An Efficient Post-Quantum One-Time Signature Scheme

Kassem Kalach[1(✉)] and Reihaneh Safavi-Naini[2]

[1] Department of Combinatorics and Optimization and Institute for Quantum Computing, University of Waterloo, Waterloo, ON, Canada
k2kalach@uwaterloo.ca
[2] Department of Computer Science, University of Calgary, Calgary, Canada

Abstract. One-time signature (OTS) schemes are important cryptographic primitives that can be constructed using one-way functions, and provide post-quantum security. They have found diverse applications including forward security and broadcast authentication. OTS schemes are time-efficient, but their space complexity is high: the sizes of signatures and keys are linear in the message size. Regular schemes (e.g., based on discrete logarithm or factoring problems, and their variants), however, have very low space complexity but are not time-efficient. Therefore, in particular, they are not suitable for resource-constraint devices. Many widely used signature schemes are not post-quantum. In this paper, we give a signature scheme that has the advantages of the previous two approaches. It provides constant-size short signatures, and is much more time-efficient than schemes in the second approach. We prove that our scheme is post-quantum secure as long as the SVP in ideal lattices is hard in the presence of a quantum computer. We use SWIFFT: a family of provable collision-resistant functions, which have efficient implementations, comparable with that of SHA-256, hence our proposed scheme could be used on resource-constrained devices.

Keywords: Post-quantum cryptography · Broadcast authentication · Digital signatures

1 Introduction

Digital signatures schemes may be grouped into two main categories or types depending on the two fundamentally distinct approaches of Lamport [25] and Diffie-Hellman [14], each with its own features and target applications. Here is a brief overview.

The first type, known as *one-time signature* (OTS) schemes, is typically founded on general one-way functions (e.g., SHA-256), and allows to sign one

K. Kalach–The bulk of this research was done when the author was a Post-doctoral Fellow at the University of Calgary. The author has been a member of the CryptoWorks21 program.

O. Dunkelman and L. Keliher (Eds.): SAC 2015, LNCS 9566, pp. 331–351, 2016.
DOI: 10.1007/978-3-319-31301-6_20

message per secret/public key pair. The first scheme of this type was described in 1976 by Diffie and Hellman [14], based on a suggestion of Lamport, and eventually published in 1979 [25]. More generally, fixed-time or t-time signature schemes allow to sign at most t messages using the same key pair; signing more messages makes the scheme insecure. The parameter t is a positive integer that must be chosen properly during the setup phase.

The second type involves one-way functions whose security is based on the computational hardness of some strongly structured mathematical problems (e.g., the discrete logarithm and factoring problems), and allows to sign an *undetermined* number (polynomially bounded) of messages using the same key pair. They are known as ordinary or regular schemes because the most deployed schemes, which are variants of ElGamal [18]–DSA, and ECDSA [24]– and RSA [41], belong to this category. Regular schemes have usually short keys and signatures. However, they involve time-consuming arithmetic operations (modular multiplications and exponentiations), particularly inconvenient for applications on devices with limited resources.

OTS schemes has attractive properties: (i) they can be constructed from general one-way functions, thus increasing the possible problems on which the security can be based [4]; (ii) they are usually very computationally efficient; and (iii) if the one-way function became insecure, then one could replace the function (e.g. SHA-1 with SHA-2). This is not possible in regular schemes; (iv) importantly, such one-way functions are quantum secure; one may only need to double the size of the domain to overcome attacks based on Grover's algorithm [21].

Applications of OTS schemes are many and diverse. They have been used to (i) construct on-line/off-line schemes that combine OTS and regular schemes [19]; (ii) transform regular unforgeable schemes into strong ones [23]; (iii) construct chosen-ciphertext composed encryption schemes [15]; (iv) construct fail-stop schemes [22]; and (v) provide forward-secure signatures [1] and one-time proxy signatures [28]. They are also used in broadcast authentication [38,40]. Importantly, they are used to construct post-quantum digital signatures [8,9].

A signature scheme *must* provide (0) some well defined security; and *should* afford at least some of the following features: (1) short signatures; (2) fast key-generation, signing and verification algorithms; (3) possibility to sign many messages per key pair; and (4) short keys. These properties, whose order depends on the target application, have driven the research on signature schemes.

Important typical shortcomings of OTS schemes are (i) the number of messages to be signed per key pair is predetermined; (ii) the signature is long; and (iii) the secret/public keys are also long. Overcoming the last two has motivated much research over the years [4,6,9,16,30,35,39,40,48]. Here we focus on schemes that are based on (general) post-quantum one-way functions, which are evolutions of Lamport scheme [25], and have a number of attractive properties.

Here is a review of some schemes that are relevant to our work. Throughout this work, let $f\colon \{0,1\}^\gamma \to \{0,1\}^\gamma$ a one-way function with a security parameter γ, and M an ℓ-bit message. Lamport scheme to sign a 1-bit message b works as follows. Choose randomly two secrets, x_0, x_1, and compute their corresponding

images under f, $y_0 = f(x_0)$ and $y_1 = f(x_1)$. The signing key is $SK := (x_0, x_1)$ and the verification key is $PK := (y_0, y_1)$. The signature is then x_b. Any party can verify the signature by evaluating f on x_b and comparing the result with y_b in the public key. The scheme is provably secure, efficient, and simple. However, there are two main disadvantages: the signature size and keys size are linear in the message size. Subsequently, this construction has been improved or generalized over decades [4–6,16,30–32,39,40,48].

The first improvement (M-OTS) is due to Merkle [30,32] who reduced the keys and signature sizes to almost half on the price of only logarithmic additional time overhead. Here is briefly the idea. The key generation consists in choosing random secrets x_i and computing the public values $y_i = f(x_i)$ for $1 \leq i \leq t$ where $t = \ell + \lfloor \log \ell \rfloor + 1$. To sign a message M, form $M' = M||c = (b_1, \ldots, b_t)$ where c is the binary representation of the number of zeros in M, then find the positions $i_1 < \cdots < i_k$ such $b_{i_j} = 1$ where $1 \leq j \leq k$. The signature is then $(s_1, \ldots, s_k)$ with $s_j = x_{i_j}$. To verify a signature $\sigma = (s_1, \ldots, s_k)$ on M, again form M', determine the positions of ones, then accept σ if and only if $y_j = f(s_j)$ for all $1 \leq j \leq k$. More details can be found in [29,46].

Winternitz [30,32] proposed to Merkle the idea, called W-OTS, of signing several bits simultaneously at the expense of more evaluations of the one-way function, thus trading time for space. More details are given in Sect. 5.4.

Implicitly based on cover-free families (defined in Sect. 3.1), Bos and Chaum gave a scheme able to reduce the public key size to almost half, or to decrease the signature size and verification time at the expense of increasing the public key length [6]. Signing is by revealing k out of $e = 2k$ secret elements. Reyzin and Reyzin [40] presented a slight generalization of BC [6], considering a general k instead of $k = e/2$. They also proposed an r-time signature scheme HORS (for Hash to Obtain Random Subset) [40]. For time and space efficiency, they use random structures instead of explicit CFF constructions. However, the security holds using a strong additional assumption: the existence of *subset-resilient* functions. The signature scheme can be used r times, for small values of r. However, the security decreases as r increases. Pieprzyk et al. [39] proposed a t-time signature scheme (HORS++) that achieves security against t-adaptive chosen-message attacks, using explicitly a t-CFF. They gave t-CFF constructions based on polynomials, error-correction codes, etc. However, it "is only of theoretical interest" [39]. They also extended the scheme to increase the number of messages to sign, using one-way hash chains.

Based on the discrete log assumption [37], van Heyst and Pedersen gave a fail-stop scheme (vHP) with signature twice as long as the security parameter [22]. They also gave methods to transform OTS scheme into t-time one. Groth [20] employed an NIZK proof system to construct a OTS scheme similar to that van Heyst and Pedersen. Inspired by Pedersen commitment scheme [37], Zaverucha and Stinson gave a OTS scheme (ZS) with signature size about 273 bits for 128-bit security, thanks to the algebraic properties of the DLP. However, the latter has an efficient quantum algorithm [44], and the scheme verification algorithm is slower than most OTS schemes and ECDSA.

1.1 Contribution

We present a new OTS (and t-time signature) scheme that combines the advantages of both reviewed approaches. Indeed, it provides signatures independent of the message size, thus much shorter signatures, compared with all schemes based on general one-way functions [3,5,6,25,30,40] (including Winternitz one) while maintaining the same time complexity.

The scheme is much more time-efficient than regular schemes, which are based on discrete logarithm or factoring problems and their variants. In particular, it has very simple and fast signing, requiring only regular additions, and a very fast verifying, requiring only additions and *one* evaluation of a hash function. We use a family of hash functions (SWIFFT), which can have efficient implementations comparable with that of SHA-256, hence our scheme could be used even on resource-constrained devices.

The security of the scheme is strongly unforgeable under adaptive chosen-message attacks and reduced to the security of knapsack functions (SWIFFT), which are provably collision-resistant assuming the hardness of the shortest vector problem in cyclic lattices. We give a formal security proof based on cover-free families and hash functions. On top of that, the scheme is post-quantum secure, meaning resistant against quantum attacks.

We use a perceived disadvantage of SWIFFT, not pseudo-random due to linearity, as a feature and use its "conditional" linearity (when the input is not binary vector) to compress several secrets into one, resulting in a short signatures. However, there is a main challenge: SWIFFT security holds if the input is a vector of small coefficients. Adding component-wise many vectors may result in a vector of large coefficients, and render the function easy to invert or collide. We give solutions to overcome this issue using some new technical ideas.

1.2 The Scheme High-Level Description

The scheme is based on 1-cover-free families and one-way functions. A w-cover-free family is a collection $\mathcal{B}$ of subsets of a set E with the property that the union of any w subsets in $\mathcal{B}$ will not include ("cover") any other subset in $\mathcal{B}$. As observed by Merkle, the one-way function can be viewed as a commitment scheme. During key generation, the signer commits to randomly chosen secret elements associated with E by computing and publishing their corresponding images. This makes the public key. Each message corresponds to a unique subset in $\mathcal{B}$. To sign a message, the signer finds its corresponding subset in $\mathcal{B}$ and typically reveals the secret elements corresponding to this subset. Using the homomorphic property of SWIFFT, we can add these secrets and effectively compress them into a single value, making up a short signature. This also provides faster verification: to verify a signed message, again find the associated subset in $\mathcal{B}$ and add mod p the corresponding public values. Finally, compare the result with the provided signature.

1.3 Paper Organization

The remaining material is organized as follows. Related work are given in the next section while preliminaries are given in Sect. 3. Our new scheme is the subject of Sects. 4, and 5 contains our conclusions and future work.

2 Related Work

The literature on OTS schemes and their variants is immense, more details about this topic can be found in [16,29,46]. Here is a review of some additional related schemes.

One main disadvantage of OTS schemes is the authentication and management of many public keys. Using a complete binary tree and a collision-resistant hash function, Merkle [30,32] introduced a solution to this problem, which allows to transform any OTS into a scheme that allows to sign many messages using one public key.

Naor et al. [36] studied whether OTS became practical by applying recent improvements in hash tree traversal to M-OTS scheme. In order to provide a practical efficient broadcast authentication protocol, Perrig proposed an r-time signature scheme [38] having fast verification and relatively short signatures when compared with the earlier related schemes. Both efficiency and security are based on the *birthday problem*, thus the name "BiBa" (for Bins and Balls). BiBa's disadvantages are the signing time, which is longer than that of most previously similar schemes, and its security considered in the *random oracle model* and decreases as r increases.

Bleichenbacher and Maurer formalized the concept of one-time signatures in terms of acyclic graphs. In particular, they unified the schemes of Lamport, Merkle, Winternitz, Vaudenay [47], and Even et al. [19]. To solve the problem of packet source authentication for multicast, Rohatgi [42] presented a hybrid signature mainly based on a collision-resistant hash function and a commitment scheme. The scheme improves over the scheme of [19]. In terms of power consumption, Seys and Preneel [43] evaluated ECDSA, M-OTS, W-OTS, HORS with or without Merkle trees.

Lyubashevsky and Micciancio gave an "asymptotically efficient" one-time signature scheme [26], which we will try to compare with our scheme. Mohassel gave a general construction for transforming any chameleon hash function to strongly unforgeable OTS schemes, in addition to instantiations based on the hardness of factoring, discrete-log, and worst-case lattice-based assumptions [35].

3 Preliminaries

The symbol '$\otimes$' is used for vector convolution, and '$\cdot$' to emphasize scalar multiplication, and $[k] = \{0, \ldots, k\}$ for integer k. The logarithm base 2 is denoted by log.

3.1 Cover-Free Families

Definition 1 (Cover-free Families). A set system $(E, \mathcal{B})$ is called a k-uniform w-cover-free family if E is a finite set of e elements (or points), $\mathcal{B}$ is a collection of s subsets (or blocks) of size k, and for all $\Delta \subset E$ with $|\Delta| = w$ and all $i \notin \Delta$, it holds that

$$\left| B_i \setminus \bigcup_{j \in \Delta} B_j \right| \geq 1.$$

A w-cover-free family with e elements and s subsets is denoted by $w\text{-}CFF(e, s)$.

In this work, we consider k-uniform 1-CFF where each subset has size k, and any two distinct subsets in $\mathcal{B}$ differ on at least one element. Importantly, a 1-CFF has optimal construction, which consists of setting $k = \lfloor e/2 \rfloor$. Indeed, there is a simple encoding algorithm that requires k subtractions (or additions) and about e comparisons using some pre-computation; it is well explained in [3,48]. In practice, encoding is twice faster than MD5 hashing, assuming MD5 requires 500 arithmetic operations [3]. The algorithm and more details are given in Sect. 5.3

3.2 Compact Knapsack Functions (SWIFFT)

A family of SWIFFT functions [27] is described by three main parameters: power of 2 security parameter n, small integer $m > 0$, and modulus $p > 0$. Define $R = \mathbb{Z}_p[\alpha]/(\alpha^n + 1)$ to be the ring of polynomials in α having integer coefficients modulo p and $\alpha^n + 1$. Any element of R can be written as a polynomial of degree smaller than n with coefficients in $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$. An instance of the family is specified by m fixed elements $a_1, \ldots, a_m \in R$. These elements (or multipliers) must be chosen uniformly and independently. The function output then corresponds to the following algebraic expression:

$$f(x) = \sum_{i=1}^{m} (a_i \otimes x_i) \in \mathbb{Z}_p^n$$

where $x_1, \ldots, x_m$ are polynomials in R with *binary* coefficients, and correspond to the input $x \in \{0,1\}^{n \times m}$. These functions are provably collision-resistant.

The security of the functions depends on the choice of the parameters, in particular the domain. In [26], $R = \mathbb{Z}_p[\alpha]/(\alpha^n + 1)$ where n is power of 2, the domain $D = \{y \in R : \|y\|_\infty \leq d\}$ for some d; $m > \frac{\log p}{\log 2d}$; and $p > 4dmn^{1.5} \log n$. Originally [34], the domain was $D = \{0, \ldots, \lfloor p^\delta \rfloor\}^n$ for positive δ. In practice, for most efficient FFT, p will be a prime such that $p - 1$ is multiple of $2n$ [27]. Here $\|y\|_\infty$ is the infinite norm of y. Unfortunately, the exact value of d is not known yet, it is a work in progress [33].

3.3 OTS Security Definition

The standard security notion for digital signatures is the (existential) *unforgeability under adaptive chosen-message attacks*, which can be modelled using the following security game between a challenger and an adversary. Consider DSS $= (\texttt{Gen}, \texttt{Sign}, \texttt{Ver})$ with some security parameter η, message space, and polynomial-time (in η) quantum adversary $\mathcal{A}$.

1. Run $\texttt{Gen}(1^\eta)$ to obtain a signing secret key SK and corresponding verification public key PK.
2. Give $\mathcal{A}$ the public key PK and a signature for at most *one* message. Let M_1 be the query asked by $\mathcal{A}$ and σ_1 its corresponding signature.
3. $\mathcal{A}$ finally outputs (M, σ).

An adversary $\mathcal{A}$ (existentially) forges a signature, or wins the unforgeability game (uf-cma) for DSS, if $\texttt{Ver}(PK, M, \sigma) = 1$ and $M \neq M_1$. Let $\text{PROB}^{\text{uf-cma}}_{\text{DSS},\mathcal{A}}$ denote the probability that $\mathcal{A}$ forges a signature taken over the random bits of the challenger and adversary; symbolically

$$\text{PROB}^{\text{uf-cma}}_{\text{DSS},\mathcal{A}} = \Pr[(M \neq M_1) \wedge \texttt{Ver}(PK, M, \sigma) = 1].$$

An adversary $\mathcal{A}$ is said to (t, ε)-win the security game if (M, σ) is output in time t such that $\text{PROB}^{\text{uf-cma}}_{\text{DSS},\mathcal{A}} = \varepsilon$.

Definition 2 (uf-cma OTS post-quantum security). A signature scheme is (existentially) unforgeable under chosen-message quantum attacks, or secure, if every polynomial-time quantum adversary forges a signature, or wins the unforgeability experiment, only with probability negligible in the security parameter.

This definition is the same as the standard classical one except that it considers quantum attacks. These attacks are captured in the security game, and model the real life scenarios of (classical) OTS schemes. More precisely, $\mathcal{A}$ can make use of any quantum computation (or communication) such as evaluating the hash function in superpositions. On the other hand, $\mathcal{A}$ cannot do queries in superpositions to the signing oracle, since the signing algorithm is controlled by a key only known to the signer, and signs one message at a time. In this definition, we assume that the scheme is implemented on a classical computer, otherwise it may require a different security model.

In this framework, systems based on DLP or factoring are not post-quantum because of Shor's quantum polynomial time algorithm for these problems [44]. However, this is not the case for SVP on which the security of SWIFFT functions are based.

Dedicated discussions about the requirements for the quantum security of classical schemes, which are satisfied in this work, are established by Song [45].

4 The New Scheme

In this section we describe our new signature scheme, give its formal proof of security, analyze its time and space complexity, and compare it with selected work. We also provide concrete parameters for a practical security level.

4.1 Scheme Description

We consider without loss of generality signatures of ℓ-bit messages M; longer messages can be adapted to this length using some secure hash function. We first give our OTS signature scheme, which can easily be extended into t-time signature one using standard techniques.

Setup: Consider an optimal k-uniform 1-CFF$(E, \mathcal{B})$ with $E = \{1, 2, \ldots, e\}$, and choose parameters e and k such that $\binom{e}{k} \geq 2^{\ell}$. An optimal 1-CFF requires that $k = \lfloor e/2 \rfloor$. Let $S\colon \{0,1\}^{\ell} \rightarrow \mathcal{B}$ be a bijection that maps a message $0 \leq M < \binom{e}{k}$ into the Mth element of $\mathcal{B}$ denoted by B_M. A very efficient constructive algorithm of this bijective is discussed in Sect. 3.1.

Let $f\colon \{0,1\}^{nm} \rightarrow \mathbb{Z}_p^n$ be a SWIFFT function where m, n, and p are the main parameters. The function also maintains its security even in a domain for small entries (see Sect. 3.2), this is needed for the security proof.

Key Generation: Choose randomly secret elements $x_1, \ldots, x_e \in \{0,1\}^{nm}$ and compute their images $y_i = f(x_i)$ for $i = 1, \ldots, e$. Output the secret signing key $SK := (x_1, \ldots, x_e)$ and the public verification key $PK := (y_1, \ldots, y_e)$.

Signing: To sign a message M using SK, compute the k-subset $B_M \in \mathcal{B}$, then

$$\sigma := \left(\sum_{i \in B_M} x_i \right) \in [k]^{nm} .$$

Verification: Given (M, σ, PK) as input, to verify whether σ is a valid signature for M using PK, compute B_M, check that $\sigma \in [k]^{nm}$ and output 1 if and only if

$$f(\sigma) = \left(\sum_{i \in B_M} y_i \mod p \right).$$

Output 0 otherwise.

Remark 1. Since $k < p$ is always the case in this work, the signature is actually the component-wise addition of k binary vectors of dimension nm, thus a vector in $[k]^{nm}$. Consequently, signing becomes much faster being reduced to component-wise counting, which can be done in parallel.

Remark 2. The functions are linear if addition is done in $\mathbb{Z}_p$ but not in $\mathbb{Z}_2$. This is straightforward to verify viewing the algebraic expression of f with nmmultipliers $a_1, \ldots, a_m$ and input $x \in \{0,1\}^{nm}$ as the matrix-vector product Ax. The matrix A is obtained from the skew-circulant matrices of a_i for $1 \leq i \leq m$ (Sect. 3.2). Consequently, `Ver` will output 1 whenever presented with a valid signature. Mathematically, the verification process is easy to validate:

$$f(\sigma) = A\sigma = A\left(\sum_{i \in B_M} x_i\right) = \sum_{i \in B_M} Ax_i = \sum_{i \in B_M} f(x_i) = \sum_{i \in B_M} y_i \mod p\,.$$

4.2 Security Proof

Theorem 1. *If $\mathcal{A}$ is a quantum adversary that (t, ε)-wins the unforgeability security game for our one-time signature scheme, then $\mathcal{A}$ can be used to $(t+c, \varepsilon c')$ find function collisions where c, c' are constants.*

Proof. Let DSS denote our one-time signature scheme, and assume that $\mathcal{A}$ queries a signature σ_1 for one message M_1 and outputs (M, σ) with $M \neq M_1$, which can be verified using PK; meaning $\Pr[\texttt{Ver}(PK, M, \sigma) = 1] \geq \varepsilon$.

Informally, forging a signature σ for a new M after obtaining a valid signature for M_1 is reduced to find collisions in SWIFFT functions. Considering the post-quantum security of the function, this cannot be done in polynomial time even using quantum computers. Otherwise, we could use the quantum forger to find collisions in the functions. Note that a quantum forger can always have some speedup over a classical one using Grover's search [21], reducing the adversary complexity by a square root at most. However, this is not considered in theory.

We now proceed with the formal proof. We devise a (quantum) algorithm $\mathcal{A}'^{\mathcal{A}}$ that uses $\mathcal{A}$ as a subroutine to find function collisions in time in the order of t with probability "close" to ε. Given access to f and other public parameters, the collision-finding algorithm $\mathcal{A}'^{\mathcal{A}}$ works as follows.

1. Create an instance of DSS using f
   (a) Choose $x_i \in \{0,1\}^{nm}$ and set $y_i = f(x_i)$ for all $i \in [e]$.
   (b) Run $\mathcal{A}$ on $PK = (y_1, \ldots, y_e)$ and all other system parameters.
2. When $\mathcal{A}$ queries a signature for M_1 do
   (a) Compute B_{M_1};
   (b) Return $\sigma_1 := \sum_{i \in B_{M_1}} x_i$.
3. When $\mathcal{A}$ outputs (M, σ)
   (a) Return σ.

We now analyze the behaviour of $\mathcal{A}'^{\mathcal{A}}$. First of all, it runs in time in the order of t (running time of $\mathcal{A}$). Indeed, the steps 1, 2 and 3 take a constant time. In Step 1, PK is exactly distributed as in the real execution. In Step 2, $\mathcal{A}'^{\mathcal{A}}$ answers similarly to a real execution the adaptive query made by $\mathcal{A}$ since it knows all the secrets. Therefore, $\mathcal{A}'^{\mathcal{A}}$ can answer any signature query with probability one.

Next, we prove that the reduction succeeds in outputting a collision when $\mathcal{A}$ makes a forgery, and compute the probability of finding a collision. Let $\bar{\sigma} = \sum_{i \in B_{M_1}} \bar{x}_i$ be the legitimate signature, and $\sigma = \sum_{i \in B_M} x_i$. There are the following cases in which $f(\sigma) = f(\bar{\sigma}) = \sum_{i \in B_M} y_i \mod p$:

1. $\sigma = \bar{\sigma}$ with $\sigma \neq \sigma_1$;
2. $\sigma = \bar{\sigma}$ with $\sigma = \sigma_1$;
3. $\sigma \neq \bar{\sigma}$.

The objective now is to show that the probability of the first two cases is negligible. Case (1) happens with probability smaller than 2^{-nm}. Indeed, $\bar{\sigma}$ is not known to the adversary, and it requires at least one uniformly distributed secret value in $\{0,1\}^{nm}$ since $|B_M \setminus B_{M_1}| \geq 1$. The only thing $\mathcal{A}$ knows about $\bar{\sigma}$ is that it is the sum of at least one uniformly secret nm-dimensional binary vector, which was not considered beforehand, and other values in $[k]^{nm}$, which may be part of σ_1 (see Lemma of Sect. 5.1).

Case (2) essentially happens with the same probability as the first case since it reduces to the problem of finding a different k-subset of elements in the SK that sums to σ. Again, this requires at least one element in $\{0,1\}^{mn}$. The probability of this even is upper bounded by one over the number of possible distinct k-subset sums, meaning in the order of 2^{-nm}.

Case (3) happens with complementary probability exponentially close to 1. Accordingly, $\mathcal{A}'^{\mathcal{A}}$ can find a collision whenever $\mathcal{A}$ makes a forgery. The probability of this event is

$$\mathrm{PROB}_f(\mathcal{A}'^{\mathcal{A}}) \geq \mathrm{PROB}^{\text{uf-cma}}_{\mathrm{DSS},\mathcal{A}} \cdot (1 - \mathrm{negl}(nm)).$$

Given that f is collision-resistant, $\mathrm{PROB}_f(\mathcal{A}'^{\mathcal{A}})$ is negl in nm when $t = poly(n)$. Therefore, $\mathrm{PROB}^{\text{uf-cma}}_{\mathrm{DSS},\mathcal{A}}$ is negligible. In conclusion, the security of the signature scheme is reduced to the collision-resistance. This ends the proof.

Keep in mind that the security proof holds for quantum adversaries. The only difference may arise in the concrete sense where the success probability may be slightly larger because of some non-significant quantum search speed-up.

Remark 3. We now discuss a tricky point related to the function security, which depends on the domain (see Sect. 3.2). The coefficients of σ are in $[k/2]$ on the average. Now, if $k/2$ is so large that σ violates the domain constraint, then the signature may be a vector in $\mathbb{Z}_p^{nm}$ for which the function is easy to invert or have collisions. If this is the case, then an adversary may choose some M, compute B_M, add the corresponding values in the public key, giving $y = \sum_{i \in B_M} y_i \mod p$, then output the inverse of y as a forgery. To avoid this concern, it is sufficient to choose k appropriately. Fortunately, this is always possible by several means. First of all, k is smaller than p for any set of parameters, in particular $k < p/2$ since $k \leq e/2$. Second, we can always trade large p for time (or space); the domain is $D = \{0, \ldots, \lfloor p^{\delta} \rfloor\}^n$ for positive δ.

Strong Unforgeability. In the strong unforgeability game means that a new signature $\sigma \neq \sigma_1$ on a previously signed message M_1 is also a forgery. Our scheme is strongly unforgeable, and the proof is reduced to the collision case. Note that this is not the standard definition, but may be useful in some cryptographic applications.

4.3 Asymptotic Evaluation

We give in this section an asymptotic comparison with the most related work, summarized in Table 1, and a concrete evaluation is given in Sect. 4.4. We compare with (stand-alone) OTS schemes because they can be used as building blocks (without Merkle tree), even beyond digital signatures, and determine the overall efficiency when used with Merkle trees [3,7,9,16,30,36]. We classify the schemes into categories, compare the schemes in each category, then compare the categories together.

In order to provide a reasonable evaluation, the time complexity is measured in terms of the number of evaluations of a general one-way function (OWF), SWIFFT function (our case) or explicit arithmetic operations. To give the reader a more concrete feeling about the timing, it is useful to recall of the following. Although SWIFFT competes with SHA-3 (see Sect. 5.2), we assume that it is multiple times slower. On the other hand, an implementation with the crypto++ library indicates that an exponentiation in a 160-bit group costs about 3 300 hashes [2]. Therefore, it is fair to assume that working in a 224-bit or 256-bit group still requires few thousands hashes of (OWF or SWIFFT) function. Finally, message encoding is twice faster than MD5 hash [3], see Sect. 5.3 for more details.

OWF-based Schemes. Efficient schemes based on general one-way functions essentially have the same time and space complexity. Indeed, the schemes BC, BCC, RR, BTT, etc. [3,6,39,40] are more general and more efficient than L-OTS [25]. We refer to them as BC (category) since they have the same time and space complexity when using 1-CFF, and BC scheme was the first to use the optimal setting. They provide post-quantum security and very efficient time complexity.

M-OTS and BC also have the same space and time complexity essentially. Indeed, we found that $\ell < e$ because $2^\ell \leq \binom{e}{k}$ is necessary to sign ℓ-bit messages, and it is known that $\binom{e}{k} < 2^e$. Therefore, we don't loose much by assuming that $e \sim \tilde{\ell} = \ell + \lfloor \log \ell \rfloor + 1$. M-OTS signature size is $\gamma\tilde{\ell}/2$ bits (on the average) while BC signature is $\gamma e/2$.

It is also relevant to discuss W-OTS, whose signature size (or communication cost) drops linearly in its parameter w while the computational cost grows exponentially. Therefore, any performance gain, if any, is only possible for small values of w. Indeed, a theoretical result [16] claims that W-OTS is most efficient when $w = 2$, and practical one recommends $w = 4$ since it is fast and give relatively short signatures. Another result [43] says that the minimum power consumption cost occurs when $w = 2$. In Table 1, the key generation costs is

$C_{gen} = (2^w - 1)\ell/\tilde{w}$ while the signing and verification are $C_{gen}/2$ on the average. However, W-OTS is unique in that the public key is not needed to be a part of the signature using Merkle tree.

The main disadvantage of such schemes is the space complexity, which is linear in the security parameter. However, the secret key can be reduced to a single seed using a pseudo-random number generator (PRNG), and the public key can be reduced to a single value using some hash function. These are common techniques [4,8,16,36]. Thus, the most challenging limitation for these schemes is the signature size, which we improve significantly in this work.

DLP-based Schemes. The scheme of van Heyst and Pedersen (vHP) (and Groth [20]) essentially provides the best balanced performance using the minimal assumption (DLP). Bellare and Shoup scheme [2] provides the shortest keys and best time complexity. However, they use a collision-resistant hash function and DLP, and did not improve the signature size. Mohassel scheme provides a nice theoretical constructions, but using a target collision-resistant function (TCRF) and without practical advanatge. Zaverucha and Stinson scheme provides the shortest signatures using only DLP, but on the expense of much slower key-generation phase and longer public-keys. None of them is post-quantum.

DLP-based Schemes vs BC. BC is post-quantum secure and provides very efficient time complexity; this is true even compared with any signature scheme. DLP-based ones provide much better space complexity, however, they are much slower and not post-quantum.

Our Scheme vs the Others. Our scheme is time and space efficient, providing the advantages of both approaches. First of all our signature size is independent of the message size, which is not the case for all schemes based on general one-way functions. Consequently, it is much shorter, which improves the main limitation of this category (BC). The key generation and signing algorithms are essentially as efficient as those of BC. The time of key generation is equivalent to e evaluations of an efficient hash function. The only difference is that we use SWIFFT functions, which are competitive with SHA-3 (see Sect. 5.2). Now, the signing is very efficient requiring only encoding, and regular additions of k binary vectors, which is very fast and can be done in parallel. Thus, it is dominated by the encoding algorithm, which is very efficient (see Sect. 5.3). Our verification algorithm requires k additions in $\mathbb{Z}_p^n$, one evaluation of SWIFFT and one comparison, in contrast with k evaluations and k comparisons in BC.

Comparing with DLP-based schemes, our scheme has (i) a much faster key generation, requiring e evaluations, in contrast with at least *two* modular exponentiations in any of the DLP-based schemes; in particular it is much faster ZS one; (ii) a faster signing, requiring regular additions, in contrast with group additions or multiplications; (iii) a much faster verification, requiring k modular additions and *one* function evaluation, in contrast with modular exponentiations. Further more, our scheme is post-quantum assuming the shortest vector

Table 1. Asymptotic comparison: γ and λ are security parameters for OWF and DL, respectively, and ℓ is the message length. Here $\tilde{x} \leq x + \log x + 2$, seed is 128 bits, and $C_{gen} = (2^w - 1)\tilde{\ell/w}$. Arithmetic: count (add binary vectors), add (modular addition), mult (modular multiplication), exp (modular exponentiation).

	Security		Time (function evaluations)			Space (bits)		
Scheme	Func	PQ	Gen	Sign	Ver	SK	PK	σ
L-OTS	OWF	Yes	2ℓ	–	ℓ	seed	$2\ell\gamma$	$\ell\gamma$
M-OTS	OWF	Yes	$\tilde{\ell}$	–	$\tilde{\ell}$	seed	$\tilde{\ell}\gamma$	$\tilde{\ell}\gamma/2$
W-OTS	OWF	Yes	C_{gen}	$\approx C_{gen}/2$	$\approx C_{gen}/2$	seed	$(\tilde{\ell/w})\gamma$	$(\tilde{\ell/w})\gamma$
BC ...	OWF	Yes	$e \approx \tilde{\ell}$	encode	k eval	seed	$e\gamma$	$k\gamma$
vHP	DLP	No	4 exp	2 mult	3 exp	4λ	2λ	2λ
			2 mult	2 add	2 mult			
BS	DLP	No	2 exp	1 mult	1 exp	2λ	λ	2λ
	CRHF							
Moh	DLP,	No	5+exp	2 mult	5+exp	5λ	4λ	2λ
	TCR			4 add				
ZS	DLP	No	$2e$ exp	k add	2 exp	seed	$O(\lambda\ell)$	$\lambda + c$
			e mult	encode	k mult			
This work	SWIFFT	Yes	e eval	k count	1 eval	seed	$en \log p$	$mn \log \frac{k}{2}$
				encode	k add			

problem in idea lattices is hard for quantum computers. However, our signature is much longer because of SWIFFT input size.

Note that there is an encoding step during verification (`Ver`) whenever there is one during signing (`Sign`). However, we do not show it in Table 1 for convenience because it is so efficient that it is dominated by the other operations.

Lattice-Based OTS Schemes. For completeness, we compare with related OTS schemes based on knapsack functions. Lyubashevsky and Micciancio gave a direct construction of OTS scheme, which is asymptotically efficient [26] and whose idea is the following. The scheme is parametrized by integers m, n, k; a ring R; subsets of matrices $\mathcal{H} \subset R^{n \times m}$, $\mathcal{K} \subseteq R^{m \times k}$; and vectors $\mathcal{M} \subseteq R^k$, $\mathcal{S} \subseteq R^m$. The parameters should satisfy certain properties for the scheme to be secure. The underlying hardness assumption is the collision resistance of a linear hash function family mapping $\mathcal{S}$ into R^n. The secret key is a matrix $\mathbf{K} \in R^{m \times k}$ while the public key consists of a matrix $\mathbf{H} \in R^{n \times m}$ along with the matrix product $\mathbf{HK}$. To sigh a message $v \in R^k$, compute $\sigma = \mathbf{K}v$. To verify (σ, v), check that $v \in \mathcal{S}$ and $\mathbf{H}\sigma = \mathbf{HK}v$. Choosing $R = \mathbb{Z}[x]/(x^n + 1)$ and $R = \mathbb{Z}_p$ produces a scheme based on the Ring-SIS and SIS problem, respectively.

There is some connection with our scheme, however, it is not easy to provide a comparison since the paper does not contain concrete or asymptotic evaluation,

or a comparison with any previous work. We can still observe the following. Our signing algorithm is simply the component-wise addition of binary vectors, instead of matrix multiplication, and our verification algorithm requires only one matrix multiplication and ring elements additions instead of two matrix multiplication. Our scheme use a different encoding technique, and able to sign a message without being encoded as a vector. It is not clear how these differences may affect the concrete efficiency. Thus, a future careful comparison is important.

SWIFFT was suggested in [10] to implement the general one-way function in W-OTS, arguing that it has provable security. However, SWIFFT requires input at least 4 times larger than SHA-3 for the same security level, thus making the signature at least 4 times more.

4.4 Concrete Parameters

For a concrete security level and more accurate comparison with other work, we select parameters to sign messages of size $\ell = 224$ bits, and provide *classical* security level of 112 bits; quantum ones may be part of a future work. Therefore, we can use SHA-224 or SHA-256, which are suitable collision-resistant hash functions for digital signatures. While 80-bit security level is disallowed after 2014, 112-bit level is acceptable until 2030 according to NIST recommendation in July 2012 [17]. We will consider two possible implementations of the OWF depending on the output lengths $\gamma = 224$ or 128.

As with elliptic curve cryptography in general, the bit size of the public key believed to be needed for ECDSA is about twice the size of the security level, thus the group order q should have 224 bits.

For SWIFFT with $n = 64$, $m = 16$ and $p = 257$, the known algorithms to invert a function have about 2^{128} time and space complexity, and those to find collisions takes time at least 2^{106} and requires almost as much space. Since the security of our scheme reduces to the collision resistance, we assume randomized hashing [12] is used so that known algorithms to find collisions require at least 2^{112} time complexity. We can always increase n, but it would be too much unless a security level much larger than 112 bits is desired. Finally, the output length is about 512 bits, which "can easily be represented using 528 bits" [27].

Considering 224-bit messages to sign, optimal 1-CFF is obtained by setting $e = 229$ and $k = 114$ so that $\binom{229}{114} > 2^{224}$. However, we consider the minimum k satisfying this inequality, which is $k = 107$. This relaxation mainly allows to have small coefficients and slightly shorter signatures.

Table 2 shows that our scheme provides signatures of 6 144 bits (0.75 KB) on the average, which is the shortest among all schemes based on general one-way functions, including Winternitz, and essentially keeps the same time efficiency. Our signing (and verification) is faster than W-OTS even for its typical value $w = 3$. In any case, our scheme provides multiple times shorter signatures (even using OWF with 128-bit output length), and the most efficient verification algorithm, which is dominated by one evaluation of SWIFFT. Our PK is 14.76 KB, which is the longest among all schemes. This fact is due to the function output length.

Table 2. Concrete evaluation with the most related efficient work. Security level 112 bits; W-OTS parameter $w = 2, 3$; $\ell = 224$ bits.

	Security		Time (function evaluation)			Space (bits)		
Scheme	OWF	PQ	Gen	Sign	Ver	SK	PK	σ
W-OTS	SHA-224	Yes	351	176	176	seed	26,208	26,208
		Yes	553	277	277	seed	17,696	17,696
BC	SHA-224	Yes	229	encode	107	seed	51,296	23,968
vHP	DLP	No	4 exp	2 mult	3 exp	896	448	448
			2 mult	2 add	2 mult			
ZS	DLP	No	458 exp	encode	2 exp,	seed	51,296	241
			229 mult	107 add	107 mult			
Ours	SWIFFT	Yes	229	107 count	107 add,	seed	120,912	6,144
				encode	1 eval			

However, the public key size is not a main concern and can be reduced using a standard technique.

Comparing with DLP-based schemes, our scheme has much better time complexity, but longer signatures due to the SWIFFT input length. The ZS signature size is 241 bits. However, the key generation time takes 458 exponentiations.

Note that our verification algorithm is the fastest one among all schemes considered in this work.

Finally, observe that the signature coefficients are now in $[0, 54]$ on the average, which are considered to be small mod 257. If smaller values are needed, then we use the solutions of Remark 3. For example, we can use $p = 641$, which slightly increases the PK size.

Using AES128-Based OWF. The OWF may be implemented using primitives with smaller output length, using AES-128 for instance. Even in this case, our scheme improves on the most efficient scheme, W-OTS with $w = 3$, where the signature is 10 112 bits (1.65 times longer than our signature), and the key generation is 553 evaluations.

4.5 General CFF

There are essentially two methods to convert a OTS scheme into t-time signature scheme, Merkle hash tree and w-CFF with $w > 1$. The first one was mentioned earlier and is a standard technique, so we only comment on the general type of CFF.

A w-CFF with $w > 1$ allows to turn a OTS scheme into many/multiple-time scheme with relatively efficient time-complexity. However, the keys size becomes so large (hundreds of Mb to sign 1000 messages) that they become impractical. In theory too, it turns out that signing w messages using w instances of a 1-CFF

instead of using a single w-CFF would reduce storage by a factor of $w/\log w$ [48]. The main advantage of 1-CFFs is their simple and efficient optimal construction, which also give easier reduction. A drawback is that the number of public keys to manage increases, but this can be solved using Merkle hash tree.

5 Conclusion and Future Work

We gave a one-time (and fixed-time) signature scheme that keeps the useful properties of those based on general one-way functions (post-quantum security and time-efficiency) while providing shorter signatures, thus improving significantly the main limitation of such schemes. Our verification algorithm is dominated by one evaluation of the hash function, which is a unique feature among all other related schemes. Accordingly, our scheme may be convenient for applications running on devices of limited resources.

Regular schemes (based on DLP and factoring) are insecure against quantum adversaries while our scheme is reduced to the security of SWIFFT functions, which are post-quantum collision-resistant as long as the SVP in ideal lattices is hard for quantum computers. Besides, our scheme is much more time-efficient. On the other hand, our signatures (and keys) are multiple times longer because of the function input and output length.

This work arises several possible extensions. A first one may be implementing the scheme, with or without Merkle tree, in order to provide more concrete evaluations. It is also important to provide a more accurate analysis of the concrete parameters involved in the security level. In particular, SWIFFT security level was estimated using the "best known" classical attack. Therefore, a future work should accomplish more detailed analysis of the parameters, and consider the best known quantum attacks.

An important work would be to improve further the signature (and preferably the keys) size. A straightforward method would be to find some family of functions having some homomorphic properties with small input (and output) size. The current function has input size that is 4 to 8 times larger than the general functions. Another important work would be to design an efficient t-time signature scheme, even for small t, without using Merkle tree.

Acknowledgments. The authors would like to thank Anne Broadbent for her comments on an earlier version of this paper, Andreas Hülsing for helping in the security proof, John Schanck for discussions on lattices, and Fang Song for discussions on the reduction of an earlier version. The authors would also like to thank the anonymous reviewers for their valuable comments.

This work is in part supported by Natural Sciences and Engineering Research Council (NSERC) of Canada, and CryptoWorks21.

Appendix: Other Useful Material

5.1 Technical Lemma

Lemma 1 ([48]). *Let χ_n be the probability distribution on $[n2^\ell]$ defined as $\chi_n = X_1 + \ldots X_n$ where X_i is the uniform distribution on $[2^\ell]$. The min-entropy of χ_n is then at least ℓ bits.*

5.2 SWIFFT Implementation

SWIFFT has efficient software implementations using number-theoretic or modular arithmetic FFT algorithm and its inherent parallelism for implementing multiplication. Importantly, the more the numbers are large the faster the FFT is when compared with the most efficient multiplications algorithms, which is in favour of FFT in the context of cryptography [13]. It was implemented using C and compiled using gcc version 4.1.2 on a PC running under Linux kernel 2.6.18 [27]. Tests on a 3.2 GHz Intel Pentium 4 show that the basic compression function can be evaluated in 1.5 μs on the above system, yielding a throughput close to 40 MB/s in a standard chaining mode of operation. For comparison, SHA-256 was tested on the same system using the highly optimized implementation in openssl version 0.9.8 (using the openssl speed benchmark), yielding a throughput of 47 MB/s when run on 8 KB blocks.

5.3 Encoding Algorithm

Cover-free family constructions and encoding algorithms have been studied in detail, and several approaches have been proposed [6,11,39,40]. In this work, we consider k-uniform 1-CFF where each subset has size k, and any two distinct subsets in $\mathcal{B}$ differ on at least one element. Importantly, 1-CFFs have an optimal construction, which consists of setting $k = \lfloor e/2 \rfloor$. Indeed, there is a simple and very efficient "ranking" algorithm, which is described in [3,48] and due to Cover [11]. The encoding algorithm requires k subtractions (or additions) and about e comparisons using some pre-computation. Here is the pseudo-code by Bicakci, Tung, and Tsudik [3] after their quotation; "To put things in perspective, consider that a single MD5 hash computation requires approximately 500 arithmetic operations. Thus, our mapping (in both directions) costs less than one MD5 hash.".

```
Input : Message m, set E=[1,e], subset size k;
Output: Unique k-subset of E (k-dimensional vector a);
q:=1;
for i=1 to k do
   while m > Binomial(e-q,k-i) do
       m:=m - Binomial(e-q,k-i);
       q:=q+1;
   end while;
```

```
  a[i]:=q;
  q:=q+1;
end for;
```

For example, to encode 3-bit messages, we need a set of size 5 and subsets of size 2 so that the total number of subsets is at least $2^3 = 8$.

```
S( 1 )=  [ 1, 2 ]
S( 2 )=  [ 1, 3 ]
S( 3 )=  [ 1, 4 ]
S( 4 )=  [ 1, 5 ]
S( 5 )=  [ 2, 3 ]
S( 6 )=  [ 2, 4 ]
S( 7 )=  [ 2, 5 ]
S( 8 )=  [ 3, 4 ]
>
```

5.4 Winternitz OTS (W-OTS)

Winternitz [30,32] suggested to Merkle the idea, called W-OTS, of signing several bits simultaneously on the expense of more evaluations of the one-way function, thus trading time for space. Given a small positive integer w, the secret key is $SK := (x_1, \ldots, x_t)$ and public key is $PK := (y_1 || \ldots || y_t)$ where $y_i = f^{2^w-1}(x_i)$. Here $t = \lceil \ell/w \rceil + \lceil \lceil \log 2^w \ell/w \rceil / w \rceil$ and f^k means the k-fold composition of f with itself. To sign an ℓ-bit M, split it into w-bit blocks (including a check sum), $d_1, \ldots, d_t$, then the signature is $(s_1, \ldots, s_t)$ where d_i is treated as an integer and $s_i = f^{d_i}(x_i)$ for $1 \leq i \leq t$. Verifying consists in forming the blocks as before, computing $y_i = f^{2^w-1-d_i}(s_i)$, and accepting if and only it verifies with PK.

5.5 Signing Many Messages

It is easy to transform our OTS scheme into a t-time signature scheme, using the same techniques as for schemes based on OWF. Indeed, the security game can be generalized as follows. Run the key generation algorithm t times to obtain $SK_1, \ldots, SK_t$ and $PK_1, \ldots, PK_t$. Give $\mathcal{A}$ the keys PK_i for $1 \leq i \leq t$ and signatures for at most t adaptive messages, one signature per key, where t is a polynomial in the security parameter. Let $Q = \{M_1, \ldots, M_t\}$ be the set of queries asked by $\mathcal{A}$ and $\{\sigma_1, \ldots, \sigma_t\}$ the corresponding signatures. Finally, $\mathcal{A}$ outputs (M, σ, i). The probability that $\mathcal{A}$ forges a signature is defined to be $\mathrm{PROB}^{\text{uf-cma}}_{\mathrm{DSS},\mathcal{A}} = \Pr[(M \notin Q) \wedge \mathtt{Ver}(PK_i, M, \sigma) = 1]$. Note that the signature now include a number i to indicate the PK with which to verify. However, the standard approach to sign many messages is to use Merkle tree.

References

1. Abdalla, M., Reyzin, L.: A new forward-secure digital signature scheme. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 116–129. Springer, Heidelberg (2000)

2. Bellare, M., Shoup, S.: Two-tier signatures, strongly unforgeable signatures, and fiat-shamir without random oracles. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 201–216. Springer, Heidelberg (2007)
3. Bicakci, K., Tung, B., Tsudik, G.: How to construct optimal one-time signatures. J. Comput. Netw. **43**(3), 339–349 (2003)
4. Bleichenbacher, D., Maurer, U.M.: Directed acyclic graphs, one-way functions and digital signatures. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 75–82. Springer, Heidelberg (1994)
5. Bleichenbacher, D., Maurer, U.M.: Optimal tree-based one-time digital signature schemes. In: Puech, C., Reischuk, R. (eds.) STACS 1996. LNCS, vol. 1046, pp. 361–374. Springer, Heidelberg (1996)
6. Bos, J.N.E., Chaum, D.: Provably unforgeable signatures. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 1–14. Springer, Heidelberg (1993)
7. Buchmann, J., Dahmen, E., Ereth, S., Hülsing, A., Rückert, M.: On the security of the winternitz one-time signature scheme. In: Nitaj, A., Pointcheval, D. (eds.) AFRICACRYPT 2011. LNCS, vol. 6737, pp. 363–378. Springer, Heidelberg (2011)
8. Buchmann, J., Dahmen, E., Hülsing, A.: XMSS - a practical forward secure signature scheme based on minimal security assumptions. In: Yang, B.-Y. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 117–129. Springer, Heidelberg (2011)
9. Buchmann, J., García, L.C.C., Dahmen, E., Döring, M., Klintsevich, E.: CMSS – an improved merkle signature scheme. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 349–363. Springer, Heidelberg (2006)
10. Buchmann, J., Lindner, R., Rückert, M., Schneider, M.: Post-quantum cryptography: lattice signatures. Computing **85**(1–2), 105–125 (2009)
11. Cover, T.: Enumerative source encoding. IEEE Trans. Inf. Theor. **19**(1), 73–77 (1973)
12. Dang, Q.: Randomized Hashing for Digital Signatures. NIST Special Publication 800–106 (2009)
13. David, J.P., Kalach, K., Tittley, N.: Hardware complexity of modular multiplication and exponentiation. IEEE Trans. Comput. **56**(10), 1308–1319 (2007)
14. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theor. **22**(6), 644–654 (1976)
15. Dodis, Y., Katz, J.: Chosen-ciphertext security of multiple encryption. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 188–209. Springer, Heidelberg (2005)
16. Dods, C., Smart, N.P., Stam, M.: Hash based digital signature schemes. In: Smart, N.P. (ed.) Cryptography and Coding 2005. LNCS, vol. 3796, pp. 96–115. Springer, Heidelberg (2005)
17. Barker, E., William Barker, W., Smid, M.: Recommendation for Key Management - Part 1: General (Revision 3), NIST Special Publication 800–57, July 2012
18. Elgamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theor. **31**(4), 469–472 (1985)
19. Even, S., Goldreich, O., Micali, S.: On-line/off-line digital signatures. J. Cryptology **9**(1), 35–67 (1996)
20. Groth, J.: Simulation-sound NIZK proofs for a practical language and constant size group signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006)
21. Grover, L.K.: Quantum mechanics helps in searching for a needle in a haystack. Phys. Rev. Lett. **79**(2), 325–328 (1997)
22. van Heyst, E., Pedersen, T.P.: How to make efficient fail-stop signatures. In: Rueppel, R.A. (ed.) EUROCRYPT 1992. LNCS, vol. 658, pp. 366–377. Springer, Heidelberg (1993)

23. Huang, Q., Wong, D.S., Zhao, Y.: Generic transformation to strongly unforgeable signatures. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 1–17. Springer, Heidelberg (2007)
24. Johnson, D., Menezes, A., Vanstone, S.: The elliptic curve digital signature algorithm (ECDSA). Int. J. Inf. Secur. **1**(1), 36–63 (2001)
25. Lamport, L.: Constructing digital signatures from a one-way function. Technical report, SRI International Computer Science Laboratory (1979)
26. Lyubashevsky, V., Micciancio, D.: Asymptotically Efficient Lattice-Based Digital Signatures. Cryptology ePrint Archive, Report 2013/746 (2013)
27. Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: SWIFFT: a modest proposal for FFT hashing. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 54–72. Springer, Heidelberg (2008)
28. Mehta, M., Harn, L.: Efficient one-time proxy signatures. IEE Proc. Commun. **152**(2), 129–133 (2005)
29. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1996)
30. Merkle, R.C.: Secrecy, Authentication, and Public Key Systems. Ph.D. thesis, Stanford University (1979)
31. Merkle, R.C.: A digital signature based on a conventional encryption function. In: Pomerance, C. (ed.) CRYPTO 1987. LNCS, vol. 293, pp. 369–378. Springer, Heidelberg (1988)
32. Merkle, R.C.: A certified digital signature. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 218–238. Springer, Heidelberg (1990)
33. Micciancio, D.: Personal communication
34. Micciancio, D.: Generalized Compact Knapsacks Cyclic Lattices and Efficient One-Way Functions. Computational Complexity **16**(4), 365–411 (2007). preliminary version in FOCS 2002
35. Mohassel, P.: One-time signatures and chameleon hash functions. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) SAC 2010. LNCS, vol. 6544, pp. 302–319. Springer, Heidelberg (2011)
36. Naor, D., Shenhav, A., Wool, A.: One-Time Signatures Revisited: Have They Become Practical? Cryptology ePrint Archive, Report 2005/442 (2005)
37. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992)
38. Perrig, A.: The BiBa one-time signature and broadcast authentication protocol. In: Proceedings of the 8th ACM Conference on Computer and Communications Security, CCS 2001, pp. 28–37 (2001)
39. Pieprzyk, J., Wang, H., Xing, C.: Multiple-time signature schemes against adaptive chosen message attacks. In: Matsui, M., Zuccherato, R.J. (eds.) SAC 2003. LNCS, vol. 3006, pp. 88–100. Springer, Heidelberg (2004)
40. Reyzin, L., Reyzin, N.: Better than BiBa: short one-time signatures with fast signing and verifying. In: Batten, L.M., Seberry, J. (eds.) ACISP 2002. LNCS, vol. 2384, pp. 144–153. Springer, Heidelberg (2002)
41. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
42. Rohatgi, P.: A compact and fast hybrid signature scheme for multicast packet authentication. In: Proceedings of the 6th ACM Conference on Computer and Communications Security, CCS 1999, pp. 93–100 (1999)

43. Seys, S., Preneel, B.: Power consumption evaluation of efficient digital signature schemes for low power devices. In: IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, vol. 1, pp. 79–86 (2005)
44. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997)
45. Song, F.: A note on quantum security for post-quantum cryptography. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 246–265. Springer, Heidelberg (2014)
46. Stinson, D.R.: Cryptography: Theory and Practice, 3rd edn. Chapman and Hall/CRC, Boca Raton (2005)
47. Vaudenay, S.: One-time identification with low memory. In: Camion, P., Charpin, P., Harari, S. (eds.) EUROCODE 1992. International Centre for Mechanical Sciences, CISM Courses and Lectures, vol. 339, pp. 217–228. Springer, Heidelberg (1992)
48. Zaverucha, G.M., Stinson, D.R.: Short one-time signatures. Adv. Math. Commun. **5**, 473–488 (2011)