

# Revocable Quantum Digital Signatures

Tomoyuki Morimae<sup>1</sup>, Alexander Poremba<sup>2,3</sup>, Takashi Yamakawa<sup>1,4,5</sup>

<sup>1</sup>Yukawa Institute for Theoretical Physics, Kyoto University, Kyoto, Japan  
tomoyuki.morimae@yukawa.kyoto-u.ac.jp

<sup>2</sup>Computing and Mathematical Sciences, Caltech, Pasadena, CA, USA

<sup>3</sup>CSAIL and Department of Mathematics, MIT, Cambridge, MA, USA  
poremba@mit.edu

<sup>4</sup>NTT Social Informatics Laboratories, Tokyo, Japan  
takashi.yamakawa@ntt.com

<sup>5</sup>NTT Research Center for Theoretical Quantum Information, Atsugi, Japan

## Abstract

We study digital signatures with *revocation capabilities* and show two results. First, we define and construct digital signatures with revocable signing keys from the LWE assumption. In this primitive, the signing key is a quantum state which enables a user to sign many messages and yet, the quantum key is also *revocable*, i.e., it can be collapsed into a classical certificate which can later be verified. Once the key is successfully revoked, we require that the initial recipient of the key loses the ability to sign. We construct digital signatures with revocable signing keys from a newly introduced primitive which we call *two-tier one-shot signatures*, which may be of independent interest. This is a variant of one-shot signatures, where the verification of a signature for the message “0” is done publicly, whereas the verification for the message “1” is done in private. We give a construction of two-tier one-shot signatures from the LWE assumption. As a complementary result, we also construct digital signatures with *quantum* revocation from group actions, where the quantum signing key is simply “returned” and then verified as part of revocation.

Second, we define and construct digital signatures with revocable signatures from OWFs. In this primitive, the signer can produce quantum signatures which can later be revoked. Here, the security property requires that, once revocation is successful, the initial recipient of the signature loses the ability to find accepting inputs to the signature verification algorithm. We construct this primitive using a newly introduced *two-tier* variant of tokenized signatures. For the construction, we show a new lemma which we call the adaptive hardcore bit property for OWFs, which may enable further applications.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Background	3
1.2	Our Results	4
1.3	Comparison with Existing Works	4
1.4	Technical Overview	5
1.5	Related Works	7
1.6	Organization	7
<b>2</b>	<b>Preliminaries</b>	<b>8</b>
2.1	Basic Notation	8
2.2	Cryptography	8
2.3	Noisy Trapdoor Claw-Free Hash Function Family	9
<b>3</b>	<b>Two-tier One-shot Signatures</b>	<b>11</b>
3.1	Definition	11
3.2	Construction	12
<b>4</b>	<b>Digital Signatures with Revocable Signing Keys</b>	<b>12</b>
4.1	Definition	13
4.2	One-Time Construction for Single-Bit Messages	14
4.3	From Single-Bit to Multi-Bit Messages	17
4.4	From One-Time Schemes to Many-Time Schemes	19
<b>5</b>	<b>Adaptive Hardcore Bit Property for OWFs</b>	<b>23</b>
5.1	Statements	23
5.2	Theorem of [BKM <sup>+</sup> 23]	23
5.3	Proof of Theorem 5.1	25
5.4	Proof of Theorem 5.2	26
<b>6</b>	<b>Two-Tier Tokenized Signatures</b>	<b>27</b>
6.1	Definition	27
6.2	Construction	29
<b>7</b>	<b>Digital Signatures with Revocable Signatures</b>	<b>29</b>
7.1	Definition	29
7.2	Construction	32
<b>A</b>	<b>Proof of Theorem 5.4</b>	<b>37</b>
<b>B</b>	<b>Digital Signatures with Quantum Revocable Signing Keys from Group Actions</b>	<b>38</b>
B.1	Group Actions	38
B.2	Swap-Trapdoor Functions Pairs	39
B.3	Definition of Two-Tier Quantum Lightning with Classical Semi-Verification	40
B.4	Construction from Group Actions	41
B.5	Construction of Digital Signatures with Quantum Revocable Signing Keys	45

# 1 Introduction

## 1.1 Background

The exotic nature of quantum physics, such as quantum superposition, no-cloning, entanglement, and uncertainty relations, enables many new cryptographic applications which are impossible in a classical world. These include quantum money [Wie83], copy-protection [Aar09, ALL<sup>+</sup>21], secure software leasing [AL21], unclonable encryption [Got02, BL20], certified deletion [BI20], and more. Here, a common approach is to encode information into a quantum state which prevents it from being copied by the no-cloning principle.

Following this line of research, Ananth, Poremba, and Vaikuntanathan [APV23] and Agrawal, Kitagawa, Nishimaki, Yamada, and Yamakawa [AKN<sup>+</sup>23] concurrently introduced the concept of key-revocable public key encryption (PKE),<sup>1</sup> which realizes the following functionality: a decryption capability is delegated to a user in the form of a quantum decryption key in such a way that, once the key is returned, the user loses the ability to decrypt. They constructed key-revocable PKE schemes based on standard assumptions, namely quantum hardness of the learning with errors problem (LWE assumption) [APV23] or even the mere existence of any PKE scheme [AKN<sup>+</sup>23]. They also extended the idea of revocable cryptography to pseudorandom functions [APV23] and encryption with advanced functionality such as attribute-based encryption and functional encryption [AKN<sup>+</sup>23]. However, neither of these works extended the idea to *digital signatures* despite their great importance in cryptography. This state of affairs raises the following question:

*Is it possible to construct digital signature schemes with revocation capabilities?*

The delegation of privileges is of central importance in cryptography, and the task of revoking privileges in the context of digital signatures and certificates, in particular, remains a fundamental challenge for cryptography [Stu95, Riv98]. One simple solution is to use a limited-time delegatable signature scheme, where a certified signing key is generated together with an expiration date. Note that this requires that the expiration date is known ahead of time and that the clocks be synchronized. Moreover, issuing new keys (for example, each day) could potentially also be costly. Quantum digital signature schemes with revocation capabilities could potentially resolve these difficulties by leveraging the power of quantum information.

To illustrate the use of *revocable* digital signature schemes, consider the following scenarios. Suppose that an employee at a company, say Alice, takes a temporary leave of absence and wishes to authorize her colleague, say Bob, to sign a few important documents on her behalf. One thing Alice can do is to simply use a (classical) digital signature scheme and to share her signing keys with Bob. While this naïve approach would certainly allow Bob to produce valid signatures while Alice is gone, it also means that Bob continues to have access to the signing keys—long after Alice’s return. This is because the signing key of a digital signature scheme is *classical*, and hence it can be copied at will. In particular, a malicious Bob could secretly sell Alice’s signing key to a third party for a profit. A digital signature scheme with *revocable signing keys* can remedy this situation as it enables Alice to certify that Bob has lost access to the signing key once and for all.

As a second example, consider the following scenario. Suppose that a company or a governmental organization wishes to grant a new employee certain access privileges throughout their employment; for example to various buildings or office spaces. One solution is to use an *electronic* ID card through a mobile device, where a digital signature is used for identity management. Naturally, one would like to ensure that, once the employee’s contract is terminated, their ID card is disabled in the system and no longer allows for further unauthorized access. However, if the signature corresponding to the employee’s ID is a digital object, it is conceivable that the owner of the card manages to retain their ID card even after it is disabled. This threat

---

<sup>1</sup>Agrawal et al. [AKN<sup>+</sup>23] call it PKE with secure key leasing.

especially concerns scenarios in which the verification of an ID card is performed by a device which is not connected to the internet, or simply not updated frequently enough. A digital signature scheme with revocable signatures can remedy this situation as it enables *revocable quantum* ID cards; in particular, it allows one to certify that the initial access privileges have been revoked once and for all.

## 1.2 Our Results

In this paper, we show the following two results on revocable digital signatures.

**Revocable signing keys.** First, we define digital signatures with revocable *signing keys* (DSR-Key). In this primitive, a signing key is encoded in the form of a quantum state which enables the recipient to sign many messages. However, once the key is successfully revoked from a user, they no longer have the ability to generate valid signatures. Here, we consider *classical revocation*, i.e., a classical certificate is issued once the user destroys the quantum signing key with an appropriate measurement. In addition, the verification of the revocation certificate takes place in private, which means that the verification requires a private key which should be kept secret. We construct DSR-Key based solely on the quantum hardness of the LWE problem [Reg05]. We remark that our scheme is inherently *stateful*, i.e., whenever a user generates a new signature, the user must update the signing key for the next invocation of the signing algorithm. Indeed, we believe that digital signatures with revocable signing keys must be inherently stateful since a user must keep the quantum signing key as a “state” for generating multiple signatures. An undesirable feature of our scheme is that the signing key and signature sizes grow with the number of signatures to be generated.

As complementary result, we also consider DSR-Key with *quantum* revocation. In this primitive, not a classical deletion certificate but the quantum signing key itself is returned for the revocation. We construct the primitive from group actions with the one-wayness property [JQSY19]. The existence of group actions with the one-wayness property is incomparable with the LWE assumption.

**Revocable signatures.** Second, we define digital signatures with revocable *signatures* (DSR-Sign). In this primitive, signatures are encoded as quantum states which can later be revoked. The security property guarantees that, once revocation is successful, the initial recipient of the signature loses the ability to pass the signature verification. We construct digital signatures with revocable signatures based on the existence of (quantum-secure) one-way functions (OWFs). In our scheme, the revocation is classical and private, i.e., a user can issue a classical certificate of revocation, which is verified by using a private key.

## 1.3 Comparison with Existing Works

To our knowledge, there is no prior work that studies digital signatures with quantum signatures. On the other hand, there are several existing works that study digital signatures with quantum signing keys. We review them and compare them with our DSR-Key.

- **Tokenized signatures** [BDS23, CLLZ21]. In a tokenized signature scheme, the signing key corresponds to a quantum state which can be used to generate a signature on at most one message. At first sight, the security notion seems to imply the desired security guarantee for DSR-Key, since a signature for a dummy message may serve as the classical deletion certificate for the signing key.<sup>2</sup> However, the

---

<sup>2</sup>Note, however, that tokenized signatures offer public verification of signatures, whereas certifying revocation in our DSR-Key scheme takes place in private.

problem is that tokenized signatures do not achieve the correctness for DSR-Key; namely, in tokenized signatures, a user who receives a quantum signing key can generate only a single signature, whereas in DSR-Key, we require that a user can generate arbitrarily many signatures before the signing key is revoked. Thus, tokenized signatures are not sufficient for achieving our goal. A similar problem exists for semi-quantum tokenized signatures [Shm22] and one-shot signatures [AGKZ20] as well.

- **Copy-protection for digital signatures [LLQZ22] (a.k.a. single-signer signatures [AGKZ20]).<sup>3</sup>** In this primitive, a signing key corresponds to a quantum state which cannot be copied. More precisely, suppose that a user is given one copy of the signing key and tries to split it into two signing keys. The security property requires that at most one of these two signing keys is capable at generating a valid signature on a random message. Amos, Georgiou, Kiayias, and Zhandry [AGKZ20] constructed such a signature scheme based on one-shot signatures. However, the only known construction of one-shot signatures is relative to classical oracles, and there is no known construction without oracles. Liu, Liu, Qian, and Zhandry [LLQZ22] constructed it based on indistinguishability obfuscation (iO) and OWFs. Intuitively, copy-protection for digital signatures implies DSR-Key, because checking whether a returned signing key succeeds at generating valid signatures on random messages can serve a means of verification for revocation.<sup>4</sup> Compared with this approach, our construction has the advantage that it is based on the standard assumption (namely the LWE assumption), whereas they require the very strong assumption of iO or ideal oracles. On the other hand, a disadvantage of our construction is that revocation requires private information, whereas theirs have the potential for public revocation. Another disadvantage is that the size of the signing key (and signatures) grows with the number of signatures, whereas this is kept constant in [LLQZ22] (but not in [AGKZ20]).

## 1.4 Technical Overview

Here we give intuitive explanations of our constructions.

**Construction of DSR-Key.** Our first scheme, DSR-Key, is constructed using *two-tier one-shot signatures* (2-OSS), which is a new primitive which we introduce in this paper.<sup>5</sup> 2-OSS are variants of one-shot signatures [AGKZ20] for single-bit messages. The main difference with regard to one-shot signatures is that there are two verification algorithms, and a signature for the message “0” is verified by a public verification algorithm, whereas a signature for the message “1” is verified by a *private* verification algorithm. We believe that the notion of 2-OSS may be of independent interest. Our construction of 2-OSS is conceptually similar to the construction of two-tier quantum lightning in [KNY21], and can be based solely on the LWE assumption.

From 2-OSS, we then go on to construct DSR-Key. We first construct DSR-Key for single-bit messages from 2-OSS as follows.<sup>6</sup> The signing key  $\text{sigk}$  of DSR-Key consists of a pair  $(\text{sigk}_0, \text{sigk}_1)$  of signing keys of a 2-OSS scheme. To sign a single-bit message  $m \in \{0, 1\}$ , the message “0” is signed with the signing algorithm of a 2-OSS scheme using the signing key  $\text{sigk}_m$ . Because the signature on  $m$  corresponds to a particular signature of “0” with respect to the 2-OSS scheme, it can be verified with the public verification algorithm of 2-OSS. To delete the signing key, the message “1” is signed with the signing algorithm of the

<sup>3</sup>Technically speaking, [LLQZ22] and [AGKZ20] require slightly different security definitions, but high level ideas are the same.

<sup>4</sup>While this sounds plausible, there is a subtlety regarding the security definitions. Indeed, we believe that the security of copy-protection for digital signatures [LLQZ22] or single-signer signatures [AGKZ20] does not readily imply our security definition in Definition 4.1, though they do seem to imply some weaker but reasonable variants of security. See also Remark 4.2.

<sup>5</sup>The term “two-tier” is taken from [KNY21] where they define two-tier quantum lightning, which is a similar variant of quantum lightning [Zha19].

<sup>6</sup>The scheme can be extended to the one for multi-bit messages by using the collision resistant hash functions.

2-OSS scheme by using the signing key. The signature for the message “1” corresponds to the revocation certificate, and it can be verified using the private verification algorithm of 2-OSS.

Our aforementioned construction readily implies a *one-time version* of a DSR-Key scheme, namely, the correctness and security hold when the signing is used only once. We then upgrade it to the many-time version by using a similar chain-based construction of single-signer signatures from one-shot signatures as in [AGKZ20]. That is, it works as follows. The signing key and verification key of the many-time scheme are those of the one-time scheme, respectively. We denote them by  $(\text{ot.sig}k_0, \text{ot.vk}_0)$ . When signing on the first message  $m_1$ , the signer first generates a new key pair  $(\text{ot.sig}k_1, \text{ot.vk}_1)$  of the one-time scheme, uses  $\text{ot.sig}k_0$  to sign on the concatenation  $m_1 \parallel \text{ot.vk}_1$  of the message and the newly generated verification key to generate a signature  $\text{ot.}\sigma_1$  of the one-time scheme. Then it outputs  $(m_1, \text{ot.vk}_1, \text{ot.}\sigma_1)$  as a signature of the many-time scheme.<sup>7</sup> Similarly, when signing on the  $k$ -th message  $m_k$  for  $k \geq 2$ , the signer generates a new key pair  $(\text{ot.sig}k_k, \text{ot.vk}_k)$  and uses  $\text{ot.sig}k_{k-1}$  to sign on  $m_k \parallel \text{ot.vk}_k$  to generate a signature  $\text{ot.}\sigma_k$ . Then the signature of the many-time scheme consists of  $\{m_i, \text{ot.vk}_i, \text{ot.}\sigma_i\}_{i \in [k]}$ . The verification algorithm of the many-time scheme verifies  $\text{ot.}\sigma_i$  for all  $i \in [k]$  under the corresponding message and verification key, and accepts if all of these verification checks pass. To revoke a signing key, the signer generates revocation certificates for all of the signing keys of the one-time scheme which have previously been generated, and the verification of the revocation certificate simply verifies that all these revocation certificates are valid.<sup>8</sup> It is easy to reduce security of the above many-time scheme to that of the one-time scheme.

**Construction of DSR-Sign.** Our second scheme, DSR-Sign, is constructed from what we call *two-tier tokenized signatures* (2-TS), which is a new primitive introduced in this paper. 2-TS are variants of tokenized signatures [BDS23] for single-bit messages where two signature verification algorithms exist. One verification algorithm is used to verify signatures for the message “0”, and it uses the public key. The other verification algorithm is used to verify signatures for the message “1”, and it uses the *secret* key.

We construct 2-TS from OWFs by using a new lemma that we call the *adaptive hardcore bit property for OWFs*, inspired by a similar notion which was shown for a family of noisy trapdoor claw-free functions by Brakerski et al. [BCM<sup>+</sup>21]. We believe that our lemma may be of independent interest, and enable further applications down the line. The adaptive hardcore bit property for OWFs roughly states that given  $|x_0\rangle + (-1)^c |x_1\rangle$  and  $(f(x_0), f(x_1))$ , no QPT adversary can output  $(x, d)$  such that  $f(x) \in \{f(x_0), f(x_1)\}$  and  $d \cdot (x_0 \oplus x_1) = c$ , where  $f$  is a OWF,  $x_0, x_1 \leftarrow \{0, 1\}^\ell$ , and  $c \leftarrow \{0, 1\}$ .<sup>9</sup> The adaptive hardcore bit property for OWFs is shown by using a theorem which is implicit in a recent work [BKM<sup>+</sup>23].

From the adaptive hardcore bit property for OWFs, we construct 2-TS as follows: The quantum signing token is  $|x_0\rangle + (-1)^c |x_1\rangle$  with random  $x_0, x_1 \leftarrow \{0, 1\}^\ell$  and  $c \leftarrow \{0, 1\}$ .<sup>10</sup> The public key is  $(f(x_0), f(x_1))$ , where  $f$  is a OWF, and the secret key is  $(x_0, x_1, c)$ . To sign the message “0”, the token is measured in the computational basis to obtain either  $x_0$  or  $x_1$ . To sign the message “1”, the token is measured in the Hadamard basis to obtain a string  $d$  such that  $d \cdot (x_0 \oplus x_1) = c$ . The measurement result in the computational basis is then verified with the public key, whereas the measurement result in the Hadamard basis is verified with the secret key. Due to the adaptive hardcore bit property for OWFs (formally shown in Theorem 5.1), no QPT adversary can output both signatures at the same time.

Finally, we observe that DSR-Sign can be constructed from any 2-TS scheme by considering the quantum signature of DSR-Sign as a quantum signing token of 2-TS. To verify the quantum signature, we sign the

<sup>7</sup>We include  $m_1$  in the signature for notational convenience even though this is redundant.

<sup>8</sup>The ability to verify all previously generated signing keys (e.g., as part of a chain) may require secret *trapdoor information*.

<sup>9</sup>We actually need its amplified version, because in this case the adversary can win with probability  $1/2$  by measuring the state to get  $x_0$  or  $x_1$ , and randomly choosing  $d$ .

<sup>10</sup>Again, we actually consider its amplified version so that the winning probability of the adversary is negligibly small.

message “0” by using the quantum token, and verify it. To delete the quantum signature, we sign the message “1” by using the quantum token. The verification of the revocation certificate requires one to check whether the deletion certificate is a valid signature for message “1” or not.

## 1.5 Related Works

We have already explained relations between our results and existing works on digital signatures with quantum signing keys. Here, we give a brief review on other related quantum cryptographic primitives.

**Certified deletion and revocation.** Unruh [Unr15] first initiated the study of quantum revocable encryption. This allows the recipient of a quantum ciphertext to return the state, thereby losing all information about the encrypted message. Quantum encryption with certified deletion [HMNY21, Por23, BK22, HKM<sup>+</sup>23, BGG<sup>+</sup>23, BKP23], first introduced by Broadbent and Islam [BI20], enables the deletion of quantum ciphertexts, whereby a classical certificate is produced which can be verified. In particular, [BK22, HKM<sup>+</sup>23, HMNY22] study the certified everlasting security where the security is guaranteed even against unbounded adversary once a valid deletion certificate is issued. [KNY23] and [BKM<sup>+</sup>23] recently showed a general conversion technique to convert the certified everlasting lemma by Bartusek and Kurana [BK22] for the private verification to the public one assuming only OWFs (or even weaker assumptions such as hard quantum planted problems for **NP** or the one-way states generators [MY22]).

The notion of certified deletion has also been used to revoke cryptographic keys [KN22, AKN<sup>+</sup>23, BGG<sup>+</sup>23, APV23, CGJL23]. Here, a key is delegated to a user in the form of a quantum state which can later be revoked. Once the key is destroyed and a valid certificate is issued, the functionality associated with the key is no longer available to the user.

Finally, we remark that the notion of revocation has also been considered in the context of more general programs. Ananth and La Placa [AL21] introduced the notion of secure software leasing. Here, the security guarantees that the functionality of a piece of quantum software is lost once it is returned and verified.

**Copy-protection.** Copy-protection, introduced by Aaronson [Aar09], is a primitive which allows one to encode a functionality into a quantum state in such a way that it cannot be cloned. [ALL<sup>+</sup>21] showed that any unlearnable functionality can be copy-protected with a classical oracle. [CMP20] constructed copy-protection schemes for (multi-bit) point functions as well as compute-and-compare programs in the quantum random oracle model. [CLLZ21] constructed unclonable decryption schemes from iO and compute-and-compare obfuscation for the class of unpredictable distributions, which were previously constructed with classical oracle in [GZ20]. [CLLZ21] also constructed a copy-protection scheme for pseudorandom functions assuming iO, OWFs, and compute-and-compare obfuscation for the class of unpredictable distributions. [LLQZ22] constructed bounded collusion-resistant copy-protection for various functionalities (copy-protection of decryption, digital signatures and PRFs) with iO and LWE.

## 1.6 Organization

Our first result, DSR-Key, is given in Section 4. It is constructed from 2-OSS whose definition and construction from the LWE assumption are given in Section 3. We give a construction of DSR-Key with quantum revocation from group actions in Appendix B.

Our second result, DSR-Sign, is given in Section 7. It is constructed from 2-TS, which is defined and constructed in Section 6. 2-TS is constructed from OWFs via the adaptive hardcore bit property for OWFs, which is explained in Section 5.

## 2 Preliminaries

### 2.1 Basic Notation

We use the standard notations of quantum computing and cryptography. We use  $\lambda$  as the security parameter. For any set  $S$ ,  $x \leftarrow S$  means that an element  $x$  is sampled uniformly at random from the set  $S$ . We write  $\text{negl}$  to mean a negligible function. PPT stands for (classical) probabilistic polynomial-time and QPT stands for quantum polynomial-time. For an algorithm  $A$ ,  $y \leftarrow A(x)$  means that the algorithm  $A$  outputs  $y$  on input  $x$ . For two bit strings  $x$  and  $y$ ,  $x\|y$  means the concatenation of them. For simplicity, we sometimes omit the normalization factor of a quantum state. (For example, we write  $\frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle)$  just as  $|x_0\rangle + |x_1\rangle$ .)  $I := |0\rangle\langle 0| + |1\rangle\langle 1|$  is the two-dimensional identity operator. For the notational simplicity, we sometimes write  $I^{\otimes n}$  just as  $I$  when the dimension is clear from the context.

**Densities and Distances.** Let  $\mathcal{X}$  be a finite domain. A density  $f$  on  $\mathcal{X}$  is a function  $f : \mathcal{X} \rightarrow [0, 1]$  such that  $\sum_{x \in \mathcal{X}} f(x) = 1$ . We denote by  $\mathcal{D}_{\mathcal{X}}$  the set of densities on  $\mathcal{X}$ . For any  $f \in \mathcal{D}_{\mathcal{X}}$ , we let  $\text{Supp}(f) := \{x \in \mathcal{X} : f(x) > 0\}$ . Given two densities  $f_0, f_1$  over  $\mathcal{X}$ , the Hellinger distance between  $f_0$  and  $f_1$  is defined by

$$H^2(f_0, f_1) := 1 - \sum_{x \in \mathcal{X}} \sqrt{f_0(x)f_1(x)}.$$

For two density matrices  $\rho$  and  $\sigma$ , the trace distance is defined as

$$\|\rho - \sigma\|_{\text{tr}} := \frac{1}{2} \|\rho - \sigma\|_1 = \frac{1}{2} \text{Tr} \left[ \sqrt{(\rho - \sigma)^2} \right],$$

where  $\|\cdot\|_1$  is the trace norm. The following elementary lemma relates the Hellinger distance and the trace distance of superpositions.

**Lemma 2.1.** *Let  $\mathcal{X}$  be a finite set,  $f_0, f_1 \in \mathcal{D}_{\mathcal{X}}$  and*

$$|\psi_b\rangle := \sum_{x \in \mathcal{X}} \sqrt{f_b(x)} |x\rangle$$

for  $b \in \{0, 1\}$ . *It holds that*

$$\| |\psi_0\rangle\langle\psi_0| - |\psi_1\rangle\langle\psi_1| \|_{\text{tr}} = \sqrt{1 - (1 - H^2(f_0, f_1))^2}.$$

**Theorem 2.2 (Holevo-Helstrom, [Hol73, Hel69]).** *Consider an experiment in which one of two quantum states, either  $\rho$  or  $\sigma$ , is sent to a distinguisher with probability  $1/2$ . Then, any measurement which seeks to discriminate between  $\rho$  and  $\sigma$  has success probability  $p_{\text{succ}}$  at most*

$$p_{\text{succ}} \leq \frac{1}{2} + \frac{1}{2} \|\rho - \sigma\|_{\text{tr}}.$$

### 2.2 Cryptography

**Definition 2.3 (EUFCMA Secure Digital Signatures).** *An EUFCMA secure digital signature scheme is a set (KeyGen, Sign, Ver) of QPT algorithms such that*

- $\text{KeyGen}(1^\lambda) \rightarrow (\text{sigk}, \text{vk})$  : *It is a QPT algorithm that, on input the security parameter  $\lambda$ , outputs a classical signing key  $\text{sigk}$  and a classical verification key  $\text{vk}$ .*

- $\text{Sign}(\text{sigk}, m) \rightarrow \sigma$  : It is a QPT algorithm that, on input  $\text{sigk}$  and a message  $m$ , outputs a classical signature  $\sigma$ .
- $\text{Ver}(\text{vk}, \sigma, m) \rightarrow \top/\perp$  : It is a QPT algorithm that, on input  $\text{vk}$ ,  $\sigma$ , and  $m$ , outputs  $\top/\perp$ .

We require the following two properties.

**Correctness:** For any message  $m$ ,

$$\Pr[\top \leftarrow \text{Ver}(\text{vk}, \sigma, m) : (\text{sigk}, \text{vk}) \leftarrow \text{KeyGen}(1^\lambda), \sigma \leftarrow \text{Sign}(\text{sigk}, m)] \geq 1 - \text{negl}(\lambda). \quad (1)$$

**EUFCMA security:** For any QPT adversary  $\mathcal{A}$ ,

$$\Pr \left[ \top \leftarrow \text{Ver}(\text{vk}, \sigma, m^*) : \begin{array}{l} (\text{sigk}, \text{vk}) \leftarrow \text{KeyGen}(1^\lambda) \\ (m^*, \sigma) \leftarrow \mathcal{A}^{\text{Sign}(\text{sigk}, \cdot)}(\text{vk}) \end{array} \right] \leq \text{negl}(\lambda), \quad (2)$$

where  $\mathcal{A}$  is not allowed to query  $m^*$  to the signing oracle.

### 2.3 Noisy Trapdoor Claw-Free Hash Function Family

We now recall the notion of noisy trapdoor claw-free (NTCF) hash function family introduced by [BCM<sup>+</sup>21].

**Definition 2.4 (NTCF Hash Function Family [BCM<sup>+</sup>21]).** Let  $\mathcal{X}, \mathcal{Y}$  be finite sets, let  $\mathcal{D}_{\mathcal{Y}}$  be the set of probability densities over  $\mathcal{Y}$ , and  $\mathcal{K}_{\mathcal{F}}$  a finite set of keys. A family of functions

$$\mathcal{F} := \{f_{k,b} : \mathcal{X} \rightarrow \mathcal{D}_{\mathcal{Y}}\}_{k \in \mathcal{K}_{\mathcal{F}}, b \in \{0,1\}}$$

is a NTCF family if the following properties hold:

**Efficient Function Generation:** There exists a PPT algorithm  $\text{NTCF.Gen}_{\mathcal{F}}$  which generates a key  $k \in \mathcal{K}_{\mathcal{F}}$  and a trapdoor  $\text{td}$ .

**Trapdoor Injective Pair:** For all keys  $k \in \mathcal{K}_{\mathcal{F}}$ , the following holds:

- **Trapdoor:** For all  $b \in \{0,1\}$  and  $x \neq x' \in \mathcal{X}$ ,  $\text{Supp}(f_{k,b}(x)) \cap \text{Supp}(f_{k,b}(x')) = \emptyset$ . In addition, there exists an efficient deterministic algorithm  $\text{Inv}_{\mathcal{F}}$  such that for all  $b \in \{0,1\}$ ,  $x \in \mathcal{X}$  and  $y \in \text{Supp}(f_{k,b}(x))$ , we have  $\text{Inv}(\text{td}, b, y) = x$ .
- **Injective pair:** There exists a perfect matching relation  $\mathcal{R}_k \subseteq \mathcal{X} \times \mathcal{X}$  such that  $f_{k,0}(x_0) = f_{k,1}(x_1)$  if and only if  $(x_0, x_1) \in \mathcal{R}_k$ .

**Efficient Range Superposition:** For all keys  $k \in \mathcal{K}_{\mathcal{F}}$  and  $b \in \{0,1\}$ , there exists a function  $f'_{k,b} : \mathcal{X} \rightarrow \mathcal{D}_{\mathcal{Y}}$  such that the following holds:

- For all  $(x_0, x_1) \in \mathcal{R}_k$  and  $y \in \text{Supp}(f'_{k,b}(x_b))$ , it holds that both  $\text{Inv}_{\mathcal{F}}(\text{td}, b, y) = x_b$  and  $\text{Inv}_{\mathcal{F}}(\text{td}, b \oplus 1, y) = x_{b \oplus 1}$ .
- There exists an efficient deterministic procedure  $\text{Chk}_{\mathcal{F}}$  that takes as input  $k \in \mathcal{K}_{\mathcal{F}}, b \in \{0,1\}$ ,  $x \in \mathcal{X}$  and  $y \in \mathcal{Y}$  and outputs 1 if  $y \in \text{Supp}(f'_{k,b}(x))$  and 0 otherwise. This procedure does not need a trapdoor  $\text{td}$ .

- For all  $k \in \mathcal{K}_{\mathcal{F}}$  and  $b \in \{0, 1\}$ ,

$$\mathbb{E}_{x \leftarrow \mathcal{X}} \left[ \mathbf{H}^2(f_{k,b}(x), f'_{k,b}(x)) \right] \leq \text{negl}(\lambda).$$

Here,  $\mathbf{H}^2$  is the Hellinger distance (see Section 2.1). In addition, there exists a QPT algorithm  $\text{Samp}_{\mathcal{F}}$  that takes as input  $k$  and  $b \in \{0, 1\}$  and prepares the quantum state

$$|\psi'\rangle = \frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{(f'_{k,b}(x))(y)} |x\rangle |y\rangle.$$

This property and Lemma 2.1 immediately imply that

$$\| |\psi\rangle\langle\psi| - |\psi'\rangle\langle\psi'| \|_{\text{tr}} \leq \text{negl}(\lambda),$$

$$\text{where } |\psi\rangle = \frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{(f_{k,b}(x))(y)} |x\rangle |y\rangle.$$

**Adaptive Hardcore Bit:** For all keys  $k \in \mathcal{X}_{\mathcal{F}}$ , the following holds. For some integer  $w$  that is a polynomially bounded function of  $\lambda$ ,

- For all  $b \in \{0, 1\}$  and  $x \in \mathcal{X}$ , there exists a set  $\mathcal{G}_{k,b,x} \subseteq \{0, 1\}^w$  such that  $\Pr_{d \leftarrow \{0,1\}^w} [d \notin \mathcal{G}_{k,b,x}] \leq \text{negl}(\lambda)$ . In addition, there exists a PPT algorithm that checks for membership in  $\mathcal{G}_{k,b,x}$  given  $k, b, x$  and  $\text{td}$ .
- There exists an efficiently computable injection  $J : \mathcal{X} \rightarrow \{0, 1\}^w$  such that  $J$  can be inverted efficiently on its range, and such that the following holds. Let

$$\begin{aligned} H_k &:= \{(b, x_b, d, d \cdot (J(x_0) \oplus J(x_1))) \mid b \in \{0, 1\}, (x_0, x_1) \in \mathcal{R}_k, d \in \mathcal{G}_{k,0,x_0} \cap \mathcal{G}_{k,1,x_1}\}, \\ \bar{H}_k &:= \{(b, x_b, d, c) \mid (b, x, d, c \oplus 1) \in H_k\}, \end{aligned}$$

then, for any QPT algorithm  $\mathcal{A}$ , it holds that

$$\left| \Pr_{(k, \text{td}) \leftarrow \text{NTCF}, \text{Gen}_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k) \in H_k] - \Pr_{(k, \text{td}) \leftarrow \text{NTCF}, \text{Gen}_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k) \in \bar{H}_k] \right| \leq \text{negl}(\lambda).$$

Brakerski et al. [BCM<sup>+</sup>21] showed the following theorem.

**Theorem 2.5 ([BCM<sup>+</sup>21]).** Assuming the hardness of the LWE problem, there exists an NTCF family.

We also recall the following amplified adaptive hardcore bit property [RS19, KNY21].

**Definition 2.6 (Amplified Adaptive Hardcore Property).** We say that a NTCF family  $\mathcal{F}$  satisfies the amplified adaptive hardcore bit property if, for any QPT  $\mathcal{A}$  and  $n = \omega(\log \lambda)$ , it holds that

$$\Pr \left[ \begin{array}{l} \forall i \in [n] : x_i = x_{i,b_i}, \quad \forall i \in [n] : (k_i, \text{td}_i) \leftarrow \text{NTCF.Gen}_{\mathcal{F}}(1^\lambda) \\ d_i \in \mathcal{G}_{k_i,0,x_{i,0}} \cap \mathcal{G}_{k_i,1,x_{i,1}}, \quad : \{(b_i, x_i, y_i, d_i, c_i)\}_{i \in [n]} \leftarrow \mathcal{A}(\{k_i\}_{i \in [n]}) \\ c_i = d_i \cdot (J(x_{i,0}) \oplus J(x_{i,1})) \quad x_{i,\beta} \leftarrow \text{Inv}_{\mathcal{F}}(\text{td}_i, \beta, y_i) \text{ for } (i, \beta) \in [n] \times \{0, 1\} \end{array} \right] \leq \text{negl}(\lambda). \quad (3)$$

Finally, we make use of the following result.

**Lemma 2.7 ([RS19, KNY21]).** Any NTCF family satisfies the amplified adaptive hardcore property.

### 3 Two-tier One-shot Signatures

In this section, we define two-tier one-shot signatures (2-OSS), and construct it from the LWE assumption [Reg05]. Broadly speaking, this cryptographic primitive is a variant of one-shot signatures [AGKZZ20], where the verification of a signature for the message “0” is done publicly, whereas that for the message “1” is done only privately.

#### 3.1 Definition

The formal definition of 2-OSS is as follows.

**Definition 3.1 (Two-Tier One-Shot Signatures (2-OSS)).** *A two-tier one-shot signature scheme is a set  $(\text{Setup}, \text{KeyGen}, \text{Sign}, \text{Ver}_0, \text{Ver}_1)$  of algorithms such that*

- $\text{Setup}(1^\lambda) \rightarrow (\text{pp}, \text{sk})$  : *It is a QPT algorithm that, on input the security parameter  $\lambda$ , outputs a classical parameter  $\text{pp}$  and a classical secret key  $\text{sk}$ .*
- $\text{KeyGen}(\text{pp}) \rightarrow (\text{sigk}, \text{vk})$  : *It is a QPT algorithm that, on input  $\text{pp}$ , outputs a quantum signing key  $\text{sigk}$  and a classical verification key  $\text{vk}$ .*
- $\text{Sign}(\text{sigk}, m) \rightarrow \sigma$  : *It is a QPT algorithm that, on input  $\text{sigk}$  and a message  $m \in \{0, 1\}$ , outputs a classical signature  $\sigma$ .*
- $\text{Ver}_0(\text{pp}, \text{vk}, \sigma) \rightarrow \top/\perp$  : *It is a QPT algorithm that, on input  $\text{pp}$ ,  $\text{vk}$ , and  $\sigma$ , outputs  $\top/\perp$ .*
- $\text{Ver}_1(\text{pp}, \text{sk}, \text{vk}, \sigma) \rightarrow \top/\perp$  : *It is a QPT algorithm that, on input  $\text{pp}$ ,  $\text{sk}$ , and  $\sigma$ , outputs  $\top/\perp$ .*

We require the following properties.

**Correctness:**

$$\Pr \left[ \begin{array}{l} (\text{sk}, \text{pp}) \leftarrow \text{Setup}(1^\lambda) \\ \top \leftarrow \text{Ver}_0(\text{pp}, \text{vk}, \sigma) : (\text{sigk}, \text{vk}) \leftarrow \text{KeyGen}(\text{pp}) \\ \sigma \leftarrow \text{Sign}(\text{sigk}, 0) \end{array} \right] \geq 1 - \text{negl}(\lambda) \quad (4)$$

and

$$\Pr \left[ \begin{array}{l} (\text{sk}, \text{pp}) \leftarrow \text{Setup}(1^\lambda) \\ \top \leftarrow \text{Ver}_1(\text{pp}, \text{sk}, \text{vk}, \sigma) : (\text{sigk}, \text{vk}) \leftarrow \text{KeyGen}(\text{pp}) \\ \sigma \leftarrow \text{Sign}(\text{sigk}, 1) \end{array} \right] \geq 1 - \text{negl}(\lambda). \quad (5)$$

**Security:** For any QPT adversary  $\mathcal{A}$ ,

$$\Pr \left[ \begin{array}{l} \top \leftarrow \text{Ver}_0(\text{pp}, \text{vk}, \sigma_0) \wedge \top \leftarrow \text{Ver}_1(\text{pp}, \text{sk}, \text{vk}, \sigma_1) : (\text{sk}, \text{pp}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{vk}, \sigma_0, \sigma_1) \leftarrow \mathcal{A}(\text{pp}) \end{array} \right] \leq \text{negl}(\lambda). \quad (6)$$

### 3.2 Construction

We show that 2-OSS can be constructed from the LWE assumption [Reg05]. Specifically, we make use of NTCF families (see Section 2.3) which allow us to generate quantum states that have a nice structure in both the computational basis, as well as the Hadamard basis. Our 2-OSS scheme is based on the two-tier quantum lightning scheme in [KNY21] and leverages this structure to sign messages: to sign the message “0”, we output a measurement outcome in the computational basis, whereas if we wish to sign “1”, we output a measurement outcome in the Hadamard basis. Crucially, the so-called adaptive hardcore-bit property (see Definition 2.4) ensures that it is computationally difficult to produce the two outcomes simultaneously.

**Theorem 3.2.** *Assuming the quantum hardness of the LWE problem, there exists two-tier one-shot signatures.*

*Proof.* To construct a 2-OSS scheme from LWE, we use NTCF families (see Definition 2.4). Our construction is almost identical to the construction of two-tier quantum lightning in [KNY21].

**Construction.** Let  $n = \omega(\log \lambda)$ . Consider the following scheme.

- **Setup**( $1^\lambda$ ): Generate  $(k_i, \text{td}_i) \leftarrow \text{NTCF.Gen}_{\mathcal{F}}(1^\lambda)$  for  $i \in [n]$  and set  $(\text{pp}, \text{sk}) := (\{k_i\}_{i \in [n]}, \{\text{td}_i\}_{i \in [n]})$ .
- **KeyGen**(pp): Parse  $\text{pp} = \{k_i\}_{i \in [n]}$ . For each  $i \in [n]$ , generate a quantum state

$$|\psi'_i\rangle = \frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}, b \in \{0,1\}} \sqrt{(f'_{k_i,b}(x))(y)} |b, x\rangle |y\rangle$$

by using  $\text{Samp}_{\mathcal{F}}$ , measure the last register to obtain  $y_i \in \mathcal{Y}$ , and let  $|\phi'_i\rangle$  be the post-measurement state where the measured register is discarded. Output  $(\text{vk}, \text{sigk}) = (\{y_i\}_{i \in [n]}, \{|\phi'_i\rangle\}_{i \in [n]})$ .

- **Sign**(sigk,  $m$ ): on input  $m \in \{0, 1\}$ , parse sigk as  $\{|\phi'_i\rangle\}_{i \in [n]}$  and then proceed as follows:
  - if  $m = 0$ : measure  $|\phi'_i\rangle$  in the computational basis for  $i \in [n]$ , which results in measurement outcomes  $\{(b_i, x_i)\}_{i \in [n]}$ , and output  $\sigma = \{(b_i, x_i)\}_{i \in [n]}$  as the signature.
  - if  $m = 1$ : measure  $|\phi'_i\rangle$  in the Hadamard basis for  $i \in [n]$ , which results in measurement outcomes  $\{(c_i, d_i)\}_{i \in [n]}$ , and output  $\sigma = \{(c_i, d_i)\}_{i \in [n]}$  as the signature.
- **Ver<sub>0</sub>**(pk, vk,  $\sigma$ ): Parse  $\text{pp} = \{k_i\}_{i \in [n]}$ ,  $\text{vk} = \{y_i\}_{i \in [n]}$  and  $\sigma = \{(b_i, x_i)\}_{i \in [n]}$ . Use the procedure  $\text{Chk}_{\mathcal{F}}$  in Definition 2.4 to determine if  $y_i \in \text{Supp}(f'_{k_i,b}(x_i))$  for all  $i \in [n]$ . Output  $\top$  if it is the case, and output  $\perp$  otherwise.
- **Ver<sub>1</sub>**(pp, sk, vk,  $\sigma$ ): Parse  $\text{pp} = \{k_i\}_{i \in [n]}$ ,  $\text{sk} = \{\text{td}_i\}_{i \in [n]}$ ,  $\text{vk} = \{y_i\}_{i \in [n]}$  and  $\sigma = \{(c_i, d_i)\}_{i \in [n]}$ . Then, compute  $x_{i,\beta} \leftarrow \text{Inv}_{\mathcal{F}}(\text{td}_i, \beta, y_i)$  for every  $i \in [n]$  and  $\beta \in \{0, 1\}$ . Output  $\top$ , if  $d_i \in \mathcal{G}_{k,0,x_{i,0}} \cap \mathcal{G}_{k,1,x_{i,1}}$  and  $c_i = d_i \cdot (J(x_{i,0}) \oplus J(x_{i,1}))$  for all  $i \in [n]$ , and output  $\perp$  otherwise.

The correctness of our 2-OSS scheme follows from the properties of the NTCF in Definition 2.4. Security follows from the amplified adaptive hardcore bit property in Definition 2.6.  $\square$

## 4 Digital Signatures with Revocable Signing Keys

In this section, we define digital signatures with revocable signing keys (DSR-Key) and give its construction from 2-OSS.

## 4.1 Definition

Let us now present a formal definition of DSR-Key. Note that we consider the *stateful* setting which requires that the signer keep a *state* of all previously signed messages and keys.

**Definition 4.1 ((Stateful) Digital Signatures with Revocable Signing Keys (DSR-Key)).** A (stateful) digital signature scheme with revocable signing keys is the following set (Setup, KeyGen, Sign, Ver, Del, Cert) of algorithms:

- $\text{Setup}(1^\lambda) \rightarrow (\text{ck}, \text{pp})$  : It is a QPT algorithm that, on input the security parameter  $\lambda$ , outputs a classical key  $\text{ck}$  and a classical parameter  $\text{pp}$ .
- $\text{KeyGen}(\text{pp}) \rightarrow (\text{sigk}_0, \text{vk})$  : It is a QPT algorithm that, on input  $\text{pp}$ , outputs a quantum signing key  $\text{sigk}_0$  and a classical verification key  $\text{vk}$ .
- $\text{Sign}(\text{pp}, \text{sigk}_i, m) \rightarrow (\text{sigk}_{i+1}, \sigma)$  : It is a QPT algorithm that, on input  $\text{pp}$ , a message  $m$  and a signing key  $\text{sigk}_i$ , outputs a subsequent signing key  $\text{sigk}_{i+1}$  and a classical signature  $\sigma$ .
- $\text{Ver}(\text{pp}, \text{vk}, m, \sigma) \rightarrow \top/\perp$  : It is a QPT algorithm that, on input  $\text{pp}$ ,  $\text{vk}$ ,  $m$ , and  $\sigma$ , outputs  $\top/\perp$ .
- $\text{Del}(\text{sigk}_i) \rightarrow \text{cert}$  : It is a QPT algorithm that, on input  $\text{sigk}_i$ , outputs a classical certificate  $\text{cert}$ .
- $\text{Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, S) \rightarrow \top/\perp$  : It is a QPT algorithm that, on input  $\text{pp}$ ,  $\text{vk}$ ,  $\text{ck}$ ,  $\text{cert}$ , and a set  $S$  consisting of messages, outputs  $\top/\perp$ .

We require the following properties.

**Many-time correctness:** For any polynomial  $p = p(\lambda)$ , and any messages  $(m_1, m_2, \dots, m_p)$ ,

$$\Pr \left[ \bigwedge_{i \in [p]} \top \leftarrow \text{Ver}(\text{pp}, \text{vk}, m_i, \sigma_i) : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{sigk}_0, \text{vk}) \leftarrow \text{KeyGen}(\text{pp}) \\ (\text{sigk}_1, \sigma_1) \leftarrow \text{Sign}(\text{pp}, \text{sigk}_0, m_1) \\ (\text{sigk}_2, \sigma_2) \leftarrow \text{Sign}(\text{pp}, \text{sigk}_1, m_2) \\ \dots \\ (\text{sigk}_p, \sigma_p) \leftarrow \text{Sign}(\text{pp}, \text{sigk}_{p-1}, m_p) \end{array} \right] \geq 1 - \text{negl}(\lambda). \quad (7)$$

We say that the scheme satisfies one-time correctness if the above property is satisfied for  $p = 1$ .

**EUFCMA security:** For any QPT adversary  $\mathcal{A}$ ,

$$\Pr \left[ \top \leftarrow \text{Ver}(\text{pp}, \text{vk}, m^*, \sigma^*) : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{sigk}_0, \text{vk}) \leftarrow \text{KeyGen}(\text{pp}) \\ (m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{Sign}}}(\text{vk}) \end{array} \right] \leq \text{negl}(\lambda), \quad (8)$$

where  $\mathcal{O}_{\text{Sign}}$  is a stateful signing oracle defined below and  $\mathcal{A}$  is not allowed to query the oracle on  $m^*$ :

$\mathcal{O}_{\text{Sign}}$ : Its initial state is set to be  $(\text{pp}, \text{sigk}_0)$ . When a message  $m$  is queried, it proceeds as follows:

- Parse its state as  $(\text{pp}, \text{sigk}_i)$ .
- Run  $(\text{sigk}_{i+1}, \sigma) \leftarrow \text{Sign}(\text{pp}, \text{sigk}_i, m)$ .
- Return  $\sigma$  to  $\mathcal{A}$  and update its state to  $(\text{pp}, \text{sigk}_{i+1})$ .

We say that the scheme satisfies one-time EUFCMA security if Equation (8) holds for any  $\mathcal{A}$  that submits at most one query to the oracle.

**Many-time deletion correctness:** For any polynomial  $p = p(\lambda)$ , and any messages  $(m_1, m_2, \dots, m_p)$ ,

$$\Pr \left[ \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{sigk}_0, \text{vk}) \leftarrow \text{KeyGen}(\text{pp}) \\ (\text{sigk}_1, \sigma_1) \leftarrow \text{Sign}(\text{pp}, \text{sigk}_0, m_1) \\ (\text{sigk}_2, \sigma_2) \leftarrow \text{Sign}(\text{pp}, \text{sigk}_1, m_2) \\ \dots \\ (\text{sigk}_p, \sigma_p) \leftarrow \text{Sign}(\text{pp}, \text{sigk}_{p-1}, m_p) \\ \text{cert} \leftarrow \text{Del}(\text{sigk}_p) \end{array} \right] \geq 1 - \text{negl}(\lambda). \quad (9)$$

We remark that we require the above to also hold for the case of  $p = 0$ , in which case the fifth component of the input of  $\text{Cert}$  is the empty set  $\emptyset$ . We say that the scheme satisfies one-time deletion correctness if the above property is satisfied for  $p \leq 1$ .

**Many-time deletion security:** For any QPT adversary  $\mathcal{A}$ ,

$$\Pr \left[ \begin{array}{l} \top \leftarrow \text{Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, S) \\ \wedge m^* \notin S \\ \wedge \top \leftarrow \text{Ver}(\text{pp}, \text{vk}, m^*, \sigma^*) \end{array} : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{vk}, \text{cert}, S, m^*, \sigma^*) \leftarrow \mathcal{A}(\text{pp}) \end{array} \right] \leq \text{negl}(\lambda). \quad (10)$$

We say that the scheme satisfies one-time deletion security if the above property is satisfied if we additionally require  $|S| \leq 1$ .

*Remark 4.2.* Following the definition of single signer security in [AGKZ20] or copy-protection security in [LLQZ22], it is also reasonable to define deletion security as follows:

For any pair  $(\mathcal{A}_1, \mathcal{A}_2)$  of QPT adversaries and any distribution  $\mathcal{D}$  with super-logarithmic min-entropy over the message space,

$$\Pr \left[ \begin{array}{l} \top \leftarrow \text{Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, S) \wedge \top \leftarrow \text{Ver}(\text{pp}, \text{vk}, m, \sigma) : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{vk}, \text{cert}, S, \text{st}) \leftarrow \mathcal{A}_1(\text{pp}) \\ m \leftarrow \mathcal{D} \\ \sigma \leftarrow \mathcal{A}_2(m, \text{st}) \end{array} \right] \leq \text{negl}(\lambda). \quad (11)$$

It is easy to see that our definition implies the above, but the converse is unlikely. This is why we define deletion security as in Definition 4.1.

## 4.2 One-Time Construction for Single-Bit Messages

Here we show that we can construct one-time DSR-Key for single-bit messages from 2-OSS in a black-box way.

**Theorem 4.3.** *If two-tier one-shot signatures exist, then digital signatures with revocable signing keys with the message space  $\{0, 1\}$  that satisfy one-time variants of correctness, EUF-CMA security, deletion correctness, and deletion security in Definition 4.1 exist.*

*Proof.* Let  $(\text{OS.Setup}, \text{OS.KeyGen}, \text{OS.Sign}, \text{OS.Ver}_0, \text{OS.Ver}_1)$  be a two-tier one-shot signature scheme. From it, we construct a one-time digital signature scheme  $\Sigma := (\text{Setup}, \text{KeyGen}, \text{Sign}, \text{Ver}, \text{Del}, \text{Cert})$  with revocable signing keys as follows.

- $\text{Setup}(1^\lambda) \rightarrow (\text{ck}, \text{pp}) : \text{Run}(\text{pp}', \text{sk}) \leftarrow \text{OS.Setup}(1^\lambda)$ . Output  $\text{ck} := \text{sk}$  and  $\text{pp} := \text{pp}'$ .
- $\text{KeyGen}(\text{pp}) \rightarrow (\text{sigk}, \text{vk}) : \text{Run}(\text{sigk}_0, \text{vk}_0) \leftarrow \text{OS.KeyGen}(\text{pp})$ .  $\text{Run}(\text{sigk}_1, \text{vk}_1) \leftarrow \text{OS.KeyGen}(\text{pp})$ . Output  $\text{sigk} := (\text{sigk}_0, \text{sigk}_1)$  and  $\text{vk} := (\text{vk}_0, \text{vk}_1)$ .
- $\text{Sign}(\text{pp}, \text{sigk}, m) \rightarrow (\text{sigk}', \sigma) : \text{Parse sigk} = (\text{sigk}_0, \text{sigk}_1)$ .  $\text{Run } \sigma \leftarrow \text{OS.Sign}(\text{pp}, \text{sigk}_m, 0)$  and let  $\text{sigk}' := (1 - m, \text{sigk}_{1-m})$ . Output  $(\text{sigk}', \sigma)$ .
- $\text{Ver}(\text{pp}, \text{vk}, m, \sigma) \rightarrow \top/\perp : \text{Parse vk} = (\text{vk}_0, \text{vk}_1)$ .  $\text{Run OS.Ver}_0(\text{pp}, \text{vk}_m, \sigma)$ , and output its output.
- $\text{Del}(\text{sigk}) \rightarrow \text{cert} :$ 
  - If  $\text{sigk}$  is of the form  $(\text{sigk}_0, \text{sigk}_1)$  (i.e., no signature has been generated), generate  $\sigma'_0 \leftarrow \text{OS.Sign}(\text{sigk}_0, 1)$  and  $\sigma'_1 \leftarrow \text{OS.Sign}(\text{sigk}_1, 1)$  and output  $\text{cert} = (0, 1, \sigma'_0, \sigma'_1)$ .
  - Otherwise, parse  $\text{sigk} = (1 - m, \text{sigk}_{1-m})$ , generate  $\sigma' \leftarrow \text{OS.Sign}(\text{sigk}_{1-m}, 1)$  and output  $\text{cert} = (1 - m, \sigma')$ .
- $\text{Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, S) \rightarrow \top/\perp : \text{Parse vk} = (\text{vk}_0, \text{vk}_1)$  and  $\text{ck} = \text{sk}$ .
  - If  $S = \emptyset$ , parse  $\text{cert} = (0, 1, \sigma'_0, \sigma'_1)$ . (Output  $\perp$  if  $\text{cert}$  is not of this form.) Output  $\top$  if  $\text{OS.Ver}_1(\text{pp}, \text{sk}, \text{vk}_b, \sigma'_b) = \top$  for both  $b = 0, 1$ .
  - If  $S = \{m\}$  for some  $m \in \{0, 1\}$ , parse  $\text{cert} = (1 - m, \sigma')$ . (Output  $\perp$  if  $\text{cert}$  is not of this form.) Output  $\top$  if  $\text{OS.Ver}_1(\text{pp}, \text{sk}, \text{vk}_{1-m}, \sigma') = \top$
  - If  $S = \{0, 1\}$ , output  $\perp$ .

Let us now verify that all of the one-time properties are satisfied.

**One-time correctness.** Let  $m \in \{0, 1\}$  be an arbitrary message. Then, the correctness of  $\text{OS.Ver}_0$  immediately implies that the following one-time correctness holds, i.e.,

$$\Pr \left[ \top \leftarrow \text{Ver}(\text{pp}, \text{vk}, m, \sigma) : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{sigk}, \text{vk}) \leftarrow \text{KeyGen}(\text{pp}) \\ (\text{sigk}', \sigma) \leftarrow \text{Sign}(\text{pp}, \text{sigk}, m) \end{array} \right] \geq 1 - \text{negl}(\lambda). \quad (12)$$

**One-time EUF-CMA security.** Suppose that there exists a QPT adversary  $\mathcal{A}$  which violates one-time EUF-CMA security. In other words, there is a polynomial  $q(\lambda)$  such that for infinitely many  $\lambda \in \mathbb{N}$ ,

$$\Pr \left[ \top \leftarrow \text{Ver}(\text{pp}, \text{vk}, m^*, \sigma^*) : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{sigk}, \text{vk}) \leftarrow \text{KeyGen}(\text{pp}) \\ (m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{Sign}}}(\text{vk}) \end{array} \right] \geq \frac{1}{q(\lambda)}. \quad (13)$$

where  $\mathcal{A}$  only makes at most single query to the stateful  $\mathcal{O}_{\text{Sign}}$  in Definition 4.1 which does not equal  $m^*$ . Without loss of generality, we assume that  $\mathcal{A}$  queries on  $1 - m^*$  (otherwise, if  $\mathcal{A}$  does not submit any query, we can simply consider  $\mathcal{A}'$  which first runs  $\mathcal{A}$  and then subsequently queries on  $1 - m^*$ ). Then, we can construct the following QPT algorithm  $\mathcal{B}$  which breaks the security of the two-tier one-shot signature scheme:

1. On input  $\text{pp}$ ,  $\mathcal{B}$  generates  $(\text{sigk}, \text{vk}) \leftarrow \text{KeyGen}(\text{pp})$  and parses the signing key as  $\text{sigk} = (\text{sigk}_0, \text{sigk}_1)$  and the verification key as  $\text{vk} = (\text{vk}_0, \text{vk}_1)$ .

2.  $\mathcal{B}$  runs the adversary  $\mathcal{A}$  on input  $\text{vk}$ ; when  $\mathcal{A}$  submits a query, say of the form  $1 - m^*$ ,  $\mathcal{B}$  simulates the oracle  $\mathcal{O}_{\text{Sign}}$  by running  $(\text{sigk}', \sigma') \leftarrow \text{Sign}(\text{pp}, \text{sigk}, 1 - m^*)$  and sends back  $\sigma'$  to  $\mathcal{A}$ . Let  $(m^*, \sigma^*)$  denote the final output returned by  $\mathcal{A}^{\mathcal{O}_{\text{Sign}}}(\text{vk})$  and assign  $\sigma_0 = \sigma^*$ .
3.  $\mathcal{B}$  parses  $\text{sigk}' = (m^*, \text{sigk}_{m^*})$  and computes  $\sigma_1 \leftarrow \text{OT.Sign}(\text{pp}, \text{sigk}_{m^*}, 1)$ .
4.  $\mathcal{B}$  outputs the triplet  $(\text{vk}_{m^*}, \sigma_0, \sigma_1)$ .

We now argue that  $\mathcal{B}$  breaks the security of the two-tier one-shot signature scheme. Using Equation (13), it follows that  $\mathcal{B}$  obtains two valid signatures  $\sigma_0, \sigma_1$  (of which one corresponds to 0 and the other corresponds to 1) with probability at least

$$\Pr \left[ \top \leftarrow \text{OS.Ver}_0(\text{pp}, \text{vk}, \sigma_0) \wedge \top \leftarrow \text{OS.Ver}_1(\text{pp}, \text{sk}, \text{vk}, \sigma_1) : \begin{array}{l} (\text{sk}, \text{pp}) \leftarrow \text{OS.Setup}(1^\lambda) \\ (\text{vk}, \sigma_0, \sigma_1) \leftarrow \mathcal{B}(\text{pp}) \end{array} \right] \geq \frac{1}{\text{poly}(\lambda)}. \quad (14)$$

This violates the security of the two-tier one-shot signature according to Definition 3.1.

**One-time deletion correctness.** Here, we consider two cases. First, when no signature is produced,  $\text{sigk}$  is of the form  $(\text{sigk}_0, \text{sigk}_1)$  and  $\text{cert} = (0, 1, \sigma'_0, \sigma'_1)$ . Thus, both  $\sigma'_0 \leftarrow \text{OS.Sign}(\text{sigk}_0, 1)$  and  $\sigma'_1 \leftarrow \text{OS.Sign}(\text{sigk}_1, 1)$  pass verification via  $\text{OS.Ver}_1(\text{pp}, \text{sk}, \text{vk}_0, \cdot)$  and  $\text{OS.Ver}_1(\text{pp}, \text{sk}, \text{vk}_1, \cdot)$  with overwhelming probability. Therefore, we get the following from the union bound and the correctness of  $\text{OS.Ver}_1$  that

$$\Pr \left[ \top \leftarrow \text{Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, \emptyset) : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{sigk}, \text{vk}) \leftarrow \text{KeyGen}(\text{pp}) \\ \text{cert} \leftarrow \text{Del}(\text{sigk}) \end{array} \right] \geq 1 - \text{negl}(\lambda). \quad (15)$$

Next, we analyze the case when precisely one signature is generated on some message  $m \in \{0, 1\}$ . In this case,  $\text{sigk}'$  is of the form  $(1 - m, \text{sigk}_{1-m})$  and  $\text{cert} = (1 - m, \sigma')$ , and hence  $\sigma' \leftarrow \text{OS.Sign}(\text{sigk}_{1-m}, 1)$  passes verification via  $\text{OS.Ver}_1(\text{pp}, \text{sk}, \text{vk}_{1-m}, \cdot)$  with overwhelming probability. This implies that

$$\Pr \left[ \top \leftarrow \text{Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, \{m\}) : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{sigk}, \text{vk}) \leftarrow \text{KeyGen}(\text{pp}) \\ (\text{sigk}', \sigma) \leftarrow \text{Sign}(\text{pp}, \text{sigk}, m) \\ \text{cert} \leftarrow \text{Del}(\text{sigk}') \end{array} \right] \geq 1 - \text{negl}(\lambda). \quad (16)$$

**One-time deletion security:** Suppose that there exists a QPT adversary  $\mathcal{A}$  which violates one-time deletion security. In other words, there exists a polynomial  $q(\lambda)$  such that for infinitely many  $\lambda \in \mathbb{N}$ , it holds that

$$\Pr \left[ \begin{array}{l} \top \leftarrow \text{Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, S) \\ \wedge m^* \notin S \\ \wedge \top \leftarrow \text{Ver}(\text{pp}, \text{vk}, m^*, \sigma^*) \\ \wedge |S| \leq 1 \end{array} : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{vk}, \text{cert}, S, m^*, \sigma^*) \leftarrow \mathcal{A}(\text{pp}) \end{array} \right] \geq \frac{1}{q(\lambda)}. \quad (17)$$

Then, we can construct the following QPT algorithm  $\mathcal{B}$  which breaks the security of the underlying two-tier one-shot signature scheme according to Definition 3.1.

1. On input  $\text{pp}$ ,  $\mathcal{B}$  runs  $\mathcal{A}(\text{pp})$  and receives a tuple  $(\text{vk}, \text{cert}, S, m^*, \sigma^*)$  for some set  $S$  of size  $|S| \leq 1$  and some verification key  $\text{vk} = (\text{vk}_0, \text{vk}_1)$ .

2. Depending on the size of  $S$ ,  $\mathcal{B}$  then does the following:

- If  $|S| = 0$ , i.e.,  $S = \emptyset$ ,  $\mathcal{B}$  parses  $\text{cert} = (0, 1, \sigma'_0, \sigma'_1)$  and outputs  $(\text{vk}_{m^*}, \sigma^*, \sigma'_{m^*})$ .
- If  $|S| = 1$ , i.e.,  $S = \{m\}$  for  $m \neq m^*$ ,  $\mathcal{B}$  parses  $\text{cert} = (1 - m, \sigma')$  and outputs  $(\text{vk}_{m^*}, \sigma^*, \sigma')$ .

Note that in either case, whether  $|S| = 0$  or  $|S| = 1$ , it follows that  $\mathcal{B}$  succeeds with probability at least

$$\Pr \left[ \top \leftarrow \text{OS.Ver}_0(\text{pp}, \text{vk}, \sigma_0) \wedge \top \leftarrow \text{OS.Ver}_1(\text{pp}, \text{sk}, \text{vk}, \sigma_1) : \begin{array}{l} (\text{sk}, \text{pp}) \leftarrow \text{OS.Setup}(1^\lambda) \\ (\text{vk}, \sigma_0, \sigma_1) \leftarrow \mathcal{B}(\text{pp}) \end{array} \right] \geq \frac{1}{\text{poly}(\lambda)},$$

which violates the security of the two-tier one-shot signature according to Definition 3.1.  $\square$

### 4.3 From Single-Bit to Multi-Bit Messages

Here, we show that we can expand the message space to  $\{0, 1\}^*$  using collision-resistant hashes.

**Theorem 4.4.** *If collision-resistant hash functions and digital signatures with revocable signing keys with the message space  $\{0, 1\}$  that satisfy one-time variants of correctness, EUF-CMA security, deletion correctness, and deletion security in Definition 4.1 exist, then a similar scheme with the message space  $\{0, 1\}^*$  exists.*

*Proof.* Let  $\Sigma = (\text{Setup}, \text{KeyGen}, \text{Sign}, \text{Ver}, \text{Del}, \text{Cert})$  be a scheme with the message space  $\{0, 1\}$  which satisfies the one-time variants of correctness, EUF-CMA security, deletion correctness, and deletion security. Let  $\mathcal{H}$  be a family of collision resistant hash functions from  $\{0, 1\}^*$  to  $\{0, 1\}^\ell$ . Then we construct  $\Sigma' = (\text{Setup}', \text{KeyGen}', \text{Sign}', \text{Ver}', \text{Del}', \text{Cert}')$  with the message space  $\{0, 1\}^*$  as follows:

- $\text{Setup}'(1^\lambda) \rightarrow (\text{ck}, \text{pp})$ : Choose  $H \leftarrow \mathcal{H}$ . Run  $(\Sigma.\text{ck}, \Sigma.\text{pp}) \leftarrow \text{Setup}(1^\lambda)$  and output  $\text{ck} := \Sigma.\text{ck}$  and  $\text{pp} := (\Sigma.\text{pp}, H)$ .
- $\text{KeyGen}'(\text{pp}) \rightarrow (\text{sigk}, \text{vk})$ : Parse  $\text{pp} = (\Sigma.\text{pp}, H)$ . For  $i \in [\ell]$ , run  $(\text{sigk}_i, \text{vk}_i) \leftarrow \text{KeyGen}(\Sigma.\text{pp})$  and output  $\text{sigk} := \{\text{sigk}_i\}_{i \in [\ell]}$  as the quantum signing key and  $\text{vk} := \{\text{vk}_i\}_{i \in [\ell]}$  as the classical verification key.
- $\text{Sign}'(\text{pp}, \text{sigk}, m) \rightarrow (\text{sigk}', \sigma)$ : Parse  $\text{pp} = (\Sigma.\text{pp}, H)$  and  $\text{sigk} = \{\text{sigk}_i\}_{i \in [\ell]}$ . Compute  $m' = H(m)$  and let  $m'_i$  be the  $i$ th bit of  $m'$  for  $i \in [\ell]$ . For  $i \in [\ell]$ , run  $(\text{sigk}'_i, \sigma_i) \leftarrow \text{Sign}(\Sigma.\text{pp}, \text{sigk}_i, m'_i)$  and output  $\text{sigk}' := \{\text{sigk}'_i\}_{i \in [\ell]}$  and  $\sigma := \{\sigma_i\}_{i \in [\ell]}$ .
- $\text{Ver}'(\text{pp}, \text{vk}, m, \sigma) \rightarrow \top/\perp$ : Parse  $\text{pp} = (\Sigma.\text{pp}, H)$ ,  $\text{vk} := \{\text{vk}_i\}_{i \in [\ell]}$ , and  $\sigma = \{\sigma_i\}_{i \in [\ell]}$ . Compute  $m' = H(m)$  and let  $m'_i$  be the  $i$ th bit of  $m'$  for  $i \in [\ell]$ . Output  $\top$  if  $\text{Ver}(\Sigma.\text{pp}, \text{vk}_i, m'_i, \sigma_i) = \top$  for every  $i \in [\ell]$ .
- $\text{Del}'(\text{sigk}) \rightarrow \text{cert}$ : Parse  $\text{sigk} = \{\text{sigk}_i\}_{i \in [\ell]}$ . For  $i \in [\ell]$ , run  $\text{cert}_i \leftarrow \text{Del}(\text{sigk}_i)$  and output  $\text{cert} := \{\text{cert}_i\}_{i \in [\ell]}$ .
- $\text{Cert}'(\text{pp}, \text{vk}, \text{ck}, \text{cert}, S) \rightarrow \top/\perp$ : Parse  $\text{pp} = (\Sigma.\text{pp}, H)$ ,  $\text{vk} = \{\text{vk}_i\}_{i \in [\ell]}$ , and  $\text{cert} = \{\text{cert}_i\}_{i \in [\ell]}$ .
  - If  $S = \emptyset$ , output  $\top$  if  $\text{Cert}(\Sigma.\text{pp}, \text{vk}_i, \text{ck}, \text{cert}_i, \emptyset) = \top$  for every  $i \in [\ell]$ .
  - If  $S = \{m\}$  for some  $m \in \{0, 1\}^*$ , compute  $m' = H(m)$  and let  $m'_i$  be the  $i$ th bit of  $m'$  for  $i \in [\ell]$ . Output  $\top$  if  $\text{Cert}(\Sigma.\text{pp}, \text{vk}_i, \text{ck}, \text{cert}_i, \{m'_i\}) = \top$  for every  $i \in [\ell]$ .
  - Otherwise (i.e., if  $|S| \geq 2$ ), output  $\perp$ .

The proof of correctness is immediate and the proof of one-time EUF-CMA security follows from standard techniques which allow conventional signature schemes to handle messages of arbitrarily length, see [KL07] for example. Therefore, it suffices to show that the scheme  $\Sigma'$  satisfies the one-time variants of deletion correctness and deletion security. Let us now verify each property separately.

**One-time deletion correctness.** Because  $\Sigma$  satisfies one-time correctness, this means that there exists a negligible function  $\nu$  such that for any  $p \leq 1$  and any message  $m_p$  (possibly  $m_p = \emptyset$ , if  $p = 0$ , in which case no message is signed in the expression below) and for all  $\lambda \in \mathbb{N}$ ,

$$\Pr \left[ \begin{array}{l} \top \leftarrow \Sigma.\text{Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, \{m_p\}) : \\ (\text{sigk}_p, \sigma_p) \leftarrow \Sigma.\text{Sign}(\text{pp}, \text{sigk}_0, m_p) \\ \text{cert} \leftarrow \Sigma.\text{Del}(\text{sigk}_p) \end{array} : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \Sigma.\text{Setup}(1^\lambda) \\ (\text{sigk}_0, \text{vk}) \leftarrow \Sigma.\text{KeyGen}(\text{pp}) \end{array} \right] \geq 1 - \nu(\lambda). \quad (18)$$

Hence, by the union bound and the fact that  $\ell \cdot \nu(\lambda) = \text{negl}(\lambda)$  with  $\ell = \ell(\lambda) = \text{poly}(\lambda)$ , it follows that  $\Sigma'$  also satisfies one-time deletion correctness.

**One-time deletion security.** Suppose that  $\Sigma'$  does not satisfy one-time deletion security. In other words, there exists a QPT adversary  $\mathcal{A}$  and a polynomial  $p(\lambda)$  such that for infinitely many  $\lambda \in \mathbb{N}$ , it holds that:

$$\Pr \left[ \begin{array}{l} \top \leftarrow \Sigma'.\text{Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, S) \\ \wedge m^* \notin S \\ \wedge \top \leftarrow \Sigma'.\text{Ver}(\text{pp}, \text{vk}, m^*, \sigma^*) \\ \wedge |S| \leq 1 \end{array} : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \Sigma'.\text{Setup}(1^\lambda) \\ (\text{vk}, \text{cert}, S, m^*, \sigma^*) \leftarrow \mathcal{A}(\text{pp}) \end{array} \right] \geq \frac{1}{p(\lambda)}. \quad (19)$$

In the following, we parse  $\text{pp} \leftarrow \Sigma'.\text{Setup}(1^\lambda)$  as  $\text{pp} = (\Sigma.\text{pp}, H)$  and distinguish between two cases:

**Case 1:** A collision occurs; namely, when  $\mathcal{A}(\text{pp})$  outputs  $(\text{vk}, \text{cert}, S, m^*, \sigma^*)$ , there exists a message  $m \in S$  such that  $H(m) = H(m^*)$ .

**Case 2:** No collision occurs; in other words, when  $\mathcal{A}(\text{pp})$  outputs  $(\text{vk}, \text{cert}, S, m^*, \sigma^*)$ , there does not exist a message  $m \in S$  such that  $H(m) = H(m^*)$ .

Note that Equation (19) implies that either **Case 1** or **Case 2** occurs with probability at least  $1/p(\lambda)$ .

Suppose that **Case 1** occurs, i.e., it holds that

$$\Pr \left[ \begin{array}{l} \top \leftarrow \Sigma'.\text{Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, S) \\ \wedge m^* \notin S \\ \wedge \top \leftarrow \Sigma'.\text{Ver}(\text{pp}, \text{vk}, m^*, \sigma^*) \\ \wedge |S| \leq 1 \\ \wedge (\exists m \in S \text{ s.t. } H(m) = H(m^*)) \end{array} : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \Sigma'.\text{Setup}(1^\lambda) \\ (\Sigma.\text{pp}, H) = \text{pp} \\ (\text{vk}, \text{cert}, S, m^*, \sigma^*) \leftarrow \mathcal{A}(\text{pp}) \end{array} \right] \geq \frac{1}{p(\lambda)}. \quad (20)$$

We can then construct the following QPT algorithm which breaks the collision-resistance of  $\mathcal{H}$ :

1. On input  $(H, 1^\lambda)$  run  $(\Sigma.\text{ck}, \Sigma.\text{pp}) \leftarrow \text{Setup}(1^\lambda)$  and output  $\text{ck} := \Sigma.\text{ck}$  and  $\text{pp} := (\Sigma.\text{pp}, H)$ .
2. Run  $\mathcal{A}(\text{pp})$  who outputs  $(\text{vk}, \text{cert}, S, m^*, \sigma^*)$ .
3. If there exists an  $m \in S$  such that  $H(m) = H(m^*)$ , output  $(m, m^*)$ . Otherwise, output  $\perp$ .

Hence, by our assumption in Equation (20), we obtain a collision with probability at least  $1/p(\lambda)$ , thereby breaking the collision-resistance of  $\mathcal{H}$ .

Suppose that **Case 2** occurs, i.e., it holds that

$$\Pr \left[ \begin{array}{l} \top \leftarrow \Sigma'.\text{Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, S) \\ \wedge m^* \notin S \\ \wedge \top \leftarrow \Sigma'.\text{Ver}(\text{pp}, \text{vk}, m^*, \sigma^*) \\ \wedge |S| \leq 1 \\ \wedge (\nexists m \in S \text{ s.t. } H(m) = H(m^*)) \end{array} : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \Sigma'.\text{Setup}(1^\lambda) \\ (\Sigma.\text{pp}, H) = \text{pp} \\ (\text{vk}, \text{cert}, S, m^*, \sigma^*) \leftarrow \mathcal{A}(\text{pp}) \end{array} \right] \geq \frac{1}{p(\lambda)}. \quad (21)$$

We can then construct the following QPT algorithm which breaks the one-time deletion security of  $\Sigma$ .

1. On input  $\Sigma.\text{pp}$ , where  $(\Sigma.\text{pp}, \Sigma.\text{ck}) \leftarrow \Sigma.\text{Setup}(1^\lambda)$ , sample  $H \leftarrow \mathcal{H}$ . and let  $\text{pp} := (\Sigma.\text{pp}, H)$ .
2. Run  $(\text{vk}, \text{cert}, S, m^*, \sigma^*) \leftarrow \mathcal{A}(\text{pp})$ .
3. Depending on the size of  $S$ , do the following:
  - If  $|S| = 0$ , i.e.,  $S = \emptyset$ , parse  $\text{vk} = \{\text{vk}_i\}_{i \in [\ell]}$ ,  $\text{cert} = \{\text{cert}_i\}_{i \in [\ell]}$ , and  $\sigma^* = \{\sigma_i^*\}_{i \in [\ell]}$ . Output  $(\text{vk}_j, \text{cert}_j, \emptyset, \bar{m}_j^*, \sigma_j^*)$ , where  $\bar{m}^* = H(m^*)$  and  $j \in [\ell]$  is a random index.
  - If  $|S| = 1$ , i.e.,  $S = \{m\}$  for  $m \neq m^*$ , parse  $\text{vk} = \{\text{vk}_i\}_{i \in [\ell]}$ ,  $\text{cert} = \{\text{cert}_i\}_{i \in [\ell]}$ , and  $\sigma^* = \{\sigma_i^*\}_{i \in [\ell]}$ . Output  $(\text{vk}_j, \text{cert}_j, \{\bar{m}_j\}, \bar{m}_j^*, \sigma_j^*)$ , where  $\bar{m} = H(m)$  and  $\bar{m}^* = H(m^*)$  and where  $j$  is an index such that the  $j$ -th bits of  $\bar{m}$  and  $\bar{m}^*$  are distinct.

Note that in either case, whether  $|S| = 0$  or  $|S| = 1$ , our reduction breaks the one-time deletion security of  $\Sigma$  with inverse-polynomial success probability; in particular, if  $|S| = 1$  occurs,  $\bar{m}$  and  $\bar{m}^*$  will be distinct (and hence differ in at least one bit) since we are in **Case 2**. This proves the claim.  $\square$

#### 4.4 From One-Time Schemes to Many-Time Schemes

In this section, we show how to extend any one-time scheme into a proper many-time scheme as in Definition 4.1. The transformation is inspired by the chain-based approach for constructing many-time digital signatures, see [KL07] for example.<sup>11</sup>

Let  $\text{OT} = (\text{OT.Setup}, \text{OT.KeyGen}, \text{OT.Sign}, \text{OT.Ver}, \text{OT.Del}, \text{OT.Cert})$  be a scheme which satisfies the one-time variants of correctness, EUF-CMA security, deletion correctness, and deletion security according to in Definition 4.1, and has the message space  $\{0, 1\}^*$ . Then, we construct  $\text{MT} = (\text{MT.Setup}, \text{MT.KeyGen}, \text{MT.Sign}, \text{MT.Ver}, \text{MT.Del}, \text{MT.Cert})$  with the message space  $\{0, 1\}^n$  as follows:

- $\text{MT.Setup}(1^\lambda) \rightarrow (\text{ck}, \text{pp})$ : This is the same as  $\text{OT.Setup}$ .
- $\text{MT.KeyGen}(\text{pp}) \rightarrow (\text{sigk}, \text{vk})$ : run  $(\text{ot.sigk}_0, \text{ot.vk}_0) \leftarrow \text{OT.KeyGen}(\text{pp})$  and output  $\text{sigk} := \text{ot.sigk}_0$  as the quantum signing key and  $\text{vk} := \text{ot.vk}_0$  as the classical verification key.
- $\text{MT.Sign}(\text{pp}, \text{sigk}_i, m) \rightarrow (\text{sigk}_{i+1}, \sigma)$ : on input the public parameter  $\text{pp}$ , a quantum signing key  $\text{sigk}_i$ , and a message  $m \in \{0, 1\}^n$  proceed as follows:
  1. Parse  $\text{sigk}_i$  as  $(\text{ot.sigk}_i, \{\text{ot.sigk}'_j\}_{j \in \{0, 1, \dots, i-1\}}, \{m_j, \text{ot.vk}_j, \text{ot}\sigma_j\}_{j \in [i]})$

<sup>11</sup>We could also use the tree-based construction [Mer88], which has a shorter (logarithmic) signature length. We describe the chain-based construction here for ease of presentation.

2. Generate  $(\text{ot.sigk}_{i+1}, \text{ot.vk}_{i+1}) \leftarrow \text{OT.KeyGen}(\text{pp})$ .

3. Run

$$(\text{ot.sigk}'_i, \text{ot.}\sigma_{i+1}) \leftarrow \text{OT.Sign}(\text{pp}, \text{ot.sigk}_i, m \parallel \text{ot.vk}_{i+1}).$$

4. Set  $m_{i+1} := m$  and output a subsequent signing key

$$\text{sigk}_{i+1} := (\text{ot.sigk}_{i+1}, \{\text{ot.sigk}'_j\}_{j \in \{0,1,\dots,i\}}, \{m_j, \text{ot.vk}_j, \text{ot.}\sigma_j\}_{j \in [i+1]})$$

and a signature

$$\sigma := \{m_j, \text{ot.vk}_j, \text{ot.}\sigma_j\}_{j \in [i+1]}.$$

- $\text{MT.Ver}(\text{pp}, \text{vk}, m, \sigma) \rightarrow \top/\perp$  : on input pp, a key vk, a message  $m$ , and signature  $\sigma$ , proceed as follows.
  1. Parse  $\sigma$  as  $\{m_j, \text{ot.vk}_j, \text{ot.}\sigma_j\}_{j \in [i]}$  and let  $\text{ot.vk}_0 = \text{vk}$ .
  2. Output  $\top$  if  $m = m_i$  and  $\text{OT.Ver}(\text{pp}, \text{ot.vk}_{j-1}, m_j \parallel \text{ot.vk}_j, \text{ot.}\sigma_j) = \top$  for every  $j \in [i]$ .
- $\text{MT.Del}(\text{sigk}_i) \rightarrow \text{cert}$  : on input sigk, proceed as follows:
  1. Parse sigk <sub>$i$</sub>  as  $(\text{ot.sigk}_i, \{\text{ot.sigk}'_j\}_{j \in \{0,1,\dots,i-1\}}, \{m_j, \text{ot.vk}_j, \text{ot.}\sigma_j\}_{j \in [i]})$ .
  2. For  $j \in \{0, 1, \dots, i-1\}$ , run  $\text{ot.cert}_j \leftarrow \text{OT.Del}(\text{ot.sigk}'_j)$ .
  3. Run  $\text{ot.cert}_i \leftarrow \text{OT.Del}(\text{ot.sigk}_i)$ .
  4. Output  $\text{cert} := \{\text{ot.cert}_j, m_j, \text{ot.vk}_j, \text{ot.}\sigma_j\}_{j \in [i]}$ .
- $\text{MT.Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, S) \rightarrow \top/\perp$  : on input pp, vk, ck, cert, and  $S$ , parse the certificate cert as a tuple  $\{\text{ot.cert}_j, m_j, \text{ot.vk}_j, \text{ot.}\sigma_j\}_{j \in [i]}$ , let  $\text{ot.vk}_0 = \text{vk}$ , and output  $\top$  if the following holds:
  - $S = \{m_1, m_2, \dots, m_i\}$ ,
  - $\text{OT.Cert}(\text{ot.vk}_{j-1}, \text{ck}, \text{ot.cert}_{j-1}, \{m_j \parallel \text{ot.vk}_j\}) = \top$  for every  $j \in [i]$ , and
  - $\text{OT.Cert}(\text{ot.vk}_i, \text{ck}, \text{ot.cert}_i, \emptyset) = \top$ .

We now prove the following theorem.

**Theorem 4.5.** *Suppose that  $(\text{OT.Setup}, \text{OT.KeyGen}, \text{OT.Sign}, \text{OT.Ver}, \text{OT.Del}, \text{OT.Cert})$  satisfies the one-time variants of correctness, EUF-CMA security, deletion correctness, and deletion security in Definition 4.1. Then, the scheme  $(\text{MT.Setup}, \text{MT.KeyGen}, \text{MT.Sign}, \text{MT.Ver}, \text{MT.Del}, \text{MT.Cert})$  satisfies many-time variants of each of the properties.*

*Proof.* Let  $(\text{OT.Setup}, \text{OT.KeyGen}, \text{OT.Sign}, \text{OT.Ver}, \text{OT.Del}, \text{OT.Cert})$  be a scheme which satisfies the one-time variants in Definition 4.1. We now verify that each of the many-time properties are satisfied.

**Many-time correctness.** Because the OT scheme satisfies one-time correctness, this means that there exists a negligible function  $\nu$  such that for any message  $m$  and for all  $\lambda \in \mathbb{N}$ ,

$$\Pr \left[ \begin{array}{l} \top \leftarrow \text{OT.Ver}(\text{vk}, m, \sigma_1) : \\ (\text{pp}, \text{ck}) \leftarrow \text{OT.Setup}(1^\lambda) \\ (\text{sigk}, \text{vk}) \leftarrow \text{OT.KeyGen}(\text{pp}) \\ (\text{sigk}_1, \sigma_1) \leftarrow \text{OT.Sign}(\text{pp}, \text{sigk}, m) \end{array} \right] \geq 1 - \nu(\lambda). \quad (22)$$

Let  $p = p(\lambda)$  be any polynomial and let  $(m_1, m_2, \dots, m_p)$  be any collection of messages. Then, it follows from the union bound and Equation (22) that

$$\Pr \left[ \bigwedge_{i \in [p]} \top \leftarrow \text{MT.Ver}(\text{vk}, m_i, \sigma_i) : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{sigk}_0, \text{vk}) \leftarrow \text{MT.KeyGen}(\text{pp}) \\ (\text{sigk}_1, \sigma_1) \leftarrow \text{MT.Sign}(\text{pp}, \text{sigk}_0, m_1) \\ (\text{sigk}_2, \sigma_2) \leftarrow \text{MT.Sign}(\text{pp}, \text{sigk}_1, m_2) \\ \vdots \\ (\text{sigk}_p, \sigma_p) \leftarrow \text{MT.Sign}(\text{pp}, \text{sigk}_{p-1}, m_p) \end{array} \right] \geq 1 - p(\lambda) \cdot (p(\lambda) + 1) \cdot \nu(\lambda)/2. \quad (23)$$

Finally, because  $p(\lambda) = \text{poly}(\lambda)$  and  $\nu(\lambda) = \text{negl}(\lambda)$  it follows that the above expression in Equation (23) is at least  $1 - \text{negl}(\lambda)$ . This proves the claim.

**EUFCMA security.** The proof is virtually identical to the standard proof that the chain-based signature scheme achieves many-time EUFCMA security, see [KL07], for example. We choose to omit it for brevity.

**Many-time deletion correctness.** Because the OT scheme satisfies one-time correctness, this means that there exists a negligible function  $\nu$  such that for any  $p \leq 1$  and any message  $m_p$  (possibly  $m_p = \emptyset$ , if  $p = 0$ , in which case no message is signed in the expression below) and for all  $\lambda \in \mathbb{N}$ ,

$$\Pr \left[ \top \leftarrow \text{OT.Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, \{m_p\}) : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \text{OT.Setup}(1^\lambda) \\ (\text{sigk}_0, \text{vk}) \leftarrow \text{OT.KeyGen}(\text{pp}) \\ (\text{sigk}_p, \sigma_p) \leftarrow \text{OT.Sign}(\text{pp}, \text{sigk}_0, m_p) \\ \text{cert} \leftarrow \text{OT.Del}(\text{sigk}_p) \end{array} \right] \geq 1 - \nu(\lambda). \quad (24)$$

Now let  $p = p(\lambda)$  be any polynomial and let  $(m_1, m_2, \dots, m_p)$  be any collection of messages. In the case when  $p = 0$  the claim follows immediately from the one-time deletion correctness of the OT scheme. More generally in the case when  $p(\lambda) \geq 1$  it follows from the union bound and Equation (25) that

$$\Pr \left[ \top \leftarrow \text{MT.Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, \{m_1, m_2, \dots, m_p\}) : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{sigk}_0, \text{vk}) \leftarrow \text{MT.KeyGen}(\text{pp}) \\ (\text{sigk}_1, \sigma_1) \leftarrow \text{MT.Sign}(\text{pp}, \text{sigk}_0, m_1) \\ (\text{sigk}_2, \sigma_2) \leftarrow \text{MT.Sign}(\text{pp}, \text{sigk}_1, m_2) \\ \vdots \\ (\text{sigk}_p, \sigma_p) \leftarrow \text{MT.Sign}(\text{pp}, \text{sigk}_{p-1}, m_p) \\ \text{cert} \leftarrow \text{MT.Del}(\text{sigk}_p) \end{array} \right] \geq 1 - (p(\lambda) + 1) \cdot \nu(\lambda)/2. \quad (25)$$

Finally, because  $p(\lambda) = \text{poly}(\lambda)$  and  $\nu(\lambda) = \text{negl}(\lambda)$  it follows that the above expression above is at least  $1 - \text{negl}(\lambda)$ . This proves the claim.

**Many-time deletion security.** Suppose that there exists a QPT adversary  $\mathcal{A}$  which breaks the many-time deletion security of MT. In other words, there exists a polynomial  $q(\lambda)$  such that for infinitely many  $\lambda \in \mathbb{N}$ ,

$$\Pr \left[ \begin{array}{l} \top \leftarrow \text{MT.Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, S) \\ \wedge m^* \notin S \\ \wedge \top \leftarrow \text{MT.Ver}(\text{vk}, m^*, \sigma^*) \end{array} : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \text{MT.Setup}(1^\lambda) \\ (\text{vk}, \text{cert}, S, m^*, \sigma^*) \leftarrow \mathcal{A}(\text{pp}) \end{array} \right] \geq \frac{1}{q(\lambda)}. \quad (28)$$

We now show that we can use  $\mathcal{A}$  which satisfies (28) to break the one-time deletion security (according to Definition 4.1) of the OT scheme. Consider the following reduction  $\mathcal{B}$ :

1. On input  $\text{pp}$ ,  $\mathcal{B}$  runs  $\mathcal{A}(\text{pp})$  and parses the output  $(\text{vk}, \text{cert}, S, m^*, \sigma^*)$  as follows:

- $\text{vk} = \text{ot.vk}_0$
- $\text{cert} = \{\text{ot.cert}_j, m_j, \text{ot.vk}_j, \text{ot.}\sigma_j\}_{j \in [t]}$
- $\sigma^* = \{m_j^*, \text{ot.vk}_j^*, \text{ot.}\sigma_j^*\}_{j \in [t^*]}$

for some integers  $t, t^*$ . Let  $i^* \leq \min\{t, t^*\}$  be the largest integer such that  $(m_j, \text{ot.vk}_i) = (m_j^*, \text{ot.vk}_i^*)$  for all  $i \in [i^*]$ .

2.  $\mathcal{B}$  proceeds as follows (depending on which case it encounters):

**Case 1:**  $i^* = t < t^*$ . Let  $\text{cert}' = (\text{ot.cert}_{i^*+1}, m_{i^*+1}, \text{ot.vk}_{i^*+1}, \text{ot.}\sigma_{i^*+1})$  and output

$$\text{out} = (\text{ot.vk}_{i^*}, \text{cert}', \emptyset, m_{i^*+1}^*, \sigma_{i^*+1}^*).$$

**Case 2:**  $i^* < \min\{t, t^*\}$ . Let  $\text{cert}' = (\text{ot.cert}_{i^*+1}, m_{i^*+1}, \text{ot.vk}_{i^*+1}, \text{ot.}\sigma_{i^*+1})$  and output

$$\text{out} = (\text{ot.vk}_{i^*}, \text{cert}', \{m_{i^*+1}\}, m_{i^*+1}^*, \sigma_{i^*+1}^*).$$

**Case 3:** ( $i^* = t^* < t$ ) or ( $i^* = t^* = t$ ). Abort and output  $\perp$ .

We now analyze the success probability of  $\mathcal{B}$ . Note that, by our assumption in (28), either **Case 1** or **Case 2** must occur with probability at least  $1/\text{poly}(\lambda)$ . Otherwise, if **Case 3** occurs with overwhelming probability, then  $\mathcal{A}$  does not constitute a successful adversary which contradicts our initial assumption in (28).

Suppose that **Case 1** occurs. In this case, with probability at least  $1/\text{poly}(\lambda)$ , we observe that  $\text{OT.Cert}(\text{pp}, \text{vk}_{i^*}, \text{ck}, \text{cert}', S') = \top$  and  $m_{i^*+1}^* \notin S'$  for  $S' = \emptyset$ , and  $\text{OT.Ver}(\text{vk}_{i^*}, m_{i^*+1}^*, \sigma_{i^*+1}^*) = \top$ .

Finally, suppose that **Case 2** occurs. By the definition of  $i^*$ , it follows that  $(m_{i^*+1}, \text{ot.vk}_{i^*+1}) \neq (m_{i^*+1}^*, \text{ot.vk}_{i^*+1}^*)$ . Hence, with probability at least  $1/\text{poly}(\lambda)$ , we find that  $\text{OT.Cert}(\text{pp}, \text{vk}_{i^*}, \text{ck}, \text{cert}', S') = \top$  and  $m_{i^*+1}^* \notin S'$  with respect to  $S' = \{m_{i^*+1}\}$ , as well as  $\text{OT.Ver}(\text{vk}_{i^*}, m_{i^*+1}^*, \sigma_{i^*+1}^*) = \top$ .

Therefore, we have shown that there exists a QPT adversary  $\mathcal{B}$  such that

$$\Pr \left[ \begin{array}{l} \top \leftarrow \text{OT.Cert}(\text{pp}, \text{vk}, \text{ck}, \text{cert}, S) \\ \wedge m^* \notin S \\ \wedge \top \leftarrow \text{OT.Ver}(\text{vk}, m^*, \sigma^*) \\ \wedge |S| \leq 1 \end{array} : \begin{array}{l} (\text{pp}, \text{ck}) \leftarrow \text{OT.Setup}(1^\lambda) \\ (\text{vk}, \text{cert}, S, m^*, \sigma^*) \leftarrow \mathcal{B}(\text{pp}) \end{array} \right] \geq \frac{1}{\text{poly}(\lambda)}.$$

This proves the claim. □

## 5 Adaptive Hardcore Bit Property for OWFs

In this section, we introduce a new concept, which we call *adaptive hardcore bit property for OWFs*, and show it from the existence of OWFs. This property is inspired by the adaptive hardcore bit property which was shown for a family of noisy trapdoor claw-free functions by Brakerski et al. [BCM<sup>+</sup>21]. Our notion of the adaptive hardcore bit property for OWFs will be used to construct two-tier tokenized signatures.

### 5.1 Statements

The formal statement of the adaptive hardcore bit property for OWFs is given as follows. (Its proof is given later.)

**Theorem 5.1 (Adaptive Hardcore Bit Property for OWFs).** *Let  $\lambda \in \mathbb{N}$  be the security parameter and let  $\ell(\lambda), \kappa(\lambda) \in \mathbb{N}$  be polynomials. Let  $f : \{0, 1\}^{\ell(\lambda)} \rightarrow \{0, 1\}^{\kappa(\lambda)}$  be a (quantumly-secure) OWF. Then, for any QPT adversary  $\{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$ , it holds that*

$$\Pr \left[ \begin{array}{l} f(x) \in \{f(x_0), f(x_1)\} \\ \wedge \\ d \cdot (x_0 \oplus x_1) = c \end{array} \quad \begin{array}{l} x_0 \leftarrow \{0, 1\}^{\ell(\lambda)}, x_1 \leftarrow \{0, 1\}^{\ell(\lambda)} \\ : c \leftarrow \{0, 1\} \\ (x, d) \leftarrow \mathcal{A}_\lambda \left( \frac{|x_0\rangle + (-1)^c |x_1\rangle}{\sqrt{2}}, f(x_0), f(x_1) \right) \end{array} \right] \leq \frac{1}{2} + \text{negl}(\lambda). \quad (29)$$

We actually use its amplified version, which is given as follows. (Its proof is given later.)

**Theorem 5.2 (Amplified Adaptive Hardcore Bit Property for OWFs).** *Let  $\lambda \in \mathbb{N}$  be the security parameter and let  $\ell(\lambda), \kappa(\lambda), n(\lambda) \in \mathbb{N}$  be polynomials. Let  $f : \{0, 1\}^{\ell(\lambda)} \rightarrow \{0, 1\}^{\kappa(\lambda)}$  be a (quantumly-secure) OWF. Then, for any QPT adversary  $\{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$ , it holds that,*

$$\Pr \left[ \begin{array}{l} \wedge_{i \in [n]} f(x_i) \in \{f(x_i^0), f(x_i^1)\} \\ \wedge \\ \wedge_{i \in [n]} d_i \cdot (x_i^0 \oplus x_i^1) = c_i \end{array} \quad \begin{array}{l} \forall i \in [n] : x_i^0 \leftarrow \{0, 1\}^{\ell(\lambda)}, x_i^1 \leftarrow \{0, 1\}^{\ell(\lambda)} \\ : \forall i \in [n] : c_i \leftarrow \{0, 1\} \\ \{x_i, d_i\}_{i \in [n]} \leftarrow \mathcal{A}_\lambda \left( \bigotimes_{i=1}^n \frac{|x_i^0\rangle + (-1)^{c_i} |x_i^1\rangle}{\sqrt{2}}, \{f(x_i^b)\}_{i,b} \right) \end{array} \right] \leq \text{negl}(\lambda). \quad (30)$$

### 5.2 Theorem of [BKM<sup>+</sup>23]

In order to show adaptive hardcore bit property for OWFs, we use the following theorem which is implicit in [BKM<sup>+</sup>23, Theorem 3.1]. The only difference is that we additionally reveal both pre-images as part of the distribution  $\{\tilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(b)\}_{\lambda \in \mathbb{N}, b \in \{0,1\}}$ . We remark that the proof is the same.

**Theorem 5.3 (Implicit in [BKM<sup>+</sup>23], Theorem 3.1).** *Let  $\lambda \in \mathbb{N}$  be the security parameter, and let  $\ell(\lambda), \kappa(\lambda) \in \mathbb{N}$  be polynomials. Let  $f : \{0, 1\}^{\ell(\lambda)} \rightarrow \{0, 1\}^{\kappa(\lambda)}$  be a OWF secure against QPT adversaries. Let  $\{\mathcal{Z}_\lambda(\cdot, \cdot, \cdot, \cdot)\}_{\lambda \in \mathbb{N}}$  be a quantum operation with four arguments: an  $\ell(\lambda)$ -bit string  $z$ , two  $\kappa(\lambda)$ -bit strings  $y_0, y_1$ , and an  $\ell(\lambda)$ -qubit quantum state  $|\psi\rangle$ . Suppose that for any QPT adversary  $\{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$ ,  $z \in \{0, 1\}^{\ell(\lambda)}$ ,  $y_0, y_1 \in \{0, 1\}^{\kappa(\lambda)}$ , and  $\ell(\lambda)$ -qubit state  $|\psi\rangle$ ,*

$$\left| \Pr [\mathcal{A}_\lambda(\mathcal{Z}_\lambda(z, y_0, y_1, |\psi\rangle)) = 1] - \Pr [\mathcal{A}_\lambda(\mathcal{Z}_\lambda(0^{\ell(\lambda)}, y_0, y_1, |\psi\rangle)) = 1] \right| = \text{negl}(\lambda).$$

*That is,  $\mathcal{Z}_\lambda$  is semantically-secure with respect to its first input. Now, for any QPT adversary  $\{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$ , consider the distribution  $\{\tilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(b)\}_{\lambda \in \mathbb{N}, b \in \{0,1\}}$  over quantum states, obtained by running  $\mathcal{A}_\lambda$  as follows.*

- Sample  $x_0, x_1 \leftarrow \{0, 1\}^{\ell(\lambda)}$ , define  $y_0 = f(x_0), y_1 = f(x_1)$  and initialize  $\mathcal{A}_\lambda$  with

$$\mathcal{Z}_\lambda \left( x_0 \oplus x_1, y_0, y_1, \frac{|x_0\rangle + (-1)^b |x_1\rangle}{\sqrt{2}} \right).$$

- $\mathcal{A}_\lambda$ 's output is parsed as a string  $x' \in \{0, 1\}^{\ell(\lambda)}$  and a residual state on register  $A'$ .
- If  $f(x') \in \{y_0, y_1\}$ , then output  $(x_0, x_1, A')$ , and otherwise output  $\perp$ .

Then, it holds that

$$\left\| \tilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(0) - \tilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(1) \right\|_{\text{tr}} \leq \text{negl}(\lambda). \quad (31)$$

We can show the following parallel version. (It can be shown by the standard hybrid argument. A detailed proof is given in Appendix A.)

**Theorem 5.4 (Parallel version of Theorem 5.3).** *Let  $\lambda \in \mathbb{N}$  be the security parameter. Let  $\ell(\lambda), \kappa(\lambda), n(\lambda) \in \mathbb{N}$  be polynomials. Let  $f : \{0, 1\}^{\ell(\lambda)} \rightarrow \{0, 1\}^{\kappa(\lambda)}$  be a OWF secure against QPT adversaries. Let  $\{\mathcal{Z}_\lambda(\cdot, \cdot, \cdot, \cdot)\}_{\lambda \in \mathbb{N}}$  be a quantum operation with four arguments: an  $\ell(\lambda)$ -bit string  $z$ , two  $\kappa(\lambda)$ -bit strings  $y_0, y_1$ , and an  $\ell(\lambda)$ -qubit quantum state  $|\psi\rangle$ . Suppose that for any QPT adversary  $\{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$ ,  $z \in \{0, 1\}^{\ell(\lambda)}$ ,  $y_0, y_1 \in \{0, 1\}^{\kappa(\lambda)}$ , and  $\ell(\lambda)$ -qubit state  $|\psi\rangle$ ,*

$$\left| \Pr [\mathcal{A}_\lambda(\mathcal{Z}_\lambda(z, y_0, y_1, |\psi\rangle)) = 1] - \Pr [\mathcal{A}_\lambda(\mathcal{Z}_\lambda(0^{\ell(\lambda)}, y_0, y_1, |\psi\rangle)) = 1] \right| = \text{negl}(\lambda).$$

That is,  $\mathcal{Z}_\lambda$  is semantically-secure with respect to its first input. Now, for any QPT adversary  $\{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$ , consider the distribution  $\left\{ \tilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(b_1, \dots, b_{n(\lambda)}) \right\}_{\lambda \in \mathbb{N}, b_i \in \{0, 1\}}$  over quantum states, obtained by running  $\mathcal{A}_\lambda$  as follows.

- Sample  $x_i^0, x_i^1 \leftarrow \{0, 1\}^{\ell(\lambda)}$  for each  $i \in [n(\lambda)]$ , define  $y_i^0 = f(x_i^0), y_i^1 = f(x_i^1)$  and initialize  $\mathcal{A}_\lambda$  with

$$\bigotimes_{i \in [n(\lambda)]} \mathcal{Z}_\lambda \left( x_i^0 \oplus x_i^1, y_i^0, y_i^1, \frac{|x_i^0\rangle + (-1)^{b_i} |x_i^1\rangle}{\sqrt{2}} \right). \quad (32)$$

- $\mathcal{A}_\lambda$ 's output is parsed as strings  $x'_i \in \{0, 1\}^{\ell(\lambda)}$  for  $i \in [n(\lambda)]$  and a residual state on register  $A'$ .
- If  $f(x'_i) \in \{y_i^0, y_i^1\}$  for all  $i \in [n(\lambda)]$ , output  $(\{x_i^0\}_{i \in [n(\lambda)]}, \{x_i^1\}_{i \in [n(\lambda)]}, A')$ , and otherwise output  $\perp$ .

Then, there exists a negligible function  $\text{negl}(\lambda)$  such that for any  $b_1, \dots, b_{n(\lambda)} \in \{0, 1\}$ ,

$$\left\| \tilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(b_1, \dots, b_{n(\lambda)}) - \tilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(0, \dots, 0) \right\|_{\text{tr}} \leq \text{negl}(\lambda). \quad (33)$$

### 5.3 Proof of Theorem 5.1

By using Theorem 5.3, we can show Theorem 5.1 as follows. Here, we leverage the fact that any algorithm that simultaneously produces both a valid pre-image of the OWF, as well as a string which leaks information about the relative phase between the respective pre-images, must necessarily violate Theorem 5.3.

*Proof of Theorem 5.1.* Let  $\ell(\lambda), \kappa(\lambda) \in \mathbb{N}$  be polynomials, and let  $f : \{0, 1\}^{\ell(\lambda)} \rightarrow \{0, 1\}^{\kappa(\lambda)}$  be a (quantumly-secure) OWF. Suppose there exist a QPT algorithm  $\{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$  and a polynomial  $p(\lambda)$  such that, for random  $x_0, x_1 \leftarrow \{0, 1\}^\ell$  and  $c \leftarrow \{0, 1\}$ , it holds that

$$\Pr \left[ \bigwedge_{\substack{f(x) \in \{f(x_0), f(x_1)\} \\ d \cdot (x_0 \oplus x_1) = c}} : (x, d) \leftarrow \mathcal{A}_\lambda \left( \frac{|x_0\rangle + (-1)^c |x_1\rangle}{\sqrt{2}}, f(x_0), f(x_1) \right) \right] \geq \frac{1}{2} + \frac{1}{p(\lambda)} \quad (34)$$

for infinitely many  $\lambda$ . We now show how to construct an algorithm that violates Theorem 5.3. For simplicity, we define the quantum operation  $\{\mathcal{Z}_\lambda(\cdot, \cdot, \cdot, \cdot)\}_{\lambda \in \mathbb{N}}$  in Theorem 5.3 as

$$\mathcal{Z}_\lambda \left( x_0 \oplus x_1, f(x_0), f(x_1), \frac{|x_0\rangle + (-1)^c |x_1\rangle}{\sqrt{2}} \right) := \left( f(x_0), f(x_1), \frac{|x_0\rangle + (-1)^c |x_1\rangle}{\sqrt{2}} \right).$$

Evidently, our choice of  $\mathcal{Z}_\lambda$  is trivially semantically secure with respect to the first argument. Consider the following QPT algorithm  $\mathcal{B}_\lambda$ :

1. On input  $\left( f(x_0), f(x_1), \frac{|x_0\rangle + (-1)^c |x_1\rangle}{\sqrt{2}} \right)$ , run

$$(x, d_c) \leftarrow \mathcal{A}_\lambda \left( \frac{|x_0\rangle + (-1)^c |x_1\rangle}{\sqrt{2}}, f(x_0), f(x_1) \right).$$

2. Output  $x$  and assign  $|d_c\rangle\langle d_c|$  as the residual state.<sup>12</sup>

Adopting the notation from Theorem 5.3, we define  $\tilde{\mathcal{Z}}_\lambda^{\mathcal{B}_\lambda}(c)$ .<sup>13</sup> Consider the following distinguisher that distinguishes  $\tilde{\mathcal{Z}}_\lambda^{\mathcal{B}_\lambda}(c)$  for  $c \in \{0, 1\}$ :

1. Get  $\tilde{\mathcal{Z}}_\lambda^{\mathcal{B}_\lambda}(c)$  as input.
2. If it is  $\perp$ , output  $\perp$  and abort.
3. Output  $d_c \cdot (x_0 \oplus x_1) \pmod{2}$ .

From Equation (34), there exists a polynomial  $p(\lambda)$  such that both  $f(x) \in \{f(x_0), f(x_1)\}$  and  $d_c \cdot (x_0 \oplus x_1) = c \pmod{2}$  occur with probability at least  $\frac{1}{2} + \frac{1}{p(\lambda)}$ . Thus, the distinguisher can distinguish  $\tilde{\mathcal{Z}}_\lambda^{\mathcal{B}_\lambda}(0)$  and  $\tilde{\mathcal{Z}}_\lambda^{\mathcal{B}_\lambda}(1)$  with probability at least  $\frac{1}{2} + \frac{1}{p(\lambda)}$ , but this means

$$\left\| \tilde{\mathcal{Z}}_\lambda^{\mathcal{B}_\lambda}(0) - \tilde{\mathcal{Z}}_\lambda^{\mathcal{B}_\lambda}(1) \right\|_{\text{tr}} \geq \frac{2}{p(\lambda)}.$$

from Theorem 2.2. This violates Theorem 5.3. □

<sup>12</sup>Note that we can think of  $d_c$  as a classical mixture (i.e., density matrix) over the randomness of  $x_0, x_1 \leftarrow \{0, 1\}^\ell$ ,  $c \leftarrow \{0, 1\}$  and the internal randomness of the algorithm  $\mathcal{A}_\lambda$ .

<sup>13</sup>It is, roughly speaking,  $|x_0\rangle\langle x_0| \otimes |x_1\rangle\langle x_1| \otimes |d_c\rangle\langle d_c|$  for  $c \in \{0, 1\}$  when  $f(x) \in \{f(x_0), f(x_1)\}$ , and is  $\perp$  when  $f(x) \notin \{f(x_0), f(x_1)\}$ .

## 5.4 Proof of Theorem 5.2

In this subsection, we show Theorem 5.2 by using Theorem 5.4.

*Proof of Theorem 5.2.* For the sake of contradiction, assume that there is a QPT adversary  $\{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$  such that

$$\Pr \left[ \begin{array}{l} \bigwedge_{i \in [n]} f(x_i) \in \{f(x_i^0), f(x_i^1)\} \\ \bigwedge_{i \in [n]} d_i \cdot (x_i^0 \oplus x_i^1) = c_i \end{array} : \begin{array}{l} \forall i \in [n] : x_i^0 \leftarrow \{0, 1\}^\ell, x_i^1 \leftarrow \{0, 1\}^\ell, c_i \leftarrow \{0, 1\} \\ \{x_i, d_i\}_{i \in [n]} \leftarrow \mathcal{A}_\lambda \left( \bigotimes_{i=1}^n \frac{|x_i^0\rangle + (-1)^{c_i} |x_i^1\rangle}{\sqrt{2}}, \{f(x_i^b)\}_{i,b} \right) \end{array} \right] \geq \frac{1}{\text{poly}(\lambda)} \quad (35)$$

for infinitely many  $\lambda$ . We consider the quantum operation  $\{\mathcal{Z}_\lambda(\cdot, \cdot, \cdot, \cdot)\}_{\lambda \in \mathbb{N}}$  in Theorem 5.4 as

$$\mathcal{Z}_\lambda \left( x_0 \oplus x_1, f(x_0), f(x_1), \frac{|x_0\rangle + (-1)^c |x_1\rangle}{\sqrt{2}} \right) := \left( f(x_0), f(x_1), \frac{|x_0\rangle + (-1)^c |x_1\rangle}{\sqrt{2}} \right), \quad (36)$$

which is trivially semantically secure with respect to the first argument. From such  $\{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$  and  $\{\mathcal{Z}_\lambda\}_{\lambda \in \mathbb{N}}$ , we construct the following QPT adversary  $\{\mathcal{B}_\lambda\}_{\lambda \in \mathbb{N}}$  for fixed each  $(c_1, \dots, c_n) \in \{0, 1\}^n$ :

1. Get  $\{f(x_i^b)\}_{i \in [n], b \in \{0,1\}}$  and  $\bigotimes_{i \in [n]} \frac{|x_i^0\rangle + (-1)^{c_i} |x_i^1\rangle}{\sqrt{2}}$  as input.
2. Run  $(\{x_i\}_{i \in [n]}, \{d_i\}_{i \in [n]}) \leftarrow \mathcal{A}_\lambda \left( \bigotimes_{i=1}^n \frac{|x_i^0\rangle + (-1)^{c_i} |x_i^1\rangle}{\sqrt{2}}, \{f(x_i^b)\}_{i \in [n], b \in \{0,1\}} \right)$ .
3. Output  $\{x_i\}_{i \in [n]}$ . Set its residual state as  $\bigotimes_{i \in [n]} |d_i\rangle \langle d_i|$ .

Then, by using the notation of Theorem 5.4, we define  $\tilde{\mathcal{Z}}_\lambda^{\mathcal{B}_\lambda}(c_1, \dots, c_n)$ .<sup>14</sup> Let us consider the following QPT distinguisher  $\{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$ :

1. Get  $\tilde{\mathcal{Z}}_\lambda^{\mathcal{B}_\lambda}(c_1, \dots, c_n)$  as input.
2. If it is  $\perp$ , output  $\perp$ . Otherwise, parse it as  $\left( \bigotimes_{i \in [n], b \in \{0,1\}} |x_i^b\rangle \langle x_i^b| \right) \otimes \left( \bigotimes_{i \in [n]} |d_i\rangle \langle d_i| \right)$ .
3. Compute  $c'_i := d_i \cdot (x_i^0 \oplus x_i^1)$  for each  $i \in [n]$ . Output  $\{c'_i\}_{i \in [n]}$ .

Then, from Equation (35),

$$\frac{1}{2^n} \sum_{(c_1, \dots, c_n) \in \{0,1\}^n} \Pr[(c_1, \dots, c_n) \leftarrow \mathcal{D}(\tilde{\mathcal{Z}}_\lambda^{\mathcal{B}_\lambda}(c_1, \dots, c_n))] \geq \frac{1}{\text{poly}(\lambda)} \quad (37)$$

for infinitely many  $\lambda$ . Now we show that it contradicts Theorem 5.4.

If Theorem 5.4 is correct, there exists a negligible function  $\text{negl}$  such that

$$\left\| \tilde{\mathcal{Z}}_\lambda^{\mathcal{B}_\lambda}(c_1, \dots, c_n) - \tilde{\mathcal{Z}}_\lambda^{\mathcal{B}_\lambda}(0, \dots, 0) \right\|_{\text{tr}} \leq \text{negl}(\lambda) \quad (38)$$

<sup>14</sup>Roughly speaking, it is  $\left( \bigotimes_{i \in [n], b \in \{0,1\}} |x_i^b\rangle \langle x_i^b| \right) \otimes \left( \bigotimes_{i \in [n]} |d_i\rangle \langle d_i| \right)$  if  $f(x_i) \in \{f(x_i^0), f(x_i^1)\}$  for all  $i \in [n]$ , and it is  $\perp$  otherwise.

for all  $(c_1, \dots, c_n) \in \{0, 1\}^n$ . However, in that case, there exists a negligible function  $\text{negl}$  such that

$$\left| \Pr \left[ (c_1, \dots, c_n) \leftarrow \mathcal{D}(\tilde{\mathcal{Z}}_\lambda^{\mathcal{B}^\lambda}(c_1, \dots, c_n)) \right] - \Pr \left[ (c_1, \dots, c_n) \leftarrow \mathcal{D}(\tilde{\mathcal{Z}}_\lambda^{\mathcal{B}^\lambda}(0, \dots, 0)) \right] \right| \leq \text{negl}(\lambda) \quad (39)$$

for all  $(c_1, \dots, c_n) \in \{0, 1\}^n$ . Then we have

$$\frac{1}{\text{poly}(\lambda)} \leq \frac{1}{2^n} \sum_{(c_1, \dots, c_n) \in \{0, 1\}^n} \Pr[(c_1, \dots, c_n) \leftarrow \mathcal{D}(\tilde{\mathcal{Z}}_\lambda^{\mathcal{B}^\lambda}(c_1, \dots, c_n))] \quad (40)$$

$$\leq \frac{1}{2^n} \sum_{(c_1, \dots, c_n) \in \{0, 1\}^n} \left( \Pr[(c_1, \dots, c_n) \leftarrow \mathcal{D}(\tilde{\mathcal{Z}}_\lambda^{\mathcal{B}^\lambda}(0, \dots, 0))] + \text{negl}(\lambda) \right) \quad (41)$$

$$\leq \frac{1}{2^n} \sum_{(c_1, \dots, c_n) \in \{0, 1\}^n} \Pr[(c_1, \dots, c_n) \leftarrow \mathcal{D}(\tilde{\mathcal{Z}}_\lambda^{\mathcal{B}^\lambda}(0, \dots, 0))] + \text{negl}(\lambda) \quad (42)$$

$$\leq \frac{1}{2^n} + \text{negl}(\lambda) \quad (43)$$

for infinitely many  $\lambda$ , which is the contradiction. Here, the first inequality is from Equation (37), the second inequality is from Equation (39), and the last inequality is from the fact that  $\sum_{(c_1, \dots, c_n) \in \{0, 1\}^n} \Pr[(c_1, \dots, c_n) \leftarrow \mathcal{A}] = 1$  for any algorithm  $\mathcal{A}$ . □

## 6 Two-Tier Tokenized Signatures

In this section, we will first give the formal definition of two-tier tokenized signatures (2-TS), and then show that they can be constructed from OWFs. For the construction, we use the (amplified) adaptive hardcore bit property for OWFs (Theorem 5.2).

### 6.1 Definition

The formal definition is as follows.

**Definition 6.1 (Two-Tier Tokenized Signatures (2-TS)).** A two-tier tokenized signature scheme is a tuple  $(\text{KeyGen}, \text{StateGen}, \text{Sign}, \text{Ver}_0, \text{Ver}_1)$  of algorithms such that

- $\text{KeyGen}(1^\lambda) \rightarrow (\text{sk}, \text{pk})$  : It is a QPT algorithm that, on input the security parameter  $\lambda$ , outputs a classical secret key  $\text{sk}$  and a classical public key  $\text{pk}$ .
- $\text{StateGen}(\text{sk}) \rightarrow \psi$  : It is a QPT algorithm that, on input  $\text{sk}$ , outputs a quantum state  $\psi$ .
- $\text{Sign}(\psi, m) \rightarrow \sigma$  : It is a QPT algorithm that, on input  $\psi$  and a message  $m \in \{0, 1\}$ , outputs a classical signature  $\sigma$ .
- $\text{Ver}_0(\text{pk}, \sigma) \rightarrow \top/\perp$  : It is a QPT algorithm that, on input  $\text{pk}$  and  $\sigma$ , outputs  $\top/\perp$ .
- $\text{Ver}_1(\text{sk}, \sigma) \rightarrow \top/\perp$  : It is a QPT algorithm that, on input  $\text{sk}$  and  $\sigma$ , outputs  $\top/\perp$ .

We require the following properties.

**Correctness:**

$$\Pr \left[ \top \leftarrow \text{Ver}_0(\text{pk}, \sigma) : \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda) \\ \psi \leftarrow \text{StateGen}(\text{sk}) \\ \sigma \leftarrow \text{Sign}(\psi, 0) \end{array} \right] \geq 1 - \text{negl}(\lambda) \quad (44)$$

and

$$\Pr \left[ \top \leftarrow \text{Ver}_1(\text{sk}, \sigma) : \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda) \\ \psi \leftarrow \text{StateGen}(\text{sk}) \\ \sigma \leftarrow \text{Sign}(\psi, 1) \end{array} \right] \geq 1 - \text{negl}(\lambda). \quad (45)$$

**Security:** For any QPT adversary  $\mathcal{A}$ ,

$$\Pr \left[ \top \leftarrow \text{Ver}_0(\text{pk}, \sigma_0) \wedge \top \leftarrow \text{Ver}_1(\text{sk}, \sigma_1) : \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda) \\ \psi \leftarrow \text{StateGen}(\text{sk}) \\ (\sigma_0, \sigma_1) \leftarrow \mathcal{A}(\psi, \text{pk}) \end{array} \right] \leq \text{negl}(\lambda). \quad (46)$$

We can show that the following type of security, which we call *one-wayness*, is also satisfied by two-tier tokenized signatures.

**Lemma 6.2 (One-wayness of two-tier tokenized signatures).** For any QPT adversary  $\mathcal{A}$ ,

$$\Pr \left[ \top \leftarrow \text{Ver}_0(\text{pk}, \sigma) : \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda) \\ \psi \leftarrow \mathcal{A}(\text{pk}) \\ \sigma \leftarrow \text{Sign}(\psi, 0) \end{array} \right] \leq \text{negl}(\lambda). \quad (47)$$

*Proof.* Assume that there exists a QPT adversary  $\mathcal{A}$  such that

$$\Pr \left[ \top \leftarrow \text{Ver}_0(\text{pk}, \sigma) : \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda) \\ \psi \leftarrow \mathcal{A}(\text{pk}) \\ \sigma \leftarrow \text{Sign}(\psi, 0) \end{array} \right] \geq \frac{1}{\text{poly}(\lambda)} \quad (48)$$

for infinitely many  $\lambda$ . Then, from such  $\mathcal{A}$ , we can construct a QPT adversary  $\mathcal{B}$  that breaks the security of the two-tier tokenized signature scheme as follows:

1. Get  $\psi$  and  $\text{pk}$  as input.
2. Run  $\psi' \leftarrow \mathcal{A}(\text{pk})$ .
3. Run  $\sigma_0 \leftarrow \text{Sign}(\psi', 0)$  and  $\sigma_1 \leftarrow \text{Sign}(\psi, 1)$ .
4. Output  $(\sigma_0, \sigma_1)$ .

It is clear that  $\mathcal{B}$  breaks the security of the two-tier tokenized signature scheme.  $\square$

## 6.2 Construction

We show that 2-TS can be constructed from OWFs.

**Theorem 6.3.** *If OWFs exist, then two-tier tokenized signatures exist.*

*Proof.* Let  $f$  be a OWF. From it, we construct a two-tier tokenized signature scheme as follows:

- $\text{KeyGen}(1^\lambda) \rightarrow (\text{sk}, \text{pk})$  : Choose  $x_i^0, x_i^1 \leftarrow \{0, 1\}^\ell$  for each  $i \in [n]$ . Choose  $c_i \leftarrow \{0, 1\}$  for each  $i \in [n]$ . Output  $\text{sk} := \{c_i, x_i^0, x_i^1\}_{i \in [n]}$  and  $\text{pk} := \{f(x_i^0), f(x_i^1)\}_{i \in [n]}$ .
- $\text{StateGen}(\text{sk}) \rightarrow \psi$  : Parse  $\text{sk} = \{c_i, x_i^0, x_i^1\}_{i \in [n]}$ . Output  $\psi := \bigotimes_{i \in [n]} \frac{|x_i^0\rangle + (-1)^{c_i} |x_i^1\rangle}{\sqrt{2}}$ .
- $\text{Sign}(\psi, m) \rightarrow \sigma$  : If  $m = 0$ , measure  $\psi$  in the computational basis to get the result  $\{z_i\}_{i \in [n]}$  (where  $z_i \in \{0, 1\}^\ell$  for each  $i \in [n]$ ), and output it as  $\sigma$ . If  $m = 1$ , measure  $\psi$  in the Hadamard basis to get the result  $\{d_i\}_{i \in [n]}$  (where  $d_i \in \{0, 1\}^\ell$  for each  $i \in [n]$ ), and output it as  $\sigma$ .
- $\text{Ver}_0(\text{pk}, \sigma) \rightarrow \top/\perp$  : Parse  $\text{pk} = \{f(x_i^0), f(x_i^1)\}_{i \in [n]}$  and  $\sigma = \{z_i\}_{i \in [n]}$ . If  $f(z_i) \in \{f(x_i^0), f(x_i^1)\}$  for all  $i \in [n]$ , output  $\top$ . Otherwise, output  $\perp$ .
- $\text{Ver}_1(\text{sk}, \sigma) \rightarrow \top/\perp$  : Parse  $\text{sk} = \{c_i, x_i^0, x_i^1\}_{i \in [n]}$  and  $\sigma = \{d_i\}_{i \in [n]}$ . If  $d_i \cdot (x_i^0 \oplus x_i^1) = c_i$  for all  $i \in [n]$ , output  $\top$ . Otherwise, output  $\perp$ .

The correctness is clear. The security is also clear from Theorem 5.2. □

## 7 Digital Signatures with Revocable Signatures

In this section, we define digital signatures with revocable signatures (DSR-Sign). We also show that it can be constructed from 2-TS, and therefore from OWFs.

### 7.1 Definition

We first give its formal definition as follows.

**Definition 7.1 (Digital Signatures with Revocable Signatures (DSR-Sign)).** *A digital signature scheme with revocable signatures is a set  $(\text{KeyGen}, \text{Sign}, \text{Ver}, \text{Del}, \text{Cert})$  of algorithms that satisfy the following.*

- $\text{KeyGen}(1^\lambda) \rightarrow (\text{sigk}, \text{vk})$  : *It is a QPT algorithm that, on input the security parameter  $\lambda$ , outputs a classical signing key  $\text{sigk}$  and a classical public verification key  $\text{vk}$ .*
- $\text{Sign}(\text{sigk}, m) \rightarrow (\psi, \text{ck})$  : *It is a QPT algorithm that, on input a message  $m$  and  $\text{sigk}$ , outputs a quantum signature  $\psi$  and a classical check key  $\text{ck}$ .*
- $\text{Ver}(\text{vk}, \psi, m) \rightarrow \top/\perp$  : *It is a QPT algorithm that, on input  $\text{vk}$ ,  $m$ , and  $\psi$ , outputs  $\top/\perp$ .*
- $\text{Del}(\psi) \rightarrow \text{cert}$  : *It is a QPT algorithm that, on input  $\psi$ , outputs a classical certificate  $\text{cert}$ .*
- $\text{Cert}(\text{ck}, \text{cert}) \rightarrow \top/\perp$  : *It is a QPT algorithm that, on input  $\text{ck}$  and  $\text{cert}$ , outputs  $\top/\perp$ .*

*We require the following properties.*

**Correctness:** For any message  $m$ ,

$$\Pr \left[ \top \leftarrow \text{Ver}(\text{vk}, \psi, m) : \begin{array}{l} (\text{sigk}, \text{vk}) \leftarrow \text{KeyGen}(1^\lambda) \\ (\psi, \text{ck}) \leftarrow \text{Sign}(\text{sigk}, m) \end{array} \right] \geq 1 - \text{negl}(\lambda). \quad (49)$$

**Deletion correctness:** For any message  $m$ ,

$$\Pr \left[ \top \leftarrow \text{Cert}(\text{ck}, \text{cert}) : \begin{array}{l} (\text{sigk}, \text{vk}) \leftarrow \text{KeyGen}(1^\lambda) \\ (\psi, \text{ck}) \leftarrow \text{Sign}(\text{sigk}, m) \\ \text{cert} \leftarrow \text{Del}(\psi) \end{array} \right] \geq 1 - \text{negl}(\lambda). \quad (50)$$

**Many-time deletion security:** For any adversary  $\mathcal{A}$  consisting of a pair of QPT algorithms  $(\mathcal{A}_1, \mathcal{A}_2)$ :

$$\Pr \left[ \top \leftarrow \text{Cert}(\text{ck}^*, \text{cert}) \wedge \top \leftarrow \text{Ver}(\text{vk}, \psi, m^*) : \begin{array}{l} (\text{sigk}, \text{vk}) \leftarrow \text{KeyGen}(1^\lambda) \\ (m^*, \text{st}) \leftarrow \mathcal{A}_1^{\text{Sign}(\text{sigk}, \cdot)}(\text{vk}) \\ (\psi^*, \text{ck}^*) \leftarrow \text{Sign}(\text{sigk}, m^*) \\ (\text{cert}, \psi) \leftarrow \mathcal{A}_2^{\text{Sign}(\text{sigk}, \cdot)}(\text{st}, \psi^*) \end{array} \right] \leq \text{negl}(\lambda), \quad (51)$$

where  $\mathcal{A}$  is not allowed to query  $m^*$  to the signing oracle.

*Remark 7.2.* The above definition does not capture the situation where the adversary gets more than one signatures on  $m^*$  but deleted all of them. Actually, our construction seems to also satisfy security in such a setting. However, we choose to not formalize it for simplicity.

*Remark 7.3.* We can define the standard EUF-CMA security as follows, but it is trivially implied by many-time deletion security, and therefore we do not include EUF-CMA security in the definition of digital signatures with revocable signatures.

**Definition 7.4 (EUF-CMA Security).** For any QPT adversary  $\mathcal{A}$ ,

$$\Pr \left[ \top \leftarrow \text{Ver}(\text{vk}, \psi^*, m^*) : \begin{array}{l} (\text{sigk}, \text{vk}) \leftarrow \text{KeyGen}(1^\lambda) \\ (m^*, \psi^*) \leftarrow \mathcal{A}^{\text{Sign}(\text{sigk}, \cdot)}(\text{vk}) \end{array} \right] \leq \text{negl}(\lambda), \quad (52)$$

where  $\mathcal{A}$  is not allowed to query  $m^*$  to the signing oracle.

We define a weaker version of many-time deletion security, which we call no-query deletion security as follows.

**Definition 7.5 (No-Query Deletion Security).** It is the same as many-time deletion security, Equation (51), except that  $\mathcal{A}$  cannot query the signing oracle.

The no-query security notion actually implies the many-time case:

**Lemma 7.6 (Many-Time Deletion Security from No-Query Deletion Security).** Assume that EUF-CMA secure digital signature schemes exist. Then following holds: if there exists a digital signature scheme with revocable signatures which satisfies no-query deletion security, then there is a scheme that satisfies many-time deletion security.

*Proof.* Let  $(\text{NQ.KeyGen}, \text{NQ.Sign}, \text{NQ.Ver}, \text{NQ.Del}, \text{NQ.Cert})$  be a digital signature scheme with revocable signatures that satisfies no-query deletion security. Let  $(\text{MT.KeyGen}, \text{MT.Sign}, \text{MT.Ver})$  be a plain EUF-CMA secure digital signature scheme. From them, we can construct a digital signature scheme  $\Sigma := (\text{KeyGen}, \text{Sign}, \text{Ver}, \text{Del}, \text{Cert})$  with revocable signatures that satisfies many-time deletion security as follows.

- $\text{KeyGen}(1^\lambda) \rightarrow (\text{sigk}, \text{vk})$  : Run  $(\text{mt.sigk}, \text{mt.vk}) \leftarrow \text{MT.KeyGen}(1^\lambda)$ . Output  $\text{sigk} := \text{mt.sigk}$  and  $\text{vk} := \text{mt.vk}$ .
- $\text{Sign}(\text{sigk}, m) \rightarrow (\psi, \text{ck})$  : Parse  $\text{sigk} = \text{mt.sigk}$ . Run  $(\text{nq.sigk}, \text{nq.vk}) \leftarrow \text{NQ.KeyGen}(1^\lambda)$ . Run  $(\phi, \text{nq.ck}) \leftarrow \text{NQ.Sign}(\text{nq.sigk}, m)$ . Run  $\sigma \leftarrow \text{MT.Sign}(\text{mt.sigk}, \text{nq.vk} \| m)$ . Output  $\psi := (\phi, \sigma, \text{nq.vk})$  and  $\text{ck} := \text{nq.ck}$ .
- $\text{Ver}(\text{vk}, \psi, m) \rightarrow \top / \perp$  : Parse  $\text{vk} = \text{mt.vk}$  and  $\psi = (\phi, \sigma, \text{nq.vk})$ . Run  $\text{MT.Ver}(\text{mt.vk}, \sigma, \text{nq.vk} \| m)$ . If the output is  $\perp$ , output  $\perp$  and abort. Run  $\text{NQ.Ver}(\text{nq.vk}, \phi, m)$ . If the output is  $\top$ , output  $\top$ . Otherwise, output  $\perp$ .
- $\text{Del}(\psi) \rightarrow \text{cert}$  : Parse  $\psi = (\phi, \sigma, \text{nq.vk})$ . Run  $\text{cert}' \leftarrow \text{NQ.Del}(\phi)$ . Output  $\text{cert} := \text{cert}'$ .
- $\text{Cert}(\text{ck}, \text{cert}) \rightarrow \top / \perp$  : Parse  $\text{ck} = \text{nq.ck}$ . Run  $\text{NQ.Cert}(\text{nq.ck}, \text{cert})$ , and output its output.

We show that  $\Sigma$  satisfies many-time deletion security. In other words, we show that if the many-time deletion security of  $\Sigma$  is broken, then either the no-query deletion security of the digital signature scheme NQ is broken or the EUF-CMA security of the digital signature scheme MT is broken. Assume that there exists a pair of QPT algorithms  $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$  such that

$$\Pr \left[ \begin{array}{l} \top \leftarrow \text{Cert}(\text{ck}^*, \text{cert}) \wedge \top \leftarrow \text{Ver}(\text{vk}, \psi, m^*) : \\ \begin{array}{l} (\text{sigk}, \text{vk}) \leftarrow \text{KeyGen}(1^\lambda) \\ (m^*, \text{st}) \leftarrow \mathcal{A}_1^{\text{Sign}(\text{sigk}, \cdot)}(\text{vk}) \\ (\psi^*, \text{ck}^*) \leftarrow \text{Sign}(\text{sigk}, m^*) \\ (\text{cert}, \psi) \leftarrow \mathcal{A}_2^{\text{Sign}(\text{sigk}, \cdot)}(\text{st}, \psi^*) \end{array} \end{array} \right] \geq \frac{1}{\text{poly}(\lambda)} \quad (53)$$

for infinitely many  $\lambda$ , where  $\mathcal{A}$  is not allowed to query  $m^*$  to the signing oracle. From such  $\mathcal{A}$ , we construct a QPT adversary  $\mathcal{B}$  that breaks the no-query deletion security of the scheme NQ as follows: Let  $\mathcal{C}$  be the challenger of the security game of the no-query deletion security.

1.  $\mathcal{C}$  runs  $(\text{nq.sigk}^*, \text{nq.vk}^*) \leftarrow \text{NQ.KeyGen}(1^\lambda)$ .
2.  $\mathcal{C}$  sends  $\text{nq.vk}^*$  to  $\mathcal{B}$ .
3.  $\mathcal{B}$  runs  $(\text{mt.sigk}, \text{mt.vk}) \leftarrow \text{MT.KeyGen}(1^\lambda)$ .
4.  $\mathcal{B}$  runs  $(m^*, \text{st}) \leftarrow \mathcal{A}_1^{\text{Sign}(\text{sigk}, \cdot)}(\text{mt.vk})$ . When  $\mathcal{A}_1$  queries  $m$  to the signing oracle,  $\mathcal{B}$  simulates it as follows:
  - (a) Run  $(\text{nq.sigk}, \text{nq.vk}) \leftarrow \text{NQ.KeyGen}(1^\lambda)$ .
  - (b) Run  $(\phi, \text{nq.ck}) \leftarrow \text{NQ.Sign}(\text{nq.sigk}, m)$ .
  - (c) Run  $\sigma \leftarrow \text{MT.Sign}(\text{mt.sigk}, \text{nq.vk} \| m)$ .
  - (d) Output  $\psi := (\phi, \sigma, \text{nq.vk})$  and  $\text{ck} := \text{nq.ck}$ .
5.  $\mathcal{B}$  sends  $m^*$  to  $\mathcal{C}$ .

6.  $\mathcal{C}$  runs  $(\phi^*, \text{nq.ck}^*) \leftarrow \text{NQ.Sign}(\text{nq.sigk}^*, m^*)$ , and sends  $\phi^*$  to  $\mathcal{B}$ .
7.  $\mathcal{B}$  runs  $\sigma^* \leftarrow \text{MT.Sign}(\text{mt.sigk}, \text{nq.vk}^* \| m^*)$ .
8.  $\mathcal{B}$  runs  $(\text{cert}, \psi) \leftarrow \mathcal{A}_2^{\text{Sign}(\text{sigk}, \cdot)}((\phi^*, \sigma^*, \text{nq.vk}^*))$ . When  $\mathcal{A}_2$  queries  $m$  to the signing oracle,  $\mathcal{B}$  simulates it as follows:
  - (a) Run  $(\text{nq.sigk}, \text{nq.vk}) \leftarrow \text{NQ.KeyGen}(1^\lambda)$ .
  - (b) Run  $(\phi, \text{nq.ck}) \leftarrow \text{NQ.Sign}(\text{nq.sigk}, m)$ .
  - (c) Run  $\sigma \leftarrow \text{MT.Sign}(\text{mt.sigk}, \text{nq.vk} \| m)$ .
  - (d) Output  $\psi := (\phi, \sigma, \text{nq.vk})$  and  $\text{ck} := \text{nq.ck}$ .
9. Parse  $\psi = (\phi, \sigma, \eta)$ .  $\mathcal{B}$  outputs  $\text{cert}$  and  $\phi$ .

Due to the EUF-CMA security of the scheme MT,  $\top \leftarrow \text{MT.Ver}(\text{mt.vk}, \sigma, \eta \| m^*)$  occurs only when  $\eta = \text{nq.vk}^*$  except for a negligible probability. Therefore, Equation (53) means that both  $\Pr[\top \leftarrow \text{NQ.Ver}(\text{nq.vk}^*, \phi, m^*)]$  and  $\Pr[\top \leftarrow \text{NQ.Cert}(\text{nq.ck}^*, \text{cert})]$  are non-negligible for the above  $\mathcal{B}$ , which breaks the no-query deletion security of the scheme NQ. □

## 7.2 Construction

Here we show the following result.

**Theorem 7.7.** *If two-tier tokenized signatures exist, then digital signatures with revocable signatures that satisfy no-query deletion security exist.*

From Lemma 7.6, it also means the following:

**Corollary 7.8.** *Digital signatures with revocable signatures (that satisfy many-time deletion security) exist if two-tier tokenized signatures and EUF-CMA secure digital signatures exist.*

*Proof of Theorem 7.7.* Here, we construct the scheme for the single-bit message space. It is clear that this can be extended to any fixed multi-bit message space case by the parallel execution of the protocol. Moreover, by using universal one-way hash functions, it can be extended to unbounded poly-length message space case [NY89].

Let  $(\text{TS.KeyGen}, \text{TS.StateGen}, \text{TS.Sign}, \text{TS.Ver}_0, \text{TS.Ver}_1)$  be a two-tier tokenized signature scheme. From it, we construct a digital signature scheme with revocable signatures  $\Sigma := (\text{KeyGen}, \text{Sign}, \text{Ver}, \text{Del}, \text{Cert})$  that satisfies no-query deletion security for the single bit message space as follows.

- $\text{KeyGen}(1^\lambda) \rightarrow (\text{sigk}, \text{vk})$  : Run  $(\text{sk}_0, \text{pk}_0) \leftarrow \text{TS.KeyGen}(1^\lambda)$ . Run  $(\text{sk}_1, \text{pk}_1) \leftarrow \text{TS.KeyGen}(1^\lambda)$ . Output  $\text{sigk} := (\text{sk}_0, \text{sk}_1)$  and  $\text{vk} := (\text{pk}_0, \text{pk}_1)$ .
- $\text{Sign}(\text{sigk}, m) \rightarrow (\psi, \text{ck})$  : Parse  $\text{sigk} = (\text{sk}_0, \text{sk}_1)$ . Run  $\psi' \leftarrow \text{TS.StateGen}(\text{sk}_m)$ . Output  $\psi := \psi'$  and  $\text{ck} := \text{sk}_m$ .
- $\text{Ver}(\text{vk}, \psi, m) \rightarrow \top/\perp$  : Parse  $\text{vk} := (\text{pk}_0, \text{pk}_1)$ . Run  $\sigma \leftarrow \text{TS.Sign}(\psi, 0)$ . Run  $\text{TS.Ver}_0(\text{pk}_m, \sigma)$ , and output its output.<sup>15</sup>

<sup>15</sup>The verification algorithm destroys the signature, but it can be done in a non-destructive way by coherently applying this procedure and then doing the uncomputation.

- $\text{Del}(\psi) \rightarrow \text{cert} : \text{Run } \sigma \leftarrow \text{TS.Sign}(\psi, 1)$ , and output  $\text{cert} := \sigma$ .
- $\text{Cert}(\text{ck}, \text{cert}) \rightarrow \top/\perp : \text{Parse } \text{ck} = \text{sk}_m$ . Run  $\text{TS.Ver}_1(\text{sk}_m, \text{cert})$ , and output its output.

Correctness and the deletion correctness are clear. Let us show the no-query deletion security. Assume that there is a pair of QPT algorithms  $(\mathcal{A}_1, \mathcal{A}_2)$  such that

$$\Pr \left[ \begin{array}{l} \top \leftarrow \text{Cert}(\text{ck}^*, \text{cert}) \wedge \top \leftarrow \text{Ver}(\text{vk}, \psi, m^*) : \\ \begin{array}{l} (\text{sigk}, \text{vk}) \leftarrow \text{KeyGen}(1^\lambda) \\ (m^*, \text{st}) \leftarrow \mathcal{A}_1(\text{vk}) \\ (\psi^*, \text{ck}^*) \leftarrow \text{Sign}(\text{sigk}, m^*) \\ (\text{cert}, \psi) \leftarrow \mathcal{A}_2(\text{st}, \psi^*) \end{array} \end{array} \right] \geq \frac{1}{\text{poly}(\lambda)} \quad (54)$$

for infinitely many  $\lambda$ . From  $\mathcal{A}$ , we can construct a QPT adversary  $\mathcal{B}$  that breaks the original two-tier tokenized signature scheme as follows:

1. Get  $\psi^*$  and  $\text{pk}$  as input.
2. Run  $(\text{sk}', \text{pk}') \leftarrow \text{TS.KeyGen}(1^\lambda)$ . Choose  $r \leftarrow \{0, 1\}$ . If  $r = 0$ , set  $\text{vk} := (\text{pk}, \text{pk}')$ . If  $r = 1$ , set  $\text{vk} := (\text{pk}', \text{pk})$ .
3. Run  $(m^*, \text{st}) \leftarrow \mathcal{A}_1(\text{vk})$ . If  $r \neq m^*$ , output  $\perp$  and abort.
4. Run  $(\text{cert}, \psi) \leftarrow \mathcal{A}_2(\text{st}, \psi^*)$ .
5. Run  $\sigma_0 \leftarrow \text{TS.Sign}(\psi, 0)$ . Define  $\sigma_1 := \text{cert}$ .
6. Output  $(\sigma_0, \sigma_1)$ .

It is clear that  $\Pr[\mathcal{B} \text{ breaks the two-tier tokenized signature scheme}] \geq \frac{1}{2} \Pr[\mathcal{A} \text{ breaks } \Sigma]$ . Therefore, from Equation (54),  $\mathcal{B}$  breaks the two-tier tokenized signature scheme.  $\square$

**Acknowledgments.** TM is supported by JST CREST JPMJCR2313, JST Moonshot R&D JPMJMS2061-5-1-1, JST FOREST, MEXT QLEAP, the Grant-in-Aid for Scientific Research (B) No.JP19H04066, the Grant-in-Aid for Transformative Research Areas (A) 21H05183, and the Grant-in-Aid for Scientific Research (A) No.22H00522.

## References

- [Aar09] Scott Aaronson. Quantum copy-protection and quantum money. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 229–242. IEEE Computer Society, 2009. (Cited on page 3, 7.)
- [AGKZZ20] Ryan Amos, Marios Georgiou, Aggelos Kiayias, and Mark Zhandry. One-shot signatures and applications to hybrid quantum/classical authentication. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *52nd ACM STOC*, pages 255–268. ACM Press, June 2020. (Cited on page 5, 6, 11, 14.)
- [AKN<sup>+</sup>23] Shweta Agrawal, Fuyuki Kitagawa, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Public key encryption with secure key leasing. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part I*, volume 14004 of *LNCS*, pages 581–610. Springer, Heidelberg, April 2023. (Cited on page 3, 7.)

- [AL21] Prabhanjan Ananth and Rolando L. La Placa. Secure software leasing. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 501–530. Springer, Heidelberg, October 2021. (Cited on page 3, 7.)
- [ALL<sup>+</sup>21] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 526–555, Virtual Event, August 2021. Springer, Heidelberg. (Cited on page 3, 7.)
- [APV23] Prabhanjan Ananth, Alexander Poremba, and Vinod Vaikuntanathan. Revocable cryptography from learning with errors. In Guy Rothblum and Hoeteck Wee, editors, *Theory of Cryptography*, pages 93–122, Cham, 2023. Springer Nature Switzerland. (Cited on page 3, 7.)
- [BCM<sup>+</sup>21] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. *Journal of the ACM*, 68(5):31:1–31:47, 2021. (Cited on page 6, 9, 10, 23.)
- [BDS23] Shalev Ben-David and Or Sattath. Quantum tokens for digital signatures. *Quantum*, 2023. (Cited on page 4, 6.)
- [BGG<sup>+</sup>23] James Bartusek, Sanjam Garg, Vipul Goyal, Dakshita Khurana, Giulio Malavolta, Justin Raizes, and Bhaskar Roberts. Obfuscation and outsourced computation with certified deletion. Cryptology ePrint Archive, Paper 2023/265, 2023. <https://eprint.iacr.org/2023/265>. (Cited on page 7.)
- [BI20] Anne Broadbent and Rabib Islam. Quantum encryption with certified deletion. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 92–122. Springer, Heidelberg, November 2020. (Cited on page 3, 7.)
- [BK22] James Bartusek and Dakshita Khurana. Cryptography with certified deletion. Cryptology ePrint Archive, Paper 2022/1178, 2022. <https://eprint.iacr.org/2022/1178>. (Cited on page 7.)
- [BKM<sup>+</sup>23] James Bartusek, Dakshita Khurana, Giulio Malavolta, Alexander Poremba, and Michael Walter. Weakening assumptions for publicly-verifiable deletion. In *Theory of Cryptography: 21st International Conference, TCC 2023, Taipei, Taiwan, November 29–December 2, 2023, Proceedings, Part IV*, page 183–197, Berlin, Heidelberg, 2023. Springer-Verlag. (Cited on page 2, 6, 7, 23.)
- [BKP23] James Bartusek, Dakshita Khurana, and Alexander Poremba. Publicly-verifiable deletion via target-collapsing functions. In *Advances in Cryptology – CRYPTO 2023: 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20–24, 2023, Proceedings, Part V*, page 99–128, Berlin, Heidelberg, 2023. Springer-Verlag. (Cited on page 7.)
- [BL20] Anne Broadbent and Sébastien Lord. Uncloneable quantum encryption via oracles. In Steven T. Flammia, editor, *15th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2020, June 9-12, 2020, Riga, Latvia*, volume 158 of *LIPICs*, pages 4:1–4:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. (Cited on page 3.)

- [CGJL23] Orestis Chardouvelis, Vipul Goyal, Aayush Jain, and Jiahui Liu. Quantum key leasing for pke and fhe with a classical lessor. Cryptology ePrint Archive, Paper 2023/1640, 2023. <https://eprint.iacr.org/2023/1640>. (Cited on page 7.)
- [CLLZ21] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. Hidden cosets and applications to unclonable cryptography. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 556–584, Virtual Event, August 2021. Springer, Heidelberg. (Cited on page 4, 7.)
- [CMP20] Andrea Coladangelo, Christian Majenz, and Alexander Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. Cryptology ePrint Archive, Paper 2020/1194, 2020. <https://eprint.iacr.org/2020/1194>. (Cited on page 7.)
- [Got02] Daniel Gottesman. Unclonable encryption, 2002. (Cited on page 3.)
- [GZ20] Marios Georgiou and Mark Zhandry. Unclonable decryption keys. Cryptology ePrint Archive, Paper 2020/877, 2020. <https://eprint.iacr.org/2020/877>. (Cited on page 7.)
- [Hel69] Carl W. Helstrom. Quantum detection and estimation theory. *Journal of Statistical Physics*, 1:231–252, 1969. (Cited on page 8.)
- [HKM<sup>+</sup>23] Taiga Hiroka, Fuyuki Kitagawa, Tomoyuki Morimae, Ryo Nishimaki, Tapas Pal, and Takashi Yamakawa. Certified everlasting secure collusion-resistant functional encryption, and more. Cryptology ePrint Archive, Paper 2023/236, 2023. <https://eprint.iacr.org/2023/236>. (Cited on page 7.)
- [HMNY21] Taiga Hiroka, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. Quantum encryption with certified deletion, revisited: Public key, attribute-based, and classical communication. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 606–636. Springer, Heidelberg, December 2021. (Cited on page 7.)
- [HMNY22] Taiga Hiroka, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. Certified everlasting zero-knowledge proof for QMA. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 239–268. Springer, Heidelberg, August 2022. (Cited on page 7.)
- [HMY23] Minki Hhan, Tomoyuki Morimae, and Takashi Yamakawa. From the hardness of detecting superpositions to cryptography: Quantum public key encryption and commitments. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part I*, volume 14004 of *LNCS*, pages 639–667. Springer, Heidelberg, April 2023. (Cited on page 38, 39, 40.)
- [Hol73] A.S Holevo. Statistical decision theory for quantum systems. *Journal of Multivariate Analysis*, 3(4):337–394, 1973. (Cited on page 8.)
- [JQSY19] Zhengfeng Ji, Youming Qiao, Fang Song, and Aaram Yun. General linear group action on tensors: A candidate for post-quantum cryptography. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part I*, volume 11891 of *LNCS*, pages 251–281. Springer, Heidelberg, December 2019. (Cited on page 4, 38, 39.)

- [KL07] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007. (Cited on page 18, 19, 21.)
- [KN22] Fuyuki Kitagawa and Ryo Nishimaki. Functional encryption with secure key leasing. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part IV*, volume 13794 of *LNCS*, pages 569–598. Springer, Heidelberg, December 2022. (Cited on page 7.)
- [KNY21] Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Secure software leasing from standard assumptions. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part I*, volume 13042 of *LNCS*, pages 31–61. Springer, Heidelberg, November 2021. (Cited on page 5, 10, 12, 40.)
- [KNY23] Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Publicly verifiable deletion from minimal assumptions. *Cryptology ePrint Archive*, Paper 2023/538, 2023. <https://eprint.iacr.org/2023/538>. (Cited on page 7.)
- [LLQZ22] Jiahui Liu, Qipeng Liu, Luowen Qian, and Mark Zhandry. Collusion resistant copy-protection for watermarkable functionalities. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 294–323. Springer, Heidelberg, November 2022. (Cited on page 5, 7, 14.)
- [Mer88] Ralph C. Merkle. A digital signature based on a conventional encryption function. In Carl Pomerance, editor, *CRYPTO '87*, volume 293 of *LNCS*, pages 369–378. Springer, Heidelberg, August 1988. (Cited on page 19.)
- [MY22] Tomoyuki Morimae and Takashi Yamakawa. Quantum commitments and signatures without one-way functions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 269–295. Springer, Heidelberg, August 2022. (Cited on page 7.)
- [NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *21st ACM STOC*, pages 33–43. ACM Press, May 1989. (Cited on page 32.)
- [Por23] Alexander Poremba. Quantum proofs of deletion for learning with errors. *ITCS 2023: 14th Innovations in Theoretical Computer Science*, 2023. (Cited on page 7.)
- [Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93. ACM Press, 2005. (Cited on page 4, 11, 12.)
- [Riv98] Ronald L. Rivest. Can we eliminate certificate revocation lists? In Rafael Hirschfeld, editor, *Proceedings Financial Cryptography '98*, volume 1465 of *Lecture Notes in Computer Science*, pages 178–183. Springer, 1998. (Cited on page 3.)
- [RS19] Roy Radian and Or Sattath. Semi-quantum money. *arXiv/1908.08889*, 2019. (Cited on page 10.)
- [Shm22] Omri Shmueli. Semi-quantum tokenized signatures. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 296–319. Springer, Heidelberg, August 2022. (Cited on page 5.)
- [Stu95] S. Stubblebine. Recent-secure authentication: enforcing revocation in distributed systems. In *2012 IEEE Symposium on Security and Privacy*, page 0224, Los Alamitos, CA, USA, may 1995. IEEE Computer Society. (Cited on page 3.)

- [Unr15] Dominique Unruh. Revocable quantum timed-release encryption. *J. ACM*, 62(6):49:1–49:76, 2015. (Cited on page 7.)
- [Wie83] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, 1983. (Cited on page 3.)
- [Zha19] Mark Zhandry. Quantum lightning never strikes the same state twice. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 408–438. Springer, Heidelberg, May 2019. (Cited on page 5.)

## A Proof of Theorem 5.4

Here we show Theorem 5.4. Assume that there exist a QPT adversary  $\mathcal{A}_\lambda$ , a polynomial poly, and bits  $b_1, \dots, b_n \in \{0, 1\}$  such that

$$\left\| \tilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(b_1, \dots, b_n) - \tilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(0, \dots, 0) \right\|_{\text{tr}} \geq \frac{1}{\text{poly}(\lambda)} \quad (55)$$

for infinitely often  $\lambda$ . Then due to the triangle inequality of the trace distance, there exist an integer  $k$  and bits  $a_1, \dots, a_{k-1}, a_{k+1}, \dots, a_n \in \{0, 1\}$  such that

$$\left\| \tilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(a_1, \dots, a_{k-1}, 0, a_{k+1}, \dots, a_n) - \tilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(a_1, \dots, a_{k-1}, 1, a_{k+1}, \dots, a_n) \right\|_{\text{tr}} \geq \frac{1}{\text{poly}(\lambda)}. \quad (56)$$

From such  $\mathcal{A}_\lambda$ , we can construct a QPT adversary  $\mathcal{B}_\lambda$  that breaks Theorem 5.3 as follows:

1. Get

$$\mathcal{Z}_\lambda \left( x_0 \oplus x_1, y_0, y_1, \frac{|x_0\rangle + (-1)^b |x_1\rangle}{\sqrt{2}} \right) \quad (57)$$

as input.

2. Sample  $x_i^0, x_i^1 \leftarrow \{0, 1\}^{\ell(\lambda)}$  for each  $i \in [n] \setminus \{k\}$ . Define  $y_i^0 = f(x_i^0)$  and  $y_i^1 = f(x_i^1)$  for each  $i \in [n] \setminus \{k\}$ . Initialize  $\mathcal{A}_\lambda$  with

$$\bigotimes_{i=1}^{k-1} \mathcal{Z}_\lambda \left( x_i^0 \oplus x_i^1, y_i^0, y_i^1, \frac{|x_i^0\rangle + (-1)^{a_i} |x_i^1\rangle}{\sqrt{2}} \right) \quad (58)$$

$$\otimes \mathcal{Z}_\lambda \left( x_0 \oplus x_1, y_0, y_1, \frac{|x_0\rangle + (-1)^b |x_1\rangle}{\sqrt{2}} \right) \quad (59)$$

$$\otimes \bigotimes_{i=k+1}^n \mathcal{Z}_\lambda \left( x_i^0 \oplus x_i^1, y_i^0, y_i^1, \frac{|x_i^0\rangle + (-1)^{a_i} |x_i^1\rangle}{\sqrt{2}} \right). \quad (60)$$

3.  $\mathcal{A}_\lambda$ 's output is parsed as strings  $x'_i \in \{0, 1\}^{\ell(\lambda)}$  for  $i \in [n]$  and a residual state on register  $A'$ .
4. If  $f(x'_i) \in \{y_i^0, y_i^1\}$  for all  $i \in [n] \setminus \{k\}$ , then output  $x'_k$  and  $A'$ . Otherwise output  $\perp$ .

It is clear that the output of the above game for  $b \in \{0, 1\}$  is equivalent to  $\tilde{\mathcal{Z}}_\lambda^{\mathcal{A}_\lambda}(a_1, \dots, a_{k-1}, b, a_{k+1}, \dots, a_n)$  (up to  $\{x_i^0, x_i^1\}_{i \in [n] \setminus \{k\}}$ ). Therefore, from Equation (56),  $\mathcal{B}_\lambda$  breaks Theorem 5.3.

## B Digital Signatures with Quantum Revocable Signing Keys from Group Actions

In this section, we construct digital signatures with *quantum* revocable signing keys from group actions. Note that in this case, the revocation is quantum, i.e., not a classical deletion certificate but the quantum signing key itself is returned. The syntax, correctness, and security are the same as those of the classical revocation case, Definition 4.1, except that there is no deletion algorithm, Del, and cert is not a classical bit string but the signing key  $\text{sig}k_i$  itself.

The outline of this section is as follows. In Appendix B.1, we review basics of group actions. We then review claw-free swap-trapdoor function pairs [HMY23] and mention that it can be constructed from group actions with the one-wayness property in Appendix B.2. We next define two-tier quantum lightning with classical semi-verification in Appendix B.3, and construct it from claw-free swap-trapdoor function pairs (and therefore from group actions with the one-wayness property) in Appendix B.4. We finally construct digital signatures with quantum revocable signing keys from two-tier quantum lightning with classical semi-verification in Appendix B.5.

### B.1 Group Actions

Here we review basics of group actions [JQSY19]. Content of this subsection is taken from [HMY23]. First, group actions are defined as follows.

**Definition B.1 (Group actions).** *Let  $G$  be a (not necessarily abelian) group,  $S$  be a set, and  $\star : G \times S \rightarrow S$  be a function where we write  $g \star s$  to mean  $\star(g, s)$ . We say that  $(G, S, \star)$  is a group action if it satisfies the following:*

1. *For the identity element  $e \in G$  and any  $s \in S$ , we have  $e \star s = s$ .*
2. *For any  $g, h \in G$  and any  $s \in S$ , we have  $(gh) \star s = g \star (h \star s)$ .*

To be useful for cryptography, we have to at least assume that basic operations about  $(G, S, \star)$  have efficient algorithms. We require the following efficient algorithms similarly to [JQSY19].

**Definition B.2 (Group actions with efficient algorithms).** *We say that a group action  $(G, S, \star)$  has efficient algorithms if it satisfies the following:<sup>16</sup>*

**Unique representations:** *Each element of  $G$  and  $S$  can be represented as a bit string of length  $\text{poly}(\lambda)$  in a unique manner. Thus, we identify these elements and their representations.*

**Group operations:** *There are classical deterministic polynomial-time algorithms that compute  $gh$  from  $g \in G$  and  $h \in G$  and  $g^{-1}$  from  $g \in G$ .*

**Group action:** *There is a classical deterministic polynomial-time algorithm that computes  $g \star s$  from  $g \in G$  and  $s \in S$ .*

**Efficient recognizability:** *There are classical deterministic polynomial-time algorithms that decide if a given bit string represents an element of  $G$  or  $S$ , respectively.*

---

<sup>16</sup>Strictly speaking, we have to consider a family  $\{(G_\lambda, S_\lambda, \star_\lambda)\}_{\lambda \in \mathbb{N}}$  of group actions parameterized by the security parameter to meaningfully define the efficiency requirements. We omit the dependence on  $\lambda$  for notational simplicity throughout the paper.

**Random sampling:** *There are PPT algorithms that sample almost uniform elements of  $G$  or  $S$  (i.e., the distribution of the sample is statistically close to the uniform distribution), respectively.*

**Superposition over  $G$ :** *There is a QPT algorithm that generates a state whose trace distance from  $|G\rangle := \sum_{g \in G} |g\rangle$  is  $\text{negl}(\lambda)$ .*

*Remark B.3* (A convention on “Random sampling” and “Superposition over  $G$ ” properties). In the rest of this paper, we assume that we can sample elements from *exactly* uniform distributions of  $G$  and  $S$ . Similarly, we assume that we can *exactly* generate  $|G\rangle$  in QPT. They are just for simplifying the presentations of our results, and all the results hold with the above imperfect version with additive negligible loss for security or correctness.

*Remark B.4.* The above requirements are identical to those in [JQSY19] except for the “superposition over  $G$ ” property. We remark that all candidate constructions proposed in [JQSY19] satisfy this property.

We define one-wayness [JQSY19] as follows.

**Definition B.5 (One-wayness).** *We say that a group action  $(G, S, \star)$  with efficient algorithms is one-way if for any non-uniform QPT adversary  $\mathcal{A}$ , we have*

$$\Pr \left[ g' \star s = g \star s : s \leftarrow S, g \leftarrow G, g' \leftarrow \mathcal{A}(s, g \star s) \right] = \text{negl}(\lambda).$$

## B.2 Swap-Trapdoor Functions Pairs

In order to show the construction of two-tier quantum lightning with classical semi-verification from group actions, it is convenient to use *swap-trapdoor function pairs* (STF), which was introduced in [HMY23]. In this subsection, we define STF and point out that STF can be constructed from group actions [JQSY19, HMY23]. Content of this subsection is taken from [HMY23].

The formal definition of STF is given as follows.

**Definition B.6 (Swap-Trapdoor Function Pair [HMY23]).** *A swap-trapdoor function pair (STF) consists a set (Setup, Eval, Swap) of algorithms such that*

- $\text{Setup}(1^\lambda) \rightarrow (\text{pp}, \text{td})$ : *It is a PPT algorithm that takes the security parameter  $\lambda$  as input, and outputs a public parameter  $\text{pp}$  and a trapdoor  $\text{td}$ . The public parameter  $\text{pp}$  specifies functions  $f_b^{(\text{pp})} : \mathcal{X} \rightarrow \mathcal{Y}$  for each  $b \in \{0, 1\}$ . We often omit the dependence on  $\text{pp}$  and simply write  $f_b$  when it is clear from the context.*
- $\text{Eval}(\text{pp}, b, x) \rightarrow y$ : *It is a deterministic classical polynomial-time algorithm that takes a public parameter  $\text{pp}$ , a bit  $b \in \{0, 1\}$ , and an element  $x \in \mathcal{X}$  as input, and outputs  $y \in \mathcal{Y}$ .*
- $\text{Swap}(\text{td}, b, x) \rightarrow x'$ : *It is a deterministic classical polynomial-time algorithm that takes a trapdoor  $\text{td}$ , a bit  $b \in \{0, 1\}$ , and an element  $x \in \mathcal{X}$  as input, and outputs  $x' \in \mathcal{X}$ .*

We require a STF to satisfy the following:

**Evaluation correctness.** *For any  $(\text{pp}, \text{td}) \leftarrow \text{Setup}(1^\lambda)$ ,  $b \in \{0, 1\}$ , and  $x \in \mathcal{X}$ , we have  $\text{Eval}(\text{pp}, b, x) = f_b(x)$ .*

**Swapping correctness.** *For any  $(\text{pp}, \text{td}) \leftarrow \text{Setup}(1^\lambda)$ ,  $b \in \{0, 1\}$ , and  $x \in \mathcal{X}$ , if we let  $x' \leftarrow \text{Swap}(\text{td}, b, x)$ , then we have  $f_{b \oplus 1}(x') = f_b(x)$  and  $\text{Swap}(\text{td}, b \oplus 1, x') = x$ . In particular,  $\text{Swap}(\text{td}, b, \cdot)$  induces an efficiently computable and invertible one-to-one mapping between  $f_0^{-1}(y)$  and  $f_1^{-1}(y)$  for any  $y \in \mathcal{Y}$ .*

**Efficient random sampling over  $\mathcal{X}$ .** There is a PPT algorithm that samples an almost uniform element of  $\mathcal{X}$  (i.e., the distribution of the sample is statistically close to the uniform distribution).

**Efficient superposition over  $\mathcal{X}$ .** There is a QPT algorithm that generates a state whose trace distance from  $|\mathcal{X}\rangle = \frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}} |x\rangle$  is  $\text{negl}(\lambda)$ .

**Claw-freeness.** For any non-uniform QPT algorithm  $\mathcal{A}$ ,

$$\Pr[f_0(x_0) = f_1(x_1) : (\text{pp}, \text{td}) \leftarrow \text{Setup}(1^\lambda), (x_0, x_1) \leftarrow \mathcal{A}(\text{pp})] = \text{negl}(\lambda).$$

*Remark B.7* (A convention on “Efficient random sampling over  $\mathcal{X}$ ” and “Efficient superposition over  $\mathcal{X}$ ” properties). In the rest of this paper, we assume that we can sample elements from *exactly* the uniform distribution of  $\mathcal{X}$ . Similarly, we assume that we can *exactly* generate  $|\mathcal{X}\rangle$  in QPT. They are just for simplifying the presentations of our results, and all the results hold with the above imperfect version with additive negligible loss for security or correctness.

A STF can be constructed from group actions. Let  $(G, S, \star)$  be a group action with efficient algorithms (as defined in Definition B.2). Then, we construct a STF as follows.

- $\text{Setup}(1^\lambda)$ : Generate  $s_0 \leftarrow S$  and  $g \leftarrow G$ , set  $s_1 := g \star s_0$ , and output  $\text{pp} := (s_0, s_1)$  and  $\text{td} := g$ . For  $b \in \{0, 1\}$ , we define  $f_b : G \rightarrow S$  by  $f_b(h) := h \star s_b$ .
- $\text{Eval}(\text{pp} = (s_0, s_1), b, h)$ : Output  $f_b(h) = h \star s_b$ .
- $\text{Swap}(\text{td} = g, b, h)$ : If  $b = 0$ , output  $hg^{-1}$ . If  $b = 1$ , output  $hg$ .

In [HMY23], the following theorem is shown.

**Theorem B.8** ([HMY23]). *If  $(G, S, \star)$  is one-way, then  $(\text{Setup}, \text{Eval}, \text{Swap})$  is claw-free.*

### B.3 Definition of Two-Tier Quantum Lightning with Classical Semi-Verification

The formal definition of two-tier quantum lightning with classical semi-verification is given as follows. The difference from the original two-tier quantum lightning [KNY21] is that the semi-verification algorithm  $\text{SemiVer}$  accepts a classical certificate  $\text{cert}$  instead of a quantum state, and there is an additional QPT algorithm  $\text{Del}$  that outputs  $\text{cert}$  on input  $\psi$ .

**Definition B.9 (Two-Tier Quantum Lightning with Classical Semi-Verification).** *Two-tier quantum lightning with classical semi-verification is a set  $(\text{Setup}, \text{StateGen}, \text{Del}, \text{SemiVer}, \text{FullVer})$  of algorithms such that*

- $\text{Setup}(1^\lambda) \rightarrow (\text{sk}, \text{pk})$  : *It is a QPT algorithm that, on input the security parameter  $\lambda$ , outputs a classical secret key  $\text{sk}$  and a classical public key  $\text{pk}$ .*
- $\text{StateGen}(\text{pk}) \rightarrow (\psi, \text{snum})$  : *It is a QPT algorithm that, on input  $\text{pk}$ , outputs a quantum state  $\psi$  and a classical serial number  $\text{snum}$ .*
- $\text{Del}(\psi) \rightarrow \text{cert}$  : *It is a QPT algorithm that, on input  $\psi$ , outputs a classical certificate  $\text{cert}$ .*
- $\text{SemiVer}(\text{pk}, \text{snum}, \text{cert}) \rightarrow \top/\perp$  : *It is a classical deterministic polynomial-time algorithm that, on input  $\text{pk}$ ,  $\text{cert}$ , and  $\text{snum}$ , outputs  $\top/\perp$ .*
- $\text{FullVer}(\text{sk}, \text{snum}, \psi) \rightarrow \top/\perp$  : *It is a QPT algorithm that, on input  $\text{sk}$ ,  $\psi$ , and  $\text{snum}$ , outputs  $\top/\perp$ .*

We require the following properties.

**Correctness:**

$$\Pr \left[ \top \leftarrow \text{SemiVer}(\text{pk}, \text{snum}, \text{cert}) : \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{Setup}(1^\lambda) \\ (\psi, \text{snum}) \leftarrow \text{StateGen}(\text{pk}) \\ \text{cert} \leftarrow \text{Del}(\psi) \end{array} \right] \geq 1 - \text{negl}(\lambda) \quad (61)$$

and

$$\Pr \left[ \top \leftarrow \text{FullVer}(\text{sk}, \text{snum}, \psi) : \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{Setup}(1^\lambda) \\ (\psi, \text{snum}) \leftarrow \text{StateGen}(\text{pk}) \end{array} \right] \geq 1 - \text{negl}(\lambda). \quad (62)$$

**Security:** For any QPT adversary  $\mathcal{A}$ ,

$$\Pr \left[ \top \leftarrow \text{SemiVer}(\text{pk}, \text{snum}, \text{cert}) \wedge \top \leftarrow \text{FullVer}(\text{sk}, \text{snum}, \psi) : \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{cert}, \psi, \text{snum}) \leftarrow \mathcal{A}(\text{pk}) \end{array} \right] \leq \text{negl}(\lambda). \quad (63)$$

## B.4 Construction from Group Actions

Here we construct two-tier quantum lightning with classical semi-verification from STF (and therefore from group actions).

**Theorem B.10.** *If group actions with the one-wayness property exist, then two-tier quantum lightning with classical semi-verification exists.*

*Proof of Theorem B.10.* Let (Setup, Eval, Swap) be a SFT. From Theorem B.8, a SFT exists if group actions that satisfy one-wayness exist. From it, we construct two-tier quantum lightning with classical semi-verification (Setup, StateGen, Del, SemiVer, FullVer) as follows.

- $\text{Setup}(1^\lambda) \rightarrow (\text{sk}, \text{pk})$  : Run  $(\text{pp}, \text{td}) \leftarrow \text{Setup}(1^\lambda)$ . Output  $\text{pk} := \text{pp}$  and  $\text{sk} := (\text{pp}, \text{td})$ .
- $\text{StateGen}(\text{pk}) \rightarrow (\psi, \text{snum})$  : Parse  $\text{pk} := \text{pp}$ . For each  $i \in [n]$ , repeat the following:
  1. Generate  $\sum_{b \in \{0,1\}} \sum_{x \in \mathcal{X}} |b\rangle |x\rangle |\text{Eval}(\text{pp}, b, x)\rangle$ .
  2. Measure the third register in the computational basis to get the measurement result  $y_i$ . The post-measurement state is  $\psi_i := |0\rangle \otimes \sum_{x \in f_0^{-1}(y_i)} |x\rangle + |1\rangle \otimes \sum_{x \in f_1^{-1}(y_i)} |x\rangle$ .
 Output  $\psi := \otimes_{i \in [n]} \psi_i$  and  $\text{snum} := \{y_i\}_{i \in [n]}$ .
- $\text{Del}(\psi) \rightarrow \text{cert}$  : Parse  $\psi = \otimes_{i \in [n]} \psi_i$ . Measure  $\psi_i$  in the computational basis to get the result  $(b_i, z_i) \in \{0, 1\} \times \mathcal{X}$  for each  $i \in [n]$ . Output  $\text{cert} := \{b_i, z_i\}_{i \in [n]}$ .
- $\text{SemiVer}(\text{pk}, \text{cert}, \text{snum}) \rightarrow \top/\perp$  : Parse  $\text{pk} := \text{pp}$ ,  $\text{cert} = \{b_i, z_i\}_{i \in [n]}$  and  $\text{snum} = \{y_i\}_{i \in [n]}$ . If  $\text{Eval}(\text{pp}, b_i, z_i) = y_i$  for all  $i \in [n]$ , output  $\top$ . Otherwise, output  $\perp$ .
- $\text{FullVer}(\text{sk}, \psi, \text{snum}) \rightarrow \top/\perp$  : Parse  $\text{sk} := (\text{pp}, \text{td})$ ,  $\psi = \otimes_{i \in [n]} \psi_i$ , and  $\text{snum} = \{y_i\}_{i \in [n]}$ . For each  $i \in [n]$ , do the following:
  1. Apply unitary  $U$  such that  $U |b\rangle |x\rangle |0\dots 0\rangle = |b\rangle |x\rangle |\text{Eval}(\text{pp}, b, x)\rangle$  on  $\psi_i \otimes |0\dots 0\rangle \langle 0\dots 0|$ , and measure the third register in the computational basis. If the measurement result is not  $y_i$ , output  $\perp$  and abort.

2. Apply the operation that maps  $x$  to  $\text{Swap}(\text{td}, 1, x)$  on the second register of the post-measurement state conditioned that the first register is 1.<sup>17</sup> Measure the first register in the Hadamard basis. If the result is 1 (i.e., if the result  $|-\rangle$  is obtained), output  $\perp$  and abort.

Output  $\top$ .

The correctness for the semi-verification is clear. The correctness for the full-verification is also clear because if each  $\psi_i$  is

$$\psi_i = |0\rangle \otimes \sum_{x \in f_0^{-1}(y_i)} |x\rangle + |1\rangle \otimes \sum_{x \in f_1^{-1}(y_i)} |x\rangle, \quad (64)$$

then the first operation of the semi-verification algorithm does not change it, and the second operation gives

$$|0\rangle \otimes \sum_{x \in f_0^{-1}(y_i)} |x\rangle + |1\rangle \otimes \sum_{x \in f_1^{-1}(y_i)} |\text{Swap}(\text{td}, 1, x)\rangle = |0\rangle \otimes \sum_{x \in f_0^{-1}(y_i)} |x\rangle + |1\rangle \otimes \sum_{x \in f_0^{-1}(y_i)} |x\rangle \quad (65)$$

$$= |+\rangle \otimes \sum_{x \in f_0^{-1}(y_i)} |x\rangle. \quad (66)$$

To show the security, we define the sequence of hybrids. Hybrid 0 (Figure 1) is the original security game. For the sake of contradiction, we assume that there exists a QPT adversary  $\mathcal{A}$  such that

$$\Pr[\text{Hybrid 0} \rightarrow \top] \geq \frac{1}{\text{poly}(\lambda)} \quad (67)$$

for infinitely many  $\lambda$ .

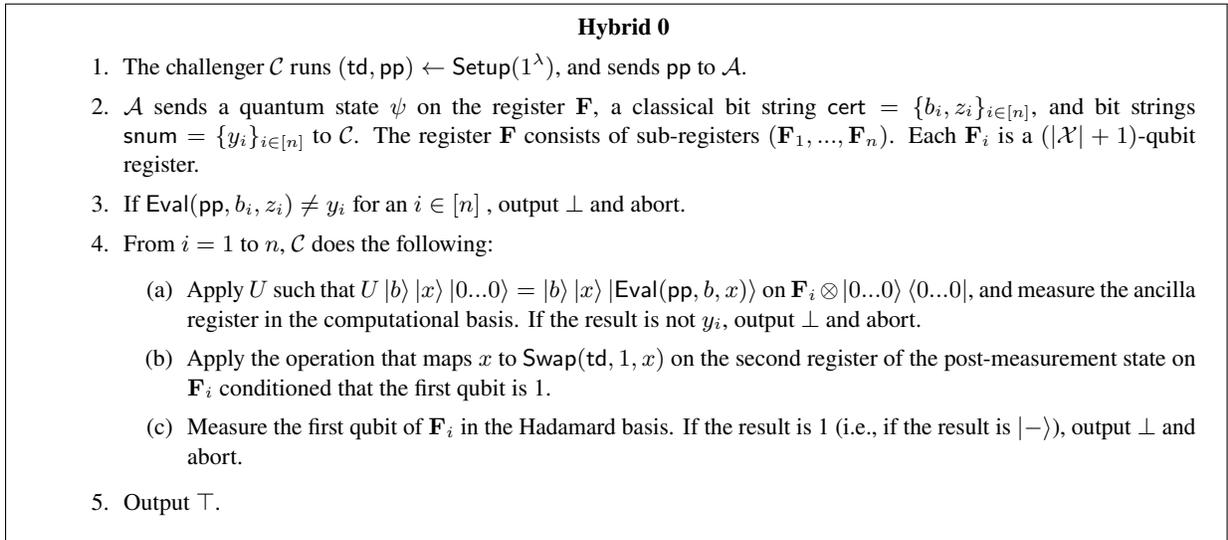


Figure 1: Hybrid 0

<sup>17</sup>Such an operation can be done in the following way: First, change  $|x\rangle |0\dots 0\rangle$  into  $|x\rangle |\text{Swap}(\text{td}, 1, x)\rangle$ . Then, change this state into  $|x \oplus \text{Swap}(\text{td}, 0, \text{Swap}(\text{td}, 1, x))\rangle |\text{Swap}(\text{td}, 1, x)\rangle = |0\dots 0\rangle |\text{Swap}(\text{td}, 1, x)\rangle$ . Finally, if we swap the first and the second register, we get  $|\text{Swap}(\text{td}, 1, x)\rangle |0\dots 0\rangle$ .

**Hybrid 1**

1. The same as Hybrid 0.
2. The same as Hybrid 0.
3. The same as Hybrid 0.
4. From  $i = 1$  to  $n$ ,  $\mathcal{C}$  does the following:
  - (a) The same as Hybrid 0.
  - (b) The same as Hybrid 0.
  - (c) Measure all qubits of  $\mathbf{F}_i$  in the computational basis to get the result  $(e_i, w_i) \in \{0, 1\} \times \mathcal{X}$ .
5. If there is  $i \in [n]$  such that  $b_i \neq e_i$ , output  $\top$ . Otherwise, output  $\perp$ .

Figure 2: Hybrid 1

**Hybrid 2**

1. The same as Hybrid 1.
2. The same as Hybrid 1.
3. The same as Hybrid 1.
4. The same as Hybrid 1 except that the step (b) is skipped for each  $i \in [n]$ .
5. The same as Hybrid 1.

Figure 3: Hybrid 2

**Lemma B.11.** *If  $\Pr[\text{Hybrid 0} \rightarrow \top] \geq \frac{1}{\text{poly}(\lambda)}$  for infinitely many  $\lambda$ , then  $\Pr[\text{Hybrid 1} \rightarrow \top] \geq \frac{1}{\text{poly}(\lambda)}$  for infinitely many  $\lambda$ .*

*Proof.* Let  $\Pr[b_1, \dots, b_n]$  be the probability that  $\mathcal{A}$  outputs  $\{b_i\}_{i \in [n]}$  and  $\mathcal{C}$  finishes Item 4b of Figure 1 without aborting. Then

$$\epsilon := \Pr[\text{Hybrid 0} \rightarrow \top] \tag{68}$$

$$= \sum_{b_1, \dots, b_n} \Pr[b_1, \dots, b_n] \text{Tr}[(|+\rangle \langle +|^{\otimes n} \otimes I) \psi_{b_1, \dots, b_n}], \tag{69}$$

where  $\psi_{b_1, \dots, b_n}$  is the state of the register  $\mathbf{F}$  after the end of Item 4b of Figure 1. From the standard average argument, we have

$$\sum_{(b_1, \dots, b_n) \in G} \Pr[b_1, \dots, b_n] \geq \frac{\epsilon}{2}, \tag{70}$$

where

$$G := \left\{ (b_1, \dots, b_n) : \text{Tr}[(|+\rangle \langle +|^{\otimes n} \otimes I) \psi_{b_1, \dots, b_n}] \geq \frac{\epsilon}{2} \right\}. \tag{71}$$

Define

$$\Lambda_{b_1, \dots, b_n} := \sum_{(e_1, \dots, e_n) \in \{0, 1\}^n \setminus (b_1, \dots, b_n)} |e_1, e_2, \dots, e_n\rangle \langle e_1, e_2, \dots, e_n|. \tag{72}$$

**Hybrid 3**

1. The same as Hybrid 2. In addition,  $\mathcal{C}$  chooses  $i^* \leftarrow [n]$ .
2. The same as Hybrid 2.
3. The same as Hybrid 2.
4. The same as Hybrid 2.
5. If  $b_{i^*} \neq e_{i^*}$ , output  $\top$ . Otherwise, output  $\perp$ .

Figure 4: Hybrid 3

As is shown later, for any  $m$ -qubit state  $\rho$ , where  $m \geq n$ , and any  $(b_1, \dots, b_n) \in \{0, 1\}^n$ , we have

$$\text{Tr}[(|+\rangle\langle +|^{\otimes n} \otimes I)\rho] \leq 2\text{Tr}[(\Lambda_{b_1, \dots, b_n} \otimes I)\rho] + 2^{-n+1}. \quad (73)$$

Then

$$\Pr[\text{Hybrid 1} \rightarrow \top] = \sum_{b_1, \dots, b_n} \Pr[b_1, \dots, b_n] \text{Tr}[(\Lambda_{b_1, \dots, b_n} \otimes I)\psi_{b_1, \dots, b_n}] \quad (74)$$

$$\geq \sum_{(b_1, \dots, b_n) \in G} \Pr[b_1, \dots, b_n] \text{Tr}[(\Lambda_{b_1, \dots, b_n} \otimes I)\psi_{b_1, \dots, b_n}] \quad (75)$$

$$\geq \sum_{(b_1, \dots, b_n) \in G} \Pr[b_1, \dots, b_n] \left( \frac{1}{2} \text{Tr}[(|+\rangle\langle +|^{\otimes n} \otimes I)\psi_{b_1, \dots, b_n}] - 2^{-n} \right) \quad (76)$$

$$\geq \frac{\epsilon}{2} \left( \frac{\epsilon}{4} - 2^{-n} \right) \quad (77)$$

$$\geq \frac{1}{\text{poly}(\lambda)}. \quad (78)$$

Here, to derive Equation (76), we have used Equation (73). To derive Equation (77), we have used Equation (70) and Equation (71).  $\square$

**Lemma B.12.**  $\Pr[\text{Hybrid 2} \rightarrow \top] = \Pr[\text{Hybrid 1} \rightarrow \top]$ .

*Proof.* The controlled Swap(td, 1,  $\cdot$ ) operations do not change the distribution of  $(e_1, \dots, e_n)$ .  $\square$

**Lemma B.13.**  $\Pr[\text{Hybrid 3} \rightarrow \top] \geq \frac{1}{n} \Pr[\text{Hybrid 2} \rightarrow \top]$ .

*Proof.* It is trivial.  $\square$

In summary, from Equation (67) and Lemmata B.11 to B.13, we conclude that  $\Pr[\text{Hybrid 3} \rightarrow \top] \geq \frac{1}{\text{poly}(\lambda)}$  for infinitely many  $\lambda$ . From such  $\mathcal{A}$  of Hybrid 3, we construct a QPT adversary  $\mathcal{B}$  that breaks the claw-freeness of STF as in Figure 5. It is easy to verify that

$$\Pr[\mathcal{B} \text{ wins}] \geq \Pr[\text{Hybrid 3} \rightarrow \top] \quad (79)$$

$$\geq \frac{1}{\text{poly}(\lambda)} \quad (80)$$

for infinitely many  $\lambda$ . Therefore,  $\mathcal{B}$  breaks the claw-freeness of the STF.  $\square$

$\mathcal{B}$

1. Get  $pp$  as input.
2. Run  $(\psi, \{b_i, z_i\}_{i \in [n]}, \{y_i\}_{i \in [n]}) \leftarrow \mathcal{A}(pp)$ . Choose  $i^* \leftarrow [n]$ .
3. Measure all qubits of  $\mathbf{F}_{i^*}$  in the computational basis to get the result  $(e_{i^*}, w_{i^*})$ .
4. If  $b_{i^*} = 0$  and  $e_{i^*} = 1$ , output  $(z_{i^*}, w_{i^*})$ . If  $b_{i^*} = 1$  and  $e_{i^*} = 0$ , output  $(w_{i^*}, z_{i^*})$ . Otherwise, output  $\perp$ .

Figure 5:  $\mathcal{B}$

*Proof of Equation (73).* Because of the linearity of  $\text{Tr}$ , we have only to show it for pure states. We have

$$\text{Tr}[(|+\rangle \langle +|^{\otimes n} \otimes I) |\psi\rangle \langle \psi|] = \left\| \langle +|^{\otimes n} \otimes I |\psi\rangle \right\|^2 \quad (81)$$

$$= \frac{1}{2^n} \left\| \sum_{e_1, \dots, e_n} (\langle e_1, \dots, e_n | \otimes I) |\psi\rangle \right\|^2 \quad (82)$$

$$= \frac{1}{2^n} \left\| \sum_{(e_1, \dots, e_n) \neq (b_1, \dots, b_n)} (\langle e_1, \dots, e_n | \otimes I) |\psi\rangle + (\langle b_1, \dots, b_n | \otimes I) |\psi\rangle \right\|^2 \quad (83)$$

$$\leq \frac{1}{2^{n-1}} \left\| \sum_{(e_1, \dots, e_n) \neq (b_1, \dots, b_n)} (\langle e_1, \dots, e_n | \otimes I) |\psi\rangle \right\|^2 \quad (84)$$

$$+ \frac{1}{2^{n-1}} \left\| (\langle b_1, \dots, b_n | \otimes I) |\psi\rangle \right\|^2 \quad (85)$$

$$\leq \frac{1}{2^{n-1}} (2^n - 1) \sum_{(e_1, \dots, e_n) \neq (b_1, \dots, b_n)} \left\| (\langle e_1, \dots, e_n | \otimes I) |\psi\rangle \right\|^2 + 2^{-n+1} \quad (86)$$

$$= \frac{2^n - 1}{2^{n-1}} \text{Tr}[(\Lambda_{b_1, \dots, b_n} \otimes I) |\psi\rangle \langle \psi|] + 2^{-n+1} \quad (87)$$

$$\leq 2 \text{Tr}[(\Lambda_{b_1, \dots, b_n} \otimes I) |\psi\rangle \langle \psi|] + 2^{-n+1}. \quad (88)$$

Here, we have used the triangle inequality and Jensen's inequality. □

## B.5 Construction of Digital Signatures with Quantum Revocable Signing Keys

We construct digital signatures with quantum revocable signing keys. Its definition is given as follows.

**Definition B.14 (Digital Signatures with quantum revocable signing keys).** *The syntax, correctness, and security are the same as those of Definition 4.1, except that there is no deletion algorithm,  $\text{Del}$ , and  $\text{cert}$  is not a classical bit string, but the signing key  $\text{sigk}_i$  itself.*

Digital signatures with quantum revocable signing keys can be constructed from group actions as follows.

**Theorem B.15.** *If group actions with the one-wayness property exist, then digital signatures with quantum revocable signing keys exist.*

*Proof.* Let  $(\text{QL.Setup}, \text{QL.StateGen}, \text{QL.Del}, \text{QL.SemiVer}, \text{QL.FullVer})$  be two-tier quantum lightning with classical semi-verification, which exists if group actions with the one-wayness property exist (Theorem B.10). From it, we construct digital signatures with quantum revocable signing keys with the message space  $\{0, 1\}$  that satisfy one-time variants of correctness, EUF-CMA security, deletion correctness, and deletion security

as follows. With similar proofs given in Section 4, we can upgrade it to the full version of digital signatures with quantum revocable signing keys.

- $\text{Setup}(1^\lambda) \rightarrow (\text{ck}, \text{pp})$  : Run  $(\text{sk}, \text{pk}) \leftarrow \text{QL.Setup}(1^\lambda)$ . Output  $\text{ck} := \text{sk}$  and  $\text{pp} := \text{pk}$ .
- $\text{KeyGen}(\text{pp}) \rightarrow (\text{sigk}, \text{vk})$  : Parse  $\text{pp} = \text{pk}$ . Run  $(\psi_0, \text{snum}_0) \leftarrow \text{QL.StateGen}(\text{pk})$  and  $(\psi_1, \text{snum}_1) \leftarrow \text{QL.StateGen}(\text{pk})$ . Output  $\text{sigk} := (\psi_0, \psi_1)$  and  $\text{vk} := (\text{snum}_0, \text{snum}_1)$ .
- $\text{Sign}(\text{pp}, \text{sigk}, m) \rightarrow (\text{sigk}', \sigma)$  : Parse  $\text{pp} = \text{pk}$  and  $\text{sigk} = (\psi_0, \psi_1)$ . Run  $\text{ql.cert} \leftarrow \text{QL.Del}(\psi_m)$ . Output  $\sigma := \text{ql.cert}$  and  $\text{sigk}' := (1 - m, \psi_{1-m})$ .
- $\text{Ver}(\text{pp}, \text{vk}, m, \sigma) \rightarrow \top/\perp$  : Parse  $\text{pp} = \text{pk}$ ,  $\text{vk} = (\text{snum}_0, \text{snum}_1)$ , and  $\sigma = \text{ql.cert}$ . Run  $\text{SemiVer}(\text{pk}, \text{snum}_m, \text{ql.cert})$  and output its output.
- $\text{Cert}(\text{pp}, \text{vk}, \text{ck}, \text{sigk}, S) \rightarrow \top/\perp$  : Parse  $\text{pp} = \text{pk}$ ,  $\text{vk} = (\text{snum}_0, \text{snum}_1)$ ,  $\text{ck} = \text{sk}$ .
  - If  $S = \emptyset$ , parse  $\text{sigk} = (0, 1, \psi_0, \psi_1)$ . (Output  $\perp$  if  $\text{sigk}$  is not of this form.) Output  $\top$  if  $\text{QL.FullVer}(\text{sk}, \text{snum}_b, \psi_b) = \top$  for both  $b = 0, 1$ .
  - If  $S = \{m\}$  for some  $m \in \{0, 1\}$ , parse  $\text{sigk} = (1 - m, \psi_{1-m})$ . (Output  $\perp$  if  $\text{sigk}$  is not of this form.) Output  $\top$  if  $\text{QL.FullVer}(\text{sk}, \text{snum}_{1-m}, \psi_{1-m}) = \top$ .
  - If  $S = \{0, 1\}$ , output  $\perp$ .

One-time correctness, one-time EUF-CMA security, one-time deletion correctness, and one-time deletion security are shown in similar ways as those in Section 4.2.  $\square$