W-OTS$^+$– Shorter Signatures for Hash-Based Signature Schemes

Andreas Hülsing[⋆]
huelsing@cdc.informatik.tu-darmstadt.de

Cryptography and Computeralgebra
Department of Computer Science
TU Darmstadt

Abstract. We present W-OTS$^+$, a Winternitz type one-time signature scheme (W-OTS). We prove that W-OTS$^+$ is strongly unforgeable under chosen message attacks in the standard model. Our proof is exact and tight. The first property allows us to compute the security of the scheme for given parameters. The second property allows for shorter signatures than previous proposals without lowering the security. This improvement in signature size directly carries over to all recent hash-based signature schemes. I.e. we can reduce the signature size by more than 50% for XMSS$^+$ at a security level of 80 bits. As the main drawback of hash-based signature schemes is assumed to be the signature size, this is a further step in making hash-based signatures practical.

Keywords digital signatures, one-time signature schemes, hash-based signatures, provable security, hash functions

1 Introduction

Digital signatures are among the most important cryptographic primitives in practice. They have many applications, including the use in SSL/TLS and securing software updates. Hash-based or Merkle signature schemes (MSS) are an interesting alternative to the signature schemes used today, not only because they are assumed to resist quantum computer aided attacks, but also because of their fast signature generation and verification times as well as their strong security guarantees. Most MSS come with a standard model security proof and outperform RSA in many settings regarding runtimes. The main drawback of MSS is the signature size which to a large extent depends on the used one-time signature scheme (OTS).

[⋆] Supported by grant no. BU 630/19-1 of the German Research Foundation (www.dfg.de). An extended abstract of this work appeared in the proceedings of Africacrypt 2013

Recent MSS proposals [BDH11,HBB13] use a variant of the Winternitz OTS (W-OTS) introduced in [BDE$^+$11]. The main reason for this choice is the reduced signature size. Using W-OTS, a MSS signature does not have to contain the OTS public key as it can be computed given the W-OTS signature. Moreover, W-OTS type signature schemes allow for a trade-off between signature size and runtime.

In this work we introduce W-OTS$^+$, a W-OTS type OTS that allows to reduce the signature size more than previous W-OTS variants and reaches a higher level of security. We prove that W-OTS$^+$ is strongly unforgeable under adaptive chosen message attacks (SU-CMA) in the standard model, if the used hash function is second-preimage resistant, undetectable and one-way (Indeed, we only present the proof for EU-CMA security in this extended abstract). Previous proposals require non-standard assumptions to achieve SU-CMA security (i.e. „key-collision resistance“ in case of [BDE$^+$11]). Besides the SU-CMA secure variants there exist W-OTS that achieve EU-CMA security, either using a collision resistant, undetectable hash function [HM02,DSS05] or a pseudorandom function family [BDE$^+$11]. The first security requirement is strictly stronger than that of W-OTS$^+$. While the second is comparable, the corresponding proof is less tight. However, both cases result in larger signatures.

Besides provable security we are also concerned with the practical performance of the scheme. We show how to use the exact security proof to compute the security level of W-OTS$^+$ for a given set of parameters. Moreover we discuss how to instantiate W-OTS$^+$ in practice and present parameter sizes for recent MSS (XMSS [BDH11], XMSS$^+$ [HBB13]) when instantiated with W-OTS$^+$.

Organization. We start by introducing W-OTS$^+$ in Section 2. Afterwards we state our main result about the security of W-OTS$^+$ and prove it in Section 3. In Section 4 we discuss possible instantiations and compare W-OTS$^+$ with previous proposals. Finally, we conclude in Section 5.

2 The Winternitz One-Time Signature Scheme

In this section we describe W-OTS$^+$. The core idea of all W-OTS is to use a certain number of function chains starting from random inputs. These random inputs are the secret key. The public key consists of the final outputs of the chains, i.e. the end of each chain. A signature is computed by mapping the message to one intermediate value of each function chain. All previous variants of W-OTS constructed the function chains as plain iteration of the used function (or function family in case of [BDE$^+$11]). In contrast, for W-OTS$^+$ we use a special mode of iteration which enables the tight security proof without requiring the used hash function family to be collision resistant. We start with some preliminaries. Afterwards we present W-OTS$^+$.

2.1 Signature Schemes

We now fix some notation and define digital signature schemes and existential unforgeability under adaptive chosen message attacks (EU-CMA). Through out the paper we write $x \overset{\$}{\longleftarrow} \mathcal{X}$ if x is randomly chosen from the set $\mathcal{X}$ using the uniform distribution. We further write log for $\log_2$.

Digital signature schemes 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 $\forall(\mathsf{pk}, \mathsf{sk}) \longleftarrow \mathsf{Kg}(1^n), \forall(M \in \mathcal{M}) : \mathsf{Vf}(\mathsf{pk}, \mathsf{Sign}(\mathsf{sk}, M), M) = 1$.

EU-CMA Security 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 $\mathsf{Exp}^{\mathsf{EU\text{-}CMA}}_{\text{Dss}(1^n)}(\mathcal{A})$
    $(\mathsf{sk}, \mathsf{pk}) \longleftarrow \mathsf{Kg}(1^n)$
    $(M^\star, \sigma^\star) \longleftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk}, \cdot)}(\mathsf{pk})$
    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

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

Using this, we define EU-CMA the following way.

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

$$\text{InSec}^{\text{EU-CMA}}\left(\text{Dss}(1^n); t, q\right) \overset{def}{=} \max_{\mathcal{A}}\{\text{Succ}^{\text{EU-CMA}}_{\text{Dss}(1^n)}\left(\mathcal{A}\right)\} = negl(n)\,.$$

An EU-CMA secure one-time signature scheme (OTS) is a Dss that is EU-CMA secure as long as the number of oracle queries of the adversary is limited to one, i.e. $q = 1$.

2.2 W-OTS$^+$

Now we present W-OTS$^+$. Like all previous variants of W-OTS, W-OTS$^+$ is parameterized by security parameter $n \in \mathbb{N}$, the message length m and the Winternitz parameter $w \in \mathbb{N}, w > 1$, which determines the time-memory trade-off. The last two parameters are used to compute

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

Furthermore, W-OTS$^+$ uses a family of functions $\mathcal{F}_n : \{f_k : \{0,1\}^n \rightarrow \{0,1\}^n | k \in \mathcal{K}_n\}$ with key space $\mathcal{K}_n$. The reader might think of it as a cryptographic hash function family that is non-compressing. Using $\mathcal{F}_n$ we define the following chaining function.

$c^i_k(x, \mathbf{r})$*:* On input of value $x \in \{0,1\}^n$, iteration counter $i \in \mathbb{N}$, key $k \in \mathcal{K}$ and randomization elements $\mathbf{r} = (r_1, \ldots, r_j) \in \{0,1\}^{n \times j}$ with $j \geq i$, the chaining function works the following way. In case $i = 0$, c returns x ($c^0_k(x, \mathbf{r}) = x$). For $i > 0$ we define c recursively as

$$c^i_k(x, \mathbf{r}) = f_k(c^{i-1}_k(x, \mathbf{r}) \oplus r_i),$$

i.e. in every round, the function first takes the bitwise xor of the intermediate value and bitmask r and evaluates f_k on the result afterwards. We write $\mathbf{r}_{a,b}$ for the subset $r_a, \ldots, r_b$ of $\mathbf{r}$. In case $b < a$ we define $\mathbf{r}_{a,b}$ to be the empty string. We assume that the parameters m, w and the function family $\mathcal{F}_n$ are publicly known. Now we describe the three algorithms of W-OTS$^+$:

Key Generation Algorithm ($\mathsf{Kg}(1^n)$)*:* On input of security parameter n in unary the key generation algorithm choses $\ell + w - 1$ n-bit strings uniformly at random. The secret key $\mathsf{sk} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell)$ consists of the first ℓ random bit strings. The remaining $w - 1$ bit strings are used as the randomization elements $\mathbf{r} = (r_1, \ldots, r_{w-1})$ for c. Next, Kg chooses a function key $k \xleftarrow{\$} \mathcal{K}$ uniformly at random. The public verification key pk is computed as

$$\mathsf{pk} = (\mathsf{pk}_0, \mathsf{pk}_1, \ldots, \mathsf{pk}_\ell) = ((\mathbf{r}, k), c_k^{w-1}(\mathsf{sk}_1, \mathbf{r}), \ldots, c_k^{w-1}(\mathsf{sk}_\ell, \mathbf{r})).$$

Signature Algorithm ($\mathsf{Sign}(M, \mathsf{sk}, \mathbf{r})$)*:* On input of a m bit message M, secret signing key sk and the randomization elements $\mathbf{r}$, the signature algorithm first computes a base w representation of M: $M = (M_1 \ldots M_{\ell_1})$, $M_i \in \{0, \ldots, w-1\}$. Therefor, M is treated as the binary representation of a natural number x and then the w-ary representation of x is computed. Next it computes the checksum

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

and 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 concatenation of the base w representations of M and C. The signature is computed as

$$\sigma = (\sigma_1, \ldots, \sigma_\ell) = (c_k^{b_1}(\mathsf{sk}_1, \mathbf{r}), \ldots, c_k^{b_\ell}(\mathsf{sk}_\ell, \mathbf{r})).$$

Please note that the checksum guarantees that given the b_i, $0 < i \leq \ell$ corresponding to one message, the b'_i corresponding to any other message include at least one $b'_i < b_i$.

Verification Algorithm ($\mathsf{Vf}(1^n, M, \sigma, \mathsf{pk})$)*:* On input of message M of binary length m, a signature σ and a 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:

$$\begin{aligned} \mathsf{pk} &= (\mathsf{pk}_0, \mathsf{pk}_1, \ldots, \mathsf{pk}_\ell) \\ &\overset{?}{=} ((\mathbf{r}, k), c_k^{w-1-b_1}(\sigma_1, \mathbf{r}_{b_1+1,w-1}), \ldots, c_k^{w-1-b_\ell}(\sigma_\ell, \mathbf{r}_{b_\ell+1,w-1})) \end{aligned}$$

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

The runtime of all three algorithms is bounded by ℓw evaluations of f_k. The size of a signature and the secret key is $|\sigma| = |\mathsf{sk}| = \ell n$ bits. The public key size is $(\ell + w - 1)n + |k|$ bits, where $|k|$ denotes the number of bits required to represent any element of $\mathcal{K}$.

3 Security of W-OTS$^+$

In this section we analyze the security of W-OTS$^+$. We prove W-OTS$^+$ is existentially unforgeable under chosen message attacks, if the used function family is a second-preimage resistant family of undetectable one-way functions. More precisely, we prove the following theorem:

Theorem 1. *Let $n, w, m \in \mathbb{N}$, $w, m = poly(n)$, $\mathcal{F}_n : \{f_k : \{0,1\}^n \rightarrow \{0,1\}^n | k \in \mathcal{K}_n\}$ a second preimage resistant, undetectable one-way function family. Then,* $\mathrm{InSec}^{\text{EU-CMA}}\left(\textit{W-OTS}^{+}(1^n, w, m); t, 1\right)$*, the insecurity of W-OTS$^+$ against an* EU-CMA *attack is bounded by*

$$\begin{aligned}&\mathrm{InSec}^{\text{EU-CMA}}\left(\textit{W-OTS}^{+}(1^n, w, m); t, 1\right)\\&\leq w \cdot \mathrm{InSec}^{\text{UD}}\left(\mathcal{F}_n; t^\star\right) + w\ell \cdot \max\left\{\mathrm{InSec}^{\text{OW}}\left(\mathcal{F}_n; t'\right), w \cdot \mathrm{InSec}^{\text{SPR}}\left(\mathcal{F}_n; t'\right)\right\}\end{aligned}$$

with $t' = t + 3\ell w$ and $t^\star = t + 3\ell w + w - 1$.

It seems natural to assume that the existence of a function that combines these properties is equivalent to the existence of a one-way function. As the function has to be one-way itself, the one direction is trivial. On the other hand, we know that second-preimage resistant functions exist if a one-way function exists [Rom90] and we know the same for undetectable functions, i.e. pseudorandom generators [HILL99]. We leave the question if this also implies the existence of a function family that combines all three properties for future work. If this was the case, it would mean that W-OTS$^+$ has minimal security requirements. The practical implications of the proof are discussed in the next section.

In this extended abstract we only prove that W-OTS$^+$ is EU-CMA secure. In fact it also fulfills the stronger notion of SU-CMA, where the adversary is also allowed to return a new signature on the message send to the signature oracle. The claimed bound in Theorem 1 holds for the SU-CMA case, too. We present the EU-CMA proof, because it contains all important ideas but has less different cases to handle. Before we present the proof we give some preliminaries. At the end of this sections we show how to compute the security level of W-OTS$^+$.

3.1 Preliminaries

In this subsection we provide some more notation and formal definitions. We denote the uniform distribution over bit strings of length n by $\mathcal{U}_n$. In our proofs, we measure all runtimes counting the evaluations of elements from $\mathcal{F}_n$. In some proofs and definitions we use the (distinguishing) advantage of an adversary which we now define.

Definition 2 (Advantage). *Given two distributions $\mathcal{X}$ and $\mathcal{Y}$, we define the advantage* $\mathrm{Adv}_{\mathcal{X},\mathcal{Y}}(\mathcal{A})$ *of an adversary* $\mathcal{A}$ *in distinguishing between these two distributions as*

$$\mathrm{Adv}_{\mathcal{X},\mathcal{Y}}(\mathcal{A}) = |\mathsf{Pr}[1 \longleftarrow \mathcal{A}(\mathcal{X})] - \mathsf{Pr}[1 \longleftarrow \mathcal{A}(\mathcal{Y})]| \,.$$

Functions We now define three properties for families of functions that we use. In what follows, we only consider families $\mathcal{F}_n$ as defined in the last section.We require that it is possible given $n \in \mathbb{N}$ to sample a key k from key space $\mathcal{K}_n$ using the uniform distribution in polynomial time. Furthermore we require that all functions from $\mathcal{F}_n$ can be evaluated in polynomial time. We first recall the definitions of one-wayness (OW) and second preimage resistance (SPR).

The success probability of an adversary against the one-wayness of $\mathcal{F}_n$ is:

$$\begin{aligned}\mathrm{Succ}^{\mathrm{OW}}_{\mathcal{F}_n}(\mathcal{A}) =&\mathsf{Pr}\,[\,k \xleftarrow{\$} \mathcal{K}_n; x \xleftarrow{\$} \{0,1\}^n, y \longleftarrow f_k(x),\\ &x' \xleftarrow{\$} \mathcal{A}(k,y) : y = f_k(x')]\end{aligned} \tag{1}$$

The success probability of an adversary against the second preimage resistance of $\mathcal{F}_n$ is:

$$\begin{aligned}\mathrm{Succ}^{\mathrm{SPR}}_{\mathcal{F}_n}(\mathcal{A}) =&\mathsf{Pr}\,[\,k \xleftarrow{\$} \mathcal{K}_n; x \xleftarrow{\$} \{0,1\}^n, x' \longleftarrow \mathcal{A}(k,x) :\\ &(x \neq x') \wedge (f_k(x) = f_k(x'))]\end{aligned} \tag{2}$$

We call a function family $\mathcal{F}_n$ one-way (second preimage resistant, resp.) if the respective success probability given above of any PPT adversary is negligible in n.

Besides SPR and OW, we require $\mathcal{F}_n$ to provide another property called undetectability to proof W-OTS$^+$ secure. Intuitively, a function family is undetectable if its outputs can not be distinguished from uniformly random values. This is what we require from a pseudorandom generator, which in contrast to $\mathcal{F}_n$ has to be length expanding.

To define undetectability, assume the following two distributions over $\{0,1\}^n \times \mathcal{K}$. A sample (u,k) from the first distribution $\mathcal{D}_{\mathrm{UD},\mathcal{U}}$ is obtained by sampling $u \longleftarrow \mathcal{U}_n$ and $k \xleftarrow{\$} \mathcal{K}$ uniformly at random from the respective domain. A sample (u,k) from the second distribution $\mathcal{D}_{\mathrm{UD},\mathcal{F}}$ is obtained by sampling $k \xleftarrow{\$} \mathcal{K}$ and then evaluating f_k on a uniformly random bit string, i.e. $u \longleftarrow f_k(\mathcal{U}_n)$. The advantage of an adversary $\mathcal{A}$ against the undetectability of $\mathcal{F}_n$ is simply the distinguishing advantage for these two distributions:

$$\mathrm{Adv}^{\mathrm{UD}}_{\mathcal{F}_n}(\mathcal{A}) = \mathrm{Adv}_{\mathcal{D}_{\mathrm{UD},\mathcal{U}},\mathcal{D}_{\mathrm{UD},\mathcal{F}}}(\mathcal{A})$$

Using this we define undetectability as:

Definition 3 (Undetectability (UD)). *Let $n \in \mathbb{N}$, $\mathcal{F}_n$ a family of functions as described above. We call $\mathcal{F}_n$ undetectable, if* $\mathrm{InSec}^{\mathrm{UD}}(\mathcal{F}_n; t)$ *the advantage of any adversary $\mathcal{A}$ against the undetectability of $\mathcal{F}_n$ running in time less or equal t is negligible:*

$$\mathrm{InSec}^{\mathrm{UD}}(\mathcal{F}_n; t) \overset{def}{=} \max_{\mathcal{A}}\{\mathrm{Adv}^{\mathrm{UD}}_{\mathcal{F}_n}(\mathcal{A})\} = negl(n)\,.$$

Undetectability was already used by Dods et al. [DSS05] to prove a former version of W-OTS secure.

3.2 Security proof

We now present the proof of Theorem 1. The general idea is, that because of the checksum, a successful forgery must contain at least one intermediate value x for one chain α, that is closer to the start value of chain α than the value σ_α contained in the answer to the signature query. We try to guess the position of σ_α and place our preimage challenge y_c there. So we can answer the signature query and hopefully extract a preimage given x. We also include a second preimage challenge in the same chain α, manipulating the randomization elements. This is necessary, as x must lead to the same public key value pk_α than y_c but the chain continued from x does not need to contain y_c as an intermediate value. But in this case it contains a second preimage which we try to extract.

Manipulating the public key to place our challenges, we slightly change the distribution of the key. In the second part of the proof we show that this does not significantly change the success probability of the adversary using the undetectability of $\mathcal{F}_n$.

Proof (of Theorem 1). For the sake of contradiction assume there exists an adversary $\mathcal{A}$ that can produce existential forgeries for W-OTS$^+(1^n, w, m)$ running an adaptive chosen message attack in time $\leq t$ and with success probability $\epsilon_{\mathcal{A}} = \mathrm{Succ}^{\text{EU-CMA}}_{\text{W-OTS}(1^n,w,m)}(\mathcal{A})$ greater than the claimed bound $\mathrm{InSec}^{\text{EU-CMA}}\left(\text{W-OTS}^+(1^n, w, m); t, 1\right)$. We first show how to construct an oracle machine $\mathcal{M}^{\mathcal{A}}$ that either breaks the second preimage resistance or

Algorithm 1 $\mathcal{M}^{\mathcal{A}}$

Input: Security parameter n, function key k, one-way challenge y_c and second preimage resistance challenge x_c.
Output: A value x that is either a preimage of y_c or a second preimage for x_c under f_k or fail.

```
1. Run $\mathsf{Kg}(1^n)$ to generate W-OTS$^+$ key pair $(\mathsf{sk}, \mathsf{pk})$
2. Choose indices $\alpha \xleftarrow{\$} \{1, ..., \ell\}, \beta \xleftarrow{\$} \{1, \ldots, w-1\}$ uniformly at random
3. If $\beta = w - 1$ then set $\mathbf{r}' = \mathbf{r}$
4. Else
   (a) Choose index $\gamma \xleftarrow{\$} \{\beta + 1, \ldots, w - 1\}$ uniformly at random
   (b) Obtain $\mathbf{r}'$ from $\mathbf{r}$, replacing $r_\gamma$ by $c_k^{\gamma-\beta-1}(y_c, \mathbf{r}_{\beta+1,l}) \oplus x_c$.
5. Obtain $\mathsf{pk}'$ by setting $\mathsf{pk}'_i = c_k^{w-1}(\mathsf{sk}_i, \mathbf{r}'), 0 < i \leq \ell, i \neq \alpha$,
   $\mathsf{pk}'_\alpha = c_k^{w-1-\beta}(y_c, \mathbf{r}'_{\beta+1,w-1})$ and $\mathsf{pk}_0 = (\mathbf{r}', k)$
6. Run $\mathcal{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk}')$
7. If $\mathcal{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk}')$ queries Sign with message $M$ then
   (a) compute $B = (b_1, ..., b_\ell)$
   (b) If $b_\alpha < \beta$ then return fail
   (c) Generate signature $\sigma$ of $M$:
       i. Run $\sigma = (\sigma_1, \ldots, \sigma_\ell) \longleftarrow \mathsf{Sign}(M, \mathsf{sk}, \mathbf{r}')$
       ii. Set $\sigma_\alpha = c_k^{b_\alpha - \beta}(y_c, \mathbf{r}'_{\beta+1,w-1})$
   (d) Reply to query using $\sigma$
8. If $\mathcal{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$ returns valid $(\sigma', M')$ then
   (a) Compute $B' = (b'_1, ..., b'_\ell)$
   (b) If $b'_\alpha \geq \beta$ return fail
   (c) If $\beta = w - 1$
       i. Return preimage $c_k^{w-1-b'_\alpha-1}(\sigma'_\alpha, \mathbf{r}'_{b'_\alpha+1,w-1}) \oplus r_{w-1}$
   (d) Else
       i. If $c_k^{\beta-b'_\alpha}(\sigma'_\alpha, \mathbf{r}'_{b'_\alpha+1,w-1}) = y_c$ then
          return preimage $c_k^{\beta-b'_\alpha-1}(\sigma'_\alpha, \mathbf{r}'_{b'_\alpha+1,w-1}) \oplus r_\beta$
       ii. Else if $x' = c_k^{\gamma-b'_\alpha-1}(\sigma'_\alpha, \mathbf{r}_{b'_\alpha+1,w-1}) \oplus r_\gamma \neq x_c$ and $c_k^{\gamma-b'_\alpha}(\sigma'_\alpha, \mathbf{r}_{b'_\alpha+1,w-1}) =$
           $c_k^{\gamma-\beta}(y_c, \mathbf{r}_{\beta+1,w-1})$ return second preimage $x'$
9. In any other case return fail
```

one-wayness of $\mathcal{F}_n$ using $\mathcal{A}$ with a possibly different input distribution. A pseudo-code description of $\mathcal{M}^{\mathcal{A}}$ is given as Algorithm 1.

The oracle machine $\mathcal{M}^{\mathcal{A}}$ first runs the W-OTS$^+$ key generation to obtain a key pair $(\mathsf{sk}, \mathsf{pk})$. Then, $\mathcal{M}^{\mathcal{A}}$ selects the positions to place its challenges in the public key. Therefor it selects a random function chain choosing the index α. Second it chooses an index β to select a random intermediate value of this chain. $\mathcal{M}^{\mathcal{A}}$ places the preimage challenge at this position. This is done, setting y_c as the βth intermediate value of the chain. If $\beta < w - 1$, i.e. $\mathcal{M}^{\mathcal{A}}$ did not sample the last position in the chain, another intermediate value between β and the end of the chain is selected, sampling γ. $\mathcal{M}^{\mathcal{A}}$ places the second preimage challenge at the input of the γth evaluation of the chain continued from y_c, replacing the randomization element r_γ (Line 4b). A manipulated public key pk' is computed using the new set of randomization elements. The αth value of pk' is computed continuing the chain from y_c at position β (Line 5). Then $\mathcal{M}^{\mathcal{A}}$ runs $\mathcal{A}$ on input pk'.

W.l.o.g. we assume that $\mathcal{A}$ asks for the signature on one message M (Line 7). So $\mathcal{M}^{\mathcal{A}}$ computes the b_i as described in the signature algorithm. $\mathcal{M}^{\mathcal{A}}$ knows the secret key value sk_i for all chains with exception of chain α. For chain α $\mathcal{M}^{\mathcal{A}}$ only knows the βth intermediate value. Hence, $\mathcal{M}^{\mathcal{A}}$ can answer the query if $b_\alpha \geq \beta$ as all intermediate values $\geq \beta$ of the αth chain can be computed using y_c. If this is not the case, $\mathcal{M}^{\mathcal{A}}$ aborts.

If $\mathcal{A}$ returns an existential forgery (σ', M'), $\mathcal{M}^{\mathcal{A}}$ computes the b'_i. The forgery is only useful if $b'_\alpha < \beta$. If this is not the case, $\mathcal{M}^{\mathcal{A}}$ returns fail. Now, there are two mutually exclusive cases. If $\beta = w-1$, i.e. we selected the end of chain α, the forgery contains a preimage of y_c. This is the case because σ'_α is an intermediate value of chain alpha that ends in y_c. So, $\mathcal{M}^{\mathcal{A}}$ extracts the preimage and returns it (Line 8(c)i). Otherwise, there are again two mutually exclusive cases. The chain continued from σ'_α either has y_c as the βth intermediate value or it has not. In the first case, again a preimage can be extracted (Line 8(d)i). In the second case, the chains continued from y_c and σ'_α must collide at some position between $\beta + 1$ and $w - 1$ according to the pigeonhole principle. If they collide at position γ for the first time, a second preimage for x_c can be extracted (Line 8(d)ii). Otherwise $\mathcal{M}^{\mathcal{A}}$ aborts.

Now we compute the success probability of $\mathcal{M}^{\mathcal{A}}$. To make it easier, we only compute the probability for a certain success case. We assume that the b_α obtained from $\mathcal{A}$'s query equals β. This happens with probability w^{-1} as β was chosen uniformly at random. As our modifications might have changed the input distribution of $\mathcal{A}$, it does not necessarily succeed with probability $\epsilon_{\mathcal{A}}$. For the moment we only denote the probability that $\mathcal{A}$ returns a valid forgery when run by $\mathcal{M}^{\mathcal{A}}$ as $\epsilon'_{\mathcal{A}}$. Because of the construction of the check sum, M' leads to at least one $b'_i < b_i, 0 < i \leq \ell$. With probability ℓ^{-1} this happens for $i = \alpha$ and the condition in line 8b is fulfilled. At this point there are two mutually exclusive cases, so one of them occurs with probability p and the other one with probability $(1-p)$.

Case 1: Either $\beta = w-1$ or the chain continued from σ'_α has y_c as the βth intermediate value. In this case, $\mathcal{M}^{\mathcal{A}}$ returns a preimage for y_c with probability 1.

Case 2: $\beta < w-1$ and the chain continued from σ'_α does not have y_c as the βth intermediate value. In this case, $\mathcal{M}^{\mathcal{A}}$ returns a second preimage for x_c if the two chains collide for the first time at position γ. This happens with probability greater w^{-1} as gamma was chosen uniformly at random from within the interval $[\beta+1, w-1]$.

Using the assumptions about the one-wayness and second preimage resistance of $\mathcal{F}_n$ we can bound the success probability of $\mathcal{A}$ if called by $\mathcal{M}^{\mathcal{A}}$:

$$\epsilon'_{\mathcal{A}} \leq w\ell \cdot \max\left\{\mathrm{InSec}^{\mathrm{OW}}\left(\mathcal{F}_n; t'\right), w \cdot \mathrm{InSec}^{\mathrm{SPR}}\left(\mathcal{F}_n; t'\right)\right\} \tag{3}$$

where the time $t' = t + 3\ell w$ is an upper bound for the runtime of $\mathcal{A}$ plus the time needed to run each algorithm of W-OTS$^+$ once.

As the second step, we bound the difference between the success probability $\epsilon'_{\mathcal{A}}$ of $\mathcal{A}$ when called by $\mathcal{M}^{\mathcal{A}}$ and its success probability $\epsilon_{\mathcal{A}}$ in the original experiment. If the first is greater than the latter we already have a contradiction. Hence we assume $\epsilon_{\mathcal{A}} \geq \epsilon'_{\mathcal{A}}$ in what follows. Please note, that among the elements of pk' only the distribution of pk'_α might differ from the distribution of a public key generated by Kg. r_γ is uniformly distributed in $\{0,1\}^n$, because x_c is uniformly distributed in $\{0,1\}^n$. We define two distributions $\mathcal{D}_{\mathcal{M}}$ and $\mathcal{D}_{\mathsf{Kg}}$ over $\{0,\ldots,w-1\} \times \{0,1\}^n \times \{0,1\}^{(n\times w-1)} \times \mathcal{K}$. A sample $(\beta, u, \mathbf{r}, k)$ follows $\mathcal{D}_{\mathcal{M}}$ if the entries $\beta \xleftarrow{\$} \{0,\ldots,w-1\}$, $u \xleftarrow{\$} \{0,1\}^n$, $\mathbf{r} \xleftarrow{\$} \{0,1\}^{n\times w-1}$ and $k \xleftarrow{\$} \mathcal{K}$ are chosen uniformly at random. A sample $(\beta, u, \mathbf{r}_{1,i}, k)$ follows $\mathcal{D}_{\mathsf{Kg}}$ if $\beta \xleftarrow{\$} \{0,\ldots,w-1\}$, $\mathbf{r} \xleftarrow{\$} \{0,1\}^{n\times w-1}$ and $k \xleftarrow{\$} \mathcal{K}$ are chosen uniformly at random and $u = c_k^{\beta}(\mathcal{U}_n, \mathbf{r})$. So the two distributions only differ in the way u is chosen. We now construct an oracle machine $\mathcal{M}'^{\mathcal{A}}$ that uses the possibly different behavior of $\mathcal{A}$ when given differently distributed inputs, to distinguish between $\mathcal{D}_{\mathsf{Kg}}$ and $\mathcal{D}_{\mathcal{M}}$. Using $\mathcal{M}'^{\mathcal{A}}$ we can then upper bound $\epsilon_{\mathcal{A}}$ by a function of the distinguishing advantage of $\mathcal{M}'^{\mathcal{A}}$ and $\epsilon'_{\mathcal{A}}$. Afterwards we use a hybrid argument to bound the distinguishing advantage of $\mathcal{M}'^{\mathcal{A}}$ using the undetectability of $\mathcal{F}_n$.

The oracle machine $\mathcal{M}'^{\mathcal{A}}$ works the following way. On input of a sample $(\beta, u, \mathbf{r}, k)$ that is either chosen from $\mathcal{D}_{\mathcal{M}}$ or from $\mathcal{D}_{\mathsf{Kg}}$, $\mathcal{M}'^{\mathcal{A}}$ generates a W-OTS$^+$ key pair. Instead of using Kg, $\mathcal{M}'^{\mathcal{A}}$ samples a secret key $\mathsf{sk} \xleftarrow{\$} \{0,1\}^{n \times \ell}$ and an index $\alpha \xleftarrow{\$} \{1, \ldots, \ell\}$ uniformly at random. It computes the public key pk as $\mathsf{pk}_0 = (\mathbf{r}, k)$ and

$$\mathsf{pk}_i = \begin{cases} c_k^{w-1}(\mathsf{sk}_i, \mathbf{r}) & \text{, if } 1 \leq i \leq \ell \text{ and } i \neq \alpha \\ c_k^{w-1-\beta}(u, \mathbf{r}_{\beta+1,w-1}) & \text{, if } i = \alpha. \end{cases}$$

Then $\mathcal{M}'^{\mathcal{A}}$ runs $\mathcal{A}$ on input pk. If $\mathcal{A}$ queries $\mathcal{M}'^{\mathcal{A}}$ for the signature on a message M, $\mathcal{M}'^{\mathcal{A}}$ behaves the same way as $\mathcal{M}^{\mathcal{A}}$. If $b_\alpha \geq \beta$, $\mathcal{M}'^{\mathcal{A}}$ uses sk and u to compute the signature, otherwise it aborts. If $\mathcal{A}$ returns a valid forgery, $\mathcal{M}'^{\mathcal{A}}$ returns 1 and otherwise 0. The runtime of $\mathcal{M}'^{\mathcal{A}}$ is bounded by the runtime of $\mathcal{A}$ plus one evaluation of each algorithm of W-OTS$^+$. So we get $t'' = t + 3\ell w$ as an upper bound.

Now, we compute the distinguishing advantage $\mathrm{Adv}_{\mathcal{D}_{\mathcal{M}}, \mathcal{D}_{\mathsf{Kg}}}\left(\mathcal{M}'^{\mathcal{A}}\right)$ of $\mathcal{M}'^{\mathcal{A}}$. If the sample is taken from $\mathcal{D}_{\mathcal{M}}$, the distribution of the public keys pk generated by $\mathcal{M}'^{\mathcal{A}}$ is the same as the distribution of the public keys pk' generated by $\mathcal{M}^{\mathcal{A}}$. Hence $\mathcal{M}'^{\mathcal{A}}$ outputs 1 with probability

$$\mathsf{Pr}\left[(\beta, u, \mathbf{r}, k) \longleftarrow \mathcal{D}_{\mathcal{M}} : 1 \longleftarrow \mathcal{M}'^{\mathcal{A}}(\beta, u, \mathbf{r}, k)\right] = \epsilon'_{\mathcal{A}}.$$

If the sample was taken from $\mathcal{D}_{\mathsf{Kg}}$, the public keys pk generated by $\mathcal{M}'^{\mathcal{A}}$ follow the same distribution than those generated by Kg and so $\mathcal{M}'$ outputs 1 with probability

$$\mathsf{Pr}\left[(\beta, u, \mathbf{r}, k) \longleftarrow \mathcal{D}_{\mathsf{Kg}} : 1 \longleftarrow \mathcal{M}'^{\mathcal{A}}(\beta, u, \mathbf{r}, k)\right] = \epsilon_{\mathcal{A}}.$$

So the distinguishing advantage of $\mathcal{M}'^{\mathcal{A}}$ is

$$\mathrm{Adv}_{\mathcal{D}_{\mathsf{Kg}}, \mathcal{D}_{\mathcal{M}}}\left(\mathcal{M}'^{\mathcal{A}}\right) = \left|\epsilon_{\mathcal{A}} - \epsilon'_{\mathcal{A}}\right|.$$

As mentioned above, we only have to consider the case $\epsilon_{\mathcal{A}} \geq \epsilon'_{\mathcal{A}}$. So we obtain the following bound on $\epsilon_{\mathcal{A}}$:

$$\epsilon_{\mathcal{A}} = \mathrm{Adv}_{\mathcal{D}_{\mathsf{Kg}}, \mathcal{D}_{\mathcal{M}}}\left(\mathcal{M}'^{\mathcal{A}}\right) + \epsilon'_{\mathcal{A}} \tag{4}$$

We now limit the distinguishing advantage of $\mathcal{M}'^{\mathcal{A}}$ in our last step. We use a hybrid argument to show that this advantage is bound by the undetectability of $\mathcal{F}_n$. For a given $\beta \in \{0, \ldots, w-1\}$, we define the hybrids $H_j = (\beta, c_k^{\beta-j}(\mathcal{U}_n, \mathbf{r}_{j+1,w-1}), \mathbf{r}, k)$ with $\mathbf{r} \xleftarrow{\$} \{0,1\}^{n \times w-1}, k \xleftarrow{\$} \mathcal{K}$ for $0 \leq j \leq \beta$. Given an adversary $\mathcal{B}$ that can distinguish between H_0 and

H_β with advantage $\epsilon_{\mathcal{B}}$, a hybrid argument leads that there must exist two consecutive hybrids that $\mathcal{B}$ distinguishes with advantage $\geq \epsilon_{\mathcal{B}}/\beta$. Assume these two hybrids are H_α and $H_{\alpha+1}$. Then we can construct an oracle machine $\mathcal{M}''^{\mathcal{B}}$ that uses $\mathcal{B}$ to distinguish between $\mathcal{D}_{\text{UD},\mathcal{U}}$ and $\mathcal{D}_{\text{UD},\mathcal{F}}$ as defined in the preliminaries and thereby attacking the undetectability of $\mathcal{F}_n$. Given a distinguishing challenge (u, k), $\mathcal{M}''^{\mathcal{B}}$ selects $\mathbf{r} \longleftarrow \mathcal{U}_n^{w-1}$, computes $x = c^{\beta-(\alpha+1)}(u, \mathbf{r}_{\alpha+2,w-1})$, runs $b \longleftarrow \mathcal{B}(\beta, x, \mathbf{r}, k)$ and outputs b.

Let's analyze the advantage $\text{Adv}^{\text{UD}}_{\mathcal{F}_n}\left(\mathcal{M}''^{\mathcal{B}}\right)$ of $\mathcal{M}''^{\mathcal{B}}$. If the sample is taken from $\mathcal{D}_{\text{UD},\mathcal{U}}$, u is uniformly random and $x = c^{\beta-(\alpha+1)}(u, \mathbf{r}_{\alpha+2,w-1})$ is distributed exactly like the second element of $H_{\alpha+1}$. Otherwise, if the sample is taken from $\mathcal{D}_{\text{UD},\mathcal{F}}$, then $u \longleftarrow f_k(\mathcal{U}_n)$ is an output of f_k and we get

$$
\begin{aligned}
x &= c^{\beta-(\alpha+1)}(f_k(\mathcal{U}_n), \mathbf{r}_{\alpha+2,w-1}) = c^{\beta-(\alpha+1)+1}(\mathcal{U}_n \oplus r_{\alpha+1}, \mathbf{r}_{\alpha+1,w-1}) \\
&= c^{\beta-\alpha}(\mathcal{U}_n, \mathbf{r}_{\alpha+1,w-1}) = H_{\alpha_{(2)}}
\end{aligned}
$$

where $H_{\alpha_{(2)}}$ denotes the second element of H_α. Here we used the fact, that the xor of a uniformly distributed variable and a fixed value leads again to a uniformly distributed variable. Summing up, the input of $\mathcal{B}$, produced by $\mathcal{M}''^{\mathcal{B}}$ is either distributed like H_α or like $H_{\alpha+1}$, depending on $\mathcal{M}''^{\mathcal{B}}$s distinguishing challenge. Hence, the advantage of $\mathcal{M}''^{\mathcal{B}}$ is exactly that of $\mathcal{B}$ distinguishing between these two hybrids. So we get

$$\text{Adv}^{\text{UD}}_{\mathcal{F}_n}\left(\mathcal{M}''^{\mathcal{B}}\right) \geq \epsilon_D/\beta.$$

As the advantage of $\mathcal{M}''^{\mathcal{B}}$ is bounded by the undetectability of $\mathcal{F}_n$ per assumption, $\mathcal{M}'^{\mathcal{A}}$ does exactly what we assume $\mathcal{B}$ to do and the runtime of $\mathcal{M}''^{\mathcal{B}}$ is that of $\mathcal{B}$ plus at most $w-1$ evaluations of elements from $\mathcal{F}_n$, we get

$$\text{InSec}^{\text{UD}}\left(\mathcal{F}_n; t^\star\right) \geq \text{Adv}^{\text{UD}}_{\mathcal{F}_n}\left(\mathcal{M}''^{\mathcal{B}}\right) \geq \frac{\epsilon_{\mathcal{B}}}{i} = \frac{\text{Adv}_{\mathcal{D}_{\mathsf{Kg}},\mathcal{D}_{\mathcal{M}}}\left(\mathcal{M}'^{\mathcal{A}}\right)}{\beta}$$

where $t^\star = t'' + w - 1 = t + 3\ell w + w - 1$ is the runtime of $\mathcal{M}''^{\mathcal{B}}$. As $\beta \in \{0, \ldots, w-1\}$, we obtain the following bound on the advantage of $\mathcal{M}'^{\mathcal{A}}$:

$$\text{Adv}_{\mathcal{D}_{\mathsf{Kg}},\mathcal{D}_{\mathcal{M}}}\left(\mathcal{M}'^{\mathcal{A}}\right) \leq w \cdot \text{InSec}^{\text{UD}}\left(\mathcal{F}_n; t^\star\right). \tag{5}$$

Putting equations (3), (4) and (5) together we obtain a final bound on $\epsilon_{\mathcal{A}}$ which leads the required contradiction:

$$\epsilon_{\mathcal{A}} \leq w \cdot \text{InSec}^{\text{UD}}\left(\mathcal{F}_n; t^\star\right) + w\ell \cdot \max\left\{\text{InSec}^{\text{OW}}\left(\mathcal{F}_n; t'\right), w \cdot \text{InSec}^{\text{SPR}}\left(\mathcal{F}_n; t'\right)\right\}$$

with $t' = t + 3\ell w$ and $t^\star = t + 3\ell w + w - 1$. □

3.3 Security Level of W-OTS$^+$

Given Theorem 1, we can compute the security level in the sense of [Len04]. This allows a comparison of the security of W-OTS$^+$ with the security of a symmetric primitive like a block cipher for given security parameters. Following [Len04], we say that W-OTS$^+$ has security level b if a successful attack on the scheme can be expected to require 2^{b-1} evaluations of functions from $\mathcal{F}_n$ on average. We can compute the security level, finding a lower bound for t s.th. $1/2 \leq \mathrm{InSec}^{\text{EU-CMA}}\left(\text{W-OTS}(1^n, w, m); t, 1\right)$. According to the proof of Theorem 1, W-OTS$^+$ can only be attacked by either attacking the second preimage resistance, one-wayness or undetectability of $\mathcal{F}_n$. Following the reasoning in [Len04], we only take into account generic attacks on $\mathcal{F}_n$.

Regarding the insecurity of $\mathcal{F}(n)$ under generic attacks we assume $\mathrm{InSec}^{\text{SPR}}\left(\mathcal{F}(n); t\right) = \mathrm{InSec}^{\text{OW}}\left(\mathcal{F}(n); t\right) = \frac{t}{2^n}$ which corresponds to a brute force search for (second-)preimages. For the insecurity regarding undetectability we assume $\mathrm{InSec}^{\text{UD}}\left(\mathcal{F}(n); t\right) = \frac{t}{2^n}$ following [DSS05]. In the following we assume that the small additive increase of the attack runtime coming from the reduction is negligible, compared to the value of t for any practical attack. So we assume $t = t' = t^\star$. We compute the lower bound on t.

$$\begin{aligned}\frac{1}{2} &\leq \mathrm{InSec}^{\text{EU-CMA}}\left(\text{W-OTS}(1^n, w, m); t, 1\right)\\ &\leq w\frac{t}{2^n} + w\ell \cdot \max\left\{\frac{t}{2^n}, w \cdot \frac{t}{2^n}\right\} = \frac{tw}{2^n} + \frac{tw^2\ell}{2^n} = \frac{t(w^2\ell + w)}{2^n}\end{aligned}$$

Solving this for t gives us

$$t \geq \frac{1}{2} \cdot \frac{2^n}{w^2\ell + w} = 2^{n-1-\log(w^2\ell+w)}.$$

So, for the security level b we obtain $b \geq n - \log(w^2\ell + w)$.

4 W-OTS$^+$ in Practice

In this section we discuss the practical implications of our result. We first present practical instantiations of W-OTS$^+$. Then we discuss the implications of the new security proof, comparing W-OTS$^+$ to other W-OTS type OTS and present results for XMSS and XMSS$^+$ when instantiated using W-OTS$^+$.

4.1 Instantiations

To use W-OTS$^+$ in practice $\mathcal{F}_n$ has to be instantiated. We propose two different instantiations. The first and most obvious way to instantiate $\mathcal{F}_n$ is to simply use a cryptographic hash function like SHA2 or SHA3. These functions are assumed to fulfill all the properties we require $\mathcal{F}_n$ to provide. In case the input length of the function is bigger then the output length, we pad the inputs using the required number of zeros. As we do not allow arbitrary length messages, we do not need a more involved padding.

Another way is to use a block cipher. It is well known that a cryptographic hash function can be constructed using a block cipher. This is very useful, as many smart cards and CPUs provide hardware acceleration for AES. To construct $\mathcal{F}_n$ using a block cipher, we apply the Matyas-Meyer-Oseas (MMO) construction [MMO85] in a manner similar to [BDH11]. The MMO construction was shown to be secure by Black et al. [BRS02]. Assume we have a block cipher $E_n : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ with block and key size n. Then we construct $\mathcal{F}_n$ with key space $\mathcal{K} = \{0,1\}^n$ defining the elements of $\mathcal{F}_n$ as $f_k(x) = E_k(x) \oplus x$ where $E_K(M)$ denotes an evaluation of E using key K and message M. So, one evaluation of f_k takes either one evaluation of the used hash function or one evaluation of the underlying block cipher.

4.2 Performance Comparison

We now compare the performance of W-OTS$^+$ with that of the schemes from [DSS05] and [BDE$^+$11] which we call W-OTS^{CR} and W-OTS^{PRF}, respectively. Comparing W-OTS$^+$ with W-OTS^{CR}, the most important point is, that W-OTS^{CR} requires an undetectable collision resistant hash function. While this is a strictly stronger security requirement, it also has practical implications. Namely, collision resistance is threatened by birthday attacks. Hence, to achieve a security level of b bits, a hash function with $n = 2b$ bits output size is required. This leads to larger signatures and slows down the scheme, as in general hash functions get slower with increased output size. It is possible to reach the same signature size as for W-OTS$^+$ using a greater w, but this further slows down the scheme. On the other hand, the W-OTS$^+$ public key is bigger than that of W-OTS^{CR} which has only ℓn bits. This is because of the randomization elements. But as we will show later, this is of no relevance in many practical scenarios as we can reuse randomness.

Comparing W-OTS$^+$ with W-OTS^{PRF}, the differences are more subtle. First, looking at the instantiations, when using a hash function H to

Table 1. Parameters for signatures below 1kB for message length $m = 256$ and security level $b \geq 100$. For W-OTS^{PRF} this is impossible so we give the best possible signature size for $b \geq 100$. Runtime is given in number of evaluations of $\mathcal{F}_n$. As key generation, signature and verification times are the same, we only included the signature time $\mathsf{t}_{\mathsf{Sign}}$.

	n	w	$\|\sigma\|$	$\mathsf{t}_{\mathsf{Sign}}$	b
W-OTS^{+}	128	21	992	1,302	113
W-OTS^{CR}	256	455	992	14,105	128
W-OTS^{PRF}	128	8	1,440	720	100

instantiate W-OTS^{PRF}, two evaluations of H are needed per evaluation of $\mathcal{F}_n$ (see [BDH11]) in contrast to one for W-OTS^{+}. So the runtimes are doubled in this case. For a block cipher based instantiation the runtimes are the same. Second, at a first glance the sizes of both schemes are the same, only the W-OTS^{+} public key contains the additional randomization elements. But the bit security of W-OTS^{PRF} is $n - w - 1 - 2\log(\ell w)$, i.e. it contains w as a negative linear term while the bit security of W-OTS^{+} only looses a term logarithmic in w. In practice, the consequence of this difference is that the possible choices for w are limited if we target a certain bit security. This is best illustrated in the following example. Table 1 shows sizes and runtimes for a signature size below 1kB at a security level of 100 bit or more. Using W-OTS^{PRF} it is simply impossible to achieve a signature size below 1kB at 100 bit security. For W-OTS^{CR} it is theoretically possible, but one needs more than 10 times the number of evaluations of $\mathcal{F}_n$ which are also slower because of the bigger n.

4.3 Impact on XMSS and XMSS^{+}

OTS have numerous applications. The application that motivated this work is usage in hash-based signature schemes. Current hash-based signature schemes like XMSS [BDH11] and XMSS^{+} [HBB13] are based on W-OTS^{PRF} which turned out to be the best choice for an OTS so far. In the following we will shortly discuss what happens if we replace W-OTS^{PRF} by W-OTS^{+}. We do not describe XMSS and XMSS^{+} in detail due to the constrained space and refer the reader to the original papers. Table 2 shows a table from [HBB13] where we recomputed the results for the case that W-OTS^{+} is used. Where the values changed, we included the old values for W-OTS^{PRF} in brackets. The table shows, that in most cases the public key of the overall scheme does not change. The reason is that XMSS and XMSS^{+} public keys already contain public randomization elements that can be reused. There is only one case where

Table 2. Results for XMSS and XMSS$^+$ using W-OTS$^+$ for message length $m = 256$ on an Infineon SLE78. We use the same k and w for both trees. b denotes the security level in bits. The signature times are worst case times. Numbers in brackets are the values when using W-OTS^{PRF}.

				Timings (ms)			Sizes (byte)			
Scheme	h	k	w	KeyGen	Sign	Verify	Secret key	Public key	Signature	b
XMSS$^+$	16	2	4	5,600	106	25	3,760	544	3,476	96 (85)
XMSS$^+$	16	2	8	5,800	105	21	3,376	512	2,436	95 (81)
XMSS$^+$	16	2	16	6,700	118	22	3,200	512	1,892	93 (71)
XMSS$^+$	16	2	32	10,500	173	28	3,056	544 (480)	1,588	92 (54)
XMSS$^+$	20	4	4	22,200	106	25	4,303	608	3,540	92 (81)
XMSS$^+$	20	4	8	22,800	105	21	3,920	576	2,500	91 (77)
XMSS$^+$	20	4	16	28,300	124	22	3,744	576	1,956	89 (67)
XMSS$^+$	20	4	32	41,500	176	28	3,600	544	1,652	88 (50)
XMSS	10	4	4	14,600	86	22	1,680	608	2,292	103 (92)
XMSS	10	4	16	18,800	100	17	1,648	576	1,236	100 (78)
XMSS	16	4	4	925,400	134	23	2,448	800	2,388	97 (86)
XMSS	16	4	16	1,199,100	159	18	2,416	768	1,332	94 (72)

randomization elements have to be added. We assume that the runtimes do not change. The W-OTS^{PRF} function chains were implemented using one AES encryption per iteration. As shown above the same can be done for W-OTS$^+$, requiring one additional xor operation per AES evaluation. This should not lead any recognizable overhead. Moreover, the table shows that certain parameter sets — those with small signatures — have a very low level of security when using W-OTS^{PRF}. In practice a scheme has to provide at least a security level of 80 bits. Hence, these parameter sets could not be used before. Using W-OTS$^+$, the same parameter sets now lead to a level of security above 80 bits. Hence, they can now be used in practice.

5 Conclusion

In this work we introduced W-OTS$^+$. We proved its security, showed how to compute the security level of a given parameter set and discussed possible practical instantiations. As shown in the last section, W-OTS$^+$ can be used to decrease the signature size of hash-based signature schemes significantly without lowering the security of the scheme. I.e. we can decrease the signature size by 50% for XMSS$^+$ at a security level of 80 bits. Hopefully this leads to a broader acceptance of hash-based signature schemes, as the signature size was so far assumed to be the main drawback of these schemes. The only drawback of W-OTS$^+$ compared to previous W-OTS variants is the increased public key size. As for the case of hash-based signature schemes, it might be possible to reuse public randomness in other scenarios to mitigate this, too. An interesting question we left open is whether the existance of a one-way function implies the existence of a second-preimage resistant family of undetectable one-way functions.

References

[BDE$^+$11] 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.

[BDH11] 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 *Lecture Notes in Computer Science*, pages 117–129. Springer Berlin / Heidelberg, 2011.

[BRS02] John Black, Phillip Rogaway, and Thomas Shrimpton. Black-box analysis of the block-cipher-based hash-function constructions from PGV. In Moti Yung, editor, *Advances in Cryptology — CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 103–118. Springer Berlin / Heidelberg, 2002.

[DSS05] Chris Dods, Nigel Smart, and Martijn Stam. Hash based digital signature schemes. In Nigel Smart, editor, *Cryptography and Coding*, volume 3796 of *Lecture Notes in Computer Science*, pages 96–115. Springer Berlin / Heidelberg, 2005.

[HBB13] Andreas Hülsing, Christoph Busold, and Johannes Buchmann. Forward secure signatures on smart cards. In Lars R. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography*, volume 7707 of *Lecture Notes in Computer Science*, pages 66–80. Springer Berlin Heidelberg, 2013.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28:1364–1396, March 1999.

[HM02] Alejandro Hevia and Daniele Micciancio. The provable security of graph-based one-time signatures and extensions to algebraic signature schemes. In Yuliang Zheng, editor, *Advances in Cryptology — ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 191–196. Springer Berlin / Heidelberg, 2002.

[Len04] Arjen K. Lenstra. Key lengths. Contribution to The Handbook of Information Security, 2004.

[MMO85] Stephen Matyas, Carl Meyer, and Jonathan Oseas. Generating strong one-way functions with cryptographic algorithms. In *IBM Technical Disclosure Bulletin 27*, pages 5658–5659. IBM, 1985.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC '90: Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 387–394, New York, NY, USA, 1990. ACM Press.