On the Security of the Winternitz One-Time Signature Scheme

Full version[*]

Johannes Buchmann, Erik Dahmen, Sarah Ereth, Andreas Hülsing[**], and Markus Rückert[***]
{buchmann,dahmen,huelsing,rueckert}@cdc.informatik.tu-darmstadt.de
ereth@mais.informatik.tu-darmstadt.de

Technische Universität Darmstadt

Abstract. We show that the Winternitz one-time signature scheme is existentially unforgeable under adaptive chosen message attacks when instantiated with a family of pseudorandom functions. Our result halves the signature size at the same security level, compared to previous results, which require a collision resistant hash function. We also consider security in the strong sense and show that the Winternitz one-time signature scheme is strongly unforgeable assuming additional properties of the pseudorandom function family. In this context we formally define several key-based security notions for function families and investigate their relation to pseudorandomness. All our reductions are exact and in the standard model and can directly be used to estimate the output length of the hash function required to meet a certain security level.

Keywords Hash-based signatures, post-quantum signatures, pseudorandom functions, security reductions.

1 Introduction

Digital signatures are ubiquitous in our computer dominated society. They are basic building blocks of eGovernment and eCommerce. They are used to guarantee the integrity and authenticity of software updates and enable secure Internet connections. The security of currently used signature schemes – RSA and ECDSA – relies on the hardness of certain number theoretic problems, whereas the actual hardness of these problems remains unclear. In 1994 Shor presented a quantum algorithm that can be used to solve the factorization and discrete logarithm problems in polynomial time, thus completely breaking RSA and ECDSA [32]. Given the importance of digital signatures, the search for alternative signature schemes that resist attacks arising from algorithmic and technological advances is an important research goal.

One promising alternative are hash-based signatures. Their sole security requirement is the existence of hash function families with certain properties. Current research suggests, that the security of hash-based signatures will not be significantly harmed by quantum computer supported attacks [19]. Another benefit of hash-based signature schemes is that they are provably secure in the standard model [13,14,15,21]. A hash-based signature scheme or Merkle signature scheme (MSS) consists of the combination of a one-time signature scheme (OTS) to sign the data and Merkle's tree authentication scheme [25] which reduces the authenticity of many one-time verification keys to the authenticity of a single public key. Examples for one-time signature schemes are the Lamport-Diffie OTS [22], the Merkle OTS [25], the Winternitz OTS [25,15], the Bleichenbacher-Maurer OTS [6],

[*] An extended abstract of this work appeared as [8]

[**] Supported by grant no. BU 630/19-1 of the German Research Foundation (www.dfg.de).

[***] Supported by CASED (www.cased.de).

the BiBa OTS [27] and HORS [29]. The Winternitz OTS (W-OTS) is most suitable for combining it with Merkle's tree authentication scheme because of the small verification key size and the flexible trade-off between signature size and signature generation time. Further it is possible to compute the corresponding verification key given a W-OTS signature. So a MSS signature does not need to contain the verification key. This is not the case for all of the above mentioned schemes besides the Bleichenbacher-Maurer OTS but it reduces the MSS signature size significantly. Hence efficient variants of the Merkle signature scheme rely on W-OTS [10]. W-OTS is also used for authentication in sensor networks [24].

The size of a Winternitz signature is roughly $mn/\log_2 w$ bits and signing roughly requires $wm/\log_2 w$ hash operations, where m is the bit length of the hash value to be signed, n is the output length of the hash function used in the scheme, and w is the Winternitz parameter determining the trade-off between signature size and signature generation time. In [15,21], the authors provide security reductions for graph based one-time signature schemes, a general class of OTS which includes W-OTS. Due to the generality of graph based OTS, these security reductions require the used hash function to be collision resistant. Collision resistance is one of the strongest security notions of hash functions and admits effective generic attacks using the birthday paradox. Following these reductions, to achieve b bits of security one must use $n = 2b$ and $m = 2b$ which yields W-OTS signatures of size roughly $4b^2/\log_2 w$ bits.

Our results. In this paper we show that weaker assumptions are sufficient for the security of W-OTS. We show that W-OTS is existentially unforgeable under adaptive chosen message attacks [18] when instantiated with a family of pseudorandom functions (PRF). Since the PRF property is not affected by birthday attacks, hash functions with shorter output length can be used which in turn leads 50 % smaller signatures at the same security level, compared to [15]. This result is especially meaningful because the main issue with hash-based signatures is the signature size. Also, it has been shown that PRF exist if one way functions (OWF) exist [20,17] and further, that OWF exist if secure digital signature schemes exist [31]. So our result shows that a secure instance of W-OTS exists, as long as there exists any secure signature scheme. For collision resistant hash function families it is unknown if their existence can be based on the existence of OWF. In a companion work [9] we use this result to introduce a new variant of a MSS, solely based on the assumption that OWF exist, that provides the actually shortest signature size of all such MSS and is forward secure.

We also consider unforgeability in the strong sense by reducing the strong unforgeability of W-OTS to the intractability of finding *key collisions* (given x, find k, k' such that $k \neq k'$ and $f_k(x) = f_{k'}(x)$) or *second keys* (given x and key k, find k' such that $k \neq k'$ and $f_k(x) = f_{k'}(x)$). The notion of key-collision resistance was used before by the authors of [28] in the security analysis of the TESLA protocol. In [16], the author uses this notion as property of pseudorandom function tribe ensembles to construct a committing and key-hiding private-key encryption scheme. The authors of [12] provide a construction for perfectly one-way functions assuming key-collision resistance. We provide a thorough treatment of these key based notions and pseudorandomness. We define them formally and investigate implications and separations among them.

Our results are exact and in the standard model. Such reductions are of enormous practical value compared to asymptotic results or the random oracle model. Exact reductions allow the security level of the scheme to be estimated for fixed security parameters. The standard model uses only security notions which can be efficiently realized in practice. Exact reductions are also of theoretical interest, because they indicate the quality of a reduction and allow an easy comparison of the hardness of the problems.

Notation. Throughout the paper we stick to the following notation. We use n as the main security parameter. Efficient algorithms require only polynomial time and space in n. The statement $x \xleftarrow{\$} X$ means x is chosen uniformly at random from X. The concatenation of strings is done via $||$. We also write log for $\log_2$. During the paper we measure the runtime of an algorithm in terms of the number of evaluations of the function family used.

Organization. We prove the existential unforgeability of W-OTS using pseudorandom functions in Section 2. We prove the strong unforgeability of W-OTS using second-key resistant or key-collision resistant functions in Section 3. We examine implications and separations between the introduced security notions in Section 4. We discuss some implementation related issues in Section 5. We interpret our results and provide concluding remarks in Section 6.

2 Existential unforgeability of the Winternitz one-time signature scheme

In this section we prove that the Winternitz one-time signature scheme (W-OTS) is existentially unforgeable under adaptive chosen message attacks (EU-CMA) when instantiated with a family of pseudo-random functions. We begin by reviewing W-OTS and introduce a little tweak required by the reduction. Then we introduce the required security notions. Finally we state the reduction and use it to estimate the security level.

2.1 The Winternitz one-time signature scheme

The Winternitz one-time signature scheme was first mentioned in [25] as a generalization of Merkle's OTS also proposed in [25]. A complete description can be found in [15]. The core idea of W-OTS is to iteratively apply a function on a secret input, whereas the number of iterations depends on the message to be signed. The used functions are members of the function family

$$F(n) = \{f_k : \{0,1\}^n \to \{0,1\}^n | k \in \{0,1\}^n\} \tag{1}$$

parameterized by a key $k \in \{0,1\}^n$ and a security parameter n. For our purposes iteratively applying a function is defined as follows. We use the output of the function f_k as *key* for the next iteration. The function is always evaluated on the same input x. This is in contrast to the original construction, where the output of the function is used as input for the next iteration and the key remains fixed. We use the notation $f_k^i(x)$ to denote that the function is iterated i times on input x using key k for the first iteration and the output of the function as key for the next iteration, e.g. $f_k^2(x) = f_{f_k(x)}(x)$ and $f_k^0(x) = x$.

In the following, we only describe the generation of signatures for m-bit messages. The generalization to arbitrary sized messages is straightforward by utilizing a collision resistant hash function.

Key pair generation (Algorithm Kg). First we choose the Winternitz parameter $w \in \mathbb{N}, w > 1$, defining the compression level. Next we choose a random value $x \xleftarrow{\$} \{0,1\}^n$. The signature key consists of ℓ bit strings of length n chosen uniformly with the random distribution,

$$\mathsf{sk} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell) \xleftarrow{\$} \{0,1\}^{(n,\ell)},$$

where ℓ is computed as follows.

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

The verification key is computed using functions from the family $F(n)$. The bit strings in the signature key are used as key for the function f and the function is iterated $w-1$ times on input x.

$$\mathsf{pk} = (\mathsf{pk}_0, \mathsf{pk}_1, \ldots, \mathsf{pk}_\ell) = (x, f_{\mathsf{sk}_1}^{w-1}(x), \ldots, f_{\mathsf{sk}_\ell}^{w-1}(x))$$

Signature generation (Algorithm Sign.) We describe how to sign an m-bit message $M = (M_1, \ldots, M_{\ell_1})$ given in base-w representation, i.e. $M_i \in \{0, \ldots, w-1\}$ for $i = 1, \ldots, \ell_1$. We begin by computing the checksum

$$C = \sum_{i=1}^{\ell_1} (w - 1 - M_i) \tag{2}$$

and represent it to base w as $C = (C_1, \ldots, C_{\ell_2})$. The length of the base-w representation of C is at most ℓ_2 since $C \leq \ell_1(w-1)$. Then we set $B = (b_1, \ldots, b_\ell) = M \parallel C$. The signature of message M is computed as

$$\sigma = (\sigma_1, \ldots, \sigma_\ell) = (f_{\mathsf{sk}_1}^{b_1}(x), \ldots, f_{\mathsf{sk}_\ell}^{b_\ell}(x)). \tag{3}$$

Signature verification (Algorithm Vf.) The verifier first computes the base-w string $B = (b_1, \ldots, b_\ell)$ as described above. Then he checks whether

$$(f_{\sigma_1}^{w-1-b_1}(\mathsf{pk}_0), \ldots, f_{\sigma_\ell}^{w-1-b_\ell}(\mathsf{pk}_0)) \stackrel{?}{=} (\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell). \tag{4}$$

The signature is accepted iff the comparison holds.

2.2 Security notions for signature schemes and function families

We begin by reviewing the standard definition of digital signature schemes and the security notion existential unforgeability under adaptive chosen message attacks (EU-CMA) [18]. We then define two security notions for function families required for our reduction. The first is the well known pseudo-randomness property. The second is *key one-wayness* which states that it is hard to find a key k such that the function f_k maps a given input x to a given output y. We also state two lemmas about these notions which will be useful for the reduction of W-OTS.

Definition 1 (Digital signature schemes). *A digital signature scheme* $\mathsf{Sig} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$ *is a triple of PPT 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 the message* M*;*
- $\mathsf{Vf}(\mathsf{pk}, \sigma, M)$ *outputs 1 iff* σ *is a valid signature on* M *under* pk.

Definition 2 (Existential unforgeability (EU-CMA)). *Let* $\mathsf{Sig} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$ *be a digital signature scheme. The EU-CMA security notion is defined by the following experiment.*

Experiment $\mathsf{Exp}_{\mathsf{A},\mathsf{Sig}}^{\mathsf{EU\text{-}CMA}}(n)$

```
    $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Kg}(1^n)$
    $(M^\star, \sigma^\star) \leftarrow \mathsf{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$
    Let $\{(M_i, \sigma_i)\}_1^{q_{\mathsf{Sign}}}$ be the query-answer pairs of $\mathsf{Sign}(\mathsf{sk}, \cdot)$.
    Return 1 iff $\mathsf{Vf}(\mathsf{pk}, M^\star, \sigma^\star) = 1$ and $M^\star \notin \{M_i\}_1^{q_{\mathsf{Sign}}}$.
```

The signature scheme Sig *is* (t, ϵ, q)*-existentially unforgeable if there is no* t*-time adversary that succeeds with probability* $\geq \epsilon$ *after making* $\leq q$ *signature oracle queries.*

A $(t, \epsilon, 1)$-EU-CMA secure signature scheme is called a one-time signature scheme.

Definition 3 (Pseudorandom functions (PRF)). *A family of functions* $F(n)$ *is pseudorandom, if no efficient algorithm* Dis *is able to distinguish a randomly chosen function* $f_k \in F(n)$ *from a randomly chosen function from the set* $G(n)$ *of all functions with same domain and range as* $F(n)$*. The formal definition is taken from [5]. The distinguisher* Dis *gets access to an oracle* $\mathsf{Box}(\cdot)$ *implementing a function randomly chosen from* $F(n)$ *or* $G(n)$ *in a black-box manner. The distinguisher may adaptively query* $\mathsf{Box}(\cdot)$ *as often as he likes. Finally, the distinguisher outputs 1 if he thinks that* Box *models a function from* $F(n)$ *and 0 otherwise.*

Let $F(n)$ *be a family of functions as in (1) and* $G(n) = \{g : \{0,1\}^n \rightarrow \{0,1\}^n\}$ *the family of all functions with domain and range* $\{0,1\}^n$*. We call* $F(n)$ (t, ϵ)*-PRF, if the advantage*

$$\mathrm{Adv}_{F(n)}^{\mathrm{PRF}}(\mathsf{Dis}) = \left|\Pr[\mathsf{Box} \xleftarrow{\$} F(n) : \mathsf{Dis}^{\mathsf{Box}(\cdot)} = 1] - \Pr[\mathsf{Box} \xleftarrow{\$} G(n) : \mathsf{Dis}^{\mathsf{Box}(\cdot)} = 1]\right| \tag{5}$$

of any distinguisher Dis *that runs in time* t *is at most* ϵ*.*

Definition 4 (Key one-wayness (KOW)). *Let* $F(n)$ *be a family of functions as in (1). We call* $F(n)$ (t, ϵ)*-KOW, if the success probability*

$$\mathrm{Adv}_{\mathsf{A}}^{\mathrm{KOW}} = \Pr[(x, k) \xleftarrow{\$} \{0,1\}^n \times \{0,1\}^n, y \leftarrow f_k(x), k' \longleftarrow \mathsf{A}(x, y) : y = f_{k'}(x)] \tag{6}$$

of any adversary A *that runs in time* t *is at most* ϵ*.*

Please recall, that the time t is counted in terms of evaluations of f. We assume, that a call to Box takes the same time as an evaluation of f. The *security level* or *bit security* b the family $F(n)$ or a signature scheme Sig provides against attacks on the respective notion is computed as $b = \log(t/\epsilon)$.

A key collision of a function family $F(n)$ is defined as a pair of distinct keys (k, k') such that $f_k(x) = f_{k'}(x)$ holds for some $x \in \{0,1\}^n$. In our proofs we make use of an upper (κ) and a lower (κ') bound on the number of key collisions that occur in the family $F(n)$. We define these bounds as follows:

Definition 5. *Let* $F(n)$ *be a family of functions as in (1). We define the upper bound on the number of key collisions in* $F(n)$ *as the maximum number of keys that map the same input value to the same output value:*

$$\kappa(F(n)) = \max_{K \subseteq \{0,1\}^n} \{|K| \mid (\exists x \in \{0,1\}^n), (\forall k_1, k_2 \in K) : f_{k_1}(x) = f_{k_2}(x)\}.$$

We define the lower bound on the number of key collisions in $F(n)$ *accordingly as*

$$\kappa'(F(n)) = \min_{K \subseteq \{0,1\}^n} \{|K| \mid (\exists x \in \{0,1\}^n), (\forall k_1, k_2 \in K) : f_{k_1}(x) = f_{k_2}(x)\}.$$

We write κ (κ') instead of $\kappa(F(n))$ ($\kappa'(F(n))$) where $F(n)$ is clear from the context. The values κ and κ' restrict the number of different images y some preimage x can be mapped to by functions in $F(n)$, i.e.

$$\frac{2^n}{\kappa} \leq \big| \{ f_k(x) : k \in \{0,1\}^n \} \big| \leq \frac{2^n}{\kappa'} \tag{7}$$

for all $x \in \{0,1\}^n$. Also, given $y \xleftarrow{\$} \{0,1\}^n$ the probability that there exists a key k and preimage x such that $f_k(x) = y$ holds is at least $1/\kappa$.

The following lemma describes an interesting relation between the security level of pseudorandom functions and the value κ defined above.

Lemma 1. *Let $F(n)$ be (t, ϵ)-PRF with security level $b = \log(t/\epsilon)$ and $\kappa(F(n))$ as in Definition 5. Then $\kappa(F(n)) \leq 2^{n-b} + 1$.*

Proof. Assume $\kappa > 2^{n-b} + 1$ and let (x, y) be a pair where there exist κ keys mapping x to y. The distinguisher Dis queries Box with x. If $\mathsf{Box}(x) = y$ then Dis returns 1 and 0 otherwise. Clearly Dis runs in time $t' = 1$. Further we have $\Pr[\mathsf{Box} \xleftarrow{\$} F(n) : \mathsf{Dis}^{\mathsf{Box}(\cdot)} = 1] = \kappa/2^n > 2^{-b} + 2^{-n}$ and $\Pr[\mathsf{Box} \xleftarrow{\$} G : \mathsf{Dis}^{\mathsf{Box}(\cdot)} = 1] = 2^{-n}$ and therefore $\epsilon' = \mathrm{Adv}^{\mathrm{PRF}}_{F(n)}(\mathsf{Dis}) > 2^{-b}$ which is a contradiction. □

Following the definition of κ and κ', $\kappa' \geq 1$ always holds. The above lemma implies that for a good pseudorandom function family, i.e. a pseudorandom function family with $b = n$ bit security, $\kappa = 2$.

The following lemma states that the KOW property is implied by the PRF property. In other words, an efficient attacker against the KOW property leads to an efficient distinguisher.

Proposition 1 (PRF $\Rightarrow$ KOW)**.** *Let $F(n)$ be (t, ϵ)-PRF. Then $F(n)$ is $(t-2, \epsilon/(1/\kappa(F(n)) - 1/2^n))$ - KOW.*

Proof. Assume there exists an adversary $\mathsf{A}_{\mathrm{KOW}}(x, y)$ who finds a key k satisfying $y = f_k(x)$ in time t_{KOW} with probability ϵ_{KOW}. Then we can construct a distinguisher Dis using $\mathsf{A}_{\mathrm{KOW}}$ the following way: Dis queries $\mathsf{Box}(\cdot)$ with $x \in \{0,1\}^n$. After receiving the answer y, Dis runs $\mathsf{A}_{\mathrm{KOW}}(x, y)$ to obtain key k. Then Dis queries Box with a second value $x' \in \{0,1\}^n$. If $\mathsf{Box}(x') = f_k(x') = y'$ Dis returns 1 and 0 otherwise. In case $\mathsf{Box} \xleftarrow{\$} F(n)$, the probability that $\mathsf{A}_{\mathrm{KOW}}$ outputs a key k such that $f_k(x) = y$ holds is ϵ_{KOW}. The probability that $f_k(x') = y'$ holds is at least $1/\kappa$, because at least one of the κ functions in $F(n)$ mapping x to y also maps x' to y'. In case $\mathsf{Box} \xleftarrow{\$} G(n)$, the probability that $\mathsf{A}_{\mathrm{KOW}}$ outputs a key k such that $f_k(x) = y$ holds is at most ϵ_{KOW}. The probability that $f_k(x') = y'$ holds is $1/2^n$, because from the $2^{n(2^n-1)}$ functions in G mapping x to y, only $2^{n(2^n-2)}$ also map x' to y'. In summary we get $\epsilon \geq \mathrm{Adv}^{\mathrm{PRF}}_{F(n)}(\mathsf{Dis}) \geq \epsilon_{\mathrm{KOW}} \left(1/\kappa - 1/2^n\right)$. □

2.3 Security reduction

We now state the main result of this section.

Theorem 1. *Let $F(n)$ be a family of functions as in Equation (1). If $F(n)$ is $(t_{\mathrm{PRF}}, \epsilon_{\mathrm{PRF}})$-PRF then W-OTS is $(t, \epsilon, 1)$ EU-CMA with*

$$t = t_{\mathrm{PRF}} - t_{\mathsf{Kg}} - t_{\mathsf{Vf}} - 2 \tag{8}$$

$$\epsilon \leq \epsilon_{\mathrm{PRF}} \ell^2 w^2 \kappa(F(n))^{w-1} \frac{1}{\left(\frac{1}{\kappa(F(n))} - \frac{1}{2^n}\right)} \tag{9}$$

where t_{Kg} and t_{Vf} denote the runtime of the W-OTS key generation and verification algorithms, respectively.

Proof. The proof works as follows: First we use a forger for W-OTS to construct an adversary on the key one-wayness of $F(n)$. This adversary is then used to construct a distinguisher using Proposition 1. Algorithm 1 shows how a forger $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$ for W-OTS can be used to construct an adversary $\mathsf{A}_{\mathrm{KOW}}$ on the key one-wayness of $F(n)$. The signing oracle Sign is simulated by the adversary.

Algorithm 1 $\mathsf{A}_{\mathrm{KOW}}$

Input: Security parameters n, m, Winternitz parameter w, description of $F(n)$, KOW challenge (x, y) as in Definition 4
Output: k', such that $f_{k'}(x) = y$ or fail

```
1. generate W-OTS signature key $\mathsf{sk}$
2. choose indices $\alpha \in \{1, ..., \ell\}, \beta \in \{1, \dots, w-1\}$ uniformly at random
3. compute verification key as $\mathsf{pk}_0 = x$, $\mathsf{pk}_i = f^{w-1}_{\mathsf{sk}_i}(x)$ for $i = 1, \dots, l, i \neq \alpha$ and $\mathsf{pk}_\alpha = f^{w-1-\beta}_y(x)$
4. run $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$
5. when $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$ queries $\mathsf{Sign}$ with message $M$ then compute $B = (b_1, ..., b_\ell)$
6. if $b_\alpha < \beta$ return fail
7. generate signature $\sigma$ of $M$ as $\sigma_i = f^{b_i}_{\mathsf{sk}_i}(x)$ for $i = 1, \dots, \ell, i \neq \alpha$ and $\sigma_\alpha = f^{b_\alpha - \beta}_y(x)$
8. send $\sigma$ to $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$
9. when $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$ returns valid $(\sigma', M')$ then compute $B' = (b'_1, ..., b'_\ell)$
10. if $b'_\alpha \geq \beta$ return fail
11. compute $k' \leftarrow f^{\beta-1-b'_\alpha}_{\sigma'_\alpha}(x)$
12. if $f_{k'}(x) \neq y$ return fail
13. return $k'$
```

The goal of the adversary $\mathsf{A}_{\mathrm{KOW}}$ is to produce a key k' such that $f_{k'}(x) = y$ for x, y provided as input. $\mathsf{A}_{\mathrm{KOW}}$ begins by generating a regular W-OTS signature key pair and choosing random positions α and β (Lines 1,2). Then he computes the W-OTS verification key using value x. The bit string at position α in the verification key (pk_α) is computed by inserting y at position β in the hash chain used to compute pk_α (Line 3). Next, $\mathsf{A}_{\mathrm{KOW}}$ calls the forger and waits for it to ask an oracle query. The forgers query can only be answered if $b_\alpha \geq \beta$ holds, because $\mathsf{A}_{\mathrm{KOW}}$ doesn't know the first β entries in the corresponding hash chain (Lines 6, 7). The signature is computed as in the real scheme with one difference. For the αth hash-chain only $b_\alpha - \beta$ iterations are computed, as we placed y at position β in this chain (Line 7). The forgery produced by the forger is only meaningful to $\mathsf{A}_{\mathrm{KOW}}$ if $b'_\alpha < \beta$ holds (Line 10). Only then the bit string σ_α in the forged signature might yield a key k' such that $y = f_{k'}(x)$ holds (Lines 11,12). We now compute the success probability of $\mathsf{A}_{\mathrm{KOW}}$. W.l.o.g we assume that the forger queries the signing oracle. The probability of $b_\alpha \geq \beta$ in Line 6 is at least $(\ell w)^{-1}$. This is because of the checksum which guarantees that not all of the b_i are zero simultaneously. The probability that the forger succeeds in Line 9 is at least ϵ by definition. This probability holds under the condition that the verification key pk computed in Line 3 resembles a regular verification key which is the case if there exists a key k such that $f_k^{\beta}(x) = y$. This happens with probability at least $1/\kappa^{\beta}$ according to Definition 5. The probability of $b'_{\alpha} < \beta$ in Line 10 is at least $(\ell w)^{-1}$. This is because of $M \neq M'$ and the checksum which guarantees that $b_i > b'_i$ for some $i \in \{1, \dots, \ell\}$. The probability that $y = f_{k'}(x)$ holds in Line 12 is at least $1/\kappa^{w-1-\beta}$. This is because there exist at most $\kappa^{w-1-\beta}$ different values that are mapped to pk_{α} after $w - 1 - \beta$ iterations and at least one of them is y.

In summary we have $\epsilon_{\text{KOW}} \geq \epsilon/(\ell^2 w^2 \kappa^{\beta} \kappa^{w-1-\beta})$ and $t_{\text{KOW}} = t + t_{\mathsf{Kg}} + t_{\mathsf{Vf}}$ as the time for the signature query is already taken into account at the runtime of the forger. Combining this with Proposition 1 yields $\epsilon_{\text{PRF}} \geq \epsilon(1/\kappa - 1/2^n)/(\ell^2 w^2 \kappa^{w-1})$ and $t_{\text{PRF}} = t + t_{\mathsf{Kg}} + t_{\mathsf{Vf}} + 2$ which concludes the proof. □

2.4 Security level

We now compute the security level of W-OTS for the case that only generic attacks against the PRF property of the function family $F(n)$ exist. This reflects the security of W-OTS, if the used function family $F(n)$ has no specific weaknesses. It corresponds to the security level defined in [23]. The best known generic attack against the pseudorandomness of $F(n)$ is a brute-force key recovery attack. As before, we count the running time of an algorithm as the number of evaluations of elements from $F(n)$. A simple counting argument gives that t_{Kg} and t_{Vf} are both bounded by ℓw evaluations of elements from $F(n)$. In the following corollary we use the bound $4/(\ell w 2^{w-2}) \leq 2^{n-w-1-2\log(\ell w)}$ which is fulfilled by most practical parameter sets. Anyhow, following the proof of the corollary one can easily compute the security level for any specific set of parameters.

Corollary 1. *Let $b = \log(t/\epsilon)$ denote the security level and use ℓw as upper bound for t_{Kg} and t_{Vf}, respectively. Let $F(n)$ be $(2^{n-1-\log\kappa(F(n))}, 1/2(1/\kappa(F(n)) - 1/2^n))$-PRF with $\kappa(F(n)) = 2$ and $4/(\ell w 2^{w-2}) \leq 2^{n-w-1-2\log(\ell w)}$. Then the security level of W-OTS under generic attacks is*

$$b \geq n - w - 1 - 2\log(\ell w) \tag{10}$$

Proof. We use a $(t_{\text{PRF}}, \epsilon_{\text{PRF}})$-PRF family $F(n)$ and assume that the best attack on the pseudorandomness of $F(n)$ is a brute-force key recovery attack. An attacker that searches through $t_{\text{KOW}} = 2^{n-1-\log\kappa}$ keys has success probability $\epsilon_{\text{KOW}} = 1/2$ for recovering the correct key. By Proposition 1 this yields an $t_{\text{PRF}} = 2^{n-1-\log\kappa} + 2, \epsilon_{\text{PRF}} = 1/2(1/\kappa - 1/2^n)$ distinguisher for the pseudorandomness of $F(n)$. The security level of the PRF property of $F(n)$ in presence of this distinguisher is $b = n$ which in turn implies $\kappa \leq 2$ according to Lemma 1. The security level of W-OTS using $F(n)$ is computed as follows

$$\begin{aligned} 2^b = \frac{t}{\epsilon} &\geq \frac{t_{\text{PRF}} - t_{\mathsf{Kg}} - t_{\mathsf{Vf}} - 2}{\epsilon_{\text{PRF}} \ell^2 w^2 \kappa^{w-1}} \left(\frac{1}{\kappa} - \frac{1}{2^n} \right) \\ &\geq \frac{2^{n-\log\kappa} - 4\ell w}{\ell^2 w^2 \kappa^{w-1}} \\ &\geq 2^{n-w-2\log(\ell w)} - \frac{4}{\ell w 2^{w-1}} \end{aligned}$$

Since $4/(\ell w 2^{w-2}) \leq 2^{n-w-1-2\log(\ell w)}$ per assumption we finally obtain $b \geq n - w - 1 - 2\log(\ell w)$ as security level of W-OTS. □

3 Strong unforgeability of the Winternitz one-time signature scheme

While the reduction of the last section shows that W-OTS is EU-CMA assuming a standard security notion for hash functions, it does not provide security in the strong sense. This is accomplished by two reductions presented in this section. We show that W-OTS is strongly unforgeable under adaptive chosen message attacks (SU-CMA), if the used function family is either *second-key resistant* or *key-collision resistant*. The difference between EU-CMA and SU-CMA is, that in SU-CMA the adversary also wins if he returns a new signature for an already queried message. SU-CMA secure signature schemes have a number of applications, including the construction of chosen-ciphertext secure encryption schemes [11], and group signatures [1,7]. While the reductions in this section provide stronger security guarantees, they do not rely on standard security notions of hash functions. One is therefore confronted with a trade-off between security and requirements on the hash function. Again we begin by introducing the required security notions and then continue with the reductions and the computation of the security levels.

3.1 Security notions for signature schemes and function families II

We begin by reviewing the definition of strong unforgeability under adaptive chosen message attacks. Then, we define two security notions for function families required for our reductions. The first is *second-key resistance* which states that given key k and preimage x, it is hard to find a key $k' \neq k$ such that $f_k(x) = f_{k'}(x)$. The second is *key-collision resistance* which states that given preimage x, it is hard to find two distinct keys k, k' such that $f_k(x) = f_{k'}(x)$.

Definition 6 (Strong unforgeability (SU-CMA)). *Let* $\mathsf{Sig} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$ *be a digital signature scheme. The SU-CMA security notion is defined by the following experiment.*

```
Experiment $\mathsf{Exp}^{\mathsf{SU\text{-}CMA}}_{\mathsf{A},\mathsf{Sig}}(n)$
    $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Kg}(1^n)$
    $(M^*, \sigma^*) \leftarrow \mathsf{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$
    Let $\{(M_i, \sigma_i)\}_1^{q_{\mathsf{Sign}}}$ be the query-answer pairs of $\mathsf{Sign}(\mathsf{sk}, \cdot)$.
    Return 1 iff $\mathsf{Vf}(\mathsf{pk}, M^\star, \sigma^\star) = 1$ and $(M^\star, \sigma^\star) \notin \{(M_i, \sigma_i)\}_1^{q_{\mathsf{Sign}}}$.
```

The signature scheme Sig *is* (t, ϵ, q)*-SU-CMA if there is no* t*-time adversary that succeeds with probability* $\geq \epsilon$ *after making* $\leq q$ *signature oracle queries.*

Definition 7 (Second-key resistance (SKR)). *Let* $F(n)$ *be a family of functions as in (1). We call* $F(n)$ (t, ϵ)-SKR*, if the success probability*

$$\mathrm{Adv}^{\mathrm{SKR}}_{\mathsf{A}} = \Pr[(x, k) \xleftarrow{\$} \{0,1\}^n \times \{0,1\}^n, k' \leftarrow \mathsf{A}(x, k) : k' \neq k, f_{k'}(x) = f_k(x)] \tag{11}$$

of any adversary A *that runs in time* t *is at most* ϵ*.*

Definition 8 (Key-collision resistance (KCR)). *Let* $F(n)$ *be a a family of functions as in (1). We call* $F(n)$ (t, ϵ)-KCR*, if the success probability*

$$\mathrm{Adv}^{\mathrm{KCR}}_{\mathsf{A}} = \Pr[x \xleftarrow{\$} \{0,1\}^n, (k, k') \leftarrow \mathsf{A}(x) : k \neq k', f_k(x) = f_{k'}(x)] \tag{12}$$

of any adversary A *that runs in time* t *is at most* ϵ*.*

Proposition 2 (SKR $\Rightarrow$ KOW)**.** *Let $F(n)$ be (t, ϵ)-SKR with $\kappa' > 1$. Then $F(n)$ is $(t-1, \epsilon/(1 - 1/\kappa'))$-KOW.*

Proof. Towards contradiction, let us assume a successful adversary A that breaks KOW for $F(n)$. We show how to use A as a black-box in an algorithm $\mathcal{B}$ to break SKR. On input (x, k) from the SKR experiment, the algorithm $\mathcal{B}$ computes $y \leftarrow f_k(x)$ and runs $\mathsf{A}(x, y)$. The subroutine returns k' such that $f_k(x) = f_{k'}(x)$ with probability at least ϵ. Then, $\mathcal{B}$ returns k'. Since $\kappa'(F(n)) > 1$, the algorithm A returns a key that is different from k with probability at least $1 - 1/\kappa' \geq 1/2$. Thus, $\mathcal{B}$ is successful with probability $\epsilon(1 - 1/\kappa')$. The condition $\kappa' > 1$ is required to guarantee that a different key actually exists. □

3.2 Security reductions

We now state the main result of this section.

Theorem 2. *Let $F(n)$ be a family of functions as in Equation (1). Denote by t_{Kg} and t_{Vf} the runtime of the W-OTS key generation and verification algorithms, respectively.*
a) If $F(n)$ is $(t_{\mathrm{SKR}}, \epsilon_{\mathrm{SKR}})$-SKR then W-OTS is $(t, \epsilon, 1)$ SU-CMA with

$$t \geq t_{\mathrm{SKR}} - t_{\mathsf{Kg}} - t_{\mathsf{Vf}} - 1 \tag{13}$$

$$\epsilon \leq \epsilon_{\mathrm{SKR}} \ell^2 w^2 \kappa(F(n))^{w-2} \frac{\kappa'(F(n))}{\kappa'(F(n)) - 1} \tag{14}$$

b) If $F(n)$ is $(t_{\mathrm{KCR}}, \epsilon_{\mathrm{KCR}})$-KCR then W-OTS is $(t, \epsilon, 1)$ SU-CMA with

$$t \geq t_{\mathrm{KCR}} - t_{\mathsf{Kg}} - t_{\mathsf{Vf}} \tag{15}$$

$$\epsilon \leq \epsilon_{\mathrm{KCR}} \frac{\kappa'(F(n))}{\kappa'(F(n)) - 1} \tag{16}$$

Proof. To prove part a) we show in Algorithm 2 how a forger $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$ for W-OTS can be used to construct an adversary $\mathsf{A}_{\mathrm{SKR}}$ on the second-key resistance of $F(n)$ with non negligible advantage. The signing oracle Sign is simulated by the adversary.

On input of a challenge key k_c and value x_c, $\mathsf{A}_{\mathrm{SKR}}$ first generates a W-OTS key pair using $x = x_c$. Then he places k_c randomly in the hash chain used to compute pk_α at position β and runs $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$ on input $\mathsf{pk} = (x_c, \mathsf{pk}_1, \ldots, \mathsf{pk}_\ell)$. If the forger queries the oracle on message M, $\mathsf{A}_{\mathrm{SKR}}$ can only answer the query if $b_\alpha \geq \beta$, because $\mathsf{A}_{\mathrm{SKR}}$ can only compute the values of the αth hash chain starting from k_c. In this case the adversary returns (M, σ) otherwise he returns fail. If the forger succeeds in computing a valid signature (M', σ') there are two possible cases.

If For_{W-OTS} returns a signature for the message sent to the oracle, there has to be at least one index i such that $\sigma_i \neq \sigma'_i$ because $\sigma \neq \sigma'$. $\mathsf{A}_{\mathrm{SKR}}$ only found a second key if (1) α is one of these indices, (2) $b_\alpha = \beta$ and (3) $f_{k_c}(x_c) = f_{\sigma'_\alpha}(x_c)$. Observe that (1) implies that $k_c \neq \sigma'_\alpha$ and therefore σ'_α is a second key for the challenge (k_c, x_c). So $\mathsf{A}_{\mathrm{SKR}}$ returns σ'_α.

If the forger returns a signature for a new message the adversary can only find a second key for the challenge k_c if (1a) $b'_\alpha < \beta$ or (1b) $b'_\alpha = \beta$ and $b_\alpha > \beta$, if (2) $f_{k_c}(x_c) = f_{\sigma'_\alpha}^{\beta - b'_\alpha + 1}(x_c)$ holds and last but not least if $k_c \neq f_{\sigma'_\alpha}^{\beta - b'_\alpha}(x_c)$. If all of these conditions are fulfilled $\mathsf{A}_{\mathrm{SKR}}$ returns $f_{\sigma'_\alpha}^{\beta - b'_\alpha}(x_c)$ as second key. Otherwise $\mathsf{A}_{\mathrm{SKR}}$ returns fail.

Algorithm 2 A_{SKR}

Input: Security parameters n, m, Winternitz parameter w, description of $F(n)$, SKR challenge (x_c, k_c) ad in Definition 7

Output: $k' : f_{k_c}(x_c) = f_{k'}(x_c)$ or fail

```
1. generate W-OTS key pair $(\mathsf{sk}, \mathsf{pk})$ using $x = x_c$
2. choose indices $\alpha \in \{1, ..., \ell\}, \beta \in \{0, \dots, w-2\}$ uniformly at random
3. replace $\mathsf{pk}_\alpha$ with $f_{k_c}^{w-1-\beta}(x_c)$
4. run $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$
5. **when** $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$ queries $\mathsf{Sign}$ with message $M$ **then** compute $B = (b_1, ..., b_\ell)$
6. **if** $b_\alpha < \beta$ **return** $\mathsf{fail}$
7. generate signature $\sigma$ of $M$ as $\sigma_i = f_{\mathsf{sk}_i}^{b_i}(x)$ for $i = 1, \dots, \ell, i \neq \alpha$ and $\sigma_\alpha = f_{k_c}^{b_\alpha - \beta}(x)$
8. send $\sigma$ to $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$
9. **when** $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$ returns valid $(\sigma', M')$ **then** compute $B' = (b'_1, ..., b'_\ell)$
10. **if** $M = M'$ **and** $\sigma_\alpha \neq \sigma'_\alpha$ **and** $b_\alpha = \beta$ **and** $f_{k_c}(x_c) = f_{\sigma'_\alpha}(x_c)$ **then return** $\sigma'_\alpha$
11. **elseif** $M \neq M'$ **and** $(b'_\alpha < \beta$ **or** $(b'_\alpha = \beta$ **and** $b_\alpha > \beta))$ **and** $f_{k_c}(x_c) = f_{\sigma'_\alpha}^{\beta - b'_\alpha + 1}(x_c)$ **and** $k_c \neq f_{\sigma'_\alpha}^{\beta - b'_\alpha}(x_c)$ **then**
    **return** $f_{\sigma'_\alpha}^{\beta - b'_\alpha}(x_c)$
12. **return** $\mathsf{fail}$
```

We now compute the success probability. Like in the proof of Theorem 1 a), pk is a possible public key only with probability at least $1/\kappa^\beta$. As before we assume that $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$ queries the oracle without loss of generality. So the probability of $b_\alpha \geq \beta$ in line 6 is at least $(\ell w)^{-1}$. This is caused by the checksum which guarantees that at least one b_i is greater zero. The forger succeeds with probability at least ϵ according to the definition. Then we've got two mutual exclusive cases.

Case 1 ($M = M'$): The probability that $\sigma_\alpha \neq \sigma'_\alpha$ holds in line 10 is at least $1/\ell$ as there has to be at least one index i such that $\sigma_i \neq \sigma'_i$ because $\sigma \neq \sigma'$. For the second condition $b_\alpha = \beta$ we get a probability of $1/w$ as β was chosen at random. If the second condition holds, the last condition $f_{k_c}(x_c) = f_{\sigma'_\alpha}(x_c)$ holds at least with probability $\frac{1}{\kappa^{w-1-\beta-1}}$, because there are $\left|\left\{y \in \{0,1\}^n | f_y^{w-1-\beta-1}(x_c) = \mathsf{pk}_i\right\}\right| = \kappa^{w-1-\beta-1}$ possible values for $f_{\sigma'_\alpha}(x_c)$. So altogether we get a probability of at least $\frac{\epsilon}{\ell^2 w^2 \kappa^{w-2}}$ for finding a second key for this case.

Case 2 ($M \neq M'$): The probability in Line 11 that $(b'_\alpha < \beta$ **or** $(b'_\alpha = \beta$ **and** $b_\alpha > \beta))$ holds is greater than $(\ell w)^{-1}$. This is because of $M \neq M'$ and the checksum which guarantees that $b_i > b'_i$ for some $i \in \{1, \dots, \ell\}$. Next $f_{k_c}(x_c) = f_{\sigma'_\alpha}^{\beta - b'_\alpha + 1}(x_c)$ holds at least with probability $\frac{1}{\kappa^{w-1-\beta-1}}$ as before. And at last $k_c \neq f_{\sigma'_\alpha}^{\beta - b'_\alpha}(x_c)$ holds with probability at least $\frac{\kappa' - 1}{\kappa'}$ if the previous condition already holds. Therefore we get a success probability of $\frac{\epsilon(\kappa'-1)}{\ell^2 w^2 \kappa^{w-2} \kappa'}$ for case 2.

Since both cases are mutually exclusive, the success probability of A_{SKR} is

$$\epsilon_{\text{SKR}} \geq \min\{\frac{\epsilon}{\ell^2 w^2 \kappa^{w-2}}, \frac{\epsilon(\kappa'-1)}{\ell^2 w^2 \kappa^{w-2} \kappa'}\} = \frac{\epsilon(\kappa'-1)}{\ell^2 w^2 \kappa^{w-2} \kappa'}.$$

The time required by A_{SKR} is $t_{\text{SKR}} \leq t + t_{\mathsf{Kg}} + t_{\mathsf{Vf}}$. The time to answer the signature query is already contained in the runtime of the forger. This concludes the proof of part a).

To prove part b) we show in Algorithm 3 how a forger $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$ for W-OTS can be used to construct an adversary A_{KCR} on the key-collision resistance with non negligible advantage. Again, the signing oracle Sign is simulated by the adversary.

The goal of adversary A_{KCR} on input x_c is to find two different keys k_1, k_2 for which $f_{k_1}(x_c) = f_{k_2}(x_c)$ holds. Therefore A_{KCR} begins by generating a W-OTS key pair using $(x = x_c)$ and calling

Algorithm 3 $\mathsf{A}_{\mathrm{KCR}}$

Input: Security parameter n, Winternitz parameter w, description of $F(n)$, KCR challenge x_c as in Definition 8
Output: (k_1,k_2): $k_1 \neq k_2$ and $f_{k_1}(x_c) = f_{k_2}(x_c)$ or fail

```
1. generate W-OTS key pair $(\mathsf{sk}, \mathsf{pk})$ using $x = x_c$
2. run $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$
3. when $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$ queries Sign with message $M$ then
     generate signature $\sigma$ of $M$ as $\sigma_i = f^{b_i}_{\mathsf{sk}_i}(x_c)$ for $i = 1, \ldots, \ell$ and respond to $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$
4. when $\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$ returns valid $(\sigma', M')$ then compute $B' = (b'_1, ..., b'_\ell)$.
5. if there exists an index $i$ such that $f^{b'_i}_{\mathsf{sk}_i}(x_c) \neq \sigma'_i$ then
     compute the smallest index $j \geq 1$ such that
     $f^{b'_i+j}_{\mathsf{sk}_i}(x_c) = f^{j}_{\sigma'_i}(x_c)$ and $f^{b'_i+j-1}_{\mathsf{sk}_i}(x_c) \neq f^{j-1}_{\sigma'_i}(x_c)$
     return $(f^{b'_i+j-1}_{\mathsf{sk}_i}(x_c), f^{j-1}_{\sigma'_i}(x_c))$
6. else return fail
```

$\mathsf{For}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$ with the generated public key. If the forger queries the signing oracle Sign for the signature of a message M, $\mathsf{A}_{\mathrm{KCR}}$ answers this query with $\sigma = \mathsf{Sign}(\mathsf{sk}, M)$. If the forger succeeds in generating a signature (M', σ'), $\mathsf{A}_{\mathrm{KCR}}$ computes the vector $B' = (b'_i), 1 \leq i \leq l$ using M' as in the signing algorithm of W-OTS. Then he checks if there exists an index i with $f^{b'_i}_{\mathsf{sk}_i}(x_c) \neq \sigma'_i$. As $f^{w-1-b'_i}_{f^{b'_i}_{\mathsf{sk}_i}}(x_c) = \mathsf{pk}_i = f^{w-1-b'_i}_{\sigma'_i}(x_c)$ there must exist one $j, 0 \leq j \leq w-1-b'_i$ with $f^{b'_i+j}_{\mathsf{sk}_i}(x_c) = f^{j}_{\sigma'_i}(x_c)$ and $f^{b'_i+j-1}_{\mathsf{sk}_i}(x_c) \neq f^{j-1}_{\sigma'_i}(x_c)$. So $\mathsf{A}_{\mathrm{KCR}}$ returns the key collision $(f^{b'_i+j-1}_{\mathsf{sk}_i}(x_c), f^{j-1}_{\sigma'_i}(x_c))$.

We now compute the success probability. The forger returns a valid signature with probability ϵ per definition. Then we have two alternative cases:

If $M' = M$ the forger returned a different signature for the message M signed by Sign. In this case the forger returned a key collision with probability 1. As $\sigma \neq \sigma'$ there has to be at least one index i with $\sigma_i \neq \sigma'_i$ what implies $f^{b'_i}_{\mathsf{sk}_i}(x_c) \neq \sigma'_i$.

If $M' \neq M$ the probability that we find an index i with $f^{b'_i}_{\mathsf{sk}_i}(x_c) \neq \sigma'_i$ and therefore a key collision is at least $(\kappa' - 1)/\kappa'$. If the forger did not query Sign there is at least one index i with $b'_i > 0$ because of the checksum construction. If the forger did query Sign there is at least one index i with $b'_i < b_i$ because of the checksum construction. In both cases there is at least one value σ'_i the adversary was unable to take from prior information. As there are κ' keys mapping x_c to some fix value, there are κ'^j possibilities to map σ'_i to $\mathsf{pk}_i = f^{j}_{\sigma'_i}(x_c)$ for $j = w - 1 - b'_i$. So the probability that $f^{b'_i}_{\mathsf{sk}_i}(x_c) = \sigma'_i$ holds is $(\kappa'^j - 1)/\kappa'^j$. So in the worst case we find a collision with probability at least $(\kappa' - 1)/\kappa'$ as we stated above.

Since both cases are mutually exclusive, the probability ϵ_{KCR} of finding a key collision is at least $\epsilon_{\mathrm{KCR}} \geq ((\kappa' - 1)/\kappa')\epsilon$. the time required by $\mathsf{A}_{\mathrm{KCR}}$ is $t_{\mathrm{KCR}} \leq t + t_{\mathsf{Kg}} + t_{\mathsf{Vf}}$. The time to answer the signature query is already contained in the runtime of the forger. This concludes the proof of part b). □

3.3 Security level

We now compute the security level of W-OTS for the case that only generic attacks against the SKR or KCR property of the function family $F(n)$ exist. This reflects the security of W-OTS, if the used function family $F(n)$ has no specific weaknesses. It corresponds to the security level defined in [23]. The best known generic attack against the second-key resistance of $F(n)$ is a brute-force key recovery attack. The best known generic attack against the key-collision resistance of $F(n)$ is a birthday attack. As before, we count the running time of an algorithm as the number of evaluations of elements from $F(n)$. A simple counting argument gives that t_{Kg} and t_{Vf} are both bounded by ℓw evaluations of elements from $F(n)$. In the following corollary we use two bounds on the parameters which are fulfilled by most practical parameter sets. Anyhow, following the proof of the corollary one can easily compute the security level for any specific set of parameters.

Note, that in case of $\kappa = 1$ it is impossible to find two signatures for the same message by construction. Therefore W-OTS is SU-CMA secure if it is EU-CMA secure and $\kappa = 1$. For the computation of the security level in this section we therefore assume $\kappa, \kappa' \geq 2$, such that there exists at least one key collision for each preimage.

Corollary 2. *Let* $b = \log(t/\epsilon)$ *denote the security level and use* ℓw *as upper bound for* t_{Kg} *and* t_{Vf}*, respectively.*
a) Let $F(n)$ *be* $(2^{n-1-\log\kappa(F(n))} + 1, (\kappa'(F(n)) - 1)/(2\kappa'(F(n))))$-SKR *and* $(t_{\text{PRF}}, \epsilon_{\text{PRF}})$-PRF *with* $\log(t_{\text{PRF}}/\epsilon_{\text{PRF}}) = n$ *and* $\kappa'(F(n)) = \kappa(F(n)) = 2$*. Let* $4/(\ell w 2^{w-2}) \leq 2^{n-w-2\log(\ell w)}$*. Then the security level of W-OTS under generic attacks is*

$$b \geq n - w - 2\log(\ell w) \tag{17}$$

b) Let $F(n)$ *be* $(2^{(n-\log\kappa'(F(n)))/2}, 1/2)$-KCR *and* $(t_{\text{PRF}}, \epsilon_{\text{PRF}})$-PRF *with* $\log(t_{\text{PRF}}/\epsilon_{\text{PRF}}) = n$ *and* $\kappa'(F(n)) = \kappa(F(n)) = 2$*. Let* $2\ell w \leq 2^{(n-1)/2-1}$*. Then the security level of W-OTS under generic attacks is*

$$b \geq (n-1)/2 - 1 \tag{18}$$

Proof. a) We use a $(t_{\text{SKR}}, \epsilon_{\text{SKR}})$-SKR family $F(n)$. The best generic attack on the second-key resistance of $F(n)$ is a brute-force key recovery attack. An attacker that searches through $t_{\text{KOW}} = 2^{n-1-\log\kappa}$ keys has success probability $\epsilon_{\text{KOW}} = 1/2$ for recovering the correct key. By Proposition 2 this yields an

$$t_{\text{SKR}} = 2^{n-1-\log\kappa} + 1, \epsilon_{\text{SKR}} = \frac{1}{2} \cdot \frac{\kappa' - 1}{\kappa'}$$

adversary on the second-key resistance of $F(n)$. The security level of the SKR property of $F(n)$ in presence of this adversary is $b = n - \log(\kappa - 1)$, assuming $\kappa = \kappa'$. We further assume that $F(n)$ is $(t_{\text{PRF}}, \epsilon_{\text{PRF}})$-PRF with $\log(t_{\text{PRF}}/\epsilon_{\text{PRF}}) = n$. This justifies using $\kappa' = \kappa = 2$ since $\kappa' \geq 2$ is required to ensure that second keys actually exist. The security level of W-OTS is computed as follows

$$\begin{aligned} 2^b = \frac{t}{\epsilon} &\geq \frac{t_{\text{SKR}} - t_{\mathsf{Kg}} - t_{\mathsf{Vf}} - 1}{\epsilon_{\text{SKR}} \ell^2 w^2 \kappa^{w-2}} \cdot \frac{\kappa' - 1}{\kappa'} \\ &= \frac{2^{n-\log\kappa} - 4\ell w}{\ell^2 w^2 \kappa^{w-2}} \cdot \frac{\kappa' - 1}{\kappa'} \cdot \frac{\kappa}{\kappa - 1} \\ &\geq 2^{n-w+1-2\log(\ell w)} - \frac{4}{\ell w 2^{w-2}} \end{aligned}$$

Since $4/(\ell w 2^{w-2}) \leq 2^{n-w-2\log(\ell w)}$ per assumption we finally obtain $b \geq n - w - 2\log(\ell w)$ as security level of W-OTS.

b) We use a $(t_{\mathrm{KCR}}, \epsilon_{\mathrm{KCR}})$-KCR family $F(n)$ and assume that the best attack on the key-collision resistance of $F(n)$ is a birthday attack, i.e. an adversary that searches through $t_{\mathrm{KCR}} = 2^{(n-\log\kappa')/2}$ keys has success probability $\epsilon_{\mathrm{KCR}} = 1/2$ for finding a key collision. The security level of the KCR property of $F(n)$ in presence of this adversary is $b = (n - \log\kappa')/2 - 1$. Again we assume that $F(n)$ is $(t_{\mathrm{PRF}}, \epsilon_{\mathrm{PRF}})$-PRF with $\log(t_{\mathrm{SKR}}/\epsilon_{\mathrm{SKR}}) = n$ and use $\kappa' = \kappa = 2$. The security level of W-OTS is computed as follows

$$2^b = \frac{t}{\epsilon} \geq \frac{t_{\mathrm{KCR}} - t_{\mathsf{Kg}} - t_{\mathsf{Vf}}}{\epsilon_{\mathrm{KCR}}} \cdot \frac{\kappa' - 1}{\kappa'} \geq 2^{(n-1)/2} - 2\ell w$$

Since $2\ell w \leq 2^{(n-1)/2-1}$ per assumption we finally obtain $b \geq (n-1)/2 - 1$ as security level of W-OTS.

4 Relation between security notions

In this section we complete the analysis of implications and separations between key one-wayness (KOW), second-key resistance (SKR), key-collision resistance (KCR), and pseudorandomness (PRF) started with Propositions 1 and 2, whereas the suspected separation PRF $\not\Rightarrow$ SKR is left as an open problem. Figure 1 summarizes our findings.

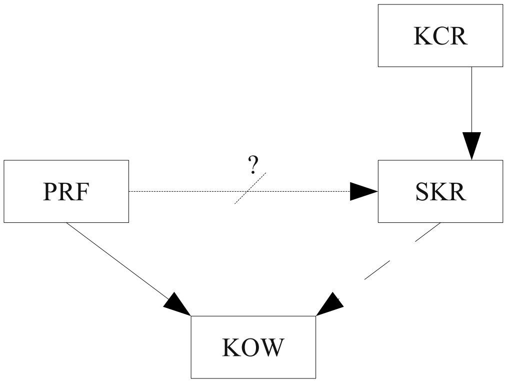

Fig. 1. Implications among PRF, KOW, SKR, and KCR. A straight arrow $A \to B$ means that property A implies property B and a dashed line means that the implication is conditional. When there is no arrow, it means that we show a separation. The suspected separation between PRF and SKR is an open problem.

Proposition 3 **(KOW $\not\Rightarrow$ PRF).** *Let $g : \{0,1\}^n \to \{0,1\}^n$ be a one-way function. Then there exists a family $F(n)$ that is* KOW *but not* PRF.

Proof. We construct the function family $F(n)$ as follows: $f_k(x) := g(k)$, $\forall k, x \in \{0,1\}^n$. $F(n)$ is not pseudorandom as there exists an distinguisher Dis querying Box t times with different values and if Box answers all queries with the same value Dis returns 1. The success probability of Dis is $\epsilon_{\mathrm{PRF}} = 1 - \frac{2^{n(2^n-t)}}{2^{n2^n}} = 1 - \frac{1}{2^{tn}}$ running in time t. $F(n)$ is KOW as we could construct an adversary A_{OW} on the one-wayness of g using any adversary $\mathsf{A}_{\mathrm{KOW}}$ on the KOW property of $F(n)$. On input y A_{OW} chooses $x \xleftarrow{\$} \{0,1\}^n$, runs $k \leftarrow \mathsf{A}_{\mathrm{KOW}}(x, y)$ and returns k. □

Proposition 4 (KOW $\not\Rightarrow$ SKR)**.** *Let $F(n)$ be (t, ϵ)-KOW. Then, there is a family $F'(n)$ that is $(t, 2\epsilon)$-KOW but not SKR.*

Proof. We denote functions in $F(n)$ and $F'(n)$ with f and f' respectively, and we define $F'(n)$ as follows. For all $k \in \{0,1\}^{n-1}$, we define $f'_{k||0} := f_{k||0} =: f'_{k||1}$ and add $\{f'_{k||0}, f'_{k||1}\}$ to $F'(n)$. Thus, we have that $f'_k = f'_{k\oplus(0^{n-1}||1)}$ for every $k \in \{0,1\}^n$. Observe that $F'(n)$ is KOW because a successful adversary against KOW in $F'(n)$ will output the correct key w.r.t. $F(n)$ with probability at least 1/2. Given the fact that key-collisions are easy to find, the new family is not SKR. On input (x, k), the adversary simply outputs $k \oplus (0^{n-1}||1)$ and wins with probability 1 in the SKR experiment. □

Proposition 5 (KOW $\not\Rightarrow$ KCR)**.** *Let $F(n)$ be (t, ϵ)-KOW. Then, there is a family $F'(n)$ of functions that is $(t, \epsilon + 2/2^n)$-KOW but not KCR.*

Proof. Let $k_1, k_2 \in \{0,1\}^n$ be distinct, fixed keys and denote functions in $F(n)$ and $F'(n)$ with f and f' respectively. We define the new family $F'(n)$ as follows. For all $k \in \{0,1\}^n \setminus \{k_2\}$, we set $f'_{k_1} := f_{k_1}$ and inject a collision $f'_{k_2} := f'_{k_1}$. The new family is still KOW because the challenge key is chosen uniformly at random and the above change does not influence the adversaries success probability but for a negligible $(2/2^n)$ amount. However, the new family is not KCR because on input a description of $F'(n)$, the adversary will simply output (k_1, k_2), under which every input collides. □

Proposition 6 (KCR $\Rightarrow$ SKR)**.** *Let $F(n)$ be (t, ϵ)-KCR. Then $F(n)$ is (t, ϵ)-SKR.*

Proof. Towards contradiction, let us assume a successful adversary A that breaks SKR for $F(n)$. We show how to use A as a block-box in an algorithm $\mathcal{B}$ to break KCR. On input x from the KCR experiment, the algorithm $\mathcal{B}$ chooses k uniformly at random and runs $\mathsf{A}(x, k)$. The subroutine returns k' such that $k' \neq k$ and $f_k(x) = f_{k'}(x)$ with probability at least ϵ. Then, $\mathcal{B}$ returns the pair (k, k') and is successful with the same probability ϵ and a negligible computational overhead. □

Proposition 7 (SKR $\not\Rightarrow$ KCR)**.** *Let $F(n)$ be (t, ϵ)-SKR. Then, there is a family $F'(n)$ of functions that is $(t, \epsilon + 2/2^n)$-SKR but not KCR.*

Proof. Let $k_1, k_2 \in \{0,1\}^n$ be distinct, fixed keys and denote functions in $F(n)$ and $F'(n)$ with f and f' respectively. We define the new family $F'(n)$ as follows. For all $k \in \{0,1\}^n \setminus \{k_2\}$, we set $f'_{k_1} := f_{k_1}$ and inject a collision $f'_{k_2} := f'_{k_1}$. The new family is still SKR because the challenge key is chosen uniformly at random and the above change does not influence the adversaries success probability but for a negligible $(2/2^n)$ amount. However, the new family is not KCR because on input a description of $F'(n)$, the adversary will simply output (k_1, k_2), under which every input collides. □

Proposition 8 (PRF $\not\Rightarrow$ KCR)**.** *Let $F(n)$ be (t, ϵ)-PRF. Then, there is a family $F'(n)$ of functions that is $(t, \epsilon + 2/2^n)$-PRF but not KCR.*

Proof. We construct F' as follows. We select $k_1, k_2 \xleftarrow{\$} \{0,1\}^n$, and define $f'_k \in F'(n)$ as $x \mapsto \begin{cases} 0^n \text{ for } k \in \{k_1, k_2\} \\ f_k(x) \text{ otherwise} \end{cases}$ for all $k \in \{0,1\}^n$.

Towards contradiction, let us assume that F' is not PRF. Then, there is a distinguisher A that breaks PRF with non-negligible probability ϵ. With access to Box, we construct an adversary

Dis against the family F. The distinguisher Dis answers all queries of A with its own oracle and simply forwards the output of A as its decision. Hence, the advantage of Dis is $\epsilon - 2/2^n$ because the probability that Box represents f_{k_1} or f_{k_2} is at most $2/2^n$, which contradicts the assumption.

Furthermore, F' is clearly not KCR because on input a preimage x, one can simply output (k_1, k_2) as the "colliding" keys. □

The following corollaries can be proven in analogy to Proposition 3.

Corollary 3 (SKR $\not\Rightarrow$ PRF)**.** *If second preimage resistant functions exist, there is a family $F(n)$ that is* SKR *but not* PRF*.*

Corollary 4 (KCR $\not\Rightarrow$ PRF)**.** *If collision resistant functions exist, there is a family $F(n)$ that is* KCR *but not* PRF*.*

5 Implementation

In this section we discuss how to implement W-OTS. First we introduce two possible heuristic instantiations for $F(n)$. Then we show how to reduce the private key size.

The main challenge implementing W-OTS, is the instantiation of $F(n)$. We propose two different possible heuristic instantiations. First, it is possible to use any hash function `Hash` with block length b and output size n that uses the the Merkle-Darmgard (M-D) construction [26]. We construct the function family $F(n)$ as

$$f_K(M) = \texttt{Hash}(\texttt{Pad}(K)||\texttt{Pad}(M)),$$

for key $K \in \{0,1\}^n$, message $M \in \{0,1\}^n$ and $\texttt{Pad}(x) = (x||10^{b-|x|-1})$ for $|x| < b$. We argue that it is reasonable to assume that this is a PRF if `Hash` is a secure cryptographic hash function. The assumptions we use are essentially those used for the security of HMAC using a practical hash function. In [3] it is assumed, that the compression function of a secure M-D hash function is a PRF if keyed using the input. In [4], it is assumed, that the compression function of a secure M-D hash function is a PRF if keyed on the chaining input. Then it is shown, that a fixed input length M-D hash function, keyed using the initialization vector (IV) is a PRF for fixed length inputs if the assumptions made before hold. In our construction the internal compression function of `hash` is evaluated twice: First on the IV and the padded key, second on the resulting chaining value and the padded message. Due to the pseudorandomness of the compression function when keyed on the message input, the first evaluation works as a pseudorandom key generation. As we have a fixed message length the second iteration is a PRF keyed using the IV input. For the new SHA-3 hash function, this construction will not be necessary, as one requirement for the candidate functions was the PRF property.

Second, as we require $F(n)$ to be a PRF it is possible to use a block cipher. The standard heuristic assumption about block ciphers is the assumption that they are pseudorandom permutation families (PRP) and PRP are a special class of PRF. Hence a straight forward use of any block cipher is possible. As today many platforms provide hardware acceleration for AES this instantiation might also lead significant speed-ups in practice. For both constructions we make the following assumption. As long as no specific attack against the pseudorandomness of the used hash function or block cipher is known that performs better than a brute-force key recovery attack, the PRF security level of the used hash function or block cipher is n bit (i.e. $\log(t_{\mathrm{PRF}}/\epsilon_{\mathrm{PRF}}) = n$). This justifies the assumption that $\kappa \leq 2$.

Another point for a practical implementation is the private key size of ℓn bits. We can use $F(n)$ to reduce the private key size to n bits, using pseudorandom key generation. This can be done using the construction

$$\mathsf{sk} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell) = (f_{\textsc{Seed}}(0), \ldots, f_{\textsc{Seed}}(\ell - 1)), f_{\textsc{Seed}} \in F(n)$$

where Seed is the new n bit private key. The security of this construction is shown in [9].

6 Conclusion

We have provided three security reductions for W-OTS. The first one shows that W-OTS provides a security level of at least $n - w - 1 - 2\log(\ell w)$, if the security level of the PRF property of the used function family is at least n. This reduction is especially appealing because it can be instantiated using any cryptographic hash function or block cipher. As a block cipher with n bit key and block size is normally assumed to provide n bit security against distinguishing attacks this justifies our assumption of $\kappa \leq 2$ given Lemma 1. The same holds for hash functions with n bit output. When using $n = 128$ (i.e. AES) and $w = 16$ the security level of W-OTS is at least 91 while the size of a signature is 560 Bytes. The more conservative approach of using $n = 160$ (i.e. SHA-1) yields a security level of at least 129, which guarantees long-term security but results in larger signatures of 860 Bytes.

However, this reduction does not guarantee strong unforgeability, except in case of $\kappa = 1$ meaning that no key collisions exist. If no key collisions exist, each message has a unique signature and the scheme is trivially SU-CMA when it is EU-CMA. Showing SU-CMA in general requires that the underlying functions are either SKR or KCR. This has been shown in the second and third reduction. The security level of W-OTS is at least $n - w - 1 - 2\log(\ell w)$ if the security level of the SKR property of the used PRF is at least $n - \log(\kappa - 1)$. When using KCR, the security level of W-OTS is at least $(n - 1)/2 - 1$ if the security level of the KCR property of the used PRF is at least $(n - \log \kappa)/2 - 1$. We remark that the last reduction also works with the original Winternitz construction using a family of collision resistant hash functions. In other words, W-OTS is SU-CMA if the used function is collision resistant. However, using a PRF with additional KCR property has the benefit that an exact value for the maximum number κ of key collisions that occur within the family is known. This is required for the estimation of the exact security level.

As a by-product we have defined three key-based security notions for function families: key one-wayness (KOW), second-key resistance (SKR), and key-collision resistance (KCR). We have analyzed implications and separations among these properties and pseudorandomness. Although, these relations have not been analyzed before, they support the common intuition. In fact, key-based and non-key-based notions share an analoguous hierarchy of implications and separations with respect to preimage resistance, second preimage resistance, and collision resistance. We refer the reader to [30] for a discussion on non-key-based notions.

We would like to point out that KCR functions f_k can easily be obtained from collision resistant functions g_k by defining $f_k(x) = g_x(k)$. If we require f to inherit the PRF property of g, we have to assume that the compression function of g is dual-PRF, meaning that it is a PRF regardless of which input it is keyed with. This is also a requirement of the security proof of HMAC [2]. SKR functions can be constructed equivalently while the KOW property is immediately implied by the PRF property. While we have shown the separation of PRF and KCR, we leave the suspected separation of PRF and SKR as an open problem. Moreover, we have studied the relation between the security level of a PRF and the maximum number of key collisions that can occur. A deeper analysis of the number of key collisions of a function family is an interesting topic for future work.

References

1. Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–270. Springer Berlin / Heidelberg, 2000.
2. Mihir Bellare. New proofs for nmac and hmac: Security without collision-resistance. In *Advances in Cryptology - CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 602–619. Springer, 2006.
3. Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In Neal Koblitz, editor, *Advances in Cryptology — CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin / Heidelberg, 1996.
4. Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *Proceedings of 37th Annual Symposium on Foundations of Computer Science*, pages 514–523. IEEE, 1996.
5. Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences*, 61(3):362–399, 2000.
6. Daniel Bleichenbacher and Ueli M. Maurer. Directed acyclic graphs, one-way functions and digital signatures. In *Advances in Cryptology - CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 75–82, 1994.
7. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matt Franklin, editor, *Advances in Cryptology - CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 227–242. Springer Berlin / Heidelberg, 2004.
8. 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.
9. 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.
10. 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, *ACNS*, volume 4521 of *Lecture Notes in Computer Science*, pages 31–45. Springer, 2007.
11. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222. Springer Berlin / Heidelberg, 2004.
12. Ran Canetti, Daniele Micciancio, and Omer Reingold. Perfectly one-way probabilistic hash functions (preliminary version). In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC '98, pages 131–140, New York, NY, USA, 1998. ACM.
13. L. C. Coronado García. On the security and the efficiency of the merkle signature scheme. Technical Report 2005/192, Cryptology ePrint Archive, 2005. Available at `http://eprint.iacr.org/2005/192/`.
14. Erik Dahmen, Katsuyuki Okeya, Tsuyoshi Takagi, and Camille Vuillaume. Digital signatures out of second-preimage resistant hash functions. In *2nd International Workshop on Post-Quantum Cryptography (PQCrypto)*, volume 5299 of *Lecture Notes in Computer Science*, pages 109–123, 2008.
15. C. Dods, N. Smart, and M. 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.
16. Marc Fischlin. Pseudorandom function tribe ensembles based on one-way permutations: Improvements and applications. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 432–445. Springer Berlin / Heidelberg, 1999.
17. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.
18. Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.
19. Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual Symposium on the Theory of Computing*, pages 212–219, New York, 1996. ACM Press.

20. 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.
21. 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.
22. Leslie Lamport. Constructing digital signatures from a one way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, 1979.
23. Arjen K. Lenstra. Key lengths. Contribution to The Handbook of Information Security, 2004.
24. Mark Luk, Adrian Perrig, and Bram Whillock. Seven cardinal properties of sensor network broadcast authentication. In *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, 2006.
25. Ralph Merkle. A certified digital signature. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO' 89 Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 218–238. Springer Berlin / Heidelberg, 1990.
26. Ralph Merkle. One way hash functions and DES. In Gilles Brassard, editor, *Advances in Cryptology — CRYPTO' 89 Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446. Springer Berlin / Heidelberg, 1990.
27. Adrian Perrig. The BiBa one-time signature and broadcast authentication protocol. In *ACM Conference on Computer and Communications Security*, pages 28–37, 2001.
28. Adrian Perrig, Ran Canetti, J. D. Tygar, and Dawn Xiaodong Song. Efficient authentication and signing of multicast streams over lossy channels. In *IEEE Symposium on Security and Privacy*, pages 56–73, 2000.
29. 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*, volume 2384 of *Lecture Notes in Computer Science*, pages 1–47. Springer Berlin / Heidelberg, 2002.
30. Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In Bimal Roy and Willi Meier, editors, *Fast Software Encryption*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388. Springer Berlin / Heidelberg, 2004.
31. 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.
32. 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.