Sakai-Ohgishi-Kasahara Identity-Based Non-Interactive Key Exchange Scheme, Revisited

Yu Chen[1], Qiong Huang[2], and Zongyang Zhang[3,4,⋆]

[1] State Key Laboratory of Information Security (SKLOIS),
Institute of Information Engineering, Chinese Academy of Sciences, China
chenyu@iie.ac.cn

[2] College of Informatics, South China Agricultural University, China
csqhuang@gmail.com

[3] National Institute of Advanced Industrial Science and Technology, Japan
[4] Shanghai Jiao Tong University, China
zongyang.zhang@aist.go.jp

Abstract. Identity-based non-interactive key exchange (IB-NIKE) is a powerful but a bit overlooked primitive in identity-based cryptography. While identity-based encryption and signature have been extensively investigated over the past three decades, IB-NIKE has remained largely unstudied. Currently, there are only few IB-NIKE schemes in the literature. Among them, Sakai-Ohgishi-Kasahara (SOK) scheme is the first efficient and secure IB-NIKE scheme, which has great influence on follow-up works. However, the SOK scheme required its identity mapping function to be modeled as a random oracle to prove security. Moreover, the existing security proof heavily relies on the ability of programming the random oracle. It is unknown whether such reliance is inherent.

In this work, we intensively revisit the SOK IB-NIKE scheme, and present a series of possible and impossible results in the random oracle model and the standard model. In the random oracle model, we first improve previous security analysis for the SOK IB-NIKE scheme by giving a tighter reduction. We then use meta-reduction technique to show that the SOK scheme is unlikely proven to be secure based on the computational bilinear Diffie-Hellman (CBDH) assumption without programming the random oracle. In the standard model, we show how to instantiate the random oracle in the SOK scheme with a concrete hash function from admissible hash functions (AHFs) and indistinguishability obfuscation. The resulting scheme is fully adaptive-secure based on the decisional bilinear Diffie-Hellman inversion (DBDHI) assumption. To the best of our knowledge, this is first fully adaptive-secure IB-NIKE scheme in the standard model that does not explicitly require multilinear maps. Previous schemes in the standard model either have merely selective security or use multilinear maps as a key ingredient. Of particular interest, we generalize the definition of AHFs, and propose a generic construction which enables AHFs with previously unachieved parameters.

[⋆] Corresponding author.

W. Susilo and Y. Mu (Eds.): ACISP 2014, LNCS 8544, pp. 274–289, 2014.

1 Introduction

Identity-based non-interactive key exchange (IB-NIKE) is a natural extension of NIKE [11] in the identity-based setting, which enables any two parties registered in the same key generator center (KGC) to agree on a unique shared key without any interaction. IB-NIKE has important applications in managing keys and enabling secure communications in mobile ad hoc and sensor networks. The advantages of IB-NIKE, in terms of reducing communication costs and latency in a realistic adversarial environment, are demonstrated in [8].

In 2000, Sakai, Ohgishi and Kasahara [22] proposed the first efficient and secure IB-NIKE scheme in the random oracle model, namely the SOK scheme (with security models and formal proofs in follow up works [12, 20]). Despite the appearing of IB-NIKE in this celebrated work on identity-based cryptography, it had received less attention as a fundamental primitive in its own right over the past decade. In the last year, we have seen remarkable progress on this topic. Freire et al. [15] constructed $(\mathsf{poly}, 2)$-programmable hash functions (PHFs) from multilinear maps. By substituting the random oracle in the original SOK scheme with $(\mathsf{poly}, 2)$-PHFs, they obtained the first IB-NIKE scheme in the standard model. Boneh and Waters [6] demonstrated that constrained pseudorandom functions that support left/right predicate imply IB-NIKE. Particularly, they constructed such specific constrained PRFs based on the decisional bilinear Diffie-Hellman (DBDH) assumption, and the resulting IB-NIKE scheme (the BW scheme) can be viewed as a variant of the SOK scheme, which is also only proven secure in the random oracle model. Boneh and Zhandry [7] proposed a construction of multiparty IB-NIKE from PRG, constrained PRFs, and indistinguishability obfuscation. However, their construction only has selective security. Hnece, how to achieve fully adaptive security is left as an open problem.

1.1 Motivations

For a security reduction $\mathcal{R}$ that converts any adversary $\mathcal{A}$ with advantage $\mathsf{Adv}_{\mathcal{A}}$ against some hard problem in running time $\mathsf{Time}_{\mathcal{A}}$ to an algorithm $\mathcal{B}$ with advantage $\mathsf{Adv}_{\mathcal{B}}$ against the target cryptographic scheme in running time $\mathsf{Time}_{\mathcal{B}}$, we say it is tight if $\mathsf{Adv}_{\mathcal{B}}/\mathsf{Adv}_{\mathcal{A}}$ (advantage loose factor) is close to 1 and $\mathsf{Time}_{\mathcal{B}} - \mathsf{Time}_{\mathcal{A}}$ (time loose factor) is close to 0, and loose otherwise. It has been well known that besides theoretical interest, a tighter reduction is of utmost practical importance. To obtain the same security level, cryptographic schemes with tighter reduction generally admits more efficient implementations [1]. The exisiting proof [20] for the SOK scheme programs the random oracle H (acting as the identity mapping function in the construction) with "all-but-one" technique to implement partitioning strategy.[1] As a consequence, the advantage loose

[1] In the case of IB-NIKE, the partitioning strategy is to partition the set of all identities into "extractable" and "unextractable" ones. The reduction hopes that all identities for which an adversary requests for a secret key are extractable, while the target identities are unextractable.

factor is around $1/2^{180}$, which is far from tight. It is interesting to know if we can provide an alternative proof with tighter reduction.

Both the original security reduction [20] and our new security reduction (as we will show in Section 3.1) for the SOK scheme exploit full programmability of the random oracle model (ROM) to implement partitioning strategy. Such property allows the reduction to program the random oracle (RO) arbitrarily as long as the output distributes uniformly and independently over the range. This full-fledged model is usually refereed as fully programming ROM (FPROM). Full programmability is a strong property in that it does not quite match with the features of cryptographic hash functions. Therefore, two weaker random oracle models are proposed by constraining the ability of the reduction to program the RO. The randomly programming ROM (RPROM) [14] allows the reduction to program the RO with random instead of arbitrary values, while the non-programming ROM (NPROM) forbids the reduction to program the RO. Since the NPROM is the weakest one among the above three random oracle models and is closest to the standard model, it is curious to know if the SOK scheme could be proven secure in the NPROM.

As previously mentioned, Freire et al. [15] successfully instantiated the SOK scheme in the standard model by substituting the random oracle H with $(\mathsf{poly}, 2)$-programmable hash functions (PHFs). However, the construction of $(\mathsf{poly}, 2)$-PHFs requires multilinear maps [16]. So far, we do not have candidates for multilinear maps between groups with cryptographically hard problems. Instead, we only have concrete candidate for an "approximation" of multilinear maps, named graded encoding systems [16]. Hence, we are motivated to find an alternative approach of substituting the random oracle in the SOK scheme, with the hope that the replacements are not explicitly involved with multilinear maps. Recently, Hohenberger, Sahai and Waters [19] gave a way to instantiate the random oracle with concrete hash functions from indistinguishability obfuscation[2] in the "full domain hash" signatures. It is natural to ask if their approach can extend to other applications, and in particular, the SOK scheme.

1.2 Our Results

In the remainder of this paper, we give negative or affirmative answers to the above questions. We summarize our main results as below.

Being aware of the usage of "all-but-one" programming technique is the reason that makes the original reduction loose, we are motivated to find an alternative programming technique that admits tighter reduction. Observing the structural similarities between the SOK IB-NIKE scheme and the Boneh-Franklin [4] IBE scheme and the Boneh-Lynn-Shacham (BLS) [5] short signature, we are inspired to program the random oracle H in the SOK scheme with the flipping coin technique developed in [10], which were successfully employed in the reductions for

[2] Although currently the only known construction of indistinguishability obfuscation ($i\mathcal{O}$) is from multilinear maps [18], it is still possible that $i\mathcal{O}$ can be constructed from other primitives.

the latter two well-known schemes. Roughly speaking, the flipping coin technique usually conducts as follows: to program $\mathsf{H}(x)$ (x is an identity in the IBC setting or a message in the signature setting), the reduction flips a random coin once, then programs $\mathsf{H}(x)$ according to the coin value in two different manners. One allows the reduction to embed a trapdoor in order to extract a secret key or produce a signature, while the other allows the reduction to embed some fixed component of the challenge instance. However, this approach does not work well in the case of the SOK scheme. This is because the reduction has to embed two group elements g_2 and g_3 from the CBDH instance to $\mathsf{H}(id_a^*)$ and $\mathsf{H}(id_b^*)$ respectively, where id_a^* and id_b^* are two target identities adaptively chosen by the adversary. We overcome this difficulty by flipping random coins twice. Looking ahead, to program $\mathsf{H}(x)$, the reduction first flips a random biased coin to determine the partitioning, namely either embedding a trapdoor or embedding a component from the CBDH instance. If the first round coin value indicates the latter choice, then $\mathcal{R}$ further flips an independent and unbiased coin to determine which component is going to be embedded. As a result, we obtain a new reduction with a loose factor around $1/2^{120}$, which significantly improves the original result. The same technique can also be used to improve Boneh-Waters constrained PRFs supporting left/right predicate [6], by minimizing the number of RO and tightening the reduction.

Following the work of Fischlin and Fleischhacker [13], we use meta-reduction technique to show that the SOK scheme is unlikely proven secure to be based on the CBDH assumption in NPROM, assuming the hardness of an intractable problem called one-more CBDH problem. We obtain this result by showing that if there is a black-box reduction $\mathcal{R}$ basing the fully adaptive security of the SOK IB-NIKE scheme on the CBDH assumption in NPROM, then there exists a meta-reduction $\mathcal{M}$ breaking the one-more CBDH assumption. Our black-box separation result holds with respect to single-instance reduction which invokes only one instance of the adversary and can rewind it arbitrarily to the point after sending over the master public key. Though single-instance reduction is a slightly restricted type of reductions, it is still general enough to cover the original reduction [20] and our new reduction shown in Section 3.1. Moreover, our result holds even for selective semi-static one-way security.

Realizing the technical heart of Hohenberger-Sahai-Waters approach [19] is to replace the programmable RO with a specific hash function H satisfying suitable programmability, we successfully extend their approach in the case of IB-NIKE, going beyond the "full domain hash" signatures. More precisely, we first create a replacement hash function H for RO from puncturable PRFs. The resulting IB-NIKE scheme is selective-secure in the standard model. To attain fully adaptive security, we hope to create a specific hash function H with $(\mathsf{poly}, 2)$-programmability from admissible hash functions (AHFs). This potentially requires the AHF to be $(\mathsf{poly}, 2)$-admissible, which is not met by current AHF constructions. We circumvent this technical difficulty by giving a generic construction of (poly, c)-AHF (c could be any constant integer) from any $(\mathsf{poly}, 1)$-AHF, which utilizes Cartesian product as the key mathematical tool. We note that beyond the usage in the above construction, (poly, c)-AHF may find more important applications as a purely statistical cryptographic primitive.

2 Preliminaries and Definitions

Notations. For a distribution or random variable X, we write $x \xleftarrow{\text{R}} X$ to denote the operation of sampling a random x according to X. For a set X, we use $x \xleftarrow{\text{R}} X$ to denote the operation of sampling x uniformly at random from X, use U_X to denote the uniform distribution over set X, and use $|X|$ to denote its size. We write κ to denote the security parameter, and all algorithms (including the adversary) are implicitly given κ as input. We write $\mathsf{poly}(\kappa)$ to denote an arbitrary polynomial function in κ. We write $\mathsf{negl}(\kappa)$ to denote an arbitrary negligible function in κ, one that vanishes faster than the inverse of any polynomial. A probability is said to be overwhelming if it is $1 - \mathsf{negl}(\kappa)$, and said to be noticeable if it is $1/\mathsf{poly}(\kappa)$. A probabilistic polynomial-time (PPT) algorithm is a randomized algorithm that runs in time $\mathsf{poly}(\kappa)$.

2.1 Cartesian Product and Power of Vectors

The Cartesian product of a m-dimension vector $X = (x_1, \dots, x_m)$ and a n-dimension vector $Y = (y_1, \dots, y_n)$ over some finite set S is defined as:

$$X \times Y = \{z_{ij} := z_{(i-1)n+j} = (x_i, y_j)\}_{1 \le i \le m, 1 \le j \le n},$$

where $\times$ denotes the Cartesian product operation. $X \times Y$ can be viewed as a mn-dimension vector over S^2 or a $2mn$-dimension vector over S. The Cartesian k-power of a m-dimension vector $X = (x_1, \dots, x_n)$ over S is defined as:

$$X^k = \underbrace{X \times \cdots \times X}_{k},$$

where X^k can be viewed as a m^k-dimension vector over S^k or a km^k-dimension vector over S.

2.2 Bilinear Maps and Related Hardness Assumptions

A bilinear group system consists of two cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p, with a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ which satisfies the following properties:

- bilinear: $\forall g \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}_p$, we have $e(g^a, g^b) = e(g, g)^{ab}$.
- non-degenerate: $\forall g \in \mathbb{G}^*$, we have $e(g, g) \neq 1_{\mathbb{G}_T}$.

In the following, we write $\mathsf{BLGroupGen}$ to denote bilinear group system generator which on input security parameter κ, output $(p, \mathbb{G}, \mathbb{G}_T, e)$.

Assumption 2.1. *The computational bilinear Diffie-Hellman (CBDH) assumption in bilinear group system* $(p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathsf{BLGroupGen}(\kappa)$ *is that for any PPT adversary* $\mathcal{A}$*, it holds that:*

$$\Pr[\mathcal{A}(g, g^x, g^y, g^z) = e(g,g)^{xyz}] \leq \mathsf{negl}(\kappa),$$

where the probability is taken over the choice of $g \xleftarrow{R} \mathbb{G}$*,* $x, y, z \xleftarrow{R} \mathbb{Z}_p$*. Hereafter, we write* $\overrightarrow{v}$ *to denote a CBDH instance* $(g, g^x, g^y, g^y) \in \mathbb{G}^4$*. The decisional bilinear Diffie-Hellman (DBDH) assumption is that the two distributions* (g, g^x, g^y, g^z, T_0) *and* (g, g^x, g^y, g^z, T_1) *are computationally indistinguishable, where* $T_0 \xleftarrow{R} \mathbb{G}_T$ *and* $T_1 = e(g,g)^{xyz}$*.*

Assumption 2.2. *The n-one-more CBDH (n-omCBDH) assumption in bilinear group system* $(p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathsf{BLGroupGen}(\kappa)$ *is that for any PPT adversary* $\mathcal{A}$*, it holds that:*

$$\Pr[\mathcal{A}^{\mathsf{DL}_g(\cdot)}(g, \{g^{x_i}, g^{y_i}, g^{z_i}\}_{i=1}^{n+1}) = (\{e(g,g)^{x_i y_i z_i}\}_{i=1}^{n+1})] \leq \mathsf{negl}(\kappa),$$

where the probability is taken over the choices of $g \xleftarrow{R} \mathbb{G}$*, and* $x_i, y_i, z_i \xleftarrow{R} \mathbb{Z}_p$ *for* $i \in [n+1]$*. To solve* $n+1$ *CBDH instances,* $\mathcal{A}$ *is allowed to query* $\mathsf{DL}_g(\cdot)$ *at most n times, where* $\mathsf{DL}_g(\cdot)$ *is a discrete logarithm oracle which outputs* $t \in \mathbb{Z}_p$ *on input* $h = g^t$*. The hardness of the omCBDH problem is demonstrated by a recent result [23].*

Assumption 2.3. *The n-decisional bilinear Diffie-Hellman inversion (n-DBDHI) assumption in bilinear group system* $(p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathsf{BLGroupGen}(\kappa)$ *is that for any PPT adversary* $\mathcal{A}$*, it holds that:*

$$|\Pr[\mathcal{A}(g, g^x, \ldots, g^{x^n}, T_\beta) = 1] - 1/2| \leq \mathsf{negl}(\kappa),$$

where $T_0 \xleftarrow{R} \mathbb{G}_T$*,* $T_1 = e(g,g)^{1/x} \in \mathbb{G}_T$*, and the probability is taken over the choices of* $g \xleftarrow{R} \mathbb{G}$*,* $x \xleftarrow{R} \mathbb{Z}_p$*, and* $\beta \xleftarrow{R} \{0,1\}$*.*

As observed in [2], the n-DBDHI assumption is equivalent to the n-DBDHI assumption, which is identical to the standard one except that* T_1 *is set as* $e(g,g)^{x^{2n+1}}$ *instead of* $e(g,g)^{1/x}$*. We will, for notational convenience, base our proofs on the n-DBDHI* assumption in this work.*

2.3 Indistinguishability Obfuscation

We recall the definition of indistinguishability obfuscator from [17] as below.

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

- ***Functionality Preserving:*** *For all security parameters* $\kappa \in \mathbb{N}$*, for all* $C \in \mathcal{C}_\kappa$*, for all inputs x, we have that:*

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

- ***Indistinguishability Obfuscation:*** *For any pairs of PPT adversaries* $(\mathcal{S}, \mathcal{D})$*, there exists a negligible function* α *such that if* $\Pr[\forall x, C_0(x) = C_1(x) : (C_0, C_1, state) \leftarrow \mathcal{S}(\kappa)] \geq 1 - \alpha(\kappa)$*, then we have:*

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

2.4 Puncturable PRFs

We then recall the notion of *puncturable* PRFs [19,21], in which the key owner is able to generate a constrained key for all but polynomial number of elements in the domain.

Definition 2. *A family of puncturable PRFs* $\mathsf{F}_k : X \to Y$*, where* X *and* Y *may be parameterized by* κ*, is efficiently evaluable itself with secret key* k*. In addition, it consists of three polynomial-time algorithms* KeyGen*,* Puncture*, and* Eval *satisfying the following properties:*

- ***Evaluable under puncturing:*** *For any* $S \subseteq \{0,1\}^n$ *(containing polynomial number of punctured points), and any* $x \in X$ *but* $x \notin S$*, we have:*

$$\Pr[\mathsf{Eval}(k_S, x) = \mathsf{F}_k(x) : k_S \leftarrow \mathsf{Puncture}(k, S)] = 1$$

- ***Pseudorandom at punctured points:*** *For any PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *such that* $\mathcal{A}_1(\kappa)$ *outputs a set* $S \subseteq X$ *and state* τ*, we have:*

$$|\Pr[\mathcal{A}_2(\tau, k_S, S, \mathsf{F}_k(S)) = 1] - \Pr[\mathcal{A}_2(\tau, k_S, S, U_{Y^{|S|}}) = 1]| \leq \mathsf{negl}(\kappa)$$

  where $S = \{x_1, \ldots, x_t\}$ *is the enumeration of the elements of* S *in lexicographic order,* $k_S \leftarrow \mathsf{Puncture}(k, S)$*,* $\mathsf{F}_k(S)$ *denotes the concatenation of* $\mathsf{F}_k(x_1), \ldots, \mathsf{F}_k(x_t)$*. The probability is defined over the choice of* $k \leftarrow \mathsf{KeyGen}(\kappa)$*.*

For ease of notation, sometimes we write $\mathsf{F}_{k_S}(x)$ *to represent* $\mathsf{Eval}(k_S, x)$*, and write* $k(S)$ *to represent the punctured key* $k_S \leftarrow \mathsf{Puncture}(k, S)$*.*

2.5 Non-Interactive Identity-Based Key Exchange

An non-interactive identity-based key exchange (IB-NIKE) scheme consists of the following polynomial-time algorithms:

- $\mathsf{Setup}(\kappa)$: on input security parameter κ, output master public key mpk and master secret key msk. Let I be the identity space and SHK be the shared key space.
- $\mathsf{Extract}(msk, id)$: on input msk and identity $id \in I$, output a secret key sk_{id} for id.
- $\mathsf{Share}(sk_{id_a}, id_b)$: on input secret key sk_{id_a} for identity id_a and another identity id_b, output a shared key shk for (id_a, id_b).

Correctness: For any $\kappa \in \mathbb{N}$, any $(mpk, msk) \leftarrow \mathsf{Setup}(\kappa)$, any pair of identities (id_a, id_b), any $sk_{id_a} \leftarrow \mathsf{Extract}(msk, id_a)$, $sk_{id_b} \leftarrow \mathsf{Extract}(msk, id_b)$, we have:

$$\mathsf{Share}(sk_{id_a}, id_b) = \mathsf{Share}(sk_{id_b}, id_a)$$

Security: Let $\mathcal{A}$ be an adversary against IB-NIKE and define its advantage as:

$$\mathrm{Adv}_{\mathcal{A}}(\kappa) = \Pr\left[\beta = \beta' : \begin{array}{l} (mpk, msk) \leftarrow \mathsf{Setup}(\kappa); \\ (id_a^*, id_b^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{extract}}(\cdot), \mathcal{O}_{\mathsf{reveal}}(\cdot,\cdot)}(mpk); \\ shk_0^* \xleftarrow{\mathrm{R}} SHK, shk_1^* \leftarrow \mathsf{Share}(id_a^*, id_b^*); \\ \beta \xleftarrow{\mathrm{R}} \{0,1\}; \\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{extract}}(\cdot), \mathcal{O}_{\mathsf{reveal}}(\cdot,\cdot)}(shk_\beta^*); \end{array}\right] - \frac{1}{2},$$

where $\mathcal{O}_{\mathsf{extract}}(id) = \mathsf{Extract}(msk, id)$, $\mathcal{O}_{\mathsf{reveal}}(id_a, id_b) = \mathsf{Share}(sk_{id_a}, id_b)$, and $\mathcal{A}$ is not allowed to query $\mathcal{O}_{\mathsf{extract}}(\cdot)$ for the target identities id_a^* and id_b^* and query $\mathcal{O}_{\mathsf{reveal}}(\cdot,\cdot)$ for (id_a^*, id_b^*) and (id_b^*, id_a^*). We say IB-NIKE is fully adaptive-secure if no PPT adversary has non-negligible advantage in the above security experiment. The fully adaptive security is the strongest security notion for IB-NIKE so far. The selective security can be defined similarly as above by requiring the adversary to commit the target identities (id_a^*, id_b^*) before it seeing mpk, while the semi-static security can be defined similarly above by discarding $\mathcal{O}_{\mathsf{reveal}}(\cdot,\cdot)$.

3 Revisit Sakai-Ohgishi-Kasahara IB-NIKE

We begin this section by recalling the SOK IB-NIKE scheme [22], which is given by the following three algorithms:

- $\mathsf{Setup}(\kappa)$: run $\mathsf{BLGroupGen}(\kappa)$ to generate $(p, \mathbb{G}, \mathbb{G}_T, e)$, pick $x \xleftarrow{\mathrm{R}} \mathbb{Z}_p$, set $h = g^x$; output $mpk = (h, \mathsf{H}, \mathsf{G})$ and $msk = x$, where $\mathsf{H} : I \rightarrow \mathbb{G}$ is the identity mapping function and $\mathsf{G} : \mathbb{G}_T \rightarrow \{0,1\}^n$ is the key mapping function.
- $\mathsf{Extract}(msk, id)$: on input $msk = x$ and $id \in I$, output $sk_{id} \leftarrow \mathsf{H}(id)^x$.
- $\mathsf{Share}(sk_{id_a}, id_b)$: on input sk_{id_a} and id_b, output $shk \leftarrow \mathsf{G}(e(sk_{id_a}, \mathsf{H}(id_b)))$.

3.1 An Improved Proof for the SOK IB-NIKE

The original reduction [20] for the SOK IB-NIKE lose a factor of $1/Q_1^2 Q_2$. In this subsection, we show that the fully adaptive security for the SOK scheme can be reduced to the CBDH problem with a tighter security reduction.

Theorem 1. *The SOK IB-NIKE scheme is fully adaptive-secure in the random oracle model assuming the CBDH assumption holds in bilinear group system generated by* $\mathsf{BLGroupGen}(\kappa)$. *Suppose* H *and* G *are random oracles, for any adversary* $\mathcal{A}$ *breaking the SOK IB-NIKE scheme with advantage* $\mathsf{Adv}_{\mathcal{A}}(\kappa)$ *that makes at most* Q_e *extraction queries and* Q_r *reveal queries and* Q_2 *random oracle queries to* G, *there is an algorithm* $\mathcal{B}$ *that solves the CBDH problem with advantage* $4\mathsf{Adv}_{\mathcal{A}}(\kappa)/e^2(Q_e + Q_r)^2 Q_2$, *where e is the natural logarithm.*

Due to space limitation, we defer the proof of Theorem 1 in the full version.

3.2 SOK IB-NIKE Is Not Provably Secure under NPROM

We now show that the SOK IB-NIKE can not be proven secure without programming the random oracle with respect to a slightly restricted type of reductions, which is called *single-instance* reduction in [13]. In the case of identity-based schemes (including IBE, IBS as well as IB-NIKE), the restrictions lie at such a type of reductions can only invoke a single instance of the adversary and, can not rewind the adversary to a point before it hands over *mpk* for the first time. We have the following theorem whose proof appears in the full version.

Theorem 2 (Non-Programming Irreducibility for SOK IB-NIKE). *Assume the 1-omCBDH assumption holds in bilinear group system generated by* $\mathsf{BLGroupGen}(\kappa)$*, then there exists no non-programming single-instance fully-black-box reduction that reduces the fully adaptive security of SOK IB-NIKE to the CBDH problem. More precisely, assume there exists such a reduction* $\mathcal{R}$ *that converts any adversary* $\mathcal{A}$ *against the SOK IB-NIKE into an algorithm against the CBDH problem. Assume further that the reduction* $\mathcal{R}$ *has success probability* $\mathsf{Succ}^{\mathrm{CBDH}}_{\mathcal{R}^{\mathcal{A}}}$ *for given* $\mathcal{A}$ *and runtime* $\mathsf{Time}_{\mathcal{R}}(\kappa)$*. Then, there exists a family* $\mathbb{A}$ *of successful (but possibly inefficient) adversaries* $\mathcal{A}_{\mathcal{R},a}$ *against fully adaptive security of SOK IB-NIKE and a meta-reduction* $\mathcal{M}$ *that breaks the 1-omCBDH assumption with success probability* $\mathsf{Succ}^{\text{1-omCBDH}}_{\mathcal{M}}(\kappa) \geq (\mathsf{Succ}^{\mathrm{CBDH}}_{\mathcal{R}^{\mathcal{A}_{\mathcal{R},a}}}(\kappa))^2$ *for a random* $\mathcal{A}_{\mathcal{R},a} \in \mathbb{A}$ *and runtime* $\mathsf{Time}_{\mathcal{M}}(\kappa) = 2 \cdot \mathsf{Time}_{\mathcal{R}}(\kappa) + \mathsf{poly}(\kappa)$*.*

4 IB-NIKE from Indistinguishability Obfuscation

4.1 Warmup: Selectively Secure IB-NIKE from *iO*

As a warmup, we show how to create a replacement for the RO $\mathsf{H}(\cdot)$ in the SOK scheme from puncturable PRFs and $i\mathcal{O}$. The resulting scheme is selective-secure in the standard model.

Selectively Secure Construction from $i\mathcal{O}$

- $\mathsf{Setup}(\kappa)$: run $\mathsf{BLGroupGen}(\kappa)$ to generate $(p, \mathbb{G}, \mathbb{G}_T, e)$, pick $x \xleftarrow{\text{R}} \mathbb{Z}_p$ and $g \xleftarrow{\text{R}} \mathbb{G}^*$; pick a secret key k for puncturable PRF $\mathsf{F} : I \to \mathbb{Z}_p$; then create an obfuscation of the program H shown in Fig. 1. The size of the program is padded to be the maximum of itself and the program H^* shown in Fig. 2. We refer to the obfuscated program as the function $\mathsf{H} : I \to \mathbb{G}$, which acts as the random oracle type hash function in the SOK scheme. The *msk* is x, whereas *mpk* is the hash function $\mathsf{H}(\cdot)$.
- Algorithm $\mathsf{Extract}$ and Share are identical to that in the SOK scheme.

Theorem 3. *The above IB-NIKE scheme is selective-secure if the obfuscation scheme is indistinguishably secure,* F *is a secure punctured PRF, and the DBDH assumption holds.*

Due to space limitation, we defer the proof of Theorem 3 in the full version.

Selective Hash H

Constants: Punctured PRF key k, $g \in \mathbb{G}^*$.
Input: Identity id.

```
1. Output $g^{\mathsf{F}_k(id)}$.
```

Fig. 1. Selective Hash H

Selective Hash H^*

Constants: Punctured PRF key $k(S)$ for $S = \{id_a^*, id_b^*\}$, $id_a^*, id_b^* \in I$, $z_1^*, z_2^* \in \mathbb{G}$, $g \in \mathbb{G}^*$.
Input: Identity id.

```
1. If $id = id_a^*$ output $z_1^*$ and exit.
2. If $id = id_b^*$ output $z_2^*$ and exit.
3. Else output $g^{\mathsf{F}_{k(S)}(id)}$.
```

Fig. 2. Selective Hash H^*

4.2 Main Result: Adaptively Secure IB-NIKE from $i\mathcal{O}$

We now show how to create a replacement for the RO $\mathsf{H}(\cdot)$ in the SOK IB-NIKE scheme from $(\mathsf{poly}, 2)$-AHF and $i\mathcal{O}$ to attain adaptive security in the standard model. We first recap the definition of AHF and present a generic construction of $(\mathsf{poly}, 2)$-AHF.

Admissible Hash Functions. Our definition below is generalization of "admissible hash function"(AHF) [3,9,15].

Definition 3 (AHF). *Let ℓ, l, and θ be efficiently computable univariate polynomials of κ. For an efficiently computable function $\mathsf{AHF} : \{0,1\}^\ell \rightarrow \{0,1\}^l$, define the predicate $P_u : \{0,1\}^\ell \rightarrow \{0,1\}$ for any $u \in \{0,1,\perp\}^l$ as $P_u(x) = 0 \iff \forall i : \mathsf{AHF}(x)_i \neq u_i$, where $\mathsf{AHF}(x)_i$ denotes the i-th component of $\mathsf{AHF}(x)$. We say that AHF is (m, n)-admissible if there exists a PPT algorithm $\mathsf{AdmSample}$ and a polynomial $\theta(\kappa)$, such that for all $x_1, \ldots, x_m, z_1, \ldots, z_n \in \{0,1\}^\ell$, where $x_i \neq z_j$ for all $1 \leq i \leq m$ and $1 \leq j \leq n$, we have that:*

$$\Pr[P_u(x_1) = \cdots = P_u(x_m) = 1 \wedge P_u(z_1) = \cdots = P_u(z_n) = 0] \geq 1/\theta(\kappa) \qquad (1)$$

where the probability is over the choice of $u \leftarrow \mathsf{AdmSample}(\kappa)$. Particularly, we say that AHF is (poly, n)-admissible if AHF is (q, n)-admissible for any polynomial $q = q(\kappa)$ and constant $n > 0$. Note that in the standard definition of AHF, the second parameter n is fixed to 1. To show the existence of (q, n)-AHF for $n \geq 1$, we present the following theorem.

Theorem 4. *Let $q = q(\kappa)$ be a polynomial, n be a constant, and* AHF *(with* $\mathsf{AdmSample}$*) be a $(q, 1)$-AHF from $\{0,1\}^\ell$ into $\{0,1\}^l$. Then* AHF' *with:*

- $\mathsf{AHF}'(x) = \underbrace{\mathsf{AHF}(x) \times \cdots \times \mathsf{AHF}(x)}_{n}$.
- $P'_u : \{0,1\}^\ell \to \{0,1\}$ *for any $u \in \{0,1,\perp\}^{nl^n}$ is defined as $P'_u(x) = 0 \iff \forall i : \mathsf{AHF}'(x)_i \neq u_i$, where $\mathsf{AHF}'(x)_i$ denotes the i-th component of $\mathsf{AHF}'(x)$.*
- $\mathsf{AdmSample}'(\kappa)$*: run $\mathsf{AdmSample}(\kappa)$ independently n times to generate u_1, $\ldots$, $u_n \in \{0,1\}^l$, output $u = \underbrace{u_1 \times \cdots \times u_n}_{n}$.*

is a (q, n)-AHF from $\{0,1\}^\ell$ into $\{0,1\}^{nl^n}$. Here $\times$ denotes the Cartesian product defined in Section 2.1. $\mathsf{AHF}'(x)$ can be viewed as a nl^n-dimension vector over $\{0,1\}$, and u can be viewed as a nl^n-dimension vector over $\{0,1,\perp\}$.

Proof. We first note that the definition of P'_u for AHF' is compatible with that of P_u for AHF. According the construction of AHF' and $\mathsf{AdmSample}'(\kappa)$, we have $P'_u(x) = P_{u_1}(x) \wedge \cdots \wedge P_{u_n}(x)$. Now fix $q+n$ distinct elements $x_1, \ldots, x_q, z_1, \ldots, z_n \in \{0,1\}^\ell$. For each $i \in [n]$, define event A_i as: $P_{u_i}(x_j) = 1$ for all $1 \leq j \leq q$ and $P_{u_i}(z_i) = 0$ (the predicate values on the rest $n-1$ elements could be either 0 or 1). Define event A as: $P'_u(x_j) = 1$ for all $1 \leq j \leq q$ and $P'_u(z_i) = 0$ for all $1 \leq i \leq n$. According to the definition of P'_u, we have: $A \supseteq A_1 \wedge \cdots \wedge A_n$. Since AHF is a $(q, 1)$-AHF, thus each event A_i happens independently with probability at least $1/\theta(\kappa)$ (over the choice of $u_i \leftarrow \mathsf{AdmSample}(\kappa)$). Therefore, we have: $\Pr[A] \geq \prod_{i=1}^{n} \Pr[A_i] \geq 1/(\theta(\kappa))^n$, which indicates AHF' is a (q, n)-AHF. This proves the theorem.

Adaptively Secure Construction from $i\mathcal{O}$

- $\mathsf{Setup}(\kappa)$: run $\mathsf{BLGroupGen}(\kappa)$ to generate $(p, \mathbb{G}, \mathbb{G}_T, e)$, pick $x \xleftarrow{\text{R}} \mathbb{Z}_p$ and $g \xleftarrow{\text{R}} \mathbb{G}^*$; pick a secret key k for puncturable PRF $\mathsf{F} : I \to \mathbb{Z}_p$; pick uniformly at random $(c_{1,0}, c_{1,1})$, $\ldots$, $(c_{n,0}, c_{n,1})$ each from $\mathbb{Z}_p$; then create an obfuscation of the program H shown in Fig. 3, where the size of the program is padded to be the maximum of itself and the program of H^*shown in Fig. 4. The *msk* is x, whereas *mpk* is the hash function $\mathsf{H}(\cdot)$.
- Algorithms $\mathsf{Extract}$ and Share are identical to that in the SOK IB-NIKE.

Theorem 5. *The above IB-NIKE scheme is adaptively secure if the obfuscation scheme is indistinguishable secure and the n-DBDHI assumption holds in bilinear group system.*

Proof. We proceed via a sequence of hybrid games, where the first game corresponds to the standard adaptive security game. We first prove that any two successive games are computationally indistinguishable. We then show that any PPT adversary in the final game that succeeds with non-negligible probability can be used to break the n-DBDHI assumption.

Adaptive Hash H

```
Constants: $g \in \mathbb{G}^*$, exponents $c_{i,\alpha} \in \mathbb{Z}_p$ for $i \in [n]$ and $\alpha \in \{0,1\}$.
Input: Identity $id$.
  1. Compute $w \leftarrow \mathsf{AHF}(id)$.
  2. Output $g^{\prod_{i=1}^{n} c_{i,w_i}}$.
```

Fig. 3. Adaptive Hash H

Adaptive Hash H^*

```
Constants: $g \in \mathbb{G}^*$, $g^x, \dots, g^{x^n} \in \mathbb{G}$ for some $x \in \mathbb{Z}_p$, exponents $y_{i,\alpha} \in \mathbb{Z}_p$ for $i \in [n]$ and $\alpha \in \{0,1\}$, $u \in \{0,1,\perp\}^n$.
Input: Identity $id$.
  1. Compute $w \leftarrow \mathsf{AHF}(id)$.
  2. Compute the set size $|\mu(w)|$, where $\mu(w)$ is the set $i$ such that $w_i \neq u_i$.
  3. Output $(g^{x^{|\mu(w)|}})^{\prod_{i=1}^{n} y_{i,w_i}}$.
```

Fig. 4. Adaptive Hash H^*

Game 0: This game is identical to standard adaptive security game played between adversary $\mathcal{A}$ and challenger $\mathcal{CH}$:

- Setup: $\mathcal{CH}$ runs $\mathsf{BLGroupGen}(\kappa)$ to generate $(p, \mathbb{G}, \mathbb{G}_T, e)$, picks $x \xleftarrow{\text{R}} \mathbb{Z}_p$ and $g \xleftarrow{\text{R}} \mathbb{G}^*$, then chooses exponents $c_{i,\alpha}$ uniformly at random $\mathbb{Z}_p$ for $i \in [n]$ and $\alpha \in \{0,1\}$, creates the hash function $\mathsf{H}(\cdot)$ as an obfuscation of the program of H shown in Fig. 3, and pads its size to be the maximum of itself and the program of H^* shown in Fig. 4. $\mathcal{CH}$ sets $msk = x$ and $mpk = \mathsf{H}$.
- Phase 1: $\mathcal{A}$ can issue the following two types of queries:
  - extraction query $\langle id \rangle$: $\mathcal{CH}$ responds with $sk_{id} = \mathsf{H}(id)^x$.
  - reveal query $\langle id_a, id_b \rangle$: $\mathcal{CH}$ first extracts secret key sk_{id_a} for id_a, then responds with $shk \leftarrow \mathsf{Share}(sk_{id_a}, id_b)$.
- Challenge: $\mathcal{A}$ submits id_a^* and id_b^* as the target identities with the restriction that either id_a^* or id_b^* has not been queried for secret key. $\mathcal{CH}$ picks $shk_0^* \xleftarrow{\text{R}} SHK$ and computes $shk_1^* \leftarrow \mathsf{Share}(sk_{id_a^*}, id_b^*)$, then picks $\beta \xleftarrow{\text{R}} \{0,1\}$ and sends shk_β^* to $\mathcal{A}$ as the challenge.
- Phase 2: $\mathcal{A}$ can continue to issue the extraction queries and the reveal queries, $\mathcal{CH}$ proceeds the same way as in Phase 1 except that the extraction queries to id_a^* or id_b^* and reveal query for (id_a^*, id_b^*) are not allowed.
- Guess: $\mathcal{A}$ outputs its guess β' and wins if $\beta = \beta'$.

Game 1: same as Game 0 except that $\mathcal{CH}$ generates the exponents $c_{i,\alpha}$ as follows: first samples $u \in (\{0,1,\perp\})^n$ via $\mathsf{AdmSample}(\kappa, Q)$, where Q is the upper bound on the number of queries made by $\mathcal{A}$ (including extraction queries and reveal queries), then for $i \in [n]$ and $\alpha \in \{0,1\}$ chooses $y_{i,\alpha} \xleftarrow{\text{R}} \mathbb{Z}_p$, and sets:

$$c_{i,\alpha} = \begin{cases} y_{i,\alpha} & \text{if } \alpha = u_i \\ x \cdot y_{i,\alpha} & \text{if } \alpha \neq u_i \end{cases}$$

Game 2: same as Game 1 except that $\mathcal{CH}$ creates the hash function $\mathsf{H}(\cdot)$ as an obfuscation of program H^* shown in Fig. 4.

Lemma 1. *Game 0 and Game 1 are statistically indistinguishable.*

Proof. This lemma immediately follows from the facts: (1) in Game 1 the sampling of u only determines the generation of $c_{i,\alpha}$ and it is independent of the rest game; (2) the value of $c_{i,\alpha}$ distributes uniformly at random from $\mathbb{Z}_p$ in both Game 0 and Game 1.

Lemma 2. *Game 1 and Game 2 are computationally indistinguishable if the underlying obfuscation scheme is indistinguishability secure.*

Proof. We prove this lemma by giving a reduction to the indistinguishability security of the obfuscator. More precisely, suppose there is an PPT adversary $\mathcal{A}$ can distinguish Game 1 and Game 2, then we can build algorithms $(\mathcal{S}, \mathcal{D})$ against the indistinguishability of the obfuscator by interacting with $\mathcal{A}$ as follows.

Sample: $\mathcal{S}$ runs $\mathsf{BLGroupGen}(\kappa)$ to generate $(p, \mathbb{G}, \mathbb{G}_T, e)$, picks $x \xleftarrow{\text{R}} \mathbb{Z}_p$ and $g \xleftarrow{\text{R}} \mathbb{G}$, prepares g^{x^i} for $i \in [n]$, runs $\mathsf{AdmSample}(\kappa, Q)$ to obtain a string $u \in (\{0,1,\perp\})^n$, and for $i \in [n]$ and $\alpha \in \{0,1\}$ chooses $y_{i,\alpha} \xleftarrow{\text{R}} \mathbb{Z}_p$, then sets:

$$c_{i,\alpha} = \begin{cases} y_{i,\alpha} & \text{if } \alpha = u_i \\ x \cdot y_{i,\alpha} & \text{if } \alpha \neq u_i \end{cases}$$

It sets $\tau = (c_{i,\alpha}, y_{i,\alpha}, u)$ and builds C_0 as the program of H, and C_1 as the program of H^*. Before describing $\mathcal{D}$, we observe that by construction, the circuits C_0 and C_1 always behave identically on every input. To show program equivalence, note that for all $w \in \{0,1\}^n$, we have that:

$$g^{\prod_i^n c_{i,\alpha_i}} = g^{x^{|\mu(w)|} \cdot \prod_i^n y_{i,w_i}} = (g^{x^{|\mu(w)|}})^{\prod_i^n y_{i,w_i}}$$

With suitable padding, both C_0 and C_1 have the same size. Thus, $\mathcal{S}$ satisfies the conditions needed for invoking the indistinguishability property of the obfuscator. Now, we can describe the algorithm $\mathcal{D}$, which takes as input τ as given above, and the obfuscation of either C_0 or C_1.

Distinguish: $\mathcal{D}$ sets $msk = x$ and builds mpk from C_β, then invokes $\mathcal{A}$ in the adaptive security game for IB-NIKE. When $\mathcal{A}$ issues extraction queries and reveal queries, $\mathcal{D}$ responds with msk. If $\mathcal{A}$ wins, $\mathcal{D}$ outputs 1.

By construction, if $\mathcal{D}$ receives an obfuscation of C_0, then the probability that $\mathcal{D}$ outputs 1 is exactly the probability that $\mathcal{A}$ wins in Game 1. On the other hand, if $\mathcal{D}$ receives an obfuscation of C_1, then the probability that $\mathcal{D}$ outputs 1 is the probability that $\mathcal{A}$ wins in Game 2. The indistinguishability of the obfuscator implies Game 1 and Game 2 are computationally indistinguishable. The lemma immediately follows.

Lemma 3. *$\mathcal{A}$'s advantage in Game 2 is negligible in κ.*

Proof. We prove this lemma by showing that any adversary $\mathcal{A}$ has non-negligible advantage in Game 2 implies an algorithm $\mathcal{B}$ that has non-negligible advantage against the n-DBDHI problem. Given a n-DBDHI instance $(g, g^x, \ldots, g^{x^n}, T_\beta)$, $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

- Setup: $\mathcal{B}$ first runs $\mathsf{AdmSample}(\kappa, Q)$ to obtain $u \in \{0, 1, \perp\}^n$, where Q is the sum of Q_e (the maximum number of extraction queries) and Q_r (the maximum number of the reveal queries). For $i \in [n]$ and $\alpha \in \{0, 1\}$, $\mathcal{B}$ chooses random $y_{i,\alpha} \in \mathbb{Z}_p$, then creates the hash function $\mathsf{H}(\cdot)$ as an obfuscation of the program H^* using the input DBDHI instance as well as $y_{i,\alpha}$ and u.
- Phase 1: $\mathcal{A}$ can issue the following two types of queries:
  - extraction queries $\langle id \rangle$: If $P_u(id) = 0$, then $\mathcal{B}$ aborts and outputs a random guess for β. Else, $\mathcal{B}$ extracts the secret key from the input n-DBDHI instance and the $y_{i,\alpha}$ values. $\mathcal{B}$ could to do so since $P_u(id) = 1$ implies there exists at least one i such that $w_i = u_i$. In this case $\mathsf{H}(id)$ will contain a power of x that is strictly less than n.
  - reveal queries $\langle id_a, id_b \rangle$: If $P_u(id_a) = 0 \wedge P_u(id_b) = 0$, then $\mathcal{B}$ aborts and outputs a random guess for β. Otherwise, either $P_u(id_a) = 1$ or $P_u(id_b) = 1$. Therefore, $\mathcal{B}$ can at least extract a secret key for one identity and then computes the shared key.
- Challenge: $\mathcal{A}$ outputs the target identities (id_a^*, id_b^*). If $P_u(id_a^*) = 1 \vee P_u(id_b^*) = 1$, then $\mathcal{B}$ aborts and outputs a random guess for β. Else, we have $P_u(id_a^*) = 0 \wedge P_u(id_b^*) = 0$, which means $\mathsf{AHF}(id_a^*)_i \neq u_i$ and $\mathsf{AHF}(id_b^*) \neq u_i$ for all $i \in [n]$. In this situation, both the hash values of id_a^* and id_b^* will be g^{a^n} raised to some known product of some $y_{i,\alpha}$ values. Denote the products by y_a^* and y_b^*, respectively. $\mathcal{B}$ thus sends $shk_\beta^* = (T_\beta)^{y_a^* y_b^*}$ to $\mathcal{A}$ as the challenge. It is easy to verify that if $T_\beta \xleftarrow{\text{R}} \mathbb{G}_T$ then shk_β^* also distributes uniformly over $\mathbb{G}_T$, else if $T_\beta = e(g, g)^{x^{2n+1}}$ then $shk_\beta^* = e(\mathsf{H}(id_a^*), \mathsf{H}(id_b^*))^a$.
- Phase 2: same as in Phase 1 except that the extraction queries $\langle id_a^* \rangle$, $\langle id_b^* \rangle$ and the reveal query $\langle id_a^*, id_b^* \rangle$ are not allowed.
- Guess: When $\mathcal{A}$ outputs its guess β', $\mathcal{B}$ forwards β' to its own challenger.

Since the choice of $u \leftarrow \mathsf{AdmSample}(\kappa, Q)$ determines whether or not $\mathcal{B}$ aborts and it is independent of the rest of the interaction. We conclude that conditioned on $\mathcal{B}$ does not abort, $\mathcal{A}$'s view in the above game is identical to that in Game 2. Let F be the event that $\mathcal{B}$ does not abort, we have $\mathsf{Adv}_{\mathcal{B}}(\kappa) = \Pr[F] \cdot \mathsf{Adv}_{\mathcal{A}}(\kappa)$. In what follows, we estimate the low bound of $\Pr[F]$. Let $\{id_i\}_{1 \leq i \leq Q_e}$ be Q_e distinct extraction queries, $\{(id_{j,1}, id_{j,2})\}_{1 \leq j \leq Q_r}$ be Q_r distinct reveal queries. During the game, $\mathcal{B}$ will abort if one of the following events does not happen.

$$\begin{aligned}&F_1 : \bigwedge_{i=1}^{Q_e}(P(id_i) = 1)\\&F_2 : \bigwedge_{j=1}^{Q_r}(P(id_{j,1}) = 1 \vee P(id_{j,2}) = 1)\\&F_3 : P_u(id_1^*) = 0 \wedge P_u(id_2^*) = 0\end{aligned}$$

We have $F = F_1 \wedge F_2 \wedge F_3$. Note that in each extraction query, there exists at least one identity different from both id_1^* and id_2^*. Suppose $Q_e + Q_r \leq Q$, then according to the fact that AHF is a $(Q, 2)$-AHF, we have $\Pr[F] \geq \theta(\kappa)$. The lemma immediately follows.

Combining the above three lemma, our main theorem immediately follows.

Acknowledgement. We greatly thank Dennis Hofheinz for helpful clarifications on admissible hash functions. In particular, we thank Dennis for suggesting the construction of (poly, c)-AHFs in Section 4.2. The first author is supported the National Natural Science Foundation of China under Grant No. 61303257, the Strategic Priority Research Program of CAS (Chinese Academy of Sciences) under Grant No. XDA06010701, and the National 973 Program of China under Grant No. 2011CB302400. The second author is supported by the National Natural Science Foundation of China under Grant No. 61103232, the Guangdong Natural Science Foundation under Grant No. S2013010011859, and the Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20114404120027. The third author is an International Research Fellow of JSPS and supported by the National Natural Science Foundation of China under Grant No. 61303201.

References

1. Bellare, M., Ristenpart, T.: Simulation without the artificial abort: Simplified proof and improved concrete security for waters' ibe scheme. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 407–424. Springer, Heidelberg (2009)
2. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
3. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)
4. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
5. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001)
6. Boneh, D., Waters, B.: Constrained pseudorandom functions and their applications. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 280–300. Springer, Heidelberg (2013)
7. Boneh, D., Zhandry, M.: Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation (2013), `http://eprint.iacr.org/2013/642`

8. Capar, C., Goeckel, D., Paterson, K.G., Quaglia, E.A., Towsley, D., Zafer, M.: Signal-flow-based analysis of wireless security protocols. Information and Computation 226, 37–56 (2013)
9. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010)
10. Coron, J.S.: On the exact security of full domain hash. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 229–235. Springer, Heidelberg (2000)
11. Diffie, W., Hellman, M.E.: New directions in cryptograpgy. IEEE Transactions on Infomation Theory 22(6), 644–654 (1976)
12. Dupont, R., Enge, A.: Provably secure non-interactive key distribution based on pairings. Discrete Applied Mathematics 154(2), 270–276 (2006)
13. Fischlin, M., Fleischhacker, N.: Limitations of the meta-reduction technique: The case of schnorr signatures. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 444–460. Springer, Heidelberg (2013)
14. Fischlin, M., Lehmann, A., Ristenpart, T., Shrimpton, T., Stam, M., Tessaro, S.: Random oracles with(out) programmability. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 303–320. Springer, Heidelberg (2010)
15. Freire, E.S.V., Hofheinz, D., Paterson, K.G., Striecks, C.: Programmable hash functions in the multilinear setting. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 513–530. Springer, Heidelberg (2013)
16. Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 1–17. Springer, Heidelberg (2013)
17. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits (2013), http://eprint.iacr.org/2013/451
18. Garg, S., Gentry, C., Halevi, S., Sahai, A., Waters, B.: Attribute-based encryption for circuits from multilinear maps. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 479–499. Springer, Heidelberg (2013)
19. Hohenberger, S., Sahai, A., Waters, B.: Replacing a random oracle: Full domain hash from indistinguishability obfuscation (2013), http://eprint.iacr.org/2013/509
20. Paterson, K.G., Srinivasan, S.: On the relations between non-interactive key distribution, identity-based encryption and trapdoor discrete log groups. Des. Codes Cryptography 52(2), 219–241 (2009)
21. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: Deniable encryption, and more (2013), http://eprint.iacr.org/2013/454
22. Sakai, R., Ohgishi, K., Kasahara, M.: Cryptosystems based on pairing. In: The 2000 Symposium on Cryptography and Information Security, Japan, vol. 45, pp. 26–28 (2000)
23. Zhang, J., Zhang, Z., Chen, Y., Guo, Y., Zhang, Z.: Generalized "one-more" problems and black-box separations. In: CRYPTO 2014 (submitted, 2014)