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

Hossein Oraei, Massoud Hadian Dehkordi

Cryptography and Data Security Laboratory, School of Mathematics, Iran University of Science & Technology, Narmak, Tehran, 1684613114*, Iran*

Abstract

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. This work introduces $\mathrm{WOTS}^{\mathrm{GES}}$, a new WOTS type signature scheme in which the need for computing all of the intermediate values of the chains is eliminated. This significantly reduces the number of required operations needed to calculate the algorithms of $\mathrm{WOTS}^{\mathrm{GES}}$. To achieve this results, we have used the concept of "leveled" multilinear maps which is also referred to as graded encoding schemes. In the context of provable security, we reduce the hardness of graded discrete-logarithm (GDL) problem to the EU-CMA security of $\mathrm{WOTS}^{\mathrm{GES}}$ in the standard model.

Keywords: graded encoding schemes, multi-linear maps, GDL problem, digital signatures, one-time signature schemes, EU-CMA security.
Mathematics Subject Classification 2010: 94A60

1. Introduction

Multilinear maps are useful tools that provide many applications in cryptography such as one-round multi-party key exchange protocol and broadcast encryption scheme [6]. The notion of multilinear maps was introduced by Boneh and Silverberg [6] as an extension of bilinear maps. Different from bilinear maps, which can be built from pairing of elliptic curves, constructing multilinear maps was a long-standing open problem. This problem was eventually solved by Garg et al. [14], who constructed the first approximate construction of multilinear maps. They introduced the notion of graded encoding scheme as a variant of multilinear maps, and proposed a candidate construction by using ideal lattices. This proposed instantiation of graded encoding schemes is called GGH13.

Email addresses: `hossein_oraei@mathdep.iust.ac.ir` (Hossein Oraei), `mhadian@iust.ac.ir` (Massoud Hadian Dehkordi)

Graded encoding schemes are one of the most important cryptographic tools, enabling many important applications, such as functional encryption [27], obfuscation [15, 3] and also cryptanalysis of obfuscation [10], witness encryption [16], multipartite key exchange [14], aggregate signature scheme [19] and so on. In this paper, we offer another application of graded encoding schemes in signature schemes.

1.1. Background

Digital signature schemes [5, 9, 11, 1, 21] 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 [25]. These schemes are important cryptographic primitives that used as the core of the design of many-time digital signature schemes. One-time signature schemes have other important applications like digital signatures with forward security property [28, 8], network routing protocols [17] 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 [29]. One-time signature schemes designed using this technique play important roles in the design of both stateless and stateful many-time signature schemes [8, 22, 23, 5, 24, 4, 21]. 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 [8]. In addition, in WOTS type signature schemes, it is possible to make a trade-off between the runtime and the size of signature.

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, all of which begin at some 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.

1.2. Related work

Along the years, several different versions of WOTS scheme have been presented for various purposes [29, 18, 12, 7, 20, 24]. The main idea of the WOTS scheme was first presented in [29]. Using this basic idea, the one-time digital signature schemes [18, 12] 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 [7].

Under the name WOTS$^+$, Hülsing [20] 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 [5] and SPHINCS$^+$ [21] 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 [24]. 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.

1.3. Motivation

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 [29, 18, 12, 7, 20, 24], 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.

1.4. Contribution

In this work, we introduce WOTS$^{\mathrm{GES}}$, 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 of the 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.

Another important part of this work is the tight security proof that we provide for WOTS$^{\mathrm{GES}}$ by giving the exact relation between the graded discrete-logarithm (GDL) problem and the security of WOTS$^{\mathrm{GES}}$. More formally, we prove that WOTS$^{\mathrm{GES}}$ has EU-CMA security, if the GDL problem is hard.

1.5. Organization

This paper is organized as follows. Section 2, describes the required tools. In section 3, we give description of the generic WOTS scheme. In section 4, we propose W-OTS$^{\mathrm{GES}}$ based on graded encoding schemes and its security is discussed in section 5. In section 6, the instantiation of W-OTS$^{\mathrm{GES}}$ using GGH13 is discussed and finally in section 7, we conclude the paper.

2. Preliminaries

Here, we review some basic concepts about multilinear maps, graded encoding schemes and also digital signature schemes. The definitions and concepts presented in this section are used throughout this paper.

2.1. Multilinear maps

The notion of multilinear maps is defined as follows [6]. For cyclic groups $G_1, \ldots, G_k$ and G_T of the same prime order q, a k-multilinear map $e : G_1 \times \cdots \times G_k \longrightarrow G_T$ is a map such that:

1. **Multilinear:** For all $g_1 \in G_1, \ldots, g_k \in G_k$ and $a_1, \ldots, a_k \in \mathbb{Z}_q^*$ we have $e(g_1^{a_1}, \ldots, g_k^{a_k}) = e(g_1, \ldots, g_k)^{a_1 \cdots a_k}$.
2. **Non-degenerate:** If for $1 \leq i \leq k$, $g_i \in G_i$ be a generator of the group G_i, then $e(g_1, \ldots, g_k)$ is a generator of G_T.
3. **Computable:** For all $g_1 \in G_1, \ldots, g_k \in G_k$, the value $e(g_1, \ldots, g_k)$ is computed efficiently.

2.2. Graded encoding schemes

Garg et al. [14] defined the notion of k-graded encoding schemes as an approximation of k-multilinear maps as follows:

Definition 1 (k-graded encoding scheme). *Let R be a ring and $\mathcal{S} = \{S_i^{(\alpha)} \subset \{0,1\}^* \mid 0 \leq i \leq k, \alpha \in R\}$ be a family of sets such that for each constant index i, the sets $\{S_i^{(\alpha)} \mid \alpha \in R\}$ are disjoint. Then a k-graded encoding scheme* $\mathrm{GES}(R, \mathcal{S})$ *with the ring R and the family of sets $\mathcal{S}$ consists of the following procedures:*

- $\mathsf{InstGen}(1^\lambda, k)$: The randomized instance-generation procedure takes as input a security parameter λ and also multilinearity parameter k. It outputs $(\mathsf{params}, P_{zt})$, where P_{zt} is a zero-test parameter (as below) and params is description of the k-graded encoding scheme.

- $\mathsf{Samp}(\mathsf{params})$: The randomized ring sampler procedure outputs $a \in S_0^{(\alpha)}$ which is a "level-zero encoding", where $\alpha \in R$ is a random and nearly uniform element.

- $\mathsf{Enc}(\mathsf{params}, i, a)$: The (possibly randomized) encoding procedure takes as input an index $i \le k$ and a "level-zero" encoding $a \in S_0^{(\alpha)}$ and outputs $u \in S_i^{(\alpha)}$ which is a "level-i" encoding for the same $\alpha \in R$.

- $\mathsf{Add}(\mathsf{params}, i, u_1, u_2), \mathsf{Neg}(\mathsf{params}, i, u_1)$: On input of params, an index $i \le k$ and two level-i encodings $u_1 \in S_i^{(\alpha_1)}$ and $u_2 \in S_i^{(\alpha_2)}$, the addition and negation procedures compute $\mathsf{Add}(\mathsf{params}, i, u_1, u_2) = u_1 + u_2 \in S_i^{(\alpha_1+\alpha_2)}$ and $\mathsf{Neg}(\mathsf{params}, i, u_1) = -u_1 \in S_i^{(-\alpha_1)}$, respectively. Here, $-\alpha_1$ and $\alpha_1 + \alpha_2$ are negation and addition in the ring R.

  This implies that for a collection of h encodings $u_j \in S_i^{(\alpha_j)}$ where $j = 1, \dots, h$, we get $u_1 + \cdots + u_h \in S_i^{(\alpha_1+\cdots+\alpha_h)}$.

- $\mathsf{Mul}(\mathsf{params}, i_1, u_1, i_2, u_2)$: On input of params, two indices i_1, i_2 with $i_1 + i_2 \le k$, a level-i_1 encoding $u_1 \in S_{i_1}^{(\alpha_1)}$ and a level-i_2 encoding $u_2 \in S_{i_2}^{(\alpha_2)}$, the multiplication procedure computes $\mathsf{Mul}(\mathsf{params}, i_1, u_1, i_2, u_2) = u_1 \times u_2 \in S_{i_1+i_2}^{(\alpha_1 \cdot \alpha_2)}$, where $i_1 + i_2$ is integer addition and $\alpha_1 \cdot \alpha_2$ is multiplication in the ring R.

  This implies that for a collection of h encodings $u_j \in S_{i_j}^{(\alpha_j)}$ where $j = 1, \dots, h$, we get $u_1 \times \cdots \times u_h \in S_{i_1+\cdots+i_h}^{(\prod_{j=1}^{h} \alpha_j)}$ as long as $\sum_{j=1}^{h} i_j \le k$.

- $\mathsf{isZero}(\mathsf{params}, P_{zt}, u)$: On input of params, the zero-test parameter P_{zt} and u, the zero-test procedure outputs 1 if $u \in S_k^{(0)}$ and 0 otherwise. In other words, this procedure only outputs 1 if u be the level-k encoding of 0.

- $\mathsf{Ext}(\mathsf{params}, P_{zt}, u)$: On input of params, the zero-test parameter P_{zt} and $u \in S_k^{(\alpha)}$, the extraction procedure outputs $s \in \{0,1\}^\lambda$ such that:

  a) For every $\alpha \in R$ and two level-k encodings $u_1, u_2 \in S_k^{(\alpha)}$,

  $$\mathsf{Ext}(\mathsf{params}, P_{zt}, u_1) = \mathsf{Ext}(\mathsf{params}, P_{zt}, u_2). \tag{1}$$

  b) The distribution $\{\mathsf{Ext}(\mathsf{params}, P_{zt}, u) \mid u \in S_k^{(\alpha)}, \alpha \in R\}$ over $\{0,1\}^\lambda$ is nearly uniform, where λ is the security parameter.

We now give a more precise description of the above procedures: Garg et al. proposed GGH13 which is a k-graded encoding scheme over ideal lattices and is parameterized by the security parameter λ and the multilinearity parameter $k \le \text{poly}(\lambda)$. Their realization of the above procedures has two changes in the zero-test and extraction procedures as follows:

- **Zero-test:** This procedure sometime allows false positives, but not false negatives. More precisely, for every $u \in S_k^{(0)}$ it is hold that $\mathsf{isZero}(\mathsf{params}, u) = 1$, but for $\alpha \in R$ which is a nearly uniform random element,

  $$\Pr[\exists\ u \in S_k^{(\alpha)} \mid \mathsf{isZero}(\mathsf{params}, u) = 1] = \text{negl}(\lambda).$$

- **Extraction:** According to the ring sampler and encoding procedures, in order to get a level-i encoding of a nearly uniform random element $\alpha \in R$ where $1 \leq i \leq k$, we first run the ring sampler procedure to get a level-0 encoding of α, and then run the encoding procedure to get a level-i encoding of α. On the other hand, in the extraction procedure of GGH13, there is a good probability of generating the same output for any two different level-k encodings of α.

  Thus, the properties a) and b) of the extraction procedure are replaced by two weaker requirements:

  a′) For every $a \leftarrow \mathsf{Samp}(\mathsf{params})$ where $a \in S_0^{(\alpha)}$, if the (randomized) encoding procedure is run twice on a to obtain two level-k encodings $u_1, u_2 \in S_k^{(\alpha)}$, then:

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

  b′) The following distribution over $\{0,1\}^\lambda$ is nearly uniform:

  $$\{\mathsf{Ext}(\mathsf{params}, P_{zt}, u) \mid a \leftarrow \mathsf{Samp}(\mathsf{params}), u \leftarrow \mathsf{Enc}(\mathsf{params}, i, a)\}.$$

Remark 1. As explained in the extraction procedure, for every $\alpha \in R$ and two level-k encodings $u_1, u_2 \in S_k^{(\alpha)}$, it is hold that $\mathsf{Ext}(\mathsf{params}, P_{zt}, u_1) = \mathsf{Ext}(\mathsf{params}, P_{zt}, u_2) \in \{0,1\}^\lambda$ (with high probability in the real-life version). In this paper, for simplicity we work with the dream version of extraction procedure in which this probability is one. If we want to use the real-life version, 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 write $\mathsf{Ext}(\mathsf{params}, P_{zt}, S_k^{(\alpha)})$ to denote this λ bit string.

Remark 2. A k-graded encoding scheme may be consist of some secret parameters (for example, see the description of GGH13 in [13]). Nevertheless, anyone can use a k-graded encoding scheme without knowing their secret parameters.

Remark 3. We can assume that a level 1 encoding of $1 \in R$ is published as part of the instance-generation procedure, namely an element $y \in S_1^{(1)}$ [13].

2.2.1. Graded Discrete-Logarithm (GDL) problem

The analog of discrete logarithm problem in a k-graded encoding scheme $\mathrm{GES}(R, \mathcal{S})$ can be considered as follows: Given a level-i encoding $u_i \in S_i^{(\alpha)}$, where $1 \leq i \leq k$ and $\alpha \in R$ is a random and nearly uniform element, the adversary must 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 $\mathsf{Exp}^{\mathsf{GDL}}_{\mathrm{GES}}(\mathcal{B}, \lambda)$:

1. Based on the multilinearity parameter k and the security parameter λ, 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, the challenger $\mathcal{C}$ firstly runs $a \leftarrow \mathsf{Samp}(\mathsf{params})$ to get 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 $\mathcal{B}$.
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.

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

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,

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

Note that according to the Remark 3, 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

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

2.3. 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 write $x \overset{\$}{\leftarrow} \mathcal{X}$, if x is chosen randomly from the set $\mathcal{X}$. We also write log for $\log_2$.

Definition 2. *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 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:*

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

.

2.3.1. 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 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 $\mathrm{Dss}(1^{\mathrm{n}})$ for a Dss $= (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$ with the security parameter n.

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

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

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

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

Definition 3. *Let* $n, t, q \in \mathbb{N}$ *and* $t, q = \mathrm{Poly}(n)$*. We say that a signature scheme* 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* $\mathrm{InSec}^{\mathsf{EU\text{-}CMA}}(\mathrm{Dss}(1^{\mathrm{n}}); t, q)$ *is a negligible function of* n*:*

$$\mathrm{InSec}^{\mathsf{EU\text{-}CMA}}(\mathrm{Dss}(1^{\mathrm{n}}); t, q) := \max_{\mathcal{A}}\{\mathrm{Succ}^{\mathsf{EU\text{-}CMA}}_{\mathrm{Dss}(1^{\mathrm{n}})}(\mathcal{A})\} = \mathrm{negl}(n). \tag{3}$$

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 4. *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 $\mathcal{E}_{\mathsf{ck}}^{i,j}(X)$ takes as input a public chain key ck, an interval $i, j \in \mathbb{N}$, $0 \leq i < j \leq \lambda$, 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 $0 \leq i \leq j \leq m \leq \lambda$ and every $X \in \mathcal{D}$, it must hold that

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

.

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

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

Key Generation Algorithm ($\mathsf{Kg}(1^n)$): On input of the 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

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

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 $C \leq l_1(w-1)$). Now, the signature is computed as

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

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 $b_i, 1 \leq i \leq l$ are computed as above. Next, if the following comparison holds, the verification algorithm returns **true** and **false** otherwise:

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

4. W-OTS$^{\mathbf{GES}}$

Here, we propose our digital signature scheme W-OTS$^{\text{GES}}(k, m)$ based on a k-graded encoding scheme GES$(R, \mathcal{S})$. As mentioned before, this signature scheme is a new variant of WOTS scheme. Like other versions of WOTS, W-OTS$^{\text{GES}}(k, m)$ 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 3, 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 $2 \leq i \leq k$.

Key Generation Algorithm ($\mathsf{Kg}(\mathrm{GES})$): 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 $1 \leq j \leq l$. The private signing key $\mathsf{sk} = (\mathsf{a}_1, \ldots, \mathsf{a}_l)$ consists of this level-zero encodings.

Next for $1 \leq j \leq l$, 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 $\mathsf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_l)$.

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 $1 \leq j \leq l$, 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

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

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 $1 \leq j \leq l$.

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. In this algorithm for $1 \leq j \leq l$:

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 W-OTS$^{\mathrm{GES}}$

Here, we prove the security of W-OTS$^{\mathrm{GES}}$. We explain how an adversary for W-OTS$^{\mathrm{GES}}$ can be used to define an adversary that is a solver for the GDL problem. More precisely, we reduce the hardness of GDL problem to the EU-CMA security of W-OTS$^{\mathrm{GES}}$.

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* W-OTS$^{\mathrm{GES}}(k, m)$*, then there exists a PPT adversary $\mathcal{B}$ that is a solver for the GDL problem such that*

$$\mathrm{Succ}^{\mathsf{EU\text{-}CMA}}_{\mathrm{Dss}(1^{\mathrm{n}})}(\mathcal{A}) \leq kl \cdot \mathrm{Succ}^{\mathsf{GDL}}_{\mathrm{GES}}(\mathcal{B}, \lambda). \tag{4}$$

Proof. Consider a PPT adversary $\mathcal{A}$ which acts according to the Experiment $\mathsf{Exp}^{\mathsf{EU\text{-}CMA}}_{\mathrm{Dss}(1^{\mathrm{n}})}(\mathcal{A})$ against the security of W-OTS$^{\mathrm{GES}}(k, m)$, such that his success probability $\mathrm{Succ}^{\mathsf{EU\text{-}CMA}}_{\mathrm{Dss}(1^{\mathrm{n}})}(\mathcal{A}) = \varepsilon_{\mathcal{A}}$ is non-negligible. In the rest of the proof, we will construct 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:

1. Based on the multilinearity parameter k and the security parameter λ, 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 $\mathrm{GES}(R, \mathcal{S})$. 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 $u \leftarrow \mathsf{Enc}(\mathsf{params}, i, a)$ 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 $\mathsf{Exp}^{\mathsf{EU\text{-}CMA}}_{\mathrm{Dss}(1^{\mathrm{n}})}(\mathcal{A})$. So, $\mathcal{B}$ executes the W-OTS$^{\mathrm{GES}}$ key generation algorithm $\mathsf{Kg}(1^n)$ 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 $\mathsf{Exp}^{\mathsf{EU\text{-}CMA}}_{\mathrm{Dss}(1^{\mathrm{n}})}(\mathcal{A})$ 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 $1 \leq \gamma \leq l$. 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 $1 \leq \gamma \leq l$. 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 $0 \leq j < i$:

(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 $1 \leq \gamma' \leq l$ 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 $\mathsf{a}'_{\gamma'}$. Next, $\mathcal{B}$ runs the multiplication procedure $\mathsf{Mul}(\mathsf{params}, i, \mathsf{u}_i, k-i, 1_{k-i})$ 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 $i \leq j' \leq k$ as he can run the multiplication procedure $\mathsf{Mul}(\mathsf{params}, i, \mathsf{u}_i, j'-i, 1_{j'-i})$ to compute the level-j' encoding $\mathsf{u}_{j'} \in S_{j'}^{(\alpha)}$. So, $\mathcal{B}$ can only answer the $\mathcal{A}$'s query M, if $i \leq b_{\gamma'}$.

(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 $u_{b^\star_{\gamma'}} \in S^{(\alpha)}_{b^\star_{\gamma'}}$ 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 line 2c, $\mathcal{B}$ can only answer the $\mathcal{A}$'s query M, if $i \leq b_{\gamma'}$. 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 $b^\star_\gamma < b_\gamma$, where $1 \leq \gamma \leq l$. 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 line 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 W-OTS$^{\mathrm{GES}}(k, m)$ 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* W-OTS$^{\mathrm{GES}}$ *against an* EU-CMA *attack by*

$$\mathrm{InSec}^{\mathsf{EU\text{-}CMA}}(\text{W-OTS}^{\mathrm{GES}}(k,m);t,1) \leq kl \cdot \mathrm{InSec}^{\mathsf{GDL}}(\mathrm{GES};t',\lambda). \tag{5}$$

with $t' = t + 5l$.

Proof. Firstly note that the equation 5 can be simply derived from equation 4 and also from definitions 2 and 3. The time $t' = t + 5l$ is also the maximum runtime required by the adversary $\mathcal{A}$ (which behaves according to the definition 3) plus the time required to execute the three algorithms of W-OTS$^{\mathrm{GES}}$ once (follow the proof of lemma 1).

6. Instantiation using GGH13

To use W-OTS$^{\mathrm{GES}}$, 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 $k \leq \mathrm{poly}(\lambda)$. 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 1. 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) [13]. On the other hand, the private signing key $\mathsf{sk} = (\mathsf{a}_1, \ldots, \mathsf{a}_l)$ of the signature scheme W-OTS$^{\mathrm{GES}}$ 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 W-OTS$^{\mathrm{GES}}$, 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 1, 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 [26], 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 W-OTS$^{\mathrm{GES}}$ using GGHLite can improve the efficiency of W-OTS$^{\mathrm{GES}}$.

7. Conclusion

Here, we provide a comparison for the number of operations required by the key generation, signature and verification algorithms of W-OTS$^{\mathrm{GES}}$ scheme and other WOTS scheme variants in the literature [29, 18, 12, 7, 20, 24]. We have summarized the results in Table 1.

Table 1: Comparison of the computational complexities

Step	WOTS schemes [29, 18, 12, 7, 20, 24]	proposed scheme
Public verification key generation	$lk \cdot \mathrm{T}_{\mathrm{fc}}$	$l \cdot (\mathrm{T}_{\mathrm{enc}} + \mathrm{T}_{\mathrm{ext}})$
Signature algorithm	$(\sum_{i=1}^{l} b_i) \cdot \mathrm{T}_{\mathrm{fc}}$	$l \cdot \mathrm{T}_{\mathrm{enc}}$
Verification algorithm	$(\sum_{i=1}^{l} (k - b_i)) \cdot \mathrm{T}_{\mathrm{fc}}$	$l \cdot (\mathrm{T}_{\mathrm{enc}} + \mathrm{T}_{\mathrm{ext}})$

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:

- T_{fc}: The time required to execute one iteration of the used function chain.
- $\mathrm{T}_{\mathrm{enc}}$: The time required to execute the encoding procedure of the used k-graded encoding scheme.
- $\mathrm{T}_{\mathrm{ext}}$: The time required to execute the extraction 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 W-OTS$^{\mathrm{GES}}$ is less than that of other WOTS scheme variants in the literature [29, 18, 12, 7, 20, 24].

In [2], the first practical implementation of graded encoding schemes is presented in which the efficiency of GGHLite has also been improved. Using the results of this paper, along with the practical implementations of graded encoding schemes, we can obtain an efficient one-time digital signature scheme for various applications [17, 28, 8].

References

[1] Alamlou Q., Blazy O., Cauchie S. Gaborit P.: *A code-based group signature scheme*, Designs, Codes and Cryptography, 82(1-2), pp. 469-493. (2017).

[2] Albrecht M.R., Cocis C., Laguillaumie F., Langlois A.: *Implementing candidate graded encoding schemes from ideal lattices*, In International Conference on the Theory and Application of Cryptology and Information Security, Springer, Berlin, Heidelberg, pp. 752-775. (2014).

[3] Ananth P., Jain A., Sahai A.: *Robust transforming combiners from indistinguishability obfuscation to functional encryption*, In Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Cham, pp. 91-121, (2017).

[4] Aumasson J.P., Endignoux G.: *Improving stateless hash-based signatures*, In Cryptographers Track at the RSA Conference, Springer, Cham, pp. 219-242. (2018).

[5] Bernstein D.J., Hopwood D., Hulsing A., Lange T., Niederhagen R., Papachristodoulou L., Schneider M., Schwabe P., Wilcox-OHearn Z.: *SPHINCS: practical stateless hash-based signatures*, In Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Berlin, Heidelberg, pp. 368-397. (2015).

[6] Boneh D., Silverberg A.: *Applications of multilinear forms to cryptography*, Contemporary Mathematics, 324(1), pp. 71-90, (2003).

[7] Buchmann J., Dahmen E., Ereth S., Hulsing A., Rckert M.: *On the security of the Winternitz one-time signature scheme*, In International Conference on Cryptology in Africa, Springer, Berlin, Heidelberg, pp. 363-378. (2011).

[8] Buchmann J., Dahmen E., Hulsing A.: *XMSS-a practical forward secure signature scheme based on minimal security assumptions*, In International Workshop on Post-Quantum Cryptography, Springer, Berlin, Heidelberg, pp. 117-129. (2011).

[9] Cheng S., Nguyen K., Wang H.,: *Policy-based signature scheme from lattices*, Designs, Codes and Cryptography, 81(1), pp.43-74. (2016).

[10] Cheon J.H., Hhan M., Kim J., Lee C.: *Cryptanalyses of branching program obfuscations over GGH13 multilinear map from the NTRU problem*, In Annual International Cryptology Conference, Springer, Cham, pp. 184-210. (2018).

[11] Dagdelen O., Galindo D., Veron P., Alaoui S.M.E.Y., Cayrel P.L.,: *Extended security arguments for signature schemes*, Designs, Codes and Cryptography, 78(2), pp.441-461. (2016).

[12] Dods C., Smart N.P., Stam M.: *Hash based digital signature schemes*, In IMA International Conference on Cryptography and Coding, Springer, Berlin, Heidelberg, pp. 96-115. (2005).

[13] Garg S.: *Candidate Multilinear Maps*, PhD diss., University of California Los Angeles, (2013).

[14] Garg S., Gentry C., Halevi S.: *Candidate multilinear maps from ideal lattices*, In Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Berlin, Heidelberg, pp. 1-17, (2013).

[15] Garg S., Gentry C., Halevi S., Raykova M., Sahai A., Waters B. : *Candidate indistinguishability obfuscation and functional encryption for all circuits*, SIAM Journal on Computing, 45(3), pp. 882-929, (2016).

[16] Garg S., Gentry C., Halevi S., Wichs D.: *On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input*, Algorithmica, 79(4), pp. 1353-1373, (2017).

[17] Hauser R., Przygienda A., Tsudik G.: *Reducing the cost of security in link-state routing*, In Network and Distributed System Security, 1997. Proceedings., 1997 Symposium on, IEEE, pp. 93-99. (1997).

[18] Hevia A., Micciancio D.: *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, Springer, Berlin, Heidelberg, pp. 379-396. (2002).

[19] Hohenberger S., Sahai A., Waters B.: *Full domain hash from (leveled) multilinear maps and identity-based aggregate signatures*, In Advances in Cryptology-CRYPTO 2013, Springer, Berlin, Heidelberg, pp. 494-512. (2013).

[20] Hulsing A.: *W-OTS^{+} shorter signatures for hash-based signature schemes*, In International Conference on Cryptology in Africa, Springer, Berlin, Heidelberg, pp. 173-188. (2013).

[21] Hulsing A., Bernstein D.J., Dobraunig C., Eichlseder M., Fluhrer S., Gazdag S.L., Kampanakis P. et al.: *SPHINCS^{+}*, Submission to the NISTs post-quantum cryptography standardization process. (2018).

[22] Hulsing A., Busold C., Buchmann J.: *Forward secure signatures on smart cards*, In International Conference on Selected Areas in Cryptography, Springer, Berlin, Heidelberg, pp. 66-80. (2012).

[23] Hulsing A., Rausch L., Buchmann J.: *Optimal parameters for XMSS^{MT}*, In International Conference on Availability, Reliability, and Security, Springer, Berlin, Heidelberg, pp. 194-208. (2013).

[24] Hulsing A., Rijneveld J., Song F.: *Mitigating multi-target attacks in hash-based signatures*, In Public-Key CryptographyPKC 2016, Springer, Berlin, Heidelberg, pp. 387-416. (2016).

[25] Lamport L.: *Constructing digital signatures from a one-way function*, Palo Alto: Technical Report CSL-98, SRI International,Vol. 238. (1979).

[26] Langlois A., Stehl D., Steinfeld R.: *GGHLite: More efficient multilinear maps from ideal lattices*, In Annual International Conference on the Theory and Applications of Cryptographic Techniques,Springer, Berlin, Heidelberg, pp. 239-256. (2014).

[27] Lin H., Tessaro S.: *Indistinguishability obfuscation from trilinear maps and block-wise local PRGs*, In Annual International Cryptology Conference, Springer, Cham, pp. 630-660, (2017).

[28] Malkin T., Micciancio D., Miner S.: *Efficient generic forward-secure signatures with an unbounded number of time periods*, In International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Berlin, Heidelberg, pp. 400-417. (2002).

[29] Merkle R.C.: *A certified digital signature*, In Conference on the Theory and Application of Cryptology,Springer, New York, NY, pp. 218-238. (1989).