^2 I_\ell, -c/2}$ (with seed seed) and computes $z := y + 2Sk + Sc$. It outputs z.

Reconstruction: $\mathsf{Rec}(vk, c, z)$ returns $w = Az - qJc \bmod 2q$. (Note: Rec implicitly checks whether $z \in \mathcal{Z}$ or not.)

Verifier: $\mathsf{V}(vk, w, c, z)$ checks if $w = \mathsf{Rec}(vk, c, z)$.

Simulator: $\mathsf{Sim}(vk, c)$ samples $z \leftarrow D_{\mathbb{Z}^k,\sqrt{2}\sigma}$ and outputs $w = \mathsf{Rec}(vk, c, z)$.

The protocol achieves statistical completeness ($(1 - \mathsf{negl}(\kappa), 1)$-correctness) [DPS23, Theorem 2], statistical HVZK and the high commitment min-entropy [DPS23, Theorem 3] with careful choice of parameters. It also ϵ_ℓ-lossy with appropriate parameter settings, and its key indistinguishability follows from the decisional LWE assumption [DPS23, Theorem 4]. The computational unique response follows from the SIS assumption and the LWE assumption as follows:

Lemma B.2. *Let $m \geq \ell > 0$ and $k > m + \ell$. Let a be a positive integer. Let q be the odd modulus and let $\gamma \in \mathbb{R}^+$ be bound. Let χ be a distribution over $\mathbb{Z}$. The protocol has the CUR property under the $\mathsf{SIS}_{m,k+a,q,2\gamma}$ assumption and the $\mathsf{LWE}_{k-m-\ell,m,\ell,\chi,q}$ assumption.*
Precisely speaking, for a quantum adversary $\mathcal{A}$ breaking the CUR property of the identification protocol, there exist a quantum adversary $\mathcal{A}_{\mathrm{lwe}}$ against the $\mathsf{LWE}_{k-m-\ell,m,\ell,\chi,q}$ assumption and a quantum adversary $\mathcal{A}_{\mathrm{sis}}$ against the $\mathsf{SIS}_{m,k+a,q,2\gamma}$ assumption such that

$$\mathsf{Adv}^{\mathrm{cur}}_{\mathsf{LID},\mathcal{A}}(\kappa) \leq \mathsf{Adv}_{\mathsf{LWE},\mathcal{A}_{\mathrm{lwe}}}(\kappa) + \mathsf{Adv}_{\mathsf{SIS},\mathcal{A}_{\mathrm{sis}}}(\kappa) + 2mq^{-(a+1)},$$
$$\mathsf{Time}^*(\mathcal{A}_{\mathrm{lwe}}) = \mathsf{Time}(\mathcal{A}) + O(\mathsf{Time}(\mathsf{LID})),$$
$$\mathsf{Mem}^*(\mathcal{A}_{\mathrm{lwe}}) = \mathsf{Mem}(\mathcal{A}) + O(\mathsf{Mem}(\mathsf{LID})),$$
$$\mathsf{Time}^*(\mathcal{A}_{\mathrm{sis}}) = \mathsf{Time}(\mathcal{A}) + O((k+a)^3 \log^3 q),$$
$$\mathsf{Mem}^*(\mathcal{A}_{\mathrm{sis}}) = \mathsf{Mem}(\mathcal{A}) + O(m(k+a)\log q).$$

Proof. Let us consider two games: The first one is G_0 in which the challenger generates $(vk, sk) \leftarrow \mathsf{Gen}_{\mathsf{LID}}(1^\kappa)$, runs the adversary on input vk and receives (w, c, z, z') from the adversary, and returns $[\![z \neq z' \wedge \mathsf{V}(vk, w, c, z) \wedge \mathsf{V}(vk, w, c, z')]\!]$. The second one G_1 is the same game G_0 except that $vk \leftarrow \mathsf{LossyGen}_{\mathsf{LID}}(1^\kappa)$.
By definition, we have

$$\Pr[\mathsf{G}_0 \Rightarrow \mathsf{true}] = \mathsf{Adv}^{\mathrm{cur}}_{\mathsf{LID},\mathcal{A}}(\kappa).$$

It is easy to construct an adversary $\mathcal{A}_{\mathrm{lwe}}$ such that

$$\begin{aligned}
&|\Pr[\mathrm{G}_0 \Rightarrow \mathsf{true}] - \Pr[\mathrm{G}_1 \Rightarrow \mathsf{true}]| \leq \mathsf{Adv}_{\mathsf{LWE},\mathcal{A}_{\mathrm{lwe}}}(\kappa),\\
&\mathsf{Time}^*(\mathcal{A}_{\mathrm{lwe}}) = \mathsf{Time}(\mathcal{A}) + O(\mathsf{Time}(\mathsf{LID})),\\
&\mathsf{Mem}^*(\mathcal{A}_{\mathrm{lwe}}) = \mathsf{Mem}(\mathcal{A}) + O(\mathsf{Mem}(\mathsf{LID})).
\end{aligned}$$

By using an adversary $\mathcal{A}$ in G_1, we construct an adversary $\mathcal{A}_{\mathrm{sis}}$ as follows: Given $\tilde{A} = [\tilde{A}_1 \mid \tilde{A}_2] \leftarrow \mathbb{Z}_q^{m\times(k+a)}$ with $\tilde{A}_1 \in \mathbb{Z}_q^{m\times(k-m)}$ and $\tilde{A}_2 \in \mathbb{Z}_q^{m\times(m+a)}$, $\mathcal{A}_{\mathrm{sis}}$ tries to find a set of m linearly independent vectors $\tilde{a}_{i_1}, \ldots, \tilde{a}_{i_m}$ from $\tilde{A}_2$. This set exists with probability at least $1 - 2mq^{-(a+1)}$ (see Lemma B.3 below). Let $\hat{A} = [\hat{A}_1 \mid \hat{A}_2] := [\tilde{A}_1 \mid \tilde{a}_{i_1} \ldots \tilde{a}_{i_m}] = \tilde{A} \cdot P$, where $\hat{A}_1 \in \mathbb{Z}_q^{(k-m)\times m}$, $\hat{A}_2 = [\tilde{a}_{i_1} \ldots \tilde{a}_{i_m}] \in \mathbb{Z}_q^{m\times m}$, and P is a corresponding matrix in $\{0,1\}^{(k+a)\times k}$. Notice that the Hamming weight of the columns of P is 1, and the Hamming weight of the rows of P is at most 1. It then computes $A := 2((2\hat{A}_2)^{-1}\cdot\hat{A} \bmod q) + [qJ \mid O] \bmod 2q$ and feeds it to $\mathcal{A}$. The distribution of this lossy verification key is perfect. $\mathcal{A}$ outputs (w, c, z, z'). If $z \neq z'$ and $\mathsf{V}(vk, w, c, z) = \mathsf{V}(vk, w, c, z') = \mathsf{true}$, then $\mathcal{A}_{\mathrm{sis}}$ output $P(z - z')$ as the solution of the SIS problem.
If we have (w, c, z, z') such that $z \neq z'$ and $\mathsf{V}(vk, w, c, z) = \mathsf{V}(vk, w, c, z') = \mathsf{true}$, then we have the relations

$$\|z\| \leq \gamma \wedge \|z'\| \leq \gamma \wedge w \equiv Az - qJc \equiv Az' - qJc \pmod{2q}.$$

The bounds on the norms imply $\|z - z'\| \leq 2\gamma$ and the condition $z \neq z'$ implies $z - z' \neq 0$. The last equation with the fact that q is odd implies $A(z - z') \equiv 0 \pmod{q}$ (instead of $2q$). Therefore, we have

$$A(z - z') \equiv 2(2\hat{A}_2)^{-1}\hat{A} \cdot (z - z') \equiv \hat{A}_2^{-1} \cdot \tilde{A} \cdot P(z - z') \equiv 0 \pmod{q}$$

Multiplying $\hat{A}_2$ to the both sides, we have

$$\tilde{A} \cdot P(z - z') \equiv 0 \pmod{q}.$$

Due to the property of P, we have $P(z - z') \neq 0$ and $\|P(z - z')\| = \|z - z'\| \leq 2\gamma$. Thus, $P(z - z')$ is the solution of an instance $\tilde{A}$ of the SIS problem.
Thus, we have

$$\begin{aligned}
&\Pr[\mathrm{G}_1 \Rightarrow \mathsf{true}] \leq \mathsf{Adv}_{\mathsf{SIS},\mathcal{A}_{\mathrm{sis}}}(\kappa) + O(q^{-m}),\\
&\mathsf{Time}^*(\mathcal{A}_{\mathrm{sis}}) = \mathsf{Time}(\mathcal{A}) + O(k^3 \log^3 q),\\
&\mathsf{Mem}^*(\mathcal{A}_{\mathrm{sis}}) = \mathsf{Mem}(\mathcal{A}) + O(mk \log q).
\end{aligned}$$

Lemma B.3. *Let m, a be a positive inteters. We have*

$$\Pr_{D \leftarrow \mathbb{Z}_q^{m\times(m+a)}}[\mathrm{rank}(D) < m] \geq 2mq^{-(a+1)}.$$

While the above bound is well-known, we include the proof for completeness.

Proof. By the formula in [Bel93, MMO04, FG15], we have

$$\Pr_{D \leftarrow \mathbb{Z}_q^{m\times(m+a)}}[\mathrm{rank}(D) = m - r] = \frac{1}{q^{r(a+r)}} \cdot \frac{\prod_{i=1}^{m+a}(1 - q^{-i}) \prod_{i=r+1}^{m}(1 - q^{-i})}{\prod_{i=1}^{m-r}(1 - q^{-i}) \prod_{i=1}^{a+r}(1 - q^{-i})}.$$

Since we consider the case $r = 0$, the probability is

$$\begin{aligned}
\Pr_{D \leftarrow \mathbb{Z}_q^{m\times(m+a)}}[\mathrm{rank}(D) = m] &= \frac{\prod_{i=1}^{m+a}(1 - q^{-i}) \prod_{i=1}^{m}(1 - q^{-i})}{\prod_{i=1}^{m}(1 - q^{-i}) \prod_{i=1}^{a}(1 - q^{-i})} = \frac{\prod_{i=1}^{m+a}(1 - q^{-i})}{\prod_{i=1}^{a}(1 - q^{-i})}\\
&= \prod_{i=a+1}^{m+a}(1 - q^{-i}) \geq (1 - q^{-(a+1)})^m \geq 1 - 2mq^{-(a+1)}.
\end{aligned}$$

Thus, the lemma follows. □

C Relation Between Blinded Unforgeability and Plus-One Unforgeability

Alagic et al. showed that there exists a PO-secure but BU-insecure MAC scheme by assuming a random function or qPRF [AMRS20, Section 8.1]. In the original version of [AMRS20], Alagic et al. insisted that BU security implies PO security (for MAC), but this claim was retracted in Apr. 2023. They weakened their claim as that their BU security implies quadratic PO security, where an adversary is required to output ck^2 forgeries with probability 1 by making k quantum queries for a fixed constant c [AMRS18, Section 5.2.3].
Here, we give a simple example of BU-secure but PO-insecure signature assuming the existence of BU-secure signature. Our example exploits the fact that PO security is a quantum version of *strong existential unforgeability*, but BU security does not.

Lemma C.1 (BU $\not\Rightarrow$ PO**).** *Suppose that there exists a* BU*-secure MAC/signature scheme. We then have a* BU*-secure but* PO*-insecure MAC/signature scheme.*

In the proof, we only consider the signature schemes. The lemma for MAC is obtained similarly.

Proof. Suppose that we have a BU-secure $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ whose signature space is $\mathcal{S} \subseteq \{0,1\}^\lambda$ for some $\lambda = \lambda(\kappa)$. We construct a new BU-secure signature scheme $\mathsf{SIG}' = (\mathsf{Gen}, \mathsf{Sign}', \mathsf{Vrfy}')$ as follows:

- $\mathsf{Sign}'(sk, m)$: Generate $\sigma \leftarrow \mathsf{Sign}(sk, m)$, and output $\sigma' := (\sigma, 0)$.
- $\mathsf{Vrfy}'(vk, m, \sigma')$: Parse $\sigma' = (\sigma, b)$ with $b \in \{0,1\}$ and output $\mathsf{dec} := \mathsf{Vrfy}(vk, m, \sigma)$.

Note that the new signature space is $\mathcal{S}' \subseteq \{0,1\}^{\lambda+1}$.
(BU security:) This new signature scheme is still BU-secure because we can construct an adversary $\mathcal{A}$ breaking the BU security of SIG if there exists an adversary $\mathcal{A}'$ breaking the BU security of SIG'. $\mathcal{A}$ is defined as follows: On input vk, it runs $\mathcal{A}'$ on input vk. For a hash query to the random oracle, it passes the query to its random oracle and returns the result. It also implements the blinded signing oracle $|B_\epsilon \textsc{Sign}'\rangle$ for $\mathcal{A}'$ as follows: For a signing query $|m\rangle|y\rangle|y'_0\rangle|y'_1\rangle$ to $B_\epsilon \textsc{Sign}'$, where $y \in \{0,1\}^\lambda$, $y'_0, y'_1 \in \{0,1\}$,

1. query $|m\rangle|y\rangle|y'_1\rangle$ to its signing oracle $|B_\epsilon \textsc{Sign}\rangle$
2. receive $|m\rangle|y \oplus \sigma\rangle|y'_1 \oplus b_\sigma\rangle$, where $\sigma \| b_\sigma$ is $\sigma\|0$ or $\bot = 0^\lambda\|1$
3. return $|m\rangle|y \oplus \sigma\rangle|y'_0\rangle|y'_1 \oplus b_\sigma\rangle$.

This perfectly simulates $B_\epsilon \textsc{Sign}'$. If $\mathcal{A}'$ outputs m and (σ, b) with $b \in \{0,1\}$, $\mathcal{A}$ outputs m and σ as a forgery.
(PO insecurity:) On the other hand, this new signature scheme is PO-insecure: If we obtain a signature $\sigma' = (\sigma, 0)$ on a message m by querying to $|\textsc{Sign}\rangle$, we can output *two valid distinct pairs* of message and siganture, $(m, (\sigma, 0))$ and $(m, (\sigma, 1))$. □

Remark C.1. On their security definition of MAC, Boneh and Zhandry [BZ13a] wrote that "After issuing q quantum chosen message queries the adversary wins the game if it can generate $q+1$ valid classical message-tag pairs." just before Def.1 (EUF-qCMA). While there is an ambiguity of distinctness, we treat it as $q+1$ distinct pairs, since they reviewed sEUF-CMA security of MAC as the classical security definition.

Refuting that BU implies quadratic PO: The above example can be used to show that there exists BU-seucre but quadratic PO-insecure MAC, while Alagic et al. showed that BU implies quadratic PO [AMRS18].
Let $a = \omega(\lg(\kappa))$. Suppose we have a BU-secure MAC scheme $\mathsf{MAC} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$. We then construct a new BU-secure MAC scheme $\mathsf{MAC}' = (\mathsf{Gen}, \mathsf{Sign}', \mathsf{Vrfy}')$ as follows:

- $\mathsf{Sign}'(sk, m)$: Generate $\sigma \leftarrow \mathsf{Sign}(sk, m)$, and output $\sigma' := (\sigma, 0)$.
- $\mathsf{Vrfy}'(sk, m, \sigma')$: Parse $\sigma' = (\sigma, b)$ with $b \in \{0,1\}^a$ and output $\mathsf{dec} := \mathsf{Vrfy}(sk, m, \sigma)$.

This new signature scheme is still BU-secure because we can construct an adversary breaking the BU security of MAC if there exists an adversary breaking the BU security of MAC'. On the other hand, given k pairs of distinct messages and corresponding tags, it is easy to construct ck^2 ($\le k2^a$) distinct valid pairs of messages and tags when $ck \le 2^a = 2^{\omega(\lg(\kappa))}$.

Summary: As we exemplified, BU security does not imply PO security. What we should ask is the relation between sBU security and PO security and the relation between BU security and weakened PO security, where the adversary is required to output $q+1$ distinct messages and their corresponding signatures/tags.

D Blinded Unforgeability of Signature from Lossy Identification

The security proof for msEUF-CMA1 security can be used to show sBU security of $\mathsf{DFS}_{B,\mathsf{wz}}[\mathsf{LID}, \mathsf{H}, \mathsf{PRF}]$.

Theorem D.1 (sBU **security of** $\mathsf{DFS}_{B,\mathsf{wz}}[\mathsf{LID},\mathsf{H},\mathsf{PRF}]$). *Let $B \geq 0$. Let $\mathsf{H}\colon \mathcal{M}\times\mathcal{W}\to C$ be a hash function modeled as a random oracle. Let* LID *be a lossy identification scheme that is ρ-complete, ϵ_{zk}-HVZK, and ϵ_ℓ-lossy, and has (α,ϵ_m)-commitment min-entropy. Let* $\mathsf{DS} := \mathsf{DFS}_{B,\mathsf{wz}}[\mathsf{LID},\mathsf{H},\mathsf{PRF}]$.
Then, for a quantum adversary $\mathcal{A}$ breaking the sBU *security of* DS *that issues at most q_H quantum queries to the random oracle* H*, q_S classical queries to the signing oracle, and q_F classical queries to the forgery oracle, there exist a quantum $\mathcal{F}_{\mathrm{prf}}$-oracle adversary $\mathcal{A}_{\mathrm{prf}}$ against pseudorandomness of* PRF *and quantum $\mathcal{F}$-oracle adversaries $\mathcal{A}_{\mathrm{ind}}$ against key indistinguishability of* LID *and $\mathcal{A}_{\mathrm{cur}}$ against computationally unique response of* LID *such that*

$$\begin{aligned}
\mathsf{Adv}^{\mathrm{sbu}}_{\mathsf{DS},\mathcal{A}}(\kappa) &\leq \mathsf{Adv}^{\mathrm{pr}}_{\mathsf{PRF},\mathcal{A}_{\mathrm{prf}}}(\kappa) + \mathsf{Adv}^{\mathrm{cur}}_{\mathsf{LID},\mathcal{A}_{\mathrm{cur}}}(\kappa) + \mathsf{Adv}^{\text{ind-key}}_{\mathsf{LID},\mathcal{A}_{\mathrm{ind}}}(\kappa) + 8(q+1)^2(1-\rho') \\
&\quad + 8qB2^{-\frac{-\alpha-1}{2}} + 7\epsilon_m + 4\sqrt{(6q)^3 B\epsilon_{\mathrm{zk}}} + 2q_F 2^{-\alpha} + 8(q+1)^2\epsilon_\ell, \\
\mathsf{Time}^*(\mathcal{A}_{\mathrm{prf}}) &= \mathsf{Time}(\mathcal{A}) + (q_S+q_F)\cdot O(B\mathsf{Time}(\mathsf{LID}) + \mathsf{Time}(B_\epsilon)) \\
\mathsf{Mem}^*(\mathcal{A}_{\mathrm{prf}}) &= \mathsf{Mem}(\mathcal{A}) + O(\mathsf{Mem}(\mathsf{LID})) + O(\mathsf{Mem}(B_\epsilon)) \\
\mathsf{Time}^*(\mathcal{A}_{\mathrm{ind}}) &= \mathsf{Time}(\mathcal{A}) + q\cdot O(B\mathsf{Time}(\mathsf{LID}) + B^2 + \mathsf{Time}(B_\epsilon)) \\
\mathsf{Mem}^*(\mathcal{A}_{\mathrm{ind}}) &= \mathsf{Mem}(\mathcal{A}) + O(B\mathsf{Mem}(\mathsf{LID})) + O(\mathsf{Mem}(B_\epsilon)) \\
\mathsf{Time}^*(\mathcal{A}_{\mathrm{cur}}) &= \mathsf{Time}(\mathcal{A}) + q\cdot O(B\mathsf{Time}(\mathsf{LID}) + B^2 + \mathsf{Time}(B_\epsilon)) \\
\mathsf{Mem}^*(\mathcal{A}_{\mathrm{cur}}) &= \mathsf{Mem}(\mathcal{A}) + O(B\mathsf{Mem}(\mathsf{LID})) + O(\mathsf{Mem}(B_\epsilon)),
\end{aligned}$$

where $q = q_H + q_S + q_F$, ρ' is defined as in subsection 3.2, $\mathcal{F}_{\mathrm{prf}} = \mathsf{Func}(\mathcal{M}\times\mathcal{W}\times\mathcal{Z},\mathcal{P})\times\mathsf{Func}(\mathcal{M}\times\mathcal{W},C)$, and $\mathcal{F} = \mathsf{Func}(\mathcal{M}\times\mathcal{W}\times\mathcal{Z},\mathcal{P})\times\mathsf{Func}(\mathcal{M}\times\mathcal{W},C)\times\mathsf{Func}(\mathcal{M}\times[B],C)\times\mathsf{Func}(\mathcal{M}\times[B],\mathcal{R}_{\mathsf{Sim}})$.

Roadmap: We define thirteen games G_i for $i\in\{0,1,\dots,12\}$ to show our theorem. Let W_i denote the event that the experiment outputs true in G_i.
The proof of sBU security involves *quantum* singing oracle and the filter B_ϵ. Fortunately, we can take the same approach as the proof of msEUF-CMA1 security (Theorem 4.1).
The original security game is denoted by G_0, in which the prover in GetTrans is derandomized by PRF. Hence, we replace this PRF with RF in G_1. We modify the games as in the previous proof. In G_7, GetTrans is simulated by the simulator Sim and the winning condition is $\mathsf{V}(vk,w^*,c^*,z^*)$, where $c^* = \mathsf{H}(m^*,w^*)$, and $(m^*,(w^*,z^*))\in B_\epsilon$. In G_8, we modify the winning condition as whether $\mathsf{V}(vk,w^*,c^*,z^*)$, where $c^* = \mathsf{H}(m^*,w^*)$, and $(m^*,(w^*,z^*))\in B_\epsilon$, *and* $(w^*,z^*)\neq(\tilde{w}^{(k)},\tilde{z}^{(k)})$ or not. We can argue that this modification introduces only a negligible change, as in the previous proof. After that, we continue to modify the games until G_{11}. We finally replace a normal verification key vk with a lossy one in G_{12}.

Game G_0*:* This is the original game. See Figure 14 for a concrete definition of G_0, where we expand the Sign algorithm and H is implemented by a random function RF_{H}. We have

$$\Pr[W_0] = \mathsf{Adv}^{\mathrm{sbu}}_{\mathsf{DS},\mathcal{A}}(\kappa).$$

Game G_1*:* We replace PRF with RF_{P} in the prover in GetTrans.
By straightforward argument, we have the following lemma:

Lemma D.1. *There exists a quantum $\mathcal{F}$-oracle adversary $\mathcal{A}_{\mathrm{prf}}$ such that*

$$\begin{aligned}
&|\Pr[W_0]-\Pr[W_1]| \leq \mathsf{Adv}^{\mathrm{pr}}_{\mathsf{PRF},\mathcal{A}_{\mathrm{prf}}}(\kappa), \\
&\mathsf{Time}^*(\mathcal{A}_{\mathrm{cur}}) = \mathsf{Time}(\mathcal{A}) + (q_S+q_F)\cdot O(B\mathsf{Time}(\mathsf{LID}) + \mathsf{Time}(B_\epsilon)) \\
&\mathsf{Mem}^*(\mathcal{A}_{\mathrm{cur}}) = \mathsf{Mem}(\mathcal{A}) + O(\mathsf{Mem}(\mathsf{LID})) + O(\mathsf{Mem}(B_\epsilon)),
\end{aligned}$$

where $\mathcal{F} = \mathsf{Func}(\mathcal{M}\times C\times\mathcal{Z},\mathcal{P})\times\mathsf{Func}(\mathcal{M}\times\mathcal{W},C)$.

Game G_2*:* We next let GetTrans output *all* transcripts instead of the last one. The signing oracle also takes the last one as a candidate for a signature. We have

$$\mathrm{G}_1 = \mathrm{G}_2.$$

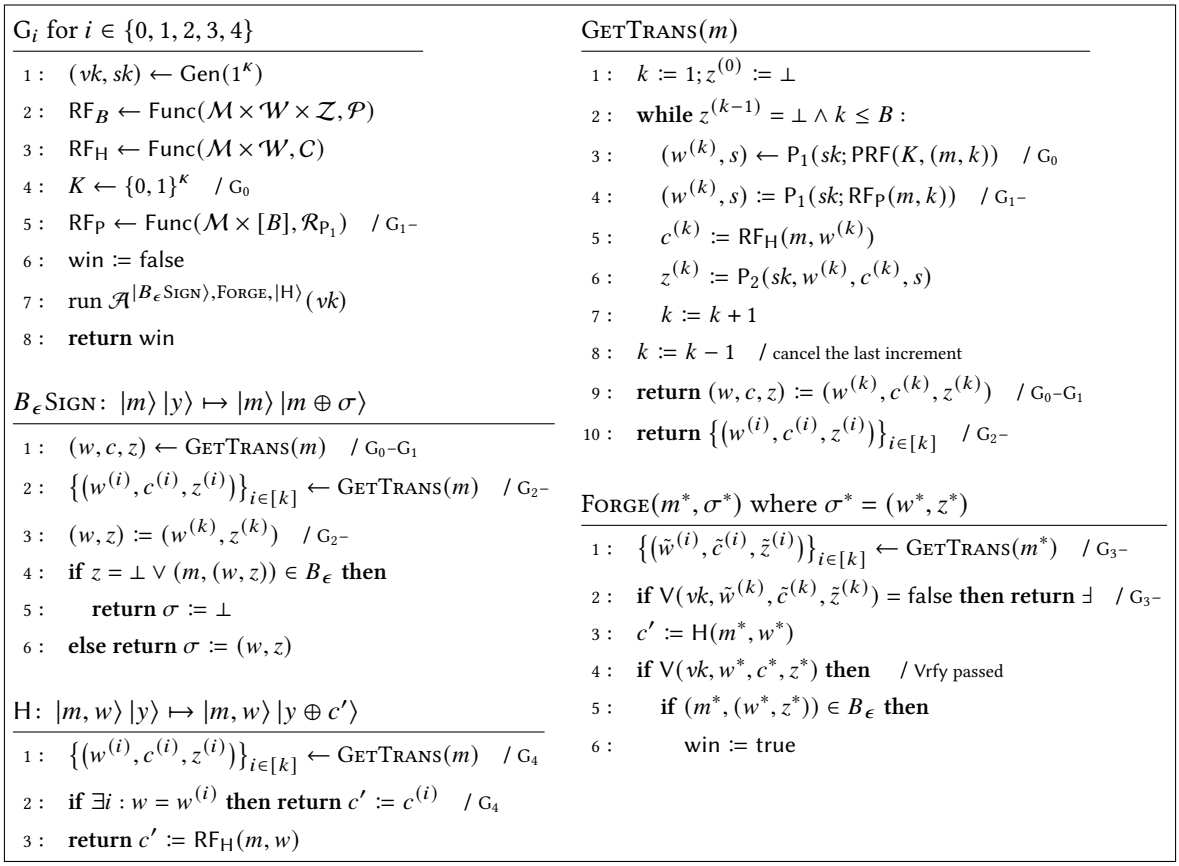

Fig. 14. G_i for $i \in \{0,1,2,3,4\}$

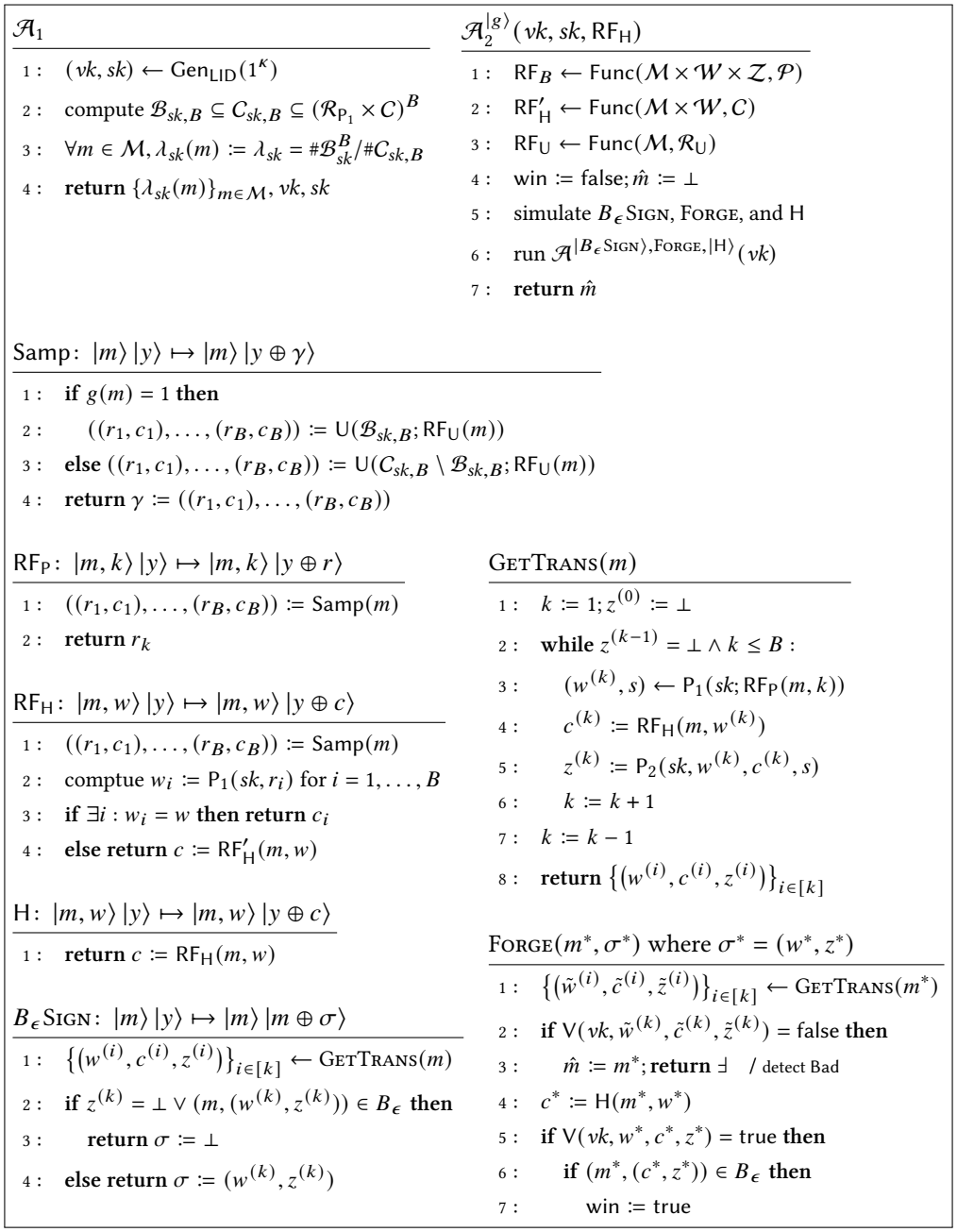

Fig. 15. Adversary $\mathcal{A}_{\text{gspb}} = (\mathcal{A}_1, \mathcal{A}_2)$ against GSPB. The set of consistent sequences $C_{sk,B}$, the set of bad sequences $\mathcal{B}_{sk,B}$, and an algorithm U are defined in the proof text.

Game G_3*:* We next modify the forge oracle as follows: Before checking the validity of submitted query (m^*, σ^*), it generates its own signature $(\tilde{w}^{(k)}, \tilde{c}^{(k)}, \tilde{z}^{(k)})$ by using $\text{GetTrans}(m^*)$. If $\text{GetTrans}(m^*)$ fails to output a valid signature, then the forge oracle returns the special symbol $\exists$.

Lemma D.2. *Let* Bad *be the event that the oracle* Forge *returns the symbol* $\exists$*. Suppose that* LID *is* (γ, β)*-correct and has* (α, ϵ_m)*-commitment min-entropy. We have*

$$|\Pr[W_2] - \Pr[W_3]| \le \Pr[\text{Bad}] \le 8(q_S + q_F + q_H + 1)^2(1 - \rho') + \epsilon_m$$

We give the proof since we omitted the corresponding proof in PO security (Lemma 5.2).

Proof. To make a proof simple, we first eliminate the bad event that the key generation algorithm outputs a bad key pair (vk, sk) making the min-entropy of commitment less than α. This elimination introduces an additional difference ϵ_m.
We define some terminology. In order to simulate RF_P and RF_H, we consider an algorithm Samp that takes B samples of a pair of randomness of P_1 and challenge in $(\mathcal{R}_{\mathsf{P}_1} \times C)^B$. For a signing key sk, we say that a sequence of B-samples $((r_1, c_1), \ldots, (r_B, c_B)) \in (\mathcal{R}_{\mathsf{P}_1} \times C)^B$ is *consistent* if

$$\forall i, j \in [B] : w_i = w_j \Longrightarrow c_i = c_j, \text{ where } (w_i, s_i) := \mathsf{P}_1(sk, r_i).$$

Let $C_{sk,B}$ be the set of all consistent sequences. We say that a consistent sequence is *bad* if 1) the signing algorithm using it fails to generate a signature with $z \neq \perp$ or 2) the signing algorithm using it succeeds to output a signature which is invalid. Let $\mathcal{B}_{sk,B}$ be the set of all bad sequences. Formally, it is defined as

$$\mathcal{B}_{sk,B} := \left\{ ((r_1, c_1), \ldots, (r_B, c_B)) \in C_{sk,B} \mid \begin{array}{c} (w_i, s_i) := \mathsf{P}_1(sk, r_i), z_i := \mathsf{P}_2(sk, w_i, c_i, s_i) : \\ (\forall i \in [B] : z_i = \perp) \vee (\exists i \in [B] : z_i \neq \perp \wedge \mathsf{V}(vk, w_i, c_i, z_i) = \text{false}) \end{array} \right\}.$$

By the definition of (γ, β)-correctness of LID and the discussion in subsection 3.2, we have

$$\operatorname*{Exp}_{(vk,sk)} \left[\#\mathcal{B}_{sk,B} / \#C_{sk,B} \right] \le 1 - \rho'.$$

For a finite set $\mathcal{S}$, U is a probabilistic sampling algorithm that returns $s \leftarrow \mathcal{S}$. For convenience, we define the output of $\mathsf{U}(\emptyset)$ as $\perp$.[4]
Let us construct an unbounded adversary $\mathcal{A}_{\text{gspb}} = (\mathcal{A}_1, \mathcal{A}_2)$ against GSPB defined in Figure 15. The first adversary $\mathcal{A}_1$ outputs a set of bounds $\{\lambda_{sk}(m)\}$, vk, sk, where $\lambda_{sk}(m) = \lambda_{sk} = \#\mathcal{B}_{sk,B}/\#C_{sk,B}$. The value of function g on m is selected according to $\text{Ber}_{\lambda_{sk}}$. The second adversary $\mathcal{A}_2$ tries to output m^* on which GetTrans fails to output a valid signature.
We first consider the success probability of $\mathcal{A}_{\text{gspb}}$ by *fixing* (vk, sk) *and* H. Let us verify the distributions of $r_1, \ldots, r_B$ in RF_P and $c_1, \ldots, c_B \in \mathsf{RF}_\mathsf{H}$. In Samp, if we took a random sample of a sequence from the set $C_{sk,B}$, then the distributions were perfectly simulated. Instead of this, we check the value of $g(m)$, and if it is 1, then we take a bad sequence uniformly at random; otherwise, we take a good sequence uniformly at random. Since the probablity that $g(m)$ takes 1 with probability $\lambda_{sk} = \#\mathcal{B}_{sk,B}/\#C_{sk,B}$, the distribution of Samp is perfect and the distribution of RF_P and RF_H are the same as those in G_2 and G_3.
We then check $\mathcal{A}$'s forgery. If $\hat{m}$ is set as m^*, then the adversary submits m^* such that m^* induces a bad sequence. Hence, $g(m^*) = 1$ and $\mathcal{A}_{\text{gspb}}$ wins the game. Thus, we have

$$\Pr[\text{Bad} \mid vk, sk] = \Pr[\text{GSPB}_{\lambda_{sk}, \mathcal{A}_{\text{gspb}}} = 1 \mid vk, sk] \le 8(q + 1)^2 \lambda_{sk},$$

where q is the number of queries to g of $\mathcal{A}_2$, which is $q \le (q_S + q_F) \cdot (\#\text{ of queries by GetTrans}) + q_H \le (q_S + q_F) \cdot 2B + q_H$. We note that we can reduce the number of queries to g by preparing $((r_1, c_1), \ldots, (r_B, c_B)) := \text{Samp}(m)$ at the first steps of GetTrans and Forge and the best bound is $q \le q_S + q_F + q_H$.
Averaging this inequality over keys, we obtain

$$\Pr[\text{Bad}] \le 8(q + 1)^2 \cdot \operatorname*{Exp}_{(vk,sk)} [\lambda_{sk}] \le 8(q + 1)^2 \cdot (1 - \rho')$$

Since $|\Pr[W_2] - \Pr[W_3]| \le \Pr[\text{Bad}]$, we obtain the bound in the lemma as we wanted. □

Game G_4*:* We next modify the random oracle as follows: On a query (m, w), the oracle first computes the transcripts. If the input w is equivalent to one of $w^{(i)}$, then it returns $c' := c^{(i)}$; otherwise, it returns $c' := \mathsf{RF}_\mathsf{H}(m, w)$. See G_4 in Figure 14 (and Figure 16) for the details. Since $c^{(i)} = \mathsf{RF}_\mathsf{H}(m, w^{(i)})$ in GetTrans, this modification changes nothing and we have

$$G_3 = G_4.$$

[4] But, this never occurs.

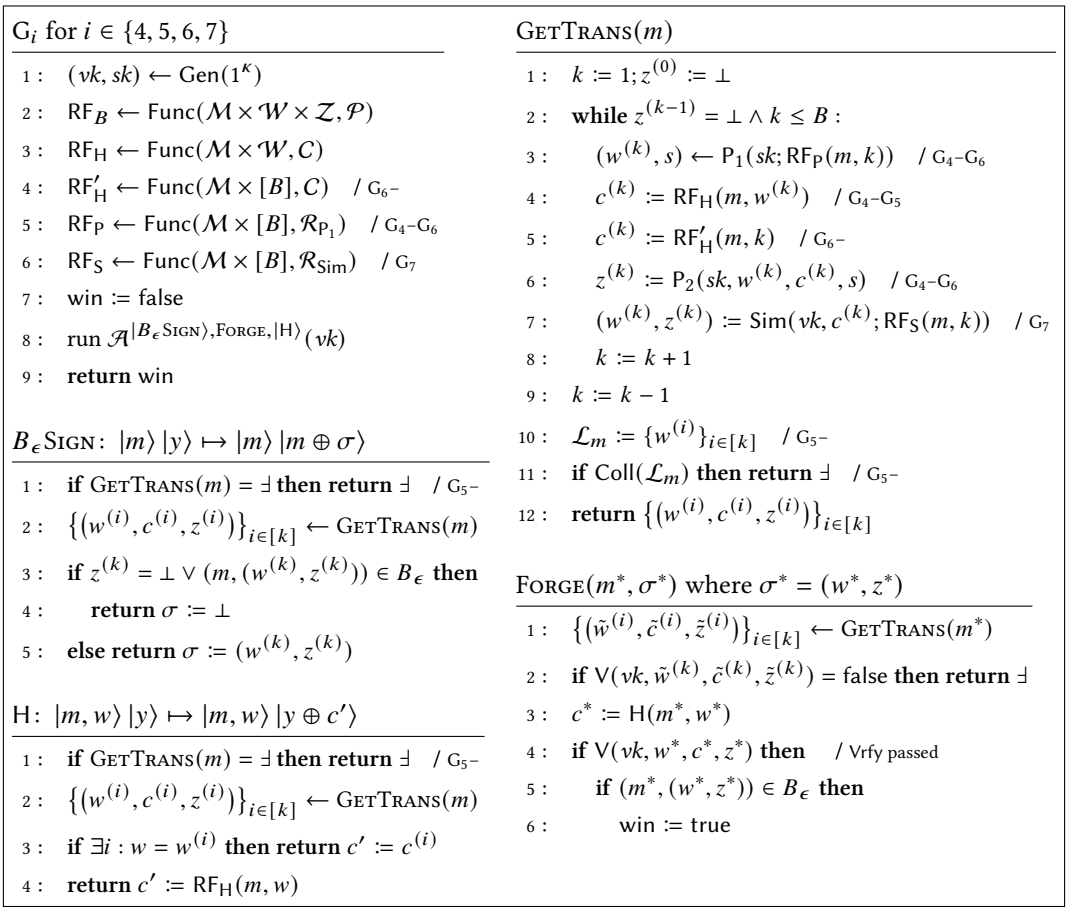

Fig. 16. G_i for $i \in \{4, 5, 6, 7\}$

G_i for $i \in \{8, 9, 10, 11, 12\}$

$(vk, sk) \leftarrow \mathsf{Gen}_{\mathsf{LID}}(1^\kappa)$ / G_8–G_{11}
$vk \leftarrow \mathsf{LossyGen}_{\mathsf{LID}}(1^\kappa)$ / G_{12}
$\mathsf{RF}_B \leftarrow \mathsf{Func}(\mathcal{M} \times \mathcal{W} \times \mathcal{Z}, \mathcal{P})$
$\mathsf{RF}_{\mathsf{H}} \leftarrow \mathsf{Func}(\mathcal{M} \times \mathcal{W}, C)$
$\mathsf{RF}'_{\mathsf{H}} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], C)$
$\mathsf{RF}_{\mathsf{S}} \leftarrow \mathsf{Func}(\mathcal{M} \times [B], \mathcal{R}_{\mathsf{Sim}})$
win $\coloneqq$ false
run $\mathcal{A}^{|B_\epsilon \textsc{Sign}\rangle, \textsc{Forge}, |\mathsf{H}\rangle}(vk)$
return win

$\textsc{Forge}(m^*, \sigma^*)$ where $\sigma^* = (w^*, z^*)$

if $\textsc{GetTrans}(m^*) = \exists$ **then return** $\exists$
$\{(\tilde{w}^{(i)}, \tilde{c}^{(i)}, \tilde{z}^{(i)})\}_{i \in [k]} \leftarrow \textsc{GetTrans}(m^*)$
if $\mathsf{V}(vk, \tilde{w}^{(k)}, \tilde{c}^{(k)}, \tilde{z}^{(k)}) = \mathsf{false}$ **then return** $\exists$
$\mathcal{L}_{m^*} \coloneqq \{w^{(i)}\}_{i \in [k]}; \mathcal{L}'_{m^*} \coloneqq \{w^{(i)}\}_{i \in [k-1]}$
if $\exists i : w^* = \tilde{w}^{(i)}$ **then** $c^* \coloneqq \tilde{c}^{(i)}$ **else** $c^* \coloneqq \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$
if $\mathsf{V}(vk, w^*, c^*, z^*) = \mathsf{true} \wedge (m^*, (c^*, z^*)) \in B_\epsilon$
  $\wedge\ (c^*, z^*) \neq (\tilde{c}^{(k)}, \tilde{z}^{(k)})$ **then**
  win $\coloneqq$ true / G_8
  if $(w^* \notin \mathcal{L}'_{m^*}) \vee (w^* \in \mathcal{L}'_{m^*} \wedge c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*))$ **then** / G_9
    win $\coloneqq$ true / G_9
  if $(w^* \notin \mathcal{L}_{m^*}) \vee (w^* \in \mathcal{L}'_{m^*} \wedge c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*))$ **then** / G_{10}
    win $\coloneqq$ true / G_{10}
  if $w^* \neq \tilde{w}^{(k)} \wedge c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ **then** / G_{11}–G_{12}
    win $\coloneqq$ true / G_{11}–G_{12}

Fig. 17. G_i for $i \in \{8, 9, 10, 11, 12\}$. We expand the computation of H in Forge. The definitions of B_ϵSign, H, GetTrans are those in G_7 in Figure 16.

Game G_5*:* The next game introduces a collision check for $w^{(i)}$'s in GetTrans. Since the min-entropy of $w^{(i)}$ is α-bit with probability $1 - \epsilon_m$, on each invocation of GetTrans, the collision occurs with probability at most $B^2 \cdot 2^{-\alpha-1}$. As Lemma 4.1, we have the following lemma using the one-sided O2H lemma.

Lemma D.3. *Suppose that* LID *has* (α, ϵ_m)*-commitment min-entropy. Then, we have that*

$$|\Pr[W_4] - \Pr[W_5]| \leq \delta_{4,5} \coloneqq 2(q_S + q_H + q_F) \cdot B \cdot 2^{\frac{-\alpha-1}{2}} + \epsilon_m.$$

Game G_6*:* We next modify how to compute $c^{(k)}$ in GetTrans, in which it is computed as $\mathsf{RF}'_{\mathsf{H}}(m, k)$ instead of $\mathsf{RF}_{\mathsf{H}}(w^{(k)}, m)$. We note that this does not change the adversary's view because $\mathsf{RF}'_{\mathsf{H}}$ is a random function, and if $w = w^{(i)}$ for the query (m, w), then consistent $c' = c^{(i)} = \mathsf{RF}'_{\mathsf{H}}(m, i)$ is output by H. (Note that excluding the collision is crucial [DFPS23].)

Game G_7*:* We next modify GetTrans to use the simulation algorithm. See G_7 in Figure 7 for the details. As Lemma 4.2, we have the following lemma.

Lemma D.4. *Suppose that* LID *is* ϵ_{zk}*-HVZK. Then, we have*

$$|\Pr[W_6] - \Pr[W_7]| \leq \delta_{6,7} \coloneqq \sqrt{(6(q_S + q_H + q_F))^3 B \epsilon_{\mathrm{zk}}}.$$

Game G_8*:* We next modify the winning condition in Forge: After checking $\mathsf{V}(vk, w^*, c^*, z^*)$, where $c^* = \mathsf{H}(m^*, w^*)$, it sets flag win as true if $(m^*, (w^*, z^*)) \notin B_\epsilon$ *and* $(w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$. See G_8 in Figure 17. We note that this modified condition equals that introduced in G_8 for the msEUF-CMA1 security (Theorem 4.1). Thus, similarly, we obtain the following lemma.

Lemma D.5. *Suppose that* LID *is* ϵ_{zk}*-HVZK and has* (α, ϵ_m)*-commitment min-entropy. Then, we have*

$$|\Pr[W_7] - \Pr[W_8]| \leq \delta_{7,8} \coloneqq q_F \cdot 2^{-\alpha} + \epsilon_m + \delta_{4,5} + \delta_{6,7}.$$

Proof. The difference occurs if the adversary queries a valid pair of message and signature $(m^*, (w^*, z^*)) \in B_\epsilon$ such that $(w^*, z^*) = (\tilde{w}^{(k)}, \tilde{z}^{(k)})$. If $(m^*, (\tilde{w}^{(k)}, \tilde{z}^{(k)}))$ is not in B_ϵ, then this contradicts with the requirement $(m^*, (w^*, z^*)) \in B_\epsilon$. Thus, $(m^*, (\tilde{w}^{(k)}, \tilde{z}^{(k)}))$ should be blinded by B_ϵ. This means that the adversary cannot obtain the signature $(\tilde{w}^{(k)}, \tilde{z}^{(k)})$ on m^* from the blinded signing oracle B_ϵSign. Thus, the adversary succeeds to guess $w^* = \tilde{w}^{(k)}$ *without knowing* $\tilde{w}^{(k)}$.
Let Bad_i be the event that in G_i the adversary submit $(m^*, (w^*, z^*))$ such that $\mathsf{V}(vk, w^* c, ^*, z^*) = \mathsf{true}$, $(m^*, (w^*, z^*)) \in B_\epsilon$, and $(w^*, z^*) = (\tilde{w}^{(k)}, \tilde{z}^{(k)})$ which implies $w^* = \tilde{w}^{(k)}$. We then have $\Pr[\mathsf{Bad}_3] =$

$q_F \cdot 2^{-\alpha} + \epsilon_m$ since in G_2 there is no leakage of $\tilde{w}^{(k)}$ from H and LID has (α, ϵ_m)-commitment min-entropy. As in the proof of Lemma 4.3, we have

$$\begin{aligned}|\Pr[W_7] - \Pr[W_8]| &\le \Pr[\mathsf{Bad}_7] \le |\Pr[\mathsf{Bad}_7] - \Pr[\mathsf{Bad}_3]| + \Pr[\mathsf{Bad}_3] \\ &\le \delta_{4,5} + \delta_{6,7} + q_F \cdot 2^{-\alpha} + \epsilon_m\end{aligned}$$

as we wanted. □

Game G_9*:* To ease the notation, let $\mathcal{L}_{m^*} := \{w^{(i)}\}_{i\in[k]}$ which are the w parts of the transcripts generated by $\textsc{GetTrans}(m^*)$. We additionally define $\mathcal{L}'_{m^*} := \{w^{(i)}\}_{i\in[k-1]}$.

Lemma D.6. *We have that*

$$|\Pr[W_8] - \Pr[W_9]| \le \delta_{4,5} + \delta_{6,7} + \delta_{7,8}.$$

Proof. In G_9, $\textsc{Forge}$ additionally check if $w^* \in \mathcal{L}'_{m^*}$ or not; if so, we additionally checks whether $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ or not. The two games may differ if the adversary queries $w^* = w^{(i)}$ for $i < k$ but $c^* \neq \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ in G_8. We call this event Bad_i in G_i. As in the proof of Lemma 4.4, we have

$$\begin{aligned}|\Pr[W_8] - \Pr[W_9]| &\le \Pr[\mathsf{Bad}_8] \le |\Pr[\mathsf{Bad}_8] - \Pr[\mathsf{Bad}_3]| + \Pr[\mathsf{Bad}_3] \\ &\le \delta_{4,5} + \delta_{6,7} + \delta_{7,8} + \Pr[\mathsf{Bad}_3].\end{aligned}$$

Notice that, in G_3, $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ always holds and Bad_3 never occurs. Thus, we obtain the bound as we wanted. □

Game G_{10}*:* We then treat the case $w^* = \tilde{w}^{(k)}$ as a special case to exclude CUR. To do so, we replace the condition $w^* \notin \mathcal{L}'_{m^*}$ with $w^* \notin \mathcal{L}_{m^*}$. See G_{10} in Figure 17 for the details.
Because of this modification, if the adversary queries $(m^*, (w^*, z^*))$ satisfying $(w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$, $w^* = \tilde{w}^{(k)}$, then two games differ. This is easily treated by the CUR property.

Lemma D.7. *There exists a quantum $\mathcal{F}$-oracle adversary $\mathcal{A}_{\mathrm{cur}}$ such that*

$$\begin{aligned}&|\Pr[W_9] - \Pr[W_{10}]| \le \mathsf{Adv}^{\mathrm{cur}}_{\mathsf{LID},\mathcal{A}_{\mathrm{cur}}}(\kappa), \\ &\mathsf{Time}^*(\mathcal{A}_{\mathrm{cur}}) = \mathsf{Time}(\mathcal{A}) + (q_H + q_S + q_F) \cdot O(B\mathsf{Time}(\mathsf{LID}) + B^2 + \mathsf{Time}(B_\epsilon)), \\ &\mathsf{Mem}^*(\mathcal{A}_{\mathrm{cur}}) = \mathsf{Mem}(\mathcal{A}) + O(B\mathsf{Mem}(\mathsf{LID})) + O(\mathsf{Mem}(B_\epsilon)),\end{aligned}$$

where $\mathcal{F} = \mathsf{Func}(\mathcal{M} \times C \times \mathcal{Z}, \mathcal{P}) \times \mathsf{Func}(\mathcal{M} \times \mathcal{W}, C) \times \mathsf{Func}(\mathcal{M} \times [B], C) \times \mathsf{Func}(\mathcal{M} \times [B], \mathcal{R}_{\mathsf{Sim}})$.

Since the proof is straightforwardly obtained, we omit it.

Game G_{11}*:* We again modify the conditions in $\textsc{Forge}$ in G_{10}: $\textsc{Forge}$ checks if $(m^*, (w^*, z^*)) \in B_\epsilon$, $(w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$, $w^* \neq \tilde{w}^{(k)}$, and $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ or not. If so, the flag is set as true. See G_{11} in Figure 17 for the details.

Lemma D.8. *We have* $G_{10} = G_{11}$.

Proof. Let us consider a valid forgery $(m^*, (w^*, z^*)) \in B_\epsilon$ satisfying $(w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$.
If $w^* \in \mathcal{L}'_{m^*}$, then there is no diference on the condition $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$ in both games. If $w^* = \tilde{w}^{(k)}$, then win is kept the same in both games. If $w^* \notin \mathcal{L}_{m^*}$, then we have $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$; both flags in G_{10} and G_{11} are set true because $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$. Summarizing those three cases, both games are the same. □

Game G_{12}*:* We finally replace a normal verification key with a lossy verification key. See G_{12} in Figure 17 for the details.

Lemma D.9. *There exists a quantum $\mathcal{F}$-oracle adversary $\mathcal{A}_{\mathrm{ind}}$ such that*

$$\begin{aligned}&|\Pr[W_{11}] - \Pr[W_{12}]| \le \mathsf{Adv}^{\mathrm{indkey}}_{\mathsf{LID},\mathcal{A}_{\mathrm{ind}}}(\kappa), \\ &\mathsf{Time}^*(\mathcal{A}_{\mathrm{ind}}) = \mathsf{Time}(\mathcal{A}) + (q_H + q_S + q_F) \cdot O(B\mathsf{Time}(\mathsf{LID}) + B^2 + \mathsf{Time}(B_\epsilon)), \\ &\mathsf{Mem}^*(\mathcal{A}_{\mathrm{ind}}) = \mathsf{Mem}(\mathcal{A}) + O(B\mathsf{Mem}(\mathsf{LID})) + O(\mathsf{Mem}(B_\epsilon)),\end{aligned}$$

where $\mathcal{F} = \mathsf{Func}(\mathcal{M} \times C \times \mathcal{Z}, \mathcal{P}) \times \mathsf{Func}(\mathcal{M} \times \mathcal{W}, C) \times \mathsf{Func}(\mathcal{M} \times [B], C) \times \mathsf{Func}(\mathcal{M} \times [B], \mathcal{R}_{\mathsf{Sim}})$.

Since the proof is obtained by a straightforward reduction, we omit it.

$\mathcal{A}_1$

$vk \leftarrow \mathsf{LossyGen}_{\mathsf{LID}}(1^\kappa)$
foreach $w \in \mathcal{W}$:
  compute set $\mathcal{G}_{vk}(w) \subseteq C$
  $\lambda'_{vk}(w) \coloneqq \#\mathcal{G}_{vk}(w)/\#C$
  foreach $m \in \mathcal{M} : \lambda_{vk}(m,w) \coloneqq \lambda'_{vk}(w)$
return $\{\lambda_{vk}(m,w)\}_{m\in\mathcal{M},w\in\mathcal{W}}, vk$

$\mathsf{RF}_\mathsf{H} \colon |m,w\rangle|y\rangle \mapsto |m,w\rangle|y \oplus c\rangle$

if $g(m,w) = 1$ **then**
  $c \coloneqq \mathsf{U}(\mathcal{G}_{vk}(w); \mathsf{RF}_\mathsf{U}(m,w))$
else $c \coloneqq \mathsf{U}(C \setminus \mathcal{G}_{vk}(w); \mathsf{RF}_\mathsf{U}(m,w))$
return c

B_ϵSIGN$\colon |m\rangle|y\rangle \mapsto |m\rangle|m \oplus \sigma\rangle$

if GETTRANS$(m) = \lightning$ **then return** $\lightning$
$\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i\in[k]} \leftarrow$ GETTRANS(m)
if $z^{(k)} = \bot \vee (m, (w^{(k)}, z^{(k)})) \in B_\epsilon$ **then**
  return $\sigma \coloneqq \bot$
else return $\sigma \coloneqq (w^{(k)}, z^{(k)})$

$\mathsf{H} \colon |m,w\rangle|y\rangle \mapsto |m,w\rangle|y \oplus c'\rangle$

if GETTRANS$(m) = \lightning$ **then return** $\lightning$
$\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i\in[k]} \leftarrow$ GETTRANS(m)
if $\exists i : w = w^{(i)}$ **then return** $c' \coloneqq c^{(i)}$
return $c' \coloneqq \mathsf{RF}_\mathsf{H}(m,w)$

$\mathcal{A}_2^{|g\rangle}(vk)$

$\mathsf{RF}_B \leftarrow \mathsf{Func}(\mathcal{M}\times\mathcal{W}\times\mathcal{Z}, \mathcal{P})$
$\mathsf{RF}_\mathsf{U} \leftarrow \mathsf{Func}(\mathcal{M}\times\mathcal{W}, \mathcal{R}_\mathsf{U})$
$\mathsf{RF}'_\mathsf{H} \leftarrow \mathsf{Func}(\mathcal{M}\times[B], C)$
$\mathsf{RF}_\mathsf{S} \leftarrow \mathsf{Func}(\mathcal{M}\times[B], \mathcal{R}_\mathsf{Sim})$
win $\coloneqq$ false; $\hat{m} \coloneqq \bot; \hat{w} \coloneqq \bot$
simulate B_ϵ SIGN, FORGE, H, and RF_H
run $\mathcal{A}^{|B_\epsilon \text{SIGN}\rangle, \text{FORGE}, |\mathsf{H}\rangle}(vk)$
if win = true **then**
  return $(\hat{m}, \hat{w})$
else return $\bot$

GETTRANS(m)

$k \coloneqq 1; z^{(0)} \coloneqq \bot$
while $z^{(k-1)} = \bot \wedge k \le B$:
  $c^{(k)} \coloneqq \mathsf{RF}'_\mathsf{H}(m,k)$
  $(w^{(k)}, z^{(k)}) \coloneqq \mathsf{Sim}(vk, c^{(k)}; \mathsf{RF}_\mathsf{S}(m,k))$
  $k \coloneqq k+1$
$k \coloneqq k-1$
$\mathcal{L}_m \coloneqq \{w^{(i)}\}_{i\in[k]}$
if Coll$(\mathcal{L}_m)$ **then return** $\lightning$
return $\{(w^{(i)}, c^{(i)}, z^{(i)})\}_{i\in[k]}$

FORGE(m^*, σ^*) where $\sigma^* = (w^*, z^*)$

if GETTRANS$(m^*) = \lightning$ **then return** $\lightning$
$\{(\tilde{w}^{(i)}, \tilde{c}^{(i)}, \tilde{z}^{(i)})\}_{i\in[k]} \leftarrow$ GETTRANS(m^*)
if $\mathsf{V}(vk, \tilde{w}^{(k)}, \tilde{c}^{(k)}, \tilde{z}^{(k)}) =$ false **then return** $\lightning$
$\mathcal{L}_{m^*} \coloneqq \{w^{(i)}\}_{i\in[k]}; \mathcal{L}'_{m^*} \coloneqq \{w^{(i)}\}_{i\in[k-1]}$
if $\exists i : w^* = \tilde{w}^{(i)}$ **then**
  return $c^* \coloneqq \tilde{c}^{(i)}$ **else** $c^* \coloneqq \mathsf{RF}_\mathsf{H}(m^*, w^*)$
if $\mathsf{V}(vk, w^*, c^*, z^*) =$ true $\wedge (m^*, (c^*, z^*)) \in B_\epsilon$
  $\wedge (w^*, z^*) \neq (\tilde{w}^{(k)}, \tilde{z}^{(k)})$ **then**
  if $w^* \neq \tilde{w}^{(k)} \wedge c^* = \mathsf{RF}_\mathsf{H}(m^*, w^*)$ **then**
    win $\coloneqq$ true; $\hat{m} \coloneqq m^*; \hat{w} \coloneqq w^*$

Fig. 18. Adversary $\mathcal{A}_{\text{gspb}} = (\mathcal{A}_1, \mathcal{A}_2)$ against GSPB. The set of good challenges $\mathcal{G}_{vk}(w)$ and an algorithm U are defined in the proof text.

Lemma D.10. *Suppose that* LID *is* ϵ_ℓ*-lossy. Then, we have*

$$\Pr[W_{12}] \le 8(q_S + q_H + q_F + 1)^2 \epsilon_\ell .$$

While we omit the proofs for the cases of msEUF-CMA1 and PO securities because the proofs are the same as in [KLS18], we here include the proof for sBU security for completeness.
Before giving the proof, we review some terminology.
For a verification key vk and commitment $w \in \mathcal{W}$, we define the set of good challenges as

$$\mathcal{G}_{vk}(w) := \{c \in C \mid \exists z \in \mathcal{Z} : \mathsf{V}(vk, w, c, z) = \mathsf{true}\}. \tag{1}$$

In [KLS18, Section 2.3], Kiltz et al. discussed that

$$\begin{aligned}\mathsf{Adv}^{\mathrm{imp}}_{\mathsf{LID},\mathcal{A}}(\kappa) &\le \operatorname*{Exp}_{vk \leftarrow \mathsf{LossyGen}_{\mathsf{LID}}(1^\kappa)} \left[\max_{w \in \mathcal{W}} \left(\Pr_{c \leftarrow C}[\exists z \in \mathcal{Z} : \mathsf{V}(vk, w, c, z) = \mathsf{true}] \right) \right] \\ &= \operatorname*{Exp}_{vk \leftarrow \mathsf{LossyGen}_{\mathsf{LID}}(1^\kappa)} \left[\max_{w \in \mathcal{W}} (\#\mathcal{G}_{vk}(w)/\#C) \right]\end{aligned}$$

and the equality holds when the adversary is optimal by choosing the best $w \in \mathcal{W}$. Thus, if LID is ϵ_ℓ-losy, we have

$$\operatorname*{Exp}_{vk \leftarrow \mathsf{LossyGen}_{\mathsf{LID}}(1^\kappa)} \left[\max_{w \in \mathcal{W}} (\#\mathcal{G}_{vk}(w)/\#C) \right] \le \epsilon_\ell . \tag{2}$$

Proof. We follow the proof by Kiltz et al. [KLS18, Theorem 3.4]. For a finite set $\mathcal{S}$, U is a probabilistic sampling algorithm that returns $s \leftarrow \mathcal{S}$. For convenience, we define the output of $\mathsf{U}(\emptyset)$ as $\perp$.[5]
Let us construct an unbounded adversary $\mathcal{A}_{\mathrm{gspb}} = (\mathcal{A}_1, \mathcal{A}_2)$ against GSPB. The first adversary $\mathcal{A}_1$ outputs a set of bounds $\{\lambda_{vk}(m, w)\}$ and vk. The value of function g on (m, w) is selected according to $\mathsf{Ber}_{\lambda_{vk}(m,w)}$. The second adversary $\mathcal{A}_2$ tries to output (m^*, w^*) as in Figure 18. We first consider the success probability of $\mathcal{A}_{\mathrm{gspb}}$ by fixing vk. Let us verify the distribution of c in L.1–3 in RF_{H}. We note that $g(m, w) = 1$ with probability $\lambda_{vk}(m, w) = \#\mathcal{G}_{vk}(w)/\#C$. We have

$$\Pr[c = \tilde{c}] = \begin{cases} \lambda_{vk}(m, w) \cdot \frac{1}{\#\mathcal{G}_{vk}(w)} & (c \in \#\mathcal{G}_{vk}(w)) \\ (1 - \lambda_{vk}(m, w)) \cdot \frac{1}{\#C - \#\mathcal{G}_{vk}(w)} & \text{o.w.,} \end{cases}$$

which is $1/\#C$ in both cases. Hence, the distribution of c in L.1–3 in RF_{H} is uniform over C (as in G_{11}). We then check $\mathcal{A}$'s forgery. Since $\mathsf{V}(vk, w^*, c^*, z^*) = \mathsf{true}$, where $c^* = \mathsf{RF}_{\mathsf{H}}(m^*, w^*)$, c^* should be a good challenge in $\mathcal{G}_{vk}(w^*)$. This means that $g(m^*, w^*) = 1$ and $\mathcal{A}_{\mathrm{gspb}}$ wins the game. Thus, we have

$$\Pr[W_{11} \mid vk] = \Pr[\mathrm{GSPB}_{\lambda_{vk}, \mathcal{A}_{\mathrm{gspb}}} = 1 \mid vk] \le 8(q + 1)^2 \lambda_{vk},$$

where $\lambda_{vk} := \max_{(m,w) \in \mathcal{M} \times \mathcal{W}} \lambda_{vk}(m, w)$. Averaging this inequality over vk generated by $\mathsf{LossyGen}_{\mathsf{LID}}(1^\kappa)$, we obtain

$$\Pr[W_{12}] \le 8(q + 1)^2 \cdot \operatorname*{Exp}_{vk \leftarrow \mathsf{LossyGen}_{\mathsf{LID}}(1^\kappa)} [\lambda_{vk}] \le 8(q + 1)^2 \cdot \epsilon_\ell$$

as we wanted, where we used $\lambda_{vk} = \max_{(m,w)} \lambda_{vk}(m, w) = \max_w (\#\mathcal{G}_{vk}(w)/\#C)$ and Equation 2. □

E Memory-Tight Proofs for PSF-(P)FDH

E.1 Preimage Sampleable Functions

Definition E.1 (Preimage sampleable function [GPV08]). *A family of preimage sampleable functions consists* PSF *of the following quadruple of PPT algorithms* $(\mathsf{Gen}_{\mathsf{PSF}}, \mathsf{F}, \mathsf{Inv}, \mathsf{Sample})$*:*

- $\mathsf{Gen}_{\mathsf{PSF}}(1^\kappa) \to (vk, sk)$*: a key-generation algorithm that on input* 1^κ *outputs a pair of keys* (vk, sk)*.*
- $\mathsf{F}(vk, x) \to y$*: a* deterministic *evaluation algorithm that takes as input* vk *and* $x \in \mathcal{X}$ *and outputs* $y \in \mathcal{Y}$*.*
- $\mathsf{Sample}(vk) \to x$*: a sampling algorithm that takes as input* vk *and outputs* $x \in \mathcal{X}$*.*
- $\mathsf{Inv}(sk, y) \to x$*: a preimage-sampling algorithm that takes as input* sk *and* $y \in \mathcal{Y}$ *and outputs* $x \in \mathcal{X}$*.*

We define properties of PSF.

[5] But, this never occurs.

$\mathsf{Gen}(1^\kappa)$	$\mathsf{Sign}(sk, m)$	$\mathsf{Vrfy}(vk, m, \sigma)$
$(vk, sk) \leftarrow \mathsf{Gen}_{\mathsf{PSF}}(1^\kappa)$	$h := \mathsf{H}(m)$	$h := \mathsf{H}(m)$
return (vk, sk)	$\sigma \leftarrow \mathsf{Inv}(sk, h)$	$h' := \mathsf{F}(vk, \sigma)$
	return σ	**return** $[\![h = h']\!]$

$\mathsf{Gen}(1^\kappa)$	$\mathsf{Sign}((sk, K), m)$	$\mathsf{Vrfy}(vk, m, \sigma)$
$(vk, sk) \leftarrow \mathsf{Gen}_{\mathsf{PSF}}(1^\kappa)$	$h := \mathsf{H}(m)$	$h := \mathsf{H}(m)$
$K \leftarrow \{0,1\}^\kappa$	$r := \mathsf{PRF}(K, m)$	$h' := \mathsf{F}(vk, \sigma)$
return $(vk, (sk, K))$	$\sigma := \mathsf{Inv}(sk, h; r)$	**return** $[\![h = h']\!]$
	return σ	

Fig. 19. FDH[PSF, H] (upper) and DFDH[PSF, H, PRF] (lower).

Definition E.2 (Simulatability [CCLM22]). *We say that* PSF *is* ϵ-simulatable *if the following two distributions are* ϵ*-close:*

$$D_1 : y \leftarrow \mathcal{Y}; x \leftarrow \mathsf{Inv}(sk, y); \textit{return } (x, y)$$
$$D_2 : x \leftarrow \mathsf{Sample}(vk); y := \mathsf{F}(vk, x); \textit{return } (x, y).$$

Definition E.3 (Preimage min-entropy). *We say that* PSF *has* α-preimage min-entropy *if for each* $y \in \mathcal{Y}$*, the conditional min-entropy of* $x \leftarrow \mathsf{Sample}(vk)$ *given* $\mathsf{F}(vk, x) = y$ *is at least* α*.*

Definition E.4 (Collision resistance). *We say that* PSF *is* collision-resistant *if for any QPT adversary* $\mathcal{A}$*, the following advantage is negligible in* κ*:*

$$\mathsf{Adv}^{\mathrm{cr}}_{\mathsf{PSF},\mathcal{A}}(\kappa) := \Pr[(vk, sk) \leftarrow \mathsf{Gen}_{\mathsf{PSF}}(1^\kappa); (x, x') \leftarrow \mathcal{A}(vk) : x \neq x' \wedge \mathsf{F}(vk, x) = \mathsf{F}(vk, x')].$$

E.2 Signature based on PSF

We review a signature scheme constructed from preimage-sampleable functions (PSF) [BR96, GPV08]. Let $\mathsf{PSF} = (\mathsf{Gen}_{\mathsf{PSF}}, \mathsf{F}, \mathsf{Inv}, \mathsf{Sample})$ be a family of preimage sampleable functions. The signature scheme obtained by applying the Full-Domain Hash FDH is depicted in Figure 19. If Inv is derandomized by PRF, then we call this conversion as DFDH and denote DFDH[PSF, H, PRF]. If we use RF instead of PRF, then we denote it as DFDH^+[PSF, H, RF]. If we apply RDS in subsection 3.1 to the obtained scheme, then we call the conversion as PFDH and denote PFDH[PSF, H, λ].

E.3 Multi-Challenge Security for PSF-(P)FDH

While we can use both approaches of Diemert et al. [DGJL21] and Ghoshal et al. [GGJT22], we here use the approach of Diemert et al. [DGJL21]: We show the msEUF-CMA1 security of FDH[PSF, H] by slightly modifying the sEUF-CMA proof of DFDH[PSF, H, PRF] in Boneh et al. [BDF+11] or the BU proof of DFDH[PSF, H, PRF] in Chatterjee et al. [CCLM22] and apply Lemma 3.1 to show the msEUF-CMA security of PFDH[PSF, H] = RDS[FDH[PSF, H], λ] via memory-tight reductions.

Theorem E.1 (msEUF-CMA1 security of FDH[PSF, H]). *Let* $\mathsf{H} : \mathcal{M} \to \mathcal{Y}$ *be a random oracle. Let* PSF *be a family of preimage-sampleable functions that is* ϵ*-simulatable and has* α*-preimage min-entropy. Let* $\mathsf{DS} := \mathsf{FDH}[\mathsf{PSF}, \mathsf{H}]$*. Then, for a quantum adversary* $\mathcal{A}$ *breaking the* msEUF-CMA1 *security of* DS *taht issues at most* q_H *quantum queried to* H*,* q_S *classical queries to the signing oracle, and* q_F *classical queries to the forgery oracle, there exists a quantum* $\mathcal{F}$*-oracle adversary* $\mathcal{A}_{\mathrm{cr}}$ *such that*

$$\mathsf{Adv}^{\mathrm{mseuf\text{-}cma1}}_{\mathsf{DS},\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\mathrm{cr}}_{\mathsf{PSF},\mathcal{A}_{\mathrm{cr}}}(\kappa) + \sqrt{(6(q_S + q_H + q_F))^3 \epsilon} + q_F \cdot 2^{-\alpha},$$
$$\mathsf{Time}^*(\mathcal{A}_{\mathrm{cr}}) = \mathsf{Time}(\mathcal{A}) + (q_H + q_S + q_F) \cdot O(\mathsf{Time}(\mathsf{PSF})),$$
$$\mathsf{Mem}^*(\mathcal{A}_{\mathrm{cr}}) = \mathsf{Mem}(\mathcal{A}) + O(\mathsf{Mem}(\mathsf{PSF})),$$

where $\mathcal{F} = \mathsf{Func}(\mathcal{M}, \mathcal{R}_{\mathsf{Sample}})$*.*

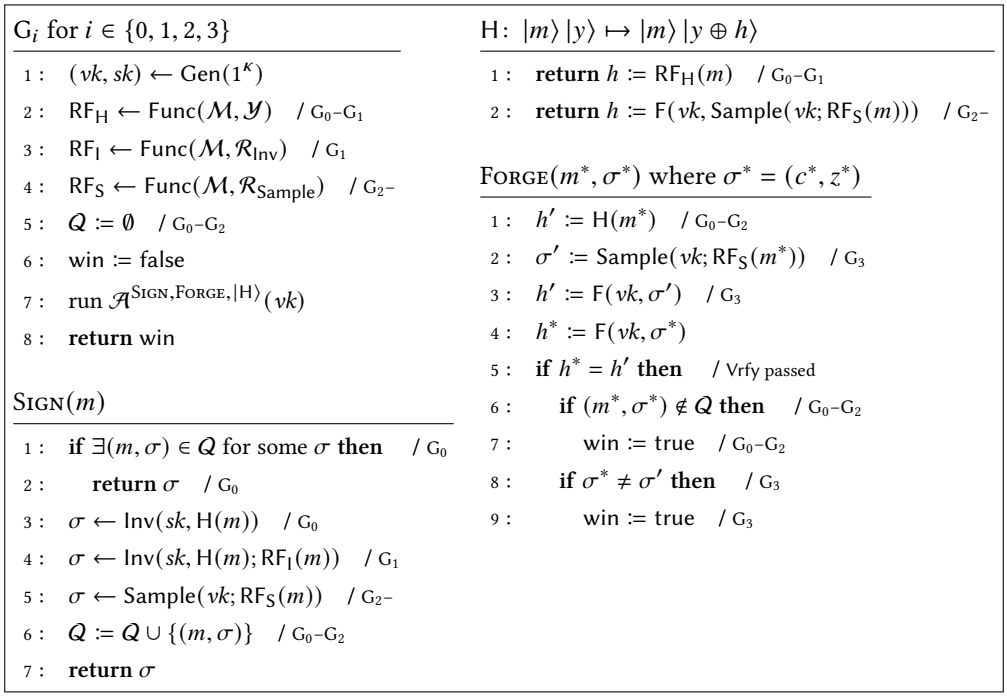

Fig. 20. G_i for $i \in \{0,1,2,3\}$

Applying Lemma 3.1, we obtain the following corollary.

Corollary E.1 (MSEUF-CMA security of PFDH[PSF, H, λ]). *Let* $\mathsf{H}\colon \mathcal{M} \times \{0,1\}^\lambda \to \mathcal{Y}$ *be a random oracle. Let* PSF *be a family of preimage-sampleable functions that is* ϵ*-simulatable and has* α*-preimage min-entropy. Let* $\mathsf{DS} := \mathsf{PFDH}[\mathsf{PSF}, \mathsf{H}, \lambda]$*. Then, for a quantum adversary* $\mathcal{A}$ *breaking the* MSEUF-CMA *security of* DS *that issues at most* q_H *quantum queried to* H*,* q_S *classical queries to the signing oracle, and* q_F *classical queries to the forgery oracle, there exists a quantum* $\mathcal{F}$*-oracle adversary* $\mathcal{A}_\mathrm{cr}$ *such that*

$$\begin{aligned}
&\mathsf{Adv}^{\text{mseuf-cma}}_{\mathsf{DS},\mathcal{A}}(\kappa) \le \mathsf{Adv}^{\mathrm{cr}}_{\mathsf{PSF},\mathcal{A}_\mathrm{cr}}(\kappa) + \sqrt{(6(q_S+q_H+q_F))^3\epsilon} + q_F \cdot 2^{-\alpha} + q_S^2 \cdot 2^{-\lambda},\\
&\mathsf{Time}^*(\mathcal{A}_\mathrm{cr}) = \mathsf{Time}(\mathcal{A}) + (q_H+q_S+q_F)\cdot O(\mathsf{Time}(\mathsf{PSF})),\\
&\mathsf{Mem}^*(\mathcal{A}_\mathrm{cr}) = \mathsf{Mem}(\mathcal{A}) + O(\mathsf{Mem}(\mathsf{PSF})),
\end{aligned}$$

where $\mathcal{F} = \mathsf{Func}(\mathcal{M}, \mathcal{R}_\mathsf{Sample})$.

Game G_0*:* This is the original game of the MSEUF-CMA1 security. See G_0 in Figure 20. By definition, we have

$$\Pr[W_0] = \mathsf{Adv}^{\text{mseuf-cma1}}_{\mathsf{DS},\mathcal{A}}(\kappa).$$

Game G_1*:* Next, we derandomize the signing oracle using a random function RF_I. By this modification, we do not need to maintain the list in the signing oracle. We have

$$G_0 = G_1.$$

Game G_2*:* We next modify the signing oracle and the random oracle. In this game, the signing oracle given m returns $\mathsf{Sample}(vk, \mathsf{RF}_\mathsf{S}(m))$ and the random oracle given m returns $\mathsf{F}(vk, \mathsf{Sample}(vk, \mathsf{RF}_\mathsf{S}(m)))$. Since $x \leftarrow \mathsf{Sample}(vk)$ conditioned on $\mathsf{F}(vk,x) = y$ is By applying Lemma 2.1 with ϵ-simulatability, we have

$$|\Pr[W_1] - \Pr[W_2]| \le \sqrt{(6(q_S+q_H+q_F))^3\epsilon}.$$

Game G_3*:* We finally modify how to update the flag win. In G_3, the flag is set true if the submitted forgery σ^* differs from the expected one σ'.

Lemma E.1. *Suppose that* PSF *has* α*-preimage min-entropy. We have*

$$|\Pr[W_2] - \Pr[W_3]| \leq q_F \cdot 2^{-\alpha}.$$

Proof. Suppose that an adversary $\mathcal{A}$ submits a valid pair (m^*, σ^*). let $\sigma' \coloneqq \mathsf{Sample}(vk; \mathsf{RF}_{\mathsf{S}}(m^*))$. Let us consider two cases:

1. If m^* is already queried to Sign, then (m^*, σ') should be contained in Q. Thus the condition $(m^*, \sigma^*) \notin Q$ is equivalent to $\sigma^* \neq \sigma'$ and the flags win are the same in the both games.
2. If m^* is not queried to Sign, then the two games differ if the adversary submits (m^*, σ') in G_2. Since the min-entropy of $\sigma' = \mathsf{Sample}(vk; \mathsf{RF}_{\mathsf{S}}(m^*))$ given $h^* = \mathsf{F}(vk, \mathsf{Sample}(vk; \mathsf{RF}_{\mathsf{S}}(m^*)))$ is at least α, this event happens with probability at most $q_F \cdot 2^{-\alpha}$.

Thus, we obtain the bound. □

Now, making a memory-tight reduction for collision-resistance of PSF is easy.

Lemma E.2. *There exists a quantum* $\mathcal{F}$*-oracle adversary* $\mathcal{A}_{\mathrm{cr}}$ *such that*

$$\begin{aligned}
&\Pr[W_3] \leq \mathsf{Adv}^{\mathrm{cr}}_{\mathsf{PSF},\mathcal{A}_{\mathrm{cr}}}(\kappa),\\
&\mathsf{Time}^*(\mathcal{A}_{\mathrm{cr}}) = \mathsf{Time}(\mathcal{A}) + (q_H + q_S + q_F) \cdot O(\mathsf{Time}(\mathsf{PSF})),\\
&\mathsf{Mem}^*(\mathcal{A}_{\mathrm{cr}}) = \mathsf{Mem}(\mathcal{A}) + O(\mathsf{Mem}(\mathsf{PSF})),
\end{aligned}$$

where $\mathcal{F} = \mathsf{Func}(\mathcal{M}, \mathcal{R}_{\mathsf{Sample}})$.

Since the reduction is straightforward, we omit it.

E.4 Plus-One Security for PSF-DFDH

The following theorem is obtained by modifying the proof of Boneh and Zhandry [BZ13b, Theorem 3.19].

Theorem E.2 (PO **security of** DFDH[PSF, H, PRF]**).** *Let* $\mathsf{H}\colon \mathcal{M} \to \mathcal{Y}$ *be a random oracle. Let* PSF *be a family of preimage-sampleable functions that is* ϵ*-simulatable and has* α*-preimage min-entropy. Let* DS $\coloneqq$ DFDH[PSF, H, PRF]*. Then, for a quantum adversary* $\mathcal{A}$ *breaking the* PO *security of* DS *that issues at most* q_H *quantum queried to* H*,* q_S *classical queries to the signing oracle, and* q_F *classical queries to the forgery oracle, there exist a quantum* $\mathcal{F}_{\mathrm{prf}}$*-oracle adversary* $\mathcal{A}_{\mathrm{prf}}$ *and a quantum* $\mathcal{F}_{\mathrm{cr}}$*-oracle adversary* $\mathcal{A}_{\mathrm{cr}}$ *and such that*

$$\begin{aligned}
\mathsf{Adv}^{\mathrm{po}}_{\mathsf{DS},\mathcal{A}}(\kappa) &\leq \mathsf{Adv}^{\mathrm{pr}}_{\mathsf{PSF},\mathcal{A}_{\mathrm{prf}}}(\kappa) + \mathsf{Adv}^{\mathrm{cr}}_{\mathsf{PSF},\mathcal{A}_{\mathrm{cr}}}(\kappa)\\
&\quad + \sqrt{(6(q_S + q_H + q_F))^3 \epsilon} + (q_S + 1)/\lfloor 2^{\alpha} \rfloor,\\
\mathsf{Time}^*(\mathcal{A}_{\mathrm{prf}}) &= \mathsf{Time}(\mathcal{A}) + (q_H + q_S + q_F) \cdot O(\mathsf{Time}(\mathsf{PSF})),\\
\mathsf{Mem}^*(\mathcal{A}_{\mathrm{prf}}) &= \mathsf{Mem}(\mathcal{A}) + O(q_F \mathsf{Mem}(\mathsf{PSF})),\\
\mathsf{Time}^*(\mathcal{A}_{\mathrm{cr}}) &= \mathsf{Time}(\mathcal{A}) + (q_H + q_S + q_F) \cdot O(\mathsf{Time}(\mathsf{PSF})),\\
\mathsf{Mem}^*(\mathcal{A}_{\mathrm{cr}}) &= \mathsf{Mem}(\mathcal{A}) + O(\mathsf{Mem}(\mathsf{PSF})),
\end{aligned}$$

where $\mathcal{F}_{\mathrm{prf}} = \mathsf{Func}(\mathcal{M}, \mathcal{Y})$ *and* $\mathcal{F}_{\mathrm{cr}} = \mathsf{Func}(\mathcal{M}, \mathcal{R}_{\mathsf{Sample}})$.

The proof becomes memory-tight if we derandomize with the random function RF.

Corollary E.2 (PO **security of** DFDH$^+$[PSF, H, RF]**).** *Let* $\mathsf{H}\colon \mathcal{M} \to \mathcal{Y}$ *be a random oracle. Let* PSF *be a family of preimage-sampleable functions that is* ϵ*-simulatable and has* α*-preimage min-entropy. Let* DS $\coloneqq$ DFDH$^+$[PSF, H, RF]*. Then, for a quantum adversary* $\mathcal{A}$ *breaking the* PO *security of* DS *that issues at most* q_H *quantum queried to* H*,* q_S *classical queries to the signing oracle, and* q_F *classical queries to the forgery oracle, there exists a quantum* $\mathcal{F}_{\mathrm{cr}}$*-oracle adversary* $\mathcal{A}_{\mathrm{cr}}$ *and such that*

$$\begin{aligned}
&\mathsf{Adv}^{\mathrm{po}}_{\mathsf{DS},\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\mathrm{cr}}_{\mathsf{PSF},\mathcal{A}_{\mathrm{cr}}}(\kappa) + \sqrt{(6(q_S + q_H + q_F))^3 \epsilon} + (q_S + 1)/\lfloor 2^{\alpha} \rfloor,\\
&\mathsf{Time}^*(\mathcal{A}_{\mathrm{cr}}) = \mathsf{Time}(\mathcal{A}) + (q_H + q_S + q_F) \cdot O(\mathsf{Time}(\mathsf{PSF})),\\
&\mathsf{Mem}^*(\mathcal{A}_{\mathrm{cr}}) = \mathsf{Mem}(\mathcal{A}) + O(\mathsf{Mem}(\mathsf{PSF})),
\end{aligned}$$

where $\mathcal{F}_{\mathrm{cr}} = \mathsf{Func}(\mathcal{M}, \mathcal{R}_{\mathsf{Sample}})$.

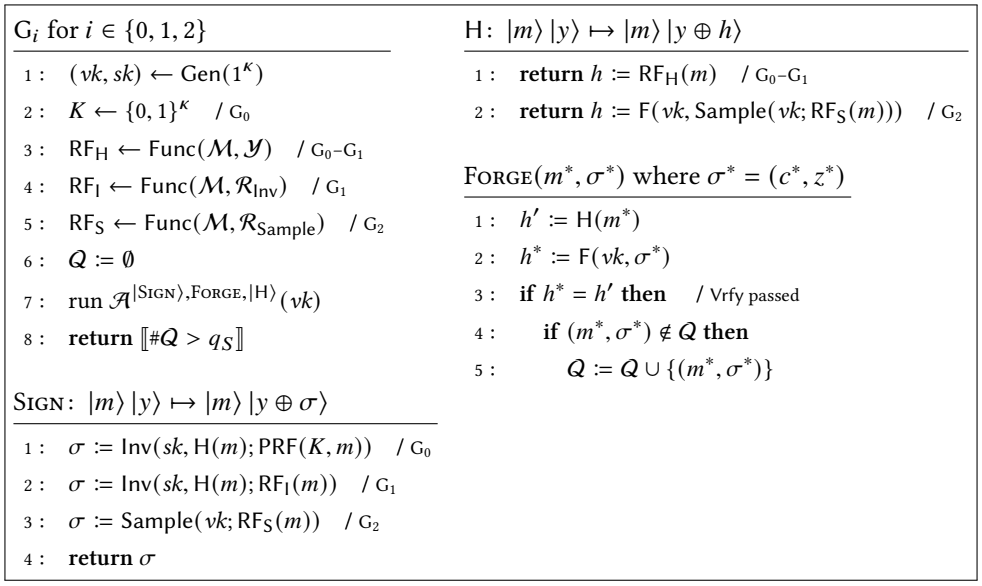

Fig. 21. G_i for $i \in \{0,1,2\}$

Game G_0*:* This is the original game of the PO security. See G_0 in Figure 21. By definition, we have

$$\Pr[W_0] = \mathsf{Adv}^{\mathrm{po}}_{\mathsf{DS},\mathcal{A}}(\kappa).$$

Game G_1*:* We next replace PRF with $\mathsf{RF}_\mathcal{I}$ in G_1. The straightforward reduction shows the following lemma, which is *memory-loose* since we need to maintain Q.

Lemma E.3. *There exists a quantum $\mathcal{F}$-oracle adversary $\mathcal{A}_{\mathrm{prf}}$ such that*

$$\begin{aligned}
&|\Pr[W_0] - \Pr[W_1]| \le \mathsf{Adv}^{\mathrm{pr}}_{\mathsf{PRF},\mathcal{A}_{\mathrm{prf}}}(\kappa),\\
&\mathsf{Time}^*(\mathcal{A}_{\mathrm{prf}}) = O(\mathsf{Time}(\mathcal{A})) + (q_H + q_S + q_F)\cdot O(\mathsf{Time}(\mathsf{PSF})),\\
&\mathsf{Mem}^*(\mathcal{A}_{\mathrm{prf}}) = O(\mathsf{Mem}(\mathcal{A})) + q_F \cdot O(\mathsf{Mem}(\mathsf{PSF})),
\end{aligned}$$

where $\mathcal{F} = \mathsf{Func}(\mathcal{M}, \mathcal{Y})$.

Game G_2*:* We next modify the signing oracle and the random oracle. In this game, the signing oracle given m returns $\sigma = \mathsf{Sample}(vk, \mathsf{RF}_\mathsf{S}(m))$ and the random oracle given m returns $\mathsf{F}(vk, \mathsf{Sample}(vk, \mathsf{RF}_\mathsf{S}(m)))$. By applying Lemma 2.1 with ϵ-simulatability, we have

$$|\Pr[W_1] - \Pr[W_2]| \le \sqrt{(6(q_S + q_H + q_F))^3 \epsilon}.$$

Game G_3*:* We next modify the winning condition as follows: We introduce a flag win which is set true if (m^*, σ^*) is valid and $\sigma^* \neq \mathsf{Sample}(vk; \mathsf{RF}_\mathsf{S}(m^*))$ in the oracle $\textsc{Forge}$. The challenger outputs $[\![\#Q > \mathsf{cnt}_s]\!] \wedge \mathsf{win}$ instead of $[\![\#Q > \mathsf{cnt}_s]\!]$. We have the following lemma as Lemma 5.4 by following the argument in the proof of [BZ13b, Theorem 3.19].

Lemma E.4. *Suppose that* PSF *has α-preimage min-entropy. We have*

$$|\Pr[W_2] - \Pr[W_3]| \le (q_S + 1)/\lfloor 2^\alpha \rfloor.$$

Proof. The two games differ if the adversary submits at least $(q_S + 1)$ distinct pairs of message/signature $\{(m_i^*, \mathsf{Sample}(vk; \mathsf{RF}_\mathsf{S}(m_i^*)))\}_i$. Since PSF has α-preimage min-entropy, even if the adversary knows $h^* = \mathsf{H}(m^*) = \mathsf{F}(vk, \mathsf{Sample}(vk; \mathsf{RF}_\mathsf{S}(m^*)))$, the min-entropy of $\mathsf{Sample}(vk; \mathsf{RF}_\mathsf{S}(m^*))$ is at least α. Thus, applying Lemma 2.2, this event happens with probability at most $(q_S + 1)/\lfloor 2^\alpha \rfloor$. □

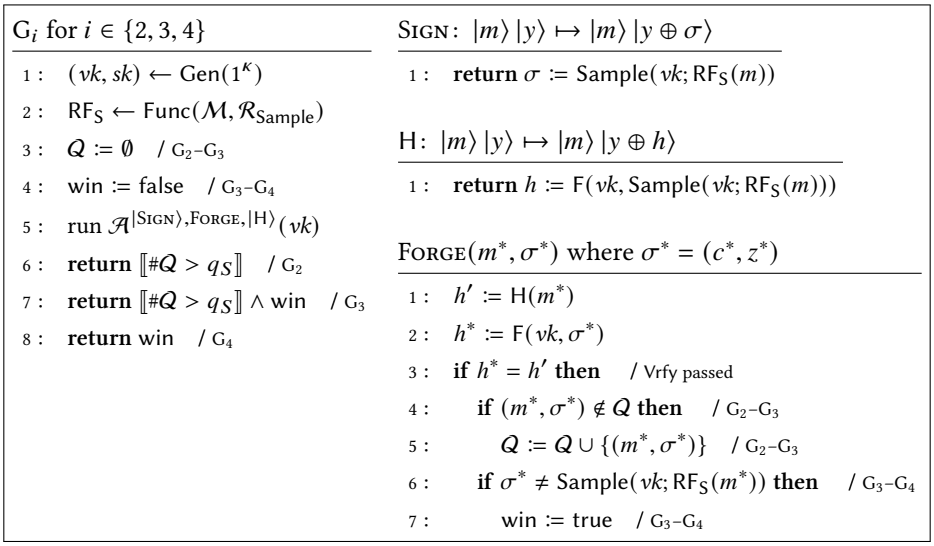

Fig. 22. G_i for $i \in \{2,3,4\}$

Game G_4*:* The challenger outputs the flag win in this game. We can remove the list Q. Since we relax the condition and this relaxation cannot be detected by the adversary, we have

$$\Pr[W_3] \leq \Pr[W_4].$$

Constructing an adversary finding a collision for $\mathsf{F}(vk, \cdot)$ is easy.

Lemma E.5. *There exists a quantum $\mathcal{F}$-oracle adversary $\mathcal{A}_{\mathrm{cr}}$ such that*

$$\begin{aligned}
&\Pr[W_4] \leq \mathsf{Adv}^{\mathrm{cr}}_{\mathsf{PSF},\mathcal{A}_{\mathrm{cr}}}(\kappa),\\
&\mathsf{Time}^*(\mathcal{A}_{\mathrm{cr}}) = \mathsf{Time}(\mathcal{A}) + (q_H + q_S + q_F)\cdot O(\mathsf{Time}(\mathsf{PSF})),\\
&\mathsf{Mem}^*(\mathcal{A}_{\mathrm{cr}}) = \mathsf{Mem}(\mathcal{A}) + O(\mathsf{Mem}(\mathsf{PSF})),
\end{aligned}$$

where $\mathcal{F} = \mathsf{Func}(\mathcal{M}, \mathcal{R}_{\mathsf{Sample}})$.

Since the reduction is straightforward, we omit it.

E.5 Strong Blinded Unforgeability for PSF-DFDH

While Chatterjee et al. [CCLM22] showed the BU security of PSF-DFDH via memory-loose reductions, we here show stronger security (sBU security) with memory-tight reduction. The proof is obtained by slightly modifying their proof for the BU security.

Theorem E.3 (sBU **security of** DFDH[PSF, H, PRF]**).** *Let* $\mathsf{H}\colon \mathcal{M} \to \mathcal{Y}$ *be a random oracle. Let* PSF *be a family of preimage-sampleable functions that is ϵ-simulatable and has α-preimage min-entropy. Let* $\mathsf{DS} := \mathsf{DFDH}[\mathsf{PSF}, \mathsf{H}, \mathsf{PRF}]$. *Then, for a quantum adversary $\mathcal{A}$ breaking the* sBU *security of* DS *that issues at most q_H quantum queried to* H*, q_S classical queries to the signing oracle, and q_F classical queries to the forgery oracle, there exist a quantum $\mathcal{F}_{\mathrm{prf}}$-oracle adversary $\mathcal{A}_{\mathrm{prf}}$ and a quantum $\mathcal{F}_{\mathrm{cr}}$-oracle adversary $\mathcal{A}_{\mathrm{cr}}$ such that*

$$\begin{aligned}
\mathsf{Adv}^{\mathrm{sbu}}_{\mathsf{DS},\mathcal{A}}(\kappa) &\leq \mathsf{Adv}^{\mathrm{pr}}_{\mathsf{PRF},\mathcal{A}_{\mathrm{prf}}}(\kappa) + \mathsf{Adv}^{\mathrm{cr}}_{\mathsf{PSF},\mathcal{A}_{\mathrm{cr}}}(\kappa)\\
&\quad + \sqrt{(6(q_S + q_H + q_F))^3\epsilon} + q_F\cdot 2^{-\alpha},\\
\mathsf{Time}^*(\mathcal{A}_{\mathrm{prf}}) &= \mathsf{Time}(\mathcal{A}) + (q_H + q_S + q_F)\cdot O(\mathsf{Time}(\mathsf{PSF}) + \mathsf{Time}(B_\epsilon)),\\
\mathsf{Mem}^*(\mathcal{A}_{\mathrm{prf}}) &= \mathsf{Mem}(\mathcal{A}) + O(\mathsf{Mem}(\mathsf{PSF})) + \mathsf{Mem}(B_\epsilon)),\\
\mathsf{Time}^*(\mathcal{A}_{\mathrm{cr}}) &= \mathsf{Time}(\mathcal{A}) + (q_H + q_S + q_F)\cdot O(\mathsf{Time}(\mathsf{PSF}) + \mathsf{Time}(B_\epsilon)),\\
\mathsf{Mem}^*(\mathcal{A}_{\mathrm{cr}}) &= \mathsf{Mem}(\mathcal{A}) + O(\mathsf{Mem}(\mathsf{PSF}) + \mathsf{Mem}(B_\epsilon)),
\end{aligned}$$

where $\mathcal{F}_{\mathrm{prf}} = \mathsf{Func}(\mathcal{M}\times\mathcal{X}, \mathcal{P}) \times \mathsf{Func}(\mathcal{M}, \mathcal{Y})$ *where* $\mathcal{F}_{\mathrm{cr}} = \mathsf{Func}(\mathcal{M}\times\mathcal{X}, \mathcal{P}) \times \mathsf{Func}(\mathcal{M}, \mathcal{R}_{\mathsf{Sample}})$.

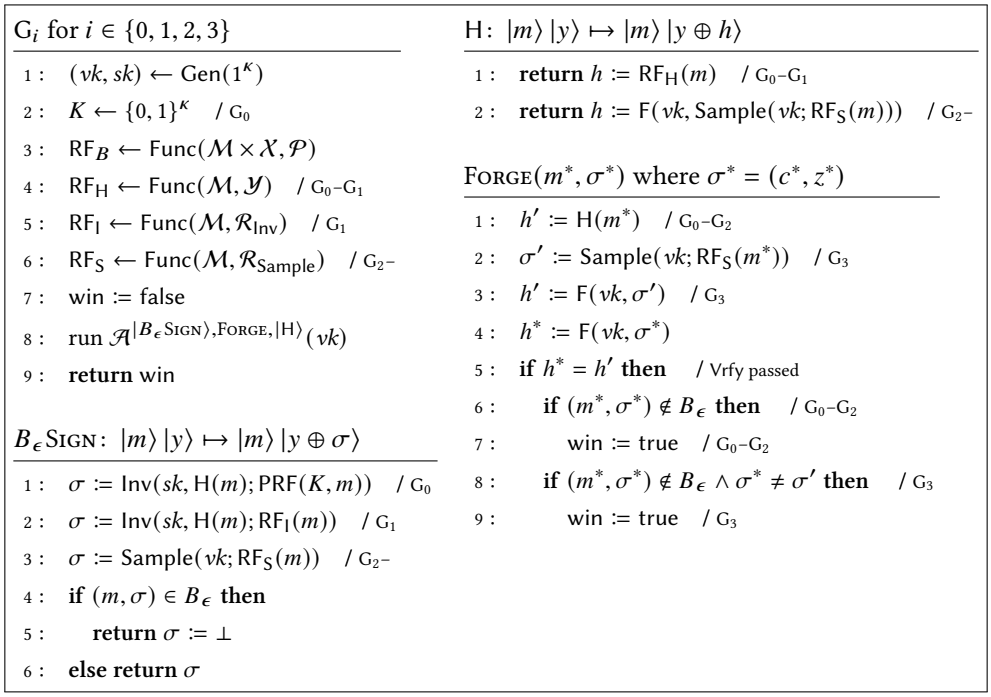

Fig. 23. G_i for $i \in \{0,1,2,3\}$

Game G_0*:* This is the original game of the sBU security. See G_0 in Figure 23. By definition, we have

$$\Pr[W_0] = \mathsf{Adv}^{\mathrm{sbu}}_{\mathsf{DS},\mathcal{A}}(\kappa).$$

Game G_1*:* We next replace PRF with RF_I in G_1. The straightforward reduction shows the following lemma.

Lemma E.6. *There exists a quantum $\mathcal{F}$-oracle adversary $\mathcal{A}_{\mathrm{prf}}$ such that*

$$|\Pr[W_0] - \Pr[W_1]| \le \mathsf{Adv}^{\mathrm{pr}}_{\mathsf{PRF},\mathcal{A}_{\mathrm{prf}}}(\kappa),$$
$$\mathsf{Time}^*(\mathcal{A}_{\mathrm{prf}}) = \mathsf{Time}(\mathcal{A}) + (q_H + q_S + q_F) \cdot O(\mathsf{Time}(\mathsf{PSF}) + \mathsf{Time}(B_\epsilon)),$$
$$\mathsf{Mem}^*(\mathcal{A}_{\mathrm{prf}}) = \mathsf{Mem}(\mathcal{A}) + O(\mathsf{Mem}(\mathsf{PSF}) + \mathsf{Mem}(B_\epsilon)),$$

where $\mathcal{F} = \mathsf{Func}(\mathcal{M} \times \mathcal{X}, \mathcal{P}) \times \mathsf{Func}(\mathcal{M}, \mathcal{Y})$.

Game G_2*:* We next modify the signing oracle and the random oracle. In this game, the signing oracle given m returns $\sigma = \mathsf{Sample}(vk, \mathsf{RF}_\mathsf{S}(m))$ (on unfiltered m) and the random oracle given m returns $\mathsf{F}(vk, \mathsf{Sample}(vk, \mathsf{RF}_\mathsf{S}(m)))$. By applying Lemma 2.1 with ϵ-simulatability, we have

$$|\Pr[W_1] - \Pr[W_2]| \le \delta_{1,2} := \sqrt{(6(q_S + q_H + q_F))^3 \epsilon}.$$

Game G_3*:* We finally modify how to update the flag win. In G_3, the flag is set true if the submitted forgery σ^* is different from the expected one σ' and $(m^*, \sigma^*) \notin B_\epsilon$.

Lemma E.7. *Suppose that* PSF *has α-preimage min-entropy. We have*

$$|\Pr[W_2] - \Pr[W_3]| \le q_F \cdot 2^{-\alpha}.$$

Proof. Let (m^*, σ^*) be a query to Forge the adversary made. Let $\sigma' := \mathsf{Sample}(vk; \mathsf{RF}_\mathsf{S}(m^*))$. The two games differ if the adversary submits a valid pair $(m^*, \sigma^*) \in B_\epsilon$ but $\sigma^* = \sigma'$. Let us consider two cases:

1. If $(m^*, \sigma') \notin B_\epsilon$, then this contradicts with $(m^*, \sigma') = (m^*, \sigma^*) \in B_\epsilon$. Thus, we do not need to consider this case.
2. If $(m^*, \sigma') \in B_\epsilon$, then the adversary cannot know σ' from B_ϵSign. Due to α-preimage min-entropy of PSF, the probability that $\sigma^* = \sigma'$ is at most $q_F \cdot 2^{-\alpha}$.

Thus, we obtain the bound. □

Now, making a memory-tight reduction for collision-resistance of PSF is easy.

Lemma E.8. *There exists a quantum $\mathcal{F}$-oracle adversary $\mathcal{A}_{\mathrm{cr}}$ such that*

$$\begin{aligned}
&\Pr[W_3] \le \mathsf{Adv}^{\mathrm{cr}}_{\mathsf{PSF},\mathcal{A}_{\mathrm{cr}}}(\kappa),\\
&\mathsf{Time}^*(\mathcal{A}_{\mathrm{cr}}) = \mathsf{Time}(\mathcal{A}) + (q_H + q_S + q_F)\cdot O(\mathsf{Time}(\mathsf{PSF}) + \mathsf{Time}(B_\epsilon)),\\
&\mathsf{Mem}^*(\mathcal{A}_{\mathrm{cr}}) = \mathsf{Mem}(\mathcal{A}) + O(\mathsf{Mem}(\mathsf{PSF}) + \mathsf{Mem}(B_\epsilon)),
\end{aligned}$$

where $\mathcal{F} = \mathsf{Func}(\mathcal{M}\times\mathcal{X},\mathcal{P}) \times \mathsf{Func}(\mathcal{M},\mathcal{R}_{\mathsf{Sample}})$.

Since the reduction is straightforward, we omit it.