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

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

[1] School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China;
[2] Guangdong Key Laboratory of Information Security Technology
[3] School of Information Management, Sun Yat-sen University, Guangzhou 510006, China;

Abstract. The bilinear map whose domain and range are identical is called self-bilinear map. Once such kind of bilinear map exists, the multilinear map can be constructed easily by using self bilinear map as a component. Yamakawa et al. have introduced the first secure self-bilinear map with auxiliary information based on the integer factoring assumption in Crypto 2014. Inspired by their work, we find that any encoding system with particular properties could be used to build self-bilinear map. We generalize them as one way encoding system and propose a generic construction of self-bilinear map. For cryptographic use, we define a new encoding division assumption to make the analog DDHP hard. We show that one level encoding of graded encoding system which is used to build multilinear map nowadays satisfy all the properties of one way encoding system. We also present an instance that is build on GGH graded encoding scheme and analyze how hard the encoding division problem is. Our self-bilinear map is believed to be quantum resistance. It seems more secure than the scheme of Yamakawa et al. Moreover, the encoding size of n-multilinear built on our self-bilinear map is smaller than that of GGH scheme.

1 Introduction

Bilinear maps were first used to solve of discrete logarithm problem for some weak elliptic curves [9, 20]. Then, these elliptic curves show powerful capability as a cryptographic primitive. It provides solutions for many cryptographic application such as identity-based encryption [1], non-interactive zero-knowledge proof systems [15], attribute-based encryption [21] and short signature, etc. A special instance of bilinear map of which the target group and preimage group are the same is called self-bilinear map. Because of the exclusive property, self-bilinear map may apply to more scenes. One direct application of self-bilinear map which is not impled by the ordinary one is to construct multilinear maps.

Lee designed the first candidate self-bilinear map [19]. After his work, Cheon and Lee remark that the self-bilinear map above is not essentially self-bilinear [5]. They also proved the impossibility result that there isn't any secure self-bilinear maps in known prime order cyclic group G. The computational Diffie-Hellman (CDH) assumption in G collapses because the map e reveals much information about group G. Excluding the known prime order cyclic group, two natural ideas to build self-bilinear map emerge. The first one takes use of the unknown composite order cyclic groups. The scheme of Yamakawa et al. [22] belongs to this classification. They choose to take use of the signed quadratic residue group $\mathbb{QR}_n^+$ of $\mathbb{Z}_n^*$ [16]. The integer factoring assumption guarantees the hardness of DDHP in G. Moreover there is a polynomial reduction from factoring problem to BCDH problem with respect to self-bilinear map. The second idea is inspired by the scheme of multilinear map nowadays, that is to build self-bilinear map from some special set rather than cyclic groups. This is the start point of our work.

Multilinear maps are generic notion from bilinear maps. Not long after the-bilinear map shows the convenience it brings to the cryptography, Boneh and Silverg [2] first image the existence of multilinear maps and their abundant applications. But, when they tried to forge such fantastic tool, they met serious obstacles. From then on, constructing multilinear map became a long-standing open problem. Until recently, three candidate multilinear maps were proposed – the GGH scheme [10] on ideal lattices, the CLT scheme [7] over the integer and the GGH14 [14] on lattices. The multilinear map is a basic component of some new cryptographic primitives such as witness encryption [13], indistinguishability obfuscation and functional encryption [11].

Multilinear maps met extremely strong challenges in 2014 and 2015. Not long after the CLT scheme was completely broken by "zerozing algorithm" [4], GGH scheme was under attack. Hu uses weak-DL attack and modified encoding/decoding algorithm that they designed in [17] to break the GMDDH assumption which is the security basis of many multilinear map applications. Two candidate fixes of multilinear maps over integers [12, 3] been proposed soon after the CLT broken. But Coron et al [8] state that these two patchs are still unsafe. They describe a new scheme over the integers [6] which is believed secure. To the ideal lattice side, there isn't any accepted scheme proposed. To construct a secure and efficient multilinear map is still a worthwhile work.

Multilinear map can be constructed trivially if the self-bilinear map exists. Say, if $e : G \times G \to G$ is a self-bilinear map, one can combine sufficient many self-bilinear maps to build a multilinear map among these groups. This is owe to the special property that the target group of self-bilinear map e is also one of the domain group of e. Thus, one of the constructing method of multilinear map can be reduced to find a good self-bilinear map.

To construct self-bilinear map which is not in cyclic group, we studied from the cyclic group G. Treat $g^\alpha \in G$ as an encoding of $\alpha \in \mathbb{Z}_{ord(G)}$. Thus, cyclic is an encoding scheme from ring $\mathbb{Z}_{ord(G)}$ to group G. $f : \mathbb{Z}_{ord(G)} \to G$, $f(x) = xg \ mod \ ord(G)$ is an efficient encoding algorithm. Note that xg is an operation between ring element and group element. $x_1 g + x_2 g = (x_1 + x_2)g \ mod \ ord(G)$ remind us that the analogous distributed law holds in encoding scheme. Following the unconventional explanation from group to encoding system, we define a more abstract edition called one way encoding system (OWES). OWES removes the restriction to domain ring and image set (in cyclic group case, $\mathbb{Z}_{ord(G)}$ is also a cyclic group with respect to addition operation, G is a cyclic group). It defines the map $f : R \rightarrow S$, where R is a finite ring with identity and S is a set with additive operation. This leads to some essential differences to cyclic group. Since the plaintext space (ring) R of OWES is not an additive cyclic group like $\mathbb{Z}_{ord(G)}$, the scheme needs a new efficient manipulation that multiply a plaintext (in R) and encoding to get a new encoding ($\alpha \in R$, $g \in S$ imply $\alpha g \in S$). This operation is quite like the scalar multiplication of vector in linear spaces. We stress that multiplying operation in S is not necessary. Thus, the sum of encoding of $\alpha \in R$ and encoding of $\beta \in R$ is the encoding of $\alpha + \beta$, the multiplication of encoding of $\alpha \in R$ multiply plaintext $\beta \in R$ is the encoding of $\alpha\beta$. Once the efficient operations are defined, we can easily simulate the DLP and CDHP in the encoding scheme. That is, DLP: given $\alpha g, g$ to compute α, CDHP: given $\alpha g, \beta g, g$ to compute $\alpha\beta g$.

Now we have the encoding system with some necessary hardness assumption. The self-bilinear map is fixed as $e(\alpha g, \beta g) = r\alpha\beta g$, where $r \in R$ is an element in R. Since the map e have to be non-degenerate, the size of $\{xg|x \in R, g \in S\}$ should equal to that of $\{rxg|x \in R, g \in S\}$. This requirement forces r not be a zero divisor of R. Since an element in finite ring with identity is either a unit or a zero divisor, it is equivalent to choose r as an invertible elements in R at random. If such a self-bilinear map is built up, we are faced with a CDH assumption challenge that similar to that in known prime order group. Namely, r is a public parameter. Given αg and βg, $r\alpha\beta g$ is efficient computable. If adversary can get $r^{-1} \in R$ efficiently or divide r from $r\alpha\beta g$, the CDHP assumption in S is broken. So we propose a new hardness assumption to stop this from happening. Moreover, Such an assumption is justifiable. We will show that GGH graded encoding scheme [10] could achieve all the property we discussed above.

Firstly, we define a generic encoding system which is called one way encoding system (OWES) and prove that any encoding schemes contains its property can build a secure self-bilinear map. Secondly, we take GGH graded encoding system as example to build a secure self-bilinear map. GGH scheme is based on ideal lattices, the security of our self-bilinear map is believed to be quantum resistance. Finally, we give some security analysis to the instance of one way encoding system which is one level of GGH encoding scheme. Since the new hardness assumption can not reduce to classical problem, we try to enhance the confidence that they are indeed hard. The attack of Hu et al. [17] which breaks the MDDH assumption in GGH scheme does not threaten our scheme.

2 Preliminaries

2.1 Notations

We use $\mathbb{Z}$ to denote the set of all integer numbers, $\mathbb{Q}$ to denote set of all rational numbers, $\mathbb{F}$ to denote number 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 $\{x_1, \cdots, x_n\}$.

$R = \mathbb{Z}[x]/(x^n + 1)$ is the residue class ring mentioned in the GGH scheme. We use $I = \langle g \rangle$ to denote the principal ideal of R, where $g \in R$. Let R/I be the palintext space of GGH map. q is a large number. All operation of GGH map is run over the ring $R_q = R/qR = \mathbb{Z}_q[x]/(x^n + 1)$, and we use $[\cdot]_q$ to denote the operation in R_q. The re-randomizer parameter are $x_i = [\frac{b_i}{z}]_q$ with $b_i \in I$. We combine x_i as a row vector of matrix X, and use r to denote a row vector sampled in D_{Z^n,σ^*}, where σ^* is a large enough standard deviation. The "generator of group" is $y = [\frac{a}{z} + rX]_q$, where a is in $1 + I$. P_{zt} is the zero-testing parameter.

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 all 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 (non necessarily uniform)* PPT *distinguisher D, there exists a negligible function α such that the following holds: For all security parameters $\lambda \in N$, for 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.

2.3 One Way Encoding System

Since we want to construct self-bilinear map over some encoding system, we have to define corresponding manipulations to imitate manipulation in group. We regard the representation of an element xg in cyclic group as an encoding of a ring element $x \in \mathbb{Z}_n$, where n is the order of the cyclic group. Thus, the plaintext space is a ring and the encoding space is a set with addition operation. On the other hand, the DL assumption is a basic requirement for group to be used in cryptography. This leads us to define a new notion of one way encoding system (OWES).

Assume that S_0 is a finite ring with identity, S_1 is an additive group, f is a surjection $f : S_0 \rightarrow S_1$.

Definition 2 (OWES). *An encoding system (S_0, S_1, f) is called one way encoding system, if it has the following properties:*

- *f is additive homomorphism: for $a, b \in S_0$, $f(a+b) = f(a) \oplus f(b)$.*
- *Palintext multiplication: for $a \in S_0$ and $f(b) \in S_1$, $a \otimes f(b) = f(ab)$.*
- *f is computational one way function: Given $f(a)$, it's hard to compute a.*

Given an OWES (S_0, S_1, f) and an element $b \in S_0$, $(S_0, b \otimes S_1, b \otimes f)$ is also an OWES.

For cryptographic use, the analogue of CDHP should hold in OWES. To achieve it, we introduce two new hardness assumption to assure. The first one is encoding computational Diffie-Hellman assumption (ECDHA).

Definition 3 (ECDHA). *For* OWES (S_0, S_1, f)*, the Encoding* CDH *problem is, on input $f(a), f(b) \in S_1$, to compute $f(ab) \in S_1$. The assumption state that there isn't any efficient algorithm to solve this problem with non-negligible probability.*

The self-bilinear map reveals extra information about CDHP. Adversary may take use of these extra information to solve CDHP. For this reason, the encoding division assumption (EDA) should hold in OWES.

Definition 4 (EDA). *For* OWES (S_0, S_1, f)*, the Encoding Division problem is, on input the $f(ab) \in S_1$ and $a \in S_0$, where a is invertible in S_0, to compute $f(b) \in S_1$. The assumption state that there isn't any efficient algorithm solve this problem with non-negligible probability.*

We state that any level of GGH graded encoding system is OWES. We can also believe that OWES is a generalization of GGH graded encoding system. Moreover, the graded encoding system which is used to build multilinear maps are likely instances of OWES.

3 Self-bilinear Map and Multilinear Map

In this section, we describe the definition of self-bilinear map and multilinear map. Then, a generic construction of self-bilinear (multilinear) map is given. After that, we propose a concrete scheme from GGH graded encoding system.

3.1 Self-bilinear Map

Cheon and Lee [5] introduced a special kind of bilinear map. The preimage group and target group are the same. They called it self-bilinear map. The self-bilinear map constructed in our work is not in groups, we give the formal definition below.

Definition 5 (Self-bilinear Map). *For an OWES (S_0, S_1, f), $e : S_1 \times S_1 \to S_1$ is called self-bilinear map, if the following properties hold.*

- *for all $g_1, g_2 \in S_1$ and $\alpha \in S_0$, it holds that*

$$e(\alpha \otimes g_1, g_2) = e(g_1, \alpha \otimes g_2) = \alpha \otimes e(g_1, g_2)$$

- *The map e is non-degenerate. Namely, e is surjective.*

3.2 Hardness Assumption

For cryptographic use, the hard problem such as discrete logarithm (DL) and computational Deffie-Hellman (CDH) Problem should keep computational unsolvable in single group.

Definition 6 (DL). *The Discrete-Log problem is hard for a bilinear map scheme, if for all probabilistic polynomial time algorithms, the discrete-logarithm advantage of $\mathcal{A}$,*

$$Adv_{\mathcal{A}}^{Dlog}(\lambda) = Pr[\mathcal{A}(g, \alpha \otimes g) = \alpha | g \in S_1, \alpha \in S_0]$$

is negligible in λ.

The Computational Diffie-Hellman assumption (CDH) should hold in the single set of self-bilinear map.

Definition 7 (CDH). *For a self-bilinear map scheme, the computational Diffie-Hellman problem is hard, if for every probabilistic polynomial time algorithms $\mathcal{A}$, the* CDH *advantage of $\mathcal{A}$,*

$$Adv_{\mathcal{A}}^{\mathrm{CDH}}(\lambda) = Pr[\mathcal{A}(g, \alpha_0 \otimes g, \alpha_1 \otimes g) = (\alpha_0\alpha_1) \otimes g | \alpha_0, \alpha_1 \in S_0, g \in S_1]$$

is negligible in λ.

Bilinear computational Diffie-Hellman assumption (BDDH) is the security basis of 3 party Diffie-Hellman key exchange scheme.

Definition 8 (BCDH). *For a self-bilinear map scheme, the computational bilinear Diffie-Hellman problem is hard, if for every probabilistic polynomial time algorithms $\mathcal{A}$, the* BCDH *advantage of $\mathcal{A}$,*

$$Adv_{\mathcal{A}}^{\mathrm{BCDH}}(\lambda) = Pr[\mathcal{A}(g, \alpha_0 \otimes g, \alpha_1 \otimes g, \alpha_2 \otimes g) = \alpha_2 \otimes e(\alpha_0 \otimes g, \alpha_1 \otimes g) | \alpha_0, \alpha_1 \in S_0, g \in S_1]$$

is negligible in λ

The BCDHP can be generalized as multilinear computational Diffie-Hellman problem which is an important hard problem in multilinear maps.

3.3 Multilinear Map

Boneh et al [2]. first give the formal definition of multilinear maps. Their multilinear map was built on cyclic groups. The multilinear map scheme nowadays don't satisfy their definition. For this reason, we describe a new definition that doesn't base on cyclic groups.

Definition 9 (Multilinear Map). *For* OWES (S_0, S_1, f)*, a map $e : S_1 \times \cdots \times S_1 \to S_1$ is called n-multilinear if it has the following properties:*

1. *For elements $\{g_i \in S_1\}_{i=1}^n$, $\alpha \in S_0$ it holds that*

$$e(g_1, \cdots, \alpha \otimes g_i, \cdots, g_n) = \alpha \otimes e(g_1, \cdots, g_n).$$

2. *The map e is non-degenerate: if $S_1 = \{\alpha \otimes g_i | \alpha \in S_0\}$, for $i = [n]$, then $S_1 = \{\alpha \otimes e(g_1, \cdots, g_n) | \alpha \in S_0\}$.*

n-mutilinear maps have symmetric case and asymmetric case. Self-bilinear map can only be used for designing the symmetric one (The domain set are all the same). The relationship between self-bilinear map and multilinear map in symmetric case can be inducted iteratively as follows

$$e_{n+1}(g_1, \ldots, g_n, g_{n+1}) := e(e_n(g_1, \ldots, g_n), g_{n+1}).$$

where e with the subscript i denote the $i-$multilinear map consists of self-bilinear map. g_i denote the element in S_1.

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

In this section, we construct a self-bilinear map $\mathcal{SBP}$ from OWES and $i\mathcal{O}$. Then, we use $\mathcal{SBP}$ to build a multilinear map scheme $\mathcal{MMP}$. We also trying to prove that the necessary hardness assumption like DL, BCDH, CDH assumption etc hold in our self-bilinear map and multilinear map scheme.

4.1 Our Construction

We describe some notations for circuits on OWES, since the scheme take $i\mathcal{O}$ as a basic component.

Notation for Circuits on OWES. For the OWES and $a_1 \in S_0$, C_{a_1} denotes the circuit that work as follows. If the input $f(a_2) \in S_1$, $C_{a_1}(f(a_2))$ computes the value of $f(a_1a_2) \in S_1$. If $f(a_2) \notin S_1$, then outputs $\perp$. For circuits C_1, C_2 whose output elements are in S_1, Plus(C_1, C_2) denotes a circuit that computes $C_{add}(C_1(x), C_2(y))$ for input (x, y) where C_{add} is a circuit that computes a addition for elements of S_1. If an input of C_{add} is not a pair of two elements in S_1, then it outputs $\perp$.

We use $\mathcal{SBP}$ to denote self-bilinear map that we construct. $\mathcal{SBP}$ contains procedures below.

params $\leftarrow$ **InstGen**$(\mathbf{1}^\lambda)$.

- On input the security parameter λ, initiate an OWES (S_0, S_1, f).
- Chooses a invertible element $r \in S_0$ at random.
- Outputs **params** $= (S_0, S_1, f, r)$ system parameters.

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

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

$(f(a), \tau_{f(a)}) \leftarrow \mathbf{Enc}(\mathbf{params}, \mathbf{a})$.

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

$f(ra_1a_2) \leftarrow \mathbf{Map}(\mathbf{params}, f(a_1), \tau_{f(a_2)})$. On input $f(a_1)$, runs the obfuscated circuit $\tau_{f(a_2)}$ to compute $\tau_{f(a_2)}(f(a_1)) = f(ra_1a_2)$.

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

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

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

As we mentioned in section 3.2, for cryptographic use, the DL, CDH, BCDH assumption should hold in $\mathcal{SBP}$. The DL assumption is directly following the one way property of OWES. THE ECD assumption and the ED assumption in OWES assure that the CDH problem is hard in $\mathcal{SBP}$. We prove that the BCHD problem is hard in $\mathcal{SBP}$ below.

Theorem 1. *If there is a PPT algorithm $\mathcal{A}$ solving BCDH problem in $\mathcal{SBP}$ efficiently, then there is an algorithm to solve ED problem in* OWES *efficiently.*

Proof. First, we introduce two games [22] played between adversary $\mathcal{A}$ and challenge $\mathcal{C}$, where $\mathcal{C}$ simulates the $\mathcal{SBP}$ scheme. We want to prove that $\mathcal{A}$ wins these games with the same advantage.

Game 1. $\mathcal{C}$ simulates the original $\mathcal{SBP}$ scheme in this game.

1. $\mathcal{C}$ takes λ and initiates an OWES (S_0, S_1, f).
   $\mathcal{C}$ chooses an invertible $r \in S_0$ and sets $\mathbf{params} = (S_0, S_1, f, r)$
2. $\mathcal{C}$ chooses $a_0, \cdots, a_n \overset{\$}{\leftarrow} S_0$ uniformly at random.
3. $\mathcal{C}$ computes $f(a_i)$ and generates its corresponding auxiliary information $\tau_{f(a_i)} = i\mathcal{O}(C_{ra_i})$, for $i \in [0, n]$.
4. $\mathcal{C}$ sends **params**, $\{f(a_i)\}_{i=0}^n$, $\{\tau_{f(a_i)}\}_{i=0}^n$ to $\mathcal{A}$. $\mathcal{A}$ takes these parameters as input, then returns U.

Game 2. Game 2 is similar to Game 1 except that parameters are chosen differently.

1. $\mathcal{C}$ takes λ and initiates an OWES (S'_0, S'_1, f').
   $\mathcal{C}$ chooses a random element $b \in S_0$ and an invertible $r \in S_0$. Then, computes rb encodes it as $f'(rb)$.
   $\mathcal{C}$ computes $S_0 = S'_0$, $f : S_0 \to S_1$, $f(x) = x \otimes f'(rb)$, $S_1 = \{f(a) | a \in S_0\}$.
   Set $\mathbf{params} = (S_0, S_1, f, r)$.

2. $\mathcal{C}$ chooses $a'_0, a'_1, a'_2 \overset{\$}{\leftarrow} S_0$. Let $ra_i = ra'_i + 1$, $i = 0, 1, 2$. (note that r is invertible in S_0, this equation is possible, and $a_i = a'_i + r^{-1}$).
3. Makes $f(ra_i) = (ra'_i + 1) \otimes f'(rb)$, $i = 0, 1, 2$.
   $\mathcal{C}$ generates the auxiliary information corresponding to $f(a_i)$, $\tau_{f(a_i)} = i\mathcal{O}(C_{ra_i}) = i\mathcal{O}(C_{ra'_i+1})$, for $i = 0, 1, 2$.
4. $\mathcal{C}$ sends $\{f(a_i), \tau_f(a_i)\}_{i=0}^2$, **params** to $\mathcal{A}$. $\mathcal{A}$ takes these parameters as input, then outputs U.

If $U = f(ra_0a_1a_2)$, we say that $\mathcal{A}$ wins games. Use G_1 and G_2 to denote the event that $\mathcal{A}$ wins game 1 and game 2 respectively. In essentially, game 1 and game 2 is the same because of the distribution of their parameters are computationally indistinguishable. In other word, the view of $\mathcal{A}$ in game 1 and 2 are the same. We analysis the differences part of game 1 and 2 below.

Step 1 simulates the instance generation step of $\mathcal{SBP}$ scheme. The instance generation step accomplish two things in essentially. First, initiate an OWES. Secondly, choose an invertible element in S_0 at random. Step 1 of game 2 is an extend to that of game 1. If (S_0, S_1, f) is an OWES, $(S_0, b \otimes S_1, b \otimes f)$ also is. Thus, Game 2 simulate the instance generation step perfectly in the view of $\mathcal{A}$.

Step 2 samples elements uniformly at random in S_0. Since r is an invertible element in S_0 and a_i obey the uniformly distribution in S_0, $a_i = a'_i + r^{-1}$ also obey the uniformly distribution in S_0. We state that game 2 simulates the sample operation perfectly in $\mathcal{SBP}$.

Step 3 was executed by the rule of original scheme, the difference between Game 1 and Game 2 is the parameters generated in step 1 and step 2.

Depending on all these facts analyzed above, $\mathcal{A}$ wins game 1 and game 2 with similar probability. Namely, $|Pr[G_1] - Pr[G_2]| = negl(\lambda)$.

We give a polynomial reduction from EDP to BCDHP by introducing an algorithm $\mathcal{B}$.

Algorithm $\mathcal{B}$: $\mathcal{B}$ takes $(f'(rb), r)$ and the OWES (S'_0, S'_1, f') as input.

1. Computes $f'(r^2b) = r \otimes f'(rb)$. Let $S_0 = S'_0$, $f : S_0 \to S_1$, $f(x) = x \otimes f'(r^2b)$, $S_1 = \{f(a) | a \in S_0\}$. Setss **params** $= (S_0, S_1, f, r)$.
2. Samples elements $a'_0, a'_1, a'_2 \overset{\$}{\leftarrow} S_0$.
3. Computes $ra_i = ra'_i + 1$. (note that r is invertible S_0, this equation is possible) Thus, $f(a_i) = (ra'_i + 1) \otimes f'(rb)$
4. Generates the auxiliary information corresponding to $f(a_i), i = 0, 1, 2$ respectively. $\tau_{f(a_i)} = i\mathcal{O}(C_{ra_i}) = i\mathcal{O}(C_{ra'_i+1})$, $i = 0, 1, 2$.
5. Runs $\mathcal{A}(\textbf{params}, f(a_0), f(a_1), f(a_2), \tau_{f(a_0)}, \tau_{f(a_1)}, \tau_{f(a_2)})$ to get U.
6. Computes $p = a'_0(ra'_1 + 1)(ra'_2 + 1)$.
7. Computes $q = \frac{(ra'_1+1)(ra'_2+1)-1}{r} = ra'_1a'_2 + a'_1 + a'_2$.
8. If $U = f(ra_0a_1a_2)$, calculates $U' = U - [p + q] \otimes f'(rb)$ and outputs it. Otherwise, abort.

Correctness of $\mathcal{B}$: We show that the step 7 of $\mathcal{B}$ outputs the right answer for ED problem.

$$\begin{aligned}
U &= f(ra_0a_1a_2) \\
&= f'(ra_0a_1a_2r^2b) \\
&= f'[a_0ra_1ra_2rb] \\
&= f'[(a'_0 + \tfrac{1}{r})(ra'_1 + 1)(ra'_2 + 1)rb] \\
&= f'[(a'_0(ra'_1 + 1)(ra'_2 + 1) + \tfrac{(ra'_1+1)(ra'_i+1)}{r})rb] \\
&= f'[(a'_0(ra'_1 + 1)(ra'_2 + 1) + \tfrac{(ra'_1+1)(ra'_i+1)-1}{r} + \tfrac{1}{r})rb] \\
U' &= U \ominus [p + q] \otimes f'(rb) \\
&= U \ominus [a'_0(ra'_1 + 1)(ra'_2 + 1) + \tfrac{(ra'_1+1)(ra'_2+1)-1}{r}] \otimes f'(rb) \\
&= U \ominus f'[(a'_0(ra'_1 + 1)(ra'_2 + 1) + \tfrac{(ra'_1+1)(ra'_2+1)-1}{r})rb] \\
&= f'[(\tfrac{1}{r})rb] \\
&= f'(b)
\end{aligned}$$

Time complexity: It's easy to see that the whole algorithm $\mathcal{B}$ except step 7 and step 5 has polynomial size operations. Because of the assumption that $\mathcal{A}$ is an efficient algorithm, we know that step 5 also runs in polynomial time. So only step 7 is needed to be analyzed. Writing $\frac{\prod_{i=1}^{n}(ra'_i+1)-1}{r}$ as sum of algebraic expression, it contains $2^n - 1$ terms, each term contain at most n multiply operations. This is a disappointed result because that we have to limited multilinear level n is a logarithmic function of secure parameter to continue our proof. But this is quite conform to the reality. If $n = O(log\lambda)$, Step 7 also runs in polynomial time. As a conclusion of the discussion above, $\mathcal{B}$ is a function that polynomial reduce ED problem to BCDH problem.

Since there is a polynomial reduction from ED problem to n-MCDH problem and we assume that ED assumption is hold, such kind of algorithm $\mathcal{A}$ which solve n-MCDH problem efficiently do not exist.

Remark If there is an efficient division algorithm in S_1 or S_1 have some particular properties, we can use this algorithm to achieve the goal of step 7. This division algorithm will also reduce the time complexity of the algorithm $\mathcal{B}$

4.3 Multilinear Maps

Self-bilinear map $\mathcal{SBP}$ can be easily extended to a multilinear map $\mathcal{MMP}$. For cryptographic use, the hardness assumption such as MCDHA [10] etc should hold. As self-bilinear map with auxiliary information is a weak variant, Yamakawa etc have defined new hardness assumption in [22] which is called multilinear computational Diffie-Hellman with auxiliary information assumption (M-CDHAI).

Definition 10 (MCDHAI assumption [22]). *We say that the* n-MCDHAI *assumption holds with respect to* $\mathcal{MMP}$ *if for any efficient algorithm* $\mathcal{A}$*,*

$$Pr[f(r^{n-1}\prod_{i=0}^{n} x_i) \leftarrow \mathcal{A}(\mathbf{params}, f(x_0), \cdots, f(x_n), \tau_{x_0}, \cdots, \tau_{x_n})] = negl(\lambda).$$

where **params** $\leftarrow$ **InstGen**(1^λ), $x_0, \cdots, x_n \leftarrow$ **Sample**(**params**), $\tau_{x_0}, \cdots, \tau_{x_0} \leftarrow$ **AIGen**(**params**, x_i).

The MCDH assumption is a generalization of BCDH.

Theorem 2. *The n-*MCDHAI *assumption with respect to $\mathcal{MMP}$ holds if the ED assumption corresponding to underlying* OWES *holds.*

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

We have stated that OWES can be initiated. One level of the graded encoding system used in multilinear maps nowadays satisfied all the properties that we required. In this section, we introduce a more concrete self-bilinear map scheme built on GGH graded encoding system [10].

5.1 Construction

An instance of this scheme is run in the polynomial ring $R = \mathbb{Z}[x]/(x^n + 1)$ depending on the GGH scheme [10]. The suggested approximate setting of the basic parameter is $n = \tilde{O}(\kappa\lambda^2)$, $q = 2^{n/\lambda}$, $m = O(n^2)$. Define $R_q = R/qR$.

InstGen(1^λ).

- Generate variant of GGH 1-graded encoding system as follows: choose the security parameter λ large enough, and set $n = \tilde{O}(\kappa\lambda^2)$, $q = 2^{n/\lambda}$, $m = O(n^2)$. Then, the ring $R = \mathbb{Z}[x]/(x^n + 1)$, $R_q = R/qR$. choose an invertible polynomial z in R_q and sample a short polynomial $g \in R$ in the distribution $D_{\mathbb{Z}^n,\sigma}$, where g should satisfied that g^{-1} is short in $\mathbb{Q}[x]/(x^n + 1)$. Use I to denote the ideal generated by g. To generate the re-randomization parameters x_i, $i = [m]$, we simply draw $b_i \leftarrow D_{I,\sigma^*}$, and compute $x_i = [\frac{b_i}{z}]_q$. Sample a from $D_{I+1,\sigma'}$, compute $y = [\frac{a}{z}]_q$. Draw element $h \leftarrow D_{\mathbb{Z}^n,\sqrt{q}}$ and let $P_{zt} = [hz/g]_q$.
- Choose a random element $\alpha \leftarrow D_{\mathbb{Z}^m,\sigma'}$.
- Choose a random element $\bar{s} \in R/I$ by sample $s \leftarrow D_{\mathbb{Z}^m,\sigma'}$, then compute $v = s \cdot y$. After v chosen, the self-bilinear map e is defined as $e(dv + rX, d'v + r'X) = \alpha dd'v + \bar{r}X$.
- Define **params** $= (R/I, R_q, v, \{x_i\}_{i=1}^m, \alpha, P_{zt})$ and makes them public.

Encode(**params**, d).

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

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

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

Map(param, $c^{(d)}, \tau_{c^{(d')}}$). Runs circuit $\tau_{c^{(d')}}(c^{(d)})$ to compute $\alpha d' c^{(d)}$ which is a valid level-1 encoding of $\alpha d d'$.

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

5.2 Setting and Parameters

In GGH graded encoding system, the setting of each parameter is given for keep security and function. The setting of our scheme is based on their conclusion.

GGH are noise encodings, the noise level should never be too large. The valid encoding is a vector of the form $[\frac{c}{z^i}]_q$, where i is the level. The "noise level" of is bounded by the numerator of encodings. The upper bound of $||c||$ is $q^{1/8}$. The adding operation cause less noise growing than multiplying operation, but the concrete differences between them didn't mentioned in GGH's paper. When two elements $[\frac{d_1}{z^i}]_q, [\frac{d_2}{z^i}]_q \in R_q$ executing multiply operation, the size of numerator bounded by $||d_1|| \cdot ||d_2|| \cdot \sqrt{n} = poly(n)$. Adding and multiplying can do any times if the sum and product of encodings is still short, namely $|| \sum_j c_j < q^{1/8}||$, $|| \prod_j c_j < q^{1/8}||$. The public parameter h in P_{zt} is sampled from $D_{\mathbb{Z}^n, \sqrt{q}}$, then g is bounded by $\sqrt{qn}$. This is the key point that make zero testing procedure work. When $P_{zt} = \left[hz^k/g\right]_q$ multiply level k encoding $u = [c/z^k]_q$, zero testing procedure checks the size of $||[h \cdot c/g]_q||$ to judge whether u is encoding zero in R/I or not. If c is a vector in I, then g can divided c in R which means $c \cdot g^{-1} \in K$ and $c/g \in R_q$ are same vector (They are not the same element since they are not in the same ring. But they are the same vector when regard them as a polynomial in R). So we can use $h \cdot c \cdot g^{-1}$ to bounded the size, $||c \cdot g^{-1}|| = ||c|| \cdot ||g^{-1}|| \cdot \sqrt{n} = ||c|| \cdot poly(n)$. then $||h \cdot c/g|| \leq ||h|| \cdot ||c|| \cdot poly(n) = q^{1/2} \cdot q^{1/8} \cdot poly(n) < q^{5/8} < q^{3/4}$.

The setting of parameters should satisfy the basic GGH requirements.

- To draw ideal generator $g \leftarrow D_{\mathbb{Z}^n, \sigma}$, set $\sigma = \sqrt{\lambda n}$, since σ should larger than the smoothing parameter $(\eta_{2^{-\lambda}}(\mathbb{Z}^n))$. As a result, the size of g bounded with $||g|| \leq \sigma\sqrt{n} = n\sqrt{\lambda}$.
- To sample a_i, b_i and level-0 elements, fix $\sigma = \lambda n^{3/2}$. Then, these elements are bounded by λn^2. GGH state that numerator in y and the x_i is bounded by σn^4.
- To sample $r \leftarrow D_{\mathbb{Z}^n, \sigma^*}$, set $\sigma^* = 2^\lambda$. As a result, the numerate size 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 $k-1$ plain-text. 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 attack , 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.

5.3 Security of Corresponding Multilinear Map

Multilinear map can be easily extended from self-bilinear map, we will analyze the the security of multilinear map below.

Modified Edcoding/Decoding Attack Shortly after the original CLT multi-linear map was fully broken by zerozing algorithm, GGH map met extreme challenge provided by Hu and Jia [17]. They name it modified encoding/decoding because of the exclusive processing routines. In fact, their algorithm analyze the GGHLite map [18] which has a little differences with GGH map mainly in the re-randomization procedure. Modified encoding/decoding algorithm almost totally break the n-GDDHA.

The necessary parameters of attack algorithm are "generator" y, re-randomization parameters $X^{(1)}$, $X^{(2)}$. They run following process when face to a $k-$GDDHP instance $(y, X^{(1)}, X^{(2)}, v^{(0)}y + u^{(0,1)}X^{(1)} + u^{(0,2)}X^{(2)}, \cdots, v^{(n)}y + u^{(n,1)}X^{(1)} + u^{(n,2)}X^{(2)}, H)$, where H is the k-level encoding of random elements in R/I or multiply of all $v^{(i)}$.

1. Use weak-DL attack to generate $n+1$ equivalent secret level-0 encodings respectively from level-1 encodings.
2. Multiply these level-0 encodings together to get level-0 encoding of the product.
3. Use modified encoding/decoding procedure to get the new level-n encoding.
4. Extract the high order bits of the result in the step 3.

The output of this procedure can help adversary to break the $n-$MDDHA, since most important bit of output is equivalent to that of P_{zt} times valid encoding.

This attack doesn't threaten our scheme. When they perform the weak-DL attack, it needs compute assistance parameters $X^{(i)} = y^{n-2}x^i x^1 p_{zt} (mod\ q)$. If the multi-linear level $n < 2$, these assistance parameters are obviously hardly generated. Our scheme exactly take use of the level-1 graded encoding system.

n-MCDHA *Assumption* Since self-bilinear map build on GGH graded encoding system is an instance of generic model $\mathcal{SBP}$, the polynomial reduction from ED problem to n-MCDHA problem is directly hold. More over, there exists efficient division in polynomial ring $\mathbb{Z}[x]$. This leads to a more optimized reduction. So, the remain work is to see the hardness of ED in GGH graded encoding system. But ED in GGH can't be reduce to some classic problem directly, we try to analysis some of its property to strengthen the idea that it is hard. This topic will discussed in next section.

6 Further Consideration to ED

In this section, we describe our attempts at cryptanalysis of our self-bilinear map. We mainly have some argue about the hardness of inverse encoding problem. We give a algorithm to solve this problem in directly, and state that these algorithm couldn't solve it efficiently. We also discuss the property of polynomials with coefficient in integer and its residue class ring. The content of them including the co-prime, Euclidean algorithm and so on.

6.1 Division Algorithm

Because ring contains nonzero elements which do not have inverses, one can't introduce division by means of inverse and multiplication. So, There is a division algorithm in ring $\mathbb{Z}$ and $\mathbb{Q}[x]$ which can be conclude as $a = qb + r$, where $r = 0$ or $\sigma(r) < \sigma(b)$, and

$$\begin{aligned} \sigma : R^* &\to N \\ c &\mapsto \sigma(c) \end{aligned}$$

R^* is the set of all elements in R except 0. In $\mathbb{Z}$, the map maps the elements to itself, and in $\mathbb{Z}[x]$, the map maps polynomial to its degree. Sometimes they call rings have such property Euclidean rings. This property ensure that Euclidean algorithm will end in finite steps.

Even though division algorithm isn't known in $\mathbb{Z}[x]$ for $\mathbb{Z}[x]$ do not obtain a map making it a Euclidean ring, the division algorithm defined in $\mathbb{Q}[X]$ is still work in $\mathbb{Z}[x]$, if b is actually a divisor of a, where a, b are inputs of division algorithm. If $r \neq 0$, we say that b don't divide a, $\frac{a}{b}$ is not an elements in $\mathbb{Z}[x]$, $\mathbb{Q}[x]$,or $\mathbb{Z}$.

6.2 Variant of Euclidean Algorithm

In this section we will state that, Euclidean algorithm is defined in $\mathbb{Q}[x]$ and also can be implement in $\mathbb{Z}[x]$. It can compute the greatest common divisor of two elements in $\mathbb{Q}[x]$, since "GCD equals 1" and "co-prime" is a equivalent relation in $\mathbb{Q}[x]$. But when we modify a bit steps to make Euclidean algorithm work in $\mathbb{Z}[x]$ in which "GCD equals 1" and "co-prime" aren't the same meaning, the output of the algorithm can only be used to judging the co-prime relation.

The property of $\mathbb{Z}[x]$ and $\mathbb{Q}[x]$ have quite many differences which is occurred by their coefficients chosen in field and unique factoring domain respectively. The famous Euclidean algorithm is defined to be worked in $\mathbb{F}[X]$, it can calculate the GCD in $\mathbb{F}[x]$. by using the GCD, we can judging the co-prime relation in $\mathbb{F}[x]$. We all knows that GCD is exists in unique factoring domain, But Euclidean algorithm doesn't mentioned in $\mathbb{D}[x]$ in textbook. This happen because that every elements in $\mathbb{F}$ (or polynomial of degree 0) are unit. On input $a, b \in \mathbb{F}[x]$ and assume that $deg(a) \geq deg(b)$, b can cancel the greatest degree term of a. Since Euclidean algorithm is defined iteratively, this happen iteratively. we only discuss Euclidean algorithm in $\mathbb{Z}[x]$ and $\mathbb{Q}[x]$ for division relation in $\mathbb{Q}$, $\mathbb{Z}$ is efficiently discovered. We state that Euclidean algorithm is also worked in $\mathbb{Z}[X]$ after adding a extra step in it, but the information it tells us have some differences. When the Euclidean run in some step and b can't cancel the greatest degree term of a, we multiply b with the leading coefficients of a. For example, on input $3x^2 + 1, 2x + 2$, we can multiply $2x + 2$ with 3, then do the normal progress of Euclidean algorithm. If the final output is c, we can know that $\exists s, t \in \mathbb{Z}[X]$ such that $sa + tb = c$. On this condition, Euclidean algorithm can judging the co-prime relation in $\mathbb{Z}[X]$, but it can't output the GCD in it.

For example. In $\mathbb{Q}[x]$, On input $x^3 + 1$ and $x^2 + 1$, the output of Euclidean algorithm are 1, it means $\exists s, t \in \mathbb{Q}[x]$, let $s(x^3 + 1) + t(x^2 + 1) = 1$. This result tells us the $x^3 + 1$ and $x^2 + 1$ are co-prime in $Q[x]$ and their GCD is also 1. Because $\mathbb{Z}[x] \in \mathbb{Q}[x]$, the GCD of $x^3 + 1$ and $x^2 + 1$ only can be 1 in $\mathbb{Z}[x]$. But when we run algorithm in $\mathbb{Z}[x]$, the final result is 2, which mean the output of it is not the GCD of two elements in $\mathbb{Z}[X]$. But we also have $\exists s, t \in \mathbb{Z}[x]$ such that $s(x^3 + 1) + t(x^2) + 1 = 2$. moreover, we have no idea to change this 2 to become 1. This tells us $x^3 + 1$ and $x^2 + 1$ are not co-prime in $\mathbb{Z}[x]$.

6.3 Co-prime in Ring and its Residue Class

In this section. we notice that if we have an algorithm to check the co-prime relation in R, so all the co-prime relation in its residue class probably can be check.

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 inverse of $\bar{a} \in R/\langle g \rangle$ if it exists, we simply use the representation of coset $a + \langle g \rangle$ and find $s, t, p \in Z[x]$ 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.

6.4 Direct Attack

We describe a direct attack on encoding division assumption. Given an instance of encoding division problem $(\alpha, y, u = \alpha\beta y + rX)$. Since we know α, the adversary can attempt to do operation below to get $\beta y + r'X$.

Assume that $\alpha \in d + I$, where $d \in R/I$. Use d^{-1} to denote inverse of $d$in R/I. We have mentioned that every non zero elements in R/I are invertible.

1. divide u by α in $R_q[x]$.
2. Find short $\alpha' \in d^{-1} + I$, and multiply it to u in $R_q[x]$.

For case 1, If there is an efficient algorithm to divide u by α in $R_q[x]$, then the re-randomization procedure in GGH map is useless. GGH state that the noise in encoding could prevent adversary knowing d from $[dy + rX]_q$ and y. If noise doesn't exist, adversary can divide $[dy]_q$ by y in R_q to get d. That is that y doesn't divide $[dy + rX]_q$ with high probability. The plain-text have been changed even though $[\frac{dy+rX}{y}]_q$ is in R_q. One can't get a short polynomial in $d + I$.

For case 2. To find short $\alpha' \in d^{-1} + I$, Its equivalent to find

$$\alpha\alpha' = 1 \ mod \ f(x) \ mod \ g(x)$$

That is $\exists s, t \in \mathbb{Z}[x]$ such that $\alpha\alpha' + sg(x) + tf(x) = 1$ and α' is short enough. In this function, α, $f(x)$, is given. Adversary could recover a not short enough $g(x)$ by knowing the public parameters of GGH map even though vector g kept secret. Adversary can fix a short α', then find s and t which has no limited conditions. But this method isn't work since the inverse of a element in R/I is unique. If α' the adversary fixed is not in $d^{-1} + I$, the function doesn't have solutions. We know that $|R/I| = det(G)$, where $G = (g, g \cdot x, \cdots, g \cdot x^{n-1})$. This means adversary fixed a valid α' with probability at most $1/det(G)$. Such a method is quite like exhaustion and we believe that to find short α^{-1} is very hard.

7 Conclusion

We described a new notion called one way encoding system (OWES) and assumed that some analog hardness assumptions hold on it. By taking use of indistinguishability obfuscation, we design a self-bilinear map over the OWES. The EBCDHP is proved to be hard if the EDP is hard. We also discussed that the any level of graded encoding system like GGH satisfies the property 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 the polynomial ring and its residue class ring. We believe that the EDP in GGH is as hard as we need.

References

1. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Advances in Cryptology–CRYPTO 2001. pp. 213–229. Springer (2001)
2. Boneh, D., Silverberg, A.: Applications of multilinear forms to cryptography. Contemporary Mathematics 324(1), 71–90 (2003)
3. 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)
4. Cheon, J.H., Han, K., Lee, C., Ryu, H., Stehlé, D.: Cryptanalysis of the multilinear map over the integers. Tech. rep., Cryptology ePrint Archive, Report 2014/906, 2014. http://eprint. iacr. org (2014)
5. Cheon, J.H., Lee, D.H.: A note on self-bilinear maps. Korean Mathematical Society 46(2), 303–309 (2009)
6. 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)
7. Coron, J.S., Lepoint, T., Tibouchi, M.: Practical multilinear maps over the integers. In: Advances in Cryptology–CRYPTO 2013, pp. 476–493. Springer (2013)
8. 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, 201 4. http://eprint. iacr. org (2014)
9. Frey, G., Rück, H.G.: A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. Mathematics of Computation 62(206), 865–874 (1994)
10. 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)
11. 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)
12. 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)
13. 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)
14. Gentry, C., Gorbunov, S., Halevi, S.: Graph-induced multilinear maps from lattices. Tech. rep., Cryptology ePrint Archive, Report 2014/645, 2014. http://eprint. iacr. org (2015)
15. Groth, J., Ostrovsky, R., Sahai, A.: Perfect non-interactive zero knowledge for np. In: Advances in Cryptology–EUROCRYPT 2006, pp. 339–358. Springer (2006)
16. Hofheinz, D., Kiltz, E.: The group of signed quadratic residues and applications. In: Advances in Cryptology–CRYPTO 2009. vol. 5677, pp. 637–653. Springer (2009)

17. Hu, Y., Jia, H.: Cryptanalysis of ggh map. Tech. rep., Cryptology ePrint Archive, Report 2015/301, 2015. http://eprint. iacr. org (2009)
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. Menezes, A., Vanstone, S., Okamoto, T.: Reducing elliptic curve logarithms to logarithms in a finite field. Information Theory IEEE Transactions on 39(5), 1639 – 1646 (1993)
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)