"Oops, I did it again" – Security of One-Time Signatures under Two-Message Attacks

Leon Groot Bruinderink and Andreas Hülsing

Department of Mathematics and Computer Science,
Technische Universiteit Eindhoven,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
authors-oops@huelsing.net

Abstract. One-time signatures (OTS) are called one-time, because the accompanying security reductions only guarantee security under single-message attacks. However, this does not imply that efficient attacks are possible under two-message attacks. Especially in the context of hash-based OTS (which are basic building blocks of recent standardization proposals) this leads to the question if accidental reuse of a one-time key pair leads to immediate loss of security or to graceful degradation.
In this work we analyze the security of the most prominent hash-based OTS, Lamport's scheme, its optimized variant, and WOTS, under different kinds of two-message attacks. Interestingly, it turns out that the schemes are still secure under two message attacks, asymptotically. However, this does not imply anything for typical parameters. Our results show that for Lamport's scheme, security only slowly degrades in the relevant attack scenarios and typical parameters are still somewhat secure, even in case of a two-message attack. As we move on to optimized Lamport and its generalization WOTS, security degrades faster and faster, and typical parameters do not provide any reasonable level of security under two-message attacks.

Keywords: Hash-based signatures, one-time signatures, few-time signatures, post-quantum cryptography, two-message attacks.

1 Introduction

The possible advent of large-scale quantum computers threatens the security of all widely deployed public key cryptography. Shor's algorithm [20] allows to factor and compute discrete logarithms in polynomial time on a quantum computer with a few thousand logical qubits. While it is not yet known for sure if it will be possible to build such a machine, it is a question of risk assessment to be prepared. The implied disastrous consequences by now also motivated standardization bodies (see e.g. [16]) and security agencies [17] to prepare the transition to post-quantum cryptography – cryptography secure against attacks using quantum-computers.

The first post-quantum signature schemes considered for standardization are hash-based Merkle Signature Schemes [13,9]. These schemes form the most confidence-inspiring post-quantum solution for digital signatures as their security only relies on some mild assumptions about properties of cryptographic hash-functions [11]. This is in contrast to all other proposals where security in addition to assumptions about the used hash function is based on rather new intractability assumptions like the $\mathcal{MQ}$-problem (see e.g. [18]) or the approximate shortest vector problem [6]. Hash-based signature schemes can

This work was supported by the Commission of the European Communities through the Horizon 2020 program under project number 645622 PQCRYPTO. Permanent ID of this document: 85629c7dc69dad1c4be4fbd7e360086c. Date: September 25, 2017

be split into stateful [15,5,4,3,10,11] and stateless [1] proposals. In this context, statefulness means that the secret key changes after every signature. In case a 'secret key state' is used twice, all security guarantees vanish. In practice it turns out that in many scenarios keeping a state becomes a complicated issue [14]. However, currently stateful schemes are the ones considered for standardization as these schemes are far more efficient in terms of signature size and signing speed than the stateless alternatives.

The reason these schemes are stateful is that their core building block is a so-called one-time signature scheme (OTS). A one-time signature scheme allows to use a key pair to sign a single (arbitrary) message. If a key pair is used to sign a second, different message, no security guarantees are given. The security reductions only apply as long as just a single message is signed. While this is commonly interpreted as the schemes are entirely broken if a key pair is used to sign twice, this is not necessarily the case. It is known that if an adversary has full control about the messages to be signed, the schemes are fully broken after two signatures, i.e. the secret key can be extracted without any effort. However, in practice the OTS causing statefulness are used to sign the digest of an adversarial chosen message. Moreover, in both recent proposals for standardization [13,9] these message digests are randomized. Hence, the actually signed message (digest) is unpredictable for an adversary.

Taking the message digest into account is one of the crucial steps in the construction of hash-based few-time signature schemes like HORS [19] that allow to use a key pair to sign a small number of messages before security drops below the acceptable limit. This opens up the question if classical hash-based OTS are still one-time when we take the message digest into account or if a similar argument applies as for HORS. For practice, this question translates to the question if reuse of a secret key state leads to a hard fail or if one is "only" facing graceful degradation of security.

Our contribution. In this work we analyze the security of hash-based one-time signature schemes under different kinds of two-message-attacks. We carry out the analysis for the most prominent proposals Lamport's scheme [12], the optimized version of Lamport's scheme [15], and the Winternitz OTS (WOTS) [15]. It turns out that actually, all three schemes are still secure under two-message attacks if we take into account that a message digest is signed – at least asymptotically (see Table 1).

The general working of these schemes is as follows. If necessary, a message M is first compressed using a cryptographic hash function H to obtain a fixed length message digest $M^* = \mathrm{H}(M)$. A mapping function G is used to map M^* to some index set $B = (B_1, \ldots, B_\ell) = \mathrm{G}(M^*)$. Finally, secret values indicated by the index set B are published as signature. Generally, the secret values are the preimages of public key values under a cryptographic hash function F. Verification works by applying F to the given values and comparing the results to the respective public key values. In case of WOTS secrets are arranged in hash chains. The end nodes of the chains are the public key values. In this case, there exists some dependency, i.e., if a value from a chain is part of the signature, all later values of that chain can be derived applying F.

After seeing two signatures, there exist two possible ways to forge a signature. First, an adversary can try to find a message that is mapped to an index set which is covered by the union of the index sets of the two seen signatures. In this case, all the required secret values are contained in the two signatures. Second, an adversary can try to compute the missing secret values for a signature from the respective public key values. However, this requires to break one of the security properties of F and would also allow to forge signatures after seeing just the public key. Parameters in practice are chosen such that this is infeasible. Consequently, we just consider the first approach in this work. The possibility and complexity of attacks of this type depends on the properties of hash function H, the message mapping function G, and possible dependencies of secret values (as in the case of WOTS). In our analysis we focus on the latter two. For H we

Table 1. Complexity for an existential forgery under a random message attack for the given signature scheme with typical parameters (see text).

Signature scheme	Attack Complexity
Lamport	$\mathcal{O}((1.34)^m)$
Optimized Lamport	$\mathcal{O}((1.14)^{m+\log m})$
Winternitz	$\mathcal{O}((1.09)^{m+\log m})$

assume that it behaves like a random oracle. This decision follows the same reasoning as above. Vulnerabilities of H would already allow for forgeries under one-message attacks. For WOTS this implies that the obtained results also apply to the recent variants of WOTS that minimize security assumptions [2,8,11] as the mapping function and the arrangement of secret values for these variants is the same as in the original scheme.

For Lamport's scheme, we obtain exact complexities for two-message attacks. For the optimized Lamport scheme and WOTS analysis becomes extremely complex when looking at the actual mapping function. This is caused by a checksum which is added to the message. This checksum introduces a lot of dependencies between probabilities, eventually leading to sums with an exponential number of summands. Therefore, we decided to analyze a simplified variant where we assume that the checksums are independent and uniformly distributed. For this simplified message mapping, we obtain exact complexities. We experimentally verified the results obtained for the simplified mapping function.

We analyze security of the OTS without initial message hash in terms of full break resistance, universal, selective, and existential unforgability under random and adaptively chosen message attacks. Please note that as we assume H to be a random oracle, existential unforgability under an adaptively chosen message attack (EU-CMA) of a scheme with initial randomized message hashing is equivalent to existential unforgability under a random message attack (EU-RMA) of the scheme without initial message hash. Accordingly, the crucial case for practice is EU-RMA security of the scheme without initial message hash. It covers the case of accidental reuse of an OTS key pair when using one of the recent proposals to standardize hash-based signatures. While all three schemes turn out to be EU-RMA-secure under two-message attacks in the asymptotic setting, we get different results for typical parameter choices. For Lamport's scheme with a message digest size of 256 bits, the complexity to produce existential forgeries under two-random-message attacks is still 2^{106} hash function calls, ignoring the costs for pairwise comparison of all message digests. Hence, in this setting a signer is still on the safe side even after using a one-time key pair twice. For the optimized Lamport OTS with 256 bit message digests, the complexity to produce existential forgeries under two-random-message attacks is already down to 2^{51}. Which means attacks are not for free, but they are possible. For WOTS in the same setting, using the parameters from [9], we are left with an attack complexity of 2^{34} hash function computations. This can be done on a modern computer within few days if not hours. These parameters use a Winternitz parameter of $w = 16$, i.e. hash chains of length 16. For bigger values of w, the attack complexity goes down even further. These results show that Lamport's scheme is still somewhat forgiving but especially for WOTS, measures have to be taken that prevent OTS key reuse in any case. However, as soon as we are considering attacks on quantum-computers, complexities drop at least by a square-root factor. In this case even Lamport's scheme has to be considered broken after two-random-message attacks for typical parameters.

Organization. In Section 2 we discuss the models we use as well as required notation. We start our analysis in Section 3 with Lamport's scheme. We continue in Section 4

with the optimized Lamport scheme and in Section 5 with WOTS. In Section 6, we experimentally verify our results.

Acknowledgement. This research was motivated in part by suggestions by Burt Kaliski of Verisign. The authors would also like to thank Aziz Mohaisen for helpful discussions.

2 The model

Security of one-time signature schemes (OTS) can be analyzed with regard to all traditional security definitions for general signature schemes. The difference is that the number of adversarial signature queries is limited to $q = 1$. Formally, any signature scheme that achieves EU-CMA-security (see definition below) when the adversary may only make a single signature query is a OTS. To understand the security of a OTS under two-message attacks in any of the models, we simply investigate the security for $q = 2$. We first discuss the traditional definitions and afterwards we discuss how to analyze security within these models.

2.1 Digital signature schemes

First, what exactly are we talking about? From a formal perspective the objects we are talking about are digital signature schemes, defined as follows:

Definition 1 (Digital Signature Scheme). *Let $\mathcal{M}$ be the message space. A digital signature scheme* $\text{DSS} = (\mathsf{kg}, \mathsf{sign}, \mathsf{vf})$ *is a triple of probabilistic polynomial time algorithms:*

- $\mathsf{kg}(1^n)$ *on input of a security parameter* 1^n *outputs a private signing key* sk *and a public verification key* pk*;*
- $\mathsf{sign}(\mathsf{sk}, M)$ *outputs a signature* σ *under* sk *for message* M*, if* $M \in \mathcal{M}$*;*
- $\mathsf{vf}(\mathsf{pk}, \sigma, M)$ *outputs 1 iff* σ *is a valid signature on* M *under* pk*;*

such that the following correctness condition is fulfilled:

$$\forall(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{kg}(1^n), \forall(M \in \mathcal{M}) : \mathsf{vf}(\mathsf{pk}, \mathsf{sign}(\mathsf{sk}, M), M) = 1.$$

Throughout this work *signature scheme* always refers to a digital signature scheme.

2.2 Security of signature schemes

The definition above is only a functional definition of the object at hand that says nothing about security. It leaves the question of how to define security for a signature scheme. In general we can split security notions into the goals an adversary $\mathcal{A}$ has to achieve (e.g., a valid signature on any new message for existential unforgeability) and the attack capabilities given to $\mathcal{A}$ (e.g., adaptively learning signatures on messages of its choice after seeing the public key). For the goals, the relevant notions[1] are:

Full break (FB): $\mathcal{A}$ can compute the secret key.
Universal forgery (UU): $\mathcal{A}$ can forge a signature for any given message. $\mathcal{A}$ can efficiently answer any signing query.
Selective forgery (SU): $\mathcal{A}$ can forge a signature for some message of its choice. In this case $\mathcal{A}$ commits itself to a message before the attack starts.

[1] We omit strong unforgeability here as it is irrelevant for this context

Existential forgery (EU): $\mathcal{A}$ can forge a signature for one arbitrary message. $\mathcal{A}$ might output a forgery for any message for which it did not learn the signature from a oracle during the attack.

On the other hand, for the attacks we got (We omit key-only attacks as these allow for no signature queries at all):

Random message attack (RMA): $\mathcal{A}$ learns the public key and the signatures on a set of random messages.

Adaptively chosen message attack (CMA): $\mathcal{A}$ learns the public key and is allowed to adaptively ask for the signatures on messages of its choice[2].

These two attacks are parameterized by the number of signature queries q the adversary is allowed to ask. For one-time schemes we only require that a notion is fulfilled for $q = 1$.

Any combination of a goal and an attack from the above sets gives a meaningful notion of security. The strength of the notion increases going down each list. Accordingly, a scheme that is only secure against a full break under a random message attack offers the weakest kind of security while a scheme that offers existential unforgeability under adaptively chosen message attacks offers the strongest security guarantees.

2.3 Formal definitions

We now give formal definitions for the notions from above. We define EU-CMA as an example. The definitions for the remaining notions can be found in Appendix A.

EU-CMA. The standard security notion for digital signature schemes is existential unforgeability under adaptive chosen message attacks (EU-CMA) which is defined using the following experiment. By $\text{Dss}(1^n)$ we denote a signature scheme with security parameter n.

Experiment $\mathsf{Exp}^{\text{EU-CMA}}_{\text{Dss}(1^n)}(\mathcal{A})$
$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{kg}(1^n)$
$(M^\star, \sigma^\star) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$
Let $\{(M_i, \sigma_i)\}_1^q$ be the query-answer pairs of $\mathsf{sign}(\mathsf{sk}, \cdot)$.
Return 1 iff $\mathsf{vf}(\mathsf{pk}, M^\star, \sigma^\star) = 1$ and $M^\star \notin \{M_i\}_1^q$.

For the success probability of an adversary $\mathcal{A}$ in the above experiment we write

$$\text{Succ}^{\text{EU-CMA}}_{\text{Dss}(1^n)}(\mathcal{A}) = \mathsf{Pr}\left[\mathsf{Exp}^{\text{EU-CMA}}_{\text{Dss}(1^n)}(\mathcal{A}) = 1\right].$$

A signature scheme is called $(t, \epsilon(t), q)$-EU-CMA-secure if any adversary running in time at most t, making no more than q queries to the signing oracle has at most a success probability of $\epsilon(t)$ for breaking the scheme:

Definition 2 (EU-CMA). *Let $n \in \mathbb{N}$, Dss a digital signature scheme as defined above. We call* Dss *$(t, \epsilon(t), q)$-EU-CMA-secure if* $\text{InSec}^{\text{EU-CMA}}(\text{Dss}(1^n); t, q)$*, the maximum success probability of all possibly probabilistic adversaries $\mathcal{A}$ running in time $\leq t$, making at most q queries to* Sign *in the above experiment, is bounded by $\epsilon(t)$:*

$$\text{InSec}^{\text{EU-CMA}}(\text{Dss}(1^n); t, q) \stackrel{def}{=} \max_{\mathcal{A}}\{\text{Succ}^{\text{EU-CMA}}_{\text{Dss}(1^n)}(\mathcal{A})\} \leq \epsilon(t).$$

[2] We omit the non-adaptive setting as it turns out that there is no difference in the given setting.

A $(t, \epsilon(t))$-EU-CMA-secure one-time signature scheme (OTS) is a DSS that is $(t, \epsilon(t), 1)$-EU-CMA secure, i.e. the number of signing oracle queries of the adversary is limited to one.

We can give similar definitions for the remaining notions. The difference between the different notions is described by a modified experiment. The definition of success probability and what it means for a scheme to fulfill the notion can be obtained replacing the experiment in the above definitions (and, of course, tracing the resulting changes through the definition). The experiments of the remaining notions are given in Appendix A.

Attack complexity. For a $(t, \epsilon(t))$-secure scheme, we define the attack complexity as $2t^*$ for $t^* = \min_t\{\epsilon(t) \geq \frac{1}{2}\}$. As the most costly operations of all attacks are calls to the message digest function H, we measure attack complexity as the number of calls to H.

Further model decisions. For our analysis we made several decisions on how we are analyzing the security in the above models. We are not interested in attacks that exploit weaknesses of the used hash-functions as these already apply in the one-message attack setting. Therefore, we model all used hash functions as random oracles. Due to this decision, RMA-attacks model the setting where randomized hashing is used for the initial message digest. Hence, we do not do a separate analysis for variants of the schemes that use randomized hashing.

3 Lamport's scheme

We start with analyzing Lamport's scheme which was the first proposal for a hash-based signature scheme. For $q = 1$ it achieves the strongest security notion EU-CMA-security when the used function is one-way (actually even the ignored stronger SU-CMA-security if the function is second-preimage resistant). This holds even without hashing the message first. Now let us look at the two-message attack case.

3.1 Scheme description

The first and most intuitive proposal for an OTS is Lamport's scheme (sometimes called Lamport-Diffie OTS) [12]. The scheme uses a one-way function $\mathrm{F} : \{0,1\}^n \to \{0,1\}^n$, and signs m bit strings. The secret key consists of $2m$ random bit strings

$$\mathsf{sk} = (\mathsf{sk}_{1,0}, \mathsf{sk}_{1,1}, \ldots, \mathsf{sk}_{m,0}, \mathsf{sk}_{m,1})$$

of length n. The public key consists of the $2m$ outputs of the one-way function

$$\mathsf{pk} = (\mathsf{pk}_{1,0}, \mathsf{pk}_{1,1}, \ldots, \mathsf{pk}_{m,0}, \mathsf{pk}_{m,1}) = (\mathrm{F}(\mathsf{sk}_{1,0}), \mathrm{F}(\mathsf{sk}_{1,1}), \ldots, \mathrm{F}(\mathsf{sk}_{m,0}), \mathrm{F}(\mathsf{sk}_{m,1}))$$

when evaluated on the elements of the secret key. Signing a message (digest) $M^* \in \{0,1\}^m$ corresponds to publishing the corresponding elements of the secret key:

$$\sigma = (\sigma_1, \ldots, \sigma_m) = (\mathsf{sk}_{1,M_1^*}, \ldots, \mathsf{sk}_{m,M_m^*}).$$

To verify a signature the verifier checks whether the elements of the signature are mapped to the right elements of the public key using F:

$$(\mathrm{F}(\sigma_1), \ldots, \mathrm{F}(\sigma_m)) \stackrel{?}{=} (\mathsf{pk}_{1,M_1^*}, \ldots, \mathsf{pk}_{m,M_m^*}).$$

For Lamport's scheme, the message mapping can be considered the identity.

Table 2. Overview of the computational complexity for two-message attacks against Lamport's scheme. If the success probability of an attack is not constant in terms complexity, we give the attack complexity to achieve a success probability of 1/2.

Security Goal	Attack Complexity	Pr[Success]
EU-CMA	$\mathcal{O}((4/3)^{m/3})$	$\frac{1}{2}$
SU-CMA	$\mathcal{O}((4/3)^{m/3})$	$\frac{1}{2}$
UU-CMA	$\mathcal{O}(2^{m/2})$	$\frac{1}{2}$
FB-CMA	$\mathcal{O}(2^{m/2})$	$\frac{1}{2}$
EU-RMA	$\mathcal{O}((4/3)^{m})$	$\frac{1}{2}$
SU-RMA	-	$(3/4)^m$
UU-RMA	-	$(3/4)^m$
FB-RMA	-	$(1/2)^{m/2}$

3.2 Security under two-message attacks

Considering a CMA setting, we cannot achieve any security without an initial message hash. An adversary $\mathcal{A}$ can choose any pair of messages (M_1^*, M_2^*) such that $M_1^* = \neg M_2^*$, where $\neg$ denotes bitwise negation, and will learn the full secret key. In the following we assume a message M is first hashed using a hash function $\mathrm{H} : \{0,1\}^* \to \{0,1\}^m$, i.e., a m-bit message digest M^* is used to select the secret key elements. Our results are summarized in Table 2.

FB-CMA. A full break requires $\mathcal{A}$ to find a pair of messages (M_1, M_2) such that $\mathrm{H}(M_1) = \neg\mathrm{H}(M_2)$. This task has the same complexity as collision finding for H. The only difference between the two tasks is that the equality condition is replaced by equality after negation. Sadly, this does not mean that we get a reduction from collision resistance as the counter example of the identity function shows: The identity function is collision resistant as no collisions exist but it is trivial to find a pair such that one message is the negation of the other. However, assuming H behaves like a random function a birthday bound argument shows that the complexity of finding such a pair is $\mathcal{O}(2^{m/2})$ which can be carried out as pre-computation as long as H is known.

EU-CMA. To produce a valid forgery in a chosen message setting, an adversary $\mathcal{A}$ has to find a triple of messages M_1, M_2, M_3 such that

$$\mathrm{break}(M_1, M_2, M_3) = (\forall i \in [0, m-1]) : \mathrm{H}(M_1)_i = \mathrm{H}(M_2)_i \vee \mathrm{H}(M_1)_i = \mathrm{H}(M_3)_i)$$

where $\mathrm{H}(\cdot)_i$ denotes the i-th bit of the message digest. In this case, we say that M_2, M_3 form a cover for M_1.

For random messages M_1, M_2, M_3, the probability that M_2, M_3 cover M_1 is the inverse probability of each bit of M_1^* not being covered by M_2^*, M_3^*:

$$\mathsf{Pr}_{M_1}[\mathrm{break}(M_1, M_2, M_3) = 1] = (1 - (1/2)^2)^m = (3/4)^m$$

For an existential forgery, $\mathcal{A}$ can start by hashing $\tau > 2$ random messages, pick a random set of two hashed message and check if these cover a hashed third message. There are $\binom{\tau}{2}$ such pairs of hashed messages, and $\tau - 2$ hashed messages that are potentially covered, leaving a total of $\binom{\tau}{2}(\tau - 2)$ possibilities. We can bound the success probability of an existential forgery by the union bound:

$$\mathsf{Pr}_{\{M_1,\ldots,M_\tau\}}[\exists (M_a, M_b, M_c) \in \{M_1, \ldots, M_\tau\} : \text{break}(M_a, M_b, M_c) = 1] \leq \binom{\tau}{2}(\tau - 2)(3/4)^m \leq \frac{1}{2}\tau^3(3/4)^m$$

We want to know for which τ this upper bound reaches $1/2$, which is $\tau = (4/3)^{m/3}$. Hence, the attack complexity is lower bound by $(4/3)^{m/3}$. As an example, if we consider $m = 256$ then $2^{36} > (4/3)^{m/3}$. It has to be noted that this is all pre-computation, which can be done before choosing a victim: no knowledge of the public key is required. It remains to show how tight our upper bound is. In Section 6, we experimentally verify that it is tight for the case of optimized Lamport and Winternitz.

SU-CMA. For selective forgeries, $\mathcal{A}$ can pick a message M for which it needs to find a cover before receiving signatures. However, since no knowledge of the public key is needed to start an attack, there is no difference between a selective forgery and an existential forgery. $\mathcal{A}$ can simply search for three messages (M_1, M_2, M_3) satisfying the break condition before the attack starts using the correct hash function. It can then commit to M_1 before learning pk, and use the signatures of M_2, M_3 to sign M_1. This means, the complexity of a selective forgery can again be lower bound by $(4/3)^{m/3}$.

UU-CMA. For universal forgeries, $\mathcal{A}$ can try to find two messages M_1, M_2 such that they have non-overlapping message digests in r indices. After the experiment, $\mathcal{A}$ can forge any message with probability $(1/2)^{m-r}$, since a messages digest has to overlap with the digests of M_1, M_2 in $m - r$ indices. The probability that any two messages M_1, M_2 have non-overlapping message digests in r indices is $\binom{m}{r}(1/2)^r(1/2)^{m-r} = \binom{m}{r}(1/2)^m$. Using similar arguments as in the EU-CMA case after τ calls to H, the probability that two messages have r non-overlapping indices is bounded by at least $1/2$ if $\binom{\tau}{2} \geq 1/2 \cdot 2^m \binom{m}{r}^{-1}$, where we can estimate that $\tau = 2^{m/2}\binom{m}{r}^{-1/2}$. It is easy to see that the more pre-computation an attacker is doing, the higher the success probability. Figure 1 shows the success probability as a function of the pre-computation carried out. For $m = 256$, a pre-computation of 2^{136} calls to H is required to reach a probability of $1/2$.

EU-RMA. In this case, the adversary gets a signature of two random messages (M_1, M_2) and has to find a third message M_3 that is covered by M_1, M_2. The difference to the CMA case is that $\mathcal{A}$ cannot optimize the choice of M_1, M_2. This means each index should be covered, which happens with probability $(3/4)^m$. In consequence, $\mathcal{A}$ has to compute $\tau = (4/3)^m$ message digests before it finds a forgery with probability $\geq 1/2$. For $m = 256$, this means the attacker has to compute about 2^{106} message digests, making this type of forgery computationally infeasible. However, for $m = 128$ bit message digests, this would mean a computational cost of 2^{53}, which is in reach for strong attackers.

SU-RMA. For SU-RMA, the adversary selects a message before it receives two signatures of two random messages. There is no way for $\mathcal{A}$ to optimize the selection of this message, as $\mathcal{A}$ does not know (or has influence on) the two random messages for which it learns the signatures. The probability that $\mathcal{A}$ can afterwards sign the selected message is $(3/4)^m$. This is also the success probability of the attack. Note that this probability is constant for fixed parameters, i.e., independent of the adversaries efforts.

UU-RMA. For random message attacks, there is no difference between universal and selective forgery attacks since the adversary has no power over the signed messages and cannot affect his success probability by choice of a target message. This means also in this case, the probability of a forgery is $(3/4)^m$.

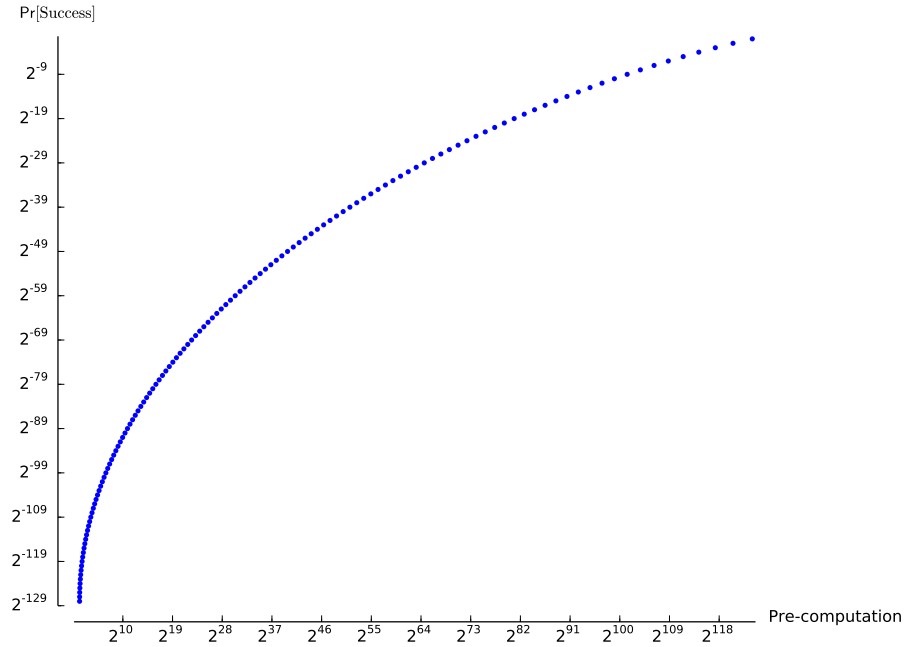

Fig. 1. This plot shows the relation between the amount of pre-computation and the success probability of a universal forgery in a chosen message attack on Lamport's One-Time Signature Scheme.

FB-RMA. The probability of a full break under a random message attack, is simply the probability that two messages are each-others negated version. This happens with probability $(1/2)^m$.

4 Optimized Lamport

The optimized Lamport scheme is very similar to Lamport's scheme and first appeared in [15]. While it is interesting on its own, it is also of interest as it can be viewed as a special, simplified version of the Winternitz OTS discussed in the next section.

4.1 Scheme description

The optimized Lamport scheme uses a one-way function $\mathrm{F} : \{0,1\}^n \to \{0,1\}^n$, and signs m bit messages. The secret key consists of $\ell = m + \log m + 1$ random bit strings

$$\mathsf{sk} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell)$$

of length n. The public key consists of the ℓ outputs of the one-way function

$$\mathsf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell) = (\mathrm{F}(\mathsf{sk}_1), \ldots, \mathrm{F}(\mathsf{sk}_\ell))$$

when evaluated on the elements of the secret key. Signing a message $M^* \in \{0,1\}^m$ corresponds to first computing and appending a checksum to M^* to obtain the message mapping $\mathrm{G}(M^*) = B = M^* \| C$ where $C = \sum_{i=1}^{m} \neg M_i^*$. The signature consists of the secret key element if the corresponding bit in B is 1, and the public key element otherwise:

$$\sigma = (\sigma_1, \ldots, \sigma_m) \text{ with } \sigma_i = \begin{cases} \mathsf{sk}_i & \text{, if } B_i = 1, \\ \mathsf{pk}_i & \text{, if } B_i = 0. \end{cases}$$

Table 3. Overview of the computational complexity for two-message attacks against the optimized Lamport scheme. If the success probability of an attack is not constant in terms of complexity, we give the attack complexity to achieve a success probability of 1/2 (aside from SU-RMA as the best we can achieve is a success probability of $\frac{3}{8}$).

Security Goal	Attack Complexity	Pr[Success]
EU-CMA	$\mathcal{O}((8/7)^{(m+\log m)/3})$	$\frac{1}{2}$
SU-CMA	$\mathcal{O}((8/7)^{(m+\log m)/3})$	$\frac{1}{2}$
UU-CMA	$\mathcal{O}((4/3)^{(m+\log m)/2})$	$\frac{1}{2}$
FB-CMA	$\mathcal{O}((4/3)^{(m+\log m)/2})$	$\frac{1}{2}$
EU-RMA	$\mathcal{O}((8/7)^{m+\log m})$	$\frac{1}{2}$
SU-RMA	$\mathcal{O}(2^{m+\log m})$	$\frac{3}{8}$
UU-RMA	-	$(7/8)^{m+\log m}$
FB-RMA	-	$(3/4)^{m+\log m}$

To verify a signature the verifier checks whether the full public key is obtained by hashing the elements of the signature that correspond to 1 bits in B:

$$\text{Return 1, iff } (\forall i \in [1,\ell]) : \mathsf{pk}_i = \begin{cases} \mathrm{F}(\sigma_i) & \text{, if } B_i = 1, \\ \sigma_i & \text{, if } B_i = 0. \end{cases}$$

4.2 Security under two-message attacks

As with the non-optimized Lamport scheme, we cannot achieve any security without initial message hash. While it is impossible to learn the whole secret key from a two-message attack for typical parameters (this is the case as for m being a power of two the most significant bit of the checksum is only 1 for the all zero message, and it is impossible to learn the remaining secret key values from the signature of a single message), it is trivial to obtain all secret key elements but the one that corresponds to the most significant bit of the checksum. This allows to sign any message but the all 0 message. An adversary can for example use the all 1 message (to learn the secret key values for the message part of B) and any message with a single one (to learn the secret key values of the checksum part of B, besides the one at the most significant position).

In the following we assume a message M is first hashed using a hash function $\mathrm{H} : \{0,1\}^* \to \{0,1\}^m$ to obtain a message digest M^* – making attacks significantly harder. It is easy to see that checksum C follows a binomial distribution. However, the analysis of the scheme as described above turned out too complex to be carried out exactly due to the dependency between C and M^*. The problem is that it would be possible to condition on two checksums to cover a third one in the existential forgery setting. These conditions would give an exact Hamming weight for the message parts. However, there would be exponentially many possibilities, each with a specific probability, rendering a very complex analysis. For that reason, we simplified the analysis assuming that C is uniformly random and thereby that digest M^* and checksum C are independent of each other. Note that the neglected dependency, and the neglected distribution of C, can make the attack both easier and harder, depending on wether the higher order bits of C are covered. Our theoretical results are summarized in Table 3. For an experimental verification of our results see Section 6.

FB-CMA. As mentioned above for m being a power of two (which is the typical setting), it is impossible to learn the whole secret key from a two-message attack. For other choices of m, an adversary $\mathcal{A}$ has to find two messages M_1, M_2 such that $(B_1)_i = 1$ or $(B_2)_i = 1$ for all $i \in \{0, \ldots, \ell - 1\}$.

As H is modeled as random oracle and we assume the checksum is uniformly random and independent of the message, every random input message M leads to a random message mapping B of length ℓ. For two random input messages M_1, M_2, the probability that at least one of the two corresponding message mappings B_1, B_2 is 1 at each position is:

$$\mathsf{Pr}[\mathrm{FB}(M_1, M_2)] = (3/4)^\ell.$$

Similar to the strategy of the existential forgery in Lamport's scheme, we can hash τ messages and check all pairs for a full break. The probability of a full break is bounded by $\binom{\tau}{2}(3/4)^\ell$. We can therefor lower bound the attack complexity of a full break by $(4/3)^{\ell/2}$ calls to H. For $m = 256$, this complexity equals 2^{54}.

EU-CMA. We will now explore forgeries for a third message, given the signatures for two messages. We define the condition for a break for three messages M_1, M_2, M_3 with message mappings B_1, B_2, B_3 as:

$$\mathrm{break}(M_1, M_2, M_3) := (\forall i \in [0, \ell - 1]) : (B_1)_i = 1 \Rightarrow (B_2)_i = 1 \vee (B_3)_i = 1 \quad (1)$$

where $(B_j)_i$ denotes the i-th bit of the mapping of message M_j. If the condition is fulfilled, we say that M_2, M_3 form a cover of M_1.

In other words: we only need the secret values for those bits of the first message mapping that are 1, so the probability for a break is higher for target messages with a low weight message mapping. Recall that we assume that M_j^* and C_j are independent, meaning we assume we have three independent random bit strings.

To get the probability that we cover a bit of B_1, we can condition on the value of that bit $b \in \{0, 1\}$:

$$\begin{aligned}
&\mathsf{Pr}[(B_1)_i \leq (B_2)_i \vee (B_1)_i \leq (B_3)_i] \\
&= \sum_{b \in \{0,1\}} \mathsf{Pr}[(B_1)_i \leq (B_2)_i \vee (B_1)_i \leq (B_3)_i \mid (B_1)_i = b] \mathsf{Pr}[(B_1)_i = b] \\
&= \frac{1}{2} \cdot \mathsf{Pr}[0 \leq (B_2)_i \vee 0 \leq (B_3)_i \mid (B_1)_i = 0] \\
&\quad + \frac{1}{2} \cdot \mathsf{Pr}[1 \leq (B_2)_i \vee 1 \leq (B_3)_i \mid (B_1)_i = 1] \\
&= \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{3}{4} = \frac{7}{8}
\end{aligned}$$

This means that the probability that the break condition is fulfilled for three random messages is $\left(\frac{7}{8}\right)^\ell$.

As with the original Lamport scheme, we can precompute τ message mappings, and calculate the upper bound for the success probability. This time, for the bound to reach $1/2$ we need to compute $\tau = (8/7)^{\ell/3}$ message mappings, using similar arguments as in the EU-CMA case for Lamport. For $m = 256$, this means the adversary needs to precompute $\tau = 2^{17}$ hash digests. For $m = 128$, this would mean $\tau = 2^9$ hash digests.

SU-CMA. As with the original Lamport scheme, the adversary does not need knowledge of the public key to compute three messages that satisfy the break condition. This means that also for the optimized Lamport scheme, a selective forgery has the same complexity as an existential forgery under chosen message attacks.

UU-CMA. The goal of the adversary is to find two messages M_1, M_2 such that their combined mappings have the highest weight possible. The probability that any two messages have weight r is equal to $\binom{\ell}{r}(3/4)^r(1/4)^{\ell-r}$, where we again assume that M^* and C are independent. Note that the mean of this distribution is at $\ell \cdot (3/4)$, which means $\mathcal{A}$ should not take any r below $\ell \cdot (3/4)$. After τ calls to H, the probability that

two of the messages M_1, M_2 have a combined weight of r is bounded by at least $1/2$ if $\binom{\tau}{2} \geq 1/2 \cdot \left(\binom{\ell}{r}(3/4)^r(1/4)^{\ell-r}\right)^{-1}$. We can estimate the pre-computation complexity as square-root of the right part of this inequality. After the online phase of the attack, $\mathcal{A}$ can sign a new message with probability $(1/2)^{\ell-r}$, since for the positions that are not covered by B_1 or B_2, the bit of the new message must be 0. The relation between the pre-computation and the success probability is given in Figure 2 for $m = 256$.

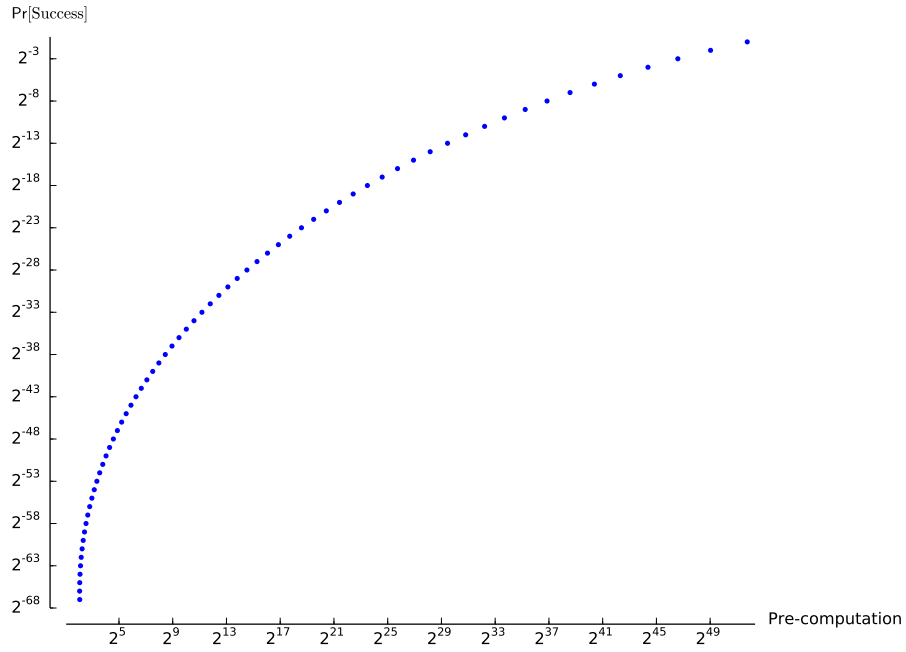

Fig. 2. This plot shows the relation between the amount of pre-computation and the success probability of a universal forgery in a chosen message attack on the optimized Lamport scheme.

EU-RMA. According to Eqn. 1, two messages M_2, M_3 have a probability of $(7/8)^\ell$ to cover a random third message M_1. This means that after receiving the signature of two random messages, the adversary has to search $\tau = (8/7)^\ell$ messages to forge a third signature (again using arguments described in earlier analyses), since it only needs the secret values for the bits of M_1 that are 1. For $m = 256$, this means a computational cost of about 2^{51}, which is in reach for a strong attacker. For $m = 128$, this would mean a computational cost of 2^{26}, which can be done within minutes on today's CPUs.

SU-RMA. Unlike with the original Lamport scheme, for the optimized Lamport scheme an adversary can optimize his selection of the target message in a random message attack. Messages that have low-weight message mappings are more likely to be covered by the mappings of two random messages. However, note that we can only select a single target message instead of a whole cover, which makes the pre-computation more costly. The probability to find a message mapping B with weight r is equal to $\binom{\ell}{r}(1/2)^\ell$, which is again symmetric around $\ell/2$. An attacker should therefor always pick a message with weight $r \leq \ell/2$. This message can be signed, after receiving the signatures of two random messages, with probability $(3/4)^r$, since all positions of B that are 1 have to be covered by the mappings of the two random messages. If we again estimate the

pre-computation as $\tau = \left(\binom{\ell}{r}(1/2)^{\ell}\right)^{-1}$ to find a message mapping with weight r with probability bounded by $1/2$, we get the relation between pre-computation and success probability for a selective forgery in Figure 3 for $m = 256$. Note that this figure looks similar to Figure 2 but a far more pre-computation is required to achieve the same bound on the success probability. Even for strong attackers, it should be infeasible to get a high success probability.

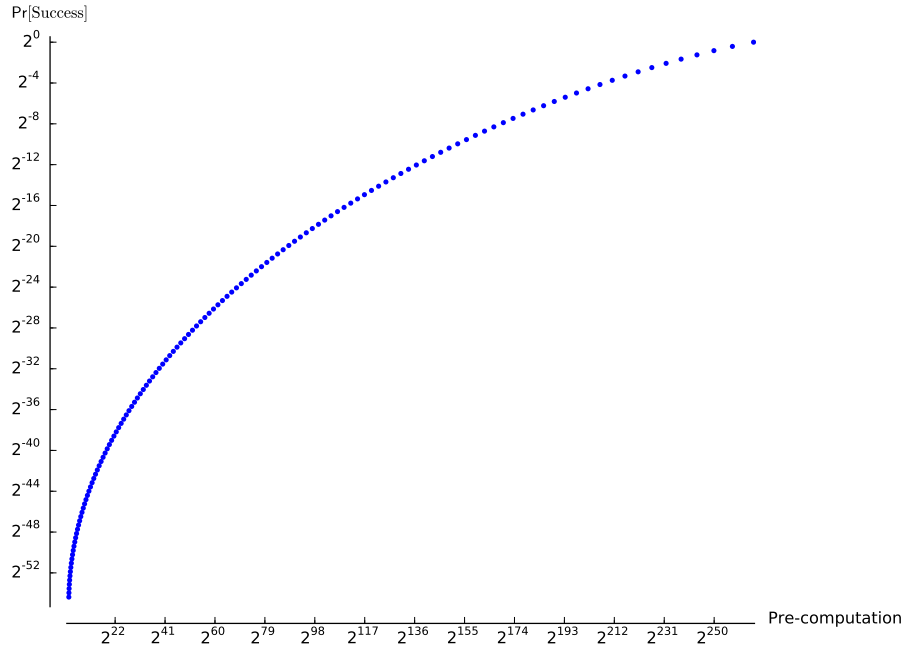

Fig. 3. This plot shows the relation between the amount of pre-computation and the succes probability of a selective forgery in a chosen message attack on the optimized Lamport's One-Time Signature Scheme.

UU-RMA. For a universal forgery under a random message attack, the attacker cannot influence anything in the experiment. This means the success probability for this forgery is simply the success probability of the conditional break: $(7/8)^{\ell}$.

FB-RMA. The probability of a full break under a random message attack, is simply the probability that all bits are covered. This happens with probability $(3/4)^{\ell}$, which is 2^{-54} when $m = 256$.

5 Winternitz OTS

The Winternitz one-time signature scheme (WOTS) is a further improvement of the optimized Lamport scheme. Instead of using the hash of each secret key value as public key, the public key values are obtained by hashing more than once, i.e. w times. That way, more than one bit can be encoded per selection of a hash value. The basic idea for the Winternitz OTS (WOTS) was proposed in [15]. What we know as WOTS today is a generalization that was proposed by Even, Goldreich, and Micali [7]. There exist several variants that reduce the assumptions made about the used hash function [2,8,11]. Recent

standardization proposals for hash-based signatures [13,9] as well as a recent proposal for stateless hash-based signatures [1] use WOTS as one-time signature scheme.

5.1 Scheme description

WOTS uses a length-preserving (cryptographic hash) function $\mathrm{F} : \{0,1\}^n \to \{0,1\}^n$. It is parameterized by the message length m and the Winternitz parameter $w \in \mathbb{N}, w > 1$, which determines the time-memory trade-off. The two parameters are used to compute

$$\ell_1 = \left\lceil \frac{m}{\log(w)} \right\rceil, \quad \ell_2 = \left\lfloor \frac{\log(\ell_1(w-1))}{\log(w)} \right\rfloor + 1, \quad \ell = \ell_1 + \ell_2.$$

The scheme uses $w-1$ iterations of F on a random input. We define them as

$$\mathrm{F}^a(x) = \mathrm{F}(\mathrm{F}^{a-1}(x))$$

and $\mathrm{F}^0(x) = x$.

Now we describe the three algorithms of the scheme:

Key generation algorithm ($\mathsf{kg}(1^n)$)*:* On input of security parameter 1^n the key generation algorithm choses ℓ n-bit strings uniformly at random. The secret key $\mathsf{sk} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell)$ consists of these ℓ random bit strings. The public verification key pk is computed as

$$\mathsf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell) = (\mathrm{F}^{w-1}(\mathsf{sk}_1), \ldots, \mathrm{F}^{w-1}(\mathsf{sk}_\ell))$$

Signature algorithm ($\mathsf{sign}(1^n, M^*, \mathsf{sk})$)*:* On input of security parameter 1^n, a message (digest) M^* of length m and the secret signing key sk, the signature algorithm first computes a base w representation of M^*: $M^* = (M_1^* \ldots M_{\ell_1}^*)$, $M_i^* \in \{0, \ldots, w-1\}$. Next it computes the check sum

$$C = \sum_{i=1}^{\ell_1} (w - 1 - M_i^*)$$

and computes its base w representation $C = (C_1, \ldots, C_{\ell_2})$. The length of the base-w representation of C is at most ℓ_2 since $C \leq \ell_1(w-1)$. We set $B = (B_1, \ldots, B_\ell) = M^* \parallel C$. The signature is computed as

$$\sigma = (\sigma_1, \ldots, \sigma_\ell) = (\mathrm{F}^{B_1}(\mathsf{sk}_1), \ldots, \mathrm{F}^{B_\ell}(\mathsf{sk}_\ell)).$$

Verification algorithm ($\mathsf{vf}(1^n, M^*, \sigma, \mathsf{pk})$)*:* On input of security parameter 1^n, a message (digest) M^* of length m, a signature σ and the public verification key pk, the verification algorithm first computes the B_i, $1 \leq i \leq \ell$ as described above. Then it does the following comparison:

$$\mathsf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell) \stackrel{?}{=} (\mathrm{F}^{w-1-B_1}(\sigma_1), \ldots, \mathrm{F}^{w-1-B_\ell}(\sigma_\ell))$$

If the comparison holds, it returns **true** and **false** otherwise.

Remark 1. The difference between the basic WOTS as described above and the variants proposed in [2,8,11] is how F is iterated. As all the attacks below are independent of this choice, our results apply to all those variants, too.

5.2 Two-message attacks

Without hashing the message, the scheme does not offer any security once an attacker can choose two messages to be signed. As always, the adversary simply chooses the all zero and the all one message to be signed, and afterwards knows all secret values (for some parameter choices it will actually be impossible to extract the whole secret key for the same reason as for optimized Lamport. However, in that case, as for the optimized Lamport scheme, it is possible to select two messages that allow learn all but one secret key element).

In the following we assume a message M is first hashed using a hash function $\mathrm{H} : \{0,1\}^* \to \{0,1\}^m$ to obtain a message digest M^* – making attacks significantly harder. As for the optimized Lamport scheme, the analysis of the scheme as described above turned out too complex to be carried out exactly due to the dependency between C and M^*. We simplified the analysis assuming that C is uniformly random and thereby that digest M^* and checksum C are independent of each other. It applies again that the neglected dependency can make the attack both easier and harder, depending on the setting. Our theoretical results are summarized in Table 4. For an experimental verification of the results see Section 6.

Table 4. Overview of the computational complexity for two-message attacks against the Winternitz OTS. If the success probability of an attack is not constant in terms of complexity, we give the attack complexity to achieve a success probability of 1/2.

Security Goal	Attack Complexity	Pr[Success]
EU-CMA	$\mathcal{O}(\left(\frac{(w+1)(4w+1)}{6w^2}\right)^{-\frac{m+\log m}{3\log w}})$	$\frac{1}{2}$
SU-CMA	$\mathcal{O}(\left(\frac{(w+1)(4w+1)}{6w^2}\right)^{-\frac{m+\log m}{3\log w}})$	$\frac{1}{2}$
UU-CMA	$\mathcal{O}(\left(1-(\frac{w-1}{w})^2\right)^{-\frac{m+\log m}{2\log w}})$	$\frac{1}{2}$
FB-CMA	$\mathcal{O}(\left(1-(\frac{w-1}{w})^2\right)^{-\frac{m+\log m}{\log w}})$	$\frac{1}{2}$
EU-RMA	$\mathcal{O}(\left(\frac{(w+1)(4w+1)}{6w^2}\right)^{-\frac{m+\log m}{\log w}})$	$\frac{1}{2}$
SU-RMA	$\mathcal{O}(\left(\frac{1}{w}\right)^{-\frac{m+\log m}{\log w}})$	$\frac{1}{2}$
UU-RMA	-	$\left(\frac{(w+1)(4w+1)}{6w^2}\right)^{\frac{m+\log m}{\log w}}$
FB-RMA	-	$\left(1-(\frac{w-1}{w})^2\right)^{\frac{m+\log m}{\log w}}$

FB-CMA. The adversary has to find messages M_1, M_2 with mappings B_1, B_2 such that for all $0 \leq i \leq \ell$: either $(B_1)_i = 0$ or $(B_2)_i = 0$. The probability to cover an index of the secret key equals $(1-(\frac{w-1}{w})^2)$ for each i, which means the probability that this is true for all i equals: $(1-(\frac{w-1}{w})^2)^\ell$. After hashing τ messages, the probability to find two messages satisfying the condition of a full break will be upper bounded by at least $1/2$ if $\binom{\tau}{2} \geq 1/2 \cdot (1-(\frac{w-1}{w})^2)^{-\ell}$, which means we can lower bound the attack complexity by $\tau \geq (1-(\frac{w-1}{w})^2)^{-\ell/2}$. As a sanity check, we see that for $w = 2$ we get $\tau = (4/3)^{\ell/2}$, which is the complexity of a full break for the optimized Lamport scheme. Typical parameters for applications are $w = 16$ and $m = 256$, which leads to $\ell = 67$ and $\tau = 2^{102}$.

EU-CMA. For an existential forgery, we first define the condition for a break for WOTS:

$$\mathrm{break}(M_1, M_2, M_3) := (\forall i \in [0, \ell-1]) : (B_1)_i \geq (B_2)_i \vee (B_1)_i \geq (B_3)_i \tag{2}$$

where $(B_j)_i$ denotes the i-th bit of the base-w values of the message mapping B_j for message $M_j; j \in \{1, 2, 3\}$. If the condition is true, we say M_2, M_3 form a cover of M_1.

We will first see what the probability is to cover one index of B_1. If we condition on the value of $(B_1)_i$, we get:

$$\begin{aligned}
&\mathsf{Pr}[(B_1)_i \geq (B_2)_i \vee (B_1)_i \geq (B_3)_i] = \\
&\sum_{x=0}^{w-1} \mathsf{Pr}[(B_1)_i \geq (B_2)_i \vee (B_1)_i \geq (B_3)_i | (B_1)_i = x] \mathsf{Pr}[(B_1)_i = x] = \\
&\sum_{x=0}^{w-1} \frac{1}{w} \left(1 - \left(\frac{w-(x+1)}{w}\right)^2\right) = \\
&\frac{(w+1)(4w-1)}{6w^2}
\end{aligned}$$

Again as a sanity check, we see that for $w = 2$, this probability equals $(7/8)$, which we already concluded for the optimized Lamport scheme.

In total we see that the probability for a conditional break is:

$$\begin{aligned}
\Pr[\text{break}(M_1, M_2, M_3) = 1] &= \left(\frac{(w+1)(4w-1)}{6w^2}\right)^{\ell} \\
&\approx \left(\frac{(w+1)(4w-1)}{6w^2}\right)^{\frac{m+\log m}{\log w}}
\end{aligned}$$

We see that for bigger w, the probability that one of the indices is not covered grows, but the number of indices shrinks. The logarithmic decrease of the exponent is in this case more important, which means the bigger the w, the bigger the probability of the conditional break (which means less computational power required for forgeries).

Similar to the arguments for the EU-CMA cases for Lamport and optimized Lamport scheme, an adversary needs to pre-compute about $\tau = \left(\left(\frac{(w+1)(4w-1)}{6w^2}\right)^{-\frac{m+\log m}{\log w}}\right)^{1/3}$ message mappings for the bound on the probability to find a cover in the list of τ message mappings to reach $1/2$. As an example, if we set $m = 256$ and $w = 16$, we have $\tau = 2^{12}$. Note that, unlike the FB-CMA setting, it is much easier to forge a third signature for bigger w: while it becomes harder to get $B_i = 0$, the probability for a message cover grows.

SU-CMA. As with Lamport's scheme and the optimized Lamport scheme, $\mathcal{A}$ does not need knowledge of the public key to start any pre-computation. This means we obtain the same complexity for a selective forgery as for an existential forgery under CMA.

UU-CMA. For a universal forgery, $\mathcal{A}$ can try to compute two message mappings B_1, B_2 such that either $(B_1)_i \leq r$ or $(B_2)_i \leq r$ for all $i \in \{0, \ldots, \ell-1\}$, where $r \in \{0, \ldots, w-1\}$. The probability that any two messages satisfy these rules equals $\left(1 - \left(\frac{w-(r+1)}{w}\right)^2\right)^{\ell}$, which means the probability that there exist two such messages in a list of τ messages is bounded by at least $1/2$ if $\binom{\tau}{2} \geq 1/2 \cdot \left(1 - \left(\frac{w-(r+1)}{w}\right)^2\right)^{-\ell}$, using again the same arguments as for Lamport and optimized Lamport. Now $\mathcal{A}$ obtains a successful forgery for M_3 with probability *at least* $\left(\frac{w-r}{w}\right)^{\ell}$, since we ignored the cases where $(B_3)_i$ is smaller than r, but still bigger than $(B_1)_i$ or $(B_2)_i$. The pre-computation τ and corresponding success probability for different values of w and $r \in \{0, \ldots, w-1\}$ are given in Figure 5.

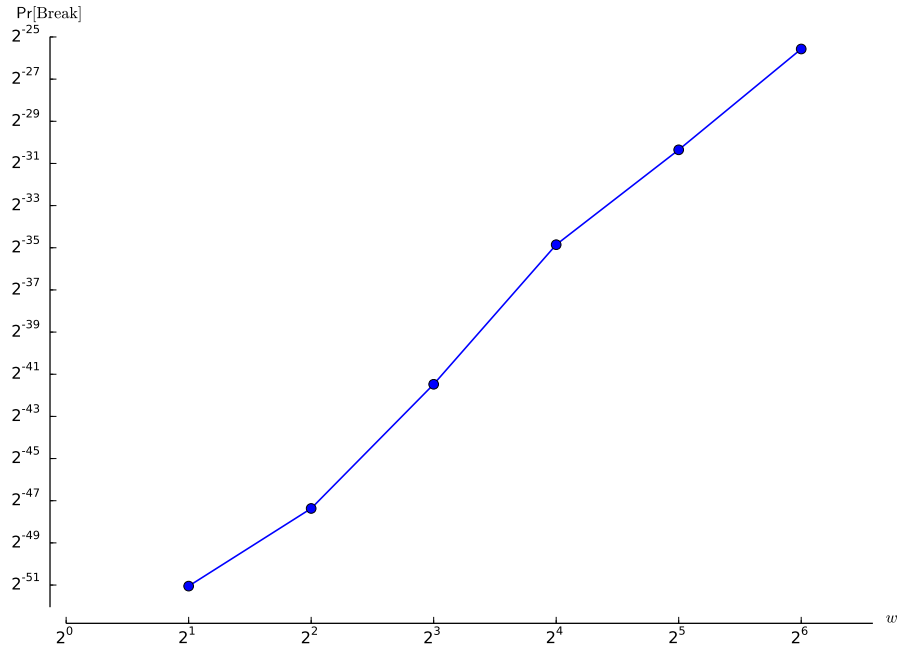

Fig. 4. This plot shows the logarithmic relation between w and Pr[break] for $w \in \{2, 4, 8, 16, 32, 64\}$. The logarithmic decrease of the exponent in Pr[break] is clearly making the probability grow faster for larger w.

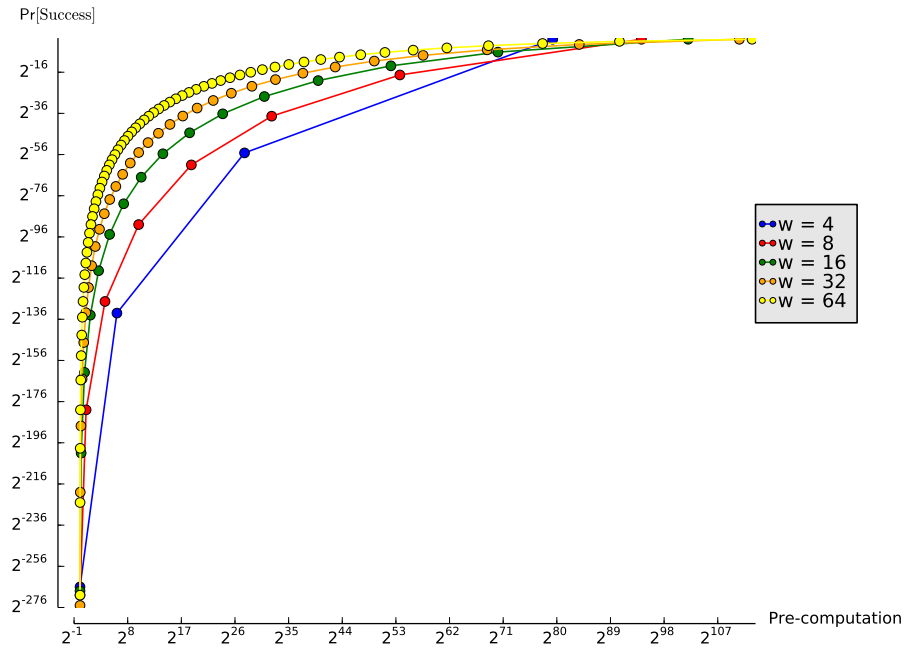

Fig. 5. This plot shows the relation between the amount of pre-computation and the lower bound for the success probability for a universal forgery under a chosen message attack on WOTS for different values of w and for each $r \in \{0, \ldots, w-1\}$.

EU-RMA. For WOTS, two messages cover a third one with probability:

$$\Pr[\text{break}(M_1, M_2, M_3) = 1] \approx \left(\frac{(w+1)(4w-1)}{6w^2}\right)^{\frac{m+\log m}{\log w}}.$$

This means that when an attacker receives two signatures of two random messages, it has to compute about $\tau = \left(\frac{(w+1)(4w-1)}{6w^2}\right)^{-\frac{m+\log m}{\log w}}$ messages to find a covered third message. For $m = 256$ and $w = 16$, this equals 2^{34}, which can be done within a few days on today's CPUs.

SU-RMA. For the selective forgery, an attacker can select an optimal message with a mapping that contains as high values as possible. For the analysis, we will use the same strategy as for the universal forgery, but in this case we want $(B_1)_i \geq r$ for all $i \in \{0, \ldots, \ell - 1\}$, which happens with probability $\left(\frac{w-r}{w}\right)^{\ell}$. Hence, the pre-computation can again be bound by $\tau \geq \left(\frac{w-r}{w}\right)^{-\ell}$ to upper bound the probability of finding such a message in a list of τ messages by at least $1/2$. The probability that the adversary can sign his selected message after he received two signatures on random messages equals $\left(1 - \left(\frac{w-(r+1)}{w}\right)^2\right)^{\ell}$ in this case. A plot of the computational costs with corresponding success probability is given in Figure 6. As for the optimized Lamport scheme, it looks similar to the graph of the universal forgery under chosen message attacks, but with lower success probabilities since $\mathcal{A}$ only has control over the selected message.

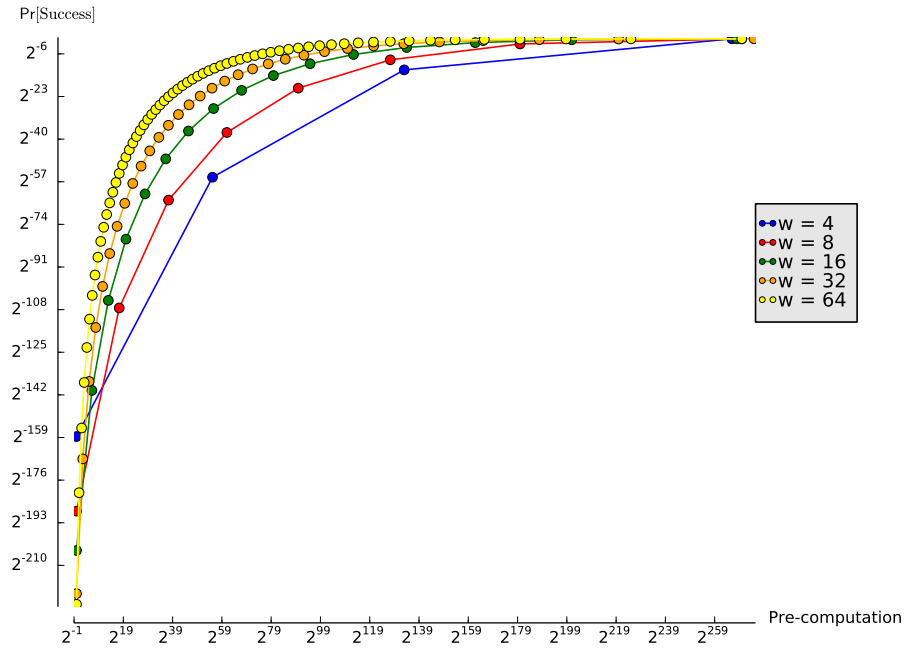

Fig. 6. This plot shows the relation between the amount of pre-computation and the success probability of a selective forgery under random message attacks on WOTS for different values of w and for each $r \in \{0, \ldots, w - 1\}$

UU-RMA. The probability of a successful universal forgery under a random message attack equals the probability that three random messages fulfill the break condition:

$$\Pr[\text{break}(M_1, M_2, M_3) = 1] \approx \left(\frac{(w+1)(4w-1)}{6w^2}\right)^{\frac{m+\log m}{\log w}}$$

The attacker has no influence on the process and cannot use any computational power before or after the online phase of the attack to increase his success probability.

FB-RMA. Similar to Lamport's and the optimized Lamport scheme, a full break occurs exactly when all secret values are exposed. For Winternitz with parameter w, this happens with probability $(1 - (\frac{w-1}{w})^2)^\ell$, which is a negligible probability for any w.

6 Experimental verifications

In sections 3, 4, and 5 we discussed the attack complexity of several different attacks. For the optimized Lamport scheme and WOTS, we assumed that the checksum is uniformly random and hence the message digest and its checksum behave as independent bit strings. However, as already mentioned there, the actual situation is that the checksum is dependent of the message digest. To verify the obtained results we carried out experiments for the EU-CMA case for optimized Lamport and WOTS.

We determined a lower bound for the number of calls τ to the message digest function H, such that a list of size τ of message digests, allows to find an existential forgery with probability upper bounded by at least 1/2. We performed several experiments for different values of τ, to see how realistic our assumption matches the real situation and how tight our bound is. We checked how many times a list of τ message mappings contained a cover for optimized Lamport scheme with digest length of $m = 128$ bits and for WOTS, with $m = 256$ and $w = 16$ (which are the parameters suggested in [9]). We performed 100 experiments per value of τ. As can be seen from the results in Table 5, the experiments closely match the theoretical results using the checksum simplification. The theoretical analysis predicts that $\tau = 2^9$ is required for the bound on the probability of an existential forgery to reach 1/2 for the optimized Lamport scheme with $m = 128$. For WOTS, the analysis suggests $\tau = 2^{12}$ when $m = 256$ and $w = 16$. From the results of the experiments, we can conclude that the simplifying assumption of independent message digests and checksums is not causing a significant difference to the real setting in the case of EU-CMA.

Remark 2. It is important to note that for extreme cases our analysis is not good enough. In the FB-CMA, UU-CMA, SU-RMA and FB-RMA settings for the optimized Lamport and Winternitz schemes, we are trying to push the message mappings to extreme cases to allow for forgeries. However, due to the inverse nature of the checksum, our analysis leads to impossible message mappings. For example, a high weight message part means a low weight checksum part for optimized Lamport, but in our analysis we are trying to push both message and checksum part to high weights. Therefor we expect the complexity to be much higher for these extreme cases, i.e. when r is very low or very high, with the meaning of r as described in optimized Lamport and Winternitz.

References

1. Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. Sphincs: Practical stateless hash-based signatures. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Computer Science*, pages 368–397. Springer Berlin Heidelberg, 2015.

Table 5. Experimental results for the success probability of an EU-CMA adversary, using a list of τ message mappings for the optimized Lamport (left table) with digest length $m = 128$ and for WOTS (right table) with $w = 16$ and digest length $m = 256$

τ	Pr[Success]
2^8	0.02
2^9	0.13
2^{10}	0.77
2^{11}	1.0
2^{12}	1.0

τ	Pr[Succes]
2^{11}	0.1
2^{12}	0.49
2^{13}	0.94
2^{14}	1.0
2^{15}	1.0

2. Johannes Buchmann, Erik Dahmen, Sarah Ereth, Andreas Hülsing, and Markus Rückert. On the security of the Winternitz one-time signature scheme. In A. Nitaj and D. Pointcheval, editors, *Africacrypt 2011*, volume 6737 of *Lecture Notes in Computer Science*, pages 363–378. Springer Berlin / Heidelberg, 2011.
3. Johannes Buchmann, Erik Dahmen, and Andreas Hülsing. XMSS - a practical forward secure signature scheme based on minimal security assumptions. In Bo-Yin Yang, editor, *Post-Quantum Cryptography*, volume 7071 of *LNCS*, pages 117–129. Springer, 2011.
4. Johannes Buchmann, Erik Dahmen, Elena Klintsevich, Katsuyuki Okeya, and Camille Vuillaume. Merkle signatures with virtually unlimited signature capacity. In Jonathan Katz and Moti Yung, editors, *Applied Cryptography and Network Security*, volume 4521 of *LNCS*, pages 31–45. Springer, 2007.
5. Johannes Buchmann, L. C. Coronado García, Erik Dahmen, Martin Döring, and Elena Klintsevich. CMSS - an improved Merkle signature scheme. In Rana Barua and Tanja Lange, editors, *Progress in Cryptology – INDOCRYPT 2006*, volume 4329 of *LNCS*, pages 349–363. Springer, 2006.
6. Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, volume 8042 of *LNCS*, pages 40–56. Springer, 2013. https://eprint.iacr.org/2013/383/.
7. S. Even, O. Goldreich, and S. Micali. On-line/off-line digital signatures. *Journal of Cryptology*, 9(1):35–67, 1996.
8. Andreas Hülsing. W-OTS+ – shorter signatures for hash-based signature schemes. In Amr Youssef, Abderrahmane Nitaj, and Aboul-Ella Hassanien, editors, *Progress in Cryptology – AFRICACRYPT 2013*, volume 7918 of *LNCS*, pages 173–188. Springer, 2013.
9. Andreas Hülsing, Denis Butin, Stefan-Lukas Gazdag, and Aziz Mohaisen. XMSS: Extended hash-based signatures. Internet Draft, IETF Crypto Forum Research Group, 2015.
10. Andreas Hülsing, Lea Rausch, and Johannes Buchmann. Optimal parameters for XMSS^{MT}. In Alfredo Cuzzocrea, Christian Kittl, Dimitris E. Simos, Edgar Weippl, and Lida Xu, editors, *Security Engineering and Intelligence Informatics*, volume 8128 of *LNCS*, pages 194–208. Springer, 2013.
11. Andreas Hülsing, Joost Rijneveld, and Fang Song. Mitigating multi-target attacks in hash-based signatures. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *Public-Key Cryptography – PKC 2016: 19th IACR International Conference on Practice and Theory in Public-Key Cryptography, Taipei, Taiwan, March 6-9, 2016, Proceedings, Part I*, pages 387–416, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.
12. Leslie Lamport. Constructing digital signatures from a one way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, 1979.
13. David McGrew and Michael Curcio. Hash-based signatures. Internet Draft, IETF, 2014.
14. David McGrew, Panos Kampanakis, Scott Fluhrer, Stefan-Lukas Gazdag, Denis Butin, and Johannes Buchmann. State management for hash based signatures. Cryptology ePrint Archive, Report 2016/357, 2016. https://eprint.iacr.org/2016/357.
15. Ralph Merkle. A certified digital signature. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO '89*, volume 435 of *LNCS*, pages 218–238. Springer, 1990.
16. NIST. Post-quantum cryptography: NIST's plan for the future, 2016. http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/pqcrypto-2016-presentation.pdf.

17. NSA. Commercial National Security Algorithm Suite. https://www.iad.gov/iad/programs/iad-initiatives/cnsa-suite.cfm, Last visited, July 2016.
18. Albrecht Petzoldt, Ming-Shing Chen, Bo-Yin Yang, Chengdong Tao, and Jintai Ding. Design principles for HFEv- based multivariate signature schemes. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015*, volume 9452 of *LNCS*, pages 311–334. Springer, 2015. http://www.iis.sinica.edu.tw/papers/byyang/19342-F.pdf.
19. Leonid Reyzin and Natan Reyzin. Better than BiBa: Short one-time signatures with fast signing and verifying. In Lynn Batten and Jennifer Seberry, editors, *Information Security and Privacy 2002*, volume 2384 of *Lecture Notes in Computer Science*, pages 1–47. Springer Berlin / Heidelberg, 2002.
20. Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1994)*, pages 124–134. IEEE Computer Society Press, 1994.

A Experiments for formal security notions

SU-CMA. Selective unforgeability is formally described by the following experiment. In this experiment $\mathcal{A}$ consists of two independent algorithms $(\mathcal{A}_1, \mathcal{A}_2)$. The first of which, $\mathcal{A}_1$, outputs the target message and some temporary state $\mathcal{S}$ that is forwarded to $\mathcal{A}_2$.

Experiment $\mathsf{Exp}^{\mathsf{SU\text{-}CMA}}_{\mathrm{Dss}(1^n)}(\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2))$
$(M_{\mathcal{A}}, \mathcal{S}) \leftarrow \mathcal{A}_1(1^n)$
$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{kg}(1^n)$
$\sigma^\star \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk}, M_{\mathcal{A}}, \mathcal{S})$
Let $\{(M_i, \sigma_i)\}_1^q$ be the query-answer pairs of $\mathsf{sign}(\mathsf{sk}, \cdot)$.
Return 1 iff $\mathsf{vf}(\mathsf{pk}, M_{\mathcal{A}}, \sigma^\star) = 1$ and $M_{\mathcal{A}} \notin \{M_i\}_1^q$.

UU-CMA. Universal unforgeability is formally described by the following experiment. The difference to the SU notion is that the target message $M_{\mathcal{A}}$ is now selected by the experiment.

Experiment $\mathsf{Exp}^{\mathsf{UU\text{-}CMA}}_{\mathrm{Dss}(1^n)}(\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2))$
$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{kg}(1^n)$
$\mathcal{S} \leftarrow \mathcal{A}_1^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$
$M_{\mathcal{A}} \xleftarrow{\$} \mathcal{M}$
$\sigma^\star \leftarrow \mathcal{A}_2(\mathcal{S}, M_{\mathcal{A}})$
Return 1 iff $\mathsf{vf}(\mathsf{pk}, M_{\mathcal{A}}, \sigma^\star) = 1$.

EU-RMA. Existential unforgeability under random message attacks (EU-RMA) is defined using the following experiment. Instead of giving the adversary oracle access as in the EU-CMA game, the experiment generates signatures on q random messages and hands these to the adversary.

Experiment $\mathsf{Exp}^{\mathsf{EU\text{-}RMA}}_{\mathrm{Dss}(1^n)}(\mathcal{A})$
$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{kg}(1^n)$
Let $\{(M_i, \sigma_i)\}_1^q$ be the set of q message signature pairs, obtained by
sampling $M_i \xleftarrow{\$} \mathcal{M}$ and computing $\sigma_i = \mathsf{sign}(\mathsf{sk}, M_i)$.
$(M^\star, \sigma^\star) \leftarrow \mathcal{A}(\mathsf{pk}, \{(M_i, \sigma_i)\}_1^q)$
Return 1 iff $\mathsf{vf}(\mathsf{pk}, M^\star, \sigma^\star) = 1$ and $M^\star \notin \{M_i\}_1^q$.

SU-RMA. Similarly to the previous notion, SU-RMA is defined be the experiment

Experiment $\mathsf{Exp}^{\mathsf{SU\text{-}RMA}}_{\mathrm{Dss}(1^n)}(\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2))$
$(M_{\mathcal{A}}, \mathcal{S}) \leftarrow \mathcal{A}_1(1^n)$
$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{kg}(1^n)$
Let $\{(M_i, \sigma_i)\}_1^q$ be the set of q message signature pairs, obtained by
sampling $M_i \xleftarrow{\$} \mathcal{M}$ and computing $\sigma_i = \mathsf{sign}(\mathsf{sk}, M_i)$.
$\sigma^\star \leftarrow \mathcal{A}(\mathsf{pk}, \{(M_i, \sigma_i)\}_1^q, M_{\mathcal{A}}, \mathcal{S})$
Return 1 iff $\mathsf{vf}(\mathsf{pk}, M_{\mathcal{A}}, \sigma^\star) = 1$.

UU-RMA. Finally, universal unforgeability under random message attacks is formally described by the following experiment.

Experiment $\mathsf{Exp}^{\mathsf{UU\text{-}RMA}}_{\mathrm{Dss}(1^n)}(\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2))$
$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{kg}(1^n)$
Let $\{(M_i, \sigma_i)\}_1^q$ be the set of q message signature pairs, obtained by
sampling $M_i \xleftarrow{\$} \mathcal{M}$ and computing $\sigma_i = \mathsf{sign}(\mathsf{sk}, M_i)$.
$\mathcal{S} \leftarrow \mathcal{A}_1(\mathsf{pk}, \{(M_i, \sigma_i)\}_1^q)$
$M_{\mathcal{A}} \xleftarrow{\$} \mathcal{M}$
$\sigma^\star \leftarrow \mathcal{A}_2(\mathcal{S}, M_{\mathcal{A}})$
Return 1 iff $\mathsf{vf}(\mathsf{pk}, M_{\mathcal{A}}, \sigma^\star) = 1$.