The ISC Int'l Journal of Information Security
ISeCure

January 2022, Volume 14, Number 1 (pp. 1–11)
http://www.isecure-journal.org

A New Variant of the Winternitz One Time Signature Based on Graded Encoding Schemes

Hossein Oraei [1,*] and Massoud Hadian Dehkordi [1]

[1] *Cryptography and Data Security Laboratory, School of Mathematics, Iran University of Science & Technology, Narmak, Tehran, Iran.*

ARTICLE INFO.

Article history:
Received: February 11, 2021
Revised: May 22, 2021
Accepted: July 17, 2021
Published Online: September 6, 2021

Keywords:
Digital Signatures, Graded Discrete-Logarithm Problem, Graded Encoding Schemes, Multi-Linear Maps, One-Time Signature Schemes

Type: Research Article

doi: 10.22042/ISECURE.2021.272908.639

dor: 20.1001.1.20082045.2022.14.1.1.1

ABSTRACT

Digital signature schemes are used to guarantee for non-repudiation and authenticity of any kind of data like documents, messages or software. The Winternitz one-time signature (WOTS) scheme, which can be described using a certain number of so-called "function chains", plays an important role in the design of both stateless and stateful many-time signature schemes. The main idea of WOTS scheme is the use of a limited number of function chains, all of which begin at some random values. This work introduces WOTS-GES, a new WOTS type signature scheme in which the need for computing all of the intermediate values of the chains is eliminated. More precisely, to compute each algorithm of the proposed scheme, we only need to calculate one intermediate value. This significantly reduces the number of required operations needed to calculate the algorithms of WOTS-GES. To achieve this results, we have used the concept of "leveled" multilinear maps which is also referred to as graded encoding schemes. We expect these results to increase the efficiency of Winternitz based digital signature schemes.

1 Introduction

Multilinear maps [1] that provide many applications in cryptography, was proposed as an extension of bilinear maps. Unlike bilinear maps, which are known to be constructed using pairing of elliptic curves, there was no method to construct a multilinear maps before the year 2013. This long-standing open problem was finally solved by Garg *et al.* [2]. They proposed the concept of graded encoding schemes, which is an approximate construction of multilinear maps. In their work, ideal lattices are used to build an instantiation of graded encoding schemes which is called GGH13.

Graded encoding schemes are one of the most important cryptographic tools that provide many applications in cryptography such as secret sharing scheme [3], witness encryption [4], multipartite key exchange [2], functional encryption [5], obfuscation [6, 7], aggregate signature scheme [8] and so on. In this paper, we offer another application of graded encoding schemes in signature schemes.

Digital signature schemes [9–11] are useful cryptographic primitives in practice. They provide many uses for data security in a variety of applications, including authenticity and non-repudiation, securing software updates, the use in secure communication protocols SSL/TLS and more. In one-time digital signature schemes the signer is limited to sign a single message [12]. These schemes are important cryptographic primitives that used as the core of the design

* Corresponding author.
Email addresses: hossein_oraei@mathdep.iust.ac.ir, mhadian@iust.ac.ir
ISSN: 2008-2045

of many-time digital signature schemes. One-time signature schemes have other important applications like digital signatures with forward security property [13, 14], network routing protocols [15] and so on.

So far, several techniques have been presented for constructing one-time digital signature schemes, one of the most interesting of which is the Winternitz one-time signature (WOTS) scheme [16]. One-time signature schemes designed using this technique play important roles in the design of both stateless and stateful many-time signature schemes [17–19]. For example, if a Merkle signature scheme is built using a WOTS type signature scheme, there is no need to put the public verification key of WOTS scheme in the signature [14]. In addition, in WOTS type signature schemes, it is possible to make a trade-off between the runtime and the size of signature [20–22].

Using the concept of "function chain", we can give a good description of WOTS scheme. A function chain, using a function (family), produces a chain of values starting from a given point. The main idea of WOTS scheme is the use of a limited number of function chains, that are all calculated starting from random values. These values are in fact the private signing key of WOTS scheme. The public verification key is also the final values of each function chain. Finally, to calculate the signature, the message is mapped to one intermediate value of each chain.

1.1 Related Work

Along the years, several different versions of WOTS scheme have been presented for various purposes. The main idea of the WOTS scheme was first presented in [16]. Using this basic idea, the one-time digital signature schemes [23, 24] were designed using an undetectable, collision resistant hash function. Afterwards, a WOTS type signature scheme was introduced that achieve existential unforgeability under adaptive chosen message attacks (EU-CMA) security using a pseudorandom function family [25].

Under the name WOTS$^+$, Hülsing [26] later introduced a WOTS type signature scheme based on minimal security requirements i.e. undetectable, second-preimage resistant, one-way hash functions. In this scheme, using the bitmasks, the need for collision resistant hash functions has been resolved. The security proof of WOTS$^+$ is tight, which allows the signature size to be reduced compared to the previous WOTS type signature schemes. Therefore, WOTS$^+$ has been given more attention than previous WOTS type signature schemes. For example, the one-time signature which is used in the structure of stateless many-time signature schemes SPHINCS [19] and SPHINCS$^+$ [22] is WOTS$^+$. The variations of WOTS scheme that have been described so far are all vulnerable to multi-target attacks. More precisely, if an adversary has several targets to attack them, then the probability of being able to attack at least one of them is more than he can attack exactly one. There is another WOTS type signature scheme which is referred to as WOTS-T [20]. This scheme is considered as an improved version of WOTS$^+$ that resists against multi-target attacks. The major difference between WOTS-T and WOTS$^+$, which makes WOTS-T resistant to multi-target attacks, is that it uses an addressing scheme. Using this, a new bitmask is produced every time the used hash function is called.

As discussed above, using the concept of function chain, there exists a good description of WOTS scheme. Considering this fact, the difference between all WOTS type signature schemes is in the method that the used function chain is constructed. In the function chain used in each of the WOTS type signature schemes, a function has been used that must be repeated a certain number of times in order to generate the intermediate values of the chains. The total number of production of each intermediate value in the key generation, signature and verification algorithms of this signature schemes is two. Thus, reducing the number of required intermediate values, can reduce the number of operations required for these algorithms.

In this work, we propose WOTS-GES, a new variant of the Winternitz one time signature scheme in which the need for computing all of the intermediate values of the chains is eliminated. More precisely, in each key generation, signature and verification algorithms of the proposed signature scheme, it is necessary to calculate only one intermediate value in each function chain. This significantly reduces the number of required operations needed to calculate these algorithms. To achieve this results, we have used the concept of "leveled" multilinear maps which is also referred to as graded encoding schemes. We also show how the used graded encoding scheme can be instantiated using GGH13.

The rest of the paper is as follows. In Section 2, we review the required concepts. In Section 3, we give description of the generic WOTS scheme. Section 4, describes WOTS-GES based on graded encoding schemes. The security of WOTS-GES is proposed in Section 5. In Section 6, the instantiation of WOTS-GES using GGH13 is discussed and finally in Section 7, conclusions are provided.

2 Preliminaries

This section reviews some required concepts which are used throughout this paper. Most of the content

in this section is devoted to graded encoding schemes. But first of all, we provide a simple definition of multilinear maps [1].

Definition 2.1. Suppose $G_1, \ldots, G_k$ and G_T be groups of prime order q. Then, a k-multilinear map $e : G_1 \times \ldots \times G_k \longrightarrow G_T$ is a map with the following properties:

(1) **Multilinear:** For $a_1, \ldots, a_k \in \mathbb{Z}_q^*$ and arbitrary elements $g_1 \in G_1, \ldots, g_k \in G_k$, it is hold that $e(g_1^{a_1}, \ldots, g_k^{a_k}) = e(g_1, \ldots, g_k)^{a_1 \ldots a_k}$.
(2) **Non-degenerate:** If $g_i \in G_i$, $1 \leq i \leq k$ be a generator of G_i, then $e(g_1, \ldots, g_k)$ is also a generator of G_T.
(3) **Computable:** For arbitrary elements $g_1 \in G_1, \ldots, g_k \in G_k$, the resulting $e(g_1, \ldots, g_k)$ can be computed efficiently.

2.1 Graded Encoding Schemes

Garg *et al.* [2] proposed the concept of graded encoding schemes, which is an approximate construction of multilinear maps as follows:

Definition 2.2. For the family of sets $\mathcal{S} = \{S_i^{(\alpha)} \subset \{0,1\}^* \mid 0 \leq i \leq k, \alpha \in R\}$ in which R is a ring, assume that for each constant $0 \leq i \leq k$, the sets $\{S_i^{(\alpha)} \mid \alpha \in R\}$ be disjoint. Using this assumption, a k-graded encoding scheme $\mathrm{GES}(R, \mathcal{S})$ can be defined with the following procedures:

- $\mathsf{InstGen}(1^\lambda, k)$: The inputs of randomized "instance-generation" procedure are a security parameter λ and also multilinearity parameter k. The corresponding output is $(\mathsf{params}, P_{zt})$ in which P_{zt} and params are a zero-test parameter and description of the k-graded encoding scheme, respectively.
- $\mathsf{Samp}(\mathsf{params})$: The input of randomized "ring sampler" procedure is params. The output of this procedure is $a \in S_0^{(\alpha)}$ that is a "level-zero encoding" (Here $\alpha \in R$ is a random and nearly uniform element).
- $\mathsf{Enc}(\mathsf{params}, i, a)$: The inputs of (possibly randomized) "encoding" procedure are params, an index $i \leq k$ and also a "level-zero" encoding $a \in S_0^{(\alpha)}$. The corresponding output is $u \in S_i^{(\alpha)}$ that is a "level-i" encoding for the same element $\alpha \in R$.

  Obviously for $1 \leq i \leq k$, in order to obtain a level-i encoding, we first get a level-0 encoding of α using the ring sampler procedure and then get a level-i encoding of α using the encoding procedure.
- $\mathsf{Add}(\mathsf{params}, i, u_1, u_2)$: The inputs of "addition" procedure are params, an index $i \leq k$ and two level-i encodings $u_1 \in S_i^{(\alpha_1)}$ and $u_2 \in S_i^{(\alpha_2)}$. This procedure computes $\mathsf{Add}(\mathsf{params}, i, u_1, u_2) = u_1 + u_2 \in S_i^{(\alpha_1+\alpha_2)}$ in which $\alpha_1 + \alpha_2$ is addition in the ring R.

  More generally, for a collection of encodings $u_j \in S_i^{(\alpha_j)}$ where $j = 1, \ldots, h$, it is hold that $u_1 + \cdots + u_h \in S_i^{(\alpha_1+\cdots+\alpha_h)}$.
- $\mathsf{Neg}(\mathsf{params}, i, u_1)$: The inputs of "negation" procedure are params, an index $i \leq k$ and a level-i encodings $u_1 \in S_i^{(\alpha_1)}$. This procedure computes $\mathsf{Neg}(\mathsf{params}, i, u_1) = -u_1 \in S_i^{(-\alpha_1)}$ in which $-\alpha_1$ is negation in the ring R.
- $\mathsf{Mul}(\mathsf{params}, i_1, u_1, i_2, u_2)$: The inputs of "multiplication" procedure are params, indices i_1, i_2 with $i_1 + i_2 \leq k$, a level-i_1 encoding $u_1 \in S_{i_1}^{(\alpha_1)}$ and also a level-i_2 encoding $u_2 \in S_{i_2}^{(\alpha_2)}$. The output of this procedure is $\mathsf{Mul}(\mathsf{params}, i_1, u_1, i_2, u_2) = u_1 \times u_2 \in S_{i_1+i_2}^{(\alpha_1 \cdot \alpha_2)}$ in which $\alpha_1 \cdot \alpha_2$ is multiplication in the ring R and $i_1 + i_2$ is integer addition.

  More generally, for a collection of h encodings $u_j \in S_{i_j}^{(\alpha_j)}$ with $\sum_{j=1}^h i_j \leq k$, it is hold that $u_1 \times \ldots \times u_h \in S_{i_1+\cdots+i_h}^{(\prod_{j=1}^h \alpha_j)}$.
- $\mathsf{isZero}(\mathsf{params}, P_{zt}, u)$: The inputs of "zero-test" procedure are params, P_{zt} and u. The output of this procedure is 1 if $u \in S_k^{(0)}$ and 0 otherwise.
- $\mathsf{Ext}(\mathsf{params}, P_{zt}, u)$: The inputs of "extraction" procedure are params, P_{zt} and $u \in S_k^{(\alpha)}$, The output of this procedure is $s \in \{0,1\}^\lambda$ with the following properties:
  a) For every $\alpha \in R$ and two level-k encodings $u_1, u_2 \in S_k^{(\alpha)}$, it is hold that
  $$\mathsf{Ext}(\mathsf{params}, P_{zt}, u_1) = \mathsf{Ext}(\mathsf{params}, P_{zt}, u_2) \quad (1)$$
  b) The following distribution over $\{0,1\}^\lambda$ is nearly uniform (λ is the security parameter):
  $$\{\mathsf{Ext}(\mathsf{params}, P_{zt}, u) \mid u \in S_k^{(\alpha)}, \alpha \in R\}$$

Garg *et al.* proposed GGH13 which is a k-graded encoding scheme that is parameterized by λ and the multilinearity parameter $k \leq \mathrm{poly}(\lambda)$. GGH13 is a realization of a k-graded encoding scheme with several changes to the above definition. The important change relevant to our signature scheme is in the extraction procedure: the probability that the output of the extraction procedure of GGH13 is the same for two different level-k encodings of α is not 1. Thus, the property a) of the extraction procedure is replaced by a weaker requirement:

a′) Suppose that $a \leftarrow \mathsf{Samp}(\mathsf{params})$ with $a \in S_0^{(\alpha)}$. Then, if the (randomized) encoding procedure is run twice on a to obtain two level-k encodings

$u_1, u_2 \in S_k^{(\alpha)}$:

$$\Pr[\mathsf{Ext}(\mathsf{params}, P_{zt}, u_1) = \mathsf{Ext}(\mathsf{params}, P_{zt}, u_2)] \geq 1 - \mathrm{negl}(\lambda)$$

In this paper, we work with the Definition 2.2, in which the probability that the output of the extraction procedure is the same λ bit string for two different level-k encodings of α is 1. If we want to use the extraction procedure of GGH13, we must consider the negligible probability that $\mathsf{Ext}(\mathsf{params}, P_{zt}, u_1) \neq \mathsf{Ext}(\mathsf{params}, P_{zt}, u_2)$. Thus, for every $\alpha \in R$, we can use $\mathsf{Ext}(\mathsf{params}, P_{zt}, S_k^{(\alpha)})$ to denote this λ bit string.

Remark 2.1. We can assume that a level 1 encoding of $1 \in R$ is published as part of the instance-generation procedure, namely an element $y \in S_1^{(1)}$ [27].

2.1.1 Graded Discrete-Logarithm (GDL) Problem

The analog of discrete logarithm problem in a k-graded encoding scheme $\mathrm{GES}(R, \mathcal{S})$ can be defined as the following: consider a level-i encoding $u_i \in S_i^{(\alpha)}$ in which $1 \leq i \leq k$ and $\alpha \in R$, it must be hard to output a level-j encoding $u_j \in S_j^{(\alpha)}$, where $j < i$. Here, the value i is chosen uniformly at random from the interval $[1, k]$.

More formally, the following experiment can be defined between a challenger $\mathcal{C}$ and an adversary $\mathcal{B}$:

Experiment $\mathsf{Exp}_{\mathrm{GES}}^{\mathsf{GDL}}(\mathcal{B}, \lambda)$:

(1) Using λ and also the multilinearity parameter k, the challenger $\mathcal{C}$ runs $(\mathsf{params}, P_{zt}) \leftarrow \mathsf{InstGen}(1^\lambda, k)$ to get description of a k-graded encoding scheme.
(2) Now, $\mathcal{C}$ firstly runs $a \in S_0^{(\alpha)} \leftarrow \mathsf{Samp}(\mathsf{params})$ to get a level-zero encoding of a random and nearly uniform element $\alpha \in R$. The challenger $\mathcal{C}$ also runs $u_i \leftarrow \mathsf{Enc}(\mathsf{params}, i, a)$ to get a level-i encoding $u_i \in S_i^{(\alpha)}$. Next, $\mathcal{C}$ sends $(\mathsf{params}, P_{zt}, u_i)$ to the adversary $\mathcal{B}$.
(3) Finally, $\mathcal{B}$ outputs a value u_j.
(4) The output is defined to be 1 iff $u_j \in S_j^{(\alpha)}$ and $j < i$.

The success probability of an adversary $\mathcal{B}$ in the experiment $\mathsf{Exp}_{\mathrm{GES}}^{\mathsf{GDL}}(\mathcal{B}, \lambda)$ can be defined as follows.

$$\mathrm{Succ}_{\mathrm{GES}}^{\mathsf{GDL}}(\mathcal{B}, \lambda) = \Pr[\mathsf{Exp}_{\mathrm{GES}}^{\mathsf{GDL}}(\mathcal{B}, \lambda) = 1]$$

We say that the GDL problem is hard, if for each polynomial time adversary $\mathcal{B}$ running in time $\leq t$, $\mathrm{Succ}_{\mathrm{GES}}^{\mathsf{GDL}}(\mathcal{B}, \lambda)$ is a negligible function of λ. In other words,

$$\mathrm{InSec}^{\mathsf{GDL}}(\mathrm{GES}; t, \lambda) := \max_{\mathcal{B}}\{\mathrm{Succ}_{\mathrm{GES}}^{\mathsf{GDL}}(\mathcal{B}, \lambda)\} \quad (2)$$

$$= \mathrm{negl}(\lambda). \quad (3)$$

Note that according to the Remark 2.1, the adversary $\mathcal{B}$ can simply get a level-j' encoding $u_{j'} \in S_{j'}^{(\alpha)}$ in the above experiment, by running the multiplication procedure

$$u_{j'} := u_i \times \underbrace{y \times \ldots \times y}_{j'-i \text{ times}} \in S_{j'}^{(\alpha)}, \qquad \text{where } i < j'$$

2.2 Digital Signature Schemes

Here, we give some required preliminaries about digital signature schemes and also security of these schemes. In the remainder of the paper, we fix some notation in order to simplify the explanation: We denote by $x \xleftarrow{\$} \mathcal{X}$, if x is chosen randomly from the set $\mathcal{X}$. We also write log for $\log_2$.

Definition 2.3. Considering a message space $\mathcal{M}$, a digital signature scheme Dss can be defined using the probabilistic polynomial time (PPT) algorithms $(\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$:

(1) Key generation algorithm $\mathsf{Kg}(1^n)$ takes n as the security parameter and outputs a private signing key sk and a public verification key pk.
(2) Signature algorithm $\mathsf{Sign}(\mathsf{sk}, M)$ takes as input a message M and also the private signing key sk. Then, if $M \in \mathcal{M}$, the algorithm outputs a signature σ for M under sk.
(3) Verification algorithm $\mathsf{Vf}(\mathsf{pk}, \sigma, M)$ takes as input the message M, the signature σ and the public verification key pk. The algorithm outputs 1 iff σ is a valid signature on M under pk.

In a $\mathrm{Dss} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$, for every sk, pk which are outputs of $\mathsf{Kg}(1^n)$ and every $M \in \mathcal{M}$, the following correctness condition must be satisfied:

$$\mathsf{Vf}(M, \mathsf{Sign}(\mathsf{sk}, M), \mathsf{pk}) = 1$$

2.2.1 EU-CMA Security

We now define "existential unforgeability under adaptive chosen message attacks (EU-CMA)" which is the standard security notion for any digital signature scheme $\mathrm{Dss} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$. EU-CMA security can be defined using the following experiment between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. In the following, we use the notation $\mathrm{Dss}(1^n)$ for a $\mathrm{Dss} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$ with the security parameter n.

Experiment $\mathsf{Exp}_{\mathrm{Dss}(1^n)}^{\mathsf{EU\text{-}CMA}}(\mathcal{A})$:

(1) $\mathcal{C}$ runs the key generation algorithm $\mathsf{Kg}(1^n)$ to generate a key pair $(\mathsf{sk}, \mathsf{pk})$ and sends the public verification key pk to $\mathcal{A}$.
(2) Suppose that $\mathsf{Sign}(\mathsf{sk}, \cdot)$ be an oracle which for every message $M \in \mathcal{M}$, returns the signature $\mathsf{Sign}(\mathsf{sk}, M)$. Here, we denote by $\mathcal{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}$ the oracle access to $\mathsf{Sign}(\mathsf{sk}, \cdot)$ for $\mathcal{A}$. Let also that $\{(M_i, \sigma_i)\}_{i=1}^q$ be the query-answer pairs of $\mathsf{Sign}(\mathsf{sk}, \cdot)$.
(3) The adversary then outputs $(M^\star, \sigma^\star)$.
(4) The output of $\mathsf{Exp}^{\mathsf{EU\text{-}CMA}}_{\mathrm{Dss}(1^n)}(\mathcal{A})$ is defined to be 1 iff $\mathsf{Vf}(M^\star, \sigma^\star, \mathsf{pk}) = 1$ and $M^\star \notin \{M_i\}_{i=1}^q$.

We define the success probability of $\mathcal{A}$ in the experiment $\mathsf{Exp}^{\mathsf{EU\text{-}CMA}}_{\mathrm{Dss}(1^n)}(\mathcal{A})$ as

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

Now, we give the definition of EU-CMA security as follows.

Definition 2.4. Let $n, t, q \in \mathbb{N}$ and $t, q = \mathrm{Poly}(n)$. We say that a signature scheme $\mathrm{Dss} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$ is EU-CMA-secure, if for all PPT adversaries $\mathcal{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}$ running in time at most t and making at most q queries, the maximum success probability $\mathrm{InSec}^{\mathsf{EU\text{-}CMA}}(\mathrm{Dss}(1^n); t, q)$ is a negligible function of n:

$$\mathrm{InSec}^{\mathsf{EU\text{-}CMA}}(\mathrm{Dss}(1^n); t, q) := \max_{\mathcal{A}}\{\mathrm{Succ}^{\mathsf{EU\text{-}CMA}}_{\mathrm{Dss}(1^n)}(\mathcal{A})\} \quad (4)$$

$$= \mathrm{negl}(n). \quad (5)$$

Note that for any one-time signature (OTS) scheme, the number of oracle queries of $\mathcal{A}$ in the above experiment is restricted to one, i.e. $q = 1$.

3 Description of the Generic WOTS

Here, we give description of the generic WOTS scheme. Before defining WOTS, we first recall the definition of function chain.

Definition 3.1. Let $n \in N$, $\mathcal{D}$ and $\mathcal{K}$ be the security parameter, domain and key space, respectively such that the length of every $X \in \mathcal{D}$ and $\mathsf{ck} \in \mathcal{K}$ be polynomial in n. A function chain $\mathcal{C} = (\mathcal{I}, \mathcal{E})$ consists of the following PPT algorithms:

- The initialization algorithm $\mathcal{I}(1^n, \lambda)$ takes as input a chain length parameter $\lambda \in \mathbb{N}$ and also the security parameter n and outputs a public value $\mathsf{ck} \in \mathcal{K}$ which is called "chain key".
- The evaluation algorithm $\mathcal{E}^{i,j}_{\mathsf{ck}}(X)$ takes as input a public chain key ck, an interval $i, j \in \mathbb{N}$, $0 \le i < j \le \lambda$, and a value $X \in \mathcal{D}$ which is the ith value of the chain and outputs $Y \in \mathcal{D}$, the jth value of the chain.

For every $n, \lambda \in \mathbb{N}$, every $\mathsf{ck} \in \mathcal{K}$ which is output of $\mathcal{I}(1^n, \lambda)$, every $i, j, m \in \mathbb{N}$ such that $0 \le i \le j \le m \le \lambda$ and every $X \in \mathcal{D}$, it must hold that

$$\mathcal{E}^{j,m}_{\mathsf{ck}}(\mathcal{E}^{i,j}_{\mathsf{ck}}(X)) = \mathcal{E}^{i,m}_{\mathsf{ck}}(X)$$

We now describe the generic W-OTS using a function chain $\mathcal{C} = (\mathcal{I}, \mathcal{E})$. This digital signature is parameterized by

- m : the binary message length.
- n : the security parameter.
- $w > 1$: the Winternitz parameter. This parameter determines the time-memory trade-off.
- l: the number of elements in a W-OTS signature, public verification key and private signing key, which is computed as

$$l_1 = \lceil \frac{m}{\log(w)} \rceil,\ l_2 = \lfloor \frac{\log(l_1(w-1))}{\log(w)} \rfloor + 1,\ l = l_1 + l_2$$

Key Generation Algorithm $(\mathsf{Kg}(1^n))$: On input of the security parameter n, this algorithm chooses the private signing key $\mathsf{sk} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_l) \xleftarrow{\$} \mathcal{D}^l$. Next, a public chain key ck is obtained using the initialization algorithm $\mathcal{I}(1^n, \lambda)$ of the function chain. Finally, the public verification key pk can be computed as

$$\mathsf{pk} = (\mathsf{pk}_0, \mathsf{pk}_1, \ldots, \mathsf{pk}_l) = (\mathsf{ck}, \mathcal{E}^{0,w-1}_{\mathsf{ck}}(\mathsf{sk}_1), \ldots, \mathcal{E}^{0,w-1}_{\mathsf{ck}}(\mathsf{sk}_l))$$

Signature Algorithm $(\mathsf{Sign}(\mathsf{sk}, M))$: This algorithm takes as input a message $M \in \{0,1\}^n$ and the private signing key sk. Firstly, the base w representation of M is computed, i.e. $M = (b_1, \ldots, b_{l_1})$ such that $b_i \in \{0, \ldots, w-1\}$. Next, the checksum

$$C = \sum_{i=1}^{l_1}(w - 1 - b_i)$$

and also its base w representation $C = (b_{l_1+1}, \ldots, b_l)$ such that $b_i \in \{0, \ldots, w-1\}$, is computed (Note that $C \le l_1(w-1)$). Now, the signature is computed as

$$\sigma = (\sigma_1, \ldots, \sigma_l) = (\mathcal{E}^{0,b_1}_{\mathsf{ck}}(\mathsf{sk}_1), \ldots, \mathcal{E}^{0,b_l}_{\mathsf{ck}}(\mathsf{sk}_l))$$

Verification Algorithm $(\mathsf{Vf}(\mathsf{pk}, \sigma, M))$: This algorithm takes as input the message M, the signature σ and also the public verification key pk. Firstly, the $b_i, 1 \le i \le l$ are computed as above. Next, if the following comparison holds, the verification algorithm returns **true** and **false** otherwise:

$$(\mathsf{pk}_1, \ldots, \mathsf{pk}_l) \stackrel{?}{=} (\mathcal{E}^{b_1,w-1}_{\mathsf{ck}}(\sigma_1), \ldots, \mathcal{E}^{b_l,w-1}_{\mathsf{ck}}(\sigma_l))$$

4 WOTS-GES

Here, we propose our digital signature scheme WOTS-GES(k, m) based on a k-graded encoding scheme $\mathrm{GES}(R, \mathcal{S})$ with the security parameter λ. As mentioned before, this signature scheme is a new variant of WOTS scheme. Like other versions of WOTS, WOTS-GES(k, m) is parameterized by

- m : the binary message length.
- $w > 1$: the Winternitz parameter. Here we suppose that $w-1$ is equal to the multilinearity parameter k of the k-graded encoding scheme, i.e. $w - 1 = k$.
- l: This parameter is calculated using the parameters m and w, as described in the previous section.

Please note that according to the Remark 2.1, we can consider that there is a level 1 encoding of 1, i.e. $1_1 \in S_1^{(1)}$. It is assumed that in the pre-computation phase, the encoding procedure $\mathsf{Enc}(\mathsf{params}, i, 1_1)$ is run to obtain the level-i encoding $1_i \in S_i^{(1)}$, where $2 \le i \le k$.

Key Generation Algorithm ($\mathsf{Kg}(\mathrm{GES}(R,\mathcal{S}))$): This algorithm takes as input the description of the k-graded encoding scheme $\mathrm{GES}(R,\mathcal{S})$. Then, the randomized ring sampler procedure $\mathsf{Samp}(\mathsf{params})$ is run to obtain l level-zero encodings $\mathsf{a}_j \in S_0^{(\alpha_j)}$, where $\alpha_1, \ldots, \alpha_l \in R$ are random and nearly uniform elements and $1 \le j \le l$. The private signing key $\mathsf{sk} = (\mathsf{a}_1, \ldots, \mathsf{a}_l)$ consists of this level-zero encodings.

Next for $1 \le j \le l$, the key generation algorithm runs the encoding procedure $\mathsf{Enc}(\mathsf{params}, k, \mathsf{a}_j)$ to get l level-k encodings $\mathsf{u}_{jk} \in S_k^{(\alpha_j)}$. Finally, the extraction procedure is run to obtain $\mathsf{pk}_j = \mathsf{Ext}(\mathsf{params}, P_{zt}, \mathsf{u}_{jk})$. Now, the public verification key pk is defined as $\mathsf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_l)$. The key generation algorithm is shown in Figure 1.

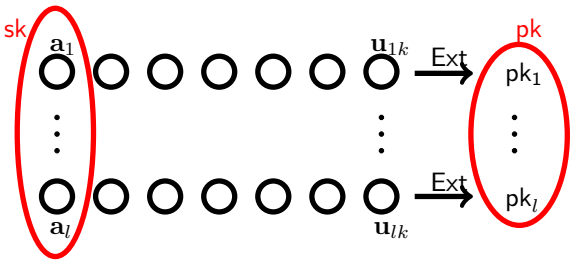

Figure 1. A schematic representation of key generation algorithm

Signature Algorithm ($\mathsf{Sign}(\mathsf{sk}, M)$): This algorithm takes as input a message $M \in \{0,1\}^n$ and also the private signing key $\mathsf{sk} = (\mathsf{a}_1, \ldots, \mathsf{a}_l)$. Firstly, the base w representation of M is computed, i.e. $M = (b_1, \ldots, b_{l_1})$ such that $b_i \in \{0, \ldots, w-1\}$. Next, the checksum

$$C = \sum_{i=1}^{l_1} (w - 1 - b_i)$$

and also its base w representation $C = (b_{l_1+1}, \ldots, b_l)$ such that $b_i \in \{0, \ldots, w-1\}$, is computed. Afterwards for $1 \le j \le l$, the signature algorithm runs the encoding procedure $\mathsf{Enc}(\mathsf{params}, b_j, \mathsf{a}_j)$ to get the level-b_j encodings $\mathsf{u}_{jb_j} \in S_{b_j}^{(\alpha_j)}$. Now, the signature σ is defined as

$$\sigma = (\sigma_1, \ldots, \sigma_l) = (\mathsf{u}_{1b_1}, \ldots, \mathsf{u}_{lb_l})$$

The signature algorithm is shown in Figure 2. Let $B = M \| C$, then we can conclude from the checksum that if $M' \neq M$ be any other message, the corresponding B' consists of at least one $b'_j < b_j$, where $1 \le j \le l$.

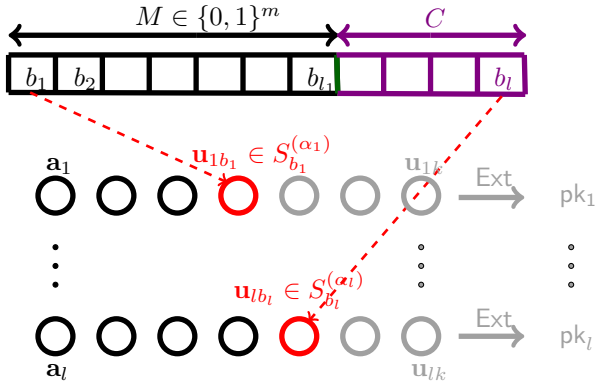

Figure 2. A schematic representation of signature algorithm

Verification Algorithm ($\mathsf{Vf}(\mathsf{pk}, \sigma, M)$): This algorithm which is shown in Figure 3, takes as input the message M, the signature σ and also the public verification key pk. In this algorithm for $1 \le j \le l$:

(1) Firstly, the b_js are computed as described above.
(2) Then, the verification algorithm runs the multiplication procedure $\mathsf{Mul}(\mathsf{params}, b_j, \mathsf{u}_{jb_j}, k - b_j, 1_{k-b_j})$ to compute the level-k encoding $\mathsf{u}'_{jk} \in S_k^{(\alpha_j)}$.
(3) Finally, the extraction procedure is run to obtain $\mathsf{pk}'_j = \mathsf{Ext}(\mathsf{params}, P_{zt}, \mathsf{u}'_{jk})$.

Now, if the following comparison holds, the verification algorithm returns **true** and **false** otherwise:

$$(\mathsf{pk}_1, \ldots, \mathsf{pk}_l) \stackrel{?}{=} (\mathsf{pk}'_1, \ldots, \mathsf{pk}'_l)$$

5 Security of WOTS-GES

Here, we prove the security of WOTS-GES. We explain how an adversary for WOTS-GES can be used to construct an adversary to solve the GDL problem. More formally, the GDL problem is reduced to the EU-CMA security of WOTS-GES.

Lemma 1. *Let $k, m \in \mathbb{N}$. Then, if there is any PPT adversary $\mathcal{A}$ who can break the proposed digital signature scheme* $\mathrm{WOTS\text{-}GES}(k,m)$*, then there exists a PPT adversary $\mathcal{B}$ that is a solver for the GDL problem such that*

$$\mathrm{Succ}^{\mathsf{EU\text{-}CMA}}_{\mathrm{Dss}(1^n)}(\mathcal{A}) \le kl \cdot \mathrm{Succ}^{\mathsf{GDL}}_{\mathrm{GES}}(\mathcal{B}, \lambda). \qquad (6)$$

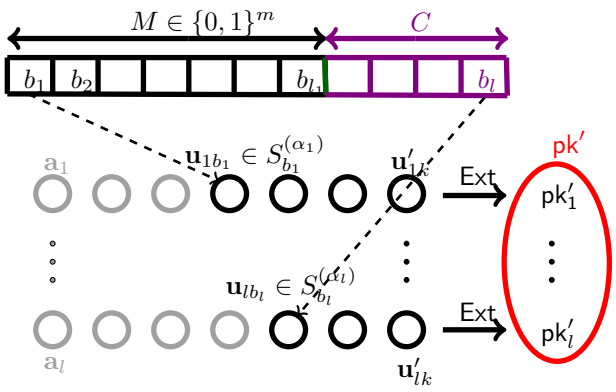

Figure 3. A schematic representation of verification algorithm

Proof. Consider a PPT adversary $\mathcal{A}$ which acts according to the Experiment $\mathsf{Exp}^{\mathsf{EU\text{-}CMA}}_{\mathrm{Dss}(1^n)}(\mathcal{A})$ against the security of WOTS-GES(k,m), such that his success probability $\mathrm{Succ}^{\mathsf{EU\text{-}CMA}}_{\mathrm{Dss}(1^n)}(\mathcal{A}) = \varepsilon_{\mathcal{A}}$ is non-negligible. In the rest of the proof, we will define an adversary $\mathcal{B}$ which acts according to the Experiment $\mathsf{Exp}^{\mathsf{GDL}}_{\mathrm{GES}}(\mathcal{B},\lambda)$ to solve the GDL problem in polynomial time with a non-negligible success probability $\mathrm{Succ}^{\mathsf{GDL}}_{\mathrm{GES}}(\mathcal{B},\lambda) = \varepsilon_{\mathcal{B}}$ and uses $\mathcal{A}$ as a sub-routine (Note that we have $n = \lambda$):

(1) Based on λ and the multilinearity parameter k, the challenger of the Experiment $\mathsf{Exp}^{\mathsf{GDL}}_{\mathrm{GES}}(\mathcal{B},\lambda)$ (that is $\mathcal{C}$) runs $(\mathsf{params}, P_{zt}) \leftarrow \mathsf{InstGen}(1^\lambda, k)$ to get an explanation of a k-graded encoding scheme GES$(R,\mathcal{S})$. Now, the challenger $\mathcal{C}$ firstly runs $a \leftarrow \mathsf{Samp}(\mathsf{params})$ to obtain a level-zero encoding $a \in S_0^{(\alpha)}$, where $\alpha \in R$ is a random and nearly uniform element. Then, $\mathcal{C}$ runs $u_i \leftarrow \mathsf{Enc}(\mathsf{params}, i, a)$ to get a level-i encoding $u_i \in S_i^{(\alpha)}$. Next, $\mathcal{C}$ sends $(\mathsf{params}, P_{zt}, u_i)$ to the adversary $\mathcal{B}$.

(2) Now, $\mathcal{B}$ is used as a challenger for $\mathcal{A}$ in the Experiment $\mathsf{Exp}^{\mathsf{EU\text{-}CMA}}_{\mathrm{Dss}(1^n)}(\mathcal{A})$. So, $\mathcal{B}$ executes the WOTS-GES key generation algorithm $\mathsf{Kg}(\mathrm{GES}(R,\mathcal{S}))$ to obtain a private signing key $\mathsf{sk} = (\mathsf{a}_1,\ldots,\mathsf{a}_l)$, where $\mathsf{a}_j \in S_0^{(\alpha_j)}$ are l level-zero encodings and $\alpha_1,\ldots,\alpha_l \in R$ are random and nearly uniform elements and also a public verification key $\mathsf{pk} = (\mathsf{pk}_1,\ldots,\mathsf{pk}_l)$. Suppose that (M,σ) be the query-answer pair of $\mathsf{Sign}(\mathsf{sk},\cdot)$ in the Step 2 of the experiment $\mathsf{Exp}^{\mathsf{EU\text{-}CMA}}_{\mathrm{Dss}(1^n)}(\mathcal{A})$ and $B = M\|C = (b_1,\ldots,b_l)$. Let also that $(M^\star,\sigma^\star)$ be the output of the adversary $\mathcal{A}$ in the Step 3 of this experiment and $B^\star = M^\star\|C^\star = (b_1^\star,\ldots,b_l^\star)$. Because of the checksum, the corresponding $B^\star$ of the successful forgery $(M^\star,\sigma^\star)$ must contain at least one $b_\gamma^\star < b_\gamma$, where $1 \le \gamma \le l$. More precisely, the γ-th components of $\sigma = (\sigma_1,\ldots,\sigma_l)$ and $\sigma^\star = (\sigma_1^\star,\ldots,\sigma_l^\star)$ i.e. σ_γ and $\sigma_\gamma^\star$ are a level-b_γ encoding $\sigma_\gamma \in S_{b_\gamma}^{(\alpha_\gamma)}$ and a level-$b_\gamma^\star$ encoding $\sigma_\gamma^\star \in S_{b_\gamma^\star}^{(\alpha_\gamma)}$, respectively, where $1 \le \gamma \le l$. In the following, the adversary $\mathcal{B}$ tries to conjecture the location of σ_γ and place the level-i encoding $u_i \in S_i^{(\alpha)}$ there. Hence, he will reply the signature query and finally extract a level-j encoding $u_j \in S_j^{(\alpha)}$ using the successful forgery $\sigma^\star$, where $0 \le j < i$:

(a) The adversary $\mathcal{B}$ selects the position of a component of the private signing key $\mathsf{sk} = (\mathsf{a}_1,\ldots,\mathsf{a}_l)$ choosing the index $1 \le \gamma' \le l$ uniformly at random.

(b) $\mathcal{B}$ considers the level-i encoding $u_i \in S_i^{(\alpha)}$ challenge as the level-i encoding of an unknown level-zero encoding $\mathsf{a}'_{\gamma'} = a$. Next, $\mathcal{B}$ runs the multiplication procedure $\mathsf{Mul}(\mathsf{params}, i, u_i, k-i, 1_{k-i})$ to compute the level-k encoding $\mathsf{u}_k \in S_k^{(\alpha)}$. Afterwards, $\mathcal{B}$ runs the extraction procedure to compute $\mathsf{pk}'_{\gamma'} = \mathsf{Ext}(\mathsf{params}, P_{zt}, \mathsf{u}_k)$ Consequently, the manipulated public verification key pk' is obtained as $\mathsf{pk}' = (\mathsf{pk}_1,\ldots,\mathsf{pk}'_{\gamma'},\ldots,\mathsf{pk}_l)$. Note that the private signing key is also changed as $\mathsf{sk} = (\mathsf{a}_1,\ldots,\mathsf{a}'_{\gamma'},\ldots,\mathsf{a}_l)$, where $\mathsf{a}'_{\gamma'}$ is unknown. Now, $\mathcal{B}$ sends the manipulated public verification key pk' to $\mathcal{A}$ (the start of the Experiment $\mathsf{Exp}^{\mathsf{EU\text{-}CMA}}_{\mathrm{Dss}(1^n)}(\mathcal{A})$).

(c) Note that $\mathcal{B}$ only knows the level-j' encodings $u_{j'} \in S_{j'}^{(\alpha)}$, where $i \le j' \le k$ as he can run the multiplication procedure $\mathsf{Mul}(\mathsf{params}, i, u_i, j'-i, 1_{j'-i})$ to compute the level-j' encoding $u_{j'} \in S_{j'}^{(\alpha)}$. So, $\mathcal{B}$ can only answer the $\mathcal{A}$'s query M, if $i \le b_{\gamma'}$.

(d) Also, the successful forgery $(M^\star,\sigma^\star)$ is only helpful if $b^\star_{\gamma'} < i$. In this case, the adversary $\mathcal{B}$ announces the level-$b^\star_{\gamma'}$ encoding $u_{b^\star_{\gamma'}} \in S_{b^\star_{\gamma'}}^{(\alpha)}$ as its output (Step 3 of the Experiment $\mathsf{Exp}^{\mathsf{GDL}}_{\mathrm{GES}}(\mathcal{B},\lambda)$).

In the following, the success probability of the adversary $\mathcal{B}$ is calculated: As we saw in Step 2c, $\mathcal{B}$ can only answer the $\mathcal{A}$'s query M, if $i \le b_{\gamma'}$. To make computation of the success probability easier, we only consider a certain success case, i.e. $i = b_{\gamma'}$. As i was selected randomly with uniform distribution from the interval $[1,k]$, the case happens with probability k^{-1}.

We also pointed out that the corresponding $B^\star$ of the successful forgery $(M^\star,\sigma^\star)$ must contain at least one $b_\gamma^\star < b_\gamma$, where $1 \le \gamma \le l$. This happens for $\gamma = \gamma'$ with probability l^{-1} . Thus we have $b^\star_{\gamma'} < b_{\gamma'}$. Consequently, we conclude that $b^\star_{\gamma'} < i$ with probability $(kl)^{-1}$ and therefore the condition in Step 2d

is fulfilled. Hence, the success probability of the adversary $\mathcal{A}$ can be bounded as follows:

$$\varepsilon_{\mathcal{A}} \leq kl \cdot \varepsilon_{\mathcal{B}}.$$

Note that because of the Equation 1 of the extraction procedure, changing the public verification key generation method to place our challenge, does not change the public verification key. More precisely, if we choose either the key generation algorithm of WOTS-GES(k, m) or the method which is used in the proof to produce public verification key, we obtain an equal value for this key. Thus, the proof is completed. □

We now conclude the following theorem using Lemma 1:

Theorem 1. *Suppose that* $k, m \in \mathbb{N}$. *Then, we can bound the insecurity of* WOTS-GES *against an* EU-CMA *attack by*

$$\mathrm{InSec}^{\text{EU-CMA}}(\text{WOTS-GES}(k,m); t, 1) \leq kl \cdot \mathrm{InSec}^{\mathsf{GDL}}(\text{GES}; t', \lambda). \tag{7}$$

with $t' = t + 5l$.

Proof. Firstly note that the Equation 7 can be simply derived from Equation 6 and also from Definitions 2 and 5. The time $t' = t + 5l$ is also the maximum runtime required by the adversary $\mathcal{A}$ (which behaves according to the Definition 2.4) plus the time required to execute the three algorithms of WOTS-GES once (follow the proof of Lemma 1). □

6 Instantiation using GGH13

To use WOTS-GES, graded encoding scheme GES$(R, \mathcal{S})$ must be instantiated. In this section, we discuss how GES$(R, \mathcal{S})$ can be instantiated using GGH13.

The graded encoding scheme GGH13 is parameterized by λ and also multilinearity parameter $k \leq \text{poly}(\lambda)$. Using these parameters, consider the cyclotomic ring $R = \frac{\mathbb{Z}}{<X^n+1>}$, in which $n = \tilde{O}(k\lambda^2)$ is a power of 2. Also, let that the modulus $q = 2^{k\lambda}$ defines the quotient ring $R_q = \frac{R}{qR}$. Finally, consider the quotient ring $QR = \frac{R}{\mathcal{I}}$ in which $\mathcal{I} =< g >$ is a principal prime ideal and g is a secret short vector drawn from the discrete Gaussian distribution $g \leftarrow D_{\mathbb{Z}^n,\sigma}$ in which $\sigma = \tilde{O}(\sqrt{n})$. There is also another secret vector $z \in R_q$, that selected uniformly at random.

In the graded encoding scheme GGH13, the quotient ring $QR = \frac{R}{\mathcal{I}}$ plays the role of ring R in Definition 2.2. More precisely, elements of QR are what are encoded.

A level-zero encoding of an arbitrary cost $r + \mathcal{I} \in QR$ is a short vector of $r + \mathcal{I}$. It can be proved that the size of level-zero encodings is bounded by λn^2 (with high probability) [27]. On the other hand, the private signing key $\mathsf{sk} = (\mathsf{a}_1, \ldots, \mathsf{a}_l)$ of the signature scheme WOTS-GES consists of l level-zero encodings. Consequently, the size of private signing key sk is bounded by $l\lambda n^2$.

Also, a level-i encoding of a cost $r+\mathcal{I} \in QR$, where $1 \leq i \leq k$, is a vector of the form $\frac{c}{z^i} \in R_q$ in which $c \in r + \mathcal{I}$ and $\|c\| < q^{\frac{1}{8}}$. Thus, the size of $\frac{c}{z^i} \in R_q$ is bounded by qn. On the other hand, we know that the signature $\sigma = (\sigma_1, \ldots, \sigma_l)$ of a given message M using WOTS-GES, consists of l level-i encodings, where $0 \leq i \leq k$. Therefore, signature σ consists of l level-i encodings which size of each is at most either λn^2 or qn.

Finally, as described in Definition 2.2, the output of the extraction procedure is a λ bit string. On the other hand, the public verification key $\mathsf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_l)$ is made up of l extraction procedure outputs. Thus, the size of the public verification key is $l\lambda$ bits.

In [28], GGHLite, an efficient version of GGH13 is presented in which the size of some parameters has been improved. Thus, instantiating the used graded encoding scheme of WOTS-GES using GGHLite can improve the efficiency of WOTS-GES.

Table 1. Comparison of the computational complexities

Step	WOTS schemes [16, 20, 23–26]	proposed scheme
Public verification key generation	$lk \cdot \mathrm{T_{fc}}$	$l \cdot (\mathrm{T_{enc}} + \mathrm{T_{ext}})$
Signature algorithm	$(\sum_{i=1}^{l} b_i) \cdot \mathrm{T_{fc}}$	$l \cdot \mathrm{T_{enc}}$
Verification algorithm	$(\sum_{i=1}^{l} (k - b_i)) \cdot \mathrm{T_{fc}}$	$l \cdot (\mathrm{T_{mul}} + \mathrm{T_{ext}})$

7 Conclusion

Here, we provide a comparison for the number of operations required by the key generation, signature and verification algorithms of WOTS-GES scheme and other WOTS scheme variants in the literature [16, 20, 23–26]. We have summarized the results in Table 1.

In this table, we have assumed that the Winternitz parameter minus one is equal to the multilinearity parameter k of the used k-graded encoding scheme, i.e. $w - 1 = k$. We have also used the following notations to analyze the complexities of the proposed scheme:

- $\mathrm{T_{fc}}$: The time required to execute one iteration of the used function chain.
- $\mathrm{T_{enc}}$: The time required to execute the encoding procedure of the used k-graded encoding scheme.
- $\mathrm{T_{ext}}$: The time required to execute the extraction procedure of the used k-graded encoding scheme.

- T_{mul}: The time required to execute the multiplication procedure of the used k-graded encoding scheme.

From the comparison in the table, we can see that the number of operations required by the three algorithms of WOTS-GES is less than that of other WOTS scheme variants. In [29], the first practical implementation of graded encoding schemes is presented in which the efficiency of GGHLite has also been improved. Using our results along with the practical implementations of graded encoding schemes, we can obtain an efficient one-time digital signature scheme for various applications [13–15].

References

[1] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324(1):71–90, 2003.

[2] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–17. Springer, 2013.

[3] Massoud Hadian Dehkordi and Hossein Oraei. How to construct a verifiable multi-secret sharing scheme based on graded encoding schemes. *IET Information Security*, 13(4):343–351, 2019.

[4] Sanjam Garg, Craig Gentry, Shai Halevi, and Daniel Wichs. On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. *Algorithmica*, 79(4):1353–1373, 2017.

[5] Huijia Lin and Stefano Tessaro. Indistinguishability obfuscation from trilinear maps and block-wise local prgs. In *Annual International Cryptology Conference*, pages 630–660. Springer, 2017.

[6] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM Journal on Computing*, 45(3):882–929, 2016.

[7] Prabhanjan Ananth, Aayush Jain, and Amit Sahai. Robust transforming combiners from indistinguishability obfuscation to functional encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 91–121. Springer, 2017.

[8] Susan Hohenberger, Amit Sahai, and Brent Waters. Full domain hash from (leveled) multilinear maps and identity-based aggregate signatures. In *Annual Cryptology Conference*, pages 494–512. Springer, 2013.

[9] Jia Yu, Hui Xia, Huawei Zhao, Rong Hao, Zhangjie Fu, and Xiangguo Cheng. Forward-secure identity-based signature scheme in untrusted update environments. *Wireless Personal Communications*, 86(3):1467–1491, 2016.

[10] Özgür Dagdelen, David Galindo, Pascal Véron, Sidi Mohamed El Yousfi Alaoui, and Pierre-Louis Cayrel. Extended security arguments for signature schemes. *Designs, Codes and Cryptography*, 78(2):441–461, 2016.

[11] Daofeng Li, Haiqiang Chen, Cheng Zhong, Taoshen Li, and Feng Wang. A new self-certified signature scheme based on ntrus ing for smart mobile communications. *Wireless Personal Communications*, 96(3):4263–4278, 2017.

[12] Leslie Lamport. Constructing digital signatures from a one-way function. Technical report, Citeseer, 1979.

[13] Tal Malkin, Daniele Micciancio, and Sara Miner. Efficient generic forward-secure signatures with an unbounded number of time periods. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 400–417. Springer, 2002.

[14] Johannes Buchmann, Erik Dahmen, and Andreas Hülsing. Xmss-a practical forward secure signature scheme based on minimal security assumptions. In *International Workshop on Post-Quantum Cryptography*, pages 117–129. Springer, 2011.

[15] Ralf Hauser, Tony Przygienda, and Gene Tsudik. Reducing the cost of security in link-state routing. In *Proceedings of SNDSS'97: Internet Society 1997 Symposium on Network and Distributed System Security*, pages 93–99. IEEE, 1997.

[16] Ralph C Merkle. A certified digital signature. In *Conference on the Theory and Application of Cryptology*, pages 218–238. Springer, 1989.

[17] Andreas Hülsing, Christoph Busold, and Johannes Buchmann. Forward secure signatures on smart cards. In *International Conference on Selected Areas in Cryptography*, pages 66–80. Springer, 2012.

[18] Andreas Hülsing, Lea Rausch, and Johannes Buchmann. Optimal parameters for xmss mt. In *International Conference on Availability, Reliability, and Security*, pages 194–208. Springer, 2013.

[19] Daniel J Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. Sphincs: practical stateless hash-based signatures. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 368–397. Springer, 2015.

[20] Andreas Hülsing, Joost Rijneveld, and Fang Song. Mitigating multi-target attacks in hash-based signatures. In *Public-Key Cryptography–*

PKC 2016, pages 387–416. Springer, 2016.

[21] Jean-Philippe Aumasson and Guillaume Endignoux. Improving stateless hash-based signatures. In *Cryptographers' Track at the RSA Conference*, pages 219–242. Springer, 2018.

[22] Jean-Philippe Aumasson, Daniel J Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Andreas Hülsing, Panos Kampanakis, Stefan Kölbl, Tanja Lange, et al. Sphincs+. 2019.

[23] Alejandro Hevia and Daniele Micciancio. The provable security of graph-based one-time signatures and extensions to algebraic signature schemes. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 379–396. Springer, 2002.

[24] Chris Dods, Nigel P Smart, and Martijn Stam. Hash based digital signature schemes. In *IMA International Conference on Cryptography and Coding*, pages 96–115. Springer, 2005.

[25] Johannes Buchmann, Erik Dahmen, Sarah Ereth, Andreas Hülsing, and Markus Rückert. On the security of the winternitz one-time signature scheme. In *International conference on cryptology in Africa*, pages 363–378. Springer, 2011.

[26] Andreas Hülsing. W-ots+–shorter signatures for hash-based signature schemes. In *International Conference on Cryptology in Africa*, pages 173–188. Springer, 2013.

[27] Sanjam Garg. *Candidate Multilinear Maps*. PhD thesis, University of California Los Angeles, 2013.

[28] Adeline Langlois, Damien Stehlé, and Ron Steinfeld. Gghlite: More efficient multilinear maps from ideal lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 239–256. Springer, 2014.

[29] Martin R Albrecht, Catalin Cocis, Fabien Laguillaumie, and Adeline Langlois. Implementing candidate graded encoding schemes from ideal lattices. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 752–775. Springer, 2015.

Hossein Oraei received the B.Sc. and M.Sc. degrees with honors in Mathematics from Qom University and Sharif University of Technology in 2012 and 2014, respectively. He also received his Ph.D. in cryptography from School of Mathematical sciences at Iran University of Science and Technology in 2020. His research interests include Symmetric Cryptography and Secret Sharing schemes.

Massoud Hadian Dehkordi received his Ph.D. degree in Mathematics from Loughborough University, UK, in 1998. He is currently a professor of mathematics at the school of Mathematical Sciences in Iran University of Science and Technology (IUST), Tehran, Iran. His research interests include Number Theory, Cryptography and other related topics.