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 · Quantum · Ring Signatures

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

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

literature (e.g., [64, 15, 65, 1]), 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., [17, 18, 67, 61, 5]). 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) [11] 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. [15], 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 [6, 62, 48, 33, 52, 32, 42].

One-More Unforgeability vs Bind Unforgeability. Starting from [65], 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., [31, 36]), 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 ad-

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

versary 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 [17, 18, 67, 61, 36, 59, 46, 43, 30, 5, 23, 28, 4, 44, 45, 9].

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 [18] 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 [57]). 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. [18] shows how to convert any PQ-EUF signatures to one-more unforgeable ones using a *chameleon hash function* [49]; it also proves that the PQ-EUF signature scheme by Gentry, Peikert, and Vaikuntanathan [38] (henceforth, GPV) is one-more unforgeable in the QROM, assuming the PRF in that construction is quantum secure (i.e., being a QPRF [65]).

As argued in [37, 5], 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. [5] 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. [5] 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, [5] show that Lamport's one-time signature [50] is BU in the QROM, assuming the OWF is modeled as a (quantum-accessible)

[5] [5] also claimed that blind-unforgeability implies one-more unforgeability. But their proof was flawed [29]. The relation between these two notions is an open problem.

random oracle. Later, [54] show that the Winternitz one-time signature [55] is BU in the QROM, assuming the underlying hash function is modeled as a (quantum-accessible) random oracle. To the best of our knowledge, they 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?*

Post-Quantum Secure Ring Signatures. In a *ring signature* scheme [58, 12], 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* [27], 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 [60, 56], and thus have received extensive attention (see, e.g., [26] and related work therein).

For ring signatures from *latticed-based* assumptions, there exist several constructions in the CROM [3, 51, 60, 10, 63, 34, 13, 53], but only two schemes are known in the plain model [21, 26]. The authors of [26] 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, [26] leave 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 [18] 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 [18] (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 [20]. This signature scheme is also short.
3. We present a new definition of post-quantum security for ring signatures, by extending blind-unforgeability from [5]. 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 [38] 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 *preimage 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 an 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 [18] then shows that these are also indistinguishable to adversaries making polynomially-many *quantum* queries.

So far, our proof is identical to that of [18], 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 [18]. 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, [18] prove a lemma ([18, 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.

Since our overall construction and proof for the QROM scheme is similar to that in [18], we provide this construction and the corresponding proof in the full version [25] due to space constraints.

2.2 BU Signatures in the Plain Model

We make use of the signature template introduced in [20], which in turn relies on key-homomorphic techniques as used in [22]. We will refer to their homomorphic evaluation procedure as $\mathsf{Eval}_{\mathrm{BV}}$. The scheme uses the 'left-right trapdoor' paradigm. Namely, the verification key contains a matrix $\mathbf{A}$ sampled with a 'trapdoor' basis $\mathbf{T}_{\mathbf{A}}$, 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 [20]). The corresponding signing key contains $\mathbf{T}_{\mathbf{A}}$. 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, $\mathbf{T}_\mathbf{A}$ allows the signer to produce a short vector for either $\mathbf{F}_{M,b}$.

To show unforgeability, one constructs a reduction that (i) replaces the left matrix with an SIS challenge (thus losing $\mathbf{T}_\mathbf{A}$), and (ii) 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 $\mathbf{A}_{\text{PRF,M}}$ 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 $\mathbf{T}_\mathbf{A}$, 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 $\mathbf{F}_\text{M} = [\mathbf{A} \mid \mathbf{A}' - \mathbf{A}_{\text{PRF,M}}]$ (where $\mathbf{A}'$ 'encodes 1'). In the reduction, after making changes just as before, we obtain that $\mathbf{F}_\text{M} = \big[\mathbf{A} \mid \mathbf{AR} - \big(1 - \mathsf{PRF}_k(M)\big)\mathbf{G}\big]$. 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 [8], 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: (1) by a *corruption query* $(\mathsf{corrupt}, i)$, $\mathcal{A}$ can corrupt a member in $\mathcal{R}$ to learn its secret key; (2) 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 $\mathcal{C}$ recording all the members in $\mathcal{R}$ that are corrupted by $\mathcal{A}$. To win the game, $\mathcal{A}$ needs to output a forgery $(\mathsf{R}^*, m^*, \Sigma^*)$ such that $\mathsf{R}^* \subseteq \mathcal{R} \setminus \mathcal{C}$, $\mathsf{RS.Verify}(\mathsf{R}^*, m^*, \Sigma^*) = 1$, and that $\mathcal{A}$ 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 maps 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 $(\mathsf{sign}, i, \sum \psi_{\mathsf{R},m} |\mathsf{R}, m\rangle)$, 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 [26] is to extend the one-more unforgeability from [18] 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. 5.1).

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

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. [26]. 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 (SPB) 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 [26] 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.

The [26] construction can thus in fact 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 [18]). Second, we want to ensure statistical similarity

of the components c_1, c_2, π (in order to use an aforementioned lemma from [18] which says that pointwise statistically close oracles are indistiguishable even with quantum queries). In particular, 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 [26] 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 [25, Section 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 $\overset{\text{c}}{\approx}$, $\overset{\text{s}}{\approx}$ and $\overset{\text{i.d.}}{=\!=}$ 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.

Quantum Oracle Indistinguishability. We will need the following lemmata.

Lemma 1 ([66]). *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 ([18]). *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 $\sqrt{8C_0q^3\varepsilon}$.*

Blind-Unforgeable Signatures. We recall in Def. 1 the definition for blind unforgeable signature schemes in [5]. 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 $\mathsf{PQAdv}^\lambda_{\mathrm{BU}}(\mathcal{A}) \coloneqq \Pr\left[\mathsf{PQExp}^\lambda_{\mathrm{BU}}(\mathcal{A}) = 1\right] \leq \mathsf{negl}(\lambda)$.*

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: (1) $m^* \in B^{\mathsf{Sig}}_\varepsilon$; (2) $\mathsf{Verify}(vk, m^*, \sigma^*) = 1$. 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 [18], 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. 4 and 5).

Remark 2. We let the adversary choose ε. This is equivalent to quantifying over all values of ε as in the definition in [5].

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

4 Blind-Unforgeable Signatures in the Plain Model

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. For standard lattice-related concepts (e.g., parameters, hardness, trapdoors), see the full version [25, Appendix A.1].

We denote the Gram-Schmidt ordered orthogonalization of a matrix $\mathbf{A} \in \mathbb{Z}^{m \times m}$ by $\widetilde{\mathbf{A}}$. 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 $\mathbf{R}$. Correspondingly, $||\mathbf{R}||_2$ denotes the operator norm of $\mathbf{R}$, defined as $||\mathbf{R}||_2 = \mathsf{sup}_{x \in \mathbb{R}^{m+1}} ||\mathbf{R}\mathbf{x}||$. 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 [2, 19] which uses two sampling algorithms SampleLeft and SampleRight. 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 1 (SampleLeft **Closeness [2, 24]**). *Let $q > 2$, $m > n$ and $s > ||\widetilde{\mathbf{T_A}}|| \cdot \omega(\sqrt{\log(m+m_1)})$. Then* SampleLeft$(\mathbf{A}, \mathbf{B}, \mathbf{T_A}, \mathbf{u}, s)$ *outputs $\mathbf{d} \in \mathbb{Z}^{m+m_1}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{F}),s}$.*

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 $\mathbf{F} = [\mathbf{A} \mid \mathbf{A}\mathbf{R} + \mathbf{B}]$. It outputs a vector $\mathbf{d} \in \mathbb{Z}^{m+m_1}$ in $\Lambda_q^{\mathbf{u}}(\mathbf{F})$.

Theorem 2 (SampleRight **Closeness [2]**). *Let $q > 2$, $m > n$ and $s > ||\widetilde{\mathbf{T_B}}|| \cdot \omega(\sqrt{\log m})$. Then* SampleRight$(\mathbf{A}, \mathbf{B}, \mathbf{R}, \mathbf{T_B}, \mathbf{u}, s)$ *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 [2, Lemma 4] (see the full version [25, Appendix A.2] for more 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 [39, 16, 22] (see the full version [25, Appendix A.3] for more 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 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 [8] is quantum-access secure (i.e., a QPRF) [65]. 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 [8] PRF relies on the quantum hardness of LWE with *super-polynomial* modulus.

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 $\mathsf{SampleLeft}$, $\mathsf{SampleRight}$ given as in Thm. 1 and Thm. 2 respectively, and $\mathsf{TrapGen}$ 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. The construction is as follows:

Construction 1: Blind-Unforgeable Signatures in the Plain Model

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 $\mathsf{TrapGen}$, and $\mathsf{sigsize}_\lambda = s\sqrt{2m}$ for the validity of $\mathsf{SampleLeft}$ output (to ensure completeness). Set $s = O(4^d m^{3/2})\omega(\sqrt{\log m})$ to ensure statistical closeness of $\mathsf{SampleLeft}$ and $\mathsf{SampleRight}$, and correspondingly set $\beta = O(16^d m^{7/2})\omega(\sqrt{\log m})$ and $q = O(16^d m^4)\big(\omega(\sqrt{\log m})\big)^2$ to have an overall reduction to an appropriately

hard instance of SIS. For further details about these choices, see the full version [25, Section 5.3].

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

1. Sample a matrix $\mathbf{A}$ along with a 'trapdoor' basis $\mathbf{T_A}$ for $\Lambda_q^\perp(\mathbf{A})$ using $\mathsf{TrapGen}$.
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 $\mathsf{C_{PRF}}$ 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 $sk = \mathbf{T_A}$.

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

1. Run the [22] 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 $\mathbf{F}_{\mathrm{M}} \coloneqq [\mathbf{A} \mid \mathbf{A}' - \mathbf{A}_{\mathrm{PRF,M}}]$; Use $\mathsf{SampleLeft}$ to obtain $\mathbf{d}_{\mathrm{M}} \leftarrow \mathcal{D}_{\Lambda_q^\perp(\mathbf{F}_{\mathrm{M}}), s}$.
3. 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.

Proof of Security. Completeness follows straightforwardly from the correctness of $\mathsf{SampleLeft}$ (Thm. 1) for $\mathcal{D}_{\Lambda_q^\perp(\mathbf{F}), s}$. In the following, we prove BU-security.

Theorem 3. *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. 5.*

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: $\mathbf{R}'_{\mathbf{A}}, \{\mathbf{R}_{\mathbf{B}_i}\}_{i=1}^{k}, \mathbf{R}_{\mathbf{C}0}, \mathbf{R}_{\mathbf{C}1} \xleftarrow{\$} \{-1,1\}^{m\times m}$.
3. Let PRF and $\mathsf{C}_{\mathsf{PRF}}$ 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 $\widetilde{\mathbf{T}}_{\mathbf{G}}$.
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 $vk = (\mathbf{A}, \mathbf{A}', \{\mathbf{B}_i\}_{i=1}^{k}, \{\mathbf{C}_0, \mathbf{C}_1\}, s, \mathsf{PRF}, \mathbf{C}_{\mathsf{PRF}})$, and $sk = (\mathbf{T}_{\mathbf{A}}, k_\varepsilon)$.

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

$H_0 \overset{s}{\approx} H_1$*:* 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.

$H_1 \overset{c}{\approx} H_2$*:* 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.)

$H_2 \overset{s}{\approx} H_3$*:* 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. 1 and 2 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 $\mathbb{Z}_q^{n\times m}$.

$\underline{H_4 \overset{s}{\approx} H_5}$*:* This follows immediately from the property of the lattice trapdoor algorithm $\mathsf{TrapGen}$.

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 [20, Theorem 3.1]. Due to space constraints, we present it in the full version [25, Section 5.4]. □

5 Post-Quantum Ring Signatures

5.1 Definitions

Classical Ring Signatures. We start by recalling the classical definition of ring signatures [12, 7].

Definition 3 (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*[8] N *(e.g.,* $N = 2^{\log^2 \lambda}$*) specifying the maximum number of members in a ring, output a verification and signing key pair* $(\mathsf{VK}, \mathsf{SK})$*.*
- $\mathsf{Sign}(\mathsf{SK}, \mathsf{R}, m)$**:** *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 $\mathcal{A}$.

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

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: (1) $\mathsf{R}^* \subseteq \mathcal{VK} \setminus \mathcal{C}$; (2) $\mathcal{A}$ never made a signing query of the form $(\mathsf{sign}, \cdot, \mathsf{R}^*, m^*)$; (3) $\mathsf{Verify}(\mathsf{R}^*, m^*, \Sigma^*) = 1$. If so, the experiment outputs 1; otherwise, 0.

We mention that the unforgeability and anonymity properties defined in Definition 3 correspond respectively to the notions of *unforgeability with insider corruption* and *anonymity with respect to full key exposure* presented in [12].

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 [26]. But their definition has certain restrictions (discussed below). In the following, we develop a new definition building on ideas from [26].

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[9]:

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

However, as observed in [26], this will lead to an unsatisfiable definition due to an attack from [18]. 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 [26] 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.

[9] 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.

Definition 4 (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. 3.* 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*

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

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, [26] 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, [26] adapts one-more unforgeability [18] 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 $\mathcal{VK}\backslash\mathcal{C}$. This definition, *when restricted to the classical setting*, seems to be weaker than the standard unforgeability in Def. 3.That is, in the classical setting, any RS satisfying the unforgeability in Def. 3 is also one-more unforgeable; but the reverse direction is unclear (we provide more discussion in [25, Appendix B]). Instead, our definition extends the blind-unforgeability for ordinary signatures (Def. 1) to the ring setting. We present this version in Def. 5. In contrast to the "one-more" unforgeability, we will show in Lem. 3 that, when restricted to the classical setting, this blind-unforgeability for ring signatures is indeed equivalent to the standard existential unforgeability in Def. 3. Its proof is almost identical to [5, Proposition 2]. Due to space constraints, we put it in [25, Section 6.1.2].

Definition 5 (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. 3. 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*

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

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

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

   $f(\mathsf{R}, m) \coloneqq \begin{cases} \mathsf{RS.Sign}(\mathsf{SK}_i, m, \mathsf{R}; r) & \text{if } \mathsf{VK}_i \in \mathsf{R} \\ \perp & \text{otherwise} \end{cases}$.
5. Finally, $\mathcal{A}$ outputs $(\mathsf{R}^*, m^*, \Sigma^*)$. The challenger checks if: (1) $\mathsf{R}^* \subseteq \mathcal{VK} \setminus \mathcal{C}$; (2) $\mathsf{Verify}(\mathsf{R}^*, m^*, \Sigma^*) = 1$; (3) $(\mathsf{R}^*, m^*) \in B_\varepsilon^{\mathsf{RS}}$. If so, it outputs 1; otherwise, 0.

Lemma 3. *Restricted to (classical) QPT adversaries, a ring signature* RS *scheme is blind-unforgeable (Def. 5) if and only if it satisfies the unforgeability requirement in Def. 3.*

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

Definition 6 (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. 3. Moreover, they also satisfy the* completeness *requirement in Def. 3, the* post-quantum anonymity *in Def. 4, and the* post-quantum blind-unforgeability *as in Def. 5.*

5.2 Building Blocks

Lossy PKEs with Special Properties. We need the following lossy PKE.

Definition 7 (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 [41]. 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.

ZAPs for Super-Complement Languages. As mentioned in Sec. 2.3, [26] 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 [26] 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 prove 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 the language the captures some *necessary conditions* for any valid forgery. 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 8 (Super-Complement [26]). *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 $\widetilde{L} \subseteq (\{0,1\}^* \setminus L) \times \{0,1\}^*$. 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. 8—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 [26] captures the general $(L, \widetilde{L})$ pairs defined in Def. 8. Since we only need the special case in Rmk. 3, we will define the ZAP only for this case.

Definition 9 (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*

$$\Pr\left[\rho \leftarrow \mathsf{V}(1^\lambda, 1^{|x|}, 1^{\widetilde{\ell}}, 1^{\widetilde{D}}); \pi \leftarrow \mathsf{P}(\rho, x, w) : \mathsf{Verify}(\rho, x, \pi) = 1\right] = 1.$$

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* $\widetilde{w} \in \{0,1\}^*$*,*

$$\Pr\left[\begin{array}{l}\rho \leftarrow \mathsf{V}(1^\lambda, 1^n, 1^{|\widetilde{w}|}, 1^{\widetilde{D}}); \\ (x, \pi^*) \leftarrow P^*_\lambda(\rho)\end{array} : \begin{array}{l}\mathsf{Verify}(\rho, x, \pi^*) = 1 \ \wedge \\ \widetilde{D} \geq \widetilde{d}(|x|, |\widetilde{w}|) \ \wedge \ (x, \widetilde{w}) \in \widetilde{R}\end{array}\right] \leq \mathsf{negl}(\lambda).$$

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*

$$\Pr[V^*_2(\rho, x, \pi, \zeta) = 1 \ \wedge \ (x, w) \in \mathcal{R} \ \wedge \ (x, w') \in \mathcal{R}]$$

in the following two experiments differ only by $\mathsf{negl}(\lambda)$*:*
   - *in experiment 1,* $(\rho, x, w, w', \zeta) \leftarrow V^*_1(1^\lambda, 1^n, 1^{\widetilde{\ell}}, 1^{\widetilde{D}}), \pi \leftarrow \mathsf{P}(\rho, x, w)$*;*
   - *in 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 4 ([26]). *Assuming QLWE, there exist ZAPs as per Def. 9 for any super-complement language as per Def. 8.*

5.3 Construction

Our construction, shown in Constr. 2, relies on the following building blocks: (1) pair-wise independent functions; a Sig satisfying Def. 1; a LE satisfying Def. 7; a ZAP satisfying Def. 9.

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. 5.4. This arrangement is because we find that the language $(L, \widetilde{L})$ becomes easier to understand once the reader has slight familiarity with Constr. 2.

Construction 2: 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 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 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. 5.4. 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 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 $\mathsf{ZAP.Verify}(\rho_1, x, \pi)$.

5.4 The Super-Complement Language Proven by the ZAP

We now define the super-complement language $(L, \widetilde{L})$ used in Constr. 2. This deviates from the $(L, \widetilde{L})$ defined in [26, Section 5], to accommodate Constr. 2.

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:

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

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

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

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:

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

Following a similar proof as for [26, Lemma 5.1], we can show that $\widetilde{L}$ is indeed a super-complement of L. (The full proof is provided in [25, Section 6.3.1].)

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

5.5 Proof of Security

The security of Constr. 2 can be established following the idea illustrated in Sec. 2.3. Due to space constraints, we refer the reader to [25, Section 6.4] for the formal security proof.

6 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.

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

1. Agarwal, A., Bartusek, J., Goyal, V., Khurana, D., Malavolta, G.: Post-quantum multi-party computation. In: EUROCRYPT 2021. pp. 435–464. Springer 2
2. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert [40], pp. 553–572. https://doi.org/10.1007/978-3-642-13190-5_28 12
3. Aguilar Melchor, C., Bettaieb, S., Boyen, X., Fousse, L., Gaborit, P.: Adapting Lyubashevsky's signature schemes to the ring signature setting. In: Youssef, A., Nitaj, A., Hassanien, A.E. (eds.) AFRICACRYPT 13. LNCS, vol. 7918, pp. 1–25. Springer, Heidelberg (Jun 2013). https://doi.org/10.1007/978-3-642-38553-7_1 4
4. Alagic, G., Brakerski, Z., Dulek, Y., Schaffner, C.: Impossibility of quantum virtual black-box obfuscation of classical circuits. In: CRYPTO 2021. Springer 3
5. Alagic, G., Majenz, C., Russell, A., Song, F.: Quantum-access-secure message authentication via blind-unforgeability. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part III. LNCS, vol. 12107, pp. 788–817. Springer, Heidelberg (May 2020). https://doi.org/10.1007/978-3-030-45727-3_27 2, 3, 5, 10, 11, 19
6. Ambainis, A., Rosmanis, A., Unruh, D.: Quantum attacks on classical proof systems: The hardness of quantum rewinding. In: 55th FOCS. pp. 474–483. IEEE Computer Society Press (Oct 2014). https://doi.org/10.1109/FOCS.2014.57 2
7. Backes, M., Döttling, N., Hanzlik, L., Kluczniak, K., Schneider, J.: Ring signatures: Logarithmic-size, no setup - from standard assumptions. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part III. LNCS, vol. 11478, pp. 281–311. Springer, Heidelberg (May 2019). https://doi.org/10.1007/978-3-030-17659-4_10 17
8. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 719–737. Springer, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_42 7, 13
9. Bartusek, J., Malavolta, G.: Indistinguishability obfuscation of null quantum circuits and applications. Cryptology ePrint Archive, Report 2021/421 (2021), https://ia.cr/2021/421 3
10. Baum, C., Lin, H., Oechsner, S.: Towards practical lattice-based one-time linkable ring signatures. In: The 2018 International Conference on Information and Communications Security. Springer 4
11. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM CCS 93. pp. 62–73. ACM Press (Nov 1993). https://doi.org/10.1145/168588.168596 2
12. Bender, A., Katz, J., Morselli, R.: Ring signatures: Stronger definitions, and constructions without random oracles. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 60–79. Springer, Heidelberg (Mar 2006). https://doi.org/10.1007/11681878_4 4, 17, 18
13. Beullens, W., Katsumata, S., Pintore, F.: Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 464–492. Springer, Heidelberg (Dec 2020). https://doi.org/10.1007/978-3-030-64834-3_16 4
14. Boldyreva, A., Micciancio, D. (eds.): CRYPTO 2019, Part II, LNCS, vol. 11693. Springer, Heidelberg (Aug 2019) 28, 29
15. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 41–69. Springer, Heidelberg (Dec 2011). https://doi.org/10.1007/978-3-642-25385-0_3 2

16. Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 533–556. Springer, Heidelberg (May 2014). https://doi.org/10.1007/978-3-642-55220-5_30 13
17. Boneh, D., Zhandry, M.: Quantum-secure message authentication codes. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 592–608. Springer, Heidelberg (May 2013). https://doi.org/10.1007/978-3-642-38348-9_35 2, 3
18. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 361–379. Springer, Heidelberg (Aug 2013). https://doi.org/10.1007/978-3-642-40084-1_21 2, 3, 5, 6, 8, 9, 10, 11, 18, 19
19. Boyen, X.: Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 499–517. Springer, Heidelberg (May 2010). https://doi.org/10.1007/978-3-642-13013-7_29 12
20. Boyen, X., Li, Q.: Towards tightly secure lattice short signature and id-based encryption. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 404–434. Springer, Heidelberg (Dec 2016). https://doi.org/10.1007/978-3-662-53890-6_14 5, 6, 16
21. Brakerski, Z., Kalai, Y.T.: A framework for efficient signatures, ring signatures and identity based encryption in the standard model. IACR Cryptol. ePrint Arch. p. 86 (2010), http://eprint.iacr.org/2010/086 4
22. Brakerski, Z., Vaikuntanathan, V.: Lattice-based FHE as secure as PKE. In: Naor, M. (ed.) ITCS 2014. pp. 1–12. ACM (Jan 2014). https://doi.org/10.1145/2554797.2554799 6, 13, 14
23. Carstens, T.V., Ebrahimi, E., Tabia, G.N., Unruh, D.: On quantum indistinguishability under chosen plaintext attack. IACR Cryptol. ePrint Arch. p. 596 (2020), https://eprint.iacr.org/2020/596 3
24. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert [40], pp. 523–552. https://doi.org/10.1007/978-3-642-13190-5_27 12
25. Chatterjee, R., Chung, K.M., Liang, X., Malavolta, G.: A note on the post-quantum security of (ring) signatures. arXiv preprint arXiv:2112.06078 (2021) 6, 10, 12, 13, 14, 16, 19, 25
26. Chatterjee, R., Garg, S., Hajiabadi, M., Khurana, D., Liang, X., Malavolta, G., Pandey, O., Shiehian, S.: Compact ring signatures from learning with errors. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12825, pp. 282–312. Springer (2021). https://doi.org/10.1007/978-3-030-84242-0_11, https://doi.org/10.1007/978-3-030-84242-0_11 4, 8, 9, 10, 18, 19, 22, 23, 24, 25
27. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT'91. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (Apr 1991). https://doi.org/10.1007/3-540-46416-6_22 4
28. Chevalier, C., Ebrahimi, E., Vu, Q.H.: On the security notions for encryption in a quantum world. IACR Cryptol. ePrint Arch. p. 237 (2020), https://eprint.iacr.org/2020/237 3

29. Communication, P.: Personal communication with the authors of [amrs20] (2021) 3
30. Czajkowski, J., Hülsing, A., Schaffner, C.: Quantum indistinguishability of random sponges. In: Boldyreva and Micciancio [14], pp. 296–325. https://doi.org/10.1007/978-3-030-26951-7_11 3
31. Damgård, I., Funder, J., Nielsen, J.B., Salvail, L.: Superposition attacks on cryptographic protocols. In: International Conference on Information Theoretic Security. pp. 142–161. Springer (2013) 2
32. Don, J., Fehr, S., Majenz, C.: The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 602–631. Springer, Heidelberg (Aug 2020). https://doi.org/10.1007/978-3-030-56877-1_21 2
33. Don, J., Fehr, S., Majenz, C., Schaffner, C.: Security of the Fiat-Shamir transformation in the quantum random-oracle model. In: Boldyreva and Micciancio [14], pp. 356–383. https://doi.org/10.1007/978-3-030-26951-7_13 2
34. Esgin, M.F., Zhao, R.K., Steinfeld, R., Liu, J.K., Liu, D.: MatRiCT: Efficient, scalable and post-quantum blockchain confidential transactions protocol. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019. pp. 567–584. ACM Press (Nov 2019). https://doi.org/10.1145/3319535.3354200 4
35. Fischlin, M., Coron, J.S. (eds.): EUROCRYPT 2016, Part II, LNCS, vol. 9666. Springer, Heidelberg (May 2016) 29, 30
36. Gagliardoni, T., Hülsing, A., Schaffner, C.: Semantic security and indistinguishability in the quantum world. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part III. LNCS, vol. 9816, pp. 60–89. Springer, Heidelberg (Aug 2016). https://doi.org/10.1007/978-3-662-53015-3_3 2, 3
37. Garg, S., Yuen, H., Zhandry, M.: New security notions and feasibility results for authentication of quantum data. In: Katz and Shacham [47], pp. 342–371. https://doi.org/10.1007/978-3-319-63715-0_12 3
38. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 197–206. ACM Press (May 2008). https://doi.org/10.1145/1374376.1374407 3, 5
39. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (Aug 2013). https://doi.org/10.1007/978-3-642-40041-4_5 13
40. Gilbert, H. (ed.): EUROCRYPT 2010, LNCS, vol. 6110. Springer, Heidelberg (May / Jun 2010) 26, 27
41. Gorbunov, S., Vaikuntanathan, V., Wichs, D.: Leveled fully homomorphic signatures from standard lattices. In: Servedio, R.A., Rubinfeld, R. (eds.) 47th ACM STOC. pp. 469–477. ACM Press (Jun 2015). https://doi.org/10.1145/2746539.2746576 21
42. Grilo, A.B., Hövelmanns, K., Hülsing, A., Majenz, C.: Tight adaptive reprogramming in the QROM. Cryptology ePrint Archive, Report 2020/1361 (2020), https://eprint.iacr.org/2020/1361 2
43. Hosoyamada, A., Iwata, T.: 4-round Luby-Rackoff construction is a qPRP. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part I. LNCS, vol. 11921, pp. 145–174. Springer, Heidelberg (Dec 2019). https://doi.org/10.1007/978-3-030-34578-5_6 3

44. Hosoyamada, A., Iwata, T.: On tight quantum security of hmac and nmac in the quantum random oracle model. In: Annual International Cryptology Conference. pp. 585–615. Springer (2021) 3
45. Hosoyamada, A., Sasaki, Y.: Quantum collision attacks on reduced SHA-256 and SHA-512. In: Annual International Cryptology Conference. pp. 616–646. Springer (2021) 3
46. Hosoyamada, A., Yasuda, K.: Building quantum-one-way functions from block ciphers: Davies-Meyer and Merkle-Damgård constructions. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part I. LNCS, vol. 11272, pp. 275–304. Springer, Heidelberg (Dec 2018). https://doi.org/10.1007/978-3-030-03326-2_10 3
47. Katz, J., Shacham, H. (eds.): CRYPTO 2017, Part II, LNCS, vol. 10402. Springer, Heidelberg (Aug 2017) 28, 29
48. Kiltz, E., Lyubashevsky, V., Schaffner, C.: A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part III. LNCS, vol. 10822, pp. 552–586. Springer, Heidelberg (Apr / May 2018). https://doi.org/10.1007/978-3-319-78372-7_18 2
49. Krawczyk, H., Rabin, T.: Chameleon signatures. In: Proceedings of the Network and Distributed System Security Symposium, NDSS 2000, San Diego, California, USA. The Internet Society (2000), https://www.ndss-symposium.org/ndss2000/chameleon-signatures/ 3
50. Lamport, L.: Constructing digital signatures from a one-way function. Tech. rep., Citeseer (1979) 3
51. Libert, B., Ling, S., Nguyen, K., Wang, H.: Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In: Fischlin and Coron [35], pp. 1–31. https://doi.org/10.1007/978-3-662-49896-5_1 4
52. Liu, Q., Zhandry, M.: Revisiting post-quantum Fiat-Shamir. In: Boldyreva and Micciancio [14], pp. 326–355. https://doi.org/10.1007/978-3-030-26951-7_12 2
53. Lyubashevsky, V., Nguyen, N.K., Seiler, G.: Smile: set membership from ideal lattices with applications to ring signatures and confidential transactions. In: Annual International Cryptology Conference. pp. 611–640. Springer (2021) 4
54. Majenz, C., Manfouo, C.M., Ozols, M.: Quantum-access security of the winternitz one-time signature scheme. arXiv preprint arXiv:2103.12448 (2021) 4
55. Merkle, R.C.: A certified digital signature. In: Brassard, G. (ed.) CRYPTO'89. LNCS, vol. 435, pp. 218–238. Springer, Heidelberg (Aug 1990). https://doi.org/10.1007/0-387-34805-0_21 4
56. Noether, S.: Ring signature confidential transactions for monero. Cryptology ePrint Archive, Report 2015/1098 (2015), https://eprint.iacr.org/2015/1098 4
57. Pointcheval, D., Stern, J.: Provably secure blind signature schemes. In: Kim, K., Matsumoto, T. (eds.) ASIACRYPT'96. LNCS, vol. 1163, pp. 252–265. Springer, Heidelberg (Nov 1996). https://doi.org/10.1007/BFb0034852 3
58. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (Dec 2001). https://doi.org/10.1007/3-540-45682-1_32 4
59. Song, F., Yun, A.: Quantum security of NMAC and related constructions - PRF domain extension against quantum attacks. In: Katz and Shacham [47], pp. 283–309. https://doi.org/10.1007/978-3-319-63715-0_10 3
60. Torres, W.A.A., Steinfeld, R., Sakzad, A., Liu, J.K., Kuchta, V., Bhattacharjee, N., Au, M.H., Cheng, J.: Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (lattice ringct v1. 0). In: Aus-

tralasian Conference on Information Security and Privacy. pp. 558–576. Springer (2018) 4
61. Unruh, D.: Computationally binding quantum commitments. In: Fischlin and Coron [35], pp. 497–527. https://doi.org/10.1007/978-3-662-49896-5_18 2, 3
62. Unruh, D.: Post-quantum security of Fiat-Shamir. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 65–95. Springer, Heidelberg (Dec 2017). https://doi.org/10.1007/978-3-319-70694-8_3 2
63. Wang, S., Zhao, R., Zhang, Y.: Lattice-based ring signature scheme under the random oracle model. Int. J. High Perform. Comput. Netw. **11**(4), 332–341 (2018). https://doi.org/10.1504/IJHPCN.2018.10014445, https://doi.org/10.1504/IJHPCN.2018.10014445 4
64. Watrous, J.: Zero-knowledge against quantum attacks. In: Kleinberg, J.M. (ed.) 38th ACM STOC. pp. 296–305. ACM Press (May 2006). https://doi.org/10.1145/1132516.1132560 2
65. Zhandry, M.: How to construct quantum random functions. In: 53rd FOCS. pp. 679–687. IEEE Computer Society Press (Oct 2012). https://doi.org/10.1109/FOCS.2012.37 2, 3, 13
66. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 758–775. Springer, Heidelberg (Aug 2012). https://doi.org/10.1007/978-3-642-32009-5_44 10
67. Zhandry, M.: A note on the quantum collision and set equality problems. Quantum Inf. Comput. **15**(7&8), 557–567 (2015). https://doi.org/10.26421/QIC15.7-8-2, https://doi.org/10.26421/QIC15.7-8-2 2, 3