A preliminary version of this paper appears in the proceedings of *the 25th International Conference on Practice and Theory of Public-Key Cryptography* (PKC 2022). This is the full version.

A Note on the Post-Quantum Security of (Ring) Signatures

Rohit Chatterjee[1], Kai-Min Chung[2], Xiao Liang[1⋆], and Giulio Malavolta[3]

[1] Stony Brook University, Stony Brook, USA
rochatterjee@cs.stonybrook.edu
xiao.crypto@gmail.com

[2] Academia Sinica, Taipei, Taiwan
kmchung@iis.sinica.edu.tw

[3] Max Planck Institute for Security and Privacy, Bochum, Germany
giulio.malavolta@hotmail.it

Abstract. This work revisits the security of classical signatures and ring signatures in a quantum world. For (ordinary) signatures, we focus on the arguably preferable security notion of *blind-unforgeability* recently proposed by Alagic et al. (Eurocrypt'20). We present two *short* signature schemes achieving this notion: one is in the quantum random oracle model, assuming quantum hardness of SIS; and the other is in the plain model, assuming quantum hardness of LWE with super-polynomial modulus. Prior to this work, the only known blind-unforgeable schemes are Lamport's one-time signature and the Winternitz one-time signature, and both of them are in the quantum random oracle model.

For ring signatures, the recent work by Chatterjee et al. (Crypto'21) proposes a definition trying to capture adversaries with quantum access to the signer. However, it is unclear if their definition, when restricted to the classical world, is as strong as the standard security notion for ring signatures. They also present a construction that only *partially* achieves (even) this seeming weak definition, in the sense that the adversary can only conduct superposition attacks over the messages, but not the rings. We propose a new definition that does not suffer from the above issue. Our definition is an analog to the blind-unforgeability in the ring signature setting. Moreover, assuming the quantum hardness of LWE, we construct a compiler converting any blind-unforgeable (ordinary) signatures to a ring signature satisfying our definition.

Keywords: Blind-Unforgeability · Post-Quantum · Ring Signatures

⋆ Part of this work was done while visiting Max Planck Institute.

Table of Contents

1 Introduction

Recent advances in quantum computing have uncovered several new threats to the existing body of cryptographic work. As demonstrated several times in the literature (e.g., [Wat06, BDF+11, Zha12a, ABG+]), building quantum-secure primitives requires more than taking existing constructions and replacing the underlying assumptions with post-quantum ones. It usually requires new techniques and analysis. Moreover, for specific primitives, even giving a meaningful security notion against quantum adversaries is a non-trivial task (e.g., [BZ13a, BZ13b, Zha15, Unr16, AMRS20]). This work focuses on *post-quantum security* of digital signature schemes, namely, classical signatures schemes for which we want to protect against quantum adversaries.

Post-Quantum Unforgeable Signatures. To build post-quantum secure signature schemes, the first step is to have a notion of unforgeability that protects against adversaries with quantum power. Probably the most natural attempt is to take the standard existential unforgeability (EUF) game, but require unforgeability against all *quantum polynomial-time* (QPT) adversaries (instead of all *probabilistic polynomial-time* (PPT) adversaries). We emphasize that the communication between the EUF challenger and the QPT adversary is still classical. Namely, the adversary is not allowed to query the challenger's circuit in a quantum manner. Herein, we refer to this notion as PQ-EUF. Usually, PQ-EUF can be achieved by existing constructions in the classical setting via replacing the underlying hardness assumptions with quantum-hard ones (e.g., hard problems on lattice or isogeny-based assumptions).

The (Quantum) Random Oracle Model. In the classical setting, the random oracle model (ROM) [BR93] has been accepted as a useful paradigm to obtain efficient signature schemes. When considering the above PQ-EUF notion in the ROM, two choices arise—one can either allow the adversary *classical* access to the RO (as in the classical setting)[4], or *quantum* access to the RO. The latter was first formalized as the *quantum random oracle model* (QROM) by Boneh et al. [BDF+11], who showed that new techniques are necessary to achieve unforgeability against QPT adversaries in this model. Then, a large body of literature has since investigated the PQ-EUF in QROM [ARU14, Unr17, KLS18, DFMS19, LZ19, DFM20, GHHM20].

One-More Unforgeability vs Bind Unforgeability. Starting from [Zha12a], people realize that the definitional approach taken by the above PQ-EUF may not be sufficient to protect against quantum adversaries. The reason is that quantum adversaries may try to attack the concerned protocol/primitive by executing it *quantumly*, even if the protocol/primitive by design is only meant to be executed classically. As argued in existing literature (e.g., [DFNS13, GHS16]), such an attack could possibly occur in a situation where the computer executing the classical protocol is a quantum machine, and an adversary somehow manages to observe the communication before measurement. Other examples include adversaries managing to trick a classical device (e.g., a smart card reader) into showing full or partial quantum behavior by, for example, cooling it down and shielding it from any external electromagnetic or thermal interference. Moreover, this concern may also arise in the security reduction (even) w.r.t. classical security games but against QPT adversaries. For example, some constructions may allow the adversary to obtain an *indistinguishability obfuscation* of, say, a PRF; the QPT adversary can then implement it as a quantum circuit to conduct superposition attacks. Recently, this issue has received an increasing amount of attention

[4] To avoid confusion, we henceforth denote this model as CROM ("C" for "classical").

[BZ13a, BZ13b, Zha15, Unr16, GHS16, SY17, HY18, HI19, CHS19, AMRS20, CETU20, CEV20, ABDS, HI21, HS21, BM21].

To address the aforementioned security threats to digital signatures, it is reasonable to give the QPT adversary $\mathcal{A}$ *quantum access* to the signing oracle in the EUF game. This raises an immediate question—How should the game decide if $\mathcal{A}$'s final forgery is valid? Recall that in the classical setting (or the PQ-EUF above), the game records all the signing queries made by $\mathcal{A}$; to decide if $\mathcal{A}$ wins, it needs to make sure that $\mathcal{A}$'s final forgery message-signature pair is different from the ones $\mathcal{A}$ learned from the signing oracle. However, this approach does not fit into the quantum setting, since it is unclear how to record $\mathcal{A}$'s *quantum* queries without irreversibly disturbing them.

Boneh and Zhandry [BZ13b] proposed the notion of *one-more unforgeability*. This requires that the adversary cannot produce $\mathsf{sq}+1$ valid message-signature pairs with only sq signing queries (an approach previously taken to define blind signatures [PS96a]). When restricted to the classical setting, this definition is equivalent to the standard unforgeability of ordinary signatures, by a simple application of the pigeonhole principle. [BZ13b] shows how to convert any PQ-EUF signatures to one-more unforgeable ones using a *chameleon hash function* [KR00]; it also proves that the PQ-EUF signature scheme by Gentry, Peikert, and Vaikuntanathan [GPV08] (henceforth, GPV) is one-more unforgeable in the QROM, assuming the PRF in that construction is quantum secure (i.e., being a QPRF [Zha12a]).

However, as argued in [GYZ17, AMRS20], one-more unforgeability does not seem to capture all that we can expect from quantum unforgeability. For example, an adversary may produce a forgery for a message in a subset A of the message space, while making queries to the signing oracle supported on a disjoint subset B. Also, an adversary may make multiple quantum signing queries, but then must consume, say, all of the answers in order to make a single valid forgery. This forgery might be for a message that is different from all the messages in all the superpositions of previous queries. This clearly violates what we intuitively expect for unforgeability, but the one-more unforgeability definition may never rule this out.

To address these problems, Alagic at el. [AMRS20] propose *blind-unforgeability* (BU). Roughly, the blind-unforgeability game modifies the (quantum-accessible) signing oracle by asking it to always return "$\perp$" for messages in a "blinded" subset of the message space. The adversary's forgery is considered valid only if it lies in the blinded subset. In this way, the adversary is forced to forge a signature for a message she has not seen a signature before, consistent with our intuition for unforgeability. [AMRS20] shows that blind-unforgeability, when restricted to the classical setting, is also equivalent to PQ-EUF; Moreover, it does not suffer from the above problems for one-more unforgeability[5].

In terms of constructions, [AMRS20] show that Lamport's one-time signature [Lam79] is BU in the QROM, assuming the OWF is modeled as a (quantum-accessible) random oracle. Later, [MMO21] show that the Winternitz one-time signature [Mer90] is BU in the QROM, assuming the underlying hash function is modeled as a (quantum-accessible) random oracle. To the best of our knowledge, these are the only schemes known to achieve BU. This gives rise to the following question:

Question 1: *Is it possible to build (multi-time) signature schemes achieving blind-unforgeability, either in the QROM or the plain model?*

[5] Although [AMRS20] claimed that blind-unforgeability implies one-more unforgeability, their proof was flawed [Com21]. The relation between these two notions is still an open problem.

Post-Quantum Secure Ring Signatures. In a *ring signature* scheme [RST01, BKM06], a user can sign a message with respect to a *ring* of public keys, with the knowledge of a signing key corresponding to any public key in the ring. It should satisfy two properties:

1. *Anonymity* requires that no user can tell which user in the ring actually produced a given signature;
2. *Unforgeability* requires that no user outside the specified ring can produce valid signatures on behalf of this ring.

In contrast to its notional predecessor, *group signatures* [Cv91], no central coordination is required for producing and verfying ring signatures. Due to these features, ring signatures (and their variants) have found natural applications related to whistleblowing, authenticating leaked information, and more recently to cryptocurrencies [TSS+18, Noe15], and thus have received extensive attention (see, e.g., [CGH+21] and related work therein).

For ring signatures from *latticed-based* assumptions, there exist several constructions in the CROM [ABB+13, LLNW16, TSS+18, BLO, WZZ18, EZS+19, BKP20, LNS21], but only two schemes are known in the plain model [BK10, CGH+21]. The authors of [CGH+21] also initiate the study of quantum security for ring signatures. They propose a definition where the QPT adversary is allowed quantum access to the signing oracle in both the anonymity and unforgeability game, where the latter is a straightforward adaption of the aforementioned one-more unforgebility for ordinary signatures. As noted in their work, this approach suffers from two disadvantages:

1. Their unforgeability definition seems weak in the sense that, when restricted to the classical setting, it is unclear if their unforgeability is equivalent to the standard one (see Sec. 2.3). This is in contrast to ordinary signatures, for which one-more unforgeability is equivalent to the standard existential unforgeability;
2. Their construction only partially achieves (even) this seemingly weak definition. In more detail, their security proof only allows the adversary to conduct superposition attacks on the messages, but not on the rings. As remarked by the authors, this is not a definitional issue, but rather a limitation of their technique. Indeed, [CGH+21] left it as an open question to have a construction protecting against superposition attacks on both the messages and the rings.

   The outlined gap begs the following natural question:

   Question 2: *Can we have a proper unforgeability notion for ring signatures that does not suffer from the above disadvantage? If so, can we have a construction achieving such a notion?*

Our Results. In this work, we resolve the aforementioned questions:

1. We show that the GPV signature, which relies on the quantum hardness of SIS (QSIS), can be proven BU-secure in the QROM. Since our adversary has quantum access to the signing oracle, we also need to replace the PRF in the original GPV scheme with a QPRF, which is also known from QSIS. As will be discussed later in Sec. 2.1, our security proof is almost identical to the proof in [BZ13b] for the one-more unforgeability of GPV, except how the desired contradiction is derived in the last hybrid. Interestingly, our proof for BU turns out to be simpler than that in [BZ13b] (for one-more unforgeability). We remark that the GPV scheme is *short* (i.e., the signature size only depends on the security parameter, but not the message size).

2. We also construct a BU-secure signature *in the plain model*, assuming quantum hardness of Learning with Errors (QLWE) with super-polynomial modulus. Our construction is inspired by the signature (and adaptive IBE) scheme by Boyen and Li [BL16]. This signature scheme is also short.
3. We present a new definition of post-quantum security for ring signatures, by extending blind-unforgeability from [AMRS20]. We show that this definition, when restricted to the classical setting, is equivalent to the standard security requirements for ring signatures.
4. We build a ring signature satisfying the above definition. Our construction is a compiler that converts any BU (ordinary) signature to a ring signature achieving the definition in Item 3, assuming QLWE.

2 Technical Overview

2.1 BU Signatures in the QROM

We show that the GPV signature scheme from [GPV08] is BU-secure in the QROM. The GPV signature scheme follows the hash-and-sign paradigm and relies crucially on the notion of *preimage sampleable functions* (PSFs). As the name indicates, these functions can be efficiently inverted given a secret inverting key in addition to being efficiently computable. Further, the joint distribution of image-preimage pairs is statistically close, no matter whether the image or the preimage is sampled first. PSFs also provide collision resistance, as well as *pre-image min-entropy*: given any image, the set of possible preimages has $\omega(\log \lambda)$ bits of min-entropy, meaning that a specific preimage can only be predicted with negligible chance.

The GPV scheme uses a hash function H modeled as a random oracle. It first hashes the message m using H to obtain a digest h. The signing key includes the PSF secret key, and the signature is a preimage of h (the signing randomness is generated using a quantum secure PRF over the message). To verify a signature, one simply computes its image under the PSF and compares it with the digest.

Notice that in the proof of (post-quantum) blind-unforgeability, the adversary has quantum access to both H and the signing algorithm. To show blind-unforgeability, we will move to a hybrid experiment where the H and the signing algorithm Sign are constructed differently, but their *joint distribution* is statistically close to that in the real execution. To do so, the hybrid will set the signature for a message m to a random preimage from the domain of the PSF (note that this procedure is "de-randomized" using the aforementioned PRF). To answer a H-oracle query on m, the hybrid will first compute its signature (i.e., the PSF preimage corresponding to m), and then return the PSF evaluation on this signature (aka preimage) as the output of $H(m)$. Observe that, in this hybrid, the (H, Sign) oracles are constructed by first sampling preimages for the PSF, and then evaluating the PSF in the "forward" direction; in contrast, in the real game, the (H, Sign) oracles can be interpreted as sampling a image for PSF first, and then evaluating the PSF in the "reverse" direction using the inverting key. From the property of PSFs given above, these two approaches induce statistically-close joint distributions of (H, Sign) on each (classical) query. A lemma from [BZ13b] then shows that these are also indistinguishable to adversaries making polynomially-many *quantum* queries.

So far, our proof is identical to that of [BZ13b], where GPV is shown to be one-more unforgeable. This final part is where we differ. In the final hybrid, if the adversary produces a successful forgery

for a message in the blind set, only two possibilities arise. Since the image of the signature under the PSF must equal the digest, the signature must either (i) provide a second preimage for h to the one computed by the challenger, creating a collision for the PSF, or (ii) equal the one the challenger itself computes, compromising preimage min-entropy of the PSF. This latter claim requires special attention in [BZ13b]. A reduction to the min-entropy condition is not immediate, since it is unclear if the earlier quantum queries of $\mathcal{A}$ already allow $\mathcal{A}$ information about the preimages for the $q+1$ forgeries it outputs. To handle this, [BZ13b] prove a lemma ([BZ13b, Lemma 2.6]) showing q quantum queries will not allow $\mathcal{A}$ to predict $q+1$ preimages, given the min-entropy condition. In contrast, this last argument is superfluous in our case, since the blind unforgeability game *automatically* prevents any information for queries in the blindset from reaching the adversary. We can therefore directly appeal to the min-entropy condition for case (ii) above.

We present the formal construction and the corresponding proof in Sec. 4.

2.2 BU Signatures in the Plain Model

To construct a BU signature *in the plain model*, we make use of the signature template introduced in [BL16], which in turn relies on key-homomorphic techniques as used in [BV14]. We will refer to the [BV14] homomorphic evaluation procedure as $\mathsf{Eval}_{\mathrm{BV}}$. The [BL16] scheme uses the 'left-right trapdoor' paradigm. Namely, the verification key contains a matrix $\mathbf{A}$ sampled with a 'trapdoor' basis $\mathbf{T_A}$, and $\mathbf{A}_0, \mathbf{C}_0, \mathbf{A}_1, \mathbf{C}_1$, which can be interpreted as BV encodings of 0 and 1 respectively, as well as similar encodings $\{\mathbf{B}_i\}_{i\in[|k|]}$ of the bits of a key k for a bit-PRF (the use of this PRF is the key innovation in [BL16]). The corresponding signing key contains $\mathbf{T_A}$. To sign, one computes BV encodings $\mathbf{C}_{M_1}, \ldots, \mathbf{C}_{M_t}$ of a t-bit message M, then computes $\mathbf{A}_{\mathrm{PRF,M}} = \mathsf{Eval}_{\mathrm{BV}}(\{\mathbf{B}_i\}_{i\in[|k|]}, \{\mathbf{C}_j\}_{j\in[t]}, \mathsf{PRF})$. Two signing matrices $\mathbf{F}_{M,b} = [\mathbf{A} \mid \mathbf{A}_b - \mathbf{A}_{\mathrm{PRF,M}}]$ $(\forall b \in \{0,1\})$ are then generated (crucially, the adversary cannot tell these apart because of the PRF). A signature is a *short non-zero* vector $\sigma \in \mathbb{Z}^{2m}$ satisfying $\mathbf{F}_{M,b} \cdot \sigma = 0$ for any one of the $\mathbf{F}_{M,b}$'s. As pointed out, $\mathbf{T_A}$ allows the signer to produce a short vector for either $\mathbf{F}_{M,b}$.

To show unforgeability, one constructs a reduction that

1. replaces the left matrix with an SIS challenge (thus losing $\mathbf{T_A}$), **and**
2. replaces the other matrices used to generate the right half with their 'puncturable' versions (e.g., $\mathbf{A}_b$ now becomes $\mathbf{AR}_b + \mathbf{G}$, where $\mathbf{R}_b$ is an uniform low-norm matrix and $\mathbf{G}$ is the gadget matrix), with the end result being that the matrix $\mathbf{A}_{\mathrm{PRF,M}}$ becomes $\mathbf{AR}' + \mathbf{G}$ and $\mathbf{F}_{M,b}$ now looks like $[\mathbf{A} \mid \mathbf{AR} + (b - \mathsf{PRF}_k(M))\mathbf{G}]$ (with $\mathbf{R}, \mathbf{R}'$ being suitable low-norm matrices).

The crucial point is this: having sacrificed $\mathbf{T_A}$, the reduction cannot sign like a normal signer. However it still retains a trapdoor for the gadget matrix $\mathbf{G}$, and for *exactly one* of the $\mathbf{F}_{M,b}$, a term in G survives in the right half. This suffices to obtain a 'right trapdoor', and in turn, valid signatures for any M. On the other hand, a forging adversary lacks the PRF key and so it cannot tell apart $\mathbf{F}_{M,0}$ from $\mathbf{F}_{M,1}$. Thus the forgery must correspond to $\mathbf{F}_{M,\mathsf{PRF}_k(M)}$ with probability around $1/2$, and the reduction can use this solution to obtain a short solution for the challenge $\mathbf{A}$.

However, the blind-unforgeability setting differs in several meaningful ways. Here, we no longer expect a forgery for any possible message, so the additional machinery to have two signing matrices for every message becomes superfluous. Indeed, for us the challenge is to disallow signing queries in the blindset (even if they are made as part of a query superposition) and to prevent forgeries in the blindset. Accordingly, we interpret the function of the PRF in a different manner. We simply have the bit-PRF act as the characteristic function for the blindset. Then we can extend

the approach above to the blind-unforgeability setting very easily: we use a single signing matrix $\mathbf{F}_{\mathrm{M}} = [\mathbf{A} \mid \mathbf{A}' - \mathbf{A}_{\mathrm{PRF,M}}]$ (where $\mathbf{A}'$ 'encodes 1'). In the reduction, after making changes just as before, we obtain that $\mathbf{F}_{\mathrm{M}} = \big[\mathbf{A} \mid \mathbf{AR} - \big(1 - \mathsf{PRF}_k(M)\big)\mathbf{G}\big]$. For messages where the PRF is not 1, we can answer signing queries using the trapdoor for $\mathbf{G}$; For messages where it is 1, we cannot, and further we can use a forgery for such a message to break the underlying SIS challenge. In effect, the reduction enforces the requisite blindset behavior naturally.

A caveat is that the bit-PRF based approach may not correctly model a blindset, which is a random ε-weight set of messages. Indeed, we require a slight modification of a normal bit-PRF to allow us the necessary latitude in approximating sets of any weight $\varepsilon \in [0, 1]$. Moreover, due to the adversary's quantum access to the signing oracle, this PRF must be quantum-access secure; and to allow the BV homomorphic evaluation, the PRF must have NC^1 implementation. Fortunately, such a *biased* bit-PRF can be built by slightly modifying the PRF from [BPR12], assuming QLWE with super-polynomial modulus.

2.3 Post-Quantum Secure Ring Signatures

Defining Post-Quantum Security. To reflect the *quantum power* of an QPT adversary $\mathcal{A}$, one needs to give $\mathcal{A}$ quantum access to the signing oracle in the security game. While this is rather straightforward for anonymity, the challenge here is to find a proper notion for unforgeability (thus, here we only focus on the latter). Let us first recall the *classical* unforgeability game for a ring signature. In this game, $\mathcal{A}$ learns a ring $\mathcal{R}$ from the challenger, and then can make two types of queries:

- by a *corruption query* $(\mathsf{corrupt}, i)$, $\mathcal{A}$ can corrupt a member in $\mathcal{R}$ to learn its secret key;
- by a *signing query* $(\mathsf{sign}, i, \mathsf{R}^*, m)$, $\mathcal{A}$ can create a ring R^*, specify a member i that is contained in both $\mathcal{R}$ and R^*, and ask the challenger to sign a message m w.r.t. R^* using the signing keys of member i.

Notice that R^* may contain (potentially malicious) keys created by $\mathcal{A}$; but as long as the member i is in both R^* and $\mathcal{R}$, the challenger is able to sign m w.r.t. R^*. The challenger also maintains a set $\mathcal{C}$, which records all the members in $\mathcal{R}$ that are corrupted by $\mathcal{A}$. To win the game, $\mathcal{A}$ needs to output a forgery $(\mathsf{R}^*, m^*, \Sigma^*)$ satisfying the following 3 requirements:

1. $\mathsf{R}^* \subseteq \mathcal{R} \setminus \mathcal{C}$,
2. $\mathsf{RS.Verify}(\mathsf{R}^*, m^*, \Sigma^*) = 1$, **and**
3. $\mathcal{A}$ never made a signing query of the form $(\mathsf{sign}, \cdot, \mathcal{R}^*, m^*)$.

To consider quantum attacks, we first require that corruption queries should remain classical. In practice, corruption queries translate to the attack where a ring member is totally taken over by $\mathcal{A}$. Since ring signatures are a de-centralized primitive, corrupting a specific party should not affect other parties in the system. This situation arguably does not change with $\mathcal{A}$'s quantum power. One could of course consider "corrupting a group of users in superposition", but the motivation and practical implications of such corruptions is unclear, and thus we defer it to future research. In this work, we restrict ourselves to classical ring member corruptions.

We will allow $\mathcal{A}$ to conduct superposition attacks over the ring and message. That is, a QPT $\mathcal{A}$ can send singing queries of the form $(\mathsf{sign}, i, \sum_{\mathsf{R},m} \psi_{\mathsf{R},m} |\mathsf{R}, m\rangle)$, where the identity i is classical for the same reason above. Given the argument above, one may wonder why we allow superpositions

over R in the signing query. The reason is that unlike for corruption queries, each signing query specifies a specific member i to run the signing algorithm for. No matter what R is, this member will only sign using her own signing key (and this is the only signing key that she knows), and this has nothing to do with other parties in the system[6]. Therefore, superposition attacks over R can be validated just as superposition attacks over m, thus should be allowed.

The next step is to determine the winning condition for QPT adversaries in the above quantum unforgeability game. The approach taken by [CGH+21] is to extend the one-more unforgeability from [BZ13b] to the ring setting. Concretely, it is required that the adversary cannot produce $(\mathsf{sq}+1)$ valid signatures by making only sq quantum sign queries. However, there is a caveat. Recall that the R^* in $\mathcal{A}$'s forgery should be a subset of uncorrupted ring members (i.e., $\mathcal{R} \setminus \mathcal{C}$). A natural generalization of the "one-more forgery" approach here is to require that, with sq quantum signing queries, the adversary cannot produce $\mathsf{sq}+1$ forgery signatures, where *all* the rings contained are subsets of $\mathcal{R} \setminus \mathcal{C}$. This requirement turns out to be so strict that, when restricted to the classical setting, this one-more unforgeability seems to be weaker than the standard unforgeability for ring signatures (more details in Sec. 6.1.2 and Appx. B).

Our idea is to extend the blind-unforgeability definition to our setting. Specifically, the challenger will create a blind set $B_\varepsilon^{\mathsf{RS}}$ by including in each ring-message pair (R, m) with probability ε. It will then blind the signing algorithm such that it always returns "$\perp$" for $(\mathsf{R}, m) \in B_\varepsilon^{\mathsf{RS}}$. In contrast to one-more unforgeability, we will show that this definition, when restricted to the classical setting, is indeed equivalent to the standard unforgeability notion for ring signatures.

Our Construction. Our starting point is the LWE-based construction by Chatterjee et al. [CGH+21]. We first recall their construction: the public key consists of a public key for a public-key encryption scheme PKE and a verification key for a standard signature scheme Sig, as well as the first round message of a (bespoke) ZAP argument. To sign a message, one first computes an ordinary signature σ and then encrypts this along with a hash key hk for a specific hash function (i.e., *somewhere perfectly-binding* hash). Two such encryptions (c_1, c_2) are produced, along with the second-round message π of the ZAP proving that one of these encryptions is properly computed using a public key that is part of the presented ring. The hash key is extraneous to our concerns here; suffice it to say that it helps encode a "hash" of the ring into the signature and is a key feature in establishing compactness of their scheme.

To show anonymity, one starts with a signature for i_0, then switches the ciphertexts c_1 and c_2 in turn to be computed using the public key for i_1 while changing the ZAP accordingly. Semantic security ensures that ciphertexts with respect to different public keys are indistinguishable, and WI of the ZAP allows us to switch whichever ciphertext is not being used to prove π, and also to switch a proof for a ciphertext corresponding to i_0 to one corresponding to i_1.

Unforgeability in [CGH+21] follows from a reduction to the unforgeability of Sig. Even though their construction uses a custom ZAP that only offers soundness for (effectively) $\mathsf{NP} \cap \mathsf{coNP}$, they develop techniques in this regard to show that even with this ZAP, one can ensure that if an adversary produces a forgery with non-negligible probability, then it also encrypts a valid signature for Sig in one of c_1 or c_2 with non-negligible probability. The reduction can extract this using a corresponding decryption key (which it can obtain during key generation for the experiment) and use this as a forgery for Sig.

[6] Indeed, R may even contain "illegitimate" or "non-existent" members faked by $\mathcal{A}$. Note that we do not require $\mathsf{R} \subseteq \mathcal{R}$.

The [CGH+21] construction can thus be seen as a compiler from ordinary to ring signatures assuming LWE. We use their template as a starting point, but there are significant differences between security notions for standard (classical) ring signatures, and our (quantum) blind-unforgeability setting. We discuss these and how to accomodate them next. The very first change that we require here is to use a blind-unforgeable signature scheme in lieu of Sig, since we reduce unforgeability to that of Sig.

Next, let us discuss post-quantum anonymity. Here, the adversary can make a challenge query that contains a *superposition* over rings and messages. We would like to use the same approach as above, but of course computational indistiguishability is compromised against superposition queries. Two clear strengthenings are needed compared to the classical scheme: first, we need to use pairwise-independent hashing to generate signing randomness (to apply quantum oracle similarity techniques from [BZ13b]). Second, we want to ensure statistical similarity of the components c_1, c_2, π (in order to use an aforementioned lemma from [BZ13b] which says that pointwise statistically close oracles are indistiguishable even with quantum queries). In particular, the PKE needs to be statistically close on different plaintexts, and the WI guarantee for the ZAP needs to be statistical. Fortunately, we can use lossy encryption for the constraint on ciphertexts, and the ZAP from [CGH+21] is already statistical WI.

Finally, we turn to blind-unforgeability. Here, the things that change are that firstly, we need to switch to injective public keys (instead of lossy ones) to carry over the reduction from the classical case. Further, we forego using SPB hashing, because our techniques require that we sign the message along with the ring, i.e. $\mathsf{Sig.Sign}(sk, \mathsf{R}\|m)$. Thus we end up compromising compactness and using an SPB would serve no purpose. The reason that we need to sign the ring too has to do with how we define the blindset and how the challenger must maintain it in the course of the unforgeability game; this turns out to be more delicate than expected (see related discussion in Sec. 6.5). With the modifications above, we can eventually reduce the blind-unforgeability to that of Sig.

3 Preliminaries

Notation. For a set $\mathcal{X}$, let $2^{\mathcal{X}}$ denote the power set of $\mathcal{X}$ (i.e., the set of all subsets of $\mathcal{X}$. Let $\lambda \in \mathbb{N}$ denote the security parameter. A non-uniform QPT adversary is defined by $\{\mathsf{QC}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, where $\{\mathsf{QC}_\lambda\}_\lambda$ is a sequence of polynomial-size non-uniform quantum circuits, and $\{\rho_\lambda\}_\lambda$ is some polynomial-size sequence of mixed quantum states. For any function $F : \{0,1\}^n \to \{0,1\}^m$, "quantum access" will mean that each oracle call to F grants an invocation of the $(n+m)$-qubit unitary gate $|x,t\rangle \mapsto |x, t\oplus F(x)\rangle$; we stipulate that for any $t \in \{0,1\}^*$, we have $t\oplus\perp = \perp$. Symbols $\overset{\mathrm{c}}{\approx}$, $\overset{\mathrm{s}}{\approx}$ and $\overset{\mathrm{i.d.}}{=\!=}$ are used to denote computational, statistical, and perfect indistinguishability respectively. Computational indistinguishability in this work is by default w.r.t. non-uniform QPT adversaries.

We provide more preliminaries on lattices and the [BV14] key-homomorphic evaluation method in Appx. A.

3.1 Quantum Oracle Indistinguishability

We will need the following lemmata.

Lemma 1 ([Zha12b]). *Let H be an oracle drawn from a $2q$-wise independent distribution. Then, the advantage of any quantum algorithm making at most q queries to H has in distinguishing H from a truly random function is 0.*

Lemma 2 ([BZ13b]). *Let $\mathcal{X}$ and $\mathcal{Y}$ be sets, and for each $x \in \mathcal{X}$, let D_x and D'_x be distributions on $\mathcal{Y}$ such that $|D_x - D'_x| \leq \varepsilon$ for some value ε that is independent of x. Let $O : \mathcal{X} \to \mathcal{Y}$ be a function where, for each x, $O(x)$ is drawn from D_x, and let $O'(x)$ be a function where, for each x, $O'(x)$ is drawn from $D'(x)$. Then any quantum algorithm making at most q queries to either O or O' cannot distinguish the two, except with probability at most $\sqrt{8C_0q^3\varepsilon}$.*

3.2 Blind-Unforgeable Signatures

We recall in Def. 1 the definition for blind unforgeable signature schemes in [AMRS20]. The authors there provide a formal definition for MACs. We extend it in the natural way to the signature setting.

Definition 1 (Blind-Unforgeable Signatures). *For any security parameter $\lambda \in \mathbb{N}$, let $\mathcal{M}_\lambda$ denote the message space and $\mathcal{T}_\lambda$ denote the signature space. A* blind-unforgeable *signature scheme* Sig *consists of the following PPT algorithms:*

- $\mathsf{Gen}(1^\lambda)$ *outputs a verification and signing key pair (vk, sk).*
- $\mathsf{Sign}(sk, m; r)$ *takes as input a signing key sk, a message $m \in \mathcal{M}_\lambda$, and a randomness r (which we avoid specifying unless pertinent). It outputs a signature $\sigma \in \mathcal{T}_\lambda$.*
- $\mathsf{Verify}(vk, m, \sigma)$ *takes as input a verification key vk, a message $m \in \mathcal{M}_\lambda$ and a signature $\sigma \in \mathcal{T}_\lambda$. It outputs a bit signifying accept (1) or reject (0).*

These algorithms satisfy the following requirements:

1. **Completeness:** *For any $\lambda \in \mathbb{N}$, any (vk, sk) in the range of $\mathsf{Gen}(1^\lambda)$, and any $m \in \mathcal{M}_\lambda$, it holds that*
$$\Pr\big[\mathsf{Verify}\big(vk, m, \mathsf{Sign}(sk, m)\big) = 1\big] = 1 - \mathsf{negl}(\lambda).$$
2. **Blind-Unforgeability:** *For any non-uniform QPT adversary $\mathcal{A}$, it holds w.r.t. Expr. 1 that*
$$\mathsf{PQAdv}^\lambda_{\mathrm{BU}}(\mathcal{A}) := \Pr\left[\mathsf{PQExp}^\lambda_{\mathrm{BU}}(\mathcal{A}) = 1\right] \leq \mathsf{negl}(\lambda).$$

Experiment 1: Blind-Unforgeability Game $\mathsf{PQExp}^\lambda_{\mathrm{BU}}(\mathcal{A})$

1. $\mathcal{A}$ sends a constant $0 \leq \varepsilon \leq 1$ to the challenger;
2. The challenger generates $(vk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ and provides vk to $\mathcal{A}$.
3. The challenger defines a *blindset* $B^{\mathsf{Sig}}_\varepsilon \subseteq \mathcal{M}_\lambda$ as follows: every $m \in \mathcal{M}_\lambda$ is put in $B^{\mathsf{Sig}}_\varepsilon$ independently with probability ε.
4. $\mathcal{A}$ is allowed to make $\mathsf{poly}(\lambda)$ quantum queries. For each query, the challenger samples a (classical) random string r and performs the following mapping:
$$\sum_{m,t} \psi_{m,t}|m,t\rangle \mapsto \sum_{m,t} \psi_{m,t}|m, t \oplus B^{\mathsf{Sig}}_\varepsilon \mathsf{Sign}(sk, m; r)\rangle,$$
where $B^{\mathsf{Sig}}_\varepsilon \mathsf{Sign}(sk, m; r) = \begin{cases} \perp & \text{if } m \in B^{\mathsf{Sig}}_\varepsilon \\ \mathsf{Sign}(sk, m; r) & \text{otherwise} \end{cases}$.
5. Finally, $\mathcal{A}$ outputs (m^*, σ^*); the challenger checks if:

i. $m^* \in B_\varepsilon^{\mathsf{Sig}}$; **and**

ii. $\mathsf{Verify}(vk, m^*, \sigma^*) = 1$.

If so, the experiment outputs 1; otherwise, it outputs 0.

3. **Shortness (Optional):** *The signature scheme is short if the signature size is at most a polynomial on the security parameter* and *the logarithm of the message size.*

Remark 1 (One randomness to rule them all[7]). The signing algorithm in our definition samples signing randomness once per every query, as opposed to sampling signing randomness for every classical message in the superposition. This was established as a reasonable definitional choice in [BZ13b], where they observed that one could "de-randomize" the signing procedure by simply using a quantum PRF to generate randomness for each possible message in superposition, and use this for signing. We stick with this convention when defining post-quantum security for both ordinary signatures (Def. 1) and ring signatures (Def. 5 and 6).

Remark 2. We let the adversary choose ε. This is equivalent to quantifying over all values of ε as in the definition in [AMRS20].

3.3 Quantum-Access Secure Biased Bit-PRF

We will need a *quantum-access secure* PRF having a *biased single-bit* output. It should also be implementable by NC^1 circuits. Let us first present the definition.

Definition 2 (Biased Bit-QPRFs). *A biased bit-QPRF on domain* $\{0,1\}^{n(\lambda)}$ *consists of:*

- $\mathsf{Gen}(1^\lambda, \varepsilon)$*: takes as input a constant* $\varepsilon \in [0,1]$*, outputs a key* k_ε*;*
- $\mathsf{PRF}_{k_\varepsilon}(x)$*: takes as input* $x \in \{0,1\}^{n(\lambda)}$*, outputs a bit* $b \in \{0,1\}$*,*

such that for any $\varepsilon \in [0,1]$ *and any QPT* $\mathcal{A}$ *having* quantum access *to its oracle,*

$$\left| \Pr\left[k_\varepsilon \leftarrow \mathsf{Gen}(1^\lambda, \varepsilon) : \mathcal{A}^{\mathsf{PRF}_{k_\varepsilon}(\cdot)} = 1\right] - \Pr\left[F \xleftarrow{\$} \mathcal{F}\big(n(\lambda), \varepsilon\big) : \mathcal{A}^{F(\cdot)} = 1\right] \right| \leq \mathsf{negl}(\lambda),$$

where $\mathcal{F}\big(n(\lambda), \varepsilon\big)$ *is the collection of all functions from* $\{0,1\}^{n(\lambda)}$ *to* $\{0,1\}$ *that output* 1 *with probability* ε*.*

It is known that the NC^1 PRF from [BPR12] is quantum-access secure (i.e., a QPRF) [Zha12a]. It can be made biased by standard techniques (e.g., using the standard QPRF to "de-randomize" a ε-biased coin-tossing circuit). Note that the [BPR12] PRF relies on the quantum hardness of LWE with *super-polynomial* modulus. It is worthing noticing that such an LWE hardness assumption is stronger than the SIS assumption with polynomial modulus (see Def. 15).

4 Blind-Unforgeable Signatures in the QROM

We show here that the signature scheme in [GPV08] is a blind-unforgeable signature in the quantum random oracle model. This construction relies on the notion of *preimage sampleable functions.*

[7] Inspired by J. R. R. Tolkien. Indeed, this is a "ring" signature paper.

Definition 3 (Preimage Sampleable Functions [GPV08]). *A* preimage sampleable function *family* PSF *consists of the following PPT algorithms:*

- $\mathsf{Gen}(1^\lambda)$ *samples a public/secret key pair* (pk, sk).

For any (pk, sk) *in the range of* $\mathsf{Gen}(1^\lambda)$*:*

- $\mathsf{F}(pk, \cdot)$ *computes a function from set* $\mathcal{X}_\lambda$ *to set* $\mathcal{Y}_\lambda$.
- $\mathsf{Sample}(1^\lambda)$ *samples an* x *from some (possibly non-uniform) distribution* $\mathcal{X}_\lambda$ *such that* $\mathsf{F}(pk, x)$ *is distributed uniformly over* $\mathcal{Y}_\lambda$.
- $\mathsf{F}^{-1}(sk, y)$ *takes as input any* $y \in \mathcal{Y}_\lambda$ *and outputs a preimage* $x \in \mathcal{X}_\lambda$ *such that* $\mathsf{F}(pk, x) = y$, *and* x *is distributed statistically close to* $\mathsf{Sample}(1^\lambda)$ *conditioned on* $\mathsf{F}(pk, x) = y$.

These algorithms satisfy the following properties:

1. **Preimage Min-entropy:** *For each* $y \in \mathcal{Y}_\lambda$, *the conditional min-entropy of* $x \leftarrow \mathsf{Sample}(1^\lambda)$ *given* $\mathsf{F}(pk, x) = y$ *is at least* $\omega(\log n)$.
2. **Collision Resistance:** *For any QPT algorithm* $\mathcal{A}$, *the probability that* $\mathcal{A}(1^\lambda, pk)$ *outputs distinct* $x, x' \in \mathcal{X}_\lambda$ *such that* $F(pk, x) = \mathsf{F}(pk, x')$ *is negligible in* λ.

$$\Pr\left[\begin{matrix}(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda);\\ (x, x') \leftarrow \mathcal{A}(1^\lambda, pk)\end{matrix} : x \neq x' \ \wedge \ \mathsf{F}(pk, x) = \mathsf{F}(pk, x')\right] = \mathsf{negl}(\lambda).$$

[GPV08] constructs such PSFs based on the hardness of the SIS problem. They also give a signature scheme using PSFs, a hash function modeled as a random oracle, and a pseudorandom function. In the following, we first recall their signature scheme in Constr. 1, and then prove in Thm. 1 that this construction satisfies Def. 1 in the QROM if the PRF is a QPRF.

Construction 1: The GPV Signature [GPV08]

Let PSF be a preimage sampleable function. Let PRF be a pseudorandom function, and H be a hash function. The signature scheme Sig is defined as follows:

$\mathsf{Gen}(\lambda)$*:*

1. Generate $(sk', pk') \leftarrow \mathsf{PSF.Gen}(\lambda)$;
2. Sample a PRF key $k \leftarrow \mathsf{PRF.Gen}(1^\lambda)$;
3. Output $sk = (sk', k)$ and $pk = pk'$.

$\mathsf{Sign}(sk, m)$*:*

1. Compute $r \leftarrow \mathsf{PRF}(k, m)$ and $h = H(m)$;
2. Compute $\sigma = \mathsf{F}^{-1}(sk', h; r)$;
3. Output σ.

$\mathsf{Verify}(pk, m, \sigma)$*:*

1. Compute $h = H(m)$ and $h' = \mathsf{F}(pk', \sigma)$;
2. If $h = h'$ output 1; otherwise, output 0.

Theorem 1. *Assume that* PSF *be a preimage sampleable function, PRF is a quantum secure pseudorandom function, and* H *realizes a random oracle. Then Constr. 1 is a short blind-unforgeable signature in the quantum random oracle model.*

Proof. Completeness and shortness of Constr. 1 are straightforward. In the following, we prove blind-unforgeability.

The Joint Oracle. First notice that in the blind-unforgeability game in the ORAM, the adversaries has quantum oracle access to *two* oracles: H and $B_\varepsilon\mathsf{Sign}$. As we will show later (in particular, in hybrids H_1 and H_2 below), we need to change the way these two oracles are sampled, without being noticed by the adversary. We will need to argue the indistinguishability of this switch by invoking Lem. 2; but Lem. 2 is for adversaries that have access to a *single* oracle ($\mathcal{O}$ or $\mathcal{O}'$). Therefore, we start by slightly changing our oracle interface.

Instead of maintaining separate random and signing oracles (i.e., H and $B_\varepsilon\mathsf{Sign}$ respectively), we will maintain a single *joint oracle* $\mathcal{O}$ that can answer both types of queries 'jointly'. In more detail, we ask the adversary to include a *flag* bit $c \in \{0,1\}$ in each query. If $c = 0$, $\mathcal{O}$ will respond as H; if $c = 1$, it will respond as $B_\varepsilon\mathsf{Sign}$. Formally, $\mathcal{O}$ implements the following mapping:

$$\sum_{c,m,t} \Psi_{c,m,t} \left|c, m, t\right\rangle \mapsto \sum_{c,m,t} \Psi_{c,m,t} \left|c, m, t \oplus G(c,m)\right\rangle,$$

where $G(c,m) = \begin{cases} H(m) & c = 0 \\ B_\varepsilon\mathsf{Sign}(m) & c = 1 \end{cases}$.

This transformation is without loss of generality—indeed, any adversary $\mathcal{A}'$ that wins the original blind-unforgeability game can be transformed into an adversary $\mathcal{A}$ that breaks blind-unforgeability w.r.t. the joint oracle $\mathcal{O}$. $\mathcal{A}$ need only forward queries from A' to $\mathcal{O}$ and corresponding responses back to $\mathcal{A}'$ by setting a proper (classical) bit c . It is straightforward to see that $\mathcal{A}'$ gets identical responses whether interacting with $\mathcal{A}$ or in the QROM challenge. We therefore conclude that $\mathcal{A}$ has the same success probability as $\mathcal{A}'$, and this validates our single joint oracle interface. For the rest of this proof, we presuppose an adversary $\mathcal{A}$ that directly interacts with the joint oracle.

We will use $\mathcal{A}$ to obtain a contradiction, by employing hybrid arguments. For ease of exposition we will also use the following shorthand: denoting $\mathsf{F}(pk,\cdot)$ by $f(\cdot)$, and $\mathsf{F}^{-1}(sk,\cdot)$ by $f^{-1}(\cdot)$. Consider the following hybrid experiments:

Hybrid H_0**:** This is simply the normal blind-unforgeability challenge with the joint oracle. In particular, for each m in the superpostion of the adversary's quantum query $\Sigma_m\Psi_m \left|m\right\rangle$, the hash h is computed (implicitly) as the random oracle output $H(m)$, and the signature σ_m is computed according to $\sigma_m = f^{-1}(h;r)$ where $r = \mathsf{PRF}_k(m)$ (note that, in accordance with the challenge, the signing algorithm will be invoked only if $m \notin B_\varepsilon$).

Hybrid H_1**:** In this hybrid we change how r is generated. Instead of computing $r = \mathsf{PRF}_k(m)$, we set $r = J(m)$ where $J(\cdot)$ is a *random* function over the range and domain of PRF.[8]

$\underline{\mathsf{Out}(H_0) \overset{c}{\approx} \mathsf{Out}(H_1)}$*:* This follows directly from the quantum security of the PRF. Any adversary that has distinguishable outputs in these two hybrids is easily converted into a QPT algorithm that

[8] Note that the adversary's query is quantum: $\Sigma_m\Psi_m \left|m\right\rangle$. To keep the notation succinct, it suffices to describe the computation for each m in the superposition. We stick to this convention for later hybrids in this proof.

can distinguish between the QPRF and a uniformly random function given quantum oracle access to them in turn.

Hybrid H_2: In this hybrid we change both components of the oracle. The sining oracle is now re-defined as $\sigma_m = \mathsf{Sig.Sign}(m) := \mathsf{Sample}\big(1^\lambda; J(m)\big)$. Further, the hash oracle query is now answered by $h = f(\sigma_m)$. That is, when the adversary asks for $H(m)$, the hybrid first computes $\sigma_m = \mathsf{Sample}\big(1^\lambda; J(m)\big)$, and then returns $h = f(\sigma_m)$ to the adversary.

$\underline{\mathsf{Out}(H_1) \overset{s}{\approx} \mathsf{Out}(H_2)}$*:* Recall that we view the random oracle H and the signing oracle together as a joint oracle. The only thing that changes in H_2 is the computation of parts of the joint oracle response. We go through these carefully. For every m (in the superposition of $\mathcal{A}$'s quantum query), the response changes from

- $\Big(H(m),\ f^{-1}\big(H(m); J(m)\big)\Big)$, i.e., the joint oracle in H_1, **to**
- $\Big(f\big(\mathsf{Sample}(\lambda; J(m))\big),\ \mathsf{Sample}\big(\lambda; J(m)\big)\Big)$, i.e, the joint oracle in H_2.

Note that both $H(m)$ and $J(m)$ are uniformly random. Therefore, by the properties of Sample and f^{-1} (Def. 3), the above two distributions are statistically close to each other. Denoting the joint oracle in H_1 by $\mathcal{O}_1$, and that in H_2 by $\mathcal{O}_2$, we conclude that for any (classical) query point m, the distributions of the responses returned by $\mathcal{O}_1$ and $\mathcal{O}_2$ conditioned on m are statistically close, say less than distance $\Delta(\lambda)$ (which is negligible in λ). Now since $\mathcal{A}$ is a quantum machine making at most polynomially (say $q(\lambda)$) many quantum queries. Then, we can use Lem. 2 to conclude that $\mathcal{A}$ distinguishes between $\mathcal{O}_1$ and $\mathcal{O}_2$ with probability at most $\sqrt{8C_0q^3\Delta}$, which is negligible in λ.

Hybrid H_3: Observe that the hybrid H_2 is *not* efficiently implementable as it needs to sample a random function $J(\cdot)$. In the classical setting, H_2 can be made efficient by lazy-sampling $J(\cdot)$; however, here we cannot resort to lazy-sampling as the adversary has *quantum* access to the oracle. Thus, we take the following alternative approach to have an efficient hybrid. Assume q is the upper-bound of the number of quantum queries made by the adversary. In hybrid H_3, we sample a $2q$-wise independent hash function $J'(\cdot)$, and replace the random function $J(\cdot)$ with $J'(\cdot)$. Everything else remains the same as in H_2.

$\underline{\mathsf{Out}(H_2) \overset{i.d.}{=\!=} \mathsf{Out}(H_3)}$: This follows immediately from Lem. 1.

Reducing to the security of PSF. Now consider the eventual forgery output by $\mathcal{A}$ in H_3, (m^*, σ_m^*). If this is valid, it follows from Def. 1 that $m^* \in B_\varepsilon$ and $\mathsf{Sig.Verify}(vk, m^*, \sigma_m^*) = 1$, which means $H(m^*) = f(\sigma_m^*)$.

Let $\sigma'_m := \mathsf{Sample}(1^\lambda; J'(m^*))$. Note that this is exactly $\mathsf{Sig.Sign}$'s output on m^* in H_3. Due to the way H_3 implements the oracles, this presents only two possibilities:

1. Either we have $\sigma'_m = \sigma_m^*$, in which $\mathcal{A}$ is able to pick out the value σ'_m among all possible preimages of $h = f(\sigma_m^*)$. Observe that $m^* \in B_\varepsilon$, which means that $\mathcal{A}$ never saw that value σ'_m before as $\mathsf{Sig.Sign}$ returns $\top$ for messages in B_ε. Therefore, by the preimage min-entropy property (Item 1) of PSF, the set of allowed pre-images for h has conditional minentropy at least $\omega(\log\lambda)$, which means that $\mathcal{A}$ can only predict this value with at most negligible probability, so this case has a negligible chance of occurrence.

2. Else, we must have $\sigma'_m \neq \sigma^*_m$ in which case $\mathcal{A}$ has obtained colliding preimages, i.e. $\sigma'_m \neq \sigma^*_m$ such that $f(\sigma'_m) = f(\sigma^*_m)$. This contradicts the collision resistance of PSF (note that this is the reason why we need H_3 to be efficiently implementable), so this case can also only arise with at most negligible probability.

We conclude that a successful forgery occurs only with negligible probability if the PSF satisfies the aforementioned properties. □

5 Blind-Unforgeable Signatures in the Plain Model

5.1 Notation and Building Blocks

We assume familiarity with standard lattice-based cryptographic notions and procedures. Here we will recall certain techniques and properties to be directly used in our plain model construction. We recall standard lattice-related concepts (e.g., parameters, hardness, trapdoors) in Appx. A.1.

For a vector $\mathbf{u}$, we let $||\mathbf{u}||$ denote its ℓ_2 norm. For a matrix $\mathbf{R} \in \mathbb{Z}^{k \times m}$, we define two matrix norms:

- $||\mathbf{R}||$ denotes the ℓ_2 norm of the largest column of $\mathbf{R}$;
- $||\mathbf{R}||_2$ denotes the operator norm of $\mathbf{R}$, defined as $||\mathbf{R}||_2 = \mathsf{sup}_{\mathbf{x} \in \mathbb{R}^{m+1}} ||\mathbf{R} \cdot \mathbf{x}||$.

We denote the Gram-Schmidt ordered orthogonalization of a matrix $\mathbf{A} \in \mathbb{Z}^{m \times m}$ by $\widetilde{\mathbf{A}}$. For a prime q, a modular matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and vector $\mathbf{u} \in \mathbb{Z}_q^n$, we define the m-dimensional (full rank) lattice $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A}\mathbf{e} = \mathbf{u} \pmod q\}$. In particular, $\Lambda_q^{\perp}(\mathbf{A})$ denotes the lattice $\Lambda_q^{\mathbf{0}}(\mathbf{A})$.

Lattice Sampling Algorithms. Our construction uses the 'left-right trapdoors' framework introduced in [ABB10, Boy10], which uses two sampling algorithms SampleLeft and SampleRight.

SampleLeft*:* The algorithm SampleLeft works as follows:

- *Inputs:* A full-rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a short basis $\mathbf{T_A}$ of $\Lambda_q^{\perp}(\mathbf{A})$, along with a matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m_1}$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a Gaussian parameter s.
- *Output:* Let $\mathbf{F} = [\mathbf{A} \mid \mathbf{B}]$. SampleLeft outputs a vector $\mathbf{d} \in \mathbb{Z}^{m+m_1}$ in $\Lambda_q^{\mathbf{u}}(\mathbf{F})$.

Theorem 2 (SampleLeft **Closeness [ABB10, CHKP10]).** *Let $q > 2$, $m > n$ and $s > ||\widetilde{\mathbf{T}}_{\mathbf{A}}|| \cdot \omega\big(\sqrt{\log(m+m_1)}\big)$. Then, the* SampleLeft$(\mathbf{A}, \mathbf{B}, \mathbf{T_A}, \mathbf{u}, s)$ *(as defined above) outputs $\mathbf{d} \in \mathbb{Z}^{m+m_1}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{F}),s}$.*

SampleRight*:* The algorithm SampleRight works as follows:

- *Inputs:* Matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times k}$ and $\mathbf{R} \in \mathbb{Z}_q^{k \times m}$, a full-rank matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$, a short basis $\mathbf{T_B}$ of $\Lambda_q^{\perp}(\mathbf{B})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a Gaussian parameter s.
- *Output:* Let $\mathbf{F} = [\mathbf{A} \mid \mathbf{AR} + \mathbf{B}]$. It outputs a vector $\mathbf{d} \in \mathbb{Z}^{m+m_1}$ in the set $\Lambda_q^{\mathbf{u}}(\mathbf{F})$.

Theorem 3 (SampleRight **Closeness [ABB10]).** *Let $q > 2$, $m > n$ and $s > ||\widetilde{\mathbf{T}}_{\mathbf{B}}|| \cdot ||\mathbf{R}||_2 \cdot \omega(\sqrt{\log m})$. Then* SampleRight$(\mathbf{A}, \mathbf{B}, \mathbf{R}, \mathbf{T_B}, \mathbf{u}, s)$ *(as defined above) outputs $\mathbf{d} \in \mathbb{Z}^{m+k}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{F}),s}$.*

Random Sampling Related. The following is a simple corollary of [ABB10, Lemma 4] (see Appx. A.2 for details).

Corollary 1. *Suppose that $m > (n+1)\log_2 q + \omega(\log n)$ and that $q > 2$ is a prime. Let $\mathbf{R}$ be an $m \times k$ matrix chosen uniformly from $\{-1,1\}^{m\times k} \mod q$ where $k = k(n)$ is polynomial in n. Let $\mathbf{A}' \in \mathbb{Z}_q^{n\times m}$ be sampled from a distribution statistically close to uniform over $\mathbb{Z}_q^{n\times m}$. Let $\mathbf{R}$ be an $m \times k$ matrix chosen uniformly from $\{-1,1\}^{m\times k} \mod q$ where $k = k(n)$ is polynomial in n. Let $\mathbf{B}$ be chosen uniformly in $\mathbb{Z}_q^{n\times k}$. Then for all vectors $\mathbf{w} \in \mathbb{Z}_q^m$, the distributions $(\mathbf{A}', \mathbf{A}'\mathbf{R}, \mathbf{R}^\top\mathbf{w})$ and $(\mathbf{A}', \mathbf{B}, \mathbf{R}^\top\mathbf{w})$ are statistically close.*

Key-Homomorphic Evaluation. We briefly recall the matrix key-homomorphic evaluation algorithm, as found in [GSW13, BGG+14, BV14] (see Appx. A.3 for details). This template evaluates NAND circuits, gate by gate, in a homomorphic manner. For a NAND gate $g(u, v; w)$ with input wires u, v and output wire w, we have (inductively) matrices $\mathbf{A}_u = \mathbf{A}\mathbf{R}_u + x_u\mathbf{G}$, and $\mathbf{A}_v = \mathbf{A}\mathbf{R}_v + x_v\mathbf{G}$ where x_u and x_v are the input bits of u and v, and the evaluation algorithm computes:

$$\mathbf{A}_w = \mathbf{G} - \mathbf{A}_\mathbf{u} \cdot \mathbf{G}^{-1}(\mathbf{A}_v) = \mathbf{G} - (\mathbf{A}\mathbf{R}_u + x_u\mathbf{G}) \cdot \mathbf{G}^{-1}(\mathbf{A}\mathbf{R}_v + x_v\mathbf{G}) = \mathbf{A}\mathbf{R}_g + (1 - x_u x_v)\mathbf{G},$$

where $1 - x_u x_v \coloneqq \mathsf{NAND}(x_u, x_v)$, and $\mathbf{R}_g = -\mathbf{R}_u \cdot \mathbf{G}^{-1}(\mathbf{A}_v) - x_u\mathbf{R}_v$ has low norm if both $\mathbf{R}_u$ and $\mathbf{R}_v$ have low norm.

Biased Bit-QPRF. We will need a single-bit-output QPRF that output 1 with a customizable probability ε. Moreover, we need it to be implementable in NC^1. Such a QPRF can be built using the PRF constructed in [BPR12] assuming QLWE with super-polynomial modulus. See Sec. 3.3 for more details and the formal definition.

5.2 Our Construction

Our signature scheme uses a biased bit QPRF PRF whose input space $\mathcal{X}$ corresponds to our message space $\mathcal{M}$, and the algorithms SampleLeft, SampleRight given as in Thm. 2 and Thm. 3 respectively, and TrapGen that can sample matrices in $\mathbb{Z}_q^{n\times m}$ statistically close to uniform, along with a corresponding 'short' or 'trapdoor' basis for the associated lattice. This is formally defined in Lem. 8. The construction is as follows:

Construction 2: Blind-Unforgeable Signatures in the Plain Model

Paramters: Set message length $t(\lambda)$ and row size $n(\lambda)$ as free parameters (polynomial in λ). PRF key size is set as $k(\lambda)$, and the depth for $\mathsf{C}_{\mathsf{PRF}}$ is given by $d(\lambda)$. We set $m = n^{1+\eta}$ for proper running of TrapGen, and $\mathsf{sigsize}_\lambda = s\sqrt{2m}$ for the validity of SampleLeft output (to ensure completeness). Set $s = O(4^d m^{3/2}) \cdot \omega(\sqrt{\log m})$ to ensure statistical closeness of SampleLeft and SampleRight, and correspondingly set $\beta = O(16^d m^{7/2}) \cdot \omega(\sqrt{\log m})$ and $q = O(16^d m^4) \cdot \left(\omega(\sqrt{\log m})\right)^2$ to have an overall reduction to an appropriately hard instance of SIS. For further details about these choices, see Sec. 5.3.

$\mathsf{Gen}(1^\lambda)$:

1. Sample a matrix $\mathbf{A}$ along with a 'trapdoor' basis $\mathbf{T}_\mathbf{A}$ for $\Lambda_q^\perp(\mathbf{A})$ using TrapGen.

2. Sample a matrix $\mathbf{A}'$, 'PRF key' matrices $\mathbf{B}_1, \ldots, \mathbf{B}_k$, and 'PRF input' matrices $\mathbf{C}_0, \mathbf{C}_1$ uniformly from $\mathbb{Z}_q^{n \times m}$ (k is the PRF key length).
3. Fix the Gaussian width parameter s as given in parameter selection.
4. Fix a Boolean circuit description $\mathsf{C}_{\mathsf{PRF}}$ of the algorithm $\mathsf{PRF}_{(\cdot)}(\cdot)$.
5. Output $vk = (\mathbf{A}, \mathbf{A}', \{\mathbf{B}_i\}_{i=1}^k, \{\mathbf{C}_0, \mathbf{C}_1\}, \mathsf{PRF}, s, \mathbf{C}_{\mathsf{PRF}})$ and $sk = \mathbf{T}_{\mathbf{A}}$.

$\underline{\mathsf{Sign}(sk, vk, M)}$: let $(M_1, \ldots, M_t) \in \{0,1\}^t$ be the bit-wise representation of M.

1. Run the [BV14] evaluation algorithm $\mathsf{Eval}_{\mathrm{BV}}$ to homomorphically evaluate the circuit $\mathbf{C}_{\mathsf{PRF}}$ using the 'encoded' PRF key bits $\{\mathbf{B}_i\}_{i \in [k]}$ and message bits $\{\mathbf{C}_{M_j}\}_{j \in [t]}$. This yields
$$\mathbf{A}_{\mathrm{PRF,M}} \coloneqq \mathsf{Eval}_{\mathrm{BV}}(\mathbf{C}_{\mathsf{PRF}}, \{\mathbf{B}_i\}_{i \in [k]}, \{\mathbf{C}_{M_j}\}_{j \in [t]}) \in \mathbb{Z}_q^{n \times m}.$$
2. Set $\mathbf{F}_{\mathrm{M}} \coloneqq [\mathbf{A} \mid \mathbf{A}' - \mathbf{A}_{\mathrm{PRF,M}}]$.
3. Use $\mathsf{SampleLeft}$ to obtain $\mathbf{d}_{\mathrm{M}}$ with distribution statistically close to $\leftarrow \mathcal{D}_{\Lambda_q^{\perp}(\mathbf{F}_{\mathrm{M}}), s}$ (see Thm. 2).
4. Output $\sigma = \mathbf{d}_{\mathrm{M}} \in \mathbb{Z}_q^{2m}$.

$\underline{\mathsf{Verify}(vk, M, \sigma)}$:

1. Compute $\mathbf{A}_{\mathrm{PRF,M}}$, $\mathbf{F}_{\mathrm{M}}$ as before.
2. Check that $\sigma \in \mathbb{Z}_q^{2m}$, $\sigma \neq 0$, and $||\sigma|| \leq \mathsf{sigsize}_\lambda$. If it fails, output 0.
3. If $\mathbf{F}_{\mathrm{M}} \cdot \sigma = 0 \mod q$, output 1, otherwise output 0.

5.3 Parameter Selection for Constr. 2

The security parameter λ is represented as before. We set message length $t(\lambda)$ and row size $n(\lambda)$ as free parameters (polynomial in λ). PRF key size is set as $k(\lambda)$, and the depth for $\mathsf{C}_{\mathsf{PRF}}$ is set to be $d(\lambda)$. We must set the parameters properly to ensure that the following conditions are satisfied:

1. We must have $m = n^{1+\eta}$, with η being given by $n^\eta > O(\log q)$. This is to ensure that $\mathsf{TrapGen}$ can run properly, by Lem. 8.
2. We require that $s > ||\widetilde{\mathbf{T}}_{\mathbf{G}}|| \cdot ||\mathbf{R}||_2 \cdot \omega(\sqrt{\log m})$, where $\mathbf{R} = \mathbf{R}_{\mathbf{A}'} - \mathbf{R}_{\mathbf{A}_{\mathrm{PRF,M}}}$ (the latter will be defined in the course of the proof), for the statistical similarity of $\mathsf{SampleLeft}$ and $\mathsf{SampleRight}$, as per Thm. 2 and 3.
3. Since signatures are vectors of length $2m$ over $\mathbb{Z}_q$ sampled from (statistically close to) $\mathcal{D}_{\Lambda_q^{\perp}(\mathbf{F}_m), s}$, for most honestly generated signatures to be valid, it is necessary to set $\mathsf{sigsize}_\lambda \geq s\sqrt{2m}$, in accordance with Lem. 9.
4. For hardness of the SIS instance, we require that the width parameter β satisfy $\beta \geq O(4^d \cdot m^{3/2} \cdot s\sqrt{2m})$.
5. Finally, for standard average-to-worst case hardness reductions to apply for SIS, we require that $q \geq \beta \cdot \omega(\sqrt{n \log n})$.

   Accordingly, we set the remaining parameters as follows:

– We set $m = n^{1+\eta}, \mathsf{sigsize}_\lambda = s\sqrt{2m}$ just as indicated.

– By the bounds on $\mathbf{R}_{\mathbf{A}_{\mathrm{PRF,M}}}$ implied by Lem. 14, it suffices to set $s = O(4^d m^{3/2}) \cdot \omega(\sqrt{\log m})$ to satisfy constraint 2 above.

– Using the above s and so as to just about satisfy constraint 4, we set $\beta = O(16^d m^{7/2}) \cdot \omega(\sqrt{\log m})$.

– We set q based on β above so as to just satisfy the final constraint, namely $q = O(16^d m^4) \cdot \omega(\sqrt{\log m})^2$.

Since we consider PRFs in NC^1, we can write $d = c \log \ell$ (for some constant c) where $\ell = t + k$ is the input length for the PRF. This yields $\beta = O(\ell^{4c} m^{7/2}) \cdot \omega(\sqrt{\log m})$ and $q = O(\ell^{4c} m^4) \cdot \omega(\sqrt{\log m})^2$.

5.4 Proof of Security

Completeness follows straightforwardly from the correctness of $\mathsf{SampleLeft}$ (Thm. 2) for $\mathcal{D}_{\Lambda_q^\perp(\mathbf{F}),s}$. In the following, we prove BU-security.

Theorem 4. *Let λ denote the security parameter, and PRF be a biased bit QPRF as defined in Def. 2 above. If the parameters n, m, q, β, s, d are picked as discussed above, and the $\mathbf{SIS}_{q,\beta,n,m}$ problem is hard for QPT adversaries, then our signature scheme Sig constructed as above, with the indicated parameters, satisfies Blind-Unforgeability as in Def. 6.*

Proof. Consider a QPT $\mathcal{A}$ that is able to produce forgeries w.r.t. Sig in the blind-unforgeability challenge. Our proof proceeds using a series of hybrid experiments. In the final hybrid we show a reduction from an adversary producing succesful forgeries to the hardness of $\mathbf{SIS}_{q,\beta,n,m}$. The hybrids are as follows:

Hybrid H_0: This is the blind-unforgeability game (Expr. 1). Namely, for an adversary-specified ε, the challenger manually samples an ε-weight set B_ε over messages, and does not answer queries in B_ε. Signing and verification keys are chosen just as in the ordinary signing procedure.

Hybrid H_1: This hybrid is identical to the previous one, except that we change the ordinary key generation into the following:

1. Sample $\mathbf{A}$ with a 'trapdoor' basis $\mathbf{T_A}$ for $\Lambda_q^\perp(\mathbf{A})$ using $\mathsf{TrapGen}$ as before.
2. Sample 'low-norm' matrices: $\mathbf{R'_A}, \{\mathbf{R_{B}}_i\}_{i=1}^k, \mathbf{R_{C}}_0, \mathbf{R_{C}}_1 \xleftarrow{\$} \{-1,1\}^{m \times m}$.
3. Let PRF and $\mathsf{C_{PRF}}$ be as before.
4. Sample a PRF key $k_\varepsilon \leftarrow \mathsf{PRF.Gen}(1^\lambda, \varepsilon)$, where $k_\varepsilon = s_1, \ldots, s_k$ (i.e. has length k).
5. Set $\mathbf{A'} = \mathbf{A}\mathbf{R}_{\mathbf{A'}} + \mathbf{G}$, where $\mathbf{G}$ the gadget matrix $\mathbf{G}$, which has a publicly-known trapdoor $\widetilde{\mathbf{T}}_{\mathbf{G}}$ (as described in Lem. 10).
6. Set $\mathbf{C}_b = \mathbf{A}\mathbf{R}_{\mathbf{C}_b} + b\mathbf{G}$ for $b \in \{0,1\}$, and sample $\mathbf{B}_i \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ for every $i \in [k]$.
7. Fix the Gaussian width parameter s as before.
8. Output $vk = (\mathbf{A}, \mathbf{A'}, \{\mathbf{B}_i\}_{i=1}^k, \{\mathbf{C}_0, \mathbf{C}_1\}, s, \mathsf{PRF}, \mathbf{C_{PRF}})$, and $sk = (\mathbf{T_A}, k_\varepsilon)$.

Note that while this hybrid generates a key k_ε, it never uses it.

$\underline{H_0 \overset{s}{\approx} H_1:}$ The only thing that changes (w.r.t. $\mathcal{A}$) is the distribution of the various components $(\mathbf{A}', \mathbf{C}_0, \mathbf{C}_1)$ of the verification key handed out by the challenger. However, by Corollary 1 these distributions are all statistically close to the corresponding distributions in H_0. Note that the verification key is picked at the start of the challenge and provided to $\mathcal{A}$, so there is no scope for $\mathcal{A}$ to have quantum access to these component distributions. Thus the outputs in these hybrids are statistically close.

Hybrid H_2: This hybrid is identical to the previous one, except that we change how the challenger picks the blindset—Instead of manually sampling B_ε as a random ε-weight set, it now sets B_ε to be the set of messages M where $\mathsf{PRF}_{k_\varepsilon}(M)$ is 1 (note that the challenger now possesses k_ε as part of sk, and can compute $\mathsf{PRF}_{k_\varepsilon}(\cdot)$). Observe that the challenger in this hybrid is now efficient.

$\underline{H_1 \overset{c}{\approx} H_2:}$ Note that setup and key generation in H_2 is identical to that in H_1—In particular, the adversary learns *no* information about the key k_ε. The indistinguishability between H_1 and H_2 then follows immediately from the security of the biased bit-QPRF (Def. 2).

Hybrid H_3: This hybrid is identical to the previous one, except that we change how the matrices $\mathbf{B}_i$'s (in Step 6) are generated. Namely, we now set

$$\forall i \in [k], \quad \mathbf{B}_i \coloneqq \mathbf{A}\mathbf{R}_{\mathbf{B}_i} + s_i \cdot \mathbf{G}.$$

(Recall that s_i is the i-th bit of the k_ε generated in Step 4.)

$\underline{H_2 \overset{s}{\approx} H_3:}$ The only things that change between these hybrids are the matrices $\{\mathbf{B}_i\}_{i\in[k]}$. Again, using Corollary 1 the distributions for $\mathbf{B}_i$ for each $i \in [k]$ are all statistically close to the corresponding distributions in H_2, and just as in the similarity argument between H_2 and H_3, we can conclude that these hybrids too have indistinguishable outputs.

Hybrid H_4: Observe that, starting from H_1, we have:

$$\begin{aligned}
\mathbf{F}_{\mathrm{M}} &= [\mathbf{A} \mid \mathbf{A}' - \mathbf{A}_{\mathrm{PRF,M}}] = \left[\mathbf{A} \mid \mathbf{A}' - \mathsf{Eval}_{\mathrm{BV}}(\mathbf{C}_{\mathsf{PRF}}, \{\mathbf{B}_i\}_{i\in[k]}, \{\mathbf{C}_{M_j}\}_{j\in[t]})\right] \\
&= \left[\mathbf{A} \mid \mathbf{A}' - \left(\mathbf{A}\mathbf{R}_{\mathrm{PRF,M}} + \mathsf{PRF}_{k_\varepsilon}(M) \cdot \mathbf{G}\right)\right] \\
&= \left[\mathbf{A} \mid \mathbf{A}(\mathbf{R}_{\mathbf{A}'} - \mathbf{R}_{\mathrm{PRF,M}}) + \left(1 - \mathsf{PRF}_{k_\varepsilon}(M)\right) \cdot \mathbf{G}\right].
\end{aligned}$$

In this hybrid, we switch to using $\mathsf{SampleRight}$ to answer signing queries, instead of using $\mathsf{SampleLeft}$. That is, we run $\mathsf{SampleRight}$ using $\mathbf{T}_{\mathbf{G}}$, the publicly available trapdoor for $\mathbf{G}$. Note this means that now the challenger cannot answer queries where the 'right half' of $\mathbf{F}_{\mathrm{M}}$ does not include $\mathbf{G}$, i.e., $\mathsf{PRF}_{k_\varepsilon}(M) = 1$. But due to the way H_2 generate the blindset, such a query is anyway answered with "$\perp$".

$\underline{H_3 \overset{c}{\approx} H_4:}$ We first show that these two hybrids answer signature queries for any *classical* query M in a *statistically* indistinguishable manner. For any query M, there are two cases: (1) if $\mathsf{PRF}_{k_\varepsilon}(M) = 1$, the challengers in both H_3 and H_4 return $\perp$. In this case, these distributions are identical. (2) Else, we have $\mathsf{PRF}_{k_\varepsilon}(M) = 0$. Since $\mathbf{F}_{\mathrm{M}}$ is computed identically in both hybrids, and by Thm. 2 and 3 both $\mathsf{SampleLeft}$ and $\mathsf{SampleRight}$ sample from distributions statistically close to $\mathcal{D}_{\Lambda_q^\perp(\mathbf{F}_{\mathrm{M}}),s}$, i.e., they are also statistically close to each other. Thus overall the distributions of signatures returned in H_3 and H_4 are statistically close to each other, say with less than distance $\Delta(\lambda)$ (which is

negligible in λ). Now since $\mathcal{A}$ is a quantum machine making at most polynomially (say $q(\lambda)$) many quantum queries. Then, we can use Lem. 2 to conclude that $\mathcal{A}$ distinguishes between H_3 and H_4 with probability at most $\sqrt{8C_0q^3\Delta}$, which is negligible.

Hybrid H_5: In this hybrid, the challenger no longer samples $\mathbf{A}$ using $\mathsf{TrapGen}$. Instead, it samples $\mathbf{A}$ uniformly from $\mathbb{Z}_q^{n\times m}$.

$H_4 \overset{s}{\approx} H_5$: This follows immediately from Lem. 8.

Reduction to QSIS. We can now describe our reduction $\mathcal{R}$ in this hybrid:

1. Asks for and recieves a uniform matrix in $\mathbb{Z}_q^{n\times m}$ as the $\mathbf{SIS}_{q,\beta,n,m}$ challenge.
2. Sets $\mathbf{A}$ to be this matrix (instead of sampling $\mathbf{A}$ by itself).
3. When the adversary returns a forgery (M^*, σ^*), $\mathcal{R}$ checks if this is valid, i.e., that (i) $M^* \in B_\varepsilon$, (ii) $\sigma^* \in \mathbb{Z}_q^{2m}$, (iii) $\sigma^* \neq 0$, (iv) $\mathbf{F}_{\mathrm{M}^*} \cdot \sigma^* = 0 \mod q$ and (v) $||\sigma|| \leq \mathsf{sigsize}_\lambda$. If any of these checks fail, it aborts.
4. Represent σ^* as $[\mathbf{d}_1^\top \mid \mathbf{d}_2^\top]^\top$, with $\mathbf{d}_1, \mathbf{d}_2 \in \mathbb{Z}_q^m$. $\mathcal{R}$ computes $\mathbf{e} = \mathbf{d}_1 + \mathbf{R}\mathbf{d}_2$ where $\mathbf{R} = \mathbf{R}_{\mathbf{A}'} - \mathbf{R}_{\mathrm{PRF,M}}$ (we will use this shorthand going forward), and presents $\mathbf{e}$ as its solution to the SIS challenge $\mathbf{A}$.

Now we can prove that $\mathbf{e}$ is indeed an SIS solution with non-negligible probability by an argument very similar as in the final reduction for [BL16, Theorem 3.1]. We present the final reduction in the following.

The Final Reduction. Before showing that the reduction's output $\mathbf{e}$ indeed breaks the given SIS challenge, we must first examine the possibility of a 'related message' attack. Namely, we want to avoid a situation where the adversary can directly use signatures for one message to get signatures on another since this would render our reduction moot. We show that this is not the case by showing that an adversary cannot come up with two messages M, M' such that $\mathbf{F}_{\mathrm{M}} = \mathbf{F}'_{\mathrm{M}}$. The following lemma accomplishes this task.

Lemma 3. *If a QPT adversary produces two distinct messages M, M' such that $\mathbf{A}_{\mathrm{PRF,M}} = \mathbf{A}_{\mathrm{PRF,M}'}$ with non-negligible probability, then we can break the $\mathbf{SIS}_{q,\beta,n,m}$ challenge.*

Proof. With the verification key in H_5 picked just as in H_2, if $\mathbf{A}_{\mathrm{PRF,M}} = \mathbf{A}_{\mathrm{PRF,M}'}$, then we have

$$\mathbf{A}\mathbf{R}_{\mathrm{PRF,M}} + \mathsf{PRF}_{k_\varepsilon}(M)\mathbf{G} = \mathbf{A}\mathbf{R}_{\mathrm{PRF,M}'} + \mathsf{PRF}_{k_\varepsilon}(M')\mathbf{G}.$$

Note that we have $\mathsf{PRF}_{k_\varepsilon}(M) \neq \mathsf{PRF}_{k_\varepsilon}(M')$ with probability $2\varepsilon \cdot (1-\varepsilon)$, which is a constant. If this holds, we have $\mathbf{A}(\mathbf{R}_{\mathrm{PRF,M}} - \mathbf{R}_{\mathrm{PRF,M}'}) \pm \mathbf{G} = 0 \mod q$. Now by Lem. 10 and using $\mathsf{SampleRight}$ we can find a low-norm vector $\mathbf{d} \in \mathbb{Z}_q^{m\times m}$ such that $\mathbf{G}\mathbf{d} = 0 \mod q$, $\mathbf{d} \neq 0$ and $||\mathbf{d}|| \leq s'\sqrt{2m}$ (for some $s' \geq \sqrt{5}\omega(\sqrt{\log m})$). Then $[\mathbf{A}(\mathbf{R}_{\mathrm{PRF,M}} - \mathbf{R}_{\mathrm{PRF,M}'}) \pm \mathbf{G}]\mathbf{d} = 0 \mod q$, yielding $\mathbf{A}(\mathbf{R}_{\mathrm{PRF,M}} - \mathbf{R}_{\mathrm{PRF,M}'})\mathbf{d} = 0 \mod q$. By our choice of parameters, $(\mathbf{R}_{\mathrm{PRF,M}} - \mathbf{R}_{\mathrm{PRF,M}'})$ has low enough norm and so $(\mathbf{R}_{\mathrm{PRF,M}} - \mathbf{R}_{\mathrm{PRF,M}'})\mathbf{d}$ is a valid SIS solution for $\mathbf{A}$. This happens with non-negligible probability using our starting assumption, and thus we break $\mathbf{SIS}_{q,\beta,n,m}$ as claimed. □

Now we can turn to validating our reduction. It is straightforward to verify that if σ^* is a valid signature, then $\mathbf{e}$ is a valid integer solution to $\mathbf{A}$. Indeed, we have $\mathbf{F}_{\mathrm{M}^*} \cdot \sigma^* = 0 \mod q$, which from

the above boils down to

$$[\mathbf{A} \mid \mathbf{A}(\mathbf{R}_{\mathbf{A}'} - \mathbf{R}_{\text{PRF,M}}) + \big(1 - \mathsf{PRF}_{k_\varepsilon}(M)\big)\mathbf{G}] \cdot \sigma^* = 0 \mod q,$$

which can be rewritten as $\mathbf{A}(\mathbf{d}_1 + \mathbf{R}\mathbf{d}_2) = 0 \mod q$, proving our claim.

It remains to verify that $\mathbf{e}$ is (i) short and (ii) nonzero. Let us begin with shortness. Since $||\sigma^*|| \leq s\sqrt{2m}$, we have $||\mathbf{d_1}||, ||\mathbf{d_2}|| \leq s\sqrt{2m}$. We then have $||\mathbf{e}|| \leq ||\mathbf{d}_1|| + ||\mathbf{d}_2|| \cdot ||\mathbf{R}||_2$. By our parameter choices, and using Lem. 14, this latter term is again at most $O(4^d m^{3/2}) s\sqrt{2m}$. By our choice of parameters, this is less that $\beta \geq O(4^d \cdot m^{3/2}) \cdot s\sqrt{m}$, and so $\mathbf{e}$ is indeed a valid solution.

Next let us show that $\mathbf{e}$ is nonzero with overwhelming probability. Note that by assumption, σ^* is nonzero so at least one of $\mathbf{d}_1$ or $\mathbf{d}_2$ must be so. If $\mathbf{d}_2$ is zero, then we have that $\mathbf{e}$ is directly is nonzero, so let us focus on the case that $\mathbf{d}_2$ is nonzero. Expressing $\mathbf{d}_2$ as $(d_1, \ldots, d_m)^\top$, we must have that at least one of the coordinates of $\mathbf{d}_2$ is nonzero. Let d_j be such a coordinate. Expressing $\mathbf{R}$ as $(\mathbf{r}_1, \ldots, \mathbf{r}_m)$, we have that

$$\mathbf{R} \cdot \mathbf{d}_2 = \mathbf{r}_j d_j + \sum_{i=1, i \neq j}^{m} \mathbf{r}_i d_i.$$

Now we note that for the (fixed) M^* for which $\mathcal{A}$ makes its forgery, $\mathbf{R}$ (and in turn $\mathbf{r}_j$) depends only on the low-norm matrices $\mathbf{R}_{\mathbf{A}'}, \{\mathbf{R}_{\mathbf{B_i}}\}_{i \in [k]}, \mathbf{R}_{\mathbf{C_0}}, \mathbf{R}_{\mathbf{C_1}}$ and k_ε. Now the *only* information about $\mathbf{R}$ (in turn, $\mathbf{r}_j$) $\mathcal{A}$ has is derived from the components of vk, namely, $\mathbf{A}', \{\mathbf{B}_i\}_{i \in [k]}, \mathbf{C}_0, \mathbf{C}_1$. This implies that *any* $\mathbf{r}'_j \in \{-1, 1\}^m$ such that $\mathbf{A}\mathbf{r}_j = \mathbf{A}\mathbf{r}'_j$ is in fact a valid vector in the sense that replacing $\mathbf{r}_j$ with $\mathbf{r}'_j$ is completely indistinguishable to the adversary. By the pigeonhole principle, there are exponentially many such distinct $\mathbf{r}'_j$'s so that $\mathbf{A}\mathbf{r}_j = \mathbf{A}\mathbf{r}'_j$, and for such an admissible $\mathbf{r}'_j$, the probability that $\mathbf{r}'_j \cdot d_j$ hits a fixed value in $\mathbb{Z}_q^m$ is exponentially small. It is straightforward to see that this implies that $\mathbf{e}$ is zero with at most negligible probability (since the chance that $\mathbf{r}'_j \cdot d_j$ equals the exact value needed to cancel out the other terms in $\mathbf{e}$ is negligible). Finally, it is straightforward to verify that H_4 runs in polynomial time, and in turn that the reduction $\mathcal{R}$ is efficient. If $\mathcal{A}$ produces a valid forgery within B_ε with probability $\nu(\lambda)$, $\mathcal{R}$ breaks the $\mathbf{SIS}_{q,\beta,n,m}$ challenge with probability $\nu(\lambda) - \mathsf{negl}(\lambda)$. We conclude that $\mathcal{A}$ wins the blind-unforgeability experiment with at most negligible probability. □

6 Post-Quantum Ring Signatures

6.1 Definition

6.1.1 Classical Ring Signatures

We start by recalling the classical definition of ring signatures [BKM06, BDH+19].

Definition 4 (Ring Signature). *A ring signature scheme* RS *is described by a triple of PPT algorithms* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *such that:*

- $\mathsf{Gen}(1^\lambda, N)$**:** *on input a security parameter* 1^λ *and a super-polynomial*[9] N *(e.g.,* $N = 2^{\log^2 \lambda}$*) specifying the maximum number of members in a ring, output a verification and signing key pair* $(\mathsf{VK}, \mathsf{SK})$*.*

[9] The N has to be super-polynomial to support rings of *arbitrary* polynomial size.

– $\mathsf{Sign}(\mathsf{SK}, \mathsf{R}, m)$**:** *given a secret key* SK*, a message* $m \in \mathcal{M}_\lambda$*, and a list of verification keys (interpreted as a ring)* $\mathsf{R} = (\mathsf{VK}_1, \cdots, \mathsf{VK}_\ell)$ *as input, and outputs a signature* Σ*.*

– $\mathsf{Verify}(\mathsf{R}, m, \Sigma)$**:** *given a ring* $\mathsf{R} = (\mathsf{VK}_1, \ldots, \mathsf{VK}_\ell)$*, message* $m \in \mathcal{M}_\lambda$ *and a signature* Σ *as input, outputs either 0 (rejecting) or 1 (accepting).*

These algorithms satisfy the following requirements:

1. **Completeness:** *for all* $\lambda \in \mathbb{N}$*,* $\ell \leq N$*,* $i^* \in [\ell]$*, and* $m \in \mathcal{M}_\lambda$*, it holds that* $\forall i \in [\ell]$ $(\mathsf{VK}_i, \mathsf{SK}_i) \leftarrow \mathsf{Gen}(1^\lambda, N)$ *and* $\Sigma \leftarrow \mathsf{Sign}(\mathsf{SK}_{i^*}, \mathsf{R}, m)$ *where* $\mathsf{R} = (\mathsf{VK}_1, \ldots, \mathsf{VK}_\ell)$*, we have*

$$\Pr[\mathsf{RS.Verify}(\mathsf{R}, m, \Sigma) = 1] = 1,$$

where the probability is taken over the random coins used by Gen *and* Sign*.*

2. **Anonymity:** *For any* $Q = \mathsf{poly}(\lambda)$ *and any PPT adversary* $\mathcal{A}$*, it holds w.r.t. Expr. 2 that*

$$\mathsf{Adv}^{\lambda,Q}_{\mathrm{ANON}}(\mathcal{A}) := \left| \Pr\left[\mathsf{Exp}^{\lambda,Q}_{\mathrm{ANON}}(\mathcal{A}) = 1\right] - 1/2 \right| \leq \mathsf{negl}(\lambda).$$

Experiment 2: Classical Anonymity $\mathsf{Exp}^{\lambda,Q}_{\mathrm{ANON}}(\mathcal{A})$

1. For each $i \in [Q]$, the challenger generates key pairs $(\mathsf{VK}_i, \mathsf{SK}_i) \leftarrow \mathsf{Gen}(1^\lambda, N; r_i)$. It sends $\{(\mathsf{VK}_i, \mathsf{SK}_i, r_i)\}_{i \in [Q]}$ to $\mathcal{A}$;
2. $\mathcal{A}$ sends a challenge to the challenger of the form $(i_0, i_1, \mathsf{R}, m)$.[a] The challenger checks if $\mathsf{VK}_{i_0} \in \mathsf{R}$ and $\mathsf{VK}_{i_1} \in \mathsf{R}$. If so, it samples a uniform bit b, computes $\Sigma \leftarrow \mathsf{Sign}(\mathsf{SK}_{i_b}, \mathsf{R}, m)$, and sends Σ to $\mathcal{A}$.
3. $\mathcal{A}$ outputs a guess b'. If $b' = b$, the experiment outputs 1, otherwise 0.

[a] We stress that R might contain keys that are not generated by the challenger in the previous step. In particular, it might contain maliciously generated keys.

3. **Unforgeability:** *for any* $Q = \mathsf{poly}(\lambda)$ *and any PPT adversary* $\mathcal{A}$*, it holds w.r.t. Expr. 3 that*

$$\mathsf{Adv}^{\lambda,Q}_{\mathrm{UNF}}(\mathcal{A}) := \Pr\left[\mathsf{Exp}^{\lambda,Q}_{\mathrm{UNF}}(\mathcal{A}) = 1\right] \leq \mathsf{negl}(\lambda).$$

Experiment 3: Classical Unforgeability $\mathsf{Exp}^{\lambda,Q}_{\mathrm{UNF}}(\mathcal{A})$

1. For each $i \in [Q]$, the challenger generates $(\mathsf{VK}_i, \mathsf{SK}_i) \leftarrow \mathsf{Gen}(1^\lambda, N; r_i)$, and stores these key pairs along with their corresponding randomness. It then sets $\mathcal{VK} = \{\mathsf{VK}_1, \ldots, \mathsf{VK}_Q\}$ and initializes a set $\mathcal{C} = \emptyset$.
2. The challenger sends $\mathcal{VK}$ to $\mathcal{A}$.
3. $\mathcal{A}$ can make polynomially-many queries of the following two types:
   – **Corruption query** $(\mathsf{corrupt}, i)$**:** The challenger adds VK_i to the set $\mathcal{C}$ and returns the randomness r_i to $\mathcal{A}$.
   – **Signing query** $(\mathsf{sign}, i, \mathsf{R}, m)$**:** The challenger first checks if $\mathsf{VK}_i \in \mathsf{R}$. If so, it computes $\Sigma \leftarrow \mathsf{Sign}(\mathsf{SK}_i, \mathsf{R}, m)$ and returns Σ to $\mathcal{A}$. It also keeps a list of all such queries made by $\mathcal{A}$.
4. Finally, $\mathcal{A}$ outputs a tuple $(\mathsf{R}^*, m^*, \Sigma^*)$. The challenger checks if:
   i. $\mathsf{R}^* \subseteq \mathcal{VK} \setminus \mathcal{C}$,

ii. $\mathcal{A}$ never made a signing query of the form $(\mathsf{sign}, \cdot, \mathsf{R}^*, m^*)$, **and**

iii. $\mathsf{Verify}(\mathsf{R}^*, m^*, \Sigma^*) = 1$.

If so, the experiment outputs 1; otherwise, 0.

We mention that the unforgeability and anonymity properties defined in Definition 4 correspond respectively to the notions of *unforgeability with insider corruption* and *anonymity with respect to full key exposure* presented in [BKM06].

6.1.2 Defining Post-Quantum Security

We aim to build a classical ring signature that is secure against adversaries making superposition queries to the signing oracle. Formalizing the security requirements in this scenario is non-trivial. An initial step toward this direction has been taken in [CGH+21]. But their definition has certain restrictions (discussed below). In the following, we develop a new definition building on ideas from [CGH+21].

Post-Quantum Anonymity. Recall that in the classical anonymity game (Expr. 2), the adversary's challenge is a quadruple $(i_0, i_1, \mathsf{R}, m)$. To define post-quantum anonymity, a natural attempt is to allow the adversary to send a superposition over components of quadruple, and to let the challenger respond using the following unitary mapping[10]:

$$\sum_{i_0,i_1,\mathsf{R},m,t} \psi_{i_0,i_1,\mathsf{R},m,t} \left|i_0, i_1, \mathsf{R}, m, t\right\rangle \mapsto \sum_{i_0,i_1,\mathsf{R},m,t} \psi_{i_0,i_1,\mathsf{R},m,t} \left|i_0, i_1, \mathsf{R}, m, t \oplus \mathsf{Sign}(\mathsf{SK}_{i_b}, m, \mathsf{R}; r)\right\rangle .$$

However, as observed in [CGH+21], this will lead to an unsatisfiable definition due to an attack from [BZ13b]. Roughly speaking, the adversary could use classical values for R, m, and i_1, but she puts a uniform superposition of all valid identities in the register for i_0. After the challenger's signing operation, observe that if $b = 0$, the last register will contain signatures in superposition (as i_0 is in superposition); if $b = 1$, it will contain a classical signature (as i_1 is classical). These two cases can be efficiently distinguished by means of a Fourier transform on the i_0's register followed by a measurement. Therefore, to obtain an achievable notion, we should not allow superpositions over (i_0, i_1).

Now, $\mathcal{A}$ only has the choice to put superpositions over R and m. The definition in [CGH+21] further forbids $\mathcal{A}$ from putting superpositions over R. But this is only because they fail to prove security if superposition attacks on R is allowed. Indeed, they leave open the problem to construct a scheme that protects against superposition attacks on R. In this work, we solve this problem: our definition allows superposition attacks on both R and m.

Definition 5 (Post-Quantum Anonymity). *Consider a triple of PPT algorithms* $\mathsf{RS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *that satisfies the same syntax as in Def. 4.* RS *achieves post-quantum anonymity if for any* $Q = \mathsf{poly}(\lambda)$ *and any QPT adversary* $\mathcal{A}$*, it holds w.r.t. Expr. 4 that*

$$\mathsf{PQAdv}^{\lambda,Q}_{\mathrm{ANON}}(\mathcal{A}) \coloneqq \left| \Pr\left[\mathsf{PQExp}^{\lambda,Q}_{\mathrm{ANON}}(\mathcal{A}) = 1\right] - 1/2 \right| \leq \mathsf{negl}(\lambda).$$

[10] Of course, the challenger also needs to check if $\mathsf{VK}_{i_0} \in \mathsf{R}$ and $\mathsf{VK}_{i_1} \in \mathsf{R}$. But we can safely ignore this for our current discussion.

Experiment 4: Post-Quantum Anonymity $\mathsf{PQExp}^{\lambda,Q}_{\mathrm{ANON}}(\mathcal{A})$

1. For each $i \in [Q]$, the challenger generates key pairs $(\mathsf{VK}_i, \mathsf{SK}_i) \leftarrow \mathsf{RS.Gen}(1^\lambda, N; r_i)$. The challenger sends $\{(\mathsf{VK}_i, \mathsf{SK}_i, r_i)\}_{i\in[Q]}$ to $\mathcal{A}$;
2. $\mathcal{A}$ sends (i_0, i_1) to the challenger, where both i_0 and i_1 are in $[Q]$;
3. $\mathcal{A}$'s challenge query is allowed to be a superposition of rings *and* messages. The challenger picks a random bit b and a random string r. It signs the message using SK_{i_b} and randomness r, while making sure that VK_{i_0} and VK_{i_1} are indeed in the ring specified by $\mathcal{A}$. Formally, the challenger implements the following mapping:

$$\sum_{\mathsf{R},m,t} \psi_{\mathsf{R},m,t} \left|\mathsf{R}, m, t\right\rangle \mapsto \sum_{\mathsf{R},m,t} \psi_{\mathsf{R},m,t} \left|\mathsf{R}, m, t \oplus f(\mathsf{R}, m)\right\rangle,$$

   where $f(\mathsf{R}, m) := \begin{cases} \mathsf{RS.Sign}(\mathsf{SK}_{i_b}, R, m; r) & \text{if } \mathsf{VK}_{i_0}, \mathsf{VK}_{i_1} \in R \\ \bot & \text{otherwise} \end{cases}$.
4. $\mathcal{A}$ outputs a guess b'. If $b' = b$, the experiment outputs 1, otherwise 0.

Post-Quantum Unforgeability. In the classical unforgeability game (Expr. 3), $\mathcal{A}$ can make both corrupt and sign queries. As discussed in Sec. 2.3, we do not consider quantum corrupt queries, or superposition attacks over the identity in $\mathcal{A}$'s sign queries. We also remark that in the unforgeability game, [CGH+21] does not allow superpositions over the ring. Instead of a definitional issue, this is again only because they are unable to prove the security of their scheme if superposition attacks on the ring is allowed. In contrast, our construction can be proven secure against such attacks; thus, this restriction is removed from our definition.

To define quantum unforgeability, [CGH+21] adapts one-more unforgeability [BZ13b] to the ring setting: they require that, with sq quantum signing queries, the adversary cannot produce $\mathsf{sq}+1$ signatures, where all the rings are subsets of $\mathcal{VK} \setminus \mathcal{C}$. This definition, *when restricted to the classical setting*, seems to be weaker than the standard unforgeability in Def. 4.That is, in the classical setting, any RS satisfying the unforgeability in Def. 4 is also one-more unforgeable; but the reverse direction is unclear (we discuss this in Appx. B). Instead, our definition extends the blind-unforgeability for ordinary signatures (Def. 1) to the ring setting. We present this definition in Def. 6.

Definition 6 (Post-Quantum Blind-Unforgeability). *Consider a triple of PPT algorithms* $\mathsf{RS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *that satisfies the same syntax as in Def. 4. For any security parameter* λ*, let* $\mathcal{R}_\lambda$ *and* $\mathcal{M}_\lambda$ *denote the ring space and message space, respectively.* RS *achieves blind-unforgeability if for any* $Q = \mathsf{poly}(\lambda)$ *and any QPT adversary* $\mathcal{A}$*, it holds w.r.t. Expr. 5 that*

$$\mathsf{PQAdv}^{\lambda,Q}_{\mathrm{BU}}(\mathcal{A}) := \Pr\left[\mathsf{PQExp}^{\lambda,Q}_{\mathrm{BU}}(\mathcal{A}) = 1\right] \leq \mathsf{negl}(\lambda).$$

Experiment 5: Post-Quantum Blind-Unforgeability $\mathsf{PQExp}^{\lambda,Q}_{\mathrm{BU}}(\mathcal{A})$

1. $\mathcal{A}$ sends a constant $0 \leq \varepsilon \leq 1$ to the challenger;

2. For each $i \in [Q]$, the challenger generates $(\mathsf{VK}_i, \mathsf{SK}_i) \leftarrow \mathsf{Gen}(1^\lambda, N; r_i)$, and stores these key pairs along with their corresponding randomness. It then sets $\mathcal{VK} = \{\mathsf{VK}_1, \ldots, \mathsf{VK}_Q\}$ and initializes a set $\mathcal{C} = \emptyset$; The challenger sends $\mathcal{VK}$ to $\mathcal{A}$;
3. The challenger defines a *blindset* $B_\varepsilon^{\mathsf{RS}} \subseteq 2^{\mathcal{R}_\lambda} \times \mathcal{M}_\lambda$: every pair $(\mathsf{R}, m) \in 2^{\mathcal{R}_\lambda} \times \mathcal{M}_\lambda$ is put in $B_\varepsilon^{\mathsf{RS}}$ with probability ε;
4. $\mathcal{A}$ can make polynomially-many queries of the following two types:
   - **Classical corruption query** $(\mathsf{corrupt}, i)$**:** The challenger adds VK_i to the set $\mathcal{C}$ and returns the randomness r_i to $\mathcal{A}$.
   - **Quantum Signing query** $(\mathsf{sign}, i, \sum \psi_{\mathsf{R},m,t} |\mathsf{R}, m, t\rangle)$**:** That is, $\mathcal{A}$ is allowed to query the signing oracle on some classical identity i and superpositions over rings and messages. The challenger samples a random string r and performs:

$$\sum_{\mathsf{R},m,t} \psi_{\mathsf{R},m,t} |\mathsf{R}, m, t\rangle \mapsto \sum_{\mathsf{R},m,t} \psi_{\mathsf{R},m,t} \left|\mathsf{R}, m, t \oplus B_\varepsilon^{\mathsf{RS}} f(\mathsf{R}, m)\right\rangle,$$

   where $B_\varepsilon^{\mathsf{RS}} f(\mathsf{R}, m) := \begin{cases} \bot & \text{if } (\mathsf{R}, m) \in B_\varepsilon^{\mathsf{RS}} \\ f(\mathsf{R}, m) & \text{otherwise} \end{cases}$, and

   $f(\mathsf{R}, m) := \begin{cases} \mathsf{RS.Sign}(\mathsf{SK}_i, m, \mathsf{R}; r) & \text{if } \mathsf{VK}_i \in \mathsf{R} \\ \bot & \text{otherwise} \end{cases}$.
5. Finally, $\mathcal{A}$ outputs $(\mathsf{R}^*, m^*, \Sigma^*)$. The challenger checks if:
   (a) $\mathsf{R}^* \subseteq \mathcal{VK} \setminus \mathcal{C}$,
   (b) $\mathsf{Verify}(\mathsf{R}^*, m^*, \Sigma^*) = 1$, **and**
   (c) $(\mathsf{R}^*, m^*) \in B_\varepsilon^{\mathsf{RS}}$.

   If so, it outputs 1; otherwise, it outputs 0.

In contrast to the "one-more" unforgeability, we show in Lem. 4 that, when restricted to the classical setting, Def. 6 (for ring signatures) is indeed equivalent to the standard existential unforgeability in Def. 4. Its proof is almost identical to that of [AMRS20, Proposition 2].

Lemma 4. *Restricted to (classical) QPT adversaries, a ring signature* RS *scheme is blind-unforgeable (Def. 6) if and only if it satisfies the unforgeability requirement in Def. 4.*

Proof. We show necessity and sufficiency in turn. In the following, by "Expr. 5", we refer to the classical version of Expr. 5, where the signing query is of the form $(\mathsf{sign}, i, \mathsf{R}, m)$ (i.e., (R, m) is classical), and is answered as $B_\varepsilon^{\mathsf{RS}} f(\mathsf{R}, m)$.

Necessity ($\Leftarrow$). Let us first show how blind-unforgeability implies standard unforgeability (for classical settings). Assume we have an adversary $\mathcal{A}_{\text{EUF}}$ that breaks standard unforgeability of RS as per Def. 4, i.e., in Expr. 3 it produces a forgery $(m^*, \mathsf{R}^*, \Sigma^*)$ that is valid with non-negligible probability $\nu(\lambda)$. We show that this is easily converted into an adversary $\mathcal{A}_{\text{BU}}$ that wins Expr. 5 with non-negligible probability as well. $\mathcal{A}_{\text{BU}}$ first sets $\varepsilon(\lambda)$ equal to $1/p(\lambda)$, where $p(\lambda)$ denotes the (polynomial) running time of $\mathcal{A}_{\text{EUF}}$ (the reasoning behind this choice will become clear very soon).

It then simply forwards all the queries from $\mathcal{A}_{\text{EUF}}$ to the blind-unforgeability challenger and the responses back to $\mathcal{A}_{\text{EUF}}$. It also outputs whatever eventual forgery $\mathcal{A}_{\text{EUF}}$ does.

Let us consider the success probability of $\mathcal{A}_{\text{BU}}$. To start with, note that $\mathcal{A}_{\text{EUF}}$ makes at most $p(\lambda)$ many queries of the signing oracle. In each such query, we know that the (R, m) pair is in the blind set independently with probability ε. Thus it is not in the blind set with probability $1-\varepsilon$, and if so the query is answered properly. In turn, the probability that all the queries made are answered properly is then at least $(1-\varepsilon)^{p(\lambda)} \approx 1/e$ (this uses independence and $\varepsilon = 1/p$), and so the probability that the forgery $(\mathsf{R}^*, m^*, \Sigma^*)$ is valid is then at least $(1-\varepsilon)^{p(\lambda)} \cdot \nu(\lambda)$. Finally, the forgery, even if successful, might lie in the blind set with probability ε. So, the total probability that $\mathcal{A}_{\text{EUF}}$ outputs a valid forgery *for the blind unforgeability game* is $(1-\varepsilon)^{p(\lambda)+1} \cdot \nu(\lambda) \approx (1-\varepsilon) \cdot \nu \cdot 1/e$, which is non-negligible since ν is non-negligible by assumption. Thus if $\mathcal{A}_{\text{EUF}}$ violates standard ring signature unforgeability according to Def. 4, then $\mathcal{A}_{\text{BU}}$ violates blind unforgeability for ring signatures according to Def. 6, as claimed.

Sufficiency ($\Rightarrow$). Let us now turn to the other direction of the equivalence. Assume now that there exists an adversary $\mathcal{A}_{\text{BU}}$ that can break blind unforgeability of RS, i.e., win Expr. 5 with non-negligible probability $\nu(\lambda)$. We show an adversary $\mathcal{A}_{\text{EUF}}$ that can win Expr. 3 with non-negligible probability. $\mathcal{A}_{\text{EUF}}$ simply simulates Expr. 5 for $\mathcal{A}_{\text{BU}}$ by answering oracle queries according to a locally-simulated version of $B_\varepsilon^{\mathsf{RS}} f(\mathsf{R}, m)$. Concretely, the adversary $\mathcal{A}_{\text{EUF}}$ proceeds by drawing a subset $B_\varepsilon^{\mathsf{RS}}$ in the same manner as the challenger in Expr. 5 and answering queries made by $\mathcal{A}_{\text{BU}}$ according to $B_\varepsilon^{\mathsf{RS}} f(\mathsf{R}, m)$. Two remarks are in order:

1. when $(\mathsf{R}, m) \in B_\lambda^{\mathsf{RS}}$, no signature needs to be done. That is, this query can be answered by $\mathcal{A}_{\text{EUF}}$ without calling its own signing oracle;
2. $\mathcal{A}_{\text{EUF}}$ can construct the set $B_\varepsilon^{\mathsf{RS}}$ by "lazy sampling", i.e., when a particular query $(\mathsf{sign}, i, \mathsf{R}, m)$ is made by $\mathcal{A}_{\text{BU}}$, whether $(\mathsf{R}, m) \in B_\varepsilon^{\mathsf{RS}}$ and "remembering" this information in case the query is asked again.

By assumption, $\mathcal{A}_{\text{BU}}$ produces a valid forgery. And it follows from Item 1 that this forgery must be on a point which was not queried by $\mathcal{A}_{\text{EUF}}$, thus, also serving as a valid forgery for $\mathcal{A}_{\text{EUF}}$'s game. □

To conclude, we present the complete definition for quantum ring signatures.

Definition 7 (Post-Quantum Secure Ring Signatures). *A post-quantum secure ring signature scheme* RS *is described by a triple of PPT algorithms* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *that share the same syntax as in Def. 4. Moreover, these algorithms also satisfy the* completeness *requirement as per Def. 4, the* post-quantum anonymity *requirement as per Def. 5, and the* post-quantum blind-unforgeability *requirement as per Def. 6.*

6.2 Building Blocks

6.2.1 Lossy PKEs with Special Properties

We need the following lossy PKE.

Definition 8 (Special Lossy PKE). *For any security parameter* $\lambda \in \mathbb{N}$*, let* $\mathcal{M}_\lambda$ *denote the message space. A special lossy public-key encryption scheme* LE *consists of the following PPT algorithms:*

- $\mathsf{MSKGen}(1^\lambda, Q)$, *on input a number* $Q \in \mathbb{N}$, *outputs* $\big(\{\mathsf{pk}_i\}_{i\in[Q]}, \mathsf{msk}\big)$. *We call* pk_i*'s the* injective public keys, *and* msk *the* master secret key.
- $\mathsf{MSKExt}(\mathsf{msk}, \mathsf{pk})$, *on input a master secret key* msk *and an injective public key* pk, *outputs a secret key* sk.
- $\mathsf{KSam}^{\mathsf{ls}}(1^\lambda)$ *outputs key* $\mathsf{pk}_{\mathsf{ls}}$, *which we call* lossy public key.
- $\mathsf{Valid}(\mathsf{pk}, \mathsf{sk})$, *on input a public* pk *and a secret key* sk, *outputs either* 1 *(accepting) or* 0 *(rejecting).*
- $\mathsf{RndExt}(\mathsf{pk})$ *outputs a* r *which we call* extracted randomness.
- $\mathsf{Enc}(\mathsf{pk}, m)$, *on input a public key* pk, *and a message* $m \in \mathcal{M}_\lambda$, *outputs* ct.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$, *on input a secret key* sk *and a ciphertext* ct, *outputs* m.

These algorithms satisfy the following properties:

1. **Completeness.** *For any* $\lambda \in \mathbb{N}$, *any* $(\mathsf{pk}, \mathsf{sk})$ *s.t.* $\mathsf{Valid}(\mathsf{pk}, \mathsf{sk}) = 1$, *and any* $m \in \mathcal{M}_\lambda$, *it holds that*
$$\Pr\big[\mathsf{Dec}\big(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)\big) = m\big] = 1.$$
2. **Lossiness of lossy keys.** *For any* $\mathsf{pk}_{\mathsf{ls}}$ *in the range of* $\mathsf{KSam}^{\mathsf{ls}}(1^\lambda)$ *and any* $m_0, m_1 \in \mathcal{M}_\lambda$, *it holds that*
$$\big\{\mathsf{Enc}(\mathsf{pk}_{\mathsf{ls}}, m_0)\big\}_{\lambda\in\mathbb{N}} \overset{s}{\approx} \big\{\mathsf{Enc}(\mathsf{pk}_{\mathsf{ls}}, m_1)\big\}_{\lambda\in\mathbb{N}}.$$
3. **Completeness of Master Secret Keys:** *for any* $Q = \mathsf{poly}(\lambda)$, *it holds that*
$$\Pr\left[\big(\{\mathsf{pk}_i\}_{i\in[Q]}, \mathsf{msk}\big) \leftarrow \mathsf{MSKGen}(1^\lambda, Q) : \begin{array}{l}\forall i \in [Q], \mathsf{Valid}\big(\mathsf{pk}_i, \mathsf{sk}_i\big) = 1, \\ \textit{where } \mathsf{sk}_i := \mathsf{MSKExt}\big(\mathsf{msk}, \mathsf{pk}_i\big)\end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$
4. **IND of $\mathsf{MSKGen}/\mathsf{KSam}^{\mathsf{ls}}$ mode:** *For any* $Q = \mathsf{poly}(\lambda)$, *the following two distributions are computationally indistinguishable:*
   - $\forall i \in [Q]$, *sample* $\mathsf{pk}_i \leftarrow \mathsf{KSam}^{\mathsf{ls}}(1^\lambda; r_i)$, *then output* $\{\mathsf{pk}_i, r_i\}_{i\in[Q]}$;
   - *Sample* $\big(\{\mathsf{pk}_i\}_{i\in[Q]}, \mathsf{msk}\big) \leftarrow \mathsf{MSKGen}(1^\lambda, Q)$ *and output* $\big\{\mathsf{pk}_i, \mathsf{RndExt}(\mathsf{pk}_i)\big\}_{i\in[Q]}$.
5. **Almost-Unique Secret Key:** *For any* $Q = \mathsf{poly}(\lambda)$, *it holds that*
$$\Pr\left[\big(\{\mathsf{pk}_i\}_{i\in[Q]}, \mathsf{msk}\big) \leftarrow \mathsf{MSKGen}(1^\lambda, Q) : \begin{array}{l}\textit{There exist } i \in [Q] \textit{ and } \mathsf{sk}'_i \textit{ such that} \\ \mathsf{sk}'_i \neq \mathsf{MSKExt}(\mathsf{msk}, \mathsf{pk}_i) \ \wedge \ \mathsf{Valid}(\mathsf{pk}_i, \mathsf{sk}'_i) = 1\end{array}\right] = \mathsf{negl}(\lambda).$$

We propose an instantiation of such a lossy PKE using dual mode LWE commitments [GVW15]. In lossy (statistically hiding) mode, the public key consists of a uniformly sampled matrix $\mathbf{A}$ and a message m is encrypted by computing $\mathbf{AR} + m\mathbf{G}$, where $\mathbf{R}$ is a low-norm matrix and $\mathbf{G}$ is the gadget matrix. Note that the random coins used to sample $\mathbf{A}$ simply consists of the matrix $\mathbf{A}$ itself. Furthermore, we can switch $\mathbf{A}$ to be an LWE-matrix (using some secret vector $\mathbf{s}$) to make the encryption scheme injective. Such a modification is computationally indistinguishable by an invocation of the LWE assumption. Note that this is true also in the presence of the output of $\mathsf{RndExt}(\mathbf{A})$, since the algorithm simply returns $\mathbf{A}$. Furthermore, by setting the dimensions appropriately, the secret $\mathbf{s}$ is uniquely determined by $\mathbf{A}$ with overwhelming probability. Finally, we note that we can define a master secret key for all keys in injective mode using a simple trick: sample a PRF key k and sample the i-th key pair using $\mathsf{PRF}(k, i)$ as the random coins. It is not hard to see that the distribution of public/secret keys is computationally indistinguishable by the pseudorandomness of PRF. Furthermore, given k one can extract the i-th secret key simply by recomputing it.

6.2.2 ZAPs for Super-Complement Languages

As mentioned in Sec. 2.3, [CGH+21] uses a ZAP (for $\mathsf{NP} \cap \mathsf{coNP}$) to prove a statement that the (ring) signature contains a ciphertext of a valid signature w.r.t. the building-block signature scheme. Let us denote this language as L. In the security proof, they need to argue that the adversary cannot prove a false statement $x^* \notin L$. However, this L is not necessarily in coNP; thus, there may not exist a non-witness $\widetilde{w}$ for the fact that $x^* \notin L$. Therefore, it is unclear how to use a ZAP for $\mathsf{NP} \cap \mathsf{coNP}$ here. To address this issue, the authors of [CGH+21] propose the notion of *super-complement languages*. This notion considers a pair of NP languages $(L, \widetilde{L})$ such that $(x \in \widetilde{L}) \Rightarrow (x \notin L)$. Their ZAP achieves soundness such that the cheating prover cannot prove $x \in L$ (except with negligible probability) once there exists a "non-witness" $\widetilde{w}$ s.t. $(x, \widetilde{w}) \in R_{\widetilde{L}}$. The $\widetilde{L}$ is set to a special language that captures some *necessary conditions* for any forged signatures to be valid. Thus, a winning adversary will break the soundness of the ZAP, leading to a contradiction.

In the following, we present the original definition of super-complement languages. But we will only need a special case of it (see Rmk. 3).

Definition 9 (Super-Complement [CGH+21]). *Let $(L, \widetilde{L})$ be two NP languages where the elements of $\widetilde{L}$ are represented as pairs of bit strings. We say $\widetilde{L}$ is a* super-complement *of L, if $\widetilde{L} \subseteq (\{0,1\}^* \backslash L) \times \{0,1\}^*$. I.e., $\widetilde{L}$ is a super complement of L if for any $x = (x_1, x_2)$, $x \in \widetilde{L} \Rightarrow x_1 \notin L$.*

Notice that, while the complement of L might not be in NP, it must hold that $\widetilde{L} \in \mathsf{NP}$. The language $\widetilde{L}$ is used to define the soundness property. Namely, producing a proof for a statement $x = (x_1, x_2) \in \widetilde{L}$, should be hard. We also use the fact that $\widetilde{L} \in \mathsf{NP}$ to mildly strengthen the soundness property. In more detail, instead of having selective soundness where the statement $x \in \widetilde{L}$ is fixed in advance, we now fix a non-witness $\widetilde{w}$ and let the statement x be adaptively chosen by the malicious prover from all statements which have $\widetilde{w}$ as a witness to their membership in $\widetilde{L}$.

Remark 3. Our application only needs a special case of the general form given in Def. 9—we will only focus on $\widetilde{L}$ where the x_2 part is an empty string. Formally, we consider the special case where $\widetilde{L} \subseteq \{0,1\}^* \setminus L$ (i.e., $x \in \widetilde{L} \Rightarrow x \notin L$).

We now define ZAPs for super-complement languages. We remark that the original definition (and construction) in [CGH+21] captures the general $(L, \widetilde{L})$ pairs defined in Def. 9. Since we only need the special case in Rmk. 3, we will define the ZAP only for this case.

Definition 10 (ZAPs for Special Super-Complement Languages). *Let $L, \widetilde{L} \in \mathsf{NP}$ be the special super-complement language in Rmk. 3. Let R and $\widetilde{R}$ denote the NP relations corresponding to L and $\widetilde{L}$ respectively. Let $\{C_{n,\ell}\}_{n,\ell}$ and $\{\widetilde{C}_{n,\widetilde{\ell}}\}_{n,\widetilde{\ell}}$ be the NP verification circuits for L and $\widetilde{L}$ respectively. Let $\widetilde{d} = \widetilde{d}(n, \widetilde{\ell})$ be the depth of $\widetilde{C}_{n,\widetilde{\ell}}$. A* ZAP *for $(L, \widetilde{L})$ is a tuple of PPT algorithms $(\mathsf{V}, \mathsf{P}, \mathsf{Verify})$ having the following interfaces (where $1^n, 1^\lambda$ are implicit inputs to P, Verify):*

- $\mathsf{V}(1^\lambda, 1^n, 1^{\widetilde{\ell}}, 1^{\widetilde{D}})$**:** *On input a security parameter λ, statement length n for L, witness length $\widetilde{\ell}$ for $\widetilde{L}$, and NP verifier circuit depth upper-bound $\widetilde{D}$ for $\widetilde{L}$, output a first message ρ.*
- $\mathsf{P}(\rho, x, w)$**:** *On input a string ρ, a statement $x \in \{0,1\}^n$, and a witness w such that $(x, w) \in R$, output a proof π.*
- $\mathsf{Verify}(\rho, x, \pi)$**:** *On input a string ρ, a statement x, and a proof π, output either 1 (accepting) or 0 (rejecting).*

The following requirements are satisfied:

1. **Completeness:** *For every $x \in L$, every $\widetilde{\ell} \in \mathbb{N}$, every $\widetilde{D} \geq \widetilde{d}(|x|, \widetilde{\ell})$, and every $\lambda \in \mathbb{N}$, it holds that*
$$\Pr\Big[\rho \leftarrow \mathsf{V}(1^\lambda, 1^{|x|}, 1^{\widetilde{\ell}}, 1^{\widetilde{D}}); \pi \leftarrow \mathsf{P}(\rho, x, w) : \mathsf{Verify}\big(\rho, x, \pi\big) = 1\Big] = 1.$$
2. **Public coin:** $\mathsf{V}(1^\lambda, 1^n, 1^{\widetilde{\ell}}, 1^{\widetilde{D}})$ *simply outputs a uniformly random string.*
3. **Selective non-witness adaptive-statement soundness:** *For any non-uniform QPT machine P^*_λ, any $n, \widetilde{D} \in \mathbb{N}$, and any non-witness $\widetilde{w} \in \{0,1\}^*$,*
$$\Pr\left[\begin{matrix}\rho \leftarrow \mathsf{V}(1^\lambda, 1^n, 1^{|\widetilde{w}|}, 1^{\widetilde{D}}); \\ (x, \pi^*) \leftarrow P^*_\lambda(\rho)\end{matrix} : \begin{matrix}\mathsf{Verify}(\rho, x, \pi^*) = 1 \ \wedge \\ \widetilde{D} \geq \widetilde{d}(|x|, |\widetilde{w}|) \ \wedge \ (x, \widetilde{w}) \in \widetilde{R}\end{matrix}\right] \leq \mathsf{negl}(\lambda).$$
4. **Statistical witness indistinguishability:** *For every (possibly unbounded) "cheating" verifier $V^* = (V^*_1, V^*_2)$ and every $n, \widetilde{\ell}, \widetilde{D} \in \mathbb{N}$, the probabilities*
$$\Pr\big[V^*_2(\rho, x, \pi, \zeta) = 1 \ \wedge \ (x, w) \in \mathcal{R} \ \wedge \ (x, w') \in \mathcal{R}\big]$$
in the following two experiments differ only by $\mathsf{negl}(\lambda)$:
   - Experiment 1: $(\rho, x, w, w', \zeta) \leftarrow V^*_1(1^\lambda, 1^n, 1^{\widetilde{\ell}}, 1^{\widetilde{D}})$, $\pi \leftarrow \mathsf{P}(\rho, x, w)$;
   - Experiment 2: $(\rho, x, w, w', \zeta) \leftarrow V^*_1(1^\lambda, 1^n, 1^{\widetilde{\ell}}, 1^{\widetilde{D}})$, $\pi \leftarrow \mathsf{P}(\rho, x, w')$.

Lemma 5 ([CGH+21]). *Assuming QLWE, there exist ZAPs as per Def. 10 for any super-complement language as per Def. 9.*

6.3 Construction

Our construction RS, shown in Constr. 3, relies on the following building blocks:

1. Pair-wise independent hash functions;
2. A blind-unforgeable signature scheme Sig satisfying Def. 1;
3. A lossy PKE scheme LE satisfying Def. 8;
4. A ZAP for special super-complement languages ZAP satisfying Def. 10.

We remark that the $\mathsf{RS.Sign}$ algorithm runs ZAP on a special super-complement language $(L, \widetilde{L})$, whose definition will appear after the construction in Sec. 6.3.1. This arrangement is because we believe that the language $(L, \widetilde{L})$ will become easier to understand once the reader has slight familiarity with Constr. 3.

Construction 3: Post-Quantum Ring Signatures

Let $\widetilde{D} = \widetilde{D}(\lambda, N)$ be the maximum depth of the NP verifier circuit for language $\widetilde{L}$ restricted to statements where the ring has at most N members, and the security parameter for Sig and LE is λ. Let $n = n(\lambda, \log N)$ denote the maximum size of the statements of language L where the ring has at most N members and the security parameter is λ. Recall that for security parameter λ, secret keys in LE have size $\widetilde{\ell} = \ell_{\mathsf{sk}}(\lambda)$. We now describe our ring signature construction:

Key Generation Algorithm $\mathsf{Gen}(1^\lambda, N)$*:*

- sample signing/verification key pair: $(vk, sk) \leftarrow \mathsf{Sig.Gen}(1^\lambda)$;
- sample obliviously an injective public key of LE: $pk \leftarrow \mathsf{LE.KSam}^{\mathsf{ls}}(1^\lambda)$;
- compute the first message $\rho \leftarrow \mathsf{ZAP.V}(1^\lambda, 1^n, 1^{\widetilde{\ell}}, 1^{\widetilde{D}})$ for ZAP;
- output the verification key $\mathsf{VK} := (vk, pk, \rho)$ and signing key $\mathsf{SK} := (sk, vk, pk, \rho)$.

Signing Algorithm $\mathsf{Sign}(\mathsf{SK}, \mathsf{R}, m)$*:*

- parse $\mathsf{R} = (\mathsf{VK}_1, \ldots, \mathsf{VK}_\ell)$; and parse $\mathsf{SK} = (sk, vk, pk, \rho)$;
- compute $\sigma \leftarrow \mathsf{Sig.Sign}(sk, \mathsf{R} \| m)$;
- let $\mathsf{VK} := \mathsf{VK}_i \in R$ be the verification key corresponding to SK;
- sample two pairwise-independent hash functions PI_1 and PI_2, and compute

$$r_{c_1} = \mathsf{PI}_1(\mathsf{R} \| m), \quad r_{c_2} = \mathsf{PI}_2(\mathsf{R} \| m).$$

- compute $c_1 \leftarrow \mathsf{LE.Enc}(pk, (\sigma, vk); r_{c_1})$ and $c_2 \leftarrow \mathsf{LE.Enc}(pk, 0^{|\sigma|+|vk|}; r_{c_2})$;
- let $\mathsf{VK}_1 = (vk_1, pk_1, \rho_1)$ denote the lexicographically smallest member of R (as a string; note that this is necessarily unique);
- fix statement $x = (\mathsf{R}, m, c_1, c_2)$ and witness $w = (vk, pk, \sigma, r_{c_1})$. We remark that this statement and witness correspond to a super-complement language $(L, \widetilde{L})$ that will be defined in Sec. 6.3.1. Looking ahead, x with witness w is a statement in the L defined in Eq. (1); x constitutes a statement that is *not* in the $\widetilde{L}$ defined in Eq. (4).
- sample another pairwise-independent hash function PI_3 and compute $r_\pi = \mathsf{PI}_3(\mathsf{R} \| m)$;
- compute $\pi \leftarrow \mathsf{ZAP.P}(\rho_1, x, w; r_\pi)$;
- output $\Sigma = (c_1, c_2, \pi)$.

Verification Algorithm $\mathsf{Verify}(\mathsf{R}, m, \Sigma)$*:*

- identify the lexicographically smallest verification key VK_1 in R;
- fix $x = (\mathsf{R}, m, c_1, c_2)$; read ρ_1 from VK_1;
- compute and output $\mathsf{ZAP.Verify}(\rho_1, x, \pi)$.

6.3.1 The Super-Complement Language Proven by the ZAP

We now define the super-complement language $(L, \widetilde{L})$ used in Constr. 3. This deviates from the $(L, \widetilde{L})$ defined in [CGH+21, Section 5], to accommodate Constr. 3.

For a statement of the form $x_1 = (\mathsf{R}, m, c)$ and witness $w = \big(\mathsf{VK} = (vk, pk, \rho), \sigma, r_c\big)$, define relations R_1, R_2, and R_3 as follows:

$$\begin{aligned}(x_1, w) \in R_1 &\;\Leftrightarrow\; \mathsf{VK} \in \mathsf{R},\\ (x_1, w) \in R_2 &\;\Leftrightarrow\; \mathsf{LE.Enc}\big(pk, (\sigma, vk); r_c\big) = c,\\ (x_1, w) \in R_3 &\;\Leftrightarrow\; \mathsf{Sig.Verify}(vk, \mathsf{R} \| m, \sigma) = 1.\end{aligned}$$

Next, define the relation R' as $R' \coloneqq R_1 \cap R_2 \cap R_3$. Let L' be the language corresponding to R'. Define language L as

$$L \coloneqq \big\{x = (\mathsf{R}, m, c_1, c_2) \;\big|\; (\mathsf{R}, m, c_1) \in L' \;\vee\; (\mathsf{R}, m, c_2) \in L'\big\}. \tag{1}$$

Now, we define another language $\widetilde{L}$ and prove that it is a super-complement of L in Claim 1. Let $x_1 = (\mathsf{R}, m, c)$ as above, but let $\widetilde{w} \coloneqq msk$. Define the following relations:

$$(x_1, \widetilde{w}) \in R_4 \;\Leftrightarrow\; \forall j \in [\ell] : \mathsf{LE.Valid}\big(pk_j, \mathsf{LE.MSKExt}(msk, pk_j)\big) = 1 \tag{2}$$

$$(x_1, \widetilde{w}) \in R_5 \;\Leftrightarrow\; \begin{cases} \exists \mathsf{VK} \in \mathsf{R} : \mathsf{VK} = (vk, pk, \rho) \text{ such that:} \\ \mathsf{LE.Valid}\big(pk, \mathsf{LE.MSKExt}(msk, pk)\big) = 1 \;\wedge \\ \mathsf{LE.Dec}\big(\mathsf{LE.MSKExt}(msk, pk), c\big) = (\sigma, vk) \;\wedge \\ \mathsf{Sig.Verify}(vk, \mathsf{R} \| m, \sigma) = 1 \end{cases} \tag{3}$$

where, for each $j \in [\ell]$, $\mathsf{VK}_j = (vk_j, pk_j, \rho_j)$ is the j-th member in R. Let L_4 and L_5 be the languages corresponding to R_4 and R_5, respectively. Define further the relation $\widehat{R}$ according to $\widehat{R} \coloneqq R_4 \setminus R_5$, and let $\widehat{L}$ be the corresponding language. Define $\widetilde{L}$ as follows:

$$\widetilde{L} \coloneqq \big\{x = (\mathsf{R}, m, c_1, c_2) \;\big|\; (\mathsf{R}, m, c_1) \in \widehat{L} \;\wedge\; (\mathsf{R}, m, c_2) \in \widehat{L}\big\}. \tag{4}$$

Following a similar proof as for [CGH+21, Lemma 5.1], we can show that $\widetilde{L}$ is indeed a super-complement of L.

Claim 1. *If* LE *satisfies the completeness defined in Item 1 of Def. 8, then the language* $\widetilde{L}$ *defined in Eq. (4) is a super-complement (as per Def. 9) of the language* L *defined in Eq. (1).*

Proof. To prove this claim, we need to show that for any statement x of the following form

$$x = (\mathsf{R}, m, c_1, c_2), \tag{5}$$

it holds that $x \in \widetilde{L} \Rightarrow x \notin L$ (see Rmk. 3). In the following, we finish the proof by showing the contrapositive: $x \in L \Rightarrow x \notin \widetilde{L}$.

For any x as in Eq. (5), we define

$$x_1 \coloneqq (\mathsf{R}, m, c_1) \text{ and } x_2 \coloneqq (\mathsf{R}, m, c_2).$$

To prove "$x \in L \Rightarrow x \notin \widetilde{L}$", it suffices to show that the following Expressions (6) and (7) hold for every $w = (\mathsf{VK} = (vk, pk, \rho), \sigma, r_c)$ and every $\widetilde{w} = msk$:

$$(x_1, w) \in R' \wedge \big(x_1, \widetilde{w}\big) \in R_4 \;\Rightarrow\; \big(x_1, \widetilde{w}\big) \in R_5 \tag{6}$$

$$(x_2, w) \in R' \wedge \big(x_2, \widetilde{w}\big) \in R_4 \;\Rightarrow\; \big(x_2, \widetilde{w}\big) \in R_5. \tag{7}$$

We first prove Expression (6). If $(x_1, \widetilde{w}) \in R_4$, then for all $\mathsf{VK} = (vk, pk, \rho) \in \mathsf{R}$, we know that $\mathsf{LE.MSKExt}(msk, pk)$ is a valid secret key for pk. This means that:

Fact: any ciphertext w.r.t. any pk (contained in any VK) in R can be decrypted correctly by $\mathsf{LE.MSKExt}(msk, pk)$.

Also, observe that $(x_1, \widetilde{w}) \in R'$ means $(x_1, w) \in R_1 \cap R_2 \cap R_3$, which says c_1 is a valid ciphertext of a signature for $\mathsf{R}\|m$, encrypted by some pk in the ring R. Then, by the above **Fact**, we must have $(x_1, \widetilde{w}) \in R_5$.

Expression (7) can be proven similarly. This finish the proof of Claim 1. □

6.4 Proof of Security

We now prove that Constr. 3 is a post-quantum secure post-quantum ring signature satisfying Def. 7. Its completeness follows straightforwardly from the completeness of ZAP and Sig. We next prove post-quantum anonymity and blind-unforgeability in Sec. 6.4.1 and Sec. 6.4.2, respectively.

6.4.1 Proving Post-Quantum Anonymity

In this section, we prove the following Lem. 6, which establishes post-quantum anonymity for Constr. 3.

Lemma 6. *Assume* LE *satisfies the lossiness (Item 2) described in Def. 8 and* ZAP *is statistically witness indistinguishable. Then, Constr. 3 satisfies the post-quantum anonymity described in Def. 5.*

Let $\mathcal{A}$ be a QPT adversary participating in Expr. 4. Recall that the classical identities specified by $\mathcal{A}$ is (i_0, i_1) and the quantum query is $\sum_{\mathsf{R},m,t} \psi_{\mathsf{R},m,t} |\mathsf{R}, m, t\rangle$. We will show a sequence of hybrids where the challenger switches from signing using i_0 to signing using i_1. It is easy to see that the scheme is post-quantum anonymous if $\mathcal{A}$ cannot tell the difference between each pair of adjacent hybrids.

Hybrid H_0**:** This hybrid simply runs the anonymity game with $b = 0$. That is, $\mathcal{A}$'s query is answered as follows:

$$\sum_{\mathsf{R},m,t} \psi_{\mathsf{R},m,t} |\mathsf{R}, m, t\rangle \mapsto \sum_{\mathsf{R},m,t} \psi_{\mathsf{R},m,t} |\mathsf{R}, m, t \oplus f(\mathsf{R}, m)\rangle ,$$

where $f(\mathsf{R}, m) := \begin{cases} \mathsf{RS.Sign}(\mathsf{SK}_{i_0}, \mathsf{R}\|m; r) & \text{if } \mathsf{VK}_{i_0}, \mathsf{VK}_{i_1} \in R \\ \bot & \text{otherwise} \end{cases}$. We remark that $f(\mathsf{R}, m)$ is performed quantumly for each (R, m) pair in the superposition. We say that $\mathcal{A}$ *wins* if it outputs $b' = b$ $(= 0)$.

It is worth noting that although $\mathsf{RS.Sign}$ is a randomized algorithm, it uses only a single random tape r for all the (R, m) pairs in the superposition (See Rmk. 1). In Constr. 3, this means that the pair-wise independent hash functions $\mathsf{PI}_1, \mathsf{PI}_2, \mathsf{PI}_3$) are sampled only once (i.e., they remain the same for all the (R, m) pairs in the superposition).

Hybrid H_1**:** In this hybrid, for each signing query from $\mathcal{A}$, instead of sampling a pair-wise indepedent function $\mathsf{PI}_2(\cdot)$ and compute $r_{c_2} = \mathsf{PI}_2(\mathsf{R}\|m)$, we compute $r_{c_2} = P_2(\mathsf{R}\|m)$, where $P_2(\cdot)$ is a *random* function. In effect, r_{c_2} is now randomly sampled for each (R, m) pairs.

$H_0 \overset{i.d.}{\equiv} H_1$*:* This follows from Lem. 1.

Hybrid H_2: Here we switch c_2 from an encryption of a zero string to $c_2 \leftarrow \mathsf{LE.Enc}(pk_{i_1}, (\sigma', vk_{i_1}); r_{c_2})$, where $\sigma' \leftarrow \mathsf{Sig.Sign}(sk_{i_1}, \mathsf{R}\|m)$. In this hybrid, it is worth noting that the previous "dummy ciphertext" c_2 becomes a valid one, i.e., it encrypts a valid signature for $\mathsf{R}\|m$ using identity i_1.

$H_1 \overset{s}{\approx} H_2$: In both H_1 and H_2, r_{c_2} is sampled (effectively) uniformly at random for each (R, m) pair in the superposition in each signing query. Consider an oracle $\mathcal{O}$ that takes (R, m) as input and returns (c_1, c_2, π) just as in H_1, and an analogous oracle $\mathcal{O}'$ that takes the same input and returns (c_1, c_2, π) computed just as in H_2. Note that the only difference between the outputs of $\mathcal{O}$ and $\mathcal{O}'$ is in c_2, which encrypts $0^{|\sigma|+|vk|}$ in H_1 and (σ', vk_{i_1}) in H_2. Recall that pk_{i_1} is produced using $\mathsf{LE.KSam}^{\mathsf{ls}}$ and therefore, by lossiness (Item 2), we have that the distributions of c_2 in H_1 and H_2 are statistically indistinguishable, implying that the outputs of $\mathcal{O}$ and $\mathcal{O}'$ are statistically close for every input (R, m), say less than distance Δ (which is negligible in λ). Then, by Lem. 2, the probability that $\mathcal{A}$ distinguishes these two oracles even with $q = \mathsf{poly}(\lambda)$ quantum queries is at most $\sqrt{8C_0 q^3 \Delta}$, which is negligible since Δ is negligible. Similarity of these hybrids is immediate.

Hybrid H_3: In this hybrid, we switch back to using the pairwise independent hash function PI_2 to compute r_{c_2}, instead of using a truly random function. Effectively we are undoing the change made in H_1.

$H_2 \overset{i.d.}{\equiv} H_3$: This again follows from Lem. 1.

Hybrid H_4: Here, we compute r_π as the output of a *random* function $r_\pi = P_3(\mathsf{R}\|m)$, instead of being computed using PI_3 as before. In effect, r_π is now uniformly random.

$H_3 \overset{i.d.}{\equiv} H_4$: This again follows from Lem. 1.

Hybrid H_5: As mentioned in H_2, the "block" (R, m, c_2) is valid. Recall that in previous hybrids, ZAP uses the witness w corresponding to the block (R, m, c_1). In this hybrid, we switch the witness used by ZAP from $w = (vk_{i_0}, pk_{i_0}, \sigma, r_{c_1})$ to $w' = (vk_{i_1}, pk_{i_1}, \sigma', r_{c_2})$, i.e., the witness corresponding to the (R, m, c_2) block.

$H_4 \overset{s}{\approx} H_5$: In both H_4 and H_5, r_π is sampled (effectively) uniformly at random for each (R, m) pairs in the superposition for each query. Consider an oracle $\mathcal{O}$ that takes (R, m) as input and returns (c_1, c_2, π) just as in H_4, and an analogous oracle $\mathcal{O}'$ that takes the same input and returns (c_1, c_2, π) computed just as in H_5. Note that the only difference between the outputs of $\mathcal{O}$ and $\mathcal{O}'$ is in π, which is generated using w in H_4 and using w' in H_5. Since both w and w' are valid witnesses, by the *statistical* witness indistinguishability of ZAP, we have that the distributions of π in H_4 and H_5 are statistically indistinguishable for every (R, m) pair (aka the input to the $\mathcal{O}$ or $\mathcal{O}'$). In other words, the outputs of $\mathcal{O}$ and $\mathcal{O}'$ are statistically close for every input (R, m), say less than distance Δ (which is negligible in λ). Then, the statistical indistinguishability follows from Lem. 2.

Hybrid H_6: In this hybrid, we switch back to using the pairwise independent hash function PI_3 to compute r_π, instead of using a truly random function. Effectively we are undoing the change made in H_4.

$H_5 \overset{i.d.}{\equiv} H_6$: This again follows from Lem. 1.

Hybrid H_7**:** In this hybrid, instead of sampling $r_{c_1} = \mathsf{PI}_1(\mathsf{R}\|m)$, we instead compute r_{c_1} as the output of a *random* function $r_{c_1} = P_1(\mathsf{R}\|m)$. In effect, r_{c_1} is now randomly sampled.

$H_6 \overset{i.d.}{=\!=} H_7$*:* This again follows from Lem. 1.

Hybrid H_8**:** In this hybrid, we switch c_1 from an encryption of (σ, vk_{i_0}) to one of (σ', vk_{i_1}).

$H_7 \overset{s}{\approx} H_8$*:* This follows from the same argument for $H_1 \overset{s}{\approx} H_2$.

Hybrid H_9**:** In this hybrid, we switch back to using the pairwise independent hash function PI_1 to compute r_{c_1}, instead of using a truly random function. Effectively we are undoing the change made in H_7.

$H_8 \overset{i.d.}{=\!=} H_9$*:* This again follows from Lem. 1.

Hybrid H_{10}**:** In this hybrid, we switch to computing r_π as the output of a *random* function $r_\pi = P_3(\mathsf{R}\|m)$, instead of being computed using PI_3.

$H_9 \overset{i.d.}{=\!=} H_{10}$*:* This again follows from Lem. 1.

Hybrid H_{11}**:** In this hybrid, we again switch the witness used to generate π, from w' to $w'' = (vk_{i_1}, pk_{i_1}, \sigma', r_{c_1})$.

$H_{10} \overset{s}{\approx} H_{11}$*:* This follows from the same argument for $H_4 \overset{s}{\approx} H_5$.

Hybrid H_{12}**:** In this hybrid, we switch back to using the pairwise independent hash function PI_3 to compute r_π, instead of using a truly random function.

$H_{11} \overset{i.d.}{=\!=} H_{12}$*:* This again follows from Lem. 1.

Hybrid H_{13}**:** Here, we switch to computing r_{c_2} as the output of a *random* function $r_{c_2} = P_2(\mathsf{R}\|m)$.

$H_{12} \overset{i.d.}{=\!=} H_{13}$*:* This again follows from Lem. 1.

Hybrid H_{14}**:** In this hybrid, we switch c_2 to an encryption of zeroes, namely $c_2 = \mathsf{LE.Enc}(pk, 0^{|\sigma|+|vk|}; r_{c_2})$, instead of an encryption of (σ', vk_{i_1}).

$H_{13} \overset{s}{\approx} H_{14}$*:* This argument is identical to that for simlarity between $H_1 \overset{s}{\approx} H_2$.

Hybrid H_{15}**:** In this hybrid, we switch back to using the pairwise independent hash function PI_2 to compute r_{c_2}, instead of using a truly random function.

$H_{14} \overset{i.d.}{=\!=} H_{15}$*:* This again follows from Lem. 1.

Observe that H_{15} corresponds to sign using identity i_1 in Expr. 4. This finishes the proof of Lem. 6.

6.4.2 Proving Post-Quantum Blind-Unforgeability

In this section, we prove the following Lem. 7, which establishes post-quantum blind-unforgeability for Constr. 3.

Lemma 7. *Assume* Sig *is blind-unforgeable as per Def. 1,* LE *satisfies the completeness of master secret keys property (Item 3) and the almost-unique secret key property (Item 5), and* ZAP *has the selective non-witness adaptive-statement soundness (Item 3). Then, Constr. 3 is blind-unforgeable as per Def. 6.*

Consider a QPT adversary $\mathcal{A}_{\mathsf{RS}}$ participating in Expr. 5. We proceed using a sequence of hybrids to set up our reduction to the blind-unforgeability of Sig.

Hybrid H_0: This is just the post-quantum blind-sunforgeability game (Expr. 5) for our construction. In particular, for all $i \in [Q]$, the encryption key pk_i is generated as $pk_i \leftarrow \mathsf{LE.KSam}^{\mathsf{ls}}(1^\lambda; r_i)$. Recall that we are in the full key exposure setting, so both the public keys and random coins $\{pk_i, r_i\}_{i\in[Q]}$ are given to $\mathcal{A}$.

Hybrid H_1: In this experiment, the only difference is that, the challenger generates the $\{pk_i\}_{i\in[Q]}$ by running $\left(\{pk_i\}_{i\in[Q]}, \mathsf{msk}\right) \leftarrow \mathsf{LE.MSKGen}(1^\lambda, Q)$. The challenger keeps msk to itself, and sends $\left\{pk_i, \mathsf{LE.RndExt}(pk_i)\right\}_{i\in[Q]}$ to $\mathcal{A}$.

$\underline{H_0 \overset{c}{\approx} H_1}$*:* This follows immediately from the IND of $\mathsf{MSKGen/KSam}^{\mathsf{ls}}$ property (Item 4) of LE as specified in Def. 8. It is worth noting that $\mathcal{A}_{\mathsf{RS}}$'s quantum access to the signing algorithm does not affect this proof at all, since the pk_i's (contained in VK_i's) are sampled classically by the challenger before $\mathcal{A}_{\mathsf{RS}}$ makes any quantum sign queries.

Reduction to the BU of Sig. We proceed to show that post-quantum blind-unforgeability holds in H_1. Consider the adversary's forgery attempt

$$\left(\mathsf{R}^*, m^*, \Sigma^* = (c_1^*, c_2^*, \pi^*)\right) \text{ satisfying } (\mathsf{R}^*, m^*) \in B_\varepsilon^{\mathsf{RS}}.$$

Let $x^* := (\mathsf{R}^*, m^*, c_1^*, c_2^*)$. Let $\mathsf{VK}_1^* = (vk_1^*, pk_1^*, \rho_1^*)$ be the lexicographically smallest verification key in R^*.

Observe that for the x^* defined above, one of the following two cases must happen: $x^* \in \widetilde{L}$ or $x^* \notin \widetilde{L}$. (Recall that $\widetilde{L}$ is the super-complement of L defined in Eq. (4).) In the following, we show two claims. Claim 2 says that it cannot be the case that $x^* \in \widetilde{L}$, unless the ZAP verification rejects (thus, the forgery is invalid). Claim 3 says that $x^* \notin \widetilde{L}$ cannot happen either. Therefore, Claims 2 and 3 together show that any QPT adversary has negligible chance of winning the blind-unforgeability game for RS in H_1. Note that winning the post-quantum blind-unforgeability game for RS is an event that can be efficiently tested. Thus, by $H_0 \overset{c}{\approx} H_1$, no QPT adversaries can win the post-quantum blind-unforgeability game for RS in hybrid H_0. This concludes the proof of Lem. 7.

Now, the only thing left is to state and prove Claims 2 and 3, which is done in the following.

Claim 2. *In* H_1*, assume that* ZAP *satisfies selective non-witness adaptive statement soundness (Item 3). Then, the following holds:*

$$\Pr\left[x^* \in \widetilde{L} \ \wedge \ \mathsf{ZAP.Verify}(\rho_1^*, x^*, \pi^*) = 1\right] = \mathsf{negl}(\lambda).$$

Proof. First, notice that by definition, the R^* in $\mathcal{A}_{\mathsf{RS}}$'s forgery contains only VK's from the set $\mathcal{VK} = \{\mathsf{VK}_j\}_{j\in[Q]}$ generated by the challenger. Therefore, it suffices to show that for each $j \in [Q]$,

$$\Pr\Big[x^* \in \widetilde{L} \ \wedge \ \mathsf{ZAP.Verify}(\rho_j, x^*, \pi^*) = 1\Big] = \mathsf{negl}(\lambda), \tag{8}$$

where ρ_j denotes the first-round message of ZAP corresponding to the j-th verification key VK_j generated in the game.

Let $\mathcal{A}_{\mathsf{RS}}$ be an adversary attempting to output a forgery such that

$$x^* \in \widetilde{L} \ \text{ and } \ \mathsf{ZAP.Verify}(\rho_j, x^*, \pi^*) = 1.$$

We build an adversary $\mathcal{A}_{\mathsf{ZAP}}$ against the selective non-witness adaptive-statement soundness of ZAP for $(L, \widetilde{L})$ (defined in Eq. (1) and (4) respectively). The algorithm $\mathcal{A}_{\mathsf{ZAP}}$ proceeds as follows:

- On input the 1st ZAP message $\widehat{\rho}$, it sets $\rho_j = \widehat{\rho}$ and proceeds exactly as H_1.
- Upon receiving the forgery attempt $\big(\mathsf{R}^*, m^*, \Sigma^* = (c_1^*, c_2^*, \pi^*)\big)$ from $\mathcal{A}$, it outputs

$$\big(x^* \coloneqq (\mathsf{R}^*, m^*, c_1^*, c_2^*),\ \pi^*\big).$$

We remark that H_1 is quantum. So, $\mathcal{A}_{\mathsf{ZAP}}$ needs to be a quantum machine to simulate H_1 for $\mathcal{A}_{\mathsf{RS}}$. This is fine since we assume that the soundness (Item 3) of ZAP in Def. 10 holds against QPT adversaries.

To finish the proof, notice that $x^* \in \widetilde{L}$ means that there exists a "non-witness" $\widetilde{w}^*$ such that $(x^*, \widetilde{w}^*) \in \widetilde{R}$. Therefore, if Eq. (8) does not hold, (ρ_j, x^*, π^*) will break the soundness (Item 3) w.r.t. the non-witness $\widetilde{w}$. □

Claim 3. *In H_1, assume that* Sig *satisfies the blind-unforgeability as per Def. 1,* LE *satisfies the completeness of master secret keys propoerty (Item 3) and the almost-unique secret key property (Item 5). Then,* $\Pr\left[x^* \notin \widetilde{L}\right] = \mathsf{negl}(\lambda)$.

Proof. Let $\mathcal{A}_{\mathsf{RS}}$ be a QPT adversary attempting to output a forgery w.r.t. our RS scheme such that $x^* \notin \widetilde{L}$. We build an algorithm $\mathcal{A}_{\mathsf{Sig}}$ against the blind-unforgeability of Sig. The algorithm $\mathcal{A}_{\mathsf{Sig}}$ (playing the blind-unforgeability game Expr. 1 for Sig) proceeds as follows:

1. invoke $\mathcal{A}_{\mathsf{RS}}$ to obtain the ε for the blind-unforgeability game of RS; give this ε to $\mathcal{A}_{\mathsf{Sig}}$'s own challenger for the blind-unforgeability game of Sig;
2. receive $\widehat{vk}$ from its own challenger; pick an index $j \xleftarrow{\$} [Q]$ uniformly at random; set $vk_j \coloneqq \widehat{vk}$; then, proceeds as in H_1 to prepare the rest of the verification keys and continue the execution with $\mathcal{A}_{\mathsf{RS}}$.
3. when $\mathcal{A}_{\mathsf{RS}}$ sends a quantum signing query $(\mathsf{sign}, i, \sum \psi_{\mathsf{R},m,t}\,|\mathsf{R}, m, t\rangle)$, if the specified identity i is not equal to j, it proceeds as in H_1; otherwise, it uses the blind-unforgeability (for Sig) game's signing oracle $\mathsf{Sig.Sign}$ to obtain a Sig signature for the j-th party and then continues exactly as in H_1; (See the paragraph right after the description of $\mathcal{A}_{\mathsf{Sig}}$.)
4. if $\mathcal{A}_{\mathsf{RS}}$ tries to corrupt the j-th party, $\mathcal{A}_{\mathsf{Sig}}$ aborts; (It is worth noting that the identities are classical. So, $\mathcal{A}_{\mathsf{RS}}$'s quantum power does not affect this step.)

5. upon receiving the forgery attempt Σ^* from $\mathcal{A}_{\mathsf{RS}}$, $\mathcal{A}_{\mathsf{Sig}}$ decrypts c_1^* using msk to recover σ_1^*. (Recall that, the secret key for pk_j can be obtained as $\mathsf{LE.MSKExt}(\mathsf{msk}, pk_j)$). If

$$\mathsf{Sig.Verify}(vk_j, \mathsf{R}^*\|m^*, \sigma_1^*) = 1,$$

it sets $\widehat{\sigma} := \sigma_1^*$. Otherwise, it decrypts c_2^* with msk to recover σ_2^*, and sets $\widehat{\sigma} := \sigma_2^*$. It outputs $(\mathsf{R}^*\|m^*, \widehat{\sigma})$.

We first remark that, up to (inclusively) Step 3, $\mathcal{A}_{\mathsf{RS}}$'s view is identical to that in H_1. Recall that in H_1, the challenger maintains a blindset $B_\varepsilon^{\mathsf{RS}}$ such that any $(\mathsf{R}, m) \in B_\varepsilon^{\mathsf{RS}}$ will not be answered (this is inherited from H_0, which is exactly Expr. 5). In contrast, in the execution of $\mathcal{A}_{\mathsf{Sig}}$ described above, $\mathcal{A}_{\mathsf{Sig}}$ first forwards the $\sum \psi_{\mathsf{R},m,t} |\mathsf{R}, m, t\rangle$ part of $\mathcal{A}_{\mathsf{RS}}$'s query to its $\mathsf{Sig.Sign}$ oracle to obtain $\sum_{\mathsf{R},m,t} \psi_{\mathsf{R},m,t} \left|\mathsf{R}, m, t \oplus B_\varepsilon^{\mathsf{Sig}} \mathsf{Sig.Sign}(sk_j, \mathsf{R}\|m)\right\rangle$ (note that $sk_j = \widehat{sk}$), and then performs the remaining computation exactly as in H_1. Note that the $B_\varepsilon^{\mathsf{Sig}}$ is the blindset maintained by the Sig signing algorithm. Importantly, since the "messages" singed by Sig are of the form $\mathsf{R}\|m$, $B_\varepsilon^{\mathsf{Sig}}$ is actually generated identically to $B_\varepsilon^{\mathsf{RS}}$—that is, both of them are generated by including each (R, m) pair in with (the same) probability ε.

To finish the proof, we show that $(\mathsf{R}^*\|m^*, \widehat{\sigma})$ is a valid forgery against Sig's blind-unforgeability game with probability at least $\frac{1}{Q}\big(\Pr\left[x^* \notin \widetilde{L}\right] - \mathsf{negl}(\lambda)\big)$.

Recall that we are focusing on the case $x^* \notin \widetilde{L}$, where $\widetilde{L}$ is defined in Eq. (4); without loss of generality, assume that $(\mathsf{R}^*, m^*, c_1^*) \notin \widehat{L}$. Then, observe that due to the way H_1 generates the public keys (more acurately, Item 3) and that $\mathsf{R}^* \subseteq \mathcal{VK} \setminus \mathcal{C}$ (in particular, $\mathsf{R}^* \subseteq \mathcal{VK}$), we have

$$\big((\mathsf{R}^*, m^*, c_1^*), \mathsf{msk}\big) \in R_4 \quad \text{(recall that } R_4 \text{ is defined in Eq. (2))}. \tag{9}$$

Since we assume that $(\mathsf{R}^*, m^*, c_1^*) \notin \widehat{L}$, Expression (9) and the definition of $\widehat{L}$ imply the existence of a string $\widetilde{w}$ such that

$$\big((\mathsf{R}^*, m^*, c_1^*), \widetilde{w}\big) \in R_5 \quad \text{(recall that } R_5 \text{ is defined in Eq. (3))}. \tag{10}$$

We remark that the $\widetilde{w}$ may not equal msk. However, note that R_5 (Eq. (3)) tests if

$$\mathsf{LE.Valid}\big(pk, \mathsf{LE.MSKExt}(\widetilde{w}, pk)\big) = 1$$

with respect to the pk contained in some VK in the ring. If this test passes, by Expression (9) and the almost-unique secret key property (Item 5) of LE, it must hold for this pk that

$$\mathsf{LE.MSKExt}(\widetilde{w}, pk) = \mathsf{LE.MSKExt}(\mathsf{msk}, pk),$$

except with negligible probability.

To summarize, the above argument implies the following facts:

1. by our assumption, $(\mathsf{R}^*, m^*) \in B_\lambda^{\mathsf{RS}}$; this also implies $(\mathsf{R}^*, m^*) \in B_\lambda^{\mathsf{Sig}}$ because $B_\varepsilon^{\mathsf{Sig}} = B_\varepsilon^{\mathsf{RS}}$ as argued earlier;
2. by Expression (10), for some $\mathsf{VK} = (vk^*, pk^*, \rho^*) \in \mathsf{R}^*$, it must hold that

$$\mathsf{LE.Dec}\big(\mathsf{LE.MSKExt}(\widetilde{w}, pk), c_1^*\big) = (\sigma^*, vk^*) \quad \text{and} \quad \mathsf{Sig.Verify}(vk^*, \mathsf{R}^*\|m^*, \sigma^*) = 1.$$

Also, as mentioned earlier, $\mathsf{LE.MSKExt}(\widetilde{w}, pk^*) = \mathsf{LE.MSKExt}(\mathsf{msk}, pk^*)$ for this pk^*.

The above means that the $\mathcal{A}_{\mathsf{RS}}$ uses a $\mathsf{VK}^* = (vk^*, pk^*, \rho^*) \in \mathsf{R}^* \subseteq \mathcal{VK} \setminus \mathcal{C}$ such that c_1^* encrypts (among other things) a signature σ^* that is valid for the forgery message $\mathsf{R}^* \| m^*$ w.r.t. key vk^* (for the blind-unforgeability game of Sig). Moreover, $\mathcal{A}_{\mathsf{Sig}}$ can extract this forgery message efficiently by decrypting c^* using $\mathsf{LE.MSKExt}(\mathsf{msk}, pk^*)$!

Finally, observe that index j is sampled uniformly. Therefore, we have that $(\widehat{vk} =)\ vk_j = vk^*$ with probability $1/Q$. □

6.5 Discussion on Compactness

Our construction of post-quantum ring signatures (i.e., Constr. 3) is currently only of theoretical interest. It is not efficient, and it does not enjoy compactness (i.e., the signatures size is independent of, or even poly-logarithmic on, the ring size). It is an interesting problem for future research to construction practical or compact ring signatures that satisfies our notion of post-quantum security. In the following, we briefly discuss why this seems non-trivial.

Efficiency. Almost all known efficient ring signatures are in the random oracle model, following the Fiat-Shamir paradigm (e.g., [ABB+13, LLNW16, TSS+18, BLO, WZZ18, EZS+19, BKP20, LNS21]). Although these constructions are based on post-quantum hardness assumptions, their security proofs can only handle adversaries making *classical* random oracle queries. Making these constructions secure in the QROM requires a quantum version of the *forking lemma* [PS96b, BN06], which seems hard to prove. Indeed, this problem is still open even for (ordinary) signatures in the post-quantum setting (e.g., see the discussion in [Unr17]). (As a side note, our construction in Sec. 4 does not face this problem as it follows the hash-and-sign paradigm, instead of Fiat-Shamir.)

Compactness. The original construction in [CGH+21] does achieve compactness. Although based on their work, our construction in Sec. 6.1 gives up compactness by using the underlying Sig to sign $\mathsf{R}\|m$ together[11]; in contrast, [CGH+21] only uses Sig to sign m. Our choice is critical to achieving BU: when proving BU for our RS, we need to reduce to the BU of Sig. The RS game will "blind" (R, m) pairs, while the Sig game only blinds messages m. If we do not use $\mathsf{R}\|m$ as the message for Sig to sign, the reduction will not be able to create the blindset in a consistent manner. This problem cannot be resolved by applying some type of "hash" function on $\mathsf{R}\|m$ and asking Sig to sign the short digest. Indeed, blinding (R, m) pairs with probability ε is different from blinding the hash result of $\mathsf{R}\|m$, unless the "hash" has pseudo-random output. Replacing the "hash" with a PRF does not work either, as the verifier also needs to evaluate the "hash" to verify the signature. We leave it as an open question to construct *compact* ring signatures achieving our post-quantum security notion.

7 Acknowledgments

We thank the anonymous PKC 2022 reviewers for their valuable comments.

Rohit Chatterjee and Xiao Liang are supported in part by Omkant Pandey's DARPA SIEVE Award HR00112020026 and NSF grants 1907908 and 2028920. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government, DARPA, or NSF.

[11] Note that our construction will not become compact even if we use a compact Sig. This is because the size of the ZAP proof for the validity of the signature for $\mathsf{R}\|m$ also depends on the size of the rings.

Kai-Min Chung is supported by Ministry of Science and Technology, Taiwan, under Grant No. MOST 109-2223-E-001-001-MY3.

Giulio Malavolta is supported by the German Federal Ministry of Education and Research BMBF (grant 16K15K042, project 6GEM).

References

ABB10. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Gilbert [Gil10], pages 553–572. 14, 15, 44, 45

ABB+13. Carlos Aguilar Melchor, Slim Bettaieb, Xavier Boyen, Laurent Fousse, and Philippe Gaborit. Adapting Lyubashevsky's signature schemes to the ring signature setting. In Amr Youssef, Abderrahmane Nitaj, and Aboul Ella Hassanien, editors, *AFRICACRYPT 13: 6th International Conference on Cryptology in Africa*, volume 7918 of *Lecture Notes in Computer Science*, pages 1–25, Cairo, Egypt, June 22–24, 2013. Springer, Heidelberg, Germany. 3, 37

ABDS. Gorjan Alagic, Zvika Brakerski, Yfke Dulek, and Christian Schaffner. Impossibility of quantum virtual black-box obfuscation of classical circuits. In *CRYPTO 2021*. Springer. 2

ABG+. Amit Agarwal, James Bartusek, Vipul Goyal, Dakshita Khurana, and Giulio Malavolta. Post-quantum multi-party computation. In *EUROCRYPT 2021*, pages 435–464. Springer. 1

Ajt96. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th Annual ACM Symposium on Theory of Computing*, pages 99–108, Philadephia, PA, USA, May 22–24, 1996. ACM Press. 43

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, *Advances in Cryptology – EUROCRYPT 2020, Part III*, volume 12107 of *Lecture Notes in Computer Science*, pages 788–817, Zagreb, Croatia, May 10–14, 2020. Springer, Heidelberg, Germany. 1, 2, 4, 9, 10, 24

ARU14. Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th Annual Symposium on Foundations of Computer Science*, pages 474–483, Philadelphia, PA, USA, October 18–21, 2014. IEEE Computer Society Press. 1

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, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69, Seoul, South Korea, December 4–8, 2011. Springer, Heidelberg, Germany. 1

BDH+19. Michael Backes, Nico Döttling, Lucjan Hanzlik, Kamil Kluczniak, and Jonas Schneider. Ring signatures: Logarithmic-size, no setup - from standard assumptions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 281–311, Darmstadt, Germany, May 19–23, 2019. Springer, Heidelberg, Germany. 20

BGG+14. Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 533–556, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany. 15, 44, 45

BK10. Zvika Brakerski and Yael Tauman Kalai. A framework for efficient signatures, ring signatures and identity based encryption in the standard model. *IACR Cryptol. ePrint Arch.*, page 86, 2010. 3

BKM06. Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 60–79, New York, NY, USA, March 4–7, 2006. Springer, Heidelberg, Germany. 3, 20, 22

BKP20. Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 464–492, Daejeon, South Korea, December 7–11, 2020. Springer, Heidelberg, Germany. 3, 37

BL16. Xavier Boyen and Qinyi Li. Towards tightly secure lattice short signature and id-based encryption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 404–434, Hanoi, Vietnam, December 4–8, 2016. Springer, Heidelberg, Germany. 4, 5, 19

BLO. Carsten Baum, Huang Lin, and Sabine Oechsner. Towards practical lattice-based one-time linkable ring signatures. In *The 2018 International Conference on Information and Communications Security*. Springer. 3, 37

BLP+13. Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 575–584, Palo Alto, CA, USA, June 1–4, 2013. ACM Press. 44

BM19. Alexandra Boldyreva and Daniele Micciancio, editors. *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany. 39, 40, 41

BM21. James Bartusek and Giulio Malavolta. Indistinguishability obfuscation of null quantum circuits and applications. Cryptology ePrint Archive, Report 2021/421, 2021. https://ia.cr/2021/421. 2

BN06. Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006: 13th Conference on Computer and Communications Security*, pages 390–399, Alexandria, Virginia, USA, October 30 – November 3, 2006. ACM Press. 37

Boy10. Xavier Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 499–517, Paris, France, May 26–28, 2010. Springer, Heidelberg, Germany. 14

BPR12. Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In Pointcheval and Johansson [PJ12], pages 719–737. 6, 10, 15

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: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press. 1

BV14. Zvika Brakerski and Vinod Vaikuntanathan. Lattice-based FHE as secure as PKE. In Moni Naor, editor, *ITCS 2014: 5th Conference on Innovations in Theoretical Computer Science*, pages 1–12, Princeton, NJ, USA, January 12–14, 2014. Association for Computing Machinery. 5, 8, 15, 16, 45, 46

BZ13a. Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 592–608, Athens, Greece, May 26–30, 2013. Springer, Heidelberg, Germany. 1, 2

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, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 361–379, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany. 1, 2, 3, 4, 5, 7, 8, 9, 10, 22, 23, 46

CETU20. Tore Vincent Carstens, Ehsan Ebrahimi, Gelo Noel Tabia, and Dominique Unruh. On quantum indistinguishability under chosen plaintext attack. *IACR Cryptol. ePrint Arch.*, page 596, 2020. 2

CEV20. Céline Chevalier, Ehsan Ebrahimi, and Quoc Huy Vu. On the security notions for encryption in a quantum world. *IACR Cryptol. ePrint Arch.*, page 237, 2020. 2

CGH+21. Rohit Chatterjee, Sanjam Garg, Mohammad Hajiabadi, Dakshita Khurana, Xiao Liang, Giulio Malavolta, Omkant Pandey, and Sina Shiehian. Compact ring signatures from learning with errors. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 282–312. Springer, 2021. 3, 7, 8, 22, 23, 27, 28, 29, 30, 37

CHKP10. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Gilbert [Gil10], pages 523–552. 14

CHS19. Jan Czajkowski, Andreas Hülsing, and Christian Schaffner. Quantum indistinguishability of random sponges. In Boldyreva and Micciancio [BM19], pages 296–325. 2

Com21. Personal Communication. Personal communication with the authors of [amrs20], 2021. 2

Cv91. David Chaum and Eugène van Heyst. Group signatures. In Donald W. Davies, editor, *Advances in Cryptology – EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265, Brighton, UK, April 8–11, 1991. Springer, Heidelberg, Germany. 3

DFM20. Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology*

– *CRYPTO 2020, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 602–631, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Heidelberg, Germany. 1

DFMS19. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In Boldyreva and Micciancio [BM19], pages 356–383. 1

DFNS13. Ivan Damgård, Jakob Funder, Jesper Buus Nielsen, and Louis Salvail. Superposition attacks on cryptographic protocols. In *International Conference on Information Theoretic Security*, pages 142–161. Springer, 2013. 1

EZS+19. Muhammed F. Esgin, Raymond K. Zhao, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. MatRiCT: Efficient, scalable and post-quantum blockchain confidential transactions protocol. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019: 26th Conference on Computer and Communications Security*, pages 567–584. ACM Press, November 11–15, 2019. 3, 37

FC16. Marc Fischlin and Jean-Sébastien Coron, editors. *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany. 41

GHHM20. Alex B. Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight adaptive reprogramming in the QROM. Cryptology ePrint Archive, Report 2020/1361, 2020. https://eprint.iacr.org/2020/1361. 1

GHS16. Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. Semantic security and indistinguishability in the quantum world. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 60–89, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany. 1, 2

Gil10. Henri Gilbert, editor. *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, French Riviera, May 30 – June 3, 2010. Springer, Heidelberg, Germany. 38, 39

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 Annual ACM Symposium on Theory of Computing*, pages 197–206, Victoria, BC, Canada, May 17–20, 2008. ACM Press. 2, 4, 10, 11

GSW13. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany. 15, 45

GVW15. Sergey Gorbunov, Vinod Vaikuntanathan, and Daniel Wichs. Leveled fully homomorphic signatures from standard lattices. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th Annual ACM Symposium on Theory of Computing*, pages 469–477, Portland, OR, USA, June 14–17, 2015. ACM Press. 26

GYZ17. Sumegha Garg, Henry Yuen, and Mark Zhandry. New security notions and feasibility results for authentication of quantum data. In Katz and Shacham [KS17], pages 342–371. 2

HI19. Akinori Hosoyamada and Tetsu Iwata. 4-round Luby-Rackoff construction is a qPRP. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019, Part I*, volume 11921 of *Lecture Notes in Computer Science*, pages 145–174, Kobe, Japan, December 8–12, 2019. Springer, Heidelberg, Germany. 2

HI21. Akinori Hosoyamada and Tetsu Iwata. On tight quantum security of hmac and nmac in the quantum random oracle model. In *Annual International Cryptology Conference*, pages 585–615. Springer, 2021. 2

HS21. Akinori Hosoyamada and Yu Sasaki. Quantum collision attacks on reduced SHA-256 and SHA-512. In *Annual International Cryptology Conference*, pages 616–646. Springer, 2021. 2

HY18. Akinori Hosoyamada and Kan Yasuda. Building quantum-one-way functions from block ciphers: Davies-Meyer and Merkle-Damgård constructions. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part I*, volume 11272 of *Lecture Notes in Computer Science*, pages 275–304, Brisbane, Queensland, Australia, December 2–6, 2018. Springer, Heidelberg, Germany. 2

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, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 552–586, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany. 1

KR00. Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2000, San Diego, California, USA*. The Internet Society, 2000. 2

KS17. Jonathan Katz and Hovav Shacham, editors. *Advances in Cryptology – CRYPTO 2017, Part II*, volume 10402 of *Lecture Notes in Computer Science*, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany. 40, 41

Lam79. Leslie Lamport. Constructing digital signatures from a one-way function. Technical report, Citeseer, 1979. 2

LLNW16. Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In Fischlin and Coron [FC16], pages 1–31. 3, 37

LNS21. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Smile: set membership from ideal lattices with applications to ring signatures and confidential transactions. In *Annual International Cryptology Conference*, pages 611–640. Springer, 2021. 3, 37

LZ19. Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat-Shamir. In Boldyreva and Micciancio [BM19], pages 326–355. 1

Mer90. Ralph C. Merkle. A certified digital signature. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 218–238, Santa Barbara, CA, USA, August 20–24, 1990. Springer, Heidelberg, Germany. 2

MMO21. Christian Majenz, Chanelle Matadah Manfouo, and Maris Ozols. Quantum-access security of the winternitz one-time signature scheme. *arXiv preprint arXiv:2103.12448*, 2021. 2

MP12. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In Pointcheval and Johansson [PJ12], pages 700–718. 43, 44

MR04. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th Annual Symposium on Foundations of Computer Science*, pages 372–381, Rome, Italy, October 17–19, 2004. IEEE Computer Society Press. 43, 44

Noe15. Shen Noether. Ring signature confidential transactions for monero. Cryptology ePrint Archive, Report 2015/1098, 2015. https://eprint.iacr.org/2015/1098. 3

Pei09. Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 333–342, Bethesda, MD, USA, May 31 – June 2, 2009. ACM Press. 44

PJ12. David Pointcheval and Thomas Johansson, editors. *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg, Germany. 39, 41

PRS17. Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-LWE for any ring and modulus. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th Annual ACM Symposium on Theory of Computing*, pages 461–473, Montreal, QC, Canada, June 19–23, 2017. ACM Press. 44

PS96a. David Pointcheval and Jacques Stern. Provably secure blind signature schemes. In Kwangjo Kim and Tsutomu Matsumoto, editors, *Advances in Cryptology – ASIACRYPT'96*, volume 1163 of *Lecture Notes in Computer Science*, pages 252–265, Kyongju, Korea, November 3–7, 1996. Springer, Heidelberg, Germany. 2

PS96b. David Pointcheval and Jacques Stern. Security proofs for signature schemes. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398, Saragossa, Spain, May 12–16, 1996. Springer, Heidelberg, Germany. 37

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93, Baltimore, MA, USA, May 22–24, 2005. ACM Press. 44

RST01. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565, Gold Coast, Australia, December 9–13, 2001. Springer, Heidelberg, Germany. 3

SY17. Fang Song and Aaram Yun. Quantum security of NMAC and related constructions - PRF domain extension against quantum attacks. In Katz and Shacham [KS17], pages 283–309. 2

TSS+18. Wilson Abel Alberto Torres, Ron Steinfeld, Amin Sakzad, Joseph K Liu, Veronika Kuchta, Nandita Bhattacharjee, Man Ho Au, and Jacob Cheng. Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (lattice ringct v1. 0). In *Australasian Conference on Information Security and Privacy*, pages 558–576. Springer, 2018. 3, 37

Unr16. Dominique Unruh. Computationally binding quantum commitments. In Fischlin and Coron [FC16], pages 497–527. 1, 2

Unr17. Dominique Unruh. Post-quantum security of Fiat-Shamir. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 65–95, Hong Kong, China, December 3–7, 2017. Springer, Heidelberg, Germany. 1, 37

Wat06. John Watrous. Zero-knowledge against quantum attacks. In Jon M. Kleinberg, editor, *38th Annual ACM Symposium on Theory of Computing*, pages 296–305, Seattle, WA, USA, May 21–23, 2006. ACM Press. 1

WZZ18. Shangping Wang, Ru Zhao, and Yaling Zhang. Lattice-based ring signature scheme under the random oracle model. *Int. J. High Perform. Comput. Netw.*, 11(4):332–341, 2018. 3, 37

Zha12a. Mark Zhandry. How to construct quantum random functions. In *53rd Annual Symposium on Foundations of Computer Science*, pages 679–687, New Brunswick, NJ, USA, October 20–23, 2012. IEEE Computer Society Press. 1, 2, 10

Zha12b. Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 758–775, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Heidelberg, Germany. 8

Zha15. Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Inf. Comput.*, 15(7&8):557–567, 2015. 1, 2

Supplementary Material

A Additional Preliminaries

A.1 Preliminaries for Lattice

Throughout the current paper, we denote the Gram-Schmidt ordered orthogonalization of a matrix $\mathbf{A} \in \mathbb{Z}^{m \times m}$ by $\widetilde{\mathbf{A}}$.

A.1.1 Lattices

We define the notion of a lattice and integer lattice.

Definition 11 (Lattice). *Let* $\mathbf{B} = [\ \mathbf{b}_1 \mid \ldots \mid \mathbf{b}_m\ ]$ *be a basis of linearly independent vectors* $\mathbf{b}_i \in \mathbb{R}^m,\ i \in [m]$. *The* lattice *generated by* $\mathbf{B}$ *is defined as* $\Lambda = \{\mathbf{y} \in \mathbb{R}^m : \exists s_i \in \mathbb{Z}, \mathbf{y} = \sum_1^m s_i \mathbf{b}_i\}$. *The* dual lattice Λ^* *of* Λ *is defined as* $\Lambda^* = \{\mathbf{z} \in \mathbb{R}^m : \forall y \in \Lambda, \langle \mathbf{z}, \mathbf{y} \rangle \in \mathbb{Z}\}$

Definition 12 (Integer Lattice). *For a prime* q, *a modular matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *and vector* $\mathbf{u} \in \mathbb{Z}_q^n$, *we define the* m*-dimensional (full rank) integer lattice* $\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A}\mathbf{e} = 0 \ (\mod q)\}$, *and the 'shifted' lattice as the coset* $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A}\mathbf{e} = \mathbf{u} \ (\mod q)\}$

A.1.2 Lattice Trapdoors, Discrete Gaussians

The works [Ajt96, MP12] show how to sample close to uniform matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ along with a matrix trapdoor $\mathbf{T}_{\mathbf{A}}$ that consists of a basis of low norm vectors for the associated lattice $\Lambda_q^{\perp}(\mathbf{A})$. We call this sampling procedure $\mathsf{TrapGen}$.

Lemma 8 (Trapdoor Matrices). *There is a PPT algorithm* $\mathsf{TrapGen}$ *that given as input integers* $n \geq 1$, $q \geq 2$, *and (sufficiently large)* $m = O(n \log q)$, *outputs a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *and a trapdoor matrix* $\mathbf{T}_{\mathbf{A}} \in \mathbb{Z}_q^{m \times m}$, *such that* $\mathbf{A}\mathbf{T}_{\mathbf{A}} = 0$, *the distribution of* $\mathbf{A}$ *is statistically close to uniform over* $\mathbb{Z}_q^{n \times m}$, *and* $||\widetilde{\mathbf{T}}_{\mathbf{A}}|| = O(\sqrt{n \log q})$.

We now define the notion of discrete Gaussian distributions.

Definition 13 (Discrete Gaussians). *Let* $m \in \mathbb{Z}_{>0}$, $\Lambda \subset \mathbb{Z}^m$. *For any vector* $\mathbf{c} \in \mathbb{R}^m$, *and positive real* $\sigma \in \mathbb{R}_{>0}$, *define the Gaussian function* $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \mathsf{exp}(-\pi||\mathbf{x} - \mathbf{c}||^2/\sigma^2)$ *over* $\mathbb{R}^m$ *with center* $\mathbf{c}$ *and width* σ. *Define the discrete Gaussian distribution over* Λ *with center* $\mathbf{c}$ *and width* σ *as* $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}} = \rho_{\sigma,\mathbf{c}}/\rho_{\sigma}(\Lambda)$ *where* $\rho_{\sigma}(\Lambda) = \sum_{x \in Lambda} \rho_{\sigma,\mathbf{c}}$. *For convenience, we use the shorthand* ρ_{σ} *and* $\mathcal{D}_{\Lambda,\sigma}$ *for* $\rho_{\sigma,\mathbf{0}}$ *and* $\mathcal{D}_{\Lambda,\sigma,\mathbf{0}}$ *respectively.*

The following lemma is a very useful concentration bound on the norm of discrete guassian samples, depending on the basis they were sampled using.

Lemma 9 (Discrete Gaussian Concentration [MR04]). *For any lattice* Λ *of integer dimension* m *with basis* $\mathbf{T}$, $\mathbf{c} \in \mathbb{R}^m$, *and Gaussian width parameter* $\sigma \geq ||\widetilde{\mathbf{T}}|| \cdot \omega\big(\sqrt{\log m}\big)$, *we have*

$$\Pr\big[\mathbf{x} \leftarrow \mathcal{D}_{\Lambda,\sigma,\mathbf{c}} : ||\mathbf{x} - \mathbf{c}|| > \sigma\sqrt{m}\big] \leq \mathsf{negl}(n)$$

A.1.3 The Gadget Matrix

The gadget matrix $\mathbf{G}$ was defined in [MP12]. We use the following two properties of $\mathbf{G}$ in particular:

Lemma 10 ([MP12, Theorem 1]). *Let q be a prime, and n, m be integers with $m = n \log q$. There is a fixed full-rank matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ such that the lattice $\Lambda_q^{\perp}(\mathbf{G})$ has a publicly known trapdoor matrix $\mathbf{T_G} \in \mathbb{Z}^{n \times m}$ with $||\widetilde{\mathbf{T}}_{\mathbf{G}}|| \leq \sqrt{5}$.*

Lemma 11 ([BGG+14, Lemma 2.1]). *There is a deterministic algorithm, denoted by $\mathbf{G}^{-1}(\cdot) : \mathbb{Z}_q^{n \times m} \to \mathbb{Z}^{m \times m}$ that takes a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ as input, and outputs a 'preimage' $\mathbf{G}^{-1}(\mathbf{A})$ of $\mathbf{A}$ such that $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{A}) = \mathbf{A} \pmod q$ and $||\mathbf{G}^{-1}(\mathbf{A})|| \leq m$.*

A.1.4 Hardness Assumptions

We recall the LWE and SIS problems, and their hardness based on worst case lattice problems.

For a positive integer dimension n and modulus q, and an error distribution χ over $\mathbb{Z}$, the LWE distribution and decision problem are defined as follows. For an $\mathbf{s} \in \mathbb{Z}^n$, the LWE distribution $A_{\mathbf{s},\chi}$ is sampled by choosing a uniformly random $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ and an error term $e \leftarrow \chi$, and outputting $(\mathbf{a}, b = \langle \mathbf{s}, \mathbf{a} \rangle + e) \in \mathbb{Z}_q^{n+1}$.

Definition 14. *The decision-LWE$_{n,q,\chi}$ problem is to distinguish, with non-negligible advantage, between any desired (but polynomially bounded) number of independent samples drawn from $A_{\mathbf{s},\chi}$ for a single $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, and the same number of* uniformly random *and independent samples over $\mathbb{Z}_q^{n+1}$.*

A standard instantiation of LWE is to let χ be a *discrete Gaussian* distribution over $\mathbb{Z}$ with parameter $r = 2\sqrt{n}$. A sample drawn from this distribution has magnitude bounded by, say, $r\sqrt{n} = \Theta(n)$ except with probability at most 2^{-n}, and hence this tail of the distribution can be entirely removed. For this parameterization, it is known that LWE is at least as hard as *quantumly* approximating certain "short vector" problems on n-dimensional lattices, in the worst case, to within $\widetilde{O}(q\sqrt{n})$ factors [Reg05, PRS17]. Classical reductions are also known for different parameterizations [Pei09, BLP+13].

Definition 15. *The $\mathbf{SIS}_{q,\beta,n,m}$ problem is: given an uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, find a nonzero integral vector $\mathbf{z} \in \mathbb{Z}^m$ such that $\mathbf{Az} = \mathbf{0} \pmod q$, and $||\mathbf{z}|| \leq \beta$.*

When $q \geq \beta \cdot \widetilde{O}(\sqrt{n})$, solving $\mathbf{SIS}_{q,\beta,n,m}$ is at least as hard as approximating certain worst-case lattice problems (namely, SIVP) to within a $\beta \cdot \widetilde{O}(\sqrt{n})$ factor [MR04].

A.2 Random Sampling Related

We recall the following generalization of the leftover hash lemma.

Lemma 12 ([ABB10, Lemma 4]). *Suppose that $m > (n+1)\log_2 q + \omega(\log n)$ and that $q > 2$ is a prime. Let $\mathbf{R}$ be an $m \times k$ matrix chosen uniformly from $\{-1, 1\}^{m \times k} \pmod q$ where $k = k(n)$ is polynomial in n. Let $\mathbf{A}$ and $\mathbf{B}$ be matrices chosen uniformly in $\mathbb{Z}_q^{n \times m}$ and $\mathbb{Z}_q^{n \times k}$ respectively. Then for all vectors $\mathbf{w} \in \mathbb{Z}_q^m$, the distribution $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^\top \mathbf{w})$ is statistically close to the distribution $(\mathbf{A}, \mathbf{B}, \mathbf{R}^\top \mathbf{w})$.*

We will give an argument to show how Corollary 1 follows from this. This goes as follows: assume we start with $(\mathbf{A}', \mathbf{A}'\mathbf{R}, \mathbf{R}^\top \mathbf{w})$. This is statistically close to $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^\top \mathbf{w})$ since $\mathbf{A}$ is sampled uniformly, and $\mathbf{A}' \overset{s}{\approx} \mathbf{A}$. By Lem. 12 above, $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^\top \mathbf{w}) \overset{s}{\approx} (\mathbf{A}, \mathbf{B}, \mathbf{R}^\top \mathbf{w})$. The latter is in turn statistically close to $(\mathbf{A}', \mathbf{B}, \mathbf{R}^\top \mathbf{w})$. Therefore, we have $(\mathbf{A}', \mathbf{B}, \mathbf{R}^\top \mathbf{w}) \overset{s}{\approx} (\mathbf{A}', \mathbf{A}'\mathbf{R}, \mathbf{R}^\top \mathbf{w})$, concluding the proof for Corollary 1.

We also recall the following concentration bound on the operator norm for the matrices $\mathbf{R}$.

Lemma 13 ([ABB10, Lemma 5]). *Let $\mathbf{R}$ be an uniformly random chosen matrix from $\{-1, 1\}^{m \times m}$, then $\Pr\left[||\mathbf{R}||_2 > 12\sqrt{2m}\right] < e^{-m}$.*

A.3 Key-Homomorphic Evaluation Algorithms

We recall the matrix key-homomorphic evaluation algorithm from [GSW13, BGG+14, BV14] more fully. This was developed in the context of fully homorphic encryption and attribute-based encryption. This template works generally as follows: given a Boolean NAND circuit $C : \{0,1\}^\ell \to \{0,1\}$ with fan-in 2, ℓ matrices $\{\mathbf{A}_i = \mathbf{AR}_i + x_i\mathbf{G} \in \mathbb{Z}_q^{n \times m}\}_{i \in [\ell]}$ which correspond to each input wire of C where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{R}_i \xleftarrow{\$} \{-1, 1\}^{m \times m}$, $x_i \in \{0, 1\}$ and $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ is the gadget matrix, the key-homomorphic evaluation algorithm deterministically computes $\mathbf{A}_C = \mathbf{AR}_C + C(x_1, \ldots, x_\ell)\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ where $\mathbf{R}_C \in \{-1, 1\}^{m \times m}$ has low norm and $C(x_1, \ldots, x_\ell) \in \{0, 1\}$ is the output bit of C on the arguments $x_1, \ldots, x_\ell$. This is done by inductively evaluating each NAND gate. For a NAND gate $g(u, v; w)$ with input wires u, v and output wire w, we have (inductively) matrices $\mathbf{A}_u = \mathbf{AR}_u + x_u\mathbf{G}$, and $\mathbf{A}_v = \mathbf{AR}_v + x_v\mathbf{G}$ where x_u and x_v are the input bits of u and v respectively, and the evaluation algorithm computes

$$\begin{aligned} \mathbf{A}_w &= \mathbf{G} - \mathbf{A_u} \cdot \mathbf{G}^{-1}(\mathbf{A}_v) \\ &= \mathbf{G} - (\mathbf{AR}_u + x_u\mathbf{G}) \cdot \mathbf{G}^{-1}(\mathbf{AR}_v + x_v\mathbf{G}) \\ &= \mathbf{AR}_g + (1 - x_u x_v)\mathbf{G} \end{aligned} \tag{11}$$

where $1 - x_u x_v := \mathsf{NAND}(x_u, x_v)$, and $\mathbf{R}_g = -\mathbf{R}_u \cdot \mathbf{G}^{-1}(\mathbf{A}_v) - x_u\mathbf{R}_v$ has low norm if both $\mathbf{R}_u$ and $\mathbf{R}_v$ have low norm.

In [BV14], Brakerski and Vaikuntanathan observed that the norm of $\mathbf{R}_C$ in the outlined evaluation procedure grows asymmetrically (in the $\mathbf{R}$s corresponding to the input wires). They exploited this observation to design a special evaluation algorithm that evaluates circuits in NC^1 with moderate blowup in the norm of $\mathbf{R}_C$. Specifically, the observation is that any circuit with depth d can be simulated by a length 4^d and width 5 branching program by Barrington's theorem, recalled below:

Theorem 5 (Barrington's Theorem). *Every Boolean NAND circuit C that acts on ℓ inputs and has depth d can be computed by a width 5 permutation branching program of length 4^d. Given the description of the circuit C, the description of the corresponding branching program can be computed in $\mathsf{poly}(\ell, 4^d)$ time.*

Such a branching program can then be computed by multiplying 4^d many 5×5 permutation matrices. It is shown in [BV14] that homomorphically evaluating the multiplication of permutation matrices using the above procedure and the asymmetric noise growth feature only increases the noise by a polynomial factor, and thus allows us to use an LWE or SIS modulus that is polynomial

in the security parameter. In our constructions, we will use this particular evaluation method just as in [BV14] and denote it by Eval_{BV}.

We will use a claim regarding the noise growth properties of Eval_{BV}. It can be obtained from Claim 3.4.2 and Lemma 3.6 of [BV14] and Barrington's Theorem.

Lemma 14. *Let $C : \{0,1\}^\ell \to \{0,1\}$ be a* NAND *Boolean circuit. Let $\{\mathbf{A}_i = \mathbf{A}\mathbf{R}_i + x_i\mathbf{G} \in \mathbb{Z}_q^{n\times m}\}_{i\in[\ell]}$ be ℓ distinct matrices corresponding to the input wires of C, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{R}_i \xleftarrow{\$} \{-1,1\}^{m\times m}$, $x_i \in \{0,1\}$ and $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ is the gadget matrix. There is an efficient deterministic algorithm Eval_{BV} that takes as input C and $\{\mathbf{A}_i\}_{i\in[\ell]}$ and outputs a matrix $\mathbf{A}_C = \mathbf{A}\mathbf{R}_C + C(x_1,\ldots,x_\ell)\mathbf{G} = \mathsf{Eval}_{BV}(C,\mathbf{A}_1,\ldots,\mathbf{A}_\ell)$ where $\mathbf{R}_C \in \mathbb{Z}^{m\times m}$ and $C(x_1,\ldots,x_\ell)$ is the output of C on the arguments $x_1,\ldots,x_\ell$. Eval_{BV} runs in time $\mathsf{poly}(4^d,\ell,n,\log q)$. Let $||\mathbf{R}_{max}||_2 = max\{||\mathbf{R}_i||_2\}_{i\in[\ell]}$, the norm of $\mathbf{R}_C$ in $\mathbf{A}_C$ output by Eval_{BV} can be bounded with overwhelming probability by*

$$\begin{aligned} ||\mathbf{R}_C||_2 &\leq O(L\cdot||\mathbf{R}_{max}||_2.m) \\ &\leq O(L\cdot 12\sqrt{2m}\cdot m) \\ &\leq O(4^d m^{3/2}) \end{aligned} \tag{12}$$

where L is the length of the width 5 branching program which simulates C and we have used Lem. 13 to obtain $||\mathbf{R}_i||_2 \leq 12\sqrt{2m}$ for each i with overwhelming probability. In particular, if C is in NC^1 *and has depth $d = c\log l$ for a constant c, then $L = 4^d = \ell^{2c}$ and $\leq O(\ell^{2c}\cdot m^{3/2})$*

B One-More Unforeagibility vs PQ-EUF for Ring Signatures

The ring-signature analog of the one-more unforgeability by Boneh and Zhandry [BZ13b], *when restricted to the classical setting*, seems to be weaker than the standard unforgeability in Def. 4.[12] That is, in the classical setting, any RS satisfying the unforgeability in Def. 4 is also one-more unforgeable; but the reverse direction is unclear. We provide discussion in the following.

To argue that one-more unforgeability is no weaker than Def. 4, one needs to show how to convert a forger $\mathcal{A}_{\text{EUF}}$ winning in Expr. 3 to another forger $\mathcal{A}_{\text{OM}}$ winning in the (classical version of) "one-more forgery" game. Conceivably, $\mathcal{A}_{\text{OM}}$ will run $\mathcal{A}_{\text{EUF}}$ internally; thus, $\mathcal{A}_{\text{OM}}$ will make no less sign queries than $\mathcal{A}_{\text{EUF}}$. Recall that $\mathcal{A}_{\text{OM}}$ needs to forge one more signature than the total number of its queries. Also, crucially, all the ring signatures presented by $\mathcal{A}_{\text{OM}}$ at the end must have *no* corrupted members in the accompanying ring. Now ideally one might imagine that we can simply use the queries made by $\mathcal{A}_{\text{OM}}$ (which are really queries by $\mathcal{A}_{\text{EUF}}$) to meet the "one-more" challenge; however, this is thwarted immediately due to the fact that $\mathcal{A}_{\text{EUF}}$ has absolutely no obligation to make queries meeting this requirement, so even if the final forgery produced by $\mathcal{A}_{\text{EUF}}$ is valid, our attempted reduction does not have any means to provide $\mathcal{A}_{\text{OM}}$ with all the signatures it needs to win the "one-more" challenge (since not all of the queries can be reused). Indeed, it is not hard to find attacks that use this definitional gap to violate standard unforgeability, while being ruled out as a valid attack against one-more ring unforgeability. Contrast this with a comparison in the other direction: an adversary $\mathcal{A}_{\text{OM}}$ for the one-more unforgeability experiment is easily converted into a standard $\mathcal{A}_{\text{EUF}}$ adversary since not all of the signatures output by $\mathcal{A}_{\text{OM}}$ at the end can be previous queries (by the pigeonhole principle); $\mathcal{A}_{\text{EUF}}$ simple outputs the one that is not.

[12] This is in contrast to the case of ordinary signatures, where one-more unforgeability is equivalent to the standard existential unforgeability [BZ13b].

We remark however that this definitional gap between standard ring signature unforgeability and the "one-more" version may not be inherent; rather, we just do not know how to meet this gap. Our arguments here should not be interpreted as a proof showing that the former notion is strictly stronger than the latter. We leave it as an open question to either demonstrate a separation, or prove that the two are actually equivalent.