information

Article

Self-Bilinear Map from One Way Encoding System and $i\mathcal{O}$

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

[1] School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410004, China; zhanghuang@csust.edu.cn (H.Z.); 21208051577@stu.csust.edu.cn (T.H.)
[2] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China; weibd@mail.sysu.edu.cn (B.W.); duyusong@mail.sysu.edu.cn (Y.D.)
* Correspondence: isszhfg@mail.sysu.edu.cn

Abstract: A bilinear map whose domain and target sets are identical is called a self-bilinear map. Original self-bilinear maps are defined over cyclic groups. Since the map itself reveals information about the underlying cyclic group, the Decisional Diffie–Hellman Problem (DDH) and the computational Diffie–Hellman (CDH) problem may be solved easily in some specific groups. This brings a lot of limitations to constructing secure self-bilinear schemes. As a compromise, a self-bilinear map with auxiliary information was proposed in CRYPTO'2014. In this paper, we construct this weak variant of a self-bilinear map from generic sets and indistinguishable obfuscation. These sets should own several properties. A new notion, One Way Encoding System (OWES), is proposed to summarize these properties. The new Encoding Division Problem (EDP) is defined to complete the security proof. The OWES can be built by making use of one level of graded encoding systems (GES). To construct a concrete self-bilinear map scheme, Garg, Gentry, and Halvei(GGH13) GES is adopted in our work. Even though the security of GGH13 was recently broken by Hu et al., their algorithm does not threaten our applications. At the end of this paper, some further considerations for the EDP for concrete construction are given to improve the confidence that EDP is indeed hard.

Keywords: self-bilinear map; indistinguishability obfuscation; One Way Encoding System

Citation: Zhang, H.; Huang, T.; Zhang, F.; Wei, B.; Du, Y. Self-Bilinear Map from One Way Encoding System and $i\mathcal{O}$. *Information* **2024**, *15*, 54. https://doi.org/10.3390/info15010054

Academic Editor: Zoran H. Perić

Received: 19 November 2023
Revised: 29 December 2023
Accepted: 4 January 2024
Published: 17 January 2024

1. Introduction

The bilinear map is a very useful cryptographic primitive. It provides solutions for many cryptographic applications such as identity-based encryptions [1–3], non-interactive zero-knowledge proof systems [4–9], attribute-based encryptions [10] and short signatures [11–15], etc. A self-bilinear map is a special variant of bilinear maps whose domain and target groups are identical. Because of this exclusive property, a self-bilinear map may have more interesting potential. A straightforward application of a self-bilinear map is to construct multilinear maps.

A multilinear map is a generalization of the bilinear map. Not long after the bilinear map showed the convenience it brought to cryptography, Boneh and Silveberg [16] imaged applications of a multilinear map. But, they met serious obstacles, when they tried to construct such a good tool. From then on, constructing multilinear maps became a long-standing open problem. Until recently, three candidate multilinear maps were proposed, the GGH13 scheme [17] on ideal lattices, the CLT13 scheme [18] over the integer and the GGH15 [19] on lattices. a multilinear map is a basic component of various cryptographic primitives such as witness encryption [20,21], indistinguishability obfuscation and functional encryption [22], etc.

Recently, the current candidates for multilinear maps met extremely strong challenges. The CLT13 scheme was completely broken by the "zerozing algorithm" [23]. Two patches [24,25] were proposed very soon after the CLT13 was broken. But Coron et al. [26] stated that these two patches were still unsafe. Then, they described a new multilinear map over the integer [27], and this scheme was soon attacked by Cheon et al. [28]. Not long after

the CLT scheme was completely broken; the GGH scheme was also under attack. Hu and Jia designed a modified encoding/decoding algorithm [29] to break the MDDH assumption which is the security basis of various applications. Moreover, Hu and Jia solve the MCDH problem in their further work [30]. As a substrate of the current program obfuscation, the secret encoding version of the GGH13 map was threatened by Miles et al.'s "Annihilation attacks". This attack has broken the security of indistinguishability obfuscation that builds upon the GGH13 map, e.g., [31–36]. From this situation, we can see that constructing a secure and efficient multilinear map is still worthwhile work. This also highlights the study of finding a secure and efficient self-bilinear map.

The first candidate self-bilinear map was designed by Lee [37]. Cheon and Lee [38] remarked that Lee's map is not essentially a self-bilinear. They also proved the impossibility that the secure self-bilinear map could not be constructed over the 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 situation, Yamakawa et al. [39] adopted the signed quadratic residue group $\mathbb{QR}_n^+$ of $\mathbb{Z}_n^*$ where the order of this group is composite and kept secret. The security of their scheme is based on the factoring assumption and the property of indistinguishability obfuscation ($i\mathcal{O}$).

Motivation

In this paper, we build a self-bilinear map with auxiliary information over generic sets instead of cyclic groups. A new concept OWES is defined to describe the generic sets that can be used to construct the weak variant of self-bilinear maps. Besides the one-way problem, we also define an encoding division problem (EDP) in the OWES. Then, we will prove that the Bilinear Computational Diffie–Hellman with Auxiliary Information (BCDHAI) assumption of a self-bilinear map with auxiliary information is held if the EDP in the underlying OWES is hard. The OWES can be initiated by using graded encoding systems (GES). Based on the GGH13 GES [17], a concrete weak variant of the self-bilinear map is proposed. We also analyze the security of the concrete scheme.

The remainder of this paper is organized as follows. In Section 2, we provide some backgrounds of the techniques we used in this paper, including the definition of $i\mathcal{O}$, self-bilinear map with auxiliary information and problems required to be hard in a self-bilinear map with auxiliary information. Then we introduce the new notion of the One Way Encoding System (OWES) in Section 3. Our generic construction of a self-bilinear map from the OWES and $i\mathcal{O}$ is described in Section 4. By instantiating the OWES with GGH13 GES, we give a concrete self-bilinear map with auxiliary information in Section 5, and discuss whether the one-way problem and EDP are hard in GGH13 GES. Finally, we give our work a brief summary.

2. Preliminaries

In this section, we describe the notations that will be used in this paper. Then, we review the $i\mathcal{O}$.

2.1. Notations

We use $\mathbb{Z}$ to denote the set of all integer numbers and $\mathbb{Q}$ to denote the rational number field. $\mathbb{Z}[x]$ are polynomials with coefficients in $\mathbb{Z}$. For a positive integer n, $[n]$ denotes the set $\{x \in \mathbb{Z} | 1 \leq x \leq n\}$. λ is the secure parameter. We denote the discrete Gaussian distribution on S with parameter σ as $D_{S,\sigma}$. For an alphabet x, define $\{x_i\}_{i=1}^n$ as $\{x_1, \cdots, x_n\}$. If R/I is a residue class ring of a ring R, for an element $a \in R$, we use $\bar{a}$ to denote the coset of I where a is one of the representatives. For a set S, $|S|$ denotes the cardinal of S. We say that a function in λ is negligible, written $\mathrm{negl}(\lambda)$, if it vanishes faster than the reciprocal of any positive polynomial. For a polynomial r, its ith coefficient is named by r_i. If M is a probabilistic polynomial time (PPT) algorithm (Turing machine), then by $M(x;r)$ we refer to the result of running M on input x and random string r.

2.2. Indistinguishability Obfuscator

The following formulation of indistinguishability obfuscator is due to Garg et al. [22].

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, and for all security parameters $\lambda \in \mathbb{N}$, 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*

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

An indistinguishability obfuscator is an efficient randomized algorithm that makes circuits C_0 and C_1 computationally indistinguishable if they have the same functionality.

2.3. Self-Bilinear Map with Auxiliary Information

Before we formalize a self-bilinear map with auxiliary information, we recall the ideal notion of a self-bilinear map. An ideal self-bilinear map is a special kind of self-bilinear map whose domain and target groups are identical.

Definition 2 (Ideal Self-bilinear map [38]). *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 the 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 so that $e(g_1, g_2)$ generates G, if both g_1 and g_2 are generators of G.*

It is well known that a k-multilinear map can be constructed inductively from a self-bilinear map (which is essentially a 2-multilinear map). If e_{k-1} is a $(k-1)$-multilinear map from self-bilinear map e_2, a k-multilinear map e_k can be generated by setting

$$e_k(g_1, \ldots, g_{k-1}, g_k) = e_2(e_{k-1}(g_1, \ldots, g_{k-1}), g_k).$$

The fact, that constructing a self-bilinear map is a candidate approach to building a multilinear map, highlights the study of self-bilinear maps.

A self-bilinear map with auxiliary information (described in [39]) is a weak notion of the ideal one, where map e is efficiently computable only if the auxiliary information is given. That is, when one computes $e(g^x, g^y)$, the auxiliary information τ_x for g^x or τ_y for g^y is required.

2.4. Efficient Procedures

Instead of constructing an ideal self-bilinear map, we construct the weak notion of a self-bilinear map [39] which can be formalized as a set of algorithms $\mathcal{SBP}$= (**InstGen, Sample, Enc, Add, Neg, AlGen, Map, AlAdd**) and a ring R. These procedures are described below.

Instance Generation. The randomized **InstGen**(1^λ) takes as input the parameter λ, and outputs **params**, which are descriptions of the group G, the order of G and a self-bilinear map $e : G \times G \to G$.

Element Encoding. Given the instance **params** from above, and an element $a \in R$, the procedure **Enc**(**params**,a) outputs an element in G which encode a. We require that for any $a_1 \neq a_2$, $\mathbf{Enc}(\mathbf{params}, a_1) \neq \mathbf{Enc}(\mathbf{params}, a_2)$.

Group Operation. Given $x, y \in G$, $\mathbf{Add}(\mathbf{params}, x, y)$ computes $x + y \in G$, and $\mathbf{Neg}(x)$ computes $-x \in G$.

Auxiliary Information Generation. The procedure $\mathbf{AIGen}(\mathbf{params}, x)$, outputs corresponding auxiliary information τ_x, on input $x \in R$.

Self-Bilinear Map. The procedure $\mathbf{Map}(\mathbf{params}, \mathbf{Enc}(\mathbf{params}, x_1), \tau_{x_2})$ takes $\mathbf{Enc}(\mathbf{params}, x_1)$ and τ_{x_2} as input, outputs $e(\mathbf{Enc}(\mathbf{params}, x_1), \mathbf{Enc}(\mathbf{params}, x_2))$.

Auxiliary Information Operation. On input auxiliary information τ_{x_1}, τ_{x_2}, $\mathbf{AIAdd}(\mathbf{params}, \tau_{x_1}, \tau_{x_2})$ outputs $\tau_{x_1+x_2}$.

2.5. Hardness Assumptions of $\mathcal{SBP}$

For the ideal self-bilinear map to be cryptographically useful, at least the discrete logarithm (one-way problem) must be hard in the underlying group, and it usually also requires the bilinear-DDH problem to be hard. In the case of the self-bilinear map with auxiliary information, these hardness problems are defined in a slightly different way, since the auxiliary information may reveal extra information about a self-bilinear map and the underlying group. Here, we introduce the bilinear computational Diffie–Hellman with auxiliary information (BCDHAI) assumption and bilinear hashed Diffie–Hellman with auxiliary information (BHDHAI) assumption whose generalizations (if the multilinear level is 2, the BCDHAI (BHDHAI) is equivalent to the MCDHAI (resp., MHDHAI) defined in [39]) are both defined in [39].

Definition 3 (**BCDHAI assumption**). *We say that the BCDHAI assumption holds with respect to $\mathcal{SBP}$ if for any efficient algorithm $\mathcal{A}$,*

$$\Pr[e(g,g)^{a_0a_1a_2} \leftarrow \mathcal{A}(\mathbf{params}, g, g^{a_0}, g^{a_1}, g^{a_2}, \tau_{a_0}, \tau_{a_1}, \tau_{a_2})] \leq negl(\lambda),$$

where $\mathbf{params} \leftarrow \mathbf{InstGen}(1^\lambda)$, g is the generator of G. $a_i \leftarrow ord(G)$, $\tau_{a_i} \leftarrow \mathbf{AIGen}(\mathbf{params}, a_i)$ for $i = 0, 1, 2$.

The BCDHAI assumption is an analog of the classic bilinear computational Diffie–Hellman (BCDH) assumption and the following BHDHAI assumption is the analog of the bilinear hashed Diffie–Hellman assumption.

Definition 4 (**BHDHAI assumption**). *We say that the BHDHAI assumption holds with respect to $\mathcal{SBP}$ and a family of hash functions $\mathcal{H} = \{H : G \to \{0,1\}^k\}$ if for any efficient algorithm D,*

$$\begin{aligned} &|\Pr[1 \leftarrow D(\mathbf{params}, g, g^{a_0}, g^{a_1}, g^{a_2}, \tau_{a_0}, \tau_{a_1}, \tau_{a_2}, H, T)|\beta = 1] \\ &- \Pr[1 \leftarrow D(\mathbf{params}, g, g^{a_0}, g^{a_1}, g^{a_2}, \tau_{a_0}, \tau_{a_1}, \tau_{a_2}, H, T)|\beta = 0]| \leq negl(\lambda) \end{aligned}$$

where $\mathbf{params} \leftarrow \mathbf{InstGen}(1^\lambda)$, g is the generator of G, $a_i \leftarrow ord(G)$, $\tau_{a_i} \leftarrow \mathbf{AIGen}(\mathbf{params}, a_i)$, for all $i = 0, 1, 2$, $\beta \leftarrow \{0,1\}$ and $T \leftarrow \{0,1\}^k$ if $\beta = 0$, and otherwise $T = H(e(g,g)^{a_0a_1a_2})$.

Depending on the work of [39], if the MCDHAI assumption holds with respect to $\mathcal{SBP}$ then the MHDHAI assumption holds with respect to $\mathcal{SBP}$ and the Goldreich–Levin hardcore bit function [40].

3. One Way Encoding Systems

In this section, we will give the definition of the One Way Encoding System (OWES), and describe some problems which are required to be hard in the OWES.

One Way Encoding Systems

The notion of a One Way Encoding System (OWES) is generalized from graded encoding systems (GES) and cryptographic cyclic groups which formed the substrates of current candidate multilinear maps and bilinear maps, respectively. We are trying to refine all properties, which are necessary for building a self-bilinear map. We will first shape the frame of OWES by comparing it to the current GES, and then, show that the frame is also suitable for cryptographic cyclic groups or even more algebraic structures.

We begin by recalling the Modules.

Definition 5 (S_0-modules). *Let R be a commutative ring with identity* 1. *An S_0-module is an abelian group S_1 together with a map*

$$\begin{array}{rccc} \otimes: & S_0 \times S_1 & \to & S_1 \\ & (a, x) & \mapsto & a \otimes x \end{array}$$

satisfying the following properties:

1. $a \otimes (x + y) = a \otimes x + a \otimes y$,
2. $(a + b) \otimes x = a \otimes x + b \otimes x$,
3. $(ab) \otimes x = a \otimes (b \otimes x)$,
4. $1x = x$

for $x, y \in S_1$, $a, b \in S_0$.

Without loss of generality, we make the following further assumptions. Let $(S_0, +, \cdot)$ be a finite commutative integral domain with identity (S_0 is essentially a finite field) and S_0 is a residue class ring of S_0' modulo m (If S_0 is not a rigorous residue class ring, consider $S_0 = S_0'/\langle 0 \rangle = S'$, where $m = 0$). Let $(S_1, \oplus)$ be an abelian group and assume similarly that S_1 is a quotient group S_1'/H, where H is a normal subgroup of S_1' (Regarding S_1 as $S_1/\{e\}$ if it is not, where $\{e\}$ is the subgroup of S_1 which only involves identity e). We make the above assumptions because of the observation of the current graded encoding system.

In practical terms, to manipulate elements in a residue class ring S_0 (e.g, $\bar{a}, \bar{b} \in S_0$, $\bar{a} + \bar{b}$) is instead achieved by doing the corresponding computation in the complete system of coset representatives of S_0 relative to S_0' (e.g., $a, b \in S_0'$, $a + b$).

Definition 6 (Complete system of coset representatives of S_0' relative to $\langle m \rangle$). *Let S_0' be an abelian group and $\langle m \rangle$ be a subgroup of S_0'. From each coset of S_0' relative to $\langle m \rangle$ we choose a coset representative, then the set so obtained, denoted by S_0^{com}, is called a complete system of coset representatives of S_0' relative to $\langle m \rangle$.*

The residing class ring $S_0 = S_0'/\langle m \rangle$ and complete system S_0^{com} are isomorphic. But at most times, the user can hardly choose a unique representative for each coset, if the generator of ideal $\langle m \rangle$ is kept secret. For example, in GGH13 GES [17], the sampled level-0 encoding is a random (and short) representative of some ring element in R/I. Since the $I = \langle g \rangle$ is a secret system parameter, it is hard to fix $|R/I|$ representatives such that the complete system of coset representatives of R relative to I and R/I are isomorphic. Thus, in this situation, the representative of a coset $\{a + kg | a, k \in S_0'\} \in S_0$ is a random variable of the form $a + kg \in S_0'$. In our paper, we will refer a representative of coset $\{a + kg | a, k \in S_0'\}$ as the result of running a PPT Algorithm M, computing $a + kg$, on input a and random string k, where $a \in S_0^{com}$ (A normal user will obtain a representative of the form $a + kg$, but he cannot obtain the system parameter a). We often omit to write $M(a)$ for simplicity, if the context is clear. The above discussion is also suitable for group S_1' and its normal subgroup H. We assume that the complete system of coset representatives of S_1' relative to H is S_1^{com}.

Definition 7 (The representative of elements in S_0). *For any element $\bar{a} \in S_0$, the representative of $\bar{a}$ is a random variable $M(a; k)$, where M is a PPT algorithm that computes the function $a + kg$*

on input $a \in \bar{a}$, and a is a secret element in S_0^{com}. The distribution of $M(a;k)$ is dependent on the distribution of the random string k.

Now we proceed to discuss the notion of valid elements which are generalized from the notion of valid encodings in graded encoding scheme. For an algebraic structure to be cryptographically useful, at least the one-way problem (e.g., discrete logarithm problem) must be hard in it, and the notion of valid (level-0) encoding is crucial for GES to assure that. Informally speaking, if u is a level-1 encoding of $a + I$, one can hardly compute $a' \in S_0'$ such that $a' - a \in I$ efficiently and a' is a valid level-0 encoding. On our side, the level-0 encoding corresponds to a representative of a coset in S_0. The valid representative in S_0' will be defined by limiting the support set of random string r.

Definition 8 (D_0-valid representatives of S_0). *Let D_0 be a set of strings. For S_0-modules $(S_0, S_1, \otimes)$, we say that a representative of $\bar{a} \in S_0$, denoted by $M(a; r_0)$, is D_0-valid, if the support of the random variable of strings r_0, is D_0. Moreover, the set of all D_0-valid representatives in S_0' is*

$$\mathbf{Set}_{D_0} = \{M(a; r_0) | a \in S_0^{com}, r_0 \in D_0\}$$

The discussions above will cause the problem of how can users without system parameters sample valid representatives at random. Thus, we need a (S_0, D_0)-sampler, which is like the ring sampler in GGH13 GES, to solve this problem.

Definition 9 ((S_0, D_0)-sampler). *The (S_0, D_0)-sampler is a PPT algorithm* **Samp**, *which on input security parameter λ and the description of S_0', outputs a random representative $b \leftarrow \mathbf{Samp}(1^\lambda, S_0'; r_0)$ such that*

- *for any $\bar{a} \in S_0$, $\Pr[b \in \bar{a}] = \frac{1}{|S_0|}$,*
- *all representatives sampled by $\mathbf{Samp}(1^\lambda, S_0'; r)$ are in $\mathbf{Set}_{D_0}$.*

The definition shows that (S_0, D_0)-sampler draws a random element b in a residue class $\bar{a}$ relying on the random string r_0. Furthermore, the corresponding residue class $\bar{a}$ obeys the uniform distribution in S_0.

After discussing the valid "level-0 encodings", we proceed to describe the valid "level-1" encodings. A valid "level-0" encoding is a representative in group S_1' with some specific properties.

Definition 10 (D_1-valid representatives of S_1). *For S_0-modules $(S_0, S_1, \otimes)$, we say that a representative of $\bar{x} \in S_1$, denoted by $M(x; r_1)$, is D_1-valid, if the support of random variable of strings r_1 is D_1. Moreover, the set of all D_1-valid representatives in S_1' is*

$$\mathbf{Set}_{D_1} = \{M(x; r_1) | x \in S_1^{com}, r_1 \in D_1\}$$

Since the presentative of the residue class in S_0 is a random variable, we require a zero testing predicate, which is similar to the functionality of the zero testing procedure in GGH13 GES.

Definition 11 (Zero testing predicate for D_1-valid representative in S_1). *The Zero testing predicate for D_1-valid representative in S_1 is a deterministic algorithm* **isZero**(x), *which on input $x \in \bar{x}$, where $\bar{x} \in S_1$, outputs*

$$\mathbf{isZero}(x) = \begin{cases} 1 & \text{, if } x \text{ is } D_1\text{-valid and } \bar{x} = \bar{0} \\ 0 & \text{, otherwise} \end{cases}$$

Now we are ready to give the formal definition of OWES.

Definition 12 ((D_0, D_1)-OWES). *Let S_0, S_1 be the algebraic structure defined above, and $(S_0, S_1, \otimes)$ be S_0-modules. We say that a PPT Turing machine $\mathbf{E}$, which computes the map $\otimes : S_0 \times S_1 \to S_1$, is a (D_0, D_1)-*OWES *if the following properties hold:*

1. *Valid encoding: For every D_0-valid representative a and every D_1-valid representative x, $\mathbf{E}(a, x; r)$ is D_1-valid.*
2. *Valid manipulation: For all D_1-valid $\mathbf{E}(a_1, x_1; r_1)$ and $\mathbf{E}(a_2, x_2; r_2)$, the encoding $\mathbf{E}(a_1, x_1; r_1) + \mathbf{E}(a_2, x_2; r_2)$ is D_1-valid.*
3. *Hard to invert: For every PPT algorithm $\mathcal{A}$ and all sufficiently large λ,*

$$\Pr_{x \in S_0}[\mathbf{isZero}(\mathbf{E}(a', x; r) - \mathbf{E}(a, x; r)) = 1 : a' \leftarrow \mathcal{A}(\mathbf{E}(a, x; r), 1^\lambda)] < negl(\lambda)$$

If we set $S_0 = \mathbb{Z}_p$, $S_1 = G = \langle g \rangle$, $|G| = p$, and set $\otimes$ to be the power operation in G, and let D_0, D_1 be the set of bit strings with a polynomial size length, such a (D_0, D_1)-OWES becomes a cryptographic cyclic group in which the "hard to invert" property is equivalent to the DLP assumption with respect to G. In another case, if we set $S_0 = R/I$, $S_1 = R_q/I$, $\otimes$ to be the GGH13 encoding procedure, and make σ to be a predicate to tell whether an element in a residue class is short, such an σ-OWES is exactly the GGH 13 graded encoding scheme.

For completing the security proof of a self-bilinear map, we have to define a new hard problem called EDP below.

Definition 13 (EDP). *For a (D_0, D_1)-*OWES *$\mathbf{E}(a, x; r)$ with respect to the modules $(S_0, S_1, \otimes)$, the Encoding Division Problem is, on input the $\mathbf{E}(a, b \otimes x; r)$ and $a \in \bar{a}$, where $\bar{a}$ is a unit of S_0, to compute a representative $y \in S_0'$ such that $\mathbf{isZero}(\mathbf{E}(a, y; r) - \mathbf{E}(a, b \otimes x; r)) = 1$.*

The Encoding Division assumption says that there are no PPT algorithms solving the EDP with non-negligible probability.

The OWES can be constructed by making use of one level of graded encoding systems. To construct a concrete $\mathcal{SBP}$, the GGH13 is adopted in Section 5.

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

In this section, we construct the weak self-bilinear map scheme $\mathcal{SBP}$ by using the OWES and $i\mathcal{O}$.

4.1. Our Construction

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

Notation for Circuits on OWES. For the (D_0, D_1)-OWES with respect to the modules $(S_0, S_1, \otimes)$ and $a \in \bar{a}$, where $\bar{a} \in S_0$, $C_a(x)$ denotes the circuit that takes $x \in \bar{x}$, where $\bar{x} \in S_1$ is the input and output an element that is equivalent to $E(a, x; r)$. For circuits $C_a(x)$, $C_b(y)$ whose outputs can be parsed as the element in S_1', respectively, $\mathrm{Plus}(C_a(x), C_b(y))$ denotes a circuit that computes the sum of outputs of $C_a(x)$ and $C_b(y)$.

Now, we are ready to introduce the procedures of the generic constructing $\mathcal{SBP}$. The generic construction of a self-bilinear map is as follows.

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

- On inputting the security parameter λ, initiate (D_0, D_1)-OWES with respect to modules $(S_0, S_1, \otimes)$.
- Choose a random representative $x \in S_1$, where $x \in \bar{x}$ and $\bar{x} \in S_1$.
- Choose an invertible representative $r \in \bar{r}$ at random, where $\bar{r} \in S_0$.
- Output **params** $= \{S_0', S_1', \mathbf{E}(\cdot), r\}$ as the system parameters.

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

$$\begin{array}{rccc} e: & S_1 \times S_1 & \rightarrow & S_1 \\ & (\mathbf{E}(a_1, x), \mathbf{E}(a_2, x)) & \mapsto & \mathbf{E}(ra_1a_2, x) \end{array}$$

Encoding: $\mathbf{E}(a, x; r_1) \leftarrow$ **Enc**$(\mathbf{params}, \mathbf{a})$.

- On input **params** and $a \in \bar{a}$, where $\bar{a} \in S_0$, compute $\mathbf{E}(a, x; r_1)$.

Auxiliary Information Generation: $\tau_a \leftarrow$ **AIGen**$(\mathbf{params}, a)$

- On input $a \in \bar{a}$, where $\bar{a} \in S_0$, generate the corresponding $\tau_a = i\mathcal{O}(C_{ra})$.

Adding encodings:

- It is easy to see that the encoding as above is additively homomorphic, in the sense that adding encodings yields an encoding of the sum.

Auxiliary Information Manipulation: $\tau_{a+b} \leftarrow$ **AIAdd**$(\mathbf{params}, \tau_a, \tau_b)$

- On input, the auxiliary information τ_a and τ_b, compute $\tau_{a+b} \leftarrow i\mathcal{O}(\text{Plus}(\tau_a, \tau_b))$.

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

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

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

We prove that the BCDHAI assumption holds with respect to our generic construction $\mathcal{SBP}$ if $i\mathcal{O}$ is an indistinguishability obfuscator for $P/poly$ and the EDP in the corresponding OWES is hard.

The *BCDHAI* assumption holds with respect to $\mathcal{SBP}$ if the *EDP* is hard in the underlying *OWES* and $i\mathcal{O}$ is an indistinguishability obfuscator for $P/poly$.

Proof. Assume that the algorithm $\mathcal{A}$ can solve the BCDHAI problem in $\mathcal{SBP}$. We consider the following games.

Game 1. This game is the original BCDHAI problem game.

1. Initiate the (D_0, D_1)-OWES with respect to the modules $(S_0, S_1, \otimes)$. Choose a random representative $x \in S_1'$, where $x \in \bar{x}$ and $\bar{x} \in S_1$. Choose an invertible representative $r \in \bar{r}$ at random, where $\bar{r} \in S_0$. Set the **params** $= \{S_0', S_1', \mathbf{E}(\cdot), x, r\}$. **params** describe a $\mathcal{SBP}$.
2. Run the (S_0, D_0)-Sampler to obtain $a_0, a_1, a_2 \in S_0'$, so that $\bar{a}$, $\bar{b}$, and $\bar{c}$ are distributed uniformly in S_0.
3. Compute $\mathbf{E}(a_i, x)$ and its corresponding auxiliary information $\tau_{a_i} = i\mathcal{O}(C_{ra_i})$ for $i = 0, 1, 2$
4. $U \leftarrow \mathcal{A}(\mathbf{params}, \mathbf{E}(a_0, x), \mathbf{E}(a_1, x), \mathbf{E}(a_2, x), \tau_{a_0}, \tau_{a_1}, \tau_{a_2})$.

Game 2. This game is the same as Game 1 except that $a_0, a_1, a_2, \tau_{a_0}, \tau_{a_1}, \tau_{a_2}$ are set differently.

1. Initiate the (D_0, D_1)-OWES with respect to the modules $(S_0, S_1, \otimes)$. Choose a random representative $x \in S_1'$, where $x \in \bar{x}$ and $\bar{x} \in S_1$. Choose an invertible representative $r \in \bar{r}$ at random, where $\bar{r} \in S_0$. Compute $x = r \otimes y$. Output **params** $= \{S_0', S_1', \mathbf{E}(\cdot), x, r\}$. **params** describe a $\mathcal{SBP}$.
2. Choose $a_0', a_1', a_2' \in S_0'$, where $\bar{a}_0', \bar{a}_1', \bar{a}_2' \in S_0$ are distributed uniformly.
3. Let $ra_i = ra_i' + 1$. Thus, $\mathbf{E}(a_i, x) = \mathbf{E}(a_i, r \otimes y) = \mathbf{E}(ra_i' + 1, y)$, for $i = 0, 1, 2$.
4. Generate the auxiliary information $\tau_{a_i} = i\mathcal{O}(C_{ra_i'+1})$, for $i = 0, 1, 2$.
5. $U \leftarrow \mathcal{A}(\mathbf{params}, \mathbf{E}(a_0, x), \mathbf{E}(a_1, x), \mathbf{E}(a_2, x), \tau_{a_0}, \tau_{a_1}, \tau_{a_2})$.

We say that $\mathcal{A}$ wins these games if $U = \mathbf{E}(ra_0a_1a_2, x)$. Let $\Pr[T_i]$ denote the probability that $\mathcal{A}$ wins Game i, for $i = 1, 2$. Next, we will prove that $|\Pr[T_1] - \Pr[T_2]|$ is negligible if $i\mathcal{O}$ is an indistinguishability obfuscator for $P/poly$. The hybrid games $H_0, \ldots, H_3$ are considered. H_i is the same as Game 2 except that the first i auxiliary information is generated as in Game 1. Therefore, H_0 is identical to Game 2 and H_3 is identical to Game 1. If H_i is indistinguishable from H_{i+1}, for $i = 0, 1, 2$, then Game 1 is indistinguishable from Game 2. Now, we assume that $\mathcal{A}$ wins H_i and H_{i+1} with probability $\Pr[H_i]$ and $\Pr[H_{i+1}]$, respectively, and $|\Pr[H_i] - \Pr[H_{i+1}]| = \gamma(\lambda)$ is a non-negligible value, for $i = 0, 1, 2$. The newly designed Algorithm 1 works as follows.

Algorithm 1 The Games Distringuisher

1: Initiate the (D_0, D_1)-OWES with respect to the modules $(S_0', S_1', \otimes)$, $\mathcal{B}$ wants to know the circuit C^* comes from $i\mathcal{O}(C_{ra_i})$ or $i\mathcal{O}(C_{ra_i'+1})$, where $ra_i = ra_i' + 1$.
2: Compute $x = r \times y$, where $r \in S_0$ is invertible and $b \in S_1$. Then, set **params** $= (S_0', S_1', \mathbf{E}(\cdot), x, r)$. **params** describe a $\mathcal{SBP}$.
3: Choose $a_j' \in S_0, j \in \{0, 1, 2\}$ at random, and compute $ra_j = ra_j' + 1$.
4: Set $C_0 = C_{ra_{i-1}}$, $C_1 = C_{2a_{i-1}'} + 1$.
5: Set

$$\tau_{a_j} = \begin{cases} i\mathcal{O}(C_{ra_i'+1}) & , \text{ if } j = 0, \ldots, i-2 \\ C^* & , \text{ if } j = i-1 \\ i\mathcal{O}(C_{ra_j}) & , \text{ if } j = i, \ldots, 2 \end{cases}$$

6: $\mathcal{B}$ runs $\mathcal{A}(\mathbf{params}, \mathbf{E}(a_0, x), \mathbf{E}(a_1, x), \mathbf{E}(a_2, x), \tau_{a_0}, \tau_{a_1}, \tau_{a_2})$ to obtain U.
7: If $\mathbf{isZero}(U - \mathbf{E}(ra_0a_1a_2, x)) = 1$, outputs 1, and otherwise output 0.

If $C^* = i\mathcal{O}(C_{ra_i})$, $\mathcal{B}$ simulates H_{i-1} for $\mathcal{A}$, otherwise it simulates H_i. With the hypothesis, we have

$$|\Pr[1 \leftarrow \mathcal{B}(i\mathcal{O}(C_{ra_i}))] - \Pr[1 \leftarrow \mathcal{B}(i\mathcal{O}(C_{ra_i'+1}))]| = |\Pr[H_i] - \Pr[H_{i+1}]| \geq \gamma(\lambda),$$

which means $\mathcal{B}$ breaks the security of $i\mathcal{O}$ with non-negligible probability, in contradiction to the assumption. Thus H_i and H_{i+1} are computationally indistinguishable; so are Game 1 and Game 2.

At the end of the proof, we give an Algorithm 2 which reduces the EDP to the BCDHAI Problem in Game 2.

Algorithm 2 The reduction of EDP to BCDHAI problem in Game 2

1: $\mathcal{C}$ takes an EDP instance $(S_0, S_1, \otimes), \mathbf{E}(\cdot), y, r$ as input.
2: Compute $x = r \otimes y$, and output **params** $= \{S_0', S_1', \mathbf{E}(\cdot), x, r\}$.
3: Choose $a_0', a_1', a_2' \in S_0'$, where $\bar{a}_0', \bar{a}_1', \bar{a}_2' \in S_0$ are distributed uniformly.
4: Set $a_i \otimes x = (a_i' \otimes x) + y$, this implies that $a_i = a_i' + r^{-1}$, for $i = 1, 2, 3$.
5: Generate the auxiliary information $\tau_{a_i} = i\mathcal{O}(C_{ra_i}) = i\mathcal{O}(C_{ra_i'+1})$, for $i = 0, 1, 2$.
6: Send **params**, $\{a_i \otimes x\}_{i=0}^2$ and $\{\tau_{a_i}\}_{i=0}^2$ to $\mathcal{A}$. $\mathcal{A}$ outputs U.
7: Compute $q = \frac{(r \otimes a_1' + 1)(r \otimes a_2' + 1) - 1}{r} = r \otimes a_1'a_2' + a_1' + a_2'$.
8: Compute $p = a_0' \otimes (r \otimes a_1' + 1)(r \otimes a_2' + 1)$, and output $U' = U - [p + q] \otimes y$.

Correctness: If the output of Algorithm 2 is $U = \mathbf{E}(ra_0a_1a_2, x)$. Assume that $\ominus$ is the inverse operation of $\oplus$.

$$\begin{aligned}
U &= \mathbf{E}(ra_0a_1a_2, x) \\
&= \mathbf{E}(ra_0a_1a_2, ry) \\
&= \mathbf{E}[(a_0)(ra_1)(ra_2), y] \\
&= \mathbf{E}[(a'_0 + r^{-1})(ra'_1 + 1)(ra'_2 + 1), y] \\
&= \mathbf{E}[(a'_0(ra'_1 + 1)(ra'_2 + 1)) + r^{-1}(ra'_1 + 1)(ra'_2 + 1), y] \\
&= \mathbf{E}[(a'_0(ra'_1 + 1)(ra'_2 + 1)) + r^{-1}(r^2a'_1a'_2 + ra'_1 + ra'_2) + r^{-1}, y] \\
&= \mathbf{E}[(a'_0(ra'_1 + 1)(ra'_2 + 1)) + (ra'_1a'_2 + a'_1 + a'_2) + r^{-1}, y] \\
U' &= U \ominus [p + q] \otimes y \\
&= U \ominus [a'_0(ra'_1 + 1)(ra'_2 + 1) + r \otimes a'_1a'_2 + a'_1 + a'_2] \otimes y \\
&= \mathbf{E}(r^{-1}, y)
\end{aligned}$$

Time complexity: We use $T(\cdot)$ to denote the time complexity. Besides the sub-routing $\mathcal{A}$, the number of manipulations in each step of $\mathcal{C}$ is a constant. Assume that the sum of these constants 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 2 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 2 is a polynomial reduction from EDP to the BCDHAI problem. Since EDP is hard, the algorithm that can solve the BCDHAI problem with respect to Game 2 does not exist. Since Game 2 and Game 1 are computationally indistinguishable, the BCDHAI assumption also holds in Game 1 (Game 1 is the original scheme). □

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

The OWES can at least be constructed by making use of the graded encoding system (GES). To design a concrete $\mathcal{SBP}$ scheme, the GGH13 GES [17] is adopted as an example.

5.1. Relationships between GGH13 and OWES

To construct a concrete OWES, only one level of the GGH13 is needed. Even though GGH13 does not completely satisfy the property of OWES, some relaxation could lead us to our destination. We introduce the relationship between GGH13 and OWES by first recalling the GGH13.

Depending on the security parameter λ, GGH13 consists of three sets $R = \mathbb{Z}[x]/\langle f(x)\rangle$, $R_q = R/qR$, and R/I, where $f(x) = x^n + 1$, $I = \langle g\rangle$, $g \in R$, encoded elements are the short representative of elements in R/I. GGH13 outputs the public parameters $y = [a/z]_q$, $x_i = [b_i/z]_q$ where $a \in 1 + I, b \in I$ are short. For a representative $d \in d + I$, it is encoded as $[(da + b)/z]_q$ at level 1. Note, that $[(da + \sum r_ib_i)/z]_q$ is a representative of the unique element in R_q/I. A zero testing parameter is used to check whether u is the highest level encoding of I. Now we are ready to compare GGH13 to OWES.

Assume that we initiate a GGH13 GES with the multi-linearity level $\kappa = 1$. We explain what parameters in GGH13 act as S_0, S_1, f in OWES and how to define the hard problem in GGH13 as the OWES requires.

- **Explanation for** S_0: Regard the R/I as S_0 of OWES. Since R is a cyclotomic ring and I is a prime ideal of R, R/I is an integral domain. Furthermore, R/I consists of finite elements, so R/I is actually a finite field. Level-0 encoding is a short representative of $d + I$, where $d \in R$.
- **Explanation for** S_1: Let R_q/I be the S_1 of OWES. The Level-1 encoding is representative of $d + I + kq$, where $d, q, k \in R$.
- **Explanation for** f: The encoding algorithm is $f : R/I \to R_q/I$. But we cannot design this function without the representative. Thus, $f_r(d) = [dy + \sum r_ix_i]_q$, where r is a random vector sampled from discrete Gaussian distribution. Note, that for a specified

d, the output of $f_r(d)$ is a random value in R_q. f is not even a function (or map) from R to R_q. But the output of $f_r(d)$ is a unique value in R_q/I, this may be the reason why the zero-testing procedure will work in GGH13.

- **Explanation for hard problem:** In GGH13, Given a level-1 encoding $[dy + \sum r_i x_i]_q$, it is not hard for adversaries to find a not short representative in $d + I$. This contradicts the property of OWES. The same problem happens in EDP. So we make a relaxation to the one-way property and EDP for the concrete construction.

Definition 14 (**A relaxation of One Way Property**). *For the* OWES *constructed from* GGH13, *we say that the one-way property holds if the following problem is hard. Given a level-1 encoding* $[dy + \sum r_i x_i]_q$, *it is hard to find a short* $d' \in d + I$.

Definition 15 (**A relaxation of EDP**). *For the* OWES *constructed from* GGH13, *the* EDP *is, on input* $[\alpha dy + \sum r_i x_i]_q, \alpha$, *to compute* $[d'y + \sum r'_i x_i]_q$ *such that* $d' \in d + I$.

The modified one-way property is held in GGH13 since this problem is essentially the analog of a discrete logarithmic problem. We believe that the new EDP is also hard in GGH13, but we cannot reduce it to some classical hard problems. Some further consideration to EDP is given in Section 5.4.2 to improve the secure confidence.

5.2. Construction

The concrete construction is parameterized by the security parameter λ. Based on it, we generate an instance of the GGH13 with multi-linearity level $k = 1$. We will use the symbol $c^{(d)}$ to denote the level-1 encoding of $d + I$ for simplicity. The notation for the circuit on OWES is defined similarly as that in Section 4.1. The concrete $\mathcal{SBP}$ scheme is disigned below.

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

- Take as input the security parameter λ, and generate the 1-GES. It has the following parameters: $y = c^{(1)}$; re-randomization parameters $x_i = c^{(0)}$, $i = [m]$; the zero testing parameter $P_{zt} = [hz/g]_q$.
- Choose a random element $\alpha \leftarrow D_{Z^m,\sigma'}$.
- Choose a random element $s \leftarrow D_{\mathbb{Z}^m,\sigma'}$, and compute $v = s \cdot y$.
- Define **params** $= (v, \{x_i\}_{i=1}^m, \alpha, P_{zt})$ and publish them.

Even though R/I and R_q/I are not published explicitly, GGH13 provides a sampling level-zero encoding procedure to sample an element in R/I uniformly at random (choose d from $D_{\mathbb{Z}^m,\sigma'}$, $d + I$ obey the uniform distribution in R/I). Since the encoding parameters are published explicitly, R_q/I is also known by users. However, users may not know the particular representative of an element in R_q/I (like a "short" representative). P_{zt} helps to check whether two elements in R_q/I are identical. After the instance generation procedure is executed, a self-bilinear map e is defined as

$$\begin{array}{rccc} e: & R_q/I \times R_q/I & \to & R_q/I \\ & (c^{(d)}, c^{(d')}) & \mapsto & c^{(\alpha dd')} \end{array}$$

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

- Compute $c^{(d)} = [dv + \sum_{i=1}^m r_i x_i]_q$, where $r \leftarrow D_{\mathbb{Z}^m,\sigma^*}$.
- 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: $c^{(\alpha dd')} \leftarrow \mathbf{Map}(\mathbf{param}, c^{(d)}, \tau_{c^{(d')}})$. Run the circuit $\tau_{c^{(d')}}(c^{(d)})$ to compute $c^{(\alpha dd')} = [\alpha d' c^{(d)}]_q$.

We also need the additional procedure **isZero** to check whether a element is an encoding of $0 + I$.

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

5.3. Setting the Parameters

The setting of parameters should satisfy the basic requirements of GGH13.

- To sample the $g \leftarrow D_{\mathbb{Z}^n,\sigma}$, set $\sigma = \sqrt{\lambda n}$, σ should be 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 the 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 the 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 obtain 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 the Concrete Construction

The proof of the hard assumption in the concrete construction directly follows that of the generic construction with minor differences, so we omitted it here. In this section, we discussed the algorithm proposed by Hu et al. which almost totally solves the k-MDDH problem in GGH13 GES. We state that Hu's algorithm does not threaten our scheme. Then, we try to analyze the hardness of the concrete EDP in GGH13.

5.4.1. Modified Encoding/Decoding Attack

Hu et al. provided the modified encoding/decoding algorithm to solve the k-MDDHP [29] in the advanced multilinear map GGHLite [41]. If we use $c_k^{(d)}$ to denote the level-k encoding of $I + d$, $\left\{\{c_1^{(d_i)}\}_{i=1}^{k+1}, T\right\}$ is an instance of the k-MDDHP, 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 obtain the level-0 encoding $\prod_{i=1}^{k+1} d_i$.
3. Use the modified encoding/decoding procedure to obtain the parameter T' that is functionally the same as $p_{zt} c_k^{(\prod_{i=1}^{k+1} d_i)}$.
4. Compare the high order bits of T and T'. If they are the same, output 1, otherwise, output 0.

If T is computed from $\{c_1^{(d_i)}\}_{i=1}^{k+1}$, this procedure will output 1 with overwhelming probability. Even though the algorithm of Hu et al. can solve the MDDH problem, it does not threaten our scheme.

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

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

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

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 $0 \leq k \leq 2$, $k-1$ or $k-2$ will be smaller than 0. On one hand, since some

elements in the ring R_q are not invertible, y^{k-2} can not always be computed. On the other hand, if $y^{2-\kappa}$ is invertible in R_q, the invert operations cannot ensure that the coefficient of y^{k-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 k-MDDHP, for $k \geq 3$.

Our self-bilinear map scheme adopts the level-1 encoding of the GGH13. The parameter $k = 1$, which means "Modified Encoding/Decoding Attack" does not threaten our self-bilinear map.

5.4.2. Further Consideration for EDP

We discuss the hardness of EDP in the concrete OWES. An instance of EDP in the concrete OWES is denoted as $(\alpha, v, c^{(\alpha d)} = \alpha dv + \sum r_i x_i)$. Assume that $\alpha \in A$, $d \in B$, A, B are elements in R/I. Every element in R/I is invertible because $I = \langle g \rangle$ is the prime ideal of R and R/I is a finite set. Since α is public, the adversary could try to solve EDP as follows.

1. Divide $c^{(\alpha d)} = \alpha dv + \sum r_i x_i$ by α in R.
2. Divide $c^{(\alpha d)} = \alpha dv + \sum r_i x_i$ by α in R_q.
3. Find short enough $a' \in A^{-1}$, and compute $c' = [\alpha' c^{(\alpha d)}]_q$. c' is a valid level-1 encoding of B.

Case 1. We cannot conduct the division in R directly, since the Euclidean algorithm is defined in $\mathbb{Q}[X]$. Elements in R can be regarded as polynomials with degree less than n. Thus, $c^{(\alpha d)}$ divide α can be written as

$$\frac{\alpha dv + \sum r_i x_i}{\alpha} = dv + \frac{\sum r_i x_i}{\alpha}.$$

$\sum r_i x_i$ is an element in I. It can be written as a polynomial $\sum r_i x_i = k(x)g(x) + l(x)f(x)$, where $k(x), l(x) \in \mathbb{Z}[X]$. Since $\alpha(x)$ is a random polynomial, a degree smaller than n and $g(x)$ generates a prime ideal for R, $\alpha(x) \nmid g(x)$ and $\alpha(x) \nmid f(x)$ in $\mathbb{Z}[x]$ with high probability. Thus, $\frac{\sum r_i x_i}{\alpha}$ is not an element in R and the first method cannot output the right answer for EDP.

Case 2. Computing $[\frac{\alpha dv + \sum r_i x_i}{\alpha}]_q$ has a similar problem.

Case 3. If the short $a' \in A^{-1}$ is found, attack method 3 truly can solve EDP. We discuss the hardness 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

$$\begin{array}{rl} & (p + kf) + (r + k'f)(g + k''f) \\ = & p + rg + (k'g + k''f + rk'')f \\ = & p + rg + r'f \end{array} \quad (1)$$

where $r' = k'g + k''f + rk''$. Note, that (1) is a polynomial $\mathbb{Z}[X]$. This fact tells us, the element $\bar{p}$ in R/I can be written as $p + rg + r'f$, and $p \in \mathbb{Z}[X]$ is a representative of $\bar{p}$.

Thus, to find an element $\alpha' \in A^{(-1)}$ is equivalent to find polynomials $\alpha', s, t \in \mathbb{Z}[X]$ such that

$$\alpha'\alpha + sg + tf = 1 \quad (2)$$

where f is a public parameter, g is a secret parameter, but GGH13 states that a not short representation $g' \in \langle g \rangle$ could be recovered. Equation (2) has three variables, thus to find a random element α' is easy. But it is hard to output the α' with small coefficients.

Of cause adversaries can fix a short α' and find random s, t that satisfies Equation (1). But Equation (1) has solutions if and only if the fixed α' is a representative of A^{-1}. The probability $\Pr[\alpha' \in A^{(-1)}] = |R/I|^{-1}$ and $|R/I|$ should be an exponential function of the secure parameter (otherwise, the analog of the discrete logarithmic problem is easy in GGH13). So, the probability of finding the short α' in case 3 is negligible.

As a result, the EDP seems difficult in the OWES constructed from GGH13.

6. Conclusions

We described a new notion called a 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 a graded encoding system like GGH can be used to construct OWES. After that, a concrete construction from the GGH13 encoding system is proposed. To increase confidence in security, we give a simple analysis of EDP in the concrete OWES.

Author Contributions: Conceptualization, H.Z., T.H. and F.Z.; methodology, H.Z., F.Z., B.W. and Y.D; validation, T.H., F.Z. and B.W.; formal analysis, H.Z., T.H., F.Z. and Y.D.; writing—original draft preparation, H.Z., T.H., F.Z. and Y.D.; writing—review and editing, H.Z., T.H., F.Z., B.W., and Y.D.; supervision, F.Z., Y.D. and B.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by "the Natural Science Foundation of Hunan Province grant number 2023JJ40054", "the Guangdong Basic and Applied Basic Research Foundation grant number 2022A1515011512" and "the scholarship under the State Scholarship Fund of China Scholarship Council grant number 202208430100".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Boneh, D.; Franklin, M. Identity-based encryption from the Weil pairing. In *Advances in Cryptology–CRYPTO 2001 ;* Springer: Berlin/Heidelberg, Germany, 2001; pp. 213–229.
2. Lee, K.; Park, J.H.; Lee, D.H. Anonymous HIBE with short ciphertexts: Full security in prime order groups. *Des. Codes Cryptogr.* **2015**, *74*, 395–425. [CrossRef]
3. Clark, J.; van Oorschot, P.; Ruoti, S.; Seamons, K.; Zappala, D. SoK: Securing Email—A Stakeholder-Based Analysis. In *Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2021.
4. Groth, J.; Ostrovsky, R.; Sahai, A. Perfect non-interactive zero knowledge for NP. In *Advances in Cryptology–EUROCRYPT 2006*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 339–358.
5. Mahapatra, S.; Wooldridge, T.; Wang, X. A Post-quantum Zero-Knowledge Proof System Using Quantum Information Theory. In Proceedings of the Seventh International Congress on Information and Communication Technology, London, UK, 21–24 February 2022 ; Springer: Berlin/Heidelberg, Germany, 2023.
6. Eli, B.; Brent, W.; David, J. Batch Arguments to NIZKs from One-Way Functions. Technical Report, Cryptology ePrint Archive, Report 2023/1938, 2023. Available online: https://eprint.iacr.org/2023/1938 (accessed on 23 December 2023).
7. Badrinarayanan, S.; Patranabis, S.; Sarkar, P. Statistical Security in Two-Party Computation Revisited. In *Theory of Cryptography*; Springer: Berlin/Heidelberg, Germany, 2022.
8. Singh, N.; Dayama, P.; Pandit, V. Zero Knowledge Proofs Towards Verifiable Decentralized AI Pipelines. In *Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2022.
9. Cascudo, I.; Giunta, E. On Interactive Oracle Proofs for Boolean R1CS Statements. In *Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2022.
10. Sahai, A.; Waters, B. Fuzzy Identity-Based Encryption. In *Advances in Cryptology–EUROCRYPT 2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 457–473.
11. Boneh, D.; Lynn, B.; Shacham, H. Short signatures from the Weil pairing. In *Advances in Cryptology ASIACRYPT 2001*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 514–532.
12. Zhang, F.; Safavi-Naini, R.; Susilo, W. An efficient signature scheme from bilinear pairings and its applications. In *Public Key Cryptography–PKC 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 277–290.
13. Chatzigiannis, P.; Baldimtsi, F.; Chalkias, K. SoK: Blockchain Light Clients. In *Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2022.
14. Vesely, P.E.A. Plumo: An Ultralight Blockchain Client. In *Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2022.

15. Abdelhaliem, B. A Signature Scheme from Full-Distance Syndrome Decoding. Technical Report, Cryptology ePrint Archive, Report 2023/1956, 2023. Available online: https://eprint.iacr.org/2023/1956 (accessed on 24 December 2023).
16. Boneh, D.; Silverberg, A. Applications of multilinear forms to cryptography. *Contemp. Math.* **2003**, *324*, 71–90.
17. Garg, S.; Gentry, C.; Halevi, S. Candidate Multilinear Maps from Ideal Lattices. In *Advances in Cryptology–EUROCRYPT 2013*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7881, pp. 1–17.
18. Coron, J.S.; Lepoint, T.; Tibouchi, M. Practical multilinear maps over the integers. In *Advances in Cryptology–CRYPTO 2013*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 476–493.
19. Gentry, C.; Gorbunov, S.; Halevi, S. Graph-induced multilinear maps from lattices. In *Theory of Cryptography*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 498–527.
20. 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, Palo Alto, CA, USA, 2–4 June 2013; ACM: New York, NY, USA, 2013; pp. 467–476.
21. Baghery, K.; Kohlweiss, M.; Siim, J.; Volkhov, M. Another Look at Extraction and Randomization of Groth's zk-SNARK. In *Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2021.
22. Garg, S.; Gentry, C.; Halevi, S.; Raykova, M.; Sahai, A.; Waters, B. Candidate indistinguishability obfuscation and functional encryption for all circuits. In Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, Berkeley, CA, USA, 26–29 October 2019; IEEE: New York, NY, USA, 2013; pp. 40–49.
23. 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*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 3–12.
24. Garg, S.; Gentry, C.; Halevi, S.; Zhandry, M. Fully Secure Functional Encryption without Obfuscation. Technical Report, Cryptology ePrint Archive, Report 2014/666. 2014. Available online: https://eprint.iacr.org/2014/666 (accessed on 28 August 2014).
25. Boneh, D.; Wu, D.J.; Zimmerman, J. Immunizing Multilinear Maps Against Zeroizing Attacks. Technical Report, Cryptology ePrint Archive, Report 2014/930, 2014. Available online: https://eprint.iacr.org/2014/930 (accessed on 13 November 2014).
26. Coron, J.S.; Lepoint, T.; Tibouchi, M. Cryptanalysis of two candidate fixes of multilinear maps over the integers. Technical Report, Cryptology ePrint Archive, Report 2014/975, 2014. Available online: https://eprint.iacr.org/2014/975 (accessed on 1 December 2014).
27. Coron, J.S.; Lepoint, T.; Tibouchi, M. New Multilinear Maps over the Integers. In Proceedings of the Annual Cryptology Conference, Santa Barbara, CA, USA, 16–20 August 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 267–286.
28. Cheon, J.H.; Fouque, P.A.; Lee, C.; Minaud, B.; Ryu, H. Cryptanalysis of the new clt multilinear map over the integers. In *Advances in Cryptology EUROCRYPT 2016*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 509–536.
29. Hu, Y.; Jia, H. Cryptanalysis of GGH Map. In *Advances in Cryptology–EUROCRYPT 2016*; Springer: Berlin/Heidelberg, Germany, 2016.
30. Jia, H.; Hu, Y. Cryptanalysis of multilinear maps from ideal lattices: Revisited. *Des. Codes Cryptogr.* **2016**, *84*, 311–324. [CrossRef]
31. Brakerski, Z.; Rothblum, G.N. Virtual black-box obfuscation for all circuits via generic graded encoding. In *Theory of Cryptography*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 1–25.
32. Barak, B.; Garg, S.; Kalai, Y.T.; Paneth, O.; Sahai, A. Protecting obfuscation against algebraic attacks. In *Advances in Cryptology–EUROCRYPT 2014*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 221–238.
33. Pass, R.; Seth, K.; Telang, S. Indistinguishability obfuscation from semantically-secure multilinear encodings. In *Advances in Cryptology–CRYPTO 2014*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 500–517.
34. Ananth, P.; Gupta, D.; Ishai, Y.; Sahai, A. Optimizing Obfuscation: Avoiding Barrington's Theorem. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3–7 November 2014; ACM: New York, NY, USA, 2014; pp. 646–658.
35. Miles, E.; Sahai, A.; Weiss, M. Protecting obfuscation against arithmetic attacks. Technical Report, Cryptology ePrint Archive, Report 2014/878, 2014. Available online: https://eprint.iacr.org/2014/878 (accessed on 28 October 2014).
36. Badrinarayanan, S.; Miles, E.; Sahai, A.; Zhandry, M. Post-Zeroizing Obfuscation: The case of Evasive Circuits. Technical Report, Cryptology ePrint Archive, Report 2015/167, 2015. Available online: https://eprint.iacr.org/2015/167 (accessed on 27 February 2015).
37. Lee, H.S. A self-pairing map and its applications to cryptography. *Appl. Math. Comput.* **2004**, *151*, 671–678. [CrossRef]
38. Cheon, J.H.; Lee, D.H. A note on self-bilinear maps. *Korean Math. Soc.* **2009**, *46*, 303–309. [CrossRef]
39. 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*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8617, pp. 90–107.
40. Goldreich, O.; Levin, L.A. A hard-core predicate for all one-way functions. In Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, 14–17 May 1989; ACM: New York, NY, USA, 1989; pp. 25–32.
41. Langlois, A.; Stehlé, D.; Steinfeld, R. GGHLite: More efficient multilinear maps from ideal lattices. In *Advances in Cryptology–EUROCRYPT 2014*; Springer: Berlin/Heidelberg, Germany,2014; pp. 239–256.