Leighton-Micali Hash-Based Signatures in the Quantum Random-Oracle Model

Edward Eaton

ISARA Corporation <ted.eaton@isara.com>
and University of Waterloo, Canada

Abstract. Digital signatures constructed solely from hash functions offer competitive signature sizes and fast signing and verifying times. Moreover, the security of hash functions against a quantum adversary is believed to be well understood. This means that hash-based signatures are strong candidates for standard use in a post-quantum world. The Leighton-Micali signature scheme (LMS) is one such scheme being considered for standardization. However all systematic analyses of LMS have only considered a classical adversary. In this work we close this gap by showing a proof of the security of LMS in the quantum random-oracle model. Our results match the bounds imposed by Grover's search algorithm within a constant factor, and remain tight in the multi-user setting.

Keywords: Post-Quantum Cryptography, Digital Signatures, Random Oracles, Hash Functions, Multi-User Setting

1 Introduction

Hash-based signature schemes have their origins in the paper "Constructing Digital Signatures from a One Way Function", by Leslie Lamport [11]. The security of these schemes is based solely on the security properties of a standard hash function, as opposed to schemes whose security relies on problems such as the discrete-logarithm problem on finite groups, or the learning with errors problem. After Lamport's one-time scheme, Ralph Merkle improved upon the construction with the Winternitz one-time scheme and the ability to sign multiple messages with Merkle trees [14, 15]. The Leighton-Micali scheme, or LMS, proposed some modifications of Merkle's construction to improve speed and security [12].

Recently, there has been a renewed interest in hash-based signatures in general, and LMS in particular. This is partially due to the expiration of the patents LMS was covered by [12, 15], but more importantly because hash-based schemes are believed to remain secure against a quantum adversary. LMS has been proposed for standardization in a recent IETF draft [13]. In a recent paper, Jonathan Katz analyzed the security of LMS [10].

Katz's analysis used the random-oracle model to establish the security of LMS. However, as the random-oracle model is insufficient for establishing the

security of a protocol against an adversary with access to a quantum computer, we must move to the quantum random-oracle model [3].

In this paper, we reformulate and update Katz's random-oracle model proof of security for LMS to the quantum random-oracle model. As LMS is a hash-based scheme, this is particularly important as it is a strong candidate for post-quantum standardization. We also discuss some of the difficulties that need to be overcome in order to establish this proof in the quantum random-oracle model.

1.1 The Quantum Random-Oracle Model

Katz's classical proof of the security of LMS takes place in the random-oracle model. In his proof, he considers an experiment with an adversary $\mathcal{A}$, who is attacking the existential-unforgeability of the scheme. Whenever this adversary wishes to evaluate the n-bit hash function H on a point x, they must instead query an oracle for the evaluation, and are provided a response which is indistinguishable from random. Katz shows that for any adversary that makes q queries, the probability that $\mathcal{A}$ can break the existential-unforgeability of LMS is at most $3q/2^n$. He establishes this by showing that for the adversary to win a game, one of a series of events must occur. Then by upper bounding the probability of these events happening, the upper bound follows.

As the random oracle is meant to replace a hash function, an adversary should be able to interact with this oracle in a similar way to how they interact with a hash function. However it has been noted that an adversary with a quantum computer can interact with a hash function in ways very different from a 'make a single query, get a response' model [3]. If a hash function is implemented on a quantum computer, then they are able to evaluate the function in superposition, giving them access to the quantum mapping

$$U_H : \sum_{x,y} \alpha_{x,y}|x\rangle|y\rangle \mapsto \sum_{x,y} \alpha_{x,y}|x\rangle|y \oplus H(x)\rangle. \tag{1}$$

A model of security in which we provide access to this mapping to an adversary is called the *quantum random-oracle model*.

New issues arise in this model however, and Katz's proof no longer works. Katz's events are defined by considering the queries that the adversary makes and the responses they receive. However in the quantum random-oracle model, the queries the adversary makes no longer need be classical, and so the definition of these events is no longer meaningful. Instead the events must be defined by considering what classical information the adversary is able to *find*, rather than just what they *query*. Classically, the information the adversary has about an oracle is entirely specified by the queries being made. But against a quantum adversary, the information an adversary has about an oracle is much more challenging to classify.

1.2 The Multi-User Setting

The security of a protocol is generally defined in terms of a game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. If the adversary is unable to win the game

with a reasonable number of resources, the protocol is considered secure. For example in our situation, $\mathcal{C}$ may be a signing oracle with a public key, and $\mathcal{A}$ may be trying to create a forged signature on that public key.

However in the real world, attackers do not always want to break a specific individual's security. They may be happy to break the security of any of a large number of entities. To model this, we consider an adversary $\mathcal{A}$ that plays a game with a large number of independent challengers $\mathcal{C}_1, \ldots, \mathcal{C}_U$. If $\mathcal{A}$ is able to win the game with any one of these challengers, they are considered to have won. The multi-user setting was first considered in [2].

For many schemes, it is unknown if an adversary's task in winning a game in the multi-user setting is easier or not. In fact there are schemes for which the adversary's chances of winning a game increase linearly with the number of challengers [5]. If a scheme is intended for widespread use, even a linear increase can be a cause for concern that can necessitate an increase in the security parameters. Therefore it is very desirable that any adversary gains no advantage in breaking the security of a scheme in the multi-user setting.

1.3 Our Contributions

- We consider a Lemma by Unruh [17] on distinguishing quantum oracles. We make a small modification that generalizes Unruh's result and addresses oracles that are more commonly considered.
- Develop a heuristic approach to study the properties of a series of composed random oracles.
- Consider the property of undetectability in the random-oracle model.
- Discuss how these can be applied to LMS in order to upper bound any quantum adversary's abilities to break the security of the scheme in the quantum random-oracle model.
- Consider how these results apply to the multi-user setting, where an adversary attempts to break the security of one of many independent instances of the scheme.

1.4 Related Work

The approach for proving LMS in the quantum random-oracle model was largely inspired by the approach in [10], reworking and incorporating modified results from [17, 18]. The quantum-random oracle model was originally defined in [3]. The quantum security of other hash-based constructions, such as Merkle trees and XMSS (another proposed hash-based standard) has been considered before in works such as [4, 9]. In particular [9] considered quantum query bounds on multi-target search problems. A comprehensive report comparing XMSS and LMS [16] has also discussed the need for a quantum random-oracle model proof of LMS. Other works exploring post-quantum signature schemes whose security is established in the quantum random-oracle model include [1, 3, 7]. Undetectability has been considered before to consider the security of the Winternitz one-time signature scheme [6].

2 Scheme Description

2.1 One-Time Scheme

The basic component of the full scheme is the one-time (OT) LMS signature scheme, also known as the Winternitz OT signature scheme. This scheme consists of OT key generation, signing, and verifying algorithms. It uses, as a basic component, a hash function $H : \{0,1\}^* \to \{0,1\}^n$, where n is the security parameter. In our analysis, we will model H as a random oracle.

The parameters are:

- n, the security parameter.
- w, the Winternitz parameter, which is a small divisor of n less than or equal to eight.

These parameters define the following values:

- $E = 2^w - 1$
- $u_1 = n/w$
- $u_2 = \lceil \lfloor \log_2 (u_1 \cdot E) + 1 \rfloor / w \rceil$
- $p = u_1 + u_2$.

For our purposes, string concatenation is denoted by $||$.

We can parse a string of n bits as the concatenation of u_1 strings, each w bits long and representing an integer from 0 to E. This allows us to define the $\mathsf{checksum} : (\{0,1\}^w)^{u_1} \to (\{0,1\}^w)^{u_2}$ function as

$$\mathsf{checksum}(h_1, \ldots, h_{u_1}) = \sum_{i=1}^{u_1} (E - h_i). \tag{2}$$

We can then see that u_2 was chosen so that $w \cdot u_2$ is the maximum bit length of the result of the checksum function.

The checksum function is constructed so that when we compare two vectors of u_1 integers from 0 to E, $(h_1, \ldots, h_{u_1})$ and $(h'_1, \ldots, h'_{u_1})$, if $h_i \leq h'_i$ for each i (and there is at least one index where they are not equal), then when the checksum is viewed as a vector of u_2 integers from 0 to E, $(c_1, \ldots, c_{u_2})$ and $(c'_1, \ldots, c'_{u_2})$, there is an index i such that $c_i > c'_i$. This follows from the fact that if $h_i \leq h'_i$ for all i (and there is at least one index where they are not equal), then $\sum(E - h_i) > \sum(E - h'_i)$, and so when the checksums are converted into integer vectors, at least one of the c_i must be greater than the corresponding c'_i.

We define a function F as a repeated application of H, with each application also adding some additional information, such as the number of times H has been applied. We also include $s = I||Q||i$, a string consisting of an identifying string I for the owner of the public key, a string Q indicating which instance of the scheme is being used, and a number i indicating which chain of hashes we are referring to. This information is used in the multi-user and multi-instance analysis of the scheme. For $0 \leq b \leq f \leq E$, define

$$F_s(x; b, f) = \begin{cases} x & \text{if } b = f \\ F_s(H(x||s||b||00); b+1, f) & \text{if } b < f. \end{cases} \tag{3}$$

The OTLMS algorithms for key generation, signing, and verifying are then described as follows.

Algorithm 1 OTLMSKeyGen

Input: Security parameter 1^n, Winternitz parameter w, identity I, and instance number Q.
Output: Public key pk, secret key sk.

1: Choose p values $x_1^0, x_2^0, \ldots, x_p^0 \in \{0,1\}^n$, uniformly at random.
2: For $i = 1$ to p, let $s = I||Q||i$ and compute $x_i^E = F_s(x_i^0; 0, E)$.
3: Let $pk = H(x_1^E||x_2^E||\ldots||x_p^E||I||Q||01)$.
4: The one-time public key is pk, and the secret key is $sk = (x_1^0, \ldots, x_p^0)$.

Algorithm 2 OTLMSSign

Input: Message $M \in \{0,1\}^*$, secret key sk, identity I, and instance number Q.
Output: Signature σ.

1: Choose a uniformly random $r \in \{0,1\}^n$.
2: Compute $h = H(M||r||I||Q||02)$ and $c = \mathsf{checksum}(h)$. Set $v := h||c$ and parse v as p w-bit integers in $\{0, \ldots, E\}$, $v = (v_1, v_2, \ldots, v_p)$.
3: For $i = 1$ to p, let $s = I||Q||i$ and compute $\sigma_i = F_s(x_i^0; 0, v_i)$.
4: Output signature $\sigma = (r, \sigma_1, \ldots, \sigma_p)$.

Algorithm 3 OTLMSVrfy

Input: Message $M \in \{0,1\}^*$, public key pk (if being used as a standalone scheme), signature $\sigma = (r, \sigma_1, \ldots, \sigma_p)$, identity I, and instance number Q.
Output: accept or reject if being used as a standalone signature scheme, value pk' if being used as part of the full LMS scheme.

1: Compute $h' = H(M||r||I||Q||02)$ and $c' = \mathsf{checksum}(h')$. Set $v' = h'||c'$, and parse v' as p w-bit integers in $\{0, \ldots, E\}$, $v' = (v_1', v_2' \ldots, v_p')$.
2: For $i = 1$ to p, let $s = I||Q||i$ and compute ${x'}_i^E = F_s(\sigma_i; v_i', E)$.
3: Let $pk' = H({x'}_1^E||{x'}_2^E||\ldots {x'}_p^E||I||Q||01)$. If the scheme is used as part of the full scheme, output pk'. If it is being used as a standalone signature scheme, output 'accept' if and only if $pk' = pk$.

The correctness property can verified by inspection. While the OTLMS scheme can seem complicated by its description it is conceptually simple. For key generation, the n-bit random values $x_1^0, \ldots, x_p^0$ are hashed E times to generate the values $x_1^E, \ldots, x_p^E$, which are hashed together to make the public key pk. Any message (along with a random salt r) is hashed to generate a seeded digest h'.

This digest can then be parsed as a series of p integers from 0 to E. These are interpreted as p positions in a 'Winternitz chain' - the number of times x_i^0 is hashed for each i. These repeated hashes are revealed as a signature. To verify a signature, the revealed values are then hashed the correct number of times more to recover $x_1^E, \ldots, x_p^E$, which are all hashed together to get pk.

Readers may be more familiar with the Lamport one-time signature scheme. In that scheme, $2n$ uniformly random n-bit strings form the private key, $(a_{0,1}, a_{1,1}, a_{0,2}, a_{1,2}, \ldots a_{0,n}, a_{1,n})$. Each of these strings is hashed once to form the public key, which also consists of $2n$ bit strings of length n, $(b_{0,1}, b_{1,1}, \ldots, b_{0,n}, b_{1,n})$. To sign an n-bit message digest $h_1 h_2 \ldots h_n$ (with $h_i \in \{0,1\}$) we reveal $a_{h_i,i}$ for $i \in \{1, \ldots, n\}$. In this scheme, public and secret keys are both $2n^2$ bits long, and the signature is n^2 bits long.

The Winternitz one-time scheme and the Lamport one-time scheme are similar in the aspect that both interpret the message digest as a specification for what parts of the secret key should be revealed. Different messages have different digests, and so while part of the secret key has been revealed by one signature, not enough information has been revealed to sign a second message after seeing one signature.

The Winternitz one-time scheme is one of the earliest hash-based schemes, and offers a considerable advantage in terms of key and signature sizes over the Lamport one-time scheme. Its public key is only n bits, and ignoring the salt, its secret key and signature sizes are just $p \cdot n$ as opposed to n^2 or $2n^2$ (for example, for $n = 256$ and $w = 8$, this is 8448 bits as opposed to 65536 bits). It obtains this advantage (at the expense of some additional hashes) by grouping together sections of the salted digest and interpreting these sections as a numeric index in a series of hashes, rather than considering each bit of the digest separately.

2.2 Full Scheme

In the full scheme, we combine the one-time scheme as a subroutine with a Merkle tree construction in order to have a full (stateful) signature scheme.

In addition to the parameters for the one-time scheme, we have the parameter G. We will create 2^G separate instances of the one-time scheme.

Algorithm 4 LMSKeyGen

Input: Security Parameter 1^n, Winternitz parameter w, Merkle tree height 1^G, identity I

Output: Public key pk, secret key sk

1: For $i = 1$ to 2^G, obtain $(pk_i, sk_i) \leftarrow \mathsf{OTLMSKeyGen}(1^n, w, I)$.
2: For $i = 1$ to 2^G, compute $y_i^0 := H(pk_i || I || i || 03)$.
3: For $j = 1$ to G:
   1. For $k = 1$ to 2^{G-j}, compute $y_k^j := H(y_{2k-1}^{j-1} || y_{2k}^{j-1} || k || j || I || 04)$.
4: Output $pk = y_1^G$ as the public key, and $sk = (sk_1, \ldots, sk_{2^G})$ as the secret key.
5: Initialize $Q = 0$.

Algorithm 5 LMSSign

Input: Message $M \in \{0,1\}^*$, secret key sk, identity I
Output: Signature σ

1: Increment Q by 1. If $Q = 2^G + 1$, **STOP**; all signatures have been used.
2: Obtain $\sigma' \leftarrow \mathsf{OTLMSSign}(M, sk_Q, I, Q)$.
3: Let $c \leftarrow Q$. Update $\sigma \leftarrow \sigma'||Q$.
4: For $j = 0$ to $G - 1$:
   1. If c is even, let $\sigma \leftarrow \sigma||y_{c-1}^j$ and $c \leftarrow c/2$.
   2. If c is odd, let $\sigma \leftarrow \sigma||y_{c+1}^j$ and $c \leftarrow (c+1)/2$.
5: Output σ.

Algorithm 6 LMSVrfy

Input: Message $M \in \{0,1\}^*$, public key pk, signature $\sigma = \sigma'||Q||y^0||y^1||\dots||y^{G-1}$, identity I
Output: accept or reject

1: Obtain $pk' \leftarrow \mathsf{OTLMSVrfy}(M, \sigma', I, Q)$.
2: Compute $y = H(pk'||I||Q||03)$.
3: Let $c \leftarrow Q$.
4: For $j = 0$ to $G - 1$:
   1. If c is even, let $y \leftarrow H(y^j||y||c/2||j+1||04)$ and $c \leftarrow c/2$.
   2. If c is odd, let $y \leftarrow H(y||y^j||(c+1)/2||j+1||04)$ and $c \leftarrow (c+1)/2$.
5: Output accept if and only if $y = pk$. Output reject otherwise.

Again, correctness can be verified by inspection. To understand the full scheme, we consider a binary tree, the leaves of which are the public keys of individual one-time schemes. When a message is signed with a one-time scheme, we include the signature of the one-time scheme (in order to generate the public key of that instance), as well as the values of the adjacent nodes on each level of the binary tree in order to be able to recover the value of the root node, which is the overall public key. These values form what is known as the Merkle tree verification path.

3 The (Quantum) Random Oracle

In order to analyze the security of LMS, we need to formulate a few results about the hardness of various problems in the quantum random-oracle model. In Section 3.1 we establish upper bounds on the success probability in standard games such as (second-) preimage resistance in a multi-instance and multi-target setting. In Section 3.2, we consider the difficulty of a slight variant of second-preimage resistance, and in Section 3.3, we consider the properties of functions defined by a composition of random oracles.

3.1 Oracle distinguishing and marked item searching

To establish the hardness of certain fundamental problems, we need a lemma to upper bound a quantum adversary's ability to obtain any relevant information from an oracle. In order to do this, we upper bound an adversary's ability to distinguish two oracles, one which has marked items and one which does not. Furthermore, we would like this upper bound to hold when the adversary has access to multiple independent oracles.

For $\vec{x} = (x_1, \ldots, x_K) \in (\{0,1\}^n)^K$, and $z \in \{0,1\}^n$, let

$$\delta_{\vec{x}}(z) := \begin{cases} 1 & \text{if } z = x_i \text{ for some } i \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

In other words, $\delta_{\vec{x}}$ is a function that outputs 1 on any of K marked items specified by $\vec{x}$. Next we consider the case where there are M independent oracles. Each of these oracles has K marked items, which are chosen independently. We want to consider an adversary $\mathcal{A}$ capable of querying such an oracle in superposition who is attempting to tell if *any* of the oracles have *any* marked items.

Lemma 1. *For* $\mathsf{X} = (\vec{x}_1, \ldots, \vec{x}_M) \in (((\{0,1\}^n)^K)^M$, $z \in \{0,1\}^n$, $j \in \{1, \ldots, M\}$, *and* $b \in \{0,1\}$, *let* U_{X} *be the mapping*

$$U_{\mathsf{X}} : |z\rangle|j\rangle|b\rangle \mapsto |z\rangle|j\rangle|b \oplus \delta_{\vec{x}_j}(z)\rangle. \tag{5}$$

Let $\mathcal{A}$ *be a quantum algorithm making at most* q *queries to a mapping. Let* ρ_b *denote* X *along with the final state of* $\mathcal{A}$ *in the following experiment: Select* $\mathsf{X} = (\vec{x}_1, \ldots, \vec{x}_M) \xleftarrow{\$} (((\{0,1\}^n)^K)^M$. *Run* $\mathcal{A}^{(U_{\mathsf{X}})^b}()$. *Then*

$$Tr\left(\rho_0, \rho_1\right) \leq 2q\sqrt{\frac{K}{2^n}}. \tag{6}$$

This lemma is a straightforward generalization of [17, Lemma 13]. Its proof is very similar, and can be found in Appendix A.

The most straightforward application of this Lemma is to upper bound any adversary's success probability in identifying a marked item in any of a set of oracles that can be queried in superposition.

Lemma 2. *Let* $H_1, \ldots, H_M$ *be independent random oracles with domains* $D_1, \ldots, D_M$ *onto a common range. Let* U_H *be the unitary mapping*

$$U_H : \sum_{x,y,i} \alpha_{x,y,i}|x\rangle|i\rangle|y\rangle \mapsto \sum_{x,y,i} |x\rangle|i\rangle|y \oplus H_i(x)\rangle. \tag{7}$$

Let $S_1, \ldots, S_M$ *be random subsets of the respective* D_i, *such that membership in* S_i *can be tested by a query to* H_i. *We call* S_i *the marked items of* H_i. *Then for any quantum adversary making* q *queries to* U_H, *the probability that they find an* $x \in S_i$ *for any* i *is at most*

$$2q\sqrt{\max_i \left\{\frac{|S_i|}{|D_i|}\right\}}. \tag{8}$$

This lemma follows from Lemma 1 by noting that any adversary that is able to *find* a marked item can certainly distinguish whether a marked item exists. So the bounds on any adversary in Lemma 1 apply, with K being determined by the maximum fraction of marked items.

3.2 Second-preimage Resistance with Adversary Prefixes

Also important to the analysis of LMS is a slight modification of second-preimage resistance, where the adversary is able to specify a prefix of the element whose second preimage they seek. We define this in terms of a game.

Game 1 (Second-preimage Resistance with Adversary Prefixes).

1. $\mathcal{C}$ chooses a random function $H : \{0,1\}^* \to \{0,1\}^n$ from all possible mappings, as well as a random suffix $r' \leftarrow \{0,1\}^n$. $\mathcal{C}$ provides $\mathcal{A}_1$ with oracle access to H.
2. $\mathcal{A}_1$ makes some queries to H, and then outputs some quantum state ρ and a classical message M'.
3. $\mathcal{C}$ runs $\mathcal{A}_2$, with access to H, M', r', and ρ.
4. $\mathcal{A}_2$ makes some queries to H, and then submits an $M^*, r^* \in \{0,1\}^* \times \{0,1\}^n$, with $M' \neq M^*$.

We say that the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ has won if $H(M^*||r^*) = H(M'||r')$.

Classically, it is not difficult to show that an adversary does not obtain much of an advantage. In Katz's paper [10], he tackles this issue through the use of random oracle reprogramming. Specifically, he considers the challenger that, when the adversary submits their prefix M', modifies H to H' so that $H'(M'||r') = h'$, where r' and h' are uniformly random n-bit strings that were chosen at the beginning of the game. The adversary will only notice that $\mathcal{C}$ isn't playing by the 'real' rules of the game if they had previously queried $M'||r'$, and since r' is not disclosed to the adversary in advance, this happens with probability $\leq \frac{q}{2^n}$. Then the probability that an adversary queries a different $M^*||r^*$ such that $H(M^*||r^*) = h'$ is simply $q/2^n$. So we upper bound the probability that the adversary wins this game by $2q/2^n$.

It is much more difficult to prove a similar statement in the quantum setting however. In Katz's proof, an essential step was to reprogram the oracle to reduce to something that more closely resembled second-preimage resistance. Since the adversary has a limited number of queries, they don't have any information about what is reprogrammed with high probability. In the quantum case however, this is much more challenging. Since the adversary can make a quantum superposition of queries, an adversary can make a query giving them some information about the entire oracle. However, the basic approach is still sound — if $\mathcal{C}$ selects a (r', h') and sets $H'(M'||r') = h'$, any adversary should be unable to notice this reprogramming.

For any oracle H, let $H_{M'||r' \mapsto h'}$ denote the oracle identical to H except that the input $M'||r'$ maps to h'.

Game 2.

1. $\mathcal{C}$ chooses a random function $H : \{0,1\}^* \to \{0,1\}^n$ from all possible mappings, as well as a random suffix and outputs $r', h' \leftarrow \{0,1\}^n$. $\mathcal{C}$ provides $\mathcal{A}_1$ with oracle access to H.
2. $\mathcal{A}_1$ makes some queries to H, and then outputs some quantum state ρ and a classical message M'.
3. $\mathcal{C}$ runs $\mathcal{A}_2$, with access to $H_{M'||r' \mapsto h'}$, M', r', and ρ.
4. $\mathcal{A}_2$ makes some queries to $H_{M'||r' \mapsto h'}$, and then submits an $M^*, r^* \in \{0,1\}^* \times \{0,1\}^n$, with $M' \neq M^*$.

$\mathcal{A}_2$ wins Game 2 if $H(M^*||r^*) = h'$.

Lemma 3. *For any $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ making collectively at most q queries to a random oracle H,*

$$\left| \Pr_{Game\ 1}[\mathcal{A}_2 wins] - \Pr_{Game\ 2}[\mathcal{A}_2 wins] \right| \leq \frac{4q}{2^{n/2}}. \tag{9}$$

Roughly speaking, the proof of this lemma follows a technique also seen in [17]. The idea is to introduce two subgames, and show that the difference in the adversary's success probabilities for these games and Games 1 and 2 is at most $2q/2^{n/2}$. This follows from Lemma 1 by showing that any adversary distinguishing between the subgames can also win the game in Lemma 1 with the same probability. The full proof can be found in Appendix B.

We can also imagine the situation where a single adversary $\mathcal{A}$ plays Game 1 with multiple challengers $\mathcal{C}_1, \ldots, \mathcal{C}_U$ with access to multiple independent quantum random oracles $H_1, \ldots, H_U$. Then note that the adversary's chances of success do not increase at all with U. This can be established by considering the same subgames in this multi-user setting. The arguments relating how close the sub-games are still apply, because Lemma 1 does not depend on the number of oracles, as long as each oracle is independent.

3.3 Random Oracle Composition

In the description of LMS, and occasionally in other constructions, a function is defined by a composition of independent random oracles. It would be convenient for this function to itself be a random oracle, or at least have certain properties of a random oracle, from the perspective of both classical and quantum adversaries. However, this is not quite the case.

Let $\mathcal{O}_1, \ldots, \mathcal{O}_E$ be independent random oracles mapping n-bit strings to n-bit strings. Consider the oracle $\mathcal{O} = \mathcal{O}_E \circ \mathcal{O}_{E-1} \circ \cdots \circ \mathcal{O}_1$, $\mathcal{O} : \{0,1\}^n \to \{0,1\}^n$. We want to consider properties of the combined oracle $\mathcal{O}$ with respect to standard properties such as preimage resistance.

Lemma 4. *Let O be a random mapping from a domain $\mathcal{D}$ of size N to a codomain $\mathcal{R}$ of size M. Then the expected size of the image of $\mathcal{D}$ under O is*

$$M\left(1 - \left(1 - \frac{1}{M}\right)^N\right). \tag{10}$$

Proof. Let $\mathcal{R} = \{1, \ldots, M\}$. For each $1 \leq i \leq M$, let X_i be a binary random variable where X_i is 1 if there is an $x \in \mathcal{D}$ such that $O(x) = i$, and 0 otherwise. It is not hard to see that $E[X_i] = 1 - (\frac{M-1}{M})^N$. Then the expected number of elements in the codomain that are hit is $E[X_1 + X_2 + \cdots + X_M] = E[X_1] + E[X_2] + \cdots + E[X_M]$, from which the result follows. □

Writing $N = \alpha \cdot M$, for sufficiently large N and M, Lemma 4 tells us that the fraction of the codomain that is hit is very close to

$$\left(1 - \frac{1}{e^{\alpha}}\right), \tag{11}$$

where $e \approx 2.71828$ is Euler's constant. So when k oracles, each of which maps to a codomain of size 2^n, are composed, the overall oracle maps to an image that has size roughly

$$2^n \cdot \left(1 - \left(\frac{1}{e}\right)^{1-(1/e)^{1-(1/e)^{\cdots}}} \right\} k\right). \tag{12}$$

For example, for $k = 256$, this tells us that after 256 applications of independent random oracles, the final range will be very close to 2^{-7} the size of the original domain. For $k = 1024$, we have the size of the final range is close to 2^{-9} of the original size.

Remark 1. For the rest of this document we will assume that the actual compression for the composed oracles in LMS does not shrink more than four times the expected rate. We will also assume that no more than 256 oracles are used, as this is the most used in any proposed set of LMS parameters. We will assume that the size of the range of 256 applications of an oracle is no smaller than $2^{-10} \cdot 2^n$, which is over four times smaller than the expected size of roughly $2^{-7} \cdot 2^n$. This amount of compression is very unlikely to actually occur, and as actually distinguishing the number of marked items in an oracle is also a exponentially difficult problem, this approach greatly overestimates the compression and the adversary's ability to take advantage of that compression. A much more careful analysis could result in a slightly tighter bound in Theorem 1. However, as this would provide at most a few bits of security in the analysis, we leave this for future work. For further details on the compression of oracles, we refer to Appendix C.

3.4 Undetectability

Often in protocols with random oracles, a value y is selected by choosing a uniformly random point x in the domain of the random oracle H, and setting $y = H(x)$. While the distribution of y is certainly uniform (as H is uniform), the *joint* distribution of (H, y) is not uniform. Therefore an adversary $\mathcal{A}$ that has access to the random oracle may be able to tell if a point in the codomain was chosen uniformly at random or if it was chosen by hashing a uniform point in the domain. This is known as the undetectability property.

Game 3 (Undetectability).

1. $\mathcal{C}$ generates a random oracle $H : \{0,1\}^n \to \{0,1\}^n$, and selects a uniformly random bit $b \xleftarrow{\$} \{0,1\}$.
2. – If $b = 0$, $\mathcal{C}$ sends a uniformly random $y \in \{0,1\}^n$ to $\mathcal{A}$ and provides oracle access to H.
   – If $b = 1$, $\mathcal{C}$ selects a uniformly random $x \in \{0,1\}^n$ and sends $y = H(x)$ to $\mathcal{A}$, and provides oracle access to H.
3. After some queries to H, $\mathcal{A}$ outputs a bit b'.

$\mathcal{A}$ is said to have won Game 3 if $b' = b$.

Lemma 5. *Let $\mathcal{A}$ be a quantum algorithm with oracle access to a random oracle H, making at most q queries. Then*

$$\left| \Pr_{Game\ 3}[\mathcal{A}\ wins] - 1/2 \right| \leq 2q/2^{n/2}. \tag{13}$$

Roughly speaking, this lemma is shown by establishing that the only real way to distinguish whether a point in the codomain was chosen uniformly at random or by first choosing a preimage is to actually *find* that preimage. Finding the preimage can then be tightly reduced to Lemma 1. Furthermore, as Lemma 1 does not depend on the number of instances of the problem, as long as each oracle is independent, the result stays the same when $\mathcal{A}$ is playing multiple, independent instances of Game 3. The full proof can be found in Appendix D.

Similar to Lemma 3, we can imagine an adversary $\mathcal{A}$ playing multiple instances of Game 3 with independent oracles. Then note that this gives no advantage to the adversary's success probability, even if b is chosen to be the same in each game. This is because the reduction to Lemma 1 still holds, with separate marked items in separate independent oracles.

4 Scheme Proof

4.1 OTLMS Proof

Throughout this section, a variable with a $*$ will refer to a value derived from the forgery (M^*, σ^*). A variable with $'$ refers to a value derived in the course of the signing query. If neither are present, it refers to a value derived in the key generation algorithm. We define security in terms of the standard notion of existential unforgeability under chosen-message attack. This standard notion of security is defined in terms of the following interaction between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

Game 4 (One-time existential-unforgeability under chosen-message attack (OTeucma)).

1. $\mathcal{C}$ chooses a random oracle $H : \{0,1\}^* \to \{0,1\}^n$ from all possible mappings (considering that there is in principle an upper bound on the length of binary strings $\mathcal{A}$ will ask for evaluation on). $\mathcal{C}$ then creates a quantum random oracle that provides quantum access to H as in equation 1.

2. $\mathcal{C}$ runs $\mathsf{OTKeyGen}(1^n, w, I, Q)$, obtaining (pk, sk), and sends pk to $\mathcal{A}$.
3. $\mathcal{A}$ makes some queries to the quantum random oracle and then submits a message M' for signing.
4. $\mathcal{C}$ runs $\mathsf{OTSign}(M', sk, I, Q)$ and sends the resulting signature, σ' to $\mathcal{A}$.
5. $\mathcal{A}$ makes some queries to the quantum random oracle, then submits a message-signature pair, (M^*, σ^*), such that $M^* \neq M'$.

We say that $\mathcal{A}$ has won the *OTeucma* game if $\mathsf{OTVrfy}(M^*, \sigma^*, pk, I, Q) \rightarrow \mathsf{accept}$. To bound the adversary's ability to win this, we introduce a separate game:

Game 5 (One-time Simulation).

1. $\mathcal{C}$ Chooses a random oracle $H : \{0,1\}^* \rightarrow \{0,1\}^n$, as well as random strings $r', h' \in \{0,1\}^n$.
2. $\mathcal{C}$ computes $c' = \mathsf{checksum}(h')$ and sets $(v'_1, \ldots, v'_p) = h'||c'$. $\mathcal{C}$ chooses p values $x_1^{v'_1}, \ldots, x_p^{v'_p}$ uniformly at random from $\{0,1\}^n$.
3. For $i = 1$ to p, let $s = I||Q||i$ and compute $x_i^E = F_s(x_i^{v'_i}; v'_i, E)$.
4. Send $pk = H(x_1^E|| \ldots ||x_p^E||I||Q||01)$ to $\mathcal{A}$ and provide oracle access to H.
5. $\mathcal{A}$ makes oracle queries and submits a message M' for signing.
6. $\mathcal{C}$ modified H so that $H(M'||r'||I||Q||02) = h'$, and sends $(r', x_1^{v'_1}, \ldots, x_p^{v'_p})$ as the signature.
7. After further oracle queries, $\mathcal{A}$ submits a message-signature pair (M^*, σ^*) such that $M^* \neq M'$.

As before, $\mathcal{A}$ wins this game if $\mathsf{OTLMSVrfy}(M^*, \sigma^*, pk, I, Q) \rightarrow \mathsf{accept}$.

Lemma 6 (Simulation Difference). *Let $\mathcal{A}$ be a quantum adversary, making at most q queries to a quantum oracle H. Then*

$$\left| \Pr_{Game\ 4}[\mathcal{A}\ wins] - \Pr_{Game\ 5}[\mathcal{A}\ wins] \right| \leq 516q/2^{n/2}. \tag{14}$$

Proof. The difference between these two games is established by applications of Lemmas 3 and 5. There are two differences between Games 4 and 5. The first is that the value h' for the signing query is chosen uniformly at random, and H is later modified so that $H(M'||r'||I||Q||02) = h'$. This introduces a difference of at most $4q/2^{n/2}$ by Lemma 3. The second difference is that values $x_i^{v'_i}$ are chosen uniformly at random, rather than as the output of $F(x_i^0; 0, v'_i)$ for $i = 1$ to p. This introduces a difference of at most $256 \cdot 2q/2^{n/2}$. This can be seen by a game hopping argument. In the original game, $x_i^{v'_i}$ is chosen by computing $F(x_i^0; 0, v'_i)$ for a uniform x_i^0. In the next game, it is chosen by computing $F(x_i^1; 1, v'_1)$ for a uniform x_i^1. By Lemma 5, this only introduces a difference of $2q/2^{n/2}$. Then we repeatedly apply this lemma until we choose $x_i^{v'_i}$ uniformly. As E is at most 256, this needs to be applied at most 256 times, and so the difference is at most $2 \cdot 256q/2^{n/2}$. Thus the overall separation between these games is at most $(4 + 2 \cdot 256)q/2^{n/2}$. □

Theorem 1. *For any adversary $\mathcal{A}$, making at most q quantum queries to the random oracle, the probability that they win Game 4 is at most*

$$580q/2^{n/2}. \tag{15}$$

Proof. This proof is established by showing that the probability an adversary wins Game 5 is at most $64q/2^{n/2}$ so that the result follows from Lemma 6.

To upper bound $\mathcal{A}$'s chances of winning Game 5, we define a few subsets of the domain of H.

- $S_{0,i,j} := \{x \in \{0,1\}^* : x = x'||I||Q||i||j||00,\ F_{I||Q||i}(x; j, E) = x_i^E\}$
- $S_1 := \{x \in \{0,1\}^* : x = {x'}_1^E|| \dots ||{x'}_p^E||I||Q||01,\ H(x) = pk,$
  $({x'}_1^E|| \dots ||{x'}_p^E) \neq (x_1^E|| \dots ||x_p^E)\}$
- $S_2 := \{x \in \{0,1\}^* : x = M||r||I||Q||02,\ H(x) = h', M \neq M'\}$.

Then we define the following three events that may occur over the course of the game $OTeucma$.

- $E0$ is the event that $\mathcal{A}$ has complete knowledge of some $x \in S_{0,i,j}$ for some i and j where $v'_i > j$.
- $E1$ is the event that $\mathcal{A}$ has complete knowledge of some $x \in S_1$.
- $E2$ is the event that $\mathcal{A}$ has complete knowledge of some $x \in S_2$.

These sets correspond to the (second-) preimages that an adversary will have to find in order to break the security of LMS. These events then represent an adversary actually finding such a preimage. Classically, an adversary finding a relevant preimage is exactly characterized by the adversary querying such a point to the random oracle. In a quantum setting however, this equivalence fails as superposition queries are allowed. Instead we characterize the event of an adversary finding such a preimage by whether such a value is derived when running the verification algorithm $\mathsf{OTLMSVrfy}$. This is what we mean by "complete knowledge".

We will establish that if (M^*, σ^*) is a valid forgery, at least one of the three events has occurred. We do this by establishing that in the event of a forgery where events $E1$ and $E2$ did not occur, $E0$ must have happened.

We are assuming that $\mathcal{A}$ has succeeded in submitting a forgery and that events $E1$ and $E2$ have not occurred. We will examine the properties of (M^*, σ^*) and show that $E0$ must have occurred.

When the adversary submits a forgery (M^*, σ^*), we can run the verification algorithm on this pair. Then the following values are derived in the process of running the verification algorithm:

- $M^*||r^*||I||Q||02$
- $x_1^{*E}|| \dots ||x_p^{*E}||I||Q||01$
- $\sigma_i^*||I||Q||i||v_i^*||00$, for $i = 1$ to p.

As $E1$ did not occur, and since the verification algorithm accepts (M^*, σ^*), then we must have that $H(x_1^{*E}||\ldots||x_1^{*E}||I||Q||01) = pk$. So we must have that $x_1^{*E}||\ldots||x_p^{*E}||I||Q||01 \notin S_1$, and so $x_1^{*E}||\ldots||x_p^{*E} = x_1^E||\ldots||x_p^E$.

Similarly, $E2$ did not occur, and since $M^* \neq M'$, it must be the case that $H(M^*||r^*||I||Q||02) \neq h'$.

So we know that $h^* \neq h'$, and that $x_1^{*E}||\ldots||x_p^{*E} = x_1^E||\ldots||x_p^E$. Note that by the construction of the checksum, when we compare v^* and v', there must be an index i for which $v_i^* < v_i'$. But then since we have that $x_i^{*E} = x_i^E$, we can see that this means that $\sigma_i^*||I||Q||i||v_i^*||00 \in S_{0,i,v_i^*}$ and $E0$ has occurred.

All we need to do now is provide an upper bound on the probability of any of the events occurring. To do this, we establish that for any of these events to occur $\mathcal{A}$ must solve some quantum search problem on a distinct search space.

Event *E*0 For event $E0$, we want to consider the adversary's ability to find any new x, i, and j, with $j < v_i'$ and $x \in S_{0,i,j}$. Note that finding an $x \in S_{0,i,j}$ implies complete knowledge of some $x' \in S_{0,i,k}$, for $j \leq k < E$. In particular, it implies complete knowledge of some $x \in S_{0,i,v_i'-1}$. So we need to upper bound the adversaries ability to find such an x.

From the signing query, the adversary knows precisely one element of $S_{0,i,v_i'}$. However, we can imagine an adversary who knows this set entirely. We will show that finding an element of $S_{0,i,v_i'-1}$ is still difficult.

From section 3.3, we know that when considering the function F as a composition of random oracles, we have an expectation on the overall compression from the domain to the codomain, based on the number of applications of H in the construction of F. For typical parameter sets, this is less than 256 times, which corresponds to a compression of roughly 2^7 times. As noted in remark 1, we will take a conservative approach and use a compression factor of four times this, 2^{10}. One consequence is that $S_{0,i,v_i'}$ will have size less than 2^{10} (as the remaining oracles then compress this down to a point).

So we can imagine an adversary that for each i, knows entirely the set $S_{0,i,v_i'}$. The adversary then needs to find an element in $\{0,1\}^n$ that $H(\cdot||I||Q||i||v_i'-1||0)$ maps that point to an element in $S_{0,i,v_i'}$. As $S_{0,i,v_i'}$ has size less than 2^{10}, a fraction less than 2^{10} of the domain maps to these points. So the adversary needs to find a marked item where the fraction of marked items is at most 2^{10-n}.

Event *E*1 Event $E1$ is simply the adversary's ability to find some distinct $x \neq x_1^e||\ldots||x_p^e$ that maps to pk under $H(\cdot||I||Q||01)$, when the adversary is already given such an element. This is a game of second-preimage resistance, so the adversary must find a marked item in the oracle $H(\cdot||I||Q||01)$, where the fraction of marked items is 2^{-n}.

Event *E*2 Event $E2$ refers to the adversary's ability to find a distinct M^* and any r^* such that $H(M^*||r^*||I||Q||02) = H(M'||r'||I||Q||02)$, where M' is chosen by the adversary and r' is chosen uniformly at random. But this is precisely the

game of second-preimage resistance with adversary prefixes with respect to the random oracle $H(\cdot||\cdot||I||Q||02)$. So, the adversary's chances of succeeding differ at most by $4q/\sqrt{2^n}$ from the challenge of finding a marked item in the oracle $H(\cdot||\cdot||I||Q||02)$, where h' is chosen in advance, and the oracle is reprogrammed. In this case the fraction of marked items is 2^{-n}.

We have that the adversary's chances of succeeding are at most $4q/2^{n/2}$ from attempting to find a marked item in any of the distinct oracles defined by I, Q, and $i||v'_i - 1$ for $i = 1$ to p. As the fraction of marked items in any of these oracles is at most 2^{10-n}, the chances of any adversary's success are at most

$$\Pr_{\text{Game 5}}[\mathcal{A} \text{ wins}] \leq 2q\sqrt{2^{10-n}} = 64q/2^{n/2}. \tag{16}$$

And so

$$\Pr_{\text{Game 4}}[\mathcal{A} \text{ wins}] \leq 516q/2^{n/2} + 64q/2^{n/2} = 580q/2^{n/2}. \tag{17}$$

□

4.2 Security Proof for full version and in the multi-user setting

Proving the security of the full version is quite simple having developed the techniques and lemmas used to prove the security of the one-time scheme. By the construction of LMS, all oracles contain different identifying information. We can thus prove security by showing that for an adversary to break the security, they must find a marked item in one of these oracles, and calculating the largest fraction of marked items.

To do this we can use Lemmas 5 and 3 to simulate a signing algorithm similar to how we did in Game 5, but instead for each one-time instance of the signature scheme. As these lemmas can be applied in a multi-instance model without affecting the parameters, we can split up the domain by instance number and identifier information to complete the proof in the full version of the scheme and in the multi-user setting without additional theory.

Theorem 2. *Let $\mathcal{A}$ be an adversary attacking the security of the full LMS scheme in the multi-user setting. If $\mathcal{A}$ makes at most q queries, then the probability they break the existential unforgeability of any of the instances of LMS is at most*

$$580q/2^{n/2}. \tag{18}$$

The complete proof of this theorem may be found in Appendices E and F.

5 Future Work and Discussion

Grover's algorithm implies that any random-oracle analysis of LMS can show that there exists an adversary whose success probability of after q queries is $2q/2^{n/2}$. While the bounds in Theorems 1 and 2 asymptotically match this, there is a difference of a constant factor of 290, suggesting a possible loss in

roughly 8 bits of security over what is expected based off of the most obvious attacks. However it is not clear if there is an attack on LMS that gives such an advantage. This loss in tightness largely comes from applying Lemma 5 a constant number of times in the proof of Lemma 6. More careful analysis in the proof of Lemma 6 could reduce this constant factor.

In our proof, we also had to assume that the number of collisions in the Winternitz chains was much higher than should ever be the case in order to make up for the heuristic technique of assuming how much they actually decreased by. Better understanding of the statistics of repeated application of independent random mappings could greatly assist in tightening up this analysis for a simpler understanding of the Winternitz chains.

In [8], the author proved the security of LMS in a model where the *compression function* of a hash function is assumed to be a random oracle, rather than the entire hash function itself. This is particularly relevant when LMS is implemented with hash functions such as the SHA-2 series where the hash function does not entirely behave as a random oracle, due to the Merkle-Damgård construction. Elevating this analysis to the quantum random-oracle model would provide greater security assurance for the use of LMS with such a hash function in practice.

Acknowledgments

Thanks to Gus Gutoski and Alfred Menezes for insightful discussion, as well as their helpful editorial skills. Additional thanks to Philip Lafrance.

References

1. Alkim, E., Bindel, N., Buchmann, J., Dagdelen, Ö., Eaton, E., Gutoski, G., Krämer, J., Pawlega, F.: Revisiting TESLA in the quantum random oracle model. In: The Eighth International Conference on Post-Quantum Cryptography (PQCrypto). LNCS, vol. 10346, pp. 143–162. Springer (2017)
2. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Advances in Cryptology – CRYPTO '93. LNCS, vol. 775, pp. 232–249 (1994)
3. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Advances in Cryptology – Asiacrypt 2011. LNCS, vol. 7073, pp. 41–69. Springer (2011)
4. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Advances in Cryptology – CRYPTO 2013. LNCS, vol. 8043, pp. 461–478. Springer (2013)
5. Chatterjee, S., Menezes, A., Sarkar, P.: Another look at tightness. In: Selected Areas in Cryptography – SAC 2011. LNCS, vol. 7118, pp. 293–319. Springer (2012)
6. Dods, C., Smart, N., Stam, M.: Hash based digital signature schemes. In: Cryptography and Coding 2005. LNCS, vol. 3796, pp. 96–115. Springer (2005)
7. Eaton, E., Song, F.: Making existential-unforgeable signatures strongly unforgeable in the quantum random-oracle model. In: 10th Conference on the Theory of Quantum Computation, Communication, and Cryptography (TQC). pp. 147–162 (2015)

8. Fluhrer, S.: Further analysis of a proposed hash-based signature standard. Cryptology ePrint Archive, Report 2017/553 (2017), http://eprint.iacr.org/2017/553
9. Hülsing, A., Rijneveld, J., Song, F.: Mitigating multi-target attacks in hash-based signatures. In: Public-Key Cryptography – PKC 2016. LNCS, vol. 9614, pp. 387–416. Springer (2016)
10. Katz, J.: Analysis of a proposed hash-based signature standard. In: Security Standardisation Research: Third International Conference, SSR 2016. LNCS, vol. 10074, pp. 261–273. Springer (2016)
11. Lamport, L.: Constructing digital signatures from a one way function. Tech. rep. (October 1979), https://www.microsoft.com/en-us/research/publication/constructing-digital-signatures-one-way-function/
12. Leighton, F., Micali, S.: Large provably fast and secure digital signature schemes based on secure hash functions (Jul 11 1995), https://www.google.com/patents/US5432852, US Patent 5,432,852
13. McGrew, D., Curcio, M., Fluhrer, S.: Hash-Based Signatures. Internet-Draft draft-mcgrew-hash-sigs-06, Internet Engineering Task Force (March 2017), https://datatracker.ietf.org/doc/html/draft-mcgrew-hash-sigs-06, work in progress
14. Merkle, R.C.: A certified digital signature. In: Advances in Cryptology – Crypto '89. LNCS, vol. 435, pp. 218–238. Springer (1979)
15. Merkle, R.C.: Method of providing digital signatures (Jan 5 1982), https://www.google.com/patents/US4309569, US Patent 4,309,569
16. Panos Kampanakis, S.F.: LMS vs XMSS: A comparison of the stateful hash-based signature proposed standards. Cryptology ePrint Archive, Report 2017/349 (2017)
17. Unruh, D.: Quantum position verification in the random oracle model. In: Advances in Cryptology – CRYPTO 2014. LNCS, vol. 8617, pp. 1–18. Springer (2014)
18. Unruh, D.: Revocable quantum timed-release encryption. Journal of the ACM 62(6), 49:1–49:76 (Dec 2015), http://doi.acm.org/10.1145/2817206

A Proof of Lemma 1

In order to prove Lemma 1, we will need another lemma bounding the distance between states. This lemma appears as Lemma 7 in the full version of [18].

Lemma 7. *Let $|\Psi_1\rangle$ and $|\Psi_2\rangle$ be quantum states that can be written as $|\Psi_i\rangle = |\Psi_i^*\rangle + |\Phi^*\rangle$, with both $|\Psi_i^*\rangle$ orthogonal to $|\Phi^*\rangle$. Then $Tr\left(|\Psi_1\rangle, |\Psi_2\rangle\right) \leq 2||\Psi_2^*\rangle||$.*

The proof of the lemma can be found in the same source. We will now proceed to prove Lemma 1. The proof closely follows the proof of [17, Lemma 13], which addressed the case of one marked item and one oracle.

Proof (Proof of Lemma 1). Let U_i be the unitary that $\mathcal{A}$ makes on their private state after the ith query to U_{X}. If $|\Psi_0\rangle$ is the initial state of $\mathcal{A}$, we may write out the state of $A^{(U_{\mathsf{X}})^b}()$ after i queries as

$$|\Psi_i^{b,\mathsf{X}}\rangle = U_i U_{\mathsf{X}}^b U_{i-1} U_{\mathsf{X}}^b \dots U_1 U_{\mathsf{X}}^b |\Psi_0\rangle. \tag{19}$$

More completely, as the adversary is allowed to maintain private state, we should write U_{X}^b as $I \otimes U_{\mathsf{X}}^b$, where I is the identity mapping on the adversary's private space. For visual simplicity, we omit this.

Then we can write out ρ_b as

$$\rho_b = \sum_{\mathsf{X}} \frac{1}{2^{nKM}} |\mathsf{X}\rangle\langle\mathsf{X}| \otimes |\Psi_q^{b,\mathsf{X}}\rangle\langle\Psi_q^{b,\mathsf{X}}|. \tag{20}$$

Then,

$$\begin{aligned}
& Tr\left(\rho_0, \rho_1\right) && (21)\\
=& Tr\left(\sum_{\mathsf{X}} \frac{1}{2^{nKM}} |\mathsf{X}\rangle\langle\mathsf{X}| \otimes |\Psi_q^{0,\mathsf{X}}\rangle\langle\Psi_q^{0,\mathsf{X}}|, \sum_{\mathsf{X}} \frac{1}{2^{nKM}} |\mathsf{X}\rangle\langle\mathsf{X}| \otimes |\Psi_q^{1,\mathsf{X}}\rangle\langle\Psi_q^{1,\mathsf{X}}|\right) && (22)\\
=& \sum_{\mathsf{X}} \frac{1}{2^{nKM}} Tr\left(|\Psi_q^{0,\mathsf{X}}\rangle\langle\Psi_q^{0,\mathsf{X}}|, |\Psi_q^{1,\mathsf{X}}\rangle\langle\Psi_q^{1,\mathsf{X}}|\right) && (23)
\end{aligned}$$

As a slight abuse of notation, for the sake of simplicity we will write $Tr\left(|\Psi\rangle, |\Phi\rangle\right)$ when we more formally mean $Tr\left(|\Psi\rangle\langle\Psi|, |\Phi\rangle\langle\Phi|\right)$. We compute

$$\begin{aligned}
D_i^{\mathsf{X}} :=& Tr\left(|\Psi_i^{1,\mathsf{X}}\rangle, |\Psi_i^{0,\mathsf{X}}\rangle\right) && (24)\\
=& Tr\left(U_i U_{\mathsf{X}} |\Psi_{i-1}^{1,\mathsf{X}}\rangle, U_i |\Psi_{i-1}^{0,\mathsf{X}}\rangle\right) && (25)\\
=& Tr\left(U_{\mathsf{X}} |\Psi_{i-1}^{1,\mathsf{X}}\rangle, |\Psi_{i-1}^{0,\mathsf{X}}\rangle\right) && (26)\\
\leq& Tr\left(U_{\mathsf{X}} |\Psi_{i-1}^{1,\mathsf{X}}\rangle, U_{\mathsf{X}} |\Psi_{i-1}^{0,\mathsf{X}}\rangle\right) + Tr\left(U_{\mathsf{X}} |\Psi_{i-1}^{0,\mathsf{X}}\rangle, |\Psi_{i-1}^{0,\mathsf{X}}\rangle\right) && (27)\\
=& D_{i-1}^{\mathsf{X}} + Tr\left(U_{\mathsf{X}} |\Psi_{i-1}^{0,\mathsf{X}}\rangle, |\Psi_{i-1}^{0,\mathsf{X}}\rangle\right), && (28)
\end{aligned}$$

where equation 27 follows from the triangle inequality.

As $D_0^{\mathsf{X}} = 0$ for all X, we have that

$$Tr\left(|\Psi_q^{1,\mathsf{X}}\rangle, |\Psi_q^{0,\mathsf{X}}\rangle\right) = D_q^{\mathsf{X}} \leq \sum_{i=0}^{q-1} Tr\left(U_{\mathsf{X}}|\Psi_i^{0,\mathsf{X}}\rangle, |\Psi_i^{0,\mathsf{X}}\rangle\right). \tag{29}$$

We can write out $\mathsf{X} = (\vec{x}_1, \ldots \vec{x}_M)$ in full as

$$\mathsf{X} = \begin{pmatrix} x_{1,1} & x_{2,1} & \ldots & x_{M,1} \\ x_{1,2} & x_{2,2} & \ldots & x_{M,2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,K} & x_{2,K} & \ldots & x_{M,K} \end{pmatrix}. \tag{30}$$

Let Q_{X} be the projector on A's state to the $x_{j,k}$'s. That is,

$$Q_{\mathsf{X}} = \sum_{j=1}^{M}\sum_{k=1}^{K} I \otimes |x_{j,k}\rangle\langle x_{j,k}| \otimes |j\rangle\langle j| \otimes I. \tag{31}$$

Therefore,

$$\sum_{\mathsf{X}} \frac{1}{2^{nKM}} Tr\left(|\Psi_q^{1,\mathsf{X}}\rangle, |\Psi_q^{0,\mathsf{X}}\rangle\right) \leq \sum_{\mathsf{X},i} \frac{1}{2^{nKM}} Tr\left(U_{\mathsf{X}}|\Psi_i^{0,\mathsf{X}}\rangle, |\Psi_i^{0,\mathsf{X}}\rangle\right) \tag{32}$$

$$= \sum_{\mathsf{X},i} \frac{1}{2^{nKM}} Tr\left(U_{\mathsf{X}}Q_{\mathsf{X}}|\Psi_i^{0,\mathsf{X}}\rangle + (1-Q_{\mathsf{X}})|\Psi_i^{0,\mathsf{X}}\rangle, Q_{\mathsf{X}}|\Psi_i^{0,\mathsf{X}}\rangle + (1-Q_{\mathsf{X}})|\Psi_i^{0,\mathsf{X}}\rangle\right) \tag{33}$$

$$\leq \sum_{\mathsf{X},i} \frac{2}{2^{nKM}} ||Q_{\mathsf{X}}|\Psi_i^{0,\mathsf{X}}\rangle|| \tag{34}$$

$$\leq 2\sum_{i} \sqrt{\frac{1}{2^{nKM}} \sum_{\mathsf{X}} ||Q_{\mathsf{X}}|\Psi_i^{0,\mathsf{X}}\rangle||^2}, \tag{35}$$

where equation 34 comes from Lemma 7, and equation 35 comes from Jensen's inequality. Up to this point, the proof is essentially identical to Unruh's proof. Note that $|\Psi_i^{0,\mathsf{X}}\rangle$ is the same for all X, as $|\Psi_i^{0,\mathsf{X}}\rangle$ is independent of X. We drop the X quantifier and expand the state out as

$$|\Psi_i^0\rangle = \sum_{z,j} \alpha_{z,j,i}|\varphi_{z,j,i}\rangle|z\rangle|j\rangle|\phi_{z,j,i}\rangle. \tag{36}$$

Then note that $\sum_{z,j} ||\alpha_{z,j,i}||^2 = 1$, as this is a quantum state. We see that

$$Q_{\mathsf{X}}|\Psi_i^0\rangle = \sum_{j=1}^{M}\sum_{k=1}^{K} \alpha_{x_{j,k},j,i}|\varphi_{x_{j,k},j,i}\rangle|x_{j,k}\rangle|j\rangle|\phi_{x_{j,k},j,i}\rangle, \tag{37}$$

and

$$\sum_{\mathsf{X}} ||Q_{\mathsf{X}}|\Psi_i^0\rangle||^2 \leq \sum_{\mathsf{X}} \sum_{j=1}^{M} \sum_{k=1}^{K} ||\alpha_{x_{j,k},j,i}||^2 \tag{38}$$

$$= \sum_{k=1}^{K} \sum_{j=1}^{M} \sum_{x_{j,k}} \sum_{\substack{x_{j',k'} \\ (j',k') \neq (j,k)}} ||\alpha_{x_{j,k},j,i}||^2 \tag{39}$$

$$= 2^{n(MK-1)} \sum_{k=1}^{K} \sum_{j=1}^{M} \sum_{x_{j,k}} ||\alpha_{x_{j,k},j,i}||^2 \tag{40}$$

$$= K 2^{n(MK-1)}. \tag{41}$$

Applying this to equation 35, we see that

$$\sum_{\mathsf{X}} \frac{1}{2^{nKM}} Tr\left(|\Psi_q^{1,\mathsf{X}}\rangle, |\Psi_q^{0,\mathsf{X}}\rangle\right) \tag{42}$$

$$\leq 2 \sum_{i} \sqrt{\frac{1}{2^{nKM}} K 2^{n(MK-1)}} \tag{43}$$

$$= 2q\sqrt{\frac{K}{2^n}}. \tag{44}$$

□

B Proof of Lemma 3

To prove the lemma, rather than considering the probability of $\mathcal{A}_2$ winning Game 1 or 2, we consider an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that only attempts to distinguish which game they are playing. $\mathcal{A}_2$ will output a bit, and we will establish Lemma 3 by showing that

$$\Pr_{\text{Game 1}}[1 \leftarrow \mathcal{A}_2] - \Pr_{\text{Game 2}}[1 \leftarrow \mathcal{A}_2 : Game2] \leq 4q/2^{n/2} \tag{45}$$

We note that an adversary that wins Game 1 or 2 with different probabilities can be used to distinguish the games, so a bound on the distinguishing advantage also provides a bound on the difference in the probability of winning for any adversary.

Proof. Recall that for any oracle H, we let $H_{M'||r' \mapsto h'}$ denote the oracle identical to H except that the input $M'||r'$ maps to h'. Similarly, let $H_{.||r' \mapsto \perp}$ denote the oracle identical to H except that on input of the string $M||r'$ (for *any* M), we output $\perp$, a symbol distinguishable from any n-bit string. Consider the following game.

We want to show that an adversary's success probability in game 2 is nearly the same as that of game 1. We do this by considering a distinguisher $(\mathcal{A}_1, \mathcal{A}_2)$, attempting to distinguish if they are playing game 1 or 2. We modify the games so that rather than outputting a forgery attempt, $\mathcal{A}_2$ simply outputs a bit to indicate their guess of what game they are playing. Following a technique of Unruh in [17], we introduce two sub-games in order to establish the indistinguishability of the games.

Game 6.

1. $\mathcal{C}$ chooses a random function $H : \{0,1\}^* \to \{0,1\}^n$ from all possible mappings, as well as a random suffix $r' \leftarrow \{0,1\}^n$. $\mathcal{C}$ provides access to $H_{\cdot||r' \mapsto \perp}$ to $\mathcal{A}_1$.
2. $\mathcal{A}_1$ makes some queries to $H_{\cdot||r' \mapsto \perp}$, and then outputs some quantum state ρ and a classical message M'.
3. $\mathcal{C}$ runs $\mathcal{A}_2$, with access to H, M', r', and ρ.
4. $\mathcal{A}_2$ makes some queries to H, and then returns a bit b.

Game 7.

1. $\mathcal{C}$ chooses a random function $H : \{0,1\}^* \to \{0,1\}^n$ from all possible mappings, as well as a random suffix and output $r', h' \leftarrow \{0,1\}^n$. $\mathcal{C}$ provides access to $H_{\cdot||r' \mapsto \perp}$ to $\mathcal{A}_1$.
2. $\mathcal{A}_1$ makes some queries to $H_{\cdot||r' \mapsto \perp}$, and then outputs some quantum state ρ and a classical message M'.
3. $\mathcal{C}$ runs $\mathcal{A}_2$, with access to $H_{M'||r' \mapsto h'}, M', r'$, and ρ.
4. $\mathcal{A}_2$ makes some queries to $H_{M'||r' \mapsto h'}$, and then submits a bit b.

Games 1, 6, 7, and 2 are all the same except for the oracle that $\mathcal{A}_1$ and $\mathcal{A}_2$ are given access to. We summarize the differences between the games in the following table:

Game	Oracle for $\mathcal{A}_1$	Oracle for $\mathcal{A}_2$
1	H	H
6	$H_{\cdot||r' \mapsto \perp}$	H
7	$H_{\cdot||r' \mapsto \perp}$	$H_{M'||r' \mapsto h'}$
2	H	$H_{M'||r' \mapsto h'}$

First, note that

$$\Pr_{\text{Game } 6}[1 \leftarrow \mathcal{A}_2] = \Pr_{\text{Game } 7}[1 \leftarrow \mathcal{A}_2]. \tag{46}$$

This follows from the fact that because the oracle $\mathcal{A}_1$ had access to contained *no* information about $H(M||r')$ for all M, the distribution of the adversary's output with oracle H or $H_{M'||r' \mapsto h'}$ are exactly the same, even when given access to $H_{\cdot||r' \mapsto \perp}$ in order to prepare the private state ρ.

Next we note that

$$\epsilon := \Big| \Pr_{\text{Game } 1}[1 \leftarrow \mathcal{A}_2] - \Pr_{\text{Game } 6}[1 \leftarrow \mathcal{A}_2] \Big| \leq \frac{2q}{2^{n/2}}. \tag{47}$$

Consider any adversary that distinguishes between game 1 and 6. Let M'_1, ρ_1 denote the output of $\mathcal{A}_1$ in game 1, and M'_6, ρ_6 denote the output of $\mathcal{A}_1$ in game 6. As $\mathcal{A}_2$ receives identical inputs and works with the same oracle in each game, we can view it as an identical quantum channel acting on the state $|M'\rangle\langle M'| \otimes \rho$. As any quantum channel can only decrease the distinguishability between two states, we have that

$$\epsilon \leq Tr\left(|M'_1\rangle\langle M'_1| \otimes \rho_1, |M'_6\rangle\langle M'_6| \otimes \rho_6\right). \tag{48}$$

Notice that these states are the output of $\mathcal{A}_1$, which makes at most q queries to either H or $H_{.||r' \mapsto \perp}$. We can show that the trace distance between these states is small by showing that $\mathcal{A}_1$ can be used as a distinguisher in the game of Lemma 1. Consider a reduction $\mathcal{B}$ which generates a random oracle $\mathcal{O} : \{0,1\}^* \times \{0,1\}^n \to \{0,1\}^n$. The reduction is given quantum access to a function f which either does nothing, or implements the function δ_x for some uniform x. Then $\mathcal{B}$ constructs a hash oracle by

$$H(M||r) = \begin{cases} \mathcal{O}(M||r) & \text{if } f(r) = 0 \\ \perp & \text{if } f(r) = 1 \end{cases}. \tag{49}$$

Then $\mathcal{B}$ can play the game in the definition of Lemma 1, using $\mathcal{A}_1$ as a subroutine, and making one query to f for each query to H that $\mathcal{A}_1$ makes. So since the trace distance of $\mathcal{B}$'s output is bounded by $2q/2^{n/2}$ (as $K = 1$ in this case), we have that the same is true of the output of $\mathcal{A}_1$, and so $\epsilon \leq 2q/2^{n/2}$.

For the same reason, we can note that

$$\left| \Pr_{\text{Game 7}}[1 \leftarrow \mathcal{A}_2] - \Pr_{\text{Game 2}}[1 \leftarrow \mathcal{A}_2] \right| \leq \frac{2q}{2^{n/2}}. \tag{50}$$

This tells us that

$$\left| \Pr_{\text{Game 1}}[1 \leftarrow \mathcal{A}_2] - \Pr_{\text{Game 2}}[1 \leftarrow \mathcal{A}_2] \right| \leq \frac{4q}{2^{n/2}}. \tag{51}$$

□

C Notes on Oracle Compression

In Section 3.3, we found the expected size of the image of a random oracle $\mathcal{O} : \mathcal{D} \to \mathcal{R}$, where $|\mathcal{D}| = N$ and $|\mathcal{R}| = M$. We then used this to find the expected amount of 'compression' that occurs from a composition of random oracles. In Section 4.1, we used this to bound the fraction of marked items in an oracle, by assuming that actual compression was no more than four times this amount.

In this appendix, we provide a more formal argument that the compression is extremely unlikely to be more than four times the expected value. We do this by considering the variance in the size of the image of a random oracle, and applying Chebyshev's inequality.

Lemma 8. *Let O be a random mapping from a domain $\mathcal{D}$ of size N to a codomain $\mathcal{R}$ of size M. Let X be the size of the image of $\mathcal{D}$ under O. Then*

$$Var[x] = M\left(1-\frac{1}{M}\right)^N - M\left(1-\frac{2}{M}\right)^N \tag{52}$$

$$+M^2\left(1-\frac{2}{M}\right)^N - M^2\left(1-\frac{1}{M}\right)^{2N}. \tag{53}$$

Proof. Write $\mathcal{R} = \{1,\ldots,M\}$. Similar to our proof of Lemma 4, for $1 \le i \le M$, let X_i be a binary random variable where X_i is 1 if there is a $x \in \mathcal{D}$ such that $O(x) = i$, and 0 otherwise. We noted before that $E[X_i] = 1 - \left(\frac{M-1}{M}\right)^N$.

We now consider $E[X_iX_j]$. When $i = j$, this is $E[X_i^2]$, and as X_i is binary, this is just $E[X_i]$. When $i \neq j$, we have

$$E[X_iX_j] = \Pr[X_i = 1 \text{ and } X_j = 1] \tag{54}$$

$$= 1 - \Pr[X_i = 0] - \Pr[X_j = 0] + \Pr[X_i = 0 \text{ and } X_j = 0] \tag{55}$$

$$= 1 - 2\left(1-\frac{1}{M}\right)^N + \left(1-\frac{2}{M}\right)^N. \tag{56}$$

Then we calculate

$$Var[X] = Var[X_1 + \cdots + X_M] \tag{57}$$

$$= E[(X_1 + \cdots + X_M)^2] - E[X_1 + \cdots + X_M]^2 \tag{58}$$

$$= \sum_i E[X_i] + \sum_{i\neq j} E[X_iX_j] - M^2\left(1-\frac{1}{M}\right)^{2N} \tag{59}$$

$$= M\left(1-\frac{1}{M}\right)^N + M(M-1)\left(1 - 2\left(1-\frac{1}{M}\right)^N + \left(1-\frac{2}{M}\right)^N\right) \tag{60}$$

$$- M^2\left(1-\frac{1}{M}\right)^{2N} \tag{61}$$

$$= M\left(1-\frac{1}{M}\right)^N - M\left(1-\frac{2}{M}\right)^N \tag{62}$$

$$+ M^2\left(1-\frac{2}{M}\right)^N - M^2\left(1-\frac{1}{M}\right)^{2N}. \tag{63}$$

□

Note that for large values of M and N, $(1-2/M)^N$ is extremely close to $(1-1/M)^{2N}$. This follows from using the approximation $(1+x/n)^n \approx e^x$, and noting that the approximation converges very very quickly.

This allows us to simplify the variance to

$$M\left(1-\frac{1}{M}\right)^N - M\left(1-\frac{2}{M}\right)^N. \tag{64}$$

Writing $N = \alpha \cdot M$, and using the same approximation, we may further simplify this to

$$\frac{1}{e^{\alpha}} M(1 - (1 - \frac{1}{M})^N). \tag{65}$$

In other words, the variance is approximately $1/e^{N/M}$ times the expected value.

As N shrinks, we can see that this will approach 1. Therefore we put a general bound on the variance and say that it is less than or equal to the expected value.

Let $Var[x] = \sigma^2$ and $E[X] = \mu$. Then $\sigma^2 \leq \mu$. Let β be a real number. Then Chebyshev's inequality tells us

$$\Pr[|X - \mu| \geq \beta\mu] \leq \frac{1}{\beta^2 \mu}. \tag{66}$$

For LMS, the expected value of X is huge - a constant fraction of the codomain $\{0,1\}^n$. So if β is also a constant fraction, this probability is negligible. When we use the results of the oracle compression, we allow for an overall compression from the domain to the final range four times the expected value. Since the variance tells us that the actual compression stays much closer to the expected value than this, this bound is very conservative.

D Proof of Lemma 5

In order to prove Lemma 5 we will reduce Game 3 to a reduction that attempts to detect the existence of a marked item in an oracle as in Lemma 1. Specifically, we have a reduction $\mathcal{B}$ that is given oracle access to a function $O : \{0,1\}^n \to \{0,1\}$ such that with probability $1/2$ exactly one $x \in \{0,1\}^n$ satisfies $O(x) = 1$, and with probability $1/2$ O maps all strings to 0. $\mathcal{B}$ must attempt to correctly guess whether O has a marked item or not. Lemma 1 then tells us that $\mathcal{B}$'s advantage in q queries over $1/2$ is at most $2q/2^{n/2}$.

Game 8.

1. $\mathcal{B}$ chooses a uniform $y \xleftarrow{\$} \{0,1\}^n$. $\mathcal{B}$ constructs a uniformly random oracle $H : \{0,1\}^n \to \{0,1\}^n$, and constructs an oracle H' as

$$H'(x) = \begin{cases} H(x) & \text{if } O(x) = 0 \\ y & \text{if } O(x) = 1. \end{cases} \tag{67}$$

2. $\mathcal{B}$ sends y to $\mathcal{A}$ and provides oracle access to H'.
3. $\mathcal{A}$ eventually submits a bit b'. If $b' = 0$, $\mathcal{B}$ guesses that O has no marked items, and if $b' = 1$, $\mathcal{B}$ guesses that it does.

First note that

$$\Pr_{\text{Game 3}}[0 \leftarrow \mathcal{A} | b = 0] = \Pr_{\text{Game 8}}[0 \leftarrow \mathcal{A} | \text{No marked item}]. \tag{68}$$

This can be seen by considering the joint distribution of the random oracle that $\mathcal{A}$ has access to and y. In each case, the random oracle that $\mathcal{A}$ has access to is entirely uniform, and y is a uniform n-bit string independent of H.

Similarly, we can see that

$$\Pr_{\text{Game 3}}[1 \leftarrow \mathcal{A}|b = 1] = \Pr_{\text{Game 8}}[1 \leftarrow \mathcal{A}|\text{One marked item}]. \tag{69}$$

In both games, H can be thought of as a uniformly random and independent series of 2^n n-bit strings, while the distribution of y is specified by choosing a uniform index and selecting the n-bit string at that index in H.

Then,

$$\Pr_{\text{Game 3}}[\mathcal{A} \text{ wins}] \tag{70}$$

$$= \frac{1}{2} \Pr_{\text{Game 3}}[0 \leftarrow \mathcal{A}|b = 0] + \frac{1}{2} \Pr_{\text{Game 3}}[1 \leftarrow \mathcal{A}|b = 1] \tag{71}$$

$$= \frac{1}{2} \Pr_{\text{Game 8}}[0 \leftarrow \mathcal{A}|\text{No marked item}] + \frac{1}{2} \Pr_{\text{Game 8}}[1 \leftarrow \mathcal{A}|\text{One marked item}] \tag{72}$$

$$= \Pr_{\text{Game 8}}[\mathcal{B} \text{ wins}]. \tag{73}$$

Therefore,

$$\left| \Pr_{\text{Game 3}}[\mathcal{A} \text{ wins}] - \frac{1}{2} \right| = \left| \Pr_{\text{Game 8}}[\mathcal{B} \text{ wins}] - 1/2] \right| \leq 2q/2^{n/2}. \tag{74}$$

E Security Proof for LMS in the QROM

For the full version of LMS, we proceed in a similar fashion, defining a full game of existential-unforgeability under chosen-message attack, and upper bounding the success probability of any adversary in this game with respect to a random oracle.

Game 9 (LMS Existential-unforgeability under chosen message attack (eucma)).

1. $\mathcal{C}$ chooses a random function $H : \{0,1\}^* \to \{0,1\}^n$ from all possible mappings. $\mathcal{C}$ then creates a quantum random oracle that provides quantum access to H as in equation 1.
2. $\mathcal{C}$ chooses an arbitrary identity I and runs $\mathsf{LMSKeyGen}(1^n, w, 1^G, I)$, obtaining (pk, sk), and sends pk, I to $\mathcal{A}$.
3. $\mathcal{A}$ makes some queries to the quantum random oracle and then submits a message M'_1 for signing.
4. $\mathcal{C}$ runs $\mathsf{LMSSign}(M'_1, sk, I)$ and sends the resulting signature, σ'_1 to $\mathcal{A}$.
5. $\mathcal{A}$ continues to make random oracle queries, as well as signing queries M'_i for $i = 2, \ldots, 2^G$.
6. $\mathcal{A}$ submits a message-signature pair, (M^*, σ^*), such that $M^* \neq M'_i$ for all i.

As before, $\mathcal{A}$ is said to win if and only if $\mathsf{LMSVrfy}(M^*, pk, \sigma^*, I) \rightarrow$ 'accept'.

Theorem 3. *For any adversary $\mathcal{A}$, making at most q quantum queries to the random oracle, the probability that they win Game 9 is at most*

$$580q/2^{n/2}. \tag{75}$$

Proof. To upper bound $\mathcal{A}$'s probability of winning game 9, we again characterize several subsets and events based on these subsets, and characterize what a forgery may look like with respect to these subsets. We also consider a challenger that simulates the signing algorithm for each one-time instance of the scheme, independently, as in Game 5. Although we are now simulating the signing algorithm 2^G times, we still have that the difference in the adversary's success probability is at most $516q/2^{n/2}$. This follows from the fact that all of the oracles are independent and so Lemma 3 and 5 are not altered in a multi-instance setting.

Parse the forged signature as $\sigma^* = \sigma'^*, Q^*, y^{*0}, \ldots, y^{*H-1}$. σ'^* is the signature for the one-time verification algorithm, Q^* is the placement identifier, and $y^{*0}, \ldots, y^{*H-1}$ form the Merkle tree verification path. Let $pk'^* \leftarrow \mathsf{OTVrfy}(M^*, \sigma'^*)$. Define the following sets:

- $S3_{Q^*} = \{x \in \{0,1\}^* : x = pk'^*||I||Q^*||03, H(x) = y^0_{Q^*}, pk'^* \neq pk_{Q^*}\}$
- $S4_{j,k} = \{x \in \{0,1\}^* : x = y_1||y_2||k||j||I||04, H(x) = y^j_k, y_1||y_2 \neq y^{j-1}_{2k-1}||y^{j-1}_{2k}\}$.

We will decompose the event of a forgery into one of three events:

- EOT_{Q^*} is the event that $pk'^* = pk_{Q^*}$.
- $E3_{Q^*}$ is the event that the adversary has perfect knowledge of some $x \in S3_{Q^*}$.
- $E4_{j,k}$ is the event that the adversary has perfect knowledge of some $x \in S4_{j,k}$ for some j, k.

It is straightforward to establish, as we did for the one-time signature scheme, that for a forgery to occur, one of these events must happen. We establish it in the same way, that is, if EOT and $E3_{Q^*}$ do not occur for a forgery, then $E4_{j,k}$ must occur.

$E3_{Q^*}$ and EOT_{Q^*} not occurring means that $H(pk'^*||I||Q^*||03) \neq y^0_{Q^*}$, and so setting this to x', $i = Q^*$ and considering the $y^{*0}, \ldots, y^{*H-1}$, we know that since the signature verified, this verification path must lead to pk. Therefore, at some point along the verification path, there is a collision onto a value Y^j_k, and $E4_{j,k}$ has occurred.

Event EOT_{Q^*} This event corresponds to the possibility of an adversary breaking the one-time signature scheme for one of the Q^* instances of the one-time signature scheme. It therefore decomposes into the previously defined events $E0$, $E1$, or $E2$, but in a multi-instance model, as the adversary now has the freedom to choose any $Q^* \in \{1, \ldots, 2^G\}$.

Event $E3_{Q^*}$ This is the event that the adversary finds a second-preimage of one of the leaf nodes of the Merkle tree. So this event corresponds to the adversary finding a marked item in the oracle $H(\cdot||I||Q^*||03)$ for some $Q^* \in \{1, \ldots, 2^G\}$, where the fraction of marked items is 2^{-n}.

Event $E4_{j,k}$ In this event, the adversary has found a second-preimage to y_k^j under the oracle $H(\cdot|| \cdot ||k||j||I||04)$, for some j, k of their choosing. This again corresponds to finding a marked item in one of the oracles $H(\cdot|| \cdot ||k||j||I||04)$, where the fraction of marked items is 2^{-n}.

Similar to our analysis in Section 4.1, we have that any adversary's chances of succeeding are at most $516q/2^{n/2}$ from attempting to find a marked item in any of the oracles. As the fraction of marked items in any of the oracles is still at most 2^{10-n}, the chances of the adversary's success are still at most

$$580q/2^{n/2}. \tag{76}$$

□

F Security Proof in the multi-user setting

We now consider LMS in a multi-user setting. We first define security in this setting.

Game 10 (LMS eucma in the multi-user setting).

1. $\mathcal{C}$ chooses a random function $H : \{0,1\}^* \to \{0,1\}^n$ from all possible mappings. $\mathcal{C}$ then creates a quantum random oracle that provides quantum access to H as in equation 1.
2. $\mathcal{C}$ chooses U distinct identity strings, $I_1, \ldots I_U$. $\mathcal{C}$ runs $\mathsf{LMSKeyGen}(1^n, w, 1^G, I_i)$ for $i = 1, \ldots, U$, obtaining $(pk_1, sk_1), \ldots, (pk_U, sk_U)$. $\mathcal{C}$ sends the public keys to $\mathcal{A}$.
3. $\mathcal{A}$ makes queries to the random oracle as well as message queries. For $\mathcal{A}$'s message queries, they must specify which public key i they want a signature under.
4. $\mathcal{C}$ signs the message with $\mathsf{LMSSign}(M', sk_i, I_i)$, and sends the signature to $\mathcal{A}$.
5. $\mathcal{A}$ can request up to 2^G signatures on each public key pk_i.
6. $\mathcal{A}$ submits a message M^*, a signature σ^*, and an index i^* such that M^* was not one of the signed messages on pk_{i^*}.

$\mathcal{A}$ is said to have won the game if $\mathsf{LMSVrfy}(M^*, pk_i, \sigma^*, I_i) \to$ 'accept'.

Theorem 4. *For any adversary $\mathcal{A}$, making at most q quantum queries to the random oracle, the probability that they win Game 10 is at most*

$$580q/2^{n/2}. \tag{77}$$

Proof. Any forgery can be decomposed into the event that an adversary forges on public key pk_i for some $i \in \{1, \ldots, U\}$. This event can in turn be broken up into the forgery events for the LMS scheme and the one-time LMS scheme. We have previously established that the probability of an adversary causing any of these events is at most $516q/2^{n/2}$ from finding a marked item in some oracle. As each of these oracles contains the identifier I_i, and each I_i is distinct, the adversary gains no advantage in finding one of these marked items.

So similar to our analysis of LMS in the single-user setting, we have that the probability $\mathcal{A}$ wins game 10 is at most $580q/2^{n/2}$. □