Self-bilinear Map on Unknown Order Groups from Indistinguishability Obfuscation and Its Applications[⋆]

Takashi Yamakawa[1,2], Shota Yamada[2],
Goichiro Hanaoka[2], and Noboru Kunihiro[1]

The University of Tokyo
yamakawa@it.k.u-tokyo.ac.jp, kunihiro@k.u-tokyo.ac.jp
National Institute of Advanced Industrial Science and Technology (AIST)
{yamada-shota, hanaoka-goichiro}@aist.go.jp

Abstract. A self-bilinear map is a bilinear map where the domain and target groups are identical. In this paper, we introduce a *self-bilinear map with auxiliary information* which is a weaker variant of a self-bilinear map, construct it based on indistinguishability obfuscation and prove that a useful hardness assumption holds with respect to our construction under the factoring assumption. From our construction, we obtain a multilinear map with interesting properties: the level of multilinearity is not bounded in the setup phase, and representations of group elements are compact, i.e., their size is independent of the level of multilinearity. This is the first construction of a multilinear map with these properties. Note, however, that to evaluate the multilinear map, auxiliary information is required. As applications of our multilinear map, we construct multiparty non-interactive key-exchange and distributed broadcast encryption schemes where the maximum number of users is not fixed in the setup phase. Besides direct applications of our self-bilinear map, we show that our technique can also be used for constructing somewhat homomorphic encryption based on indistinguishability obfuscation and the Φ-hiding assumption.

Keywords: self-bilinear map, indistinguishability obfuscation, multilinear map

1 Introduction

1.1 Background

Bilinear maps are an important tool in the construction of many cryptographic primitives, such as identity-based encryption (IBE) [2], attribute-based encryption (ABE) [22], non-interactive zero-knowledge (NIZK) proof systems [16] etc. The bilinear maps which are mainly used in cryptography, are constructed on

[⋆] The first and second authors are supported by a JSPS Fellowship for Young Scientists.

elliptic curve groups. In these constructions, the target group is different from the domain groups.

This leads to the natural question: is it possible to construct a bilinear map where the domain and target groups are identical? Such a bilinear map is called a *self-bilinear map*, and has previously been studied by Cheon and Lee [5]. They showed that a self-bilinear map is useful to construct cryptographic primitives by highlighting that it can be used for constructing a multilinear map [3]. However, in contrast to this useful property, they also proved an impossibility result: the computational Diffie-Hellman (CDH) assumption cannot hold in a group G of *known prime order* if there exists an efficiently computable self-bilinear map on G. This is undesirable for cryptographic applications. The overview of the proof is as follows. Let $e : G \times G \to G$ be a self-bilinear map and g be a generator of G, then we have $e(g^x, g^y) = e(g, g)^{xy} = g^{cxy}$ where c is an integer such that $e(g, g) = g^c$. Then we can compute g^{xy} by computing c-th root of $e(g^x, g^y)$ since G is a prime and known order group.[1] However, their impossibility result cannot be applied for the case that G is a *composite and unknown* order group. This is the setting we focus on in this paper.

1.2 Our Contribution

In this paper, we consider a group of composite and unknown order and construct a *self-bilinear map with auxiliary information* which is a weaker variant of a self-bilinear map, by using indistinguishability obfuscation [10]. Though our self-bilinear map with auxiliary information has a limited functionality compared with a self-bilinear map, we show that it is still useful to construct various cryptographic primitives. Especially, it is sufficient to instantiate some multilinear map-based cryptographic primitives such as multiparty non-interactive key exchange (NIKE), broadcast encryption and attribute-based encryption for circuits. Our multiparty NIKE and distributed broadcast encryption schemes are the first schemes where all algorithms can be run independent of the number of users. We also show that our technique can be used for constructing a somewhat homomorphic encryption scheme for NC^1 circuits.

Applications of our self-bilinear map with auxiliary information. As applications of our self-bilinear map with auxiliary information, we construct a multilinear map. From our construction, we obtain multiparty NIKE, distributed broadcast encryption and ABE for circuits schemes. The details follow.

- **Multilinear map.** We can construct a multilinear map by iterated usage of a self-bilinear map. Since our variant of a self-bilinear map in this paper requires auxiliary information to compute the map, the resulting multilinear map also inherits this property. However, we show that it is sufficient to replace existing multilinear maps in some applications which are given

[1] Here, we consider only the case in which c is known. However, [5] proved that the CDH assumption does not hold even if c is unknown as long as G is a group of known prime order.

below. Moreover, our multilinear map has an interesting property that existing multilinear maps do not have: the level of multilinearity is *not bounded at the instance generation phase* and representations of group elements are *compact*, i.e., their sizes are independent of the level of multilinearity.

- **Multiparty NIKE.** We construct a multiparty NIKE scheme where the maximum number of users is not fixed in the setup phase. In particular, the size of both the public parameters and a public key generated by a user are independent of the number of users. The construction is a natural extension of the Diffie-Hellman key exchange by using our multilinear map [7, 3]. We note that [4] also constructed multiparty NIKE schemes based on indistinguishability obfuscation. However, in their schemes, the setup algorithm or key generation algorithm have to take the number of users as input unlike ours.
- **Distributed broadcast encryption.** Distributed broadcast encryption is broadcast encryption where a user can join the system by himself without the assistance of a (semi) trusted third party holding a master key. We construct a distributed broadcast encryption scheme where the maximum number of users is not fixed in the setup phase based on our multiparty NIKE scheme. In particular, the size of both the public parameters and a ciphertext overhead are independent of the number of users. We note that [4] also constructed a distributed broadcast encryption scheme based on indistinguishability obfuscation. However, in their scheme, the setup algorithm have to take the number of users as input unlike ours.
- **ABE for circuits.** We construct an ABE scheme for general circuits by using our multilinear map. The construction is an analogue of the scheme in [11]. Note that this is *not* the first ABE scheme for general circuits based on indistinguishability obfuscation since a indistinguishability obfuscation implies witness encryption [10], and [12] constructed ABE for circuits based on witness encryption. We also note that Gorbunov et al. [15] constructed attribute based encryption for circuit based on the standard learning with errors (LWE) assumption.

The above results can be interpreted as evidence that our multilinear map can replace existing multilinear maps in some applications based on the multilinear CDH assumption since all of the above constructions are simple analogue of multilinear map-based constructions.

Besides direct applications of our self-bilinear map with auxiliary information, we construct a somewhat homomorphic encryption scheme by using a similar technique. Our somewhat homomorphic encryption scheme is chosen plaintext (CPA) secure, NC^1 homomorphic and compact.

Note that all known candidate constructions of indistinguishability obfuscation are far from practical, and hence, the above constructions are mostly of theoretical interest.

Technical overview. Here, we give a technical overview of our result. Our basic idea is to avoid the impossibility result of self-bilinear maps which is explained above by considering a group of *composite and unknown order*. Note that even if we consider such a group, many decisional assumptions such as the decisional Diffie-Hellman (DDH) assumption cannot hold if there exists an efficiently computable self-bilinear map on the group. Therefore we consider only computational assumptions such as the CDH assumption. For a Blum integer N, we consider the group $\mathbb{QR}_N^+$ of signed quadratic residues [17]. On this group, we consider a self-bilinear map $e : \mathbb{QR}_N^+ \times \mathbb{QR}_N^+ \to \mathbb{QR}_N^+$ which is defined as $e(g^x, g^y) := g^{2xy}$. The reason why we define it in this manner is that we want to ensure that the CDH assumption holds in $\mathbb{QR}_N^+$, even if e is efficiently computable. That is, even if we can compute $e(g^x, g^y) = g^{2xy}$, it is difficult to compute g^{xy} from it since the Rabin function is hard to invert under the factoring assumption. However, given only the group elements g^x and g^y, we do not know how to compute $e(g^x, g^y)$ efficiently. To address this, we introduce *auxiliary information* τ_y for each element $g^y \in \mathbb{QR}_N^+$ which enables us to compute a map $e(\cdot, g^y)$ efficiently. This leads to the notion of *self-bilinear map with auxiliary information* which we introduce in this paper.

The problem is how to define auxiliary information τ_y which enables us to compute $e(\cdot, g^y)$ efficiently. The most direct approach is to define τ_y as a circuit that computes the $2y$-th power. However, if we define τ_y as a "natural" circuit that computes the $2y$-th power, then we can extract $2y$ from τ_y, and thus we can compute y. This clearly enables us to compute g^{xy}, which breaks the CDH assumption.

A more clever way is to define τ_y as a circuit that computes the t_y-th power where $t_y = 2y \pm \mathrm{ord}(\mathbb{QR}_N^+)$.[2] In this way, it seems that τ_y does not reveal y since t_y is a "masked" value of $2y$ by $\mathrm{ord}(\mathbb{QR}_N^+)$ which is an unknown odd number. This idea is already used by Seurin [25] to construct a trapdoor DDH group. Actually, he proved that even if t_y is given in addition to g^x and g^y, it is still difficult to compute g^{xy}. In this way, it seems that we can construct a self-bilinear map with auxiliary information. However, this creates a problem: we do not have an efficient algorithm to compute t_y from y without knowing the factorization of N. If such an algorithm does not exist, then we cannot instantiate many bilinear map-based primitives using the resulting map such as the 3-party Diffie-Hellman key exchange [19].

To overcome the above difficulty, we use *indistinguishability obfuscation*. An indistinguishability obfuscator ($i\mathcal{O}$) is an efficient randomized algorithm that makes circuits C_0 and C_1 computationally indistinguishable if they have exactly the same functionality.

We observe that a circuit that computes the $2y$-th power and a circuit that computes the t_y-th power for an element of $\mathbb{QR}_N^+$ have exactly the same functionality since we have $t_y = 2y \pm \mathrm{ord}(\mathbb{QR}_N^+)$. Therefore if we obfuscate these circuits by $i\mathcal{O}$, then the resulting circuits are computationally indistinguishable. Then we define auxiliary information τ_y as an obfuscation of a circuit that computes the $2y$-th power. With this definition, it is clear that τ_y can be computed from y efficiently, and the above mentioned problem is solved. Moreover, τ_y is compu-

[2] In the definition of t_y, whether $+$ or $-$ is used depends on y. See [25] for more details.

tationally indistinguishable from an obfuscation of a circuit that computes the t_y-th power. Therefore it must still be difficult to compute g^{xy} even if τ_y is given in addition to g^x and g^y.

Thus we obtain a self-bilinear map with auxiliary information on $\mathbb{QR}_N^+$ while ensuring that the auxiliary information does not allow the CDH assumption to be broken. Moreover, by extending this, we can prove that an analogue of multilinear CDH assumption holds with respect to a multilinear map which is constructed from our self-bilinear map with auxiliary information based on $i\mathcal{O}$ and the factoring assumption.

1.3 Related Work

In cryptography, bilinear maps on elliptic curves were first used for breaking the discrete logarithm problem on certain curves [21]. The first constructive cryptographic applications of a bilinear map are given in [19, 24, 2]. Since then, many constructions of cryptographic primitives based on a bilinear map have been proposed.

Boneh and Silverberg [3] considered a multilinear map which is an extension of a bilinear map, and showed its usefulness for constructing cryptographic primitives though they did not give a concrete construction of multilinear maps. Garg et al. [8] proposed a candidate construction of multilinear maps based on ideal lattices. Coron et al. [6] proposed another construction over the integers.

The notion of indistinguishability obfuscation was first proposed by Barak et al. [1]. The first candidate construction of indistinguishability obfuscation was proposed by Garg et al. [10]. Since then, many applications of indistinguishability obfuscation have been proposed [23, 9, 18, 14, 4].

2 Preliminaries

2.1 Notations

We use $\mathbb{N}$ to denote the set of all natural numbers, and $[n]$ to denote the set $\{1, \dots n\}$ for $n \in \mathbb{N}$. If S is a finite set, then we use $x \overset{\$}{\leftarrow} S$ to denote that x is chosen uniformly at random from S. If $\mathcal{A}$ is an algorithm, we use $x \leftarrow \mathcal{A}(y; r)$ to denote that x is output by $\mathcal{A}$ whose input is y and randomness is r. We often omit r. We say that a function $f(\cdot) : \mathbb{N} \to [0, 1]$ is negligible if for all positive polynomials $p(\cdot)$ and all sufficiently large $\lambda \in \mathbb{N}$, we have $f(\lambda) < 1/p(\lambda)$. We say f is overwhelming if $1 - f$ is negligible. We say that an algorithm $\mathcal{A}$ is efficient if there exists a polynomial p such that the running time of $\mathcal{A}$ with input length λ is less than $p(\lambda)$. For two integers $x \neq 0$ and y, we say that x and y are negligibly close if $|x - y|/x$ is negligible. For a set S and a random variable x over S, we say that x is almost random on S if the statistical distance between the distribution of x and the uniform distribution on S is negligible. For a circuit C, we denote the size of C by $|C|$. For a wire w which is an output wire of a gate, we denote the first input incoming wire of the gate by $A(w)$ and the second incoming wire of the gate by $B(w)$. We use λ to denote the security parameter.

2.2 Indistinguishability Obfuscator

Here, we recall the definition of an indistinguishability obfuscator [10, 23].

Definition 1 *(Indistinguishability Obfuscator.) Let C_λ be the class of circuits of size at most λ. An efficient randomized algorithm $i\mathcal{O}$ is called an indistinguishability obfuscator for P/poly if the following conditions are satisfied:*

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

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

- *For any (not necessarily uniform) efficient algorithm $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, there exists a negligible function α such that the following holds: if $\mathcal{A}_1(1^\lambda)$ always outputs (C_0, C_1, σ) such that we have $C_0, C_1 \in \mathcal{C}_\lambda$ and $\forall x\ C_0(x) = C_1(x)$, then we have*

$$|\Pr[\mathcal{A}_2(\sigma, i\mathcal{O}(\lambda, C_0)) = 1 : (C_0, C_1, \sigma) \leftarrow \mathcal{A}_1(1^\lambda)]$$
$$- \Pr[\mathcal{A}_2(\sigma, i\mathcal{O}(\lambda, C_1)) = 1 : (C_0, C_1, \sigma) \leftarrow \mathcal{A}_1(1^\lambda)]| \leq \alpha(\lambda)$$

Note that a candidate construction of $i\mathcal{O}$ that satisfies the above definition is given in [10].

2.3 Group of Signed Quadratic Residues

Here, we recall the definition and some properties of a group of signed quadratic residues [17] that we mainly work with in this paper. An integer $N = PQ$ is called a Blum integer if P and Q are distinct primes and $P \equiv Q \equiv 3 \bmod 4$ holds. Let $\mathsf{RSAGen}(1^\lambda)$ be an efficient algorithm which outputs a random $\ell_N(\lambda)$-bit Blum integer $N = PQ$ and its factorization (P, Q) so that the length of P and Q are the same and we have $\gcd(P-1, Q-1) = 1$. For simplicity, we often omit λ and simply denote $\ell_N(\lambda)$ as ℓ_N. We say that the factoring assumption holds with respect to RSAGen if for any efficient adversary $\mathcal{A}$, $\Pr[x \in \{P, Q\} : (N, P, Q) \leftarrow \mathsf{RSAGen}(1^\lambda), x \leftarrow \mathcal{A}(1^\lambda, N)]$ is negligible. We define the group of quadratic residues as $\mathbb{QR}_N := \{u^2 : u \in {\mathbb{Z}_N}^*\}$. Note that $\mathbb{QR}_N$ is a cyclic group of order $(P-1)(Q-1)/4$ if N is output by $\mathsf{RSAGen}(1^\lambda)$.

For any subgroup $H \in \mathbb{Z}_N^*$, we define its signed group as $H^+ := \{|x| : x \in H\}$ where $|x|$ is the absolute value of x when it is represented as an element of $\{-(N-1)/2, \ldots, (N-1)/2\}$. This is certainly a group by defining a multiplication as $x \circ y := |(xy \bmod N)|$ for $x, y \in H^+$. For simplicity, we often denote multiplications on H^+ as usual multiplication when it is clear that we are considering a signed group. If H is a subgroup of $\mathbb{QR}_N$, then $H \cong H^+$ by the natural projection since $-1 \notin \mathbb{QR}_N$. In particular, $\mathbb{QR}_N^+$ is a cyclic group of order $(P-1)(Q-1)/4$. We call $\mathbb{QR}_N^+$ a group of signed quadratic residues. A remarkable property of $\mathbb{QR}_N^+$ is that it is efficiently recognizable. That is, there exists an efficient algorithm that determines whether a given string is an element of $\mathbb{QR}_N^+$ or not [17].

3 Self-bilinear Maps

In this section, we recall the definition of a self-bilinear map [5]. Next, we introduce the notion of *self-bilinear map with auxiliary information* which is a weaker variant of a self-bilinear map. Finally we define hardness assumptions with respect to a multilinear map which is constructed from a self-bilinear map.

3.1 Definition of a Self-bilinear Map

First, we recall the definition of a self-bilinear map. A self-bilinear map is a bilinear map where the domain and target groups are identical. The formal definition is as follows.

Definition 2 *(Self-bilinear Map [5]) For a cyclic group G, a self-bilinear map $e : G \times G \to G$ has the following properties.*

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

$$e(g_1^\alpha, g_2) = e(g_1, g_2^\alpha) = e(g_1, g_2)^\alpha.$$

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

In addition to the above, we usually require that e is efficiently computable. As shown in [5], we can construct an n-multilinear map for any integer $n \geq 2$ from a self-bilinear map e. This can be seen by easy induction: suppose that an n-multilinear map e_n can be constructed from a self-bilinear map e, then we can construct an $(n+1)$-multilinear map e_{n+1} by defining

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

3.2 Self-bilinear Map with Auxiliary Information

Instead of constructing a self-bilinear map, we construct a *self-bilinear map with auxiliary information* which is a weaker variant of a self-bilinear map. In a self-bilinear map with auxiliary information, the map is efficiently computable only if "auxiliary information" is given. That is, when we compute $e(g^x, g^y)$, we require auxiliary information τ_x for g^x or τ_y for g^y. This is the difference from an"ideal" self-bilinear map in which $e(g^x, g^y)$ can be computed only from g^x and g^y. We formalize a self-bilinear map with auxiliary information as a set of algorithms $\mathcal{SBP} = (\mathsf{InstGen}, \mathsf{Sample}, \mathsf{AIGen}, \mathsf{Map}, \mathsf{AIMult})$ and a set R of integers.

$\mathsf{InstGen}(1^\lambda) \to \mathsf{params}$: $\mathsf{InstGen}$ takes the security parameter 1^λ as input and outputs the public parameters params which specifies an efficiently recognizable cyclic group G on which the group operation is efficiently computable. We require that an approximation $\mathrm{Approx}(G)$ of $\mathrm{ord}(G)$ can be computed efficiently from params and that $\mathrm{Approx}(G)$ is negligibly close to $\mathrm{ord}(G)$. Additionally, params specifies sets T_x^ℓ of auxiliary information for all integers x and $\ell \in \mathbb{N}$.

$\mathsf{Sample}(\mathsf{params}) \to g$: Sample takes params as input and outputs an almost random element g of G. The self-bilinear map $e : G \times G \to G$ is defined with respect to the element g.

$\mathsf{AIGen}(\mathsf{params}, \ell, x) \to \tau_x$: AIGen takes params, level ℓ and an integer $x \in R$ as input, and outputs corresponding auxiliary information $\tau_x \in T_x^{\ell}$.

$\mathsf{Map}(\mathsf{params}, g^x, \tau_y) \to e(g^x, g^y)$: Map takes params, $g^x \in G$ and $\tau_y \in \cup_{\ell \in \mathbb{N}} T_y^{\ell}$ as input and outputs $e(g^x, g^y)$. By using this algorithm iteratively, we can compute $e_n(g_1^{x_1}, \dots, g_n^{x_n})$ if we are given $g^{x_1}, \dots, g_n^{x_n}$ and $\tau_{x_1}, \dots, \tau_{x_n}$. (Note that not all of these elements are required to evaluate the map.)

$\mathsf{AIMult}(\mathsf{params}, \ell, \tau_x, \tau_y) \to \tau_{x+y}$: AIMult takes params, ℓ, $\tau_x \in T_x^{\ell_1}$, $\tau_y \in T_y^{\ell_2}$ such that $\ell > \max\{\ell_1, \ell_2\}$ as input and outputs $\tau_{x+y} \in T_{x+y}^{\ell}$.

In addition to the above algorithms, we require for $\mathcal{SBP}$ to satisfy the following property.

Indistinguishability of auxiliary information. We require that any efficient algorithm which is given auxiliary information cannot tell whether it is generated by AIGen or AIMult. More formally, for any $\mathsf{params} \leftarrow \mathsf{InstGen}(1^{\lambda})$, $\ell \in \mathbb{N}$ (which does not depend on λ), natural numbers $\ell_1, \ell_2 < \ell$, integers x, y and z (which are polynomially bounded in λ), such that $z \in R$ and $z \equiv x + y \bmod \mathrm{ord}(G)$, and auxiliary information $\tau_x \in T_x^{\ell_1}$ and $\tau_y \in T_y^{\ell_2}$, the following two distributions are computationally indistinguishable:

$$\mathcal{D}_1 = \{\tau_z : \tau_z \leftarrow \mathsf{AIGen}(\mathsf{params}, \ell, z)\}$$

$$\mathcal{D}_2 = \{\tau_{x+y} : \tau_{x+y} \leftarrow \mathsf{AIMult}(\mathsf{params}, \ell, \tau_x, \tau_y)\}.$$

Remark 1 *A level ℓ of auxiliary information grows by at least* 1 *when* AIMult *is applied. One can think of it as an analogue of a noise level in the GGH graded encoding [11]. In our construction, the size of auxiliary information grows exponentially in a level ℓ. Therefore an efficient algorithm can only handle auxiliary information of a constant level. Actually, in our applications in this paper, ℓ is set at most* 2.

3.3 Hardness Assumptions

For cryptographic use, we introduce some hardness assumptions. We use $\mathcal{SBP}$ to construct a multilinear map, and thus our hardness assumptions are associated with a multilinear map which is constructed from $\mathcal{SBP}$. In the following, we let $\mathcal{SBP} = (\mathsf{InstGen}, \mathsf{Sample}, \mathsf{AIGen}, \mathsf{Map}, \mathsf{AIMult})$ be self-bilinear map procedures. First, we define the multilinear computational Diffie-Hellman with auxiliary information (MCDHAI) assumption which is an analogue of the multilinear computational Diffie-Hellman (MCDH) assumption.

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

$$\Pr[e_n(g, \dots, g)^{s\Pi_{i=1}^{n} x_i} \leftarrow \mathcal{A}(\mathsf{params}, g, g^s, g^{x_1}, \dots, g^{x_n}, \tau_s, \tau_{x_1} \dots, \tau_{x_n})]$$

is negligible, where $\mathsf{params} \leftarrow \mathsf{InstGen}(1^\lambda)$*,* $g \leftarrow \mathsf{Sample}(\mathsf{params})$*,* $s, x_1, \ldots, x_n \leftarrow [\mathrm{Approx}(G)]$*,* $\tau_s \leftarrow \mathsf{AIGen}(\mathsf{params}, 1, s)$*,* $\tau_{x_i} \leftarrow \mathsf{AIGen}(\mathsf{params}, 1, x_i)$ *for all* $i \in [n]$*.*

We say that the MCDHAI assumption holds with respect to $\mathcal{SBP}$ *if the* n*-MCDHAI assumption holds with respect to* $\mathcal{SBP}$ *for any integer* n *which is polynomially bounded in* λ*.*

We also define the multilinear hashed Diffie-Hellman with auxiliary information (MHDHAI) assumption which is an analogue of the multilinear hashed Diffie-Hellman (MHDH) assumption.

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

$$|\Pr[1 \leftarrow \mathcal{D}(\mathsf{params}, g, g^s, g^{x_1}, \ldots, g^{x_n}, \tau_s, \tau_{x_1} \ldots, \tau_{x_n}, H, T)|\beta = 1] \\ - \Pr[1 \leftarrow \mathcal{D}(\mathsf{params}, g, g^s, g^{x_1}, \ldots, g^{x_n}, \tau_s, \tau_{x_1} \ldots, \tau_{x_n}, H, T)|\beta = 0]|$$

is negligible, where $\mathsf{params} \leftarrow \mathsf{InstGen}(1^\lambda)$*,* $g \leftarrow \mathsf{Sample}(\mathsf{params})$*,* $s, x_1, \ldots, x_n \leftarrow [\mathrm{Approx}(G)]$*,* $\tau_s \leftarrow \mathsf{AIGen}(\mathsf{params}, 1, s)$*,* $\tau_{x_i} \leftarrow \mathsf{AIGen}(\mathsf{params}, 1, x_i)$ *for all* $i \in [n]$*,* $\beta \xleftarrow{\$} \{0,1\}$ *and* $T \xleftarrow{\$} \{0,1\}^k$ *if* $\beta = 0$*, and otherwise* $T = H(e_n(g, \ldots, g)^{s\Pi_{i=1}^n x_i})$*.*

We say that the MHDHAI assumption holds with respect to $\mathcal{SBP}$ *and* $\mathcal{H}$ *if the* n*-MHDHAI assumption holds with respect to* $\mathcal{SBP}$ *and* $\mathcal{H}$ *for any integer* n *which is polynomially bounded in* λ*.*

Note that 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 [13].

4 Our Construction of a Self-bilinear Map

In this section, we construct a self-bilinear map with auxiliary information by giving a construction of self-bilinear map procedures $\mathcal{SBP}$. We prove that the MCDHAI assumption holds with respect to $\mathcal{SBP}$ if the factoring assumption holds and there exists an indistinguishability obfuscator for $P/poly$.

4.1 Construction

First we prepare some notations for circuits on $\mathbb{QR}_N^+$.

Notation for circuits on $\mathbb{QR}_N^+$**.** In the following, for an ℓ_N-bit RSA modulus N and an integer $x \in \mathbb{Z}$, $\mathcal{C}_{N,x}$ denotes a set of circuits $C_{N,x}$ that work as follows. For input $y \in \{0,1\}^{\ell_N}$, $C_{N,x}$ interprets y as an element of $\mathbb{Z}_N$ and returns y^x where the exponentiation is done on $\mathbb{QR}_N^+$ if $y \in \mathbb{QR}_N^+$ and otherwise returns 0^{ℓ_N} (which is interpreted as $\perp$). We define the canonical circuit $\tilde{C}_{N,x}$ in $\mathcal{C}_{N,x}$ in a natural way [3]. For circuits C_1, C_2 whose output can be interpreted as elements

[3] There is flexibility to define the canonical circuit. However, any definition works if the size of $\tilde{C}_{N,x}$ is polynomially bounded in λ and $|x|$.

of $\mathbb{QR}_N^+$, $\mathsf{Mult}(C_1, C_2)$ denotes a circuit that computes $C_{\mathsf{mult}}(C_1(x), C_2(y))$ for input (x, y) where C_{mult} is a circuit that computes a multiplication for elements of $\mathbb{QR}_N^+$. If an input of C_{mult} is not a pair of two elements in $\mathbb{QR}_N^+$, then it outputs 0^{ℓ_N}.

Now we are ready to describe our construction. The construction of $\mathcal{SBP}$ is as follows.

$\mathsf{InstGen}(1^\lambda) \to \mathsf{params}$: Run $\mathsf{RSAGen}(1^\lambda)$ to obtain (N, P, Q), and outputs $\mathsf{params} = N$. params defines the underlying group $G = \mathbb{QR}_N^+$ and $\mathrm{Approx}(G) = (N-1)/4$. For an integer x and $\ell \in \mathbb{N}$, the set T_x^ℓ is defined as $T_x^\ell = \{i\mathcal{O}(M_\ell, C_{N,2x}; r) : C_{N,2x} \in \mathcal{C}_{N,2x}$ such that $|C_{N,2x}| \leq M_\ell, r \in \{0,1\}^*\}$, where M_ℓ is defined later.

$\mathsf{Sample}(\mathsf{params}) \to g$: Choose a random element $g \in \mathbb{Z}_N^*$, computes g^2 in $\mathbb{Z}_N^*$ and outputs $|g^2|$ where the absolute value is taken when it is represented as an element of $\{-(N-1)/2, \ldots, (N-1)/2\}$. When $\mathsf{params} = N$ and a generator $g \in \mathbb{QR}_N^+$ are fixed, the self-bilinear map e is defined as $e(g^x, g^y) = g^{2xy}$.

$\mathsf{AIGen}(\mathsf{params}, \ell, x) \to \tau_x$: Define the range of x as $R := [(N-1)/2]$. Take the canonical circuit $\tilde{C}_{N,2x} \in \mathcal{C}_{N,2x}$, set $\tau_x \leftarrow i\mathcal{O}(M_\ell, \tilde{C}_{N,2x})$ and output τ_x.

$\mathsf{Map}(\mathsf{params}, g^x, \tau_y) \to e(g^x, g^y)$: Compute $\tau_y(g^x)$ and output it. (Recall that τ_y is a circuit that computes the $2y$-th power for an element of $\mathbb{QR}_N^+$.)

$\mathsf{AIMult}(\mathsf{params}, \ell, \tau_x, \tau_y) \to \tau_{x+y}$: Compute $\tau_{x+y} \leftarrow i\mathcal{O}(M_\ell, \mathsf{Mult}(\tau_x, \tau_y))$ and output it.

Definition of M_ℓ. M_ℓ represents an upper bound of the size of a circuit which is obfuscated by $i\mathcal{O}$ when auxiliary information with level ℓ is generated. To define it, we consider another integer M'_ℓ which represents an upper bound of the size of auxiliary information with level ℓ. We define M_ℓ and M'_ℓ recursively. We define M'_0 as an integer which is larger than $\max_{x \in [(N+1)/2]}\{|\tilde{C}_{N,x}|\}$. For $\ell \geq 1$, we define $M_\ell := 2M'_{\ell-1} + |C_{\mathsf{Mult}}|$ and $M'_\ell := poly(M_\ell, \lambda)$ where $poly$ is a polynomial that satisfies $|i\mathcal{O}(M, C)| < poly(M, \lambda)$ for any integer M and circuit C such that $|C| < M$.

Indistinguishability of auxiliary information. If $z \equiv x + y \bmod \mathrm{ord}(\mathbb{QR}_N^+)$ holds, then $C_{N,2z}$ and $\mathsf{Mult}(\tau_x, \tau_y)$ have exactly the same functionality. Therefore if we obfuscate these circuits by $i\mathcal{O}$, then the resulting circuits are computationally indistinguishable.

4.2 Hardness Assumptions

We prove that the MCDHAI assumption holds with respect to our construction of a self-bilinear map if $i\mathcal{O}$ is an indistinguishability obfuscator for $P/poly$ and the factoring assumption holds. From that, we can immediately see that the MHDHAI assumption also holds with respect to our construction if we use the Goldreich-Levin hardcore bit function [13] as $\mathcal{H}$.

First, we prove that the MCDHAI assumption holds if $i\mathcal{O}$ is an indistinguishability obfuscator for $P/poly$ and the factoring assumption holds.

Theorem 1 *The MCDHAI assumption holds with respect to $\mathcal{SBP}_{\mathsf{Ours}}$ if the factoring assumption holds with respect to* RSAGen *and $i\mathcal{O}$ is an indistinguishability obfuscator for $P/poly$.*

Proof. For an algorithm $\mathcal{A}$ and an integer n (which is polynomially bounded by the security parameter), we consider the following games.

Game **1.** This game is the original n-MCDHAI game. More precisely, it is as follows.

$(N, P, Q) \leftarrow \mathsf{RSAGen}(1^\lambda)$
$g \xleftarrow{\$} \mathbb{QR}_N^+$
$s, x_1, \ldots, x_n \xleftarrow{\$} [(N-1)/4]$
$\tau_s \leftarrow i\mathcal{O}(M_1, \tilde{C}_{N,2s}), \tau_{x_i} \leftarrow i\mathcal{O}(M_1, \tilde{C}_{N,2x_i})$ for $i \in [n]$
$U \leftarrow \mathcal{A}(N, g, g^s, g^{x_1}, \ldots, g^{x_n}, \tau_s, \tau_{x_1} \ldots, \tau_{x_n})$

Game $1'$ This game is the same as Game 1 except that $s, x_1, \ldots, x_n$ are chosen from $[\mathrm{ord}(\mathbb{QR}_N^+)]$.

Game $2'$ This game is the same as Game $1'$ except that $g, s, x_1, \ldots, x_n, \tau_s, \tau_{x_1}, \ldots, \tau_{x_n}$ are set differently. More precisely, it is as follows.

$(N, P, Q) \leftarrow \mathsf{RSAGen}(1^\lambda)$
$h \xleftarrow{\$} \mathbb{QR}_N^+$
$g := h^2$
$s', x'_1, \ldots, x'_n \xleftarrow{\$} [\mathrm{ord}(\mathbb{QR}_N^+)]$
$g^s := g^{s'} h$, $g^{x_i} := g^{x'_i} h$ for $i \in [n]$
(This implicitly defines $s \equiv s' + 1/2 \bmod \mathrm{ord}(\mathbb{QR}_N^+)$ and $x_i \equiv x'_i + 1/2 \bmod \mathrm{ord}(\mathbb{QR}_N^+)$).
$\tau_s \leftarrow i\mathcal{O}(M_1, \tilde{C}_{N,2s'+1}), \tau_{x_i} \leftarrow i\mathcal{O}(M_1, \tilde{C}_{N,2x'_i+1})$ for $i \in [n]$
$U \leftarrow \mathcal{A}(N, g, g^s, g^{x_1}, \ldots, g^{x_n}, \tau_s, \tau_{x_1} \ldots, \tau_{x_n})$

Game **2** This game is the same as Game $2'$ except that $s, x_1, \ldots, x_n$ are chosen from $[(N-1)/4]$.

We say that $\mathcal{A}$ wins if it outputs $U = e_n(g, \ldots, g)^{s\Pi_{i=1}^n x_i}$. For $i = 1, 2$, we let T_i and T'_i be the events that $\mathcal{A}$ wins in Game i and Game i', respectively. What we want to prove is that $\Pr[T_1]$ is negligible. We prove it by the following lemmas.

Lemma 1 $|\Pr[T_i] - \Pr[T'_i]|$ *is negligible for* $i = 1, 2$

Proof. This follows since $(N-1)/4$ is negligibly close to $\mathrm{ord}(\mathbb{QR}_N^+)$.

Lemma 2 $|\Pr[T'_1] - \Pr[T'_2]|$ *is negligible if $i\mathcal{O}$ is an indistinguishability obfuscator for $P/poly$.*

Proof. We consider hybrid games $H_0, \ldots H_{n+1}$. A hybrid game H_i is the same as Game $1'$ except that the first i auxiliary information (i.e, $\tau_s, \tau_{x_1}, \ldots, \tau_{x_{i-1}}$) are generated as in Game $2'$. It is clear that H_0 is identical to Game $1'$ and

H_{n+1} is identical to $\mathsf{Game}\ 2'$. Let S_i be the event that $\mathcal{A}$ wins in $\mathsf{Game}\ H_i$. It suffices to show that $|\Pr[S_i] - \Pr[S_{i-1}]|$ is negligible by the standard hybrid argument. We construct an algorithm $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ that breaks the security of $i\mathcal{O}$ for the security parameter M_1 by using $\mathcal{A}$ that distinguishes H_i and H_{i-1}. In the following, we use x_0 to mean s for notational convenience.

$\mathcal{B}_1(1^\lambda)$: $\mathcal{B}_1$ runs $(N, P, Q) \leftarrow \mathsf{RSAGen}(1^\lambda)$, chooses $h \xleftarrow{\$} \mathbb{QR}_N^+$ and $x_0, \dots, x_n \xleftarrow{\$} [\mathrm{ord}(\mathbb{QR}_N^+)]$ and sets $g := h^2$. $\mathcal{B}_1$ computes $x'_0, \dots, x'_n \in \mathrm{ord}(\mathbb{QR}_N^+)$ such that $x_j \equiv x'_j + 1/2 \bmod \mathrm{ord}(\mathbb{QR}_N^+)$ for $j = 0, \dots, n$. (This can be computed since $\mathcal{B}_1$ knows the factorization of N.) Then $\mathcal{B}_1$ sets $C_0 := \tilde{C}_{N,2x_{i-1}}$, $C_1 := \tilde{C}_{N,2x'_{i-1}+1}$ and $\sigma := (N, P, Q, h, g, x_0, \dots, x_n, x'_0, \dots, x'_n)$ and outputs (C_0, C_1, σ).

$\mathcal{B}_2(\sigma, C^*)$: $\mathcal{B}_2$ sets

$$\tau_{x_j} \leftarrow \begin{cases} i\mathcal{O}(M_1, \tilde{C}_{N,2x'_j+1}) & \text{if } j = 0, \dots, i-2 \\ C^* & \text{if } j = i-1 \\ i\mathcal{O}(M_1, \tilde{C}_{N,2x_j}) & \text{if } j = i, \dots, n. \end{cases}$$

Then $\mathcal{B}_2$ runs $\mathcal{A}(N, g, g^{x_0}, \dots, g^{x_n}, \tau_{x_0}, \dots, \tau_{x_n})$ to obtain U. If we have $U = e_n(g, \dots, g)^{\Pi_{i=0}^n x_i}$, then $\mathcal{B}_2$ outputs 1, and otherwise outputs 0.

The above completes the description of $\mathcal{B}$. First, we note that each of g_j ($j = 0, \dots, n$) is distributed in $\mathbb{QR}_N^+$ independently of each other in all hybrid games H_i for $i = 0, \dots, n+1$. Therefore $\mathcal{B}$ generates them in exactly the same way as those are generated in the hybrids H_{i-1} and H_i. Then we can see that $\mathcal{B}$ perfectly simulates H_{i-1} if $C^* \leftarrow i\mathcal{O}(M_1, C_0)$ and H_i if $C^* \leftarrow i\mathcal{O}(M_1, C_1)$ from the view of $\mathcal{A}$. If the difference between the probability that $\mathcal{A}$ wins in H_{i-1} and that in H_i is non-negligible, then $\mathcal{B}$ succeeds in distinguish whether C^* is computed as $C^* \leftarrow i\mathcal{O}(M_1, C_0)$ or $C^* \leftarrow i\mathcal{O}(M_1, C_1)$, with non-negligible advantage, and thus breaks the security of $i\mathcal{O}$.

Lemma 3 $\Pr[T_2]$ *is negligible if the factoring assumption holds.*

Proof. Assuming that $\mathcal{A}$ wins in $\mathsf{Game}\ 2$ with non-negligible probability, we construct an algorithm $\mathcal{B}$ that computes $h^{1/2}$ given an RSA modulus N and a random element $h \in \mathbb{QR}_N^+$ with non-negligible probability. This yields the factoring algorithm [17]. The construction of $\mathcal{B}$ is as follows.

$\mathcal{B}(N, h)$: $\mathcal{B}$ sets $g := h^2$ and chooses $s', x'_1, \dots, x'_n \xleftarrow{\$} [(N-1)/4]$. Then $\mathcal{B}$ sets $g^s := g^{s'}h$, $g^{x_i} := g^{x'_i}h$ for all $i \in [n]$, $\tau_s \leftarrow i\mathcal{O}(M_1, \tilde{C}_{N,2s'+1})$ and $\tau_{x_i} \leftarrow i\mathcal{O}(M_1, \tilde{C}_{N,2x'_i+1})$ for all $i \in [n]$. Then $\mathcal{B}$ runs $\mathcal{A}(N, g, g^s, g^{x_1}, \dots, g^{x_n}, \tau_s, \tau_{x_1} \dots, \tau_{x_n})$. Let U be the output of $\mathcal{A}$. Then $\mathcal{B}$ computes $X := \Pi_{i=1}^n (2x'_i + 1)$ and outputs $Ug^{-(s'X + (X-1)/2)}$. (Note that X is odd and therefore $(X-1)/2$ is an integer.)

Since $\mathcal{B}$ perfectly simulates $\mathsf{Game}\ 2$ from the view of $\mathcal{A}$, $\mathcal{A}$ outputs $e_n(g, \dots, g)^{s\Pi_{i=1}^n x_i}$ with non-negligible probability. If it occurs, then we have

$$\begin{aligned} U = e_n(g, \dots, g)^{s\Pi_{i=1}^n x_i} &= g^{2^{n-1} s\Pi_{i=1}^n x_i} = h^{2^n s\Pi_{i=1}^n x_i} = h^{s\Pi_{i=1}^n 2x_i} \\ &= h^{(s'+1/2)\Pi_{i=1}^n (2x'_i+1)} = h^{s'X + X/2} = h^{s'X + (X-1)/2 + 1/2} \end{aligned}$$

and therefore we have $Ug^{-(s'X+(X-1)/2)} = h^{1/2}$.

Theorem 1 is proven by the above lemmas. □

The following is immediate from Theorem 1 and the Goldreich-Levin theorem.

Theorem 2 *The MHDHAI assumption holds with respect to $\mathcal{SBP}_{\mathsf{Ours}}$ and the Goldreich-Levin hardcore bit function if the factoring assumption holds with respect to* RSAGen *and $i\mathcal{O}$ is an indistinguishability obfuscator for P/poly.*

5 Applications of Our Self-bilinear Map

In Sec. 4, we constructed a self-bilinear map with auxiliary information. In this section, we construct a multilinear map, multiparty NIKE, distributed broadcast encryption and ABE for circuits by using it.

Multilinear map. Here, we consider a multilinear map which is constructed from a self-bilinear map with auxiliary information. As shown in Sec. 3.1 we can construct a multilinear map by iterated usage of a self-bilinear map. However, if we use a self-bilinear map with auxiliary information as a building block, then the resulting multilinear map has a restricted functionality: we need auxiliary information to compute the map. The concrete formulation is as follows.

Similarly to self-bilinear map procedures in Sec. 3.2, we formalize a multilinear map with auxiliary information as a set of algorithms $\mathcal{SBP} = (\mathsf{InstGen}_{\mathsf{mult}}, \mathsf{Sample}_{\mathsf{mult}}, \mathsf{AIGen}_{\mathsf{mult}}, \mathsf{Map}_{\mathsf{mult}}, \mathsf{AIMult}_{\mathsf{mult}})$ and a set R of integers. $\mathsf{InstGen}_{\mathsf{mult}}$ takes the security parameter as input and outputs the public parameters params which specify an underlying group G and a multilinear map e on it. $\mathsf{Sample}_{\mathsf{mult}}$ takes params as input and outputs an almost random element of G. $\mathsf{AIGen}_{\mathsf{mult}}$ takes params, ℓ, and $x \in R$ as input and outputs auxiliary information τ_x of level ℓ with respect to x. $\mathsf{Map}_{\mathsf{mult}}$ takes params, $g^{x_1}, \ldots, g^{x_n}$, $\tau_{x_1}, \ldots, \tau_{x_{n-1}}$ and a level of multilinearity n as input, and outputs $e_n(g^{x_1}, \ldots, g^{x_n})$. $\mathsf{AIMult}_{\mathsf{mult}}$ takes params, an integer ℓ and auxiliary information τ_x and τ_y whose levels are less than ℓ as input and outputs auxiliary information τ_{x+y} of level ℓ with respect to $x + y$. A more precise definition is given in the full version.

In spite of the limitation that it requires auxiliary information to compute the map, a multilinear map with auxiliary information is sufficient to replace existing multilinear maps in some applications. Moreover, our multilinear map has interesting properties that existing multilinear maps do not have: the level of multilinearity is not bounded at the instance generation phase and representations of group elements are compact, i.e., their sizes are independent of the level of multilinearity. By this property, cryptographic primitives which are constructed from our multilinear map inherit these properties too.

Multiparty NIKE. By extending the Diffie-Hellman key exchange [7] to a multilinear setting as in [3], we obtain a multiparty NIKE scheme. By using our multilinear map (with auxiliary information) as a building block, we obtain a multiparty NIKE scheme where the maximum number of users is not fixed in the setup phase. In particular, the size of both the public parameters and a public key generated by a user are independent of the number of users. Note that [4] also constructed multiparty NIKE schemes based on indistinguishability obfuscation. However, in their schemes, the setup algorithm or key generation algorithm have to take the number of users as input unlike ours.

Distributed broadcast encryption. It is known that a multiparty NIKE scheme can be converted to a *distributed broadcast encryption* [3, 4], where a user can join the system by himself without the assistance of a (semi) trusted third party holding a master key. The conversion is very simple: The setup algorithm runs $\mathsf{Setup}_{\mathsf{NIKE}}(1^\lambda)$ to obtain PP and publishes it. A user who wants to join the system runs $\mathsf{Publish}_{\mathsf{NIKE}}(\mathsf{PP})$ to obtain (pk, sk), publishes pk as his public key and keeps sk as his secret key. A sender who wants to send a message M to a set S of users plays the role of a user of the underlying NIKE, shares a derived key K with users in S and encrypts M to obtain a ciphertext Ψ by a symmetric key encryption scheme using the key K. A ciphertext consists of S, the sender's public key and Ψ. It is easy to prove that the resulting broadcast encryption scheme is CPA secure if the underlying multiparty NIKE scheme is statically secure. In our scheme, as in the multiparty NIKE scheme, all algorithms can be run independently of the number of users. In particular, the size of both the public parameters and a ciphertext overhead are independent of the number of users. This is the first distributed broadcast encryption scheme with this property. Note that [4] also constructed distributed broadcast encryption schemes based on indistinguishability obfuscation. However, in their schemes, the setup algorithm or key generation algorithm have to take the number of users as input unlike ours.

Attribute based encryption for circuits. We can construct ABE for circuits based on our self-bilinear map almost similarly to the scheme in [11]. The concrete construction can be found in the full version.

6 Homomorphic Encryption

In this section, we construct a somewhat homomorphic encryption scheme by using an indistinguishability obfuscator. This is not a direct application of our self-bilinear map. However, the idea behind the construction is similar.

6.1 Definition of Homomorphic Encryption

Here, we recall some definitions for homomorphic encryption. A homomorphic encryption scheme HE consists of the four algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$. KeyGen takes the security parameter 1^λ as input and outputs a public key pk and a secret key sk. Enc takes a public key pk and a massage $m \in \{0, 1\}$ as input, and outputs a ciphertext c. Eval takes a public key pk, a circuit f with input length

ℓ and a set of ℓ ciphertexts $c_1, \dots, c_\ell$ as input, and outputs a ciphertext c_f. Dec takes a secret key sk and a ciphertext c as input, and outputs a message m. For correctness of the scheme, we require that for all $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and all $m \in \{0, 1\}$, we have $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m$ with overwhelming probability.

Next, we define some properties of homomorphic encryption such as the CPA security, $\mathcal{C}$-homomorphism, and compactness.

Definition 5 *(CPA security) We say that a scheme* HE *is CPA secure if for any efficient adversary* $\mathcal{A}$*,*

$$|\Pr[1 \leftarrow \mathcal{A}(pk, \mathsf{Enc}(pk, 0))] - \Pr[1 \leftarrow \mathcal{A}(pk, \mathsf{Enc}(pk, 1))]|$$

is negligible, where $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$*.*

Definition 6 *($\mathcal{C}$-homomorphism) Let* $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ *be a class of circuits. A scheme* HE *is* $\mathcal{C}$*-homomorphic if for any family of circuits* $\{f_\lambda\}_{\lambda \in \mathbb{N}}$ *such that* $f_\lambda \in \mathcal{C}$ *whose input length is* ℓ *and any messages* $m_1, \dots, m_\ell \in \{0, 1\}$*,*

$$\Pr[\mathsf{Dec}(sk, \mathsf{Eval}(pk, C, c_1, \dots, c_\ell)) \neq C(m_1, \dots, m_\ell)]$$

is negligible, where $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and* $c_i \leftarrow \mathsf{Enc}(pk, m_i)$*.*

Remark 2 *We can also consider the additional property that an output of* Eval *can be used as input of another homomorphic evaluation. This is called "multi-hop" homomorphism, and many fully homomorphic encryption schemes have this property. However, our scheme does not.*

Definition 7 *(Compactness) A homomorphic encryption scheme* HE *is compact if there exists a polynomial poly such that the output length of* Eval *is at most poly(λ)-bit.*

6.2 Φ-hiding Assumption

Here, we give the definition of the Φ-hiding assumption [20] as follows. Let $\mathsf{RSA}[p \equiv 1 \bmod e]$ be an efficient algorithm which takes the security parameter 1^λ as input and outputs (N, P, Q) where $N = PQ$ is an ℓ_N-bit Blum integer such that $P \equiv 1 \bmod e$ and $\mathbb{QR}_N^+$ is cyclic. Let $\mathcal{P}_\ell$ be the set of all ℓ-bit primes.

Definition 8 *For a constant c, we consider the following distributions.*

$$\mathcal{R} = \{(e, N) : e, e' \overset{R}{\leftarrow} \mathcal{P}_{c\ell_N}; N \leftarrow \mathsf{RSA}[p \equiv 1 \bmod e'](1^\lambda)\}$$
$$\mathcal{L} = \{(e, N) : e \overset{R}{\leftarrow} \mathcal{P}_{c\ell_N}; N \leftarrow \mathsf{RSA}[p \equiv 1 \bmod e](1^\lambda)\}$$

We say that the Φ-hiding assumption holds with respect to RSA *if for any efficient adversary* $\mathcal{A}$*,* $|\Pr[1 \leftarrow \mathcal{A}(\mathcal{L})] - \Pr[1 \leftarrow \mathcal{A}(\mathcal{R})]|$ *is negligible.*

Parameters. According to [20], N can be factorized in time $O(N^\epsilon)$ where $e \overset{R}{\leftarrow} \mathcal{P}_{c\ell_N}; N \leftarrow \mathsf{RSA}[p \equiv 1 \bmod e](1^k)$ and $c = 1/4 - \epsilon$. In our scheme, we set c to be the value such that $c\ell_N = \lambda$. This setting avoids the above mentioned attack in a usual parameter setting (e.g., $\ell_N = 1024$ for 80-bit security).

6.3 Our Construction

Here, we construct a somewhat homomorphic encryption scheme by using indistinguishability obfuscation. We use the notation for circuits on $\mathbb{QR}_N^+$ which is given in Sec. 4. In addition to that, here, we use the following notation. For circuits C_1 and C_2 such that an output of C_1 can be interpreted as input for C_2, $C_1 \circ C_2$ denotes the composition of C_1 and C_2, i.e, $C_1 \circ C_2$ is a circuit that computes $C_2(C_1(x))$ for input x. The construction of our homomorphic encryption $\mathsf{HE}_{\mathsf{Ours}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ is as follows.

$\mathsf{KeyGen}(1^\lambda)$**:** Choose $e \xleftarrow{\$} \mathcal{P}_\lambda$ and $(N, P, Q) \leftarrow \mathsf{RSA}[p \equiv 1 \bmod e](1^\lambda)$. Choose $g \xleftarrow{\$} \mathbb{QR}_N^+$ and compute an integer ρ such that $\rho \equiv 0 \bmod \mathrm{ord}(\mathbb{QR}_N^+)/e$ and $\rho \equiv 1 \bmod e$. It outputs a public key $pk = (N, e, g)$ and a secret key $sk = (\rho, pk)$.

$\mathsf{Enc}(pk, m \in \{0,1\})$**:** Choose $r \xleftarrow{\$} [(N-1)/4]$, set $c \leftarrow i\mathcal{O}(\mathsf{Max}, \tilde{C}_{N,m+re})$ and output c, where Max is defined as an integer larger than $\max_{m\in\{0,1\},r\in[(N-1)/4]}\{|\tilde{C}_{N,m+re}|\}$.

$\mathsf{Eval}(pk, f, c_1, \ldots, c_\ell)$**:** Work only if $c_1, \ldots, c_\ell$ are circuits (i.e., generated by Enc). Convert f into an arithmetic circuit f' on $\mathbb{Z}_e$. (That is, each gate of f' is addition, multiplication or negation on $\mathbb{Z}_e$.)[4] Compute as follows for all wires of f' from wires with lower depth.

- *Input*: Let w be the i-th input wire. Then c_i is assigned to this wire.
- *Addition*: Let w be an output wire of an addition gate. Set $c_w := \mathsf{Mult}(c_{A(w)}, c_{B(w)})$.
- *Multiplication*: Let w be an output wire of a multiplication gate. Set $c_w := c_{A(w)} \circ c_{B(w)}$.
- *Negation*: Let w be an output wire of a negation gate. Set $c_w := C_{N,inv} \circ c_{A(w)}$ where $C_{N,inv}$ is a circuit that computes an inverse on $\mathbb{QR}_N^+$.

Let v be the output wire. Compute $c_{\mathsf{eval}} = c_v(g)$ and output it. Note that it is a group element and not a circuit. Therefore we cannot evaluate it again.

$\mathsf{Dec}(sk, c)$**:** Work differently depending on whether c is an output of Enc or Eval. If c is an output of Enc, then compute $M = c(g)$. If $M^\rho = 1$, then output 0, and otherwise output 1. If c is an output of Eval, then output 0 if $c^\rho = 1$, and otherwise output 1.

First, we prove the correctness of the scheme. We have $e|\mathrm{ord}(\mathbb{QR}_N^+)$ by the choice of N. Therefore, there exists a subgroup G_e^+ of order e of $\mathbb{QR}_N^+$. We can see that for any element $h \in \mathbb{QR}_N^+$, h^ρ is the G_e^+ component of h. In the decryption, we have $M = i\mathcal{O}(\mathsf{Max}, C_{N,m+re})(g) = g^{m+re}$. Therefore M^ρ is the G_e^+ component of g^m. We can see that G_e^+ component of g is not 1 with overwhelming probability since e is a λ-bit prime. Therefore $M^\rho = 1$ is equivalent to $m = 0$ and $M^\rho \neq 1$ is equivalent to $m = 1$ with overwhelming probability. Thus the correctness follows.

The security of $\mathsf{HE}_{\mathsf{Ours}}$ relies on the Φ-hiding assumption.

[4] This can be done since we have $a \wedge b = a \cdot b \bmod e$ and $a \vee b = a + b - a \cdot b \bmod e$ if $a, b \in \{0,1\}$.

Theorem 3 $\mathsf{HE}_{\mathsf{Ours}}$ *is NC^1-homomorphic, compact and CPA secure if the Φ-hiding assumption holds with respect to* RSA *and $i\mathcal{O}$ is an indistinguishability obfuscator for $P/poly$.*

Here, we give only an intuitive explanation. The full proof can be found in the full version. The compactness is clear since an output of Eval consists of one group element of $\mathbb{QR}_N^+$. It is easy to see that evaluated ciphertexts are decrypted correctly. The problem is whether Eval works in polynomial time. To see this, we observe that the size of a circuit assigned to a wire of depth i is $O(2^i poly(\lambda))$. Thus if the depth of an evaluated circuit is $O(\log \lambda)$, then the size of the circuit assigned to an output wire is $O(poly(\lambda))$ and thus Eval works in polynomial time. The CPA security is reduced to the ϕ-hiding assumption: by the assumption, if N is replaced with N' such that e does not divide $\mathrm{ord}(\mathbb{QR}_{N'}^+)$, any efficient adversary cannot tell the difference. We can see that $(re \bmod \mathrm{ord}(\mathbb{QR}_{N'}^+))$ is distributed almost uniformly where $r \xleftarrow{\$} [(N'-1)/4]$ since $\gcd(e, \mathrm{ord}(\mathbb{QR}_N^+)) = 1$ holds. Thus $((m + re) \bmod \mathrm{ord}(\mathbb{QR}_{N'}^+))$ is uniformly distributed regardless of the value of m and the ciphertexts of 0 and 1 are distributed almost identically.

Acknowledgment

We would like to thank the anonymous reviewers and members of the study group "Shin-Akarui-Angou-Benkyou-Kai" for their helpful comments. Especially, we would like to thank Satsuya Ohata for his instructive comment on self-bilinear maps, and Takahiro Matsuda and Jacob Schuldt for their detailed proofreading.

References

1. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *CRYPTO*, pages 1–18, 2001.
2. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, pages 213–229, 2001.
3. Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324:71–90, 2002.
4. Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In *CRYPTO*, 2014.
5. Jung Hee Cheon and Dong Hoon Lee. A note on self-bilinear maps. *Bulletin of the Korean Mathematical Society*, 46, 2009.
6. Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *CRYPTO (1)*, pages 476–493, 2013.
7. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.
8. Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT*, pages 1–17, 2013.
9. Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure MPC from indistinguishability obfuscation. In *TCC*, pages 74–94, 2014.

10. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, pages 40–49, 2013.
11. Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In *CRYPTO (2)*, pages 479–499, 2013.
12. Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In *STOC*, pages 467–476, 2013.
13. Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32, 1989.
14. Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In *EUROCRYPT*, 2014.
15. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *STOC*, pages 545–554, 2013.
16. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In *EUROCRYPT*, pages 339–358, 2006.
17. Dennis Hofheinz and Eike Kiltz. The group of signed quadratic residues and applications. In *CRYPTO*, pages 637–653, 2009.
18. Susan Hohenberger, Amit Sahai, and Brent Waters. Replacing a random oracle: Full domain hash from indistinguishability obfuscation. *EUROCRYPT*, 2014.
19. Antoine Joux. A one round protocol for tripartite diffie-hellman. In *ANTS*, pages 385–394, 2000.
20. Eike Kiltz, Adam O'Neill, and Adam Smith. Instantiability of RSA-OAEP under chosen-plaintext attack. In *CRYPTO*, pages 295–313, 2010.
21. Alfred Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, 1993.
22. Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.
23. Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. In *STOC*, 2014.
24. Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. Cryptosystems based on pairing (in japanese). In *SCIS*, 2000.
25. Yannick Seurin. New constructions and applications of trapdoor DDH groups. In *Public Key Cryptography*, pages 443–460, 2013.