Analysis of a Proposed Hash-Based Signature Standard

Jonathan Katz[*]

Abstract

We analyze a signature scheme described in a recent Internet Draft, and highlight a variant (based on prior work of Micali and Leighton) that offers improved concrete security.

1 Overview

McGrew [5] recently proposed a standard for hash-based digital signatures. The proposed construction instantiates Merkle's tree-based approach [6, 7] with the one-time signature scheme of Lamport-Diffie-Winternitz-Merkle [3, 6, 7] (the *LDWM scheme*). The concrete security of the construction depends on the concrete security of the LDWM scheme in the *multi-instance setting*, where multiple public keys are generated and an attack is successful if it results in a forged signature with respect to any of those keys. Here, we analyze the concrete security of the LDWM scheme in the multi-instance setting, and highlight some modifications (previously proposed by Leighton and Micali [4]) that yield better concrete security in this setting.

2 Description of the Proposed Scheme

We begin with a detailed description of the LDWM scheme, following [5]. Let $H : \{0,1\}^* \to \{0,1\}^{8n}$ and $F : \{0,1\}^{8m} \to \{0,1\}^{8m}$ be functions that we will treat as random oracles. Let F^i, for integer $i \geq 1$, denote i-fold iterated application of F, and let F^0 denote the identity function. Fix

[*] Dept. of Computer Science, University of Maryland. Email: `jkatz@cs.umd.edu`. Work performed under a consultancy agreement with University Technical Services, Inc. on behalf of the National Security Agency. We thank Laurie E. Law and Jerome A. Solinas for bringing [4] to our attention, and for suggesting we write this paper.

$w \in \{1,2,4,8\}$ as a parameter of the scheme, and set $e \overset{\text{def}}{=} 2^w - 1$. Set $u \overset{\text{def}}{=} 8n/w$; note that outputs of H can be viewed as a sequence of u integers, each exactly w bits long. Set $v \overset{\text{def}}{=} \lceil \lfloor \log u \cdot (2^w - 1) + 1 \rfloor / w \rceil$, and define $\mathsf{checksum} : (\{0,1\}^w)^u \to \{0,1\}^{wv}$ as follows:

$$\mathsf{checksum}(h_0, \ldots, h_{u-1}) = \sum_{i=0}^{u-1} (2^w - 1 - h_i)\,,$$

where each $h_i \in \{0,1\}^w$ is viewed as an integer in the range $\{0, \ldots, 2^w - 1\}$ and the result is written as an integer using exactly wv bits. Set $p \overset{\text{def}}{=} u + v$.

Define a one-time signature scheme as follows:

Key generation

1. Choose p uniform values $x_0, \ldots, x_{p-1} \in \{0,1\}^{8m}$.
2. For $i = 0$ to $p - 1$, compute $y_i := F^e(x_i)$.
3. Compute $pk := H(y_0, \ldots, y_{p-1})$.

The public key is pk, and the private key is $x_0, \ldots, x_{p-1}$.

Signing

To sign a message $M \in \{0,1\}^*$ using private key $x_0, \ldots, x_{p-1}$ do:

1. Compute $h := H(M)$ and $c := \mathsf{checksum}(h)$. Set $V := h \| c$, and parse V as a sequence of w-bit integers $V_0, \ldots, V_{p-1}$.
2. For $i = 0, \ldots, p - 1$, compute $\sigma_i := F^{V_i}(x_i)$.
3. Return the signature $\sigma_0, \ldots, \sigma_{p-1}$.

Verifying

To verify a signature $\sigma_0, \ldots, \sigma_{p-1}$ on a message $M \in \{0,1\}^*$ with respect to the public key pk do:

1. Compute $h := H(M)$ and $c := \mathsf{checksum}(h)$. Set $V := h \| c$, and parse V as a sequence of w-bit integers $V_0, \ldots, V_{p-1}$.
2. For $i = 0, \ldots, p - 1$, compute $y_i := F^{e - V_i}(\sigma_i)$.
3. Return 1 if and only if $pk = H(y_0, \ldots, y_{p-1})$.

3 Security Analysis of the Proposed Scheme

We assume the reader is familiar with the standard notion of security for one-time signature schemes (see [2]). When the output lengths n, m (in bytes) of the hash functions are sufficiently large, as they are in [5], the LDWM scheme can be proven secure when H, F are modeled as random oracles [1]. Here, however, we are interested in *concrete security*, and so we explore how large n, m need to be in order to ensure "k-bit security," i.e., to ensure that an attacker needs to invest roughly 2^k work in order to forge a signature with probability close to 1. We measure work in terms of the number q of H- and F-evaluations performed. This is a somewhat coarse measure, and a more refined analysis would also take into account memory usage as well as the effect of parallelization. Nevertheless, this measure serves as a good first approximation to the difficulty of forging a signature.

Finding a collision in H. A signature forgery is possible if a collision in H can be found. By evaluating H a total of q times, a collision can be found with probability roughly $q^2 \cdot 2^{-8n}$ using a standard "birthday" attack. Thus, to ensure k-bit security the output length of H must be at least $2k$ bits.

Multiple public keys I. If N instances of the LDWM scheme are run, either by the same signer or by multiple signers, then security degrades linearly in N. To see this, note that the ith public key pk^i has the form

$$pk^i = H(y_0^i, \cdots, y_{p-1}^i)\,.$$

Consider computing the Q values $y_0^* = F^e(x_0^*), \ldots, y_{Q-1}^* = F^e(x_{Q-1}^*)$, for distinct x_i^*, and evaluating H on all (ordered) length-p lists of the y_i^*. (There are $q \overset{\text{def}}{=} Q!/(Q-p)!$ such lists. Note that $eQ \ll q$ for "interesting" values of Q, so the overall work is dominated by the q evaluations of H.) If any of the resulting hashes is equal to some pk^i, then it becomes trivial to forge arbitrary signatures with respect to that public key. The probability that this occurs is roughly $qN \cdot 2^{-8n}$. In particular, to ensure k-bit security the output length of H must be at least $k + \log N$ bits.

Multiple public keys II. A similar issue as above arises because F is used in all instances of the scheme. Here we show that if N instances of the LDWM scheme are run then security degrades linearly in pN.

Let pk^i, for $1 \leq i \leq N$, denote the ith public key, and assume a signature with respect to each public key has been released so that, in particular, values $y_0^i, \ldots, y_{p-1}^i$ with $pk^i = H(y_0^i, \ldots, y_{p-1}^i)$ are known for all i. Consider evaluating F^e on q/e random inputs, looking for an input x such that

$F^e(x) = y^i_j$ for some i, j. If such an x is found, a forgery becomes possible with high probability.[1] The probability that such an x is found is roughly $(q/e) \cdot pN \cdot 2^{-8m}$. (Small variants of this approach, having slightly better parameters, are also possible.) Thus, to ensure k-bit security the output length of F must be at least $k + \log N + \log p - \log e$ bits.

4 Suggested Improvements

As observed by Leighton and Micali [4], it is possible to achieve k-bit security with reduced hash lengths by modifying the LDWM scheme. (Micali and Leighton conjecture that these modifications achieve better concrete security; we provide a proof in the appendix.)

The first modification is to have the signer choose a uniform value r at the time of signing and then set $h := H(r, M)$; the rest of the signing algorithm remains the same, except that r is included as part of the signature. Now, finding an arbitrary collision in H is not sufficient to forge a signature; instead, given a signature on M using randomness r a signature forgery is possible only if one can find r', M' such that $H(r', M') = H(r, M)$. Given r, M and evaluating H a total of q times, such values r', M' can be found with probability only $q \cdot 2^{-8n}$, suggesting that k-bit security is achieved if the output length of H is k bits—half the size as before. (This assumes r is long enough so that it is infeasible to guess it in advance; see further below.)

The discussion above assumes a single instance of the scheme. When N instances of the scheme are used, however, security degrades by a factor of N even if the above modification is used. This impacts both the required length of the output of H as well as the length of r. Specifically:

- Say M^i is signed by the ith instance of the scheme using randomness r^i. A signature forgery is possible if one can find r', M' such that $H(r', M') = H(r^i, M^i)$ for any i. With q evaluations of H, this occurs with probability roughly $qN \cdot 2^{-8n}$, implying that for k-bit security the output length of H must be at least $k + \log N$ bits.

- Say values r, M, r', M' ($M' \neq M$) are found with $H(r, M) = H(r', M')$, and then a signature on M is obtained with respect to each of the N instances of the scheme. A signature forgery (on M') is now possible if any of those N signatures use randomness r; this occurs with probability roughly $N \cdot 2^{-|r|}$, implying that for k-bit security the length of r must be at least $k + \log N$ bits.

[1] A precise calculation depends on the messages that have already been signed.

This motivates using another idea suggested by Leighton and Micali [4]: ensuring that each evaluation of H and[2] F by the (honest) signers is done on an element from a distinct domain. This can be achieved by having each signer prepend their identity, an instance number (in case the same signer runs multiple instances of the scheme), and a 2-bit identifier to each H- or F-evaluation. The identity and instance number are included as part of the public key and used during signature verification.

We incorporate both the above modifications into the scheme described next. Assume $m \leq n$, and let $H'(x)$ be equal to $H(x)$ truncated to $8m$ bits. For an arbitrary string s, set $F_s^1(x) \overset{\text{def}}{=} H'(s, 1, x)$, and for integer $e > 1$ define $F_s^e(x) \overset{\text{def}}{=} H'(s, e, F_s^{e-1}(x))$ where e is encoded using exactly w bits. (F_s^0 is still the identity function.)

Key generation

Let I denote the identity of the signer, and let num denote an instance number. (We require that no signer uses the same instance number twice.) We assume that I and num are each encoded using some fixed number of bits. Key generation then proceeds as follows:

1. Choose p uniform values $x_0, \ldots, x_{p-1} \in \{0,1\}^{8m}$.
2. For $i = 0$ to $p-1$, compute $y_i := F^e_{00,I,\mathsf{num},i}(x_i)$, where i is encoded using some fixed number of bits.
3. Compute $pk := H(01, I, \mathsf{num}, y_0, \ldots, y_{p-1})$.

The public key is (I, num, pk), and the private key is $sk = (I, \mathsf{num}, x_0, \ldots, x_{p-1})$.

Signing

To sign $M \in \{0,1\}^*$ using private key $sk = (I, \mathsf{num}, x_0, \ldots, x_{p-1})$ do:

1. Choose uniform $r \in \{0,1\}^{8m}$.
2. Compute $h := H(11, I, \mathsf{num}, r, M)$ and $c := \mathsf{checksum}(h)$. Set $V := h\|c$, and parse V as a sequence of w-bit integers $V_0, \ldots, V_{p-1}$.
3. For $i = 0, \ldots, p-1$, compute $\sigma_i := F^{V_i}_{00,I,\mathsf{num},i}(x_i)$.
4. Return the signature $\sigma = (r, \sigma_0, \ldots, \sigma_{p-1})$.

[2] Note that F and H might be the same function.

Verifying

To verify a signature $\sigma = (r, \sigma_0, \ldots, \sigma_{p-1})$ on a message $M \in \{0,1\}^*$ with respect to the public key (I, num, pk) do:

1. Compute $h := H(11, I, \mathsf{num}, r, M)$ and $c := \mathsf{checksum}(h)$. Set $V := h \| c$, and parse V as a sequence of w-bit integers $V_0, \ldots, V_{p-1}$.
2. For $i = 0, \ldots, p-1$, compute $y_i := F^{e-V_i}_{00,I,\mathsf{num},i}(\sigma_i)$.
3. Return 1 if and only if $pk = H(01, I, \mathsf{num}, y_0, \ldots, y_{p-1})$.

Looking again at the scenarios discussed earlier in light of the changes above, we have:

- Say M^i is signed using the ith instance of the scheme, using randomness r^i. Now a signature forgery is possible by finding r', M' such that $H(11, I^i, \mathsf{num}^i, r', M') = H(11, I^i, \mathsf{num}^i, r^i, M^i)$ for some i, where I^i, num^i denote the values used in the ith instance. For q evaluations of H, this occurs with probability $q \cdot 2^{-8n}$, suggesting that for k-bit security the output length of H can be only k bits.
- Alternately, say r, M, r', M' are found with $H(11, I^i, \mathsf{num}^i, r, M) = H(11, I^i, \mathsf{num}^i, r', M')$ for some i, and then a signature on M is obtained for the ith instance. A signature forgery is now possible if this instance used randomness r; this occurs only with probability $2^{-|r|}$, suggesting that the length of r can be only k bits.

In the appendix we provide a proof of security for the above scheme in the random-oracle model, showing that an attacker making q hash queries is able to forge a signature with probability at most $2q \cdot 2^{-8m}$, regardless of how many instances of the scheme are run. Note that this suggests setting $n = m$, since no security is gained for $n > m$.

References

[1] J. Buchmann, E. Dahmen, and M. Szydlo. Hash-based digital signature schemes. Technical Report, Technische Universitat Darmstadt, 2008.

[2] J. Katz and Y. Lindell. *Introduction to Modern Cryptography, 2nd edition.* Chapman & Hall/CRC Press, 2014.

[3] L. Lamport. Constructing digital signatures from a one-way function. Tehcnical Report SRI-CSL-98, SRI Intl. Computer Science Laboratory, 1979.

[4] F.T. Leighton and S. Micali. Large provably fast and secure digital signature schemes based on secure hash functions. US Patent 5,432,852, July 11, 1995.

[5] D. McGrew and M. Curcio. Hash-based signatures. Internet Draft draft-mcgrew-hash-sigs-02, July 4, 2014.

[6] R.C. Merkle. Secrecy, authentication, and public-key systems. PhD Thesis, Stanford University, 1979.

[7] R.C. Merkle. A certified digital signature. *Advances in Cryptology—Crypto '89*, LNCS vol. 435, pages 218–238, Springer-Verlag, 1989.

A Proof of Security

We prove security of the scheme when H is modeled as a random oracle. Let t be an upper bound on the number of instances of the scheme overall. This means we have some set $\{(I_i, \mathsf{num}_i)\}_{i=1}^t$ of identity/instance number pairs, where $(I_i, \mathsf{num}_i) \neq (I_j, \mathsf{num}_j)$ for $i \neq j$. We let $\mathsf{id}_i = (I_i, \mathsf{num}_i)$ denote the *identifier* for the ith instance. These identifiers could be chosen adaptively by the attacker (subject to being distinct) without any significant change to the proof below, but for simplicity we treat them as fixed in advance.

We are interested in bounding the attacker's success probability in the following experiment:

1. A random function $H : \{0,1\}^* \to \{0,1\}^{8n}$ is chosen.

2. For $i = 1$ to t, the key-generation algorithm is run using identifier id_i to obtain (pk^i, sk^i). The attacker is given $(\mathsf{id}_1, pk^1), \ldots, (\mathsf{id}_t, pk^t)$.

3. The attacker is given oracle access to H. In addition, it is given access to a signing oracle $\mathsf{Sign}(\cdot,\cdot)$ such that $\mathsf{Sign}(i, M)$ returns a signature on M computed using private key sk^i. For each i, the attacker may make at most one query $\mathsf{Sign}(i, \star)$.

   Without loss of generality we assume the attacker makes exactly one signing query $\mathsf{Sign}(i, M^i)$ for each value of i. We also assume that when the attacker is given a signature, it is additionally given the answers to all the H-queries needed to verify that signature.

4. The attacker outputs (i, M, σ) with $M \neq M^i$. The attacker succeeds if σ is a valid signature on the message M with respect to id_i, pk^i.

   Without loss of generality we assume that the attacker has previously made (or has been given the answers to) all the H-queries needed to verify σ on M with respect to id_i, pk^i.

An equivalent way of viewing the above experiment is as follows (we use $\|$ for string concatenation when using commas would cause confusion):

1. Initialize an empty set H. (H will contain defined query/answer pairs for the function H. That is, if $(x, y) \in H$ then $H(x) = y$.[3])

[3] The range of H is $\{0,1\}^{8n}$ but in some cases we only care about the output of H truncated to $8m$ bits. In such cases, we allow $(x, y) \in H$ with $y \in \{0,1\}^{8m}$.

2. Do the following for $i \in \{1, \ldots, t\}$:

    (a) For $j = 0, \ldots, p-1$, choose uniform $x_j^i \in \{0,1\}^{8m}$ and define $x_{j,0}^i := x_j^i$.

    (b) For $j = 0, \ldots, p-1$ and $k = 1, \ldots, e$, choose uniform $x_{j,k}^i \in \{0,1\}^{8m}$ and add $\left(00\|\mathsf{id}_i\|j\|k\|x_{j,k-1}^i,\ x_{j,k}^i\right)$ to H. Define $y_j^i := x_{j,e}^i$.

    (c) Choose uniform $pk^i \in \{0,1\}^{8n}$. Add $\left(01\|\mathsf{id}_i\|y_0^i\|\cdots\|y_{p-1}^i,\ pk^i\right)$ to H.

    (d) Choose uniform $r^i \in \{0,1\}^{8m}$ and $h^i \in \{0,1\}^{8n}$.

    (e) Give (id_i, pk^i) to the attacker.

3. When the attacker makes a query $H(x)$, answer it as follows:

    (a) If there is an entry $(x, y) \in H$ for some y, then return y.

    (b) Otherwise, if x begins with 00 then choose uniform $y \in \{0,1\}^{8m}$ and in any other case choose uniform $y \in \{0,1\}^{8n}$. Return y to the attacker, and store (x, y) in H.

4. When the attacker makes a query $\mathsf{Sign}(i, M^i)$, answer it as follows:

    (a) (Recall that h^i was defined in step 2(d).) If there is an entry $(11\|\mathsf{id}_i\|r^i\|M^i,\ h) \in H$ for some h, then redefine $h^i := h$. Otherwise, store $(11\|\mathsf{id}_i\|r^i\|M^i,\ h^i)$ in H.

    (b) Let $c^i := \mathsf{checksum}(h^i)$, and set $V^i := h^i\|c^i$. Parse V^i as a sequence of w-bit integers $V_0^i, \ldots, V_{p-1}^i$.

    (c) Return the signature $(r^i, x_{0,V_0^i}^i, \ldots, x_{p-1,V_{p-1}^i}^i)$.

5. The attacker outputs (i, M, σ) with $M \neq M^i$. The attacker succeeds if σ is a valid signature on the message M with respect to id_i.

We define the following events in the above experiment:

- $\mathsf{Coll}_{1,i}$ is the event that the attacker ever queries $H(01, \mathsf{id}_i, y_0, \ldots, y_{p-1})$ with $(y_0, \ldots, y_{p-1}) \neq (y_0^i, \ldots, y_{p-1}^i)$, and receives the response pk^i.

- $\mathsf{Coll}_{2,i}$ is the event that the attacker ever queries $H(11, \mathsf{id}_i, r^i, \star)$ before making the query $\mathsf{Sign}(i, \star)$.

- $\mathsf{Coll}^*_{2,i}$ is the event that either $\mathsf{Coll}_{2,i}$ occurs, or either of the following occur: (1) before making the query $\mathsf{Sign}(i, \star)$, the attacker ever queries $H(11, \mathsf{id}_i, \star, \star)$, and receives the response h^i, or (2) after making the query $\mathsf{Sign}(i, M^i)$, the attacker ever queries $H(11, \mathsf{id}_i, \star, M)$ with $M \neq M^i$, and receives the response h^i. Note that $\mathsf{Coll}_{2,i} \subseteq \mathsf{Coll}^*_{2,i}$.
- $\mathsf{Coll}_{3,i,j,k}$ is the event that the attacker queries $H(00, \mathsf{id}_i, j, k+1, x^i_{j,k})$ either before making the query $\mathsf{Sign}(i, \star)$, or after making the query $\mathsf{Sign}(i, \star)$ but with $k < V^i_j$.
- $\mathsf{Coll}^*_{i,j,k}$ is the event that either $\mathsf{Coll}_{i,j,k}$ occurs, or the attacker queries $H(00, \mathsf{id}_i, j, k+1, x)$ with $x \neq x^i_{j,k}$ and receives the response $x^i_{j,k+1}$. Note that $\mathsf{Coll}_{3,i,j,k} \subseteq \mathsf{Coll}^*_{3,i,j,k}$.

We first observe that the probability of forgery can be upper-bounded by the probablity that one of the above events occurs.

Claim 1. *If the attacker succeeds, then either* $\mathsf{Coll}_{1,i}$ *or* $\mathsf{Coll}^*_{2,i}$ *occur for some* i*, or else* $\mathsf{Coll}^*_{i,j,k}$ *occurs for some* i, j*, and* $0 \leq k < e$*.*

Proof. Say the attacker outputs (i, M, σ) with $M \neq M^i$ and σ a valid signature on M with respect to id_i, pk^i. Recall that, by assumption, all the H-queries needed to verify σ on M with respect to id_i, pk^i must be defined when the attacker outputs (i, M, σ). Parse σ as $(r, \sigma_0, \ldots, \sigma_{p-1})$, define $h = H(11, \mathsf{id}_i, r, M)$ and $c = \mathsf{checksum}(h)$, and let $V_0, \ldots, V_{p-1} = h \| c$ and $y_j = F^{e-V_j}_{00,id_i,j}(\sigma_j)$ be the values computed by running the verification algorithm with respect to id_i, pk^i on the message M and signature σ. Since the attacker succeeds we have $H(01, \mathsf{id}_i, y_0, \ldots, y_{p-1}) = pk^i$.

We show that if $\mathsf{Coll}_{1,i}$ and $\mathsf{Coll}^*_{2,i}$ have not occurred, then $\mathsf{Coll}^*_{i,j,k}$ must have occurred for some j, k. If $\mathsf{Coll}_{1,i}$ has not occurred, we must have $(y_0, \ldots, y_{p-1}) = (y^i_0, \ldots, y^i_{p-1})$. If $\mathsf{Coll}^*_{2,i}$ (and hence $\mathsf{Coll}_{2,i}$) has not occurred, the value of h^i was not changed during the experiment, and we must also have $h \neq h^i$. By construction of $\mathsf{checksum}$, we must therefore have $V_j < V^i_j$ for some j. But then one can verify by inspection that $\mathsf{Coll}^*_{3,i,j,k}$ must have occurred for some k. □

Thus, to bound the success probability of the attacker it suffices to bound the probabilities of the above events. Let q be an upper bound on the number of H-queries made by the attacker, and for an arbitrary string s let q_s denote the number of queries made by the attacker of the form $H(s, \star)$. If S is a set of strings none of which is a prefix of any other, then $\sum_{s \in S} q_s \leq q$.

Claim 2. *For all i,* $\Pr[\mathsf{Coll}_{1,i}] \leq q_{01\|\mathsf{id}_i} \cdot 2^{-8n}$.

Proof. Note that each time the attacker queries $H(01, \mathsf{id}_i, y_0, \ldots, y_{p-1})$ with $(y_0, \ldots, y_{p-1}) \neq (y_0^i, \ldots, y_{p-1}^i)$, the value returned is uniformly distributed in $\{0,1\}^{8n}$ and independent of pk^i. The claim follows. □

Claim 3. *For all i,* $\Pr[\mathsf{Coll}_{2,i}] \leq q_{11\|\mathsf{id}_i} \cdot 2^{-8m}$.

Proof. Note that r^i is a uniform $8m$-bit string, and the attacker has no information about r^i until it queries $\mathsf{Sign}(i, \star)$. The claim follows. □

Claim 4. *For all i,* $\Pr[\mathsf{Coll}^*_{2,i}] \leq q_{11\|\mathsf{id}_i} \cdot (2^{-8m} + 2^{-8n})$.

Proof. We have $\Pr[\mathsf{Coll}^*_{2,i}] \leq \Pr[\mathsf{Coll}_{2,i}] + \Pr[\mathsf{Coll}^*_{2,i} \mid \neg\mathsf{Coll}_{2,i}]$. The previous claim provides an upper bound on the first term. As for the second term, when $\mathsf{Coll}_{2,i}$ does not occur, the value of h^i does not change during the experiment. Each time the attacker queries $H(11, \mathsf{id}_i, \star, \star)$ before making the query $\mathsf{Sign}(i, \star)$, or queries $H(11, \mathsf{id}_i, \star, M)$ with $M \neq M^i$ after the query $\mathsf{Sign}(i, M^i)$, the value returned is uniformly distributed in $\{0,1\}^{8n}$ and independent of h^i. The claim follows. □

Claim 5. *For all i, j, k,*

$$\Pr\left[\mathsf{Coll}_{3,i,j,k} \mid \textstyle\bigwedge_{\ell=0}^{k-1} \neg\mathsf{Coll}^*_{3,i,j,\ell}\right] \leq q_{00\|\mathsf{id}_i\|j\|k+1} \cdot 2^{-8m}.$$

Proof. When $\mathsf{Coll}^*_{3,i,j,k-1}$ does not occur, the attacker gets no information about $x^i_{j,k}$ until it queries $H(00, \mathsf{id}_i, j, k+1, x^i_{j,k})$ or $\mathsf{Sign}(i, M^i)$ with $V^i_j \leq k$. In the latter case $\mathsf{Coll}_{3,i,j,k}$ cannot occur once the signature query is made. Since $x^i_{j,k}$ is uniform in $\{0,1\}^{8m}$, the claim follows. □

Claim 6. *For all i, j, k,*

$$\Pr\left[\mathsf{Coll}^*_{3,i,j,k} \mid \textstyle\bigwedge_{\ell=0}^{k-1} \neg\mathsf{Coll}^*_{3,i,j,\ell}\right] \leq 2 \cdot q_{00\|\mathsf{id}_i\|j\|k+1} \cdot 2^{-8m}.$$

Proof. We have

$$\begin{aligned}
&\Pr\left[\mathsf{Coll}^*_{3,i,j,k} \mid \textstyle\bigwedge_{\ell=0}^{k-1} \neg\mathsf{Coll}^*_{3,i,j,\ell}\right] \\
&\quad\leq \Pr\left[\mathsf{Coll}_{3,i,j,k} \mid \textstyle\bigwedge_{\ell=0}^{k-1} \neg\mathsf{Coll}^*_{3,i,j,\ell}\right] \\
&\qquad + \Pr\left[\mathsf{Coll}^*_{3,i,j,k} \mid \textstyle\bigwedge_{\ell=0}^{k-1} \neg\mathsf{Coll}^*_{3,i,j,\ell} \bigwedge \neg\mathsf{Coll}_{3,i,j,k}\right].
\end{aligned}$$

The previous claim provides an upper bound on the first term. As for the second term, note that when $\mathsf{Coll}_{3,i,j,k}$ does not occur then whenever the attacker queries $H(00, \mathsf{id}_i, j, k+1, \star)$, the value returned is uniformly distributed in $\{0,1\}^{8m}$ independent of $x^i_{j,k+1}$. The claim follows. □

Claim 7. *For all i, j,* $\Pr\left[\bigvee_{k=0}^{e-1} \mathsf{Coll}^*_{3,i,j,k}\right] \leq \sum_{k=0}^{e-1} 2 \cdot q_{00\|\mathsf{id}_i\|j\|k+1} \cdot 2^{-8m}$.

Proof. We have

$$\begin{aligned}\Pr\left[\bigvee_{k=0}^{e-1} \mathsf{Coll}^*_{3,i,j,k}\right] &\leq \sum_{k=0}^{e-1} \Pr\left[\mathsf{Coll}^*_{3,i,j,k} \mid \bigwedge_{\ell=0}^{k-1} \neg\mathsf{Coll}^*_{3,i,j,\ell}\right] \\ &\leq \sum_{k=0}^{e-1} 2 \cdot q_{00\|\mathsf{id}_i\|j\|k+1} \cdot 2^{-8m},\end{aligned}$$

using the previous claim. □

Putting everything together, we have:

Theorem 8. *For any adversary attacking arbitrarily many instances of the one-time signature scheme, and making at most q hash queries, the probability with which the adversary is able to forge a signature with respect to any of the instances is at most $2q/2^{8m}$.*

Proof. Let t denote the number of instances of the scheme. Using Claim 1 and a union bound, the probability with the the adversary forges a signature is at most

$$\sum_{i=1}^{t} \Pr[\mathsf{Coll}_{1,i}] + \sum_{i=1}^{t} \Pr[\mathsf{Coll}^*_{2,i}] + \sum_{i=1}^{t} \sum_{j=0}^{p-1} \Pr\left[\bigvee_{k=1}^{e-1} \mathsf{Coll}^*_{3,i,j,k}\right].$$

Using Claims 2, 4, and 7, the above is at most

$$\begin{aligned}&\sum_{i=1}^{t} q_{01\|\mathsf{id}_i} \cdot 2^{-8n} + \sum_{i=1}^{t} q_{11\|\mathsf{id}_i} \cdot 2^{-8m} \\ +\ & \sum_{i=1}^{t} \sum_{j=0}^{p-1} \sum_{k=0}^{e-1} 2 \cdot q_{00\|\mathsf{id}_i\|j\|k+1} \cdot 2^{-8m} \\ \leq\ & q_{01} \cdot 2^{-8n} + q_{11} \cdot 2^{-8m} + 2 \cdot q_{00} \cdot 2^{-8m} \\ \leq\ & 2 \cdot q \cdot 2^{-8m},\end{aligned}$$

as claimed. □