Efficient Multilinear Map from Graded Encoding Scheme

Majid Salimi

Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran
M.Salimi@eng.ui.ac.ir,MajidSalimi3@gmail.com

Abstract. Though the multilinear maps have many cryptographic applications, secure and efficient construction of such maps is an open problem. Many multilinear maps like GGH, GGH15, CLT, and CLT15 have been and are being proposed, while none of them is both secure and efficient. The construction of some multilinear maps is based on the Graded Encoding Scheme (GES), where, the necessity of announcing zero-testing parameter and encoding of zero has destroyed the security of the multilinear map. Attempt is made to propose a new GES, where, instead of encoding an element, the users can obtain the encoding of an associated but unknown random element. In this new setting, there is no need to publish the encodings of zero and one. This new GES provides the actual functionality of the usual GES and can be applied in constructing a secure and efficient multilinear map and a multi-party non-interactive key exchange (MP-NIKE) scheme. We also improve the MP-NIKE scheme of [1] and turn it into an ID-based MP-NIKE scheme.

Keywords: Multilinear map, Self-multilinear map, multi-party key exchange.

1 Introduction

Multilinear map is the extension of bilinear map, where, instead of two elements, it is given many elements as the input. This map is widely applied in many cryptographic applications, like broadcast encryption, attribute-based encryption, indistinguishable obfuscation, and multi-party key exchange.
Boneh and Silverberg (2003) introduced the notion of multilinear map and showed that a multilinear map can be applied in constructing efficient broadcast encryption and non-interactive key-exchange, but they did not introduce any concrete construction for multilinear maps therein. They found that the well-known bilinear pairings cannot be generalized in constructing multilinear maps. Gentry et al. (2013) found that a multilinear map can be applied in constructing an Indistinguishable Obfuscation $i\mathcal{O}$ [17]; moreover, it can be applied in constructing other tools like Pseudo-Random Function (PRF) and Constrained PRF [13], Attribute-Based Encryption (ABE) [18] and Functional Encryption (FE).

1.1 Related Work

In 2003, Garg et al. proposed the first candidate of the multilinear map (GGH13) based on the Graded Encoding Scheme (GES) in ideal lattices. Coron et al. (2013) proposed another candidate of the multilinear map (CLT13) based on GES and over integers [4]. Both the CLT13 and GGH13 maps are inefficient [6]. Langlois et al. (2014) proposed a more efficient version of GGH, named GGH-Lite [5], and then, Albrecht et al. (2014) implemented the GGHLite and made its codes publicly available under an open-source license [6]. Their construction is inefficient in the sense that, computing the value of a 7-linear map takes about 1.75 seconds on a 16-core CPU [6]. Hu and Jia (2015) revealed that the GGH13 and GGHLite multilinear maps are not secure [7]. Following this, Cheon et al. (2015) proposed an attack on the CLT13 multilinear map [8]. In the same year, Gentry, Gorbunov and Halevi proposed a graph-based multilinear map named GGH15 [9], and immediately, Coron et al. (2014) proposed an attack against GGH15 [10]. Coron et al. (2015) improved the CLT13 scheme [11], while, immediately, Minaud and Fouque proposed an attack on their scheme [12]. All these schemes lack formal security proof. Lewi et al. (2016) constructed a framework for CLT multilinear map named 5Gen [13], which was the most efficient multilinear map [13], [14] of the date. Ma and Zhandry (2018) proposed a CLT-like scheme and proved that their scheme is secure against known attacks, but they did not prove the security in a formal model [14].

In 2016, Albrecht et al. (2016) proposed a multilinear map from obfuscation [15], [16], which is not efficient because of employing costly tools like indistinguishable obfuscation and homomorphic encryption.

1.2 Our Contribution

A new GES scheme is presented in this article. In previous graded encoding schemes each element has many encodings, while in this scheme any element has a unique encoding. In the previous schemes, any code-word could be easily decoded without re-randomization, while, this encoding scheme, even without re-randomization is irreversible. Previous schemes need encodings of zero for re-randomization, while, in this scheme, re-randomization and encodings of zero as its public parameters are not required. Next to these, they are restricted to a pre-defined level κ to extract a representation of an element, while the proposed GES scheme can extract the representation of the element in any level of encoding. In this proposed GES, users can directly check the equality of two encoded values, thus, no need for zero-testing parameter. In this proposed GES, users can add and multiply the encoded elements. Multiplication of encoded elements will increase the level of encoding similar to that of the available GES schemes. The advantage of this newly proposed GES vs. its counterparts is that users can extract a representation of encoded elements at any arbitrary level of encoding, without being limited to a predefined level κ. Contrary to the available GES candidates, this proposed GES scheme is practical and efficient in different aspects like the system parameters' size, encoded elements' size and computation cost.

After introducing this GES, we will explain how it can be applied in multilinear map and designing MP-NIKE scheme, where the key generation algorithm is not run by a Trusted Third Party (TTP).

1.3 Paper Organization

The rest of this article is organized as follows: In Section 2 the notation, required definitions and the proposed GES general model are introduced. Our proposed GES scheme is described in Section 3. In Section 4, an efficient multilinear map based on the proposed GES is constructed. An MP-NIKE scheme without key generation algorithm run by TTP is proposed in Section 5. In Section 6, the MP-NIKE scheme of [1] is converted to an ID-Based scheme. The security of the proposed multilinear map is proved in Section 7, and finally, the paper is concluded in Section 2.

2 Preliminaries

The required mathematical background and the GES general model are introduced here.

2.1 Notations

The notations applied throughout this article are listed in Table 1.

2.2 Graded Encoding Scheme

The recent constructions of multilinear maps (e.g. the GGHLite multilinear map) use an encoding scheme named Graded Encoding Scheme (GES) [5] based on ideal lattices. In the available GES, a same element has many different encodings. In this section, the GGHLite graded encoding scheme [5] is briefly described. The GES encoding scheme involves the following seven algorithms.

Instance generation InstGen(1^λ, 1^κ): Consider polynomial rings $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ and $R_q = R/qR$. The algorithm InstGen is given a security parameter λ and the multilinearity parameter κ as the input, and generates system parameters SP and one instance of level-1 encoding of 1 ($[o]_1$) and two instances of level-1 encodings of zero ($[z_1]_1, [z_2]_1$). The $[z_1]_1$ and $[z_2]_1$ will be considered as randomizers.

Level-0 sampler samp(SP): In this algorithm, any user chooses an integer y which in GES is defined as a level-0 encoding. In lattice-based GES schemes, the level-0 encoding y is an integer sampled from discrete Gaussian distribution.

Level-1 encoding enc(SP, y): Given a level-0 encoding y and SP, this algorithm computes $[y]_1 = y[o]_1$ and yields the encoded value of y at level 1, while, by having the publicly known $[o]_1$ computing $[y]_1$ becomes reversible and the value y can be computed. In this case, the user must randomize $[y]_1$ as follows:

$$\acute{[y]}_1 = [y]_1 + r_1[z_1]_1 + r_2[z_2]_1, \tag{1}$$

h

Table 1. Notations

Symbol	Description
$z(q)$	A random polynomial of q, such that $z(0) = 0$
$r^{(j)}$	j-th bit of the integer r
p, q	Two large primes
N	Modulus of computation
$\mathbb{G}$	A multiplicative subgroup of $\mathbb{Z}_N^*$
g	A generator of $\mathbb{G}$
r	Random integer
W	A set of encoded base elements which is used for encoding other elements.
u_i	Encoding of r_i
b_i	Blinded encoding of r_i
$[x]_j$	The element x is encoded at level j
SP	System parameters
msk	The KGC's master secret key
e_i	Encoding of y_i
d_i	Blinded encoding of y_i
γ	Security parameter
A	The adversary
B	The challenger algorithm

where r_1 and r_2 are two random integers (level-0 encodings). Note that the resultant $[y]\acute{}_1$ is still an encoding of y at level 1. Adding different multiples of $[z_1]_1$ and $[z_2]_1$, as their name suggests, is to randomize the encoding, and does not affect the actual value and level of the the encoding. The $[y]\acute{}_1$ is a random encoding of integer y and computing the value of y from $[y]\acute{}_1$ is intractable. In the GES encoding scheme $[y]_1$, $[y]\acute{}_1$ and $[o]_1$ are level-1 encodings.

Adding encodings add($[a]_l, [b]_l$**)**: For adding the two elements $[a]_l$ and $[b]_l$ together they must be at the same level l and the result is $[a + b]_l$ at level l.

Multiplying encodings mult($[a]_{l_1}, [b]_{l_2}$**)**: Multiplication of two elements $[a]_{l_1}$ and $[b]_{l_2}$ with any arbitrary level l_1 and l_2 is possible and the result will be $[ab]_{l_1+l_2}$ at level $l_1 + l_2$.

isZero(SP**,** p_{zt}**,** $[y]_\kappa$**)**: The isZero algorithm is applied in testing whether an encoded element $[y]_\kappa$ (at a pre-defined level κ) is a valid encoding of zero or not, applicable in equality test of two encoded elements (i.e. if $[y_1]_\kappa - [y_2]_\kappa$ is valid encoding of zero then $y_1 = y_2$). The p_{zt} is the zero testing parameter and if $[y]_\kappa . p_{zt}$ is less than specific value ($q^{3/4}$), then $[y]_\kappa$ is a valid encoding of zero, (i.e. $y = 0$).

ext(SP, pzt, $[y]_\kappa$): For extracting the random representation of an encoded element, it must be at a predefined level κ, where users can extract it by multiplying it with the *zero testing parameter*. The result is a unique image of y not its actual value. This image of y only depends on its exact value and does not depend on the randomizers and their coefficients. For exact details, refer to [5].

2.3 Definitions

To describe and prove the security of the proposed scheme, the following definitions should be defined:

Definition 1. *(Safe modulus): Let $N = \acute{p}\acute{q}$ be the product of two large primes. N is called a safe RSA modulus if $\acute{p} = 2p + 1$ and $\acute{q} = 2q + 1$, where p and q are prime numbers of the same size.*

Definition 2. *(Subset Sum Problem (SSP)). Let $W = \{e_1, e_2, \dots, e_m\}$ be a set of m random integers. The SSP is the problem of finding a subset of W, where the sum of its elements is equal to a given value u.*

The SSP is an NP-complete problem and the complexity of its best possible solutions is equal to $O(2^{m/2})$ [20], $O(u\sqrt{m})$ [21] and $O(mc)$ [22], where c is the size of the largest element of W.

Definition 3. *(Multilinear Map). Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of the same order N. Consider $g_1, g_2, \dots$ and g_t are generators of $\mathbb{G}_1$, and g_{t+1} is a generator of $\mathbb{G}_2$. The map $e : \underbrace{\mathbb{G}_1 \times \mathbb{G}_1 \times \cdots \times \mathbb{G}_1}_{t\ times} \rightarrow \mathbb{G}_2$ is named a multilinear map if it satisfies the following properties [2].*

1. *For any integer $\beta \in \mathbb{Z}$ and generators $g_i \in \mathbb{G}_1$, $i = 1, \dots, t$, and $g_{t+1} \in \mathbb{G}_2$, we have $e(g_1^{\alpha}, \dots, g_j, \dots, g_t^{\gamma})^{\beta} = e(g_1^{\alpha}, \dots, g_i^{\beta}, \dots, g_t^{\gamma})$.*
2. *The map e is non-degenerate, i.e. if $g_i \in \mathbb{G}_1$, for $i = 1, \dots, t$, are all generators of $\mathbb{G}_1$, then $g_{t+1}{=}e(g_1, \dots, g_t)$ is a generator of $\mathbb{G}_2$.*

Definition 4. *(Injective function f). A function $f : \mathbb{S} \rightarrow \mathbb{G}$ is injective if it maps each element of $\mathbb{S}$ into exactly one distinct element of $\mathbb{G}$, where $\mathbb{S}$ is a set and $\mathbb{G}$ is a group. The size of $\mathbb{G}$ must be equal or greater than $\mathbb{S}$.*

Definition 5. *(Isomorph ring). The $(\mathbb{R}_1, +, \times) \cong (\mathbb{R}_2, +, \times)$ are isomorph rings if there exists a reversible bijective function $f : \mathbb{R}_1 \rightarrow \mathbb{R}_2$ in a sense that it maps each element of $\mathbb{R}_1$ to exactly one element of $\mathbb{R}_2$ and each element of $\mathbb{R}_2$ is paired to exactly one element of $\mathbb{R}_1$. For any elements a and $b \in \mathbb{R}_1$ the following is yield.*

$$f(a) + f(b) = f(a + b) \tag{2}$$

$$f(a) \times f(b) = f(a \times b) \tag{3}$$

Definition 6. *(Quotient ring). Let $\mathcal{I}$ be an ideal in $\mathbb{Z}$ we define an equivalence relation $\sim$ on $\mathbb{Z}$ as below*

$$a + \mathcal{I} = \{a + r : r \in \mathcal{I}\} \tag{4}$$
$$a \sim b \iff a - b \in \mathcal{I} \tag{5}$$

We use $\mathbb{Z}/\mathcal{I}$ to show the set of all such equivalence classes. The $\mathbb{Z}/\mathcal{I}$ is a ring and we have

$$(a + I) + (b + I) = (a + b) + I \tag{6}$$
$$(a + I)(b + I) = (ab) + I \tag{7}$$

The multiplicative identity of $\mathbb{R} = \mathbb{Z}/\mathcal{I}$ is $\bar{1} = (1 + I)$ and its zero-element is $\bar{0} = (0 + I) = I$.

3 The Proposed Graded Encoding Scheme

Before describing our Graded Encoding Scheme, first, the Yamakawa self-bilinear map should be discussed. The bilinear map is a multilinear map that has two domain inputs ($t = 2$ in Definition 4) and a different target group, where its target group and domain groups are different. Yamakawa et al. proposed a new bilinear map, named the self-bilinear map, where the two domain groups and the target group are the same with unknown order [23]. If the order of the self-bilinear map group is known (e.g. a safe prime $2p + 1$, where p is also prime), then the CDH problem will be easy in this group, because, if $e(g, g) = g_1 = g^c \bmod 2p + 1$ then $e(g^a, g^b) = g_1^{ab} \bmod 2p + 1 = g^{abc} \bmod \ 2p + 1$. Therefore computing $g^{ab} \bmod 2p + 1$ is possible by computing $e(g^{c^{-2}}, e(g^a, g^b)) = g^{bc} \bmod 2p + 1$, while $g^{c^{-2}} = g^{c^{p-3}} \bmod 2p + 1$ [23], [24]. The $g^{c^{-2}} \bmod N$ can be computed by at most $O(logp)$ computations since $e(g^{c^i}, g^{c^i}) = g^{c^{2i+1}}$ and $e(g, g^{c^i}) = g^{c^{i+1}}$ [24]. Yamakawa revealed that in self-bilinear schemes the decisional assumptions like the decisional Diffie-Hellman (DDH) assumption cannot hold and they consider only computational assumptions like CDH problem [23]. A self-multilinear map is defined as follows:

Definition 7. *Self-Multilinear Map: Let $\mathbb{R}$ be a ring of the unknown order q and $\mathbb{G}$ be an isomorph cyclic group of the unknown order q. Then the map $e : \underbrace{\mathbb{G} \times \mathbb{R} \times \cdots \times \mathbb{R}}_{t \ \ times} \rightarrow \mathbb{G}$ is called a self-multilinear map if it satisfies the following properties:*

1. *For any integer $r \in \mathbb{S}$, where $\mathbb{S}$ is a set of integers, generator $g \in \mathbb{R}$ and generator $\acute{g} \in \mathbb{G}$ we have*

$$e(\acute{g}, g, \ldots, g, \ldots, g)^r = e(\acute{g}, g, \ldots, g^r, \ldots, g) = e(\acute{g}^r, g, \ldots, g, \ldots, g). \tag{8}$$

2. *The map e is non-degenerate; that is, if $g_i \in \mathbb{R}$, for $i = 1, \ldots, t-1$ are generators of $\mathbb{R}$ and $\acute{g}$ is a generator of $\mathbb{G}$ then $g_t = e(\acute{g}, g_1, \ldots, g_{t-1})$ is also a generator of $\mathbb{G}$.*

Let $N = \acute{p}\acute{q} = (2p+1)(2q+1)$ be a safe modulus, where $q > p$, $\mathcal{I}$ be a principal ideal over $\mathbb{Z}$ generated by q and $z(q)$ be a random polynomial of q, then $\mathbb{Z}/q\mathbb{Z} = \{0, 1, \ldots, q-1\}$ is considered as a quotient ring. In this article the $(\mathbb{R} = \mathbb{Z}/\mathcal{I}, +, \times)$ is defined as a ring of order q with generator $p + z(q)$. We have $y_1p^2 + z(q) \mod q = (y_1p^2 \bmod q) + z(q)$, so $y_1p^2 + z(q)$ is an element of $\mathbb{R}$. Let g be a generator of subgroup $\mathbb{G}_{\mathbb{1}}$ of $\mathbb{Z}_N^*$ of order pq. The $(\mathbb{G}, +)$ is defined as a group of order q with generator $g^p \bmod N$, where the ring $\mathbb{R}$ over $+$ and the group $(\mathbb{G}, +)$ are isomorph, which makes them behave a similar manner.

In the proposed GES, $[y]_1 = e = yp + kq \in \mathbb{R}$ is a level-1 public encoding of y and $[y]_1^b = d = g^{py} \bmod N \in \mathbb{G}$ is the blinded version of this encoding of y, where $y \in \{0, 1, \ldots, q-1\}$ is the plaintext and k is a random integer. The exponent of p is encoding level (e.g. $yp^2 + z(q)$ is a level-2 encoding of y, where $z(q)$ is the polynomial of q, thus making $z(0) = 0$). The $z(q)$ is considered as the noise of encoding and does not affect the result of extraction algorithm. Let $[y_{i_1}]_1^b = d_{i_1}$ be a blinded level-1 encoding of y_{i_1}, and e_{i_2}, e_{i_3} be the level-1 encodings of y_{i_2} and y_{i_3}, respectively. The blinded level-1 encoding of y_i is referred to as the representation of y_i at level 1. To extract a representation of element $F = y_{i_1}y_{i_2}y_{i_3}$ at level 3, users can simply compute $(d_{i_1})^{e_{i_2}e_{i_3}} = g^{p^3 y_{i_1}y_{i_2}y_{i_3}} \bmod N$. The final result of the extraction algorithm does not depend on the actual polynomial $z(q)$, which may be added during arithmetic operations on encoded elements; that is $z(q)$ is just the encoding noise indicating that there exist too many encodings for an integer y_i. Because 1 randomizers are not applied, there is no way for generating two different encodings for the same integer. This property prevents many attacks. If one can find three different encodings u_{i_1}, u_{i_2} and u_{i_3} of the same integer y_i, he can break the security of the proposed GES scheme by first, applying the Euclidean algorithm on $u_{i_1} - u_{i_2}$ and $u_{i_1} - u_{i_3}$, which is assumed to output a multiple of q, and second, computing the gcd of this output and $u_{i_2} - u_{i_3}$, which would be q. Next to this, because in the proposed GES, users can obtain only a unique codeword of an element in a specific level, no zero testing parameter p_{zt} is needed for equality testing.

In this GES, instead of publishing encodings of zero and one as system parameters, the level-1 and blinded level-1 encodings of m random integers are published. The value of m will be determined later. Now assume that users want to obtain an encoding of an element of $\mathbb{S}$, where $\mathbb{S} = \{2, 3, \ldots, 2^m\}$ is a set of integers, then, instead of obtaining the encoding of an element $r \in \mathbb{S}$ they can compute an encoding of the image of r $(\acute{r})$ in the ring $\mathbb{R}$ and group $\mathbb{G}$. The users have access to description of $\mathbb{S}$, with no idea about the real value of the image of r in the $\mathbb{R}$ and $\mathbb{G}$. Let $W = \{(e_1, d_1), \ldots, (e_m, d_m)\}\}$, where $e_i = [y_i]_1 = y_ip + k_iq \in \mathbb{R}$ and $d_i = [y_i]_1^b = g^{py_i} \bmod N \in \mathbb{G}$ for $i \in \{1, \ldots, m\}$, are the valid level-1 and blinded level-1 encodings of y_i respectively. The k_i's are the random integers and $\{y_1, \ldots, y_m\}$ is a super-increasing set of private integers. For any integer $r_i \in \mathbb{S}$, the image of r_i is defined in the ring $\mathbb{R}$ and group $\mathbb{G}$, symbolized by $\acute{r}_i$, as follow:

$$\acute{r}_i = \sum_{j=1}^{m} y_j r_i^{(j)}, \tag{9}$$

where $r_i^{(j)}$ is the jth bit of r_i. That is users encode r and they obtain an encoding of the image of r in the ring $\mathbb{R}$ and group $\mathbb{G}$. The encoding algorithm is explained in the Section 3.1.
The $\{y_1, \ldots, y_m\}$ must be a super-increasing set, that is, there exists an injective relationship between r_i and $\acute{r}_i$, where $\acute{r}_i = y_1 r_i^{(1)} + \cdots + y_m r_i^{(m)}$. To generate this set in a secure manner, first, the two 80-bit random integers y_1 and y_2 must be chosen, where $y_2 > y_1$, followed by choosing $y_3 > y_2 + y_1$, and $y_4 > y_3 + y_2 + y_1$ and so on.
If $\{y_1, \ldots, y_m\}$ is not a super-increasing set, then there exist no one-to-one relationship between r_i and $\acute{r}_i$. In this case, there is a chance to have the two different integers r_i and r_j with the same images $\acute{r}_i = \acute{r}_j$. However, Because of the large space of 2^m and $\mathbb{Z}_N^*$, this probability is negligible.

Definition 8. *(GCDH and Ext-GCDH Problems). Given system parameter SP and $t+1$ level-1 encodings $[y_0]_1, [y_1]_1, \ldots, [y_t]_1$ as inputs, the Graded-CDH and Extraction Graded CDH problems are defined as follows [5].*

- ***(Graded CDH Problem)**: Output $[y_1 \times \cdots \times y_t]_{t-1}$.*
- ***(Ext-Graded CDH Problem)**: Output the extracted string of level-t encoding $[y_1 \times \cdots \times y_t]_t$, which is equal to level-t blinded encoding $[y_1 \times \cdots \times y_t]_t^b$.*

Because of its self-multilinearity property, the Graded-DDH problem is not hold in the proposed GES. Based on Theorem 1, the Graded-CDH in the proposed GES is as hard as integer factorization.

Theorem 1. *Computing $[y]_{t-1} = yp^{t-1} + z(q)$ from $[y]_t = yp^t + z(q)$ (downgrading the level of encodings) is as hard as integer factorization.*

Proof. Let $N = (2p+1)(2q+1)$ be a safe RSA modulus, $e_r = (N-1)/2 = p + (2p+1)q$ and $r \in_R Z_N^*$. Suppose that there exists an adversary able to downgrade the level of encoding, then there exists an algorithm B, which can factor N. The algorithm B sends $re_r = rp + z(q)$ to the adversary. Note that re_r is a valid level-1 encoding. Assume that the adversary is able to compute $r + z(q)$ and send it back to the algorithm B, then the algorithm B can compute $z(q) = r + z(q) - r$, which is a polynomial of q and can lead the attacker to factor N. ■

The Ext-Graded-CDH problem indicates that, it is not possible to compute $[y]_t^b = g^{yp^t} \bmod N$ from $[y]_t = yp^t + z(q)$. For this computation, the attacker must remove $z(q)$ from $[y]_t$ or extract the p-root of $(g^p)^{yp^t+z(q)} \bmod N$, which needs the knowledge of factorization of N. The representation of an element at different levels of encoding is different. The Ext-Graded-CDH problem is formally proved in Theorem 3.

3.1 Description of the proposed GES

The proposed GES consists of the following five algorithms:

1. **Instance generation** InstGen(1^λ). Given security parameter λ, this algorithm determines system parameters $SP = \{N, \mathbb{G}, \mathbb{R}, \mathbb{S}, g^p, W = \{(e_1, d_1), \ldots, (e_m, d_m)\}\}$, where N is a safe RSA modulus, $\mathbb{G}$ is a subgroup of $\mathbb{Z}_N^*$ of order q, g^p is a generator of $\mathbb{G}$, $\mathbb{R}$ is a ring of order q as defined in previous section and $e_i = [y_i]_1 = y_i p + k_i q \in \mathbb{R}$, $d_i = [y_i]_1^b = g^{py_i} \bmod N \in \mathbb{G}$ for $i \in \{1, \ldots, m\}$ are the valid level-1 and blinded level-1 encodings of y_i's. The k_i's are the random integers and $\{y_1, \ldots, y_m\}$ is a super-increasing set of private integers. The Key Generation Center (KGC) keeps $\{y_1, \ldots, y_m\}$ private.

2. **Obtaining a level-1 encoding of an element** Level-1-enc(SP). Any user chooses uniformly random integer r_i and then generates a level-1 encoding ($[r_i]_1 = e_i$) and a blinded level-1 encoding ($[r_i]_1^b = b_i$) of an integer $r_i \in \mathbb{S}$ by computing the following equation:

$$\begin{aligned} u_i =& r_i^{(1)} e_1 + \cdots + r^{(m)} e_m \\ =& ((r_i^{(1)}) y_1 + \cdots + (r_i^{(m)}) y_m) p + ((r_i^{(1)}) k_1 + \cdots + (r_i^{(m)}) k_m) q \\ =& p\acute{r}_i + z(q), \end{aligned} \tag{10}$$

$$\begin{aligned} b_i =& d_1^{r_i^{(1)}} \times \cdots \times d_m^{r_i^{(m)}} \\ =& g^{p(y_1(r_i^{(1)}) + \cdots + y_m(r_i^{(m)}))} \bmod N \\ =& g^{\acute{r}_i p} \bmod N, \end{aligned} \tag{11}$$

where $b_i \in \mathbb{R}$ and $u_i \in \mathbb{G}$, are the encoding of $\acute{r}$ in $\mathbb{R}$ and $\mathbb{G}$ respectively. The user i publishes u_i as level-1 encoding of the unknown value $\acute{r}_i = \sum_{j=1}^{m} y_j r_i^{(j)}$ and keeps b_i secret for himself. It is obvious that obtaining r_i from $u_i = \sum_{j=1}^{m} r_i^{(j)} e_j$ requires solving an instance of the SSP problem which is assumed to be intractable.

3. **Adding encodings** Add(α, β). The addition procedure is given the two blinded encodings $(\alpha, \beta) = (b_i, b_j)$ or the two non-blinded encodings $(\alpha, \beta) = (u_i, u_j)$

with arbitrary level κ as input, and will produce another element with the same level through the following equation:

$$u_i + u_j = [r_i]_\kappa + [r_j]_\kappa = (\acute{r}_i + \acute{r}_j)p^\kappa + z(q) = [r_i + r_j]_\kappa \tag{12}$$

$$b_i b_j = [r_i]^b_\kappa [r_j]^b_\kappa = g^{(\acute{r}_i)p^\kappa} g^{(\acute{r}_j)p^\kappa} = g^{(\acute{r}_i+\acute{r}_j)p^\kappa} \bmod N = [r_i + r_j]^b_\kappa, \tag{13}$$

where $z(q)$ is the encoding noise and does not affect the final result. Users are not allowed to perform any operation on elements of $\mathbb{S}$.

4. **Multiplying encodings** $\mathrm{Mul}([r]_{\kappa_i}, [r_j]_{\kappa_j})$. The multiplication procedure is given two non-blinded encodings $u_i = [r_i]_{\kappa_i}$ and $u_j = [r_j]_{\kappa_j}$ with arbitrary levels κ_i and κ_j as the input and multiplies them as follows:

$$u_i u_j = [r_i]_{\kappa_i}[r_j]_{\kappa_j} = (\acute{r}_i\acute{r}_j)p^{\kappa_i+\kappa_j} + z(q) = [r_i r_j]_{\kappa_i+\kappa_j}, \tag{14}$$

   where the result is a level-$(\kappa_i + \kappa_j)$ encoding of $r_1 r_2$. Users are not allowed to perform any operation on elements of $\mathbb{S}$.

5. **Extraction** $\mathrm{Ext}(N, [r_i]^b_{\kappa_i}, [r_j]_{\kappa_j})$. This algorithm is given an encoding $u_j = [r_j]_{\kappa_j}$ with any arbitrary level κ_j and a blinded level-κ_i encoding $b_i = [r_i]^b_{\kappa_i}$ as the input and then extracts the representation (blinded encoding) of $[r_i.r_j]^b_{\kappa_i+\kappa_j}$ at level $\kappa_i + \kappa_j$.

$$(b_i)^{u_j} = g^{p^{\kappa_i+\kappa_j}\acute{r}_i\acute{r}_j} \bmod N = [r_i r_j]^b_{\kappa_i+\kappa_j} \tag{15}$$

Let $a_1, a_2 \in \mathbb{S}$ be two integers and $a_3 = a_1 + a_2$, then $[a_3]_1 \neq [a_1]_1 + [a_2]_1$ (e.g. $[2]_1 + [4]_1 \neq [6]_1$). Because, the elements of $\mathbb{S}$ is mapped to $\mathbb{R}$ and $\mathbb{G}$ by injective functions $f_1 : \mathbb{S} \to \mathbb{R}$ and $f_2 : \mathbb{S} \to \mathbb{G}$ (i.e. functions f_1 and f_2 are not morphisms), then $f(u \times v) \neq f(u) \times f(v)$ and $f(u + v) \neq f(u) + f(v)$. As a result, no operation on $\mathbb{S}$ is defined, in the first place.This property does not cause any obstacle for the applications of the resultant GES such as key exchange and multilinear map, because, in these applications, extracting the value of an encoded element is not intended, and users just wants to securely obtain a unique key by using a public tool.

Parameter setting. For $\lambda = 80$ bit security the modulus of computation must be at least 1024-bit. Computing r_i from u_i is an instance of the SSP problem. The most efficient algorithms for solving the SSP problem have the complexity $O(mc)$, where c is a size of the largest member of W, $O(u\sqrt{m})$ and $O(2^{m/2})$. At $m = 2\lambda = 160$, it is necessary to assure that $u \geq c > 2^{80}$. Here, N is a safe RSA modulus, p and q are chosen to be at least 511-bit primes and all of the members of W are at least 512-bit integers, thus $u \geq c > 2^{511}$.

4 Multilinear Map

The proposed GES can be used to construct an efficient multilinear map. In this system, a trusted third party runs $\mathrm{InstGen}(1^\lambda)$ to generate system parameters

SP, and then it will announce it publicly. Then any user can generate his encoding, do various operations on encodings, and run a multilinear map in any arbitrary multilinearity-degree.

Instance Generation(1^λ). The instance generation algorithm takes the security parameter λ as input and then runs InstGen(1^λ). Finally, it publishes $SP = \{N, \mathbb{G}, \mathbb{R}, \$, g^p, W = \{(e_1, d_1), \dots, (e_m, d_m)\}\}$ as public parameters.

Element Encoding(SP). Given system parameters SP as input, user i generates a level-1 encoding ($u_i = [r_i]_1$) and a blinded level-1 encoding ($b_i = [r_i]_1^b$) of an arbitrary element r_i by running Level-1-enc(SP) algorithm. The user keeps b_i secret and publicly publishes e_i.

Group Operation(SP, α, β). This algorithm takes the system parameters SP and two encodings (α, β) as input, where $(\alpha, \beta) = ([r_i]_\kappa, [r_j]_\kappa)$ or two encodings $(\alpha, \beta) = ([r_i]_{\kappa_i}, [r_j]_{\kappa_j})$ or two blinded encodings $(\alpha, \beta) = ([r_i]^b_\kappa, [r_j]^b_\kappa)$. Users can compute $[r_i]^b_\kappa [r_j]^b_\kappa = [r_i + r_j]^b_\kappa = g^{p(\acute{r}_i + \acute{r}_j)} \bmod N$ and $[r_i]_\kappa + [r_j]_\kappa = [r_i + r_j]_\kappa = (\acute{r}_i + \acute{r}_j)p^\kappa + z(q)$ as well as multiple of two non-blinded level-κ_i and level-κ_j encodings by computing $[r_i]_{\kappa_i}[r_j]_{\kappa_j} = [r_i r_j]_{\kappa_i + \kappa_j} = (\acute{r}_i \acute{r}_j)p^{\kappa_i + \kappa_j} + z(q)$, where the result is at level $\kappa_i + \kappa_j$ and $z(q)$ is the noise of encoding.

Multilinear Map($SP, [r_{i_1}]^b_{\kappa_{i_1}}, [r_{i_2}]_{\kappa_{i_2}}, \dots, [r_{i_{t+1}}]_{\kappa_{i_{t+1}}}$). This algorithm is given the system parameters SP, t encodings $\{u_{i_2} = [r_{i_2}]_{\kappa_{i_2}}, \dots, u_{i_{t+1}} = [r_{i_{t+1}}]_{\kappa_{i_{t+1}}}\}$ with level $\{\kappa_{i_2}, \dots, \kappa_{i_{t+1}}\}$ and one blinded level-κ_{i_1} encoding $b_{i_1} = [r_{i_1}]^b_{\kappa_{i_1}}$ as input and generates a blinded level-$(\kappa_{i_1} + \cdots + \kappa_{i_{t+1}})$ encoding (representation) of $[r_{i_1} \dots r_{i_{\kappa+1}}]_{\kappa_{i_1} + \cdots + \kappa_{i_{t+1}}}$ as below

$$\begin{aligned} e(b_{i_1}, u_{i_2}, \dots, u_{i_{t+1}}) =& (b_{i_1})^{\prod_{j=2}^{t+1} u_{i_j}} \bmod N \\ =& g^{p^{(\kappa_{i_1} + \cdots + \kappa_{i_{t+1}})} \prod_{j=1}^{t+1} u_{i_j}} \textit{ mod } N \\ =& [r_{i_1} \dots r_{i_{t+1}}]^b_{\kappa_{i_1} + \cdots + \kappa_{i_{t+1}}} \end{aligned} \tag{16}$$

The following lemma proves that the proposed multilinear map is non-degenerate.

Lemma 1. *The proposed self-multilinear map is non-degenerate.*

Proof. Let the order of $\mathbb{G}$ and g^p be a prime integer q, then for any integer $y \in \mathbb{Z} - \{q, 2q, \dots\}$, $(g^p)^y$ is also a generator of $\mathbb{G}$. Next to this, because, there is not any two integers y_1 and y_2 such that $y_1 y_2 = 0 \bmod q$, the proposed multilinear map is non-degenerate. ■

The proposed multilinear map needs an off-line trusted party to perform a trusted setup. However, the proposed self-multilinear map does not need KeyGen phase as in MP-NIKE schemes, and it is not limited to a predefined number of users. Because of self-multilinearity property, the output of the self multilinear map can be used as an input for another self-multilinear map.

5 Efficient Multi-party Non-interactive Key Exchange (MP-NIKE) scheme without key generation algorithm run by TTP

The proposed GES-based MP-NIKE scheme does not require an on-line key generation server since any user can generate his public/private keys. Suppose $k+1$ users aim to compute a shared key. The proposed MP-NIKE scheme has the following three algorithms.

1. **Setup**(1^{λ}). This algorithm takes as input the security parameter λ and runs **InstGen**(1^{λ}).
2. **Publish**(SP). In this algorithm, given the system parameters SP, every user runs **Level-1-enc**(SP). So, at the end of this algorithm, any user, for example the i-th user, will receive k encodings $u_1 = [r_1]_1, \ldots, u_{i-1} = [r_{i-1}]_1, u_{i+1} = [r_{i+1}]_1, \ldots, u_{k+1} = [r_{k+1}]_1$ from other k users. These encodings are regarded as other users' public keys. Moreover, each user, for example the i-th user, generates a blinded level-1 encoding $b_i = [r_i]_1^b$ which is regarded as his private key.
3. **ShareKey**$(N, \{[r_1]_1, \ldots, [r_{i-1}]_1, [r_{i+1}]_1, \ldots, [r_{k+1}]_1\}, [r_i]_1^b)$**.** This algorithm is given the system parameters, k level-1 encodings $\{[r_1]_1, \ldots, [r_{i-1}]_1, [r_{i+1}]_1, \ldots, [r_{k+1}]_1\}$ and a blinded level-1 encoding $b_i = [r_i]_1^b$ and outputs the representation of $r_1 \times r_2 \times \cdots \times r_{k+1}$ at level k. In the proposed MP-NIKE scheme, users are not limited to a predefined level K and the representation of encoded element can be extracted in any arbitrary level.

$$\begin{aligned} e(b_i, u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_{k+1}) =& (d_i)^{\prod_{j=1,j\neq i}^{k+1} u_j} \bmod N \\ =& g^{p^{k+1} \prod_{j=1}^{k+1} \acute{r}_{i_j}} \; mod \; N \end{aligned} \tag{17}$$

For example, suppose that at the end of **Publish**(SP) the user 1 receives three level-1 encodings $u_2 = y_2 p + k_2 q \bmod N$, $u_3 = y_3 p + k_3 q$ and $u_4 = y_4 p + k_4 q$. It wants to obtain a representation of $y_1 y_2 y_3 y_4$ at level 4. So, it computes $d_1^{u_j}$, where $u_j = u_2 u_3 u_4$. Note that the result does not depend on the coefficient of q.

6 ID-Based MP-NIKE

In ID-based cryptography, the public key of users is obtained directly from the identity of users. Fortunately, in the MP-NIKE scheme of [1] any $logN$-bit random integer r can be a valid public key and the KGC has a trapdoor to compute its respective private key, while others cannot do it. The TTP can use algorithm ID-KeyGen(SP, ID_i) to compute the private key of user i.

ID-KeyGen(N, g, msk, ID_i). This algorithm takes the modulus of computation, generator g, master secret key $msk = \{p, q\}$ and identity of user i as input and after user authentication, it generates a valid private key for user i. Let $N = \acute{p}\acute{q} = (2p + 1)(2q + 1)$ be a n-bit safe modulus, $H : \{0, 1\}^* \longrightarrow \{0, 1\}^n$ be a secure cryptographic hash function and $h_i = H(ID_i)$ be the public key of user i. Any integer bigger than Frobenius number $(pq-p-q)$ can be represented by linear combination of p and q [25]. So, any n-bit random integer r can be valid public key of MP-NIKE scheme of [1].

The KGC computes the private key of user i as below

$$\begin{aligned} h_i p^{-1} \bmod q =& y_{h_i} p p^{-1} + k_{h_i} q p^{-1} \bmod q \qquad (18) \\ =& y_{h_i} p p^{-1} \bmod q \\ =& y_{h_i} \bmod q. \end{aligned}$$

Now the algorithm ID-KeyGen generates the private key for user i as $g^{p y_{h_i}} \bmod N$.

7 Security Analysis

The security of this proposed GES scheme is proved in this Section. First, in Theorem 2, the intractability of decompositioning an encoding $u = [r]_1$ to $e_1 r^{(1)} + \cdots + r^{(j)} e_j + \cdots + r^{(m)} e_m$ and recovering r is proved.

Theorem 2. *The adversary that has access to system parameters $SP = \{N, \mathbb{G}, \mathbb{R}, \mathbb{S}, g^p, W = \{(e_1, d_1), \ldots, (e_m, d_m)\}\}$ and a non-blinded level-1 encoding $u = [r]_1 = e_1 r^{(1)} + \cdots + r^{(j)} e_j + \cdots + r^{(m)} e_m$ cannot compute r.*

Proof. Suppose that there is an adversary A which can compute r from $u = [r]_1$, then there is an algorithm B which can solve an instance of the SSP problem. Algorithm B is given a non-super-increasing set of m-bit integers $\{e_1, \ldots, e_m\}$ as input and must find all the subsets whose sum of elements is equal to u. The algorithm B generates a safe RSA modulus $N = (2p + 1)(2q + 1)$, choose a generator $g \in \mathbb{Z}_N^*$ and then computes

$$y_i = e_i p^{-1} \bmod q \qquad (19)$$

$$d_i = g^{p y_i} \bmod N. \qquad (20)$$

Note that any integer larger than Frobenius number $(pq - p - q)$ can be a valid public encoding in the proposed GES. Finally the algorithm B sends $SP = \{N, \mathbb{G}, \mathbb{R}, \mathbb{S}, g^p, W = \{(e_1, d_1), \ldots, (e_m, d_m)\}\}$ and u to the adversary A. If the adversary A can computes r such that $u = r^{(1)} e_1 + \cdots + r^{(m)} e_m$ and sends it back to B. Then algorithm B outputs $r^{(1)}, \ldots, r^{(m)}$ (r) as the solution of the SSP problem. ■

Theorem 3 proves that the security of the proposed scheme is equal to the MP-NIKE scheme of [1].

Theorem 3. *The security of the proposed GES is equal to the MP-NIKE scheme of [1].*

Proof. Suppose that there exists an adversary A that can breaks the security of the proposed GES scheme, then there exists an algorithm B that can break the security of the MP-NIKE scheme of [1]. Suppose that there is an algorithm B which is given the public parameter $PP = \{N, H(\cdot), g^p\}$ and m public/private key pairs $W = \{(e_1, d_1), \ldots, (e_m, d_m)\}$ as input and it must compute the shared key of group $W^* = \{u_1, \ldots, u_s\}$, where u_i is a valid public key of [1] MP-NIKE scheme. The algorithm B gives $SP = \{N, \mathbb{G}, \mathbb{R}, \mathbb{S}, g^p, W = \{(e_1, d_1), \ldots, (e_m, d_m)\}\}$ as the system parameters and $W^* = \{u_1, \ldots, u_s\}$ as a valid level-1 encodings to the attacker A. If the adversary A can computes Ext-Graded CDH$(u_1, u_2, \ldots, u_s)$ and return it to the algorithm B, then the algorithm B can solve the inputted problem. ■

8 Conclusion

In this article, a new efficient GES is introduced, where users are simply able to generate an encoding of the image of an element, without re-randomization. In this scheme, the users are not able to generate several encodings of the same element, while the noise of encoding is always fixed, thus, the zero testing parameter and re-randomization are not necessary. The extraction algorithm can run on any encoding with any arbitrary level. This GES can be applied in constructing a multilinear map. This GES multilinear map is secure, efficient and practical in different applications, like the multi-party non-interactive key exchange without the key generation algorithm run by TTP. The MP-NIKE scheme of [1] is improved here and turned to an ID-based scheme.

References

1. M. Salimi, H. Mala, H. Martin and P. Peris-Lopez. Full-Resilient Memory-Optimum Multi-Party Non-Interactive Key Exchange. In IEEE Access, 8(1), 2020. pp. 8821-8833.
2. D. Boneh and A. Silverberg. Applications of multilinear forms to cryptography. Contemporary Mathematics, 2002, 324, pp. 71-90.
3. S. Garg, C. Gentry and S. Halevi. Candidate multilinear maps from ideal lattices. Proc. Int. Conf. Advances in Cryptology-EUROCRYPT, LNCS 7881, 2013, pp. 1-17.
4. J. S. Coron, T. Lepoint and M. Tibouchi. Practical multilinear maps over the integers. Proc. Int. Conf. Advances in Cryptology-CRYPTO, LNCS 8042, 2013, pp. 476-493.
5. A. Langlois, D. Stehlae and R. Steinfeld. GGHLite: More Efficient Multilinear Maps from Ideal Lattices. Proc. Int. Conf. Advances in Cryptology-EUROCRYPT, LNCS 8441, 2014, pp. 239-256.
6. M. R. Albrecht, C. Cocis, F. Laguillaumie and A. Langlois. Implementing Candidate Graded Encoding Schemes from Ideal Lattices, IACR Cryptology ePrint Archive, Report 2014/928, 2014, http://eprint.iacr.org/2014/928.

7. Y. Hu and H. Jia. Cryptanalysis of GGH map. Technical report, IACR Cryptology ePrint Archive, Report 2015/301, 2015, http://eprint.iacr.org/2015/301.
8. J. H. Cheon, K. Han, C. Lee, H. Ryu and D. Stehl'e. Cryptanalysis of the multilinear map over the integers. Proc. Int. Conf. Advances in Cryptology-EUROCRYPT, LNCS 9056, 2015, pp. 3-12.
9. C. Gentry, S. Gorbunov and S. Halevi. Graph-induced multilinear maps from lattices. Proc. Int. Conf. Theory of Cryptography(TCC), LNCS 9015, 2015, pp. 498-527.
10. J. S. Coron, M. S. Lee, T. Lepoint and M. Tibouchi. Cryptanalysis of GGH15 Multilinear Maps, IACR Cryptology ePrint Archive, Report 2015/1037, 2015, http://eprint.iacr.org/2015/1037.
11. J. S. Coron, T. Lepoint and M. Tibouchi. New Multilinear Maps over the Integers. Proc. Int. Conf. Advances in Cryptology - CRYPTO 2015, LNCS 9215, 2015, pp. 267-286.
12. B. Minaud and P. A. Fouque. Cryptanalysis of the New Multilinear Map over the Integers. Cryptology ePrint Archive, Report 2015/941, 2015.
13. K. Lewi, A. J. Malozemoff, D. Apon, B. Carmer, A. Foltzer, D. Wagner, D. W. Archer, D. Boneh, J. Katz and M. Raykova. 5Gen: A Framework for Prototyping Applications Using Multilinear Maps and Matrix Branching Programs. IACR Cryptology ePrint Archive, Report 2016/619, 2016.
14. F. Ma, and M. Zhandry. The MMap strikes back: obfuscation and new multilinear maps immune to CLT13 zeroizing attacks. In Theory of Cryptography (TCC18), 2018, pp. 513-543.
15. M. R. Albrecht, P. Farshim, D. Hofheinz, E. LarraiaKenneth, G. Paterson. Multilinear Maps from Obfuscation. Proc. Int. Conf. Theory of Cryptography Conference-TCC. 2016, pp.446-473.
16. M. R. Albrecht, P. Farshim, D. Hofheinz, E. LarraiaKenneth, G. Paterson. Multilinear Maps from Obfuscation. Journal of Cryptology, 2020. pp. 1-34.
17. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. Proc. Int. Conf. Foundations of Computer Science(FOCS), 2013, pp. 40-49.
18. S. Garg, C. Gentry, S. Halevi, M. Zhandry. Fully Secure Attribute Based Encryption from Multilinear Maps. IACR Cryptology ePrint Archive 2014: 622 (2014).
19. Sanjam Garg, Craig Gentry, Shai Halevi, and Mark Zhandry. Functional Encryption without Obfuscation. Proc. Int. Conf. Theory of Cryptography Conference-TCC. 2016, pp.480-511.
20. K. Koiliaris and C. Xu. "A Faster Pseudopolynomial Time Algorithm for Subset Sum". arXiv:1507.0231, 2015.
21. J. Kleinberg and V. Tardos. Algorithm Design (2nd ed.). p. 491. ISBN 0-321-37291-3. 2006.
22. D. Pisinger. "Linear Time Algorithms for Knapsack Problems with Bounded Weights". Journal of Algorithms, Volume 33(1), pp. 114, 1999.
23. T. Yamakawa, S. Yamada, G. Hanaoka and N. Kunihiro. Self-bilinear Map on Unknown Order Groups from Indistinguishability Obfuscation and Its Applications. Proc. Int. Conf. Advances in Cryptology-CRYPTO, LNCS 8617, 2014, pp. 90-107.
24. J. H. Cheon, D. H. Lee. A NOTE ON SELF-BILINEAR MAPS. Bulletin of the Korean Mathematical Society. 46(2), 2009, pp. 303–309.
25. J. Sylvester. Mathematical questions, with their solutions. Educational Times 41. pp. 21. 1884.