Self-bilinear Map from One Way Encoding System and Indistinguishability Obfuscation

Huang Zhang[1,4], Fangguo Zhang[1,4], Baodian Wei[2,4], and Yusong Du[3]

[1] School of Information Science and Technology, Sun Yat-sen University
[2] School of Data and Computer Science, Sun Yat-sen University
[3] School of Information Management, Sun Yat-sen University
[4] Guangdong Key Laboratory of Information Security Technology
Guangzhou 510006, China
isszhfg@mail.sysu.edu.cn

Abstract. The bilinear map whose domain and target sets are identical is called the self-bilinear map. Original self-bilinear maps are defined over cyclic groups. This brings a lot of limitations to construct secure self-bilinear schemes. Since the map itself reveals information about the underlying cyclic group, hardness assumptions on DDHP and CDHP may not hold any more. In this paper, we used $i\mathcal{O}$ to construct a self-bilinear map from generic sets. These sets should own several properties. A new notion, One Way Encoding System (OWES), is proposed to describe formally the properties those sets should hold. An Encoding Division Problem is defined to complete the security proof. As an instance of the generic construction, we propose a concrete scheme built on the GGH graded encoding system and state that any 1-graded encoding system may satisfy the requirements of OWES. Finally, we discuss the hardness of EDP in the GGH graded encoding system.

Keywords: self-bilinear map, multi-linear map, indistinguishability obfuscation, one way encoding system.

1 Introduction

The bilinear map is a very useful cryptographic primitive. It provides solutions for many cryptographic applications such as identity-based encryptions [2], non-interactive zero-knowledge proof systems [16], attribute-based encryptions [21] and short signatures [3, 23], etc. The self-bilinear map is a special variant of bilinear maps. The target and preimage groups of self-bilinear maps are the same. Because of this exclusive property, the self-bilinear map may have more interesting potentials. A straightforward application of the self-bilinear map is constructing multilinear maps.

Multilinear maps are generalized notion from bilinear maps. Not long after bilinear maps show the convenience they bring to the cryptography, Boneh and Silverg [4] imaged applications of the multilinear maps. But, when they tried to find such a fantastic tool, they met serious obstacles. From then on, constructing multilinear maps became a long-standing open problem. Until recently, three candidate

multilinear maps were proposed – the GGH scheme [11] on ideal lattices, the CLT scheme [9] over the integer and the GGH14 [15] on lattices. The multilinear map is a basic component of some cryptographic primitives such as witness encryption [14], indistinguishability obfuscation and functional encryption [12], etc.

Recently, multilinear maps met extremely strong challenges. CLT schcme was completely broken by "zerozing algorithm" [6]. Two patches [13, 5] were proposed very soon after the CLT was broken. But Coron et al. [10] stated that these two patches were still unsafe. Then, they described a new multilinear map over the integer [8]. Not long after the CLT scheme was completely broken, the GGH scheme was also under attack. Hu and Jia constructed a modified encoding/decoding algorithm [17] and designed a weak-DL attack to break the MDDH assumption which is the security basis of many schemes. There isn't any patch that can fix this weakness. To construct a secure and efficient multilinear map is still a worthwhile work. This also highlights the study of finding a wonderful self-bilinear map.

Lee [19] designed the first candidate self-bilinear map. But Cheon and Lee [7] remarked that Lee's map is not essentially self-bilinear. They also proved the impossibility that the secure self-bilinear map can't be constructed over cyclic group of known prime order. The computational Diffie-Hellman (CDH) assumption collapses because the map itself reveals much information about the underlying group. To avoid this, Yamakawa et al. [22] chose the signed quadratic residue group $\mathbb{QR}_n^+$ of $\mathbb{Z}_n^*$ to be the underlying group. The order of $\mathbb{QR}_n^+$ is composite and unknown. The factoring assumption is a basic hardness assumption in their security proof.

Unlike other's work, we prefer to build self-bilinear maps over generic sets instead of cyclic groups. A new concept OWES is defined to describe the generic sets that can be used to build self-bilinear maps. In order to complete the security proof, some hard problems are assumed to be hard in the OWES. We show that the graded encoding system (GES) is an instance of the OWES. Based on the GGH grade encoding system, a concrete construction of the self-bilinear map is proposed. We also discussed the security of the concrete scheme.

Through the work of self-bilinear maps from $i\mathcal{O}$, we find that multilinear maps can be built by making use of $i\mathcal{O}$. On the contrary, it's like a paradox that the first $i\mathcal{O}$ was designed from multilinear maps. Coincidentally, some recent works try to study the relationship between the multilinear map and the obfuscation. Paneth et al. [20] defined a variant of secret sampling multilinear map. They named it the polynomial jigsaw puzzle. The polynomial jigsaw puzzle can be used to construct $i\mathcal{O}$ and $i\mathcal{O}$ implies the polynomial jigsaw puzzle. Albrecht et al. [1] proposed a multilinear map scheme from obfuscation. These work are not similar to ours. Whether multilinear map and obfuscation are essentially the same conception is still an open problem to discuss.

We organize this paper as follows. The cryptographic tools and notations we used in this paper are introduced in Section 2. We proposed a generic construction of self-bilinear map based on OWES and $i\mathcal{O}$ in Section 4. In Section 5, we build a concrete self-bilinear map from GGH graded encoding system and $i\mathcal{O}$. We analyse the EDP in Section 6.

2 Preliminaries

In this section, we describe notations used in this paper, review the $i\mathcal{O}$, and propose the new concept of OWES. We also present the formal definition of the variant of self-bilinear maps and multilinear maps.

2.1 Notations

We use $\mathbb{Z}$ to denote the set of all integer numbers, $\mathbb{F}$ to denote a field. $\mathbb{Z}[x]$ to denote all polynomials with coefficients in $\mathbb{Z}$. Let $[n]$ be the set $\{x \in \mathbb{Z} | 1 \leq x \leq n\}$ and [0,n] be the set $\{x \in \mathbb{Z} | 0 \leq x \leq n\}$. λ is the secure parameter. We use $e \leftarrow D_{S,\sigma}$ to denote that e is sampled from the discrete distribution with mean 0 and standard deviation σ in set S. $\{x_i\}_{i=1}^n$ represent the set $\{x_1, \cdots, x_n\}$. If $\bar{a}$ is an element in a residue class ring R/I, then a is its representation in R.

2.2 Indistinguishability Obfuscation

Definition 1 (Indistinguishability Obfuscator). *A uniform* PPT *machine $i\mathcal{O}$ is called an indistinguishability obfuscator for a circuit class $\mathcal{C}_\lambda$ if the following conditions are satisfied:*

- *For security parameters $\lambda \in \mathbb{N}$, all $C \in \mathcal{C}_\lambda$, and all inputs x, we have that*

$$Pr[C'(x) = C(x) : C' \leftarrow i\mathcal{O}(\lambda, C)] = 1$$

- *For any (not necessarily uniform)* PPT *distinguisher D, there exists a negligible function α such that the following holds: For all security parameters $\lambda \in N$, and all pairs of circuits $C_0, C_1 \in \mathcal{C}_\lambda$, we have that if $C_0(x) = C_1(x)$ for all inputs x, then*

$$|Pr[D(i\mathcal{O}(\lambda, C_0)) = 1] - Pr[D(i\mathcal{O}(\lambda, C_1)) = 1]| \leq \alpha(\lambda)$$

Informally speaking, $i\mathcal{O}$ can make two circuits or functions which have the same size and achieve the same goal be computationally indistinguishable.

3 Self-bilinear maps and One Way Encoding Systems

Below we define formally our notion of self-bilinear maps from OWES and $i\mathcal{O}$. To make the analogy and differences from self-bilinear maps more explicit, we begin by recalling the definition of self-bilinear maps of Cheon and Lee [7].

Definition 2 (Self-bilinear map). *For a cyclic group G of order p, a map $e : G \times G \to G$ is self-bilinear, if it has the following properties.*

- *For all $g_1, g_2 \in G$ and integer $a \in \mathbb{Z}_p$, it holds that*

$$e(g_1^a, g_2) = e(g_1, g_2^a) = e(g_1, g_2)^a.$$

- *The map e is non-degenerate, i.e, if g_1, g_2 are generators of G, then $e(g_1, g_2)$ is a generator of G.*

3.1 One Way Encoding Systems

We define the new concept of One Way Encoding Systems (OWES) to summarize all the necessary properties to build self-bilinear maps. These properties were refined from cryptographic cyclic groups. Firstly, we generalize the $ag \in G$ as an encoding of ring element $\bar{a} \in \mathbb{Z}_{ord(n)}$, where a is a representation of $\bar{a}$. The encoding algorithm is define as $f(x) = xg$. We'll regard the ring as the plaintext space, and all possible encodings as the encoding space. Secondly, in order to simulate the cyclic group operation $ag+bg = (a+b)g$, the encoding system have to be additive homomorphic. Thirdly, Observing the valid self-bilinear map operation $e((a+b)g, \gamma g) = (a+b)\gamma e(g,g)$, we find that the self-bilinear map system doesn't need the plaintext space to be additively cyclic. So we extend the plaintext space to any finite ring with identity. The encodings space will not always be a cyclic group because of the change. Fourthly, since the represenctation of the element in plaintext space is not the integer anymore, we must define a new manipulation "$\otimes$" between representations and encodings. Namely, if x is a representation of an element in the plaintext space and h is an encoding, then $x \otimes g$ is a new encoding. Such manipulation is similar to the scalar multiplication in linear spaces. Finally, for cryptographical use, given f and $f(a)$, it should be hard to compute x. Moreover, given f, $f(a)$, $f(b)$, to distinguish $f(ab)$ from a random encoding should be hard. As a conclusion of all discussed above, here comes the formal definition of OWES and its hardness assumptions.

Assume that $(S_0, +, \cdot)$ is a finite ring with identity ('$\cdot$' will be omitted for simplicity), S_1 is a set with addition operation "$\oplus$", $f : S_0 \to S_1$ is a surjection. ('$\cdot$' will be omitted for simplicity))

Definition 3 (OWES). *An One Way Encoding System consits of S_0, S_1 and f, with following properties hold:*

1. *f is additive homomorphic: for $a, b \in S_0$, $f(a+b) = f(a) \otimes f(b)$.*
2. *Plaintext multiplication '$\otimes$': for $a \in S_0$ and $f(b) \in S_1$, $a \otimes f(b) = f(a \cdot b)$.*
3. *f is one way: give $f(a)$, it's hard to compute a*

Moreover, since the self-bilinear map reveal information about the underlying set, a new hard problem should be hard in the OWES.

Definition 4 (EDP). *For* OWES *(S_0, S_1, f), the Encoding Division Problem is, on input the $f(ab) \in S_1$ and invertible $a \in S_0$, to compute $f(b) \in S_1$.*

The OWES has a property similar to the linear transformation in linear space. If we change the f by multiplying it with an element in $b \in S_0$, $(S_0, b \otimes S_1, b \otimes f)$ is also an OWES (But $b \otimes S_1 \neq S_1$, if b is a zero divisor in S_0).

Theorem 1. *If (S_0, S_1, f) is an* OWES *and the element $b \in S_0$, $(S_0, b \otimes S_1, b \otimes f)$ is also an* OWES*.*

Proof. The Property 1 and Property 2 obviously hold in $(S_0, b \otimes S_1, b \otimes f)$. We proof the Property 3 hold below. For simplicity, let $f' = b \otimes f$. Given $f'(a)$, If $\mathcal{A}$ can compute a efficiently, we design a algorithm to break Property 3 in (S_0, S_1, f).

Assume that (S_0, S_1, f) and $b \in S_0$ are the public parameters. The challenger receives a problem instance $f(a)$. The simulator computes $f'(a) = b \otimes f(a)$ and sends $f'(a)$ to $\mathcal{A}$. $\mathcal{A}$ return a with non-negligible probability. So if the Property 3 holds in (S_0, S_1, f), it also holds in $(S_0, b \otimes S_1, b \otimes f)$. Thus $(S_0, b \otimes S_1, b \otimes f)$ is an OWES. □

Self-bilinear map from OWES. The self-bilinear map with auxiliary information was first defined by Yamakawa et al [22]. We recall the definition with a little modifications. The Self-bilinear map with auxiliary information consists of efficient procedures for instance-generation, element-encoding, addition and negation, self-bilinear map, $\mathcal{SBP}$ =(**InstGen**, **Enc**, **Add**, **Neg**, **Map**). These procedures are described below.

Instance Generation. The randomized $\mathbf{InstGen}(1^\lambda)$ takes as input the parameter λ, and outputs **params**, where **params** is a description of the self-bilinear map $e : S_1 \times S_1 \to S_1$.

Encoding. The **Enc**(**params**, a) takes **params** and a plaintext $a \in S_0$, and outputs the encoding $f(a)$ and the auxiliary information $\tau_{f(a)}$.

Addition. Given **params**, two encodings $f(a), f(b) \in S_1$, $\tau_{f(a)}, \tau_{f(b)}$, we have $\mathbf{Add}(\mathbf{params}, f(a), f(b), \tau_{f(a)}, \tau_{f(b)})$ output the encoding $f(a+b)$, and the auxiliary information $\tau_{f(a+b)}$.

Self-bilinear Map. For $f(a), f(b) \in S_1$, we have $\mathbf{Map}(\mathbf{params}, f(a), f(b)) = e(f(a), f(b))$.

Hardness Assumptions. Our hardness assumptions are modeled after the discrete-logarithm, BCDH assumptions in self-bilinear groups. For example, the analog of discrete-logarithm problem is trying to obtain a from the encoding $f(a)$.

The analog of BCDH in our case roughly says that it is hard to compute map when somebody is given three or more encodings and the auxiliary information. In other words, given encodings $f(a)$, $f(b)$, $f(c)$, $\tau_{f(a)}$, $\tau_{f(b)}$, $\tau_{f(c)}$, the $c \otimes e(f(a), f(b))$ can't be obtained efficiently by BCDH challenger. We formalize the hardness assumption as follows.

Definition 5 (EBCDHP). *For a self-bilinear map $e : S_1 \times S_1 \to S_1$ and its underlying* OWES (S_0, S_1, f)*, the* EBCDHP *is, on input $f(a), f(b), f(c), \tau_{f(a)}, \tau_{f(b)}, \tau_{f(c)}$, to compute $c \otimes e(f(a), f(b))$.*

The EBCDH assumption says that for any setting of parameters, the probability of solving EBCDHP is negligible.

The OWES can be constructed. We will show in Section 5 that 1-GES (graded encoding system) of GGH scheme is an OWES. Furthermore, any proposed GES may satisfy the requirement of OWES by making some minor modifications. Unfortunately, we can't reduce EDP in GGH scheme to classical hard problems for this moment, some evidences will exhibit to increase the secure confidence.

4 Generic Construction from OWES and $i\mathcal{O}$

In this section, we construct a self-bilinear map scheme $\mathcal{SBP}$ from OWES and $i\mathcal{O}$.

4.1 Our Construction

In the $\mathcal{SBP}$, $i\mathcal{O}$ circuits act as the auxiliary information. We describe notations for circuits on OWES first.

Notation for Circuits on OWES. For the OWES (S_0, S_1, f) and $a_1 \in S_0$, $C_{a_1}(x)$ denotes the circuit that takes $x \in S_1$ as input and computes $a_1 \otimes x$. For circuits $C_{a_1}(x)$, $C_{a_2}(y)$ whose outputs can be parsed as elements in S_1, $\text{Plus}(C_{a_1}(x), C_{a_2}(y))$ denotes a circuit that computes the sum of outputs of $C_{a_1}(x)$ and $C_{a_2}(y)$.

Now we are ready to to introduce the procedures of of the generic constructing $\mathcal{SBP}$. The generic construction of self-bilnear map is shown in Fig.1.

Instance Generation: params $\leftarrow$ **InstGen**$(\mathbf{1}^\lambda)$.
- On input the security parameter λ, initiate an OWES (S_0, S_1, f).
- Choose an invertible element $r \in S_0$ at random.
- Output **params** $= (S_0, S_1, f, r)$ as the system parameters.

After the **InstGen** procedure is executed, the self-bilinear map e is defined as:

$$\begin{aligned} e: \quad S_1 \times S_1 \quad &\to \quad S_1 \\ (f(a_1), f(a_2)) &\mapsto f(ra_1a_2) \end{aligned}$$

Encoding: $(f(a), \tau_{f(a)}) \leftarrow$ **Enc**(**params**, **a**).

- On input **params** and $a \in S_0$, compute $f(a)$.
- Generate the corresponding $\tau_{f(a)} = i\mathcal{O}(C_{ra})$

Self-biliner Map: $f(ra_1a_2) \leftarrow$ **Map**(**params**, $f(a_1), \tau_{f(a_2)}$)**.**

- On input $f(a_1)$, run the obfuscated circuit $\tau_{f(a_2)}$ to compute $\tau_{f(a_2)}(f(a_1)) = f(ra_1a_2)$.

Addition: $(f(a_1{+}a_2), \tau_{f(a_1+a_2)}) \leftarrow$ **Add**(**params**, $f(a_1), f(a_2), \tau_{f(a_1)}, \tau_{f(a_2)}$)**.**

- On input $f(a_1), f(a_2) \in S_1$ and the corresponding $\tau_{f(a_1)}, \tau_{f(a_2)}$, compute $f(a_1 + a_2) = f(a_1) \oplus f(a_2)$.
- Compute $\tau_{f(a_1+a_2)} \leftarrow i\mathcal{O}(\text{Plus}(\tau_{f(a_1)}, \tau_{f(a_2)}))$

Fig. 1. The generic construction of $\mathcal{SBP}$

4.2 Security Analysis of $\mathcal{SBP}$.

We give a polynomial reduction from the EDP to the EBCDHP. In Section 6, we'll give an instance of EDP, and analyze its hardness.

Theorem 2. *If there is a PPT algorithm $\mathcal{A}$ solving the* EBCDHP *with respect to $\mathcal{SBP}$ efficiently, then there is an algorithm to solve the* EDP *in the* OWES *efficiently.*

Proof. We assume that an algorithm $\mathcal{A}$ can solve the EBCDHP in $\mathcal{SBP}$. We design an algorithm $\mathcal{C}$ to solve the EDP with high probability by running $\mathcal{A}$ as the subroutine.

Reduction Algorithm $\mathcal{C}$:

1. $\mathcal{O}$ outputs an EDP instance $(S_0', S_1', f'), f'(rb), r$.
2. Compute $f'(r^2 b) = r \otimes f'(rb)$. Then, set $S_0 = S_0'$, $f(x) = x \otimes f'(r^2 b)$, $S_1 = \{f(x) | x \in S_0\}$. Finally, output **params** $= (S_0, S_1, f, r)$. **params** describes a $\mathcal{SBP}$. (note that (S_0, S_1, f) is also an OWES).
3. Choose $a_0', a_1', a_2' \in S_0$ uniformly at random.
4. Compute $ra_i' + 1$ and sets $ra_i = ra_i' + 1$. Thus, $f(a_i) = (ra_i' + 1) \otimes f'(rb)$, for $i = 0, 1, 2$. (note that r is invertible in S_0).
5. Generate the auxiliary information $\tau_{f(a_i)} = i\mathcal{O}(C_{ra_i}) = i\mathcal{O}(C_{ra_i'+1})$, for $i = 0, 1, 2$.
6. Send **params**, $\{f(a_i)\}_{i=0}^2$ and $\{\tau_{f(a_i)}\}_{i=0}^2$ to $\mathcal{A}$.
7. If $\mathcal{A}$ thinks he is playing with a $\mathcal{SBP}$ scheme, returns U. Otherwise, $\mathcal{A}$ returns nothing and $\mathcal{C}$ aborts.
8. $\mathcal{C}$ checks the value of U, if $U \neq f(ra_0 a_1 a_2)$, aborts.
9. Compute $q = \frac{(ra_1'+1)(ra_2'+1)-1}{r} = ra_1' a_2' + a_1' + a_2'$.
10. Compute $p = a_0'(ra_1' + 1)(ra_2' + 1)$, and output $U' = U - [p + q] \otimes f'(rb)$.

Correctness: If the reduction algorithm $\mathcal{C}$ complete its execution without aborting, the output U' is the answer of EDP $(f(rb), b)$.

$$
\begin{aligned}
U &= f(ra_0 a_1 a_2) \\
&= f'(ra_0 a_1 a_2 r^2 b) \\
&= f'[a_0 r a_1 r a_2 r b] \\
&= f'[(a_0' + \tfrac{1}{r})(ra_1' + 1)(ra_2' + 1) rb] \\
&= f'[(a_0'(ra_1' + 1)(ra_2' + 1) + \tfrac{(ra_1'+1)(ra_2'+1)}{r}) rb] \\
&= f'[(a_0'(ra_1' + 1)(ra_2' + 1) + \tfrac{(ra_1'+1)(ra_2'+1)-1}{r} + \tfrac{1}{r}) rb] \\
U' &= U \ominus [p + q] \otimes f'(rb) \\
&= U \ominus [a_0'(ra_1' + 1)(ra_2' + 1) + \tfrac{(ra_1'+1)(ra_2'+1)-1}{r}] \otimes f'(rb) \\
&= U \ominus f'[(a_0'(ra_1' + 1)(ra_2' + 1) + \tfrac{(ra_1'+1)(ra_2'+1)-1}{r}) rb] \\
&= f'[(\tfrac{1}{r}) rb] \\
&= f'(b)
\end{aligned}
$$

Probability: We analyze the probability that $\mathcal{C}$ aborts. In the step 7, $\mathcal{A}$ end this algorithm if the parameters he received is not come from a $\mathcal{SBP}$ scheme. According to the theorem 3.1, (S_0, S_1, f) is an OWES if (S_0', S_1', f') is. The auxiliary $i\mathcal{O}$ circuits in $\mathcal{SBP}$ achieved the expected function. So, the probability that algorithm abort in step 7 is negligible. Since we have assumed that $\mathcal{A}$ can solve the EBCDHP in $\mathcal{SBP}$ efficiently, the algorithm $\mathcal{C}$ through the step 8 with non-negligible probability. Thus, the algorithm $\mathcal{C}$ will not abort with a non-negligible probability.

Time complexity: We use $T(\cdot)$ to denote the time complexity. Besides the subrouting $\mathcal{A}$, the times of manipulations in each step of $\mathcal{C}$ is a constant. Assume that the sum of these constant is t. The time complexity of each manipulation is a polynomial $poly(\lambda)$, since they are efficiently computable (addition in a ring etc). Thus, the time complexity of the algorithm $\mathcal{C}$ is bounded by $T(\mathcal{C}) = t{\cdot}poly(\lambda){+}T(\mathcal{A})$. Since $\mathcal{A}$ is assumed to be an efficient algorithm, $T(\mathcal{A})$ is bounded by $poly(\lambda)$. So, $T(\mathcal{C}) = poly(\lambda)$ which means $\mathcal{C}$ is efficiently computable.

In summary, the algorithm C is a polynomial reduction from EDP to EBCDHP. Since EDP is hard, the algorithm that can solve EBCDHP doesn't exist.

Remark In the algorithm $\mathcal{C}$, we compute $ra_1'a_2' + a_1' + a_2'$ instead of computing $\frac{(ra_1'+1)(ra_2'+1)-1}{r}$ in step 9. This is because of the uncovered division algorithm in S_0. If S_0 is the integer ring or polynomial ring (it has the division algorithm), the division algorithm may be used reduce the time complexity of the step 9. As the number of term increase, to compute $\frac{\prod_{i=1}^{n}(ra_i'+i)-1}{r}$ is much easier.

5 Concrete Construction from GGH and $i\mathcal{O}$

The OWES is not an unpractical concept. The GGH graded encoding system [11] is an instance of the OWES. We'll introduce a concrete construction from the graded encoding system (GES) and $i\mathcal{O}$.

5.1 An Instance of OWES

We can get an instance of the OWES out of the GGH graded encoding system, only considered the 1-GES. Let r, d, d' denote the level-0 encodings, $c^{(d)}$ denotes the level-1 encoding of $I + d$, where I is the ideal. The four properties of OWES hold in 1-GES.

1. additive homomorphic: $c^{(d)} + c^{(d')} = c^{(d+d')}$
2. plaintext multiplication: $d \otimes c^{(d)} = c^{(dd')}$
3. Given the level-1 encoding $c^{(d)}$, it is hard to compute the level-0 encoding d, where d is a short element in $I + d$.

The EDP in 1-GES is, given $c^{(rd)}, r$, to compute $c^{(d)}$. It seems that EDP is hard in 1-GES, the further consideration of EDP is in Section 6.

5.2 Construction

Depending on the security parameter λ, we choose the ring $R = \mathbb{Z}[x]/(x^n + 1)$ and $R_q = R/qR$, where q, n, m (mentioned below) are function of λ. Elements in the underlying ring can be regarded as polynomials. The system encodes elements of a quotient ring $QR = R/I$, where $I = \langle g \rangle$ and $g \in R$. We will use the symbol $c^{(d)}$ to denote the encodings of element d. Since 1-GES is an instance of the OWES, we assume that the notation for circuit on OWES defined in Section 4.1 is still worked here. The concrete self-bilinear map scheme is shown in Fig.2

Instance Generation: params $\leftarrow$ InstGen(1^λ).

- Take as input the security parameter λ, generate the 1-GES. It is described by the following parameters. $y = [\frac{a}{z}]_q$, the level-1 encoding of $I + 1$. $x_i = [\frac{b_i}{z}]_q$, $i = [m]$, the re-randomization parameters. x_i is the level-1 encoding of I. The zero testing parameter $P_{zt} = [hz/g]_q$, where h is "somewhat small".
- Choose a random element $\alpha \leftarrow D_{R,\sigma'}$.
- Choose a random element $s \leftarrow D_{\mathbb{Z}^m,\sigma'}$, and compute $v = s \cdot y$.
- Define **params** $= (R/I, R_q, v, \{x_i\}_{i=1}^m, \alpha, P_{zt})$ and makes them public.

Here, R/I acts as S_0 in OWES, R_q plays a role of S_1, and $v, \{x_i\}_{i=1}^m$ are the encoding algorithm f. P_{zt} helps to check whether an element equals to the other. After the instance generation procedure is executed, the self-bilinear map e is difined as

$$\begin{aligned} e: \ R_q \times R_q &\rightarrow R_q \\ (c^{(d)}, c^{(d')}) &\mapsto c^{(\alpha dd')} \end{aligned}$$

Encode: $(c^{(d)}, \tau_{c^{(d)}}) \leftarrow$ **Encode**$($**params**$, d)$**.**

- Compute $c^{(d)} = [dv + \sum_{i=1}^m r_i x_i]_q$, where $r \leftarrow D_{\mathbb{Z}^m,\sigma^*}$, $\sigma^* = 2^\lambda$.
- Generate the corresponding auxiliary information $\tau_{c^{(d)}} = i\mathcal{O}(C_{\alpha d})$.

Addition: $(c^{(d+d')}, \tau_{c^{(d+d')}}) \leftarrow$ **Add**$($**params**$, c^{(d)}, c^{(d')}, \tau_{c^{(d)}}, \tau_{c^{(d')}})$**.**

- Compute $c^{(d+d')} = [c^{(d)} + c^{(d')}]_q$ directly.
- Generate the auxiliary information as $\tau_{c^{(d+d')}} \leftarrow i\mathcal{O}(\text{Plus}(\tau_{c^{(d)}}, \tau_{c^{(d')}}))$.

Self-bilinear Map: $c^{(\alpha dd')} \leftarrow$ **Map**$($**param**$, c^{(d)}, \tau_{c^{(d')}})$**.** Run the circuit $\tau_{c^{(d')}}(c^{(d)})$ to compute $c^{(\alpha dd')} = [\alpha d' c^{(d)}]_q$.

isZero$($**params**$, c)$. Output 1 if $||[P_{zt} c^{(d)}]_q|| < q^{3/4}$, otherwise output 0.

Fig. 2. The concrete construction of $\mathcal{SBP}$

5.3 Setting the Parameters

GGH are noise encodings. The noise level should never be too large. The setting of parameters should satisfy the basic GGH requirements.

- To sample the $g \leftarrow D_{\mathbb{Z}^n,\sigma}$, set $\sigma = \sqrt{\lambda n}$, σ should larger than the smoothing parameter $(\eta_{2^{-\lambda}}(\mathbb{Z}^n))$. As a result, the size of g is bounded with $||g|| \leq \sigma\sqrt{n} = n\sqrt{\lambda}$.
- To sample a_i, b_i and level-0 elements, set $\sigma = \lambda n^{3/2}$. Then, these elements are bounded by λn^2. GGH states that numerator in y and the x_i are bounded by σn^4.
- To sample $r \leftarrow D_{\mathbb{Z}^n,\sigma^*}$, set $\sigma^* = 2^\lambda$. As a result, the numerator x_i is bounded by $||c|| \leq 2^\lambda \cdot poly(n)$.
- The value of k-multilinear map of k encodings is essentially the product of one level-1 encoding and $k-1$ plaintext. Hence the numerate of this final encoding is bounded by $||c|| \leq 2^\lambda \cdot poly(n) \cdot (\lambda n^{3/2})^{k-1} = \lambda 2^\lambda n^{O(k)}$.
- To get λ-level security against lattice attacks, the dimension n should be roughly fixed so that $q < 2^{n/\lambda}$, which means that $n > \tilde{O}(\kappa\lambda^2)$.
- Finally, m should be larger than $n \log q$. $m = O(n^2)$ is enough.

5.4 Security Analysis of Concrete Self-bilinear Maps

Modified Encoding/Decoding Attack Hu et al. provided the modified encoding/decoding to solve the MDDHP [17]. In fact, their algorithm analyzed the GGHLite scheme [18]. There is a little difference between the GGHLite and GGH scheme in the re-randomization procedure. GGLite only contains two re-randomization parameters $x^{(1)}$ and $x^{(2)}$. These parameters are also the level-1 encodings of I. The modified encoding/decoding algorithm almost totally solves the MDDHP. If we use κ-MDDHP to denote the problem corresponding to κ-multilinear maps, $c_k^{(d)}$ to denote the level-k encoding of $I + d$, $\{c_1^{(d_i)}\}_{i=1}^{\kappa+1}$ is the $\kappa+1$ level-1 encodings given by the κ-MDDHP oracle, then the attack procedure works as follows.

1. Use the weak-DL attack to generate the level-0 encoding d_i' of level-1 encoding $c_1^{(d_i)}$. Note that d' is not a short element.
2. Multiply these level-0 encodings together to get the level-0 encoding $\prod_{i=1}^{\kappa+1} d_i$.
3. Use the modified encoding/decoding procedure to get a level-κ decoding $p_{zt} c_\kappa^{(\prod_{i=1}^{\kappa+1} d_i)}$.
4. Extract the high order bits of the result in the step 3.

If level-k encodings encode the same coset of I, the most significant bits of their decodings are identical. So, these bits can help adversaries distinguish the $c_\kappa^{(\prod_{i=1}^{\kappa+1} d_i)}$ and a random level-κ encoding.

The attacking algorithm requires some intermediate parameters. These parameters are called special decodings that are obtained as below.

$$Y = y^{\kappa-1} x^{(1)} p_{zt} \pmod{q} = h(1 + ag)^{\kappa-1} b^{(1)}$$

$$X^{(i)} = y^{\kappa-2}x^{(i)}x^{(1)}p_{zt} \pmod{q} = h(1+ag)^{\kappa-2}(b^{(i)}g)b^{(1)},\ i = 1, 2$$

where $x^{(i)} = [b^{(i)}g/z]$, $i = 1, 2$. y=$(1 + ag)/z$. The exponent of y brings a limitation to this procedure. If $0 \leq \kappa \leq 2$, $\kappa - 1$ or $\kappa - 2$ will be smaller than 0. On one hand, since some elements in the ring R_q are not invertible, $y^{\kappa-2}$ can not always be computed. On the other hand, if $y^{2-\kappa}$ is invertible in R_q, the invert operations can't assure that the coefficient of $y^{\kappa-2}$ is smaller than q. The "mod q" operation couldn't be omitted on the right sides of the equations above. So, the attacking procedure can only solve the κ-MDDHP, for $\kappa \geq 3$.

Our self-bilinear map scheme adopts the 1-GES. The parameter $\kappa = 1$, which means "Modified Encoding/Decoding Attack" doesn't threat our self-bilinear map.

6 Further Consideration to ED

The EDP is a very important hard problem in our scheme. Unfortunately, we don't know how to give a reduction from classic hard problem to EDP. We attempted an extensive cryptanalysis of EDP instead.

6.1 Co-prime in Ring and its Residue Class

In the residue class ring $\mathbb{Z}_p$, the element $\bar{a}$ is invertible if and only if a is co-prime with p in $\mathbb{Z}$. The modified Euclidean Algorithm can be used to check whether two elements in $\mathbb{Z}$ are co-prime. Moreover the modified Euclidean Algorithm will output $\bar{a}^{-1}$ if $\bar{a}$.

In this section. we notice that if we have an algorithm to check the co-prime relation in R, all the co-prime relation in its residue class can be check. Moreover, the inversion of the element in R of its residue class ring can be computed by solving a equation with multiple variables.

At the first glance, we have an algorithm to check the co-prime relation only if we had the Euclidean algorithm in this ring. Unfortunately, even though we can check co-prime relation in $\mathbb{Z}[x]$ and $\mathbb{Q}[x]$, these case in residue class of them are more complicated so that we can't do it in directly. But this topic is quite important since if we want to calculate the inverse of $\bar{a} \in R/\langle g\rangle$, we usually find $s, t \in R$ such that $sa + tg = 1$, and pick this (s *mod* g) as the inverse of $\bar{a} \in R/\langle g\rangle$.

We obtained a simply and directly conclusion by exploring the procedure which find inverse of elements in $\mathbb{Z}_q$. That is if calculating inverse in $\mathbb{Z}_q$, one take use of Euclidean algorithms in Z. So we have conclusion below.

Theorem 3. *Let $R/\langle g\rangle$ be the residue class ring of R, where $g \in R$. Then $\bar{a}, \bar{b} \in R/\langle g\rangle$ are co-prime if and only if $\exists s, t, p \in R$ such that $\forall a + lg \in \bar{a}$ and $\forall b + ng \in \bar{b}$, $s(a + kg) + t(b + ng) + pg = 1$.*

Proof. Assume that $\oplus$ and $\odot$ denote the add and multiply in $R/\langle g\rangle$ respectively.

1. **Sufficiency.** if $\bar{a}$, $\bar{b} \in R/\langle g\rangle$ are co-prime, then $\exists \bar{s}$, $\bar{t} \in R/\langle g\rangle$ such that

$$\bar{s} \odot \bar{a} \oplus \bar{t} \odot \bar{b} = \bar{1}$$

$$(s + kg) \cdot (a + lg) + (t + mg) \cdot (b + ng) = 1 + pg$$
$$(s + kg) \cdot (a + lg) + (t + mg) \cdot (b + ng) - pg = 1$$

where $k, l, m, n \in R$. So $a + lg$ and $b + ng$ can denote any element in $\bar{a}$ and $\bar{b}$ respectively, and $\exists(s + kg), (t + mg), (-p) \in R$ to prove the result.

2. **necessity.** if $\exists s, t, p \in R$ such that $s(a + lg) + t(b + ng) + pg = 1$. we have

$$sa + tb \equiv 1 \ mod \ g$$
$$\bar{s} \odot \bar{a} \oplus \bar{t} \odot \bar{b} = \bar{1}$$

this means $\bar{s}$, $\bar{b} \in R/\langle g \rangle$ are co-prime by the definition.

By using this theorem, if we want to compute inversion of $\bar{a} \in R/\langle g \rangle$, we simply use the representation of coset $a + \langle g \rangle$ and find $s, t, p \in \mathbb{Z}[x]$ such that $sa + tg + pf = 1$, then s is the a representation of coset $a^{-1} + \langle g \rangle$. But this is a function with three variants. We wonder how hard it is, especially we limited the norm of s.

6.2 Direct Attack

We discuss the difficulty of EDP in concrete OWES. The EDP in OWES is denoted as $(a, v, c^{(ab)} = abv + rX)$. Assume that $a \in A$, $b \in B$, A, B are elements in R/I (here, we don't use the coset form to denote them). All elements in R/I is invertible, because that $I = \langle g \rangle$ is the prime element in R. Since a is public, the adversary have two direct idea to solve EDP to get $c^{(b)}$.

1. Divide $c^{(ab)}$ by a using the division algorithm in R.
2. Divide $c^{(ab)}$ by a using the division algorithm in $R_q[x]$.
3. Find short enough $a' \in A^{-1}$, and compute $c' = [a'c^{(ab)}]_q$. c' is a valid level-1 encoding of element in B.

Case 1. All these elements are regarded as elements in R, or as polynomials with degree less than n. We will get

$$\frac{abv + rX}{a} = bv + \frac{rX}{a}$$

rX is an element in I. It can be written as a polynomial $rX = k(x)g(x) + l(x)f(x)$. Since $a(x)$ ia random polynomial with degree smaller than n, $a(x) \nmid g(x)$ and $a(x) \nmid f(x)$ with high probability. Moreover, for n is a power of 2, $f(x) = x^n + 1$ is an irreducible polynomial in $Q[X]$, so is in $Z[X]$ and R. Thus, it doesn't exist $k'(x)$ and $l'(x)$ that makes $\frac{rX}{a} = k'(x)g(x) + l'(x)f(x)$ with high probability. $\frac{rX}{a} \notin I$. $\frac{abv+rX}{a} = bv + \frac{rX}{a}$ is not a valid level-1 encoding of B.

Case 2. R_q is not a Euclidean ring. There's no known division algorithm in R_q. If we want to do the division operation, we have to take use of the relation ship between R_q and R (Actually, R is not a Euclidean ring, But R is more closer to $Z[X]$ than R_q, and $\mathbb{Z}[X]$ is Euclidean ring). So, the analysis is similar to case 1. However, R_q cause mores problems than R. the modular q operation rise the coefficients up, It's even not a valid level-1 encoding sometimes.

Case 3. If the short $a' \in A^{-1}$ is found, attack method 3 truly can solve EDP. We try to measure the difficulty of finding a'.

We use f to denote the polynomial $f(x)$ for simplicity. The element in R can be written as $p + kf$, where $p, k, f \in \mathbb{Z}[x]$. The element in R/I can be written as $\bar{p} + \bar{r}\bar{g}$, where $\bar{q}, \bar{r}, \bar{g} \in R$. It can also be written as

$$\begin{aligned} & (p + kf) + (r + k'f)(g + k''f) \\ = \; & p + rg + (k'g + k''f + rk'')f \\ = \; & p + rg + r'f \end{aligned} \tag{1}$$

all elements in (1) are in $\mathbb{Z}[X]$. $r' = k'g + k''f + rk''$. This fact tell us, the element $\bar{\bar{p}}$ in R/I can be written as $p + rg + r'f$, and $p \in Z[X]$ is the representation of $\bar{\bar{p}}$.

Thus, to find an element $a' \in A^{(-1)}$, is to find polynomials $a', s, t \in Z[X]$ that satisfy the equation below

$$a'a + sg + tf = 1 \tag{2}$$

where f is a public parameter, g is a secret parameter, but GGH state that a not short representation $g' \in \langle g \rangle >$ could be recovered. Equation (2) with three variables. It seems hard to find a', s, t that satisfy equation (2), and a' is a short polynomial. Since s, t don't have to be short, adversaries can fix a short a' and find any s, t that solve the equation. Since a' is a random element, the probability $Pr[a' \in A^{(-1)}] = |R/I|^{-1}$. $|R/I|$ can be computed as $|R/I| = \det(g, gx, \cdots, gx^{n-1})$. We do believe that $|R/I| = O(2^{\lambda})$, because that the $|R/I|$ is the plaintext space of GGH graded encoding system. If $|R/I|$ is not large enough, the MDL problem in GGH is not hard any more. Thus, the probability that the short $a' \in A^{-1}$ is negligible.

7 Conclusion

We described a new notion called one way encoding system (OWES). By making use of the indistinguishability obfuscation, we construct a self-bilinear map over the OWES. The EBCDHP is proved to be hard if the EDP is hard. We also discussed that the graded encoding system like GGH can be used as an instance of OWES. After that, a concrete construction from GGH encoding system is proposed. To increase the confidence of security, we give a simple analysis about EDP in the polynomial ring and its residue class ring. We believe that the EDP in GGH is as hard as we need.

References

1. Albrecht, M.R., Farshim, P., Hofheinz, D., Larraia, E., Paterson, K.G.: Multilinear maps from obfuscation. Tech. rep., Cryptology ePrint Archive, Report 2015/780, 2015. http://eprint.iacr.org (2015)
2. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Advances in Cryptology–CRYPTO 2001. pp. 213–229. Springer (2001)

3. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: Advances in CryptologyASIACRYPT 2001, pp. 514–532. Springer (2001)
4. Boneh, D., Silverberg, A.: Applications of multilinear forms to cryptography. Contemporary Mathematics 324(1), 71–90 (2003)
5. Boneh, D., Wu, D.J., Zimmerman, J.: Immunizing multilinear maps against zeroizing attacks. Tech. rep., Cryptology ePrint Archive, Report 2014/930, 2014. http://eprint.iacr.org (2014)
6. Cheon, J.H., Han, K., Lee, C., Ryu, H., Stehlé, D.: Cryptanalysis of the multilinear map over the integers. In: Advances in Cryptology–EUROCRYPT 2015, pp. 3–12. Springer (2015)
7. Cheon, J.H., Lee, D.H.: A note on self-bilinear maps. Korean Mathematical Society 46(2), 303–309 (2009)
8. Coron, J.S., Lepoint, T., Tibouchi, M.: New multilinear maps over the integers. Tech. rep., Cryptology ePrint Archive, Report 2015/162, 2015. http://eprint.iacr.org (2015)
9. Coron, J.S., Lepoint, T., Tibouchi, M.: Practical multilinear maps over the integers. In: Advances in Cryptology–CRYPTO 2013, pp. 476–493. Springer (2013)
10. Coron, J.S., Lepoint, T., Tibouchi, M.: Cryptanalysis of two candidate fixes of multilinear maps over the integers. Tech. rep., Cryptology ePrint Archive, Report 2014/975, 2014. http://eprint.iacr.org (2014)
11. Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices. In: Advances in Cryptology–EUROCRYPT 2013. vol. 7881, pp. 1–17. Springer (2013)
12. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on. pp. 40–49. IEEE (2013)
13. Garg, S., Gentry, C., Halevi, S., Zhandry, M.: Fully secure functional encryption without obfuscation. Tech. rep., Cryptology ePrint Archive, Report 2014/666, 2014. http://eprint.iacr.org (2014)
14. Garg, S., Gentry, C., Sahai, A., Waters, B.: Witness encryption and its applications. In: Proceedings of the forty-fifth annual ACM symposium on Theory of computing. pp. 467–476. ACM (2013)
15. Gentry, C., Gorbunov, S., Halevi, S.: Graph-induced multilinear maps from lattices. In: Theory of Cryptography, pp. 498–527. Springer (2015)
16. Groth, J., Ostrovsky, R., Sahai, A.: Perfect non-interactive zero knowledge for np. In: Advances in Cryptology–EUROCRYPT 2006, pp. 339–358. Springer (2006)
17. Hu, Y., Jia, H.: Cryptanalysis of ggh map. Tech. rep., Cryptology ePrint Archive, Report 2015/301, 2015. http://eprint.iacr.org (2015)
18. Langlois, A., Stehlé, D., Steinfeld, R.: Gghlite: More efficient multilinear maps from ideal lattices. In: Advances in Cryptology–EUROCRYPT 2014, pp. 239–256. Springer (2014)
19. Lee, H.S.: A self-pairing map and its applications to cryptography. Applied Mathematics and Computation 151(3), 671C678 (2004)
20. Paneth, O., Sahai, A.: On the equivalence of obfuscation and multilinear maps. Submission to TCC 2015 (2015)
21. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Advances in Cryptology–EUROCRYPT 2005, pp. 457–473. Springer (2005)
22. Yamakawa, T., Yamada, S., Hanaoka, G., Kunihiro, N.: Self-bilinear map on unknown order groups from indistinguishability obfuscation and its applications. In: Advances in Cryptology–CRYPTO 2014, pp. 90–107. Springer (2014)

23. Zhang, F., Safavi-Naini, R., Susilo, W.: An efficient signature scheme from bilinear pairings and its applications. In: Public Key Cryptography–PKC 2004, pp. 277–290. Springer (2004)