# Blind HIBE and its Applications to Identity-Based Blind Signature and Blind Decryption

Le Trieu Phong and Wakaha Ogata

July 29, 2008

#### Abstract

We explicitly describe and analyse *blind* hierachical identity-based encryption (*blind* HIBE) schemes, which are natural generalizations of blind IBE schemes [20]. We then uses the blind HIBE schemes to construct: (1) An identity-based blind signature scheme secure in the standard model, under the computational Diffie-Hellman (CDH) assumption, and with much shorter signature size and lesser communication cost, compared to existing proposals. (2) A new mechanism supporting a user to buy digital information over the Internet without revealing what he/she has bought, while protecting the providers from cheating users.

### 1 Introduction

Blind identity-based encryption (blind IBE) is a concept introduced in [20]. It is essentially the same as an IBE scheme, except that a user can obtain his/her private key in a blind manner (namely, without revealing his/her identity). The paper [20] used blind IBE for constructing some oblivious transfer (OT) protocols.

Blind hierarchical IBE (blind HIBE) is a natural generalization of blind IBE. In a blind HIBE, identities are now vectors, and the structure of blind HIBE is now like a hierarchical tree. In a blind HIBE, a child node obtains his/her private key from the corresponding parent's node in a blind manner, like blind IBE. We explicitly describe two concrete constructions of blind HIBE, based on the HIBEs of Boneh-Boyen [2] and Chatterjee-Sakar [17].

Bind HIBE was also realized (without *explicit* constructions) by Green and Hohenberger [20], but they showed no applications of them. We note that, in contrast with IBE and HIBE, the blind versions themselves may be of no *direct* use in practice. However, deploying it as a building block for other constructions would be useful, and that is what we provide in this paper.

In this paper, we consider two applications which needs blind HIBE as a building block. In the first application of blind HIBE, we use it to construct an identitybased blind signature (IBBS) scheme enjoying several merits over previous works in the literature. In the second application, we use blind HIBE to build a new mechanism called *hierachical blind decryption protocol* (HBDP), which is useful for anonymously trading over the Internet. Let us now explain them in more details in the following subsections.

# 1.1 The first application: Shorter IBBS scheme in the standard model

The concept of blind signature was firstly suggested in [6] for Internet banking. It was later used in E-voting schemes. In a blind signature scheme, a user can obtain a signature on a message m without revealing m to the signer. In identity-based blind signature (IBBS), signers are associated with identities (e.g., e-mail addresses, instead of public keys as in standard blind signature schemes). Most of works on IBBS was in the random oracle model [35], [36], [13] (with its well-known limitations [10]), so we exclude them from the following discussion. The first work in the standard model is attributed to Galindo, Herranz, and Kiltz [21]. An important result from that paper is that IBBS can be constructed from any one-way function, in the standard model. One shortcoming of the result is that it gives long signatures (say,  $6 \cdot 170 = 1020$  bits).

Our first contribution. We propose an IBBS scheme with the following properties:

– It is secure under the well-studied CDH assumption, in the standard model.

- The signature is shorter than that of [21], with almost the same security level. Specifically, it is about  $3 \cdot 170 = 510$  bits, which is half in size compared to the cdh-based IBBS derived from [21].

- The communication cost between the signer and user is about eight times lesser than that of the cdh-based IBBS derived from [21].

Our construction, given in Section 4, is based on the blind HIBE derived from Chatterjee-Sakar HIBE of level 2.

# **1.2** The second application: Hierarchical Blind Decryption Protocol

We first consider the following scenario.

**Scenario.** Consider the following situation in trading encrypted information: over the Internet, a company owns a chain of retailers who sell digital information (e.g., music, magazine) using their websites. The information is available in encrypted form, namely ciphertexts, on the websites. A buyer, after browsing the summaries attached to the ciphertexts, must pay some money to have its choice of ciphertexts decrypted by the corresponding retailers. The problem is: the buyer wishes to hide its choice of purchases, namely he/she wants to have his/her choice of ciphertexts decrypted by the retailers in a blind manner; while the company and retailers want to make sure that the buyer cannot get more than what he/she pays. For example, the buyer cannot obtain v + 1 plaintexts while only paying for v ones. We will however formalize a stronger security notion, ensuring that even one bit of the (v + 1)-th ciphertext is not leaked.

On a higher view, the above problem can be rephrased as follows: a buyer holds ciphertexts which will be decrypted by some decrypters. The buyer wants to hide his choice of ciphertexts while the decrypters want to make sure that the buyer gains nothing more beside the plaintexts the buyer requests. When there is *only one* decrypter, this problem was known as blind decryption (aka, blind decoding) in the literature [32],[24], [31]. The case of many decrypters (controlled by a company), to the best of our knowledge, has not been examined yet before this work.

Even in the case of one decrypter, the blind decryption problem apparently has not had a satisfied solution in standard model yet. (We will explain more about this later.) Solving the problem, in standard model, in the case of many decrypters also settles a solution for the case of one decrypter, since the latter is a special case of the former. These are what we provides in this paper.

**Our second contribution.** We propose a novel mechanism to solve the above problem. We call the mechanism *Hierarchical Blind Decryption Protocol*, or HBDP for short. The description and some highlighted functionalities of HBDP are as follows:

(1) It has a hierarchical structure (like HIBE).

(2) Only leaf identities<sup>1</sup>, which represent the retailers, support blind decryption protocols (with buyers).

(3) Everyone can create ciphertexts the retailers will sell. This capability can capture situations where the retailers are not producers.<sup>2</sup>

(4) A current retailer, a leaf identity, can give up selling and freely set up new "child retailers". We expect this flexibility is extremely welcome once a retailer is overloaded with what needs to sell.

We formalize syntax and security notions for HBDP in Section 5. Our security notions are stronger than those of previous works on blind decryption. We stress that forming the decrypters security is *non-trivial*, and is novel to this work. In particular, we develop the notion of *one-more indistinguishability* for blind decryption, which is symmetric to the well-known one-more unforgeability security for blind signature. Details are in Subsection 5.1. Our construction of HBDP uses as its building block blind HIBE. The construction is generic (namely, from any blind HIBE) and is secure if the building block is secure.

As already observed in [31], in the case of trading encrypted information as above, all the prices of plaintexts must be the same; or otherwise blindness for buyers may be trivially lost since the company can guess what the buyer purchases based on the prices. We stress that although we discuss HBDP in the situation of trading encrypted information anonymously, HBDP (or BDP as a special case) may also be used in other contexts as a building block as well, including the work of [33], [28]. Another interesting idea is in [29], where BDP is used to create a mechanism making data disappear<sup>3</sup>.

**Previous and related works.** As mentioned, the problem of blind decryption has appeared in the literature [32], [24], [31]. These works considered only one retailer (decrypter). No work in the literature has considered the problem when it comes to a chain of retailers instead of one. The work of [31] is in the random oracle model. The work of Mambo, Sakurai, and Okamoto [24], improving [32], is in standard model, but whether the scheme in [24] is secure or not is suggested as an open problem [27]. Specifically, in the Concluding Remarks section of [27], Ohta wrote: "It is an open question whether the scheme proposed in [24] is secure ...". Seemingly, the authors of [24] also realized some trouble in their scheme by writing: "(From this viewpoint)

<sup>&</sup>lt;sup>1</sup>Here and hereafter, we use the term leaf identities to indicate identities who are not a parent identity of any one.

<sup>&</sup>lt;sup>2</sup>This functionality would be also welcome in the following situation: the reliability of the producers is higher than that of the sellers, and the buyers would like to check whether the ciphertexts were from the producers. This can be done by having the producers sign on the ciphertexts using a digital signature scheme. The ciphertexts, together with their signatures, will be on the the sellers' websites so that everyone is able to check their validity *before* purchasing them.

<sup>&</sup>lt;sup>3</sup>We however comment that the work of [29] is apparently not in the area of *provable* cryptography. The claim "blind decryption is novel" in that paper is not right.

the proposed blind decoding scheme is not existentially unforgeable." ([24], line 1, page 329). This is why we previously stated that the problem of blind decryption in the standard model, even in the case of one decrypter, was not settled yet before this work.

One may argue that if the blind decryption problem for one decrypter is solved, then the case of many decrypters is also settled, since one just needs to set up many of independent decrypters in parallel. We agree. However, we expect that this approach is costly, compared to the HBDP approach, since one has to set up independent system parameters for each decrypter in the former. Moreover, the functionality (4) as mentioned above would be lost.

A related concept is oblivious transfer (OT), typically considering two players, i.e., one receiver (user) and one sender (database), which totally differs from HBDP. In particular, the functionalities (3) and (4) above of HBDP are not supported by OT. Setting up many OTs in parallel may solve the problem HBDP considers, but suffers from the same weakness as many BDPs in parallel. OT itself is costly in general due to many zero-knowledge proofs of knowledge, in constrast to our HBDP, which uses only one. Moreover, as will be seen later, our security notions for HBDP are also different from those of OT, while sufficient enough for use.

**Organization.** We begin with some technical preliminaries in Section 2. We then recall HIBE and examine blind HIBE in section 3. Next, we move to our main contributions in Section 4 (about IBBS) and Section 5 (about HBDP). Finally, we conclude this paper in Section 6.

## 2 Technical Preliminaries

We will use essentially the same presentation and wording as [20] in this section and Section 3. Let BMsetup (bilinear map setup) be an algorithm that, on input a security parameter  $\kappa$ , outputs the parameters for a bilinear mapping as  $\gamma = (q, g, G, G_T, e)$ , where g generates  $G, e : G \times G \to G_T$ , and q is the order of G and  $G_T$ . We will need the following complexity assumptions made in these groups.

**Decisional Bilinear Diffie-Hellman (DBDH)** [3]. Let  $\mathsf{BMsetup}(\kappa) \to (q, g, G, G_T, e)$ . For all p.p.t adversaries  $\mathcal{A}$ , the value  $|\Pr[b' = b] - 1/2|$  is negligible in the following experiment:  $x, y, z, t \stackrel{\$}{\leftarrow} Z_q; x_0 \leftarrow e(g,g)^{xyz}; x_1 \leftarrow e(g,g)^t; b \stackrel{\$}{\leftarrow} \{0,1\}; b' \leftarrow \mathcal{A}(g, g^x, g^y, g^z, x_b).$ 

**Computational Diffie-Hellman.** For all p.p.t adversaries  $\mathcal{A}$ , its advantage defined as  $\mathbf{Adv}(\mathcal{A}) = \Pr[g \stackrel{*}{\leftarrow} G; x, y \stackrel{*}{\leftarrow} Z_q : \mathcal{A}(g, g^x, g^y) = g^{xy}]$  is negligible.

Known Discrete-Logarithm-Based, Zero Knowledge Proofs. We use known techniques for proving statements about discrete logarithms, such as (1) proof of knowledge of an element representation in a prime order group [25], (2) proof that a committed value lies in a given integer interval [8], [14], [1], and also (3) proof of conjunction of any of the previous [7]. These protocols are secure under the discrete logarithm assumption, although some implementation of (2) requires the Strong RSA assumption [4], [18].

When referring to the proofs above, we use the notation of Camenisch and Stadler [16]. For instance,  $PoK\{(x, r): y = g^x h^r \land 1 \le x \le n\}$  denotes a zero-knowledge proof of knowledge of integers x and r such that  $y = g^x h^r$  and  $1 \le x \le n$ . All values not

enclosed in ()'s are assumed to be known to the verifier. We can apply Fiat-Shamir heuristic [19] to make such proofs non-interactive in the random oracle model.

# 3 HIBE and blind HIBE

We will use essentially the same presentation as in [20], which might help readers easily realize the differences between blind HIBE and blind IBE [20].

#### 3.1 Definitions

Hierarchical Identity-based Encryption Scheme (HIBE) [22], [23]. A HIBE consists of four algorithms: Setup, Extract, Encrypt, Decrypt. In a HIBE, identities are vectors. A vector of dimension j represents an identity at depth j, denoted as  $ID_{|j} = (I_1, \ldots, I_j)$  where the components  $I_1, \ldots, I_j \in \mathcal{I}$  for some set  $\mathcal{I}$ . The detailed description and functionality of the algorithms are as follows:

- In the Setup $(\kappa, l) \rightarrow (params, msk)$  algorithm, on input a security parameter  $\kappa$  and the maximum depth l of the HIBE, the master authority outputs master parameters and a master secret key (params, msk). params is also the input of all algorithms below, but we omit writing it for the sake of clarity.

- In the Extract( $\mathcal{P}_{ID|j-1}(sk_{ID|j-1}), \mathcal{U}(ID_{|j})) \rightarrow (ID_{|j}, sk_{ID|j})$  protocol<sup>4</sup>, an honest user  $\mathcal{U}$  with identity  $ID_{|j} = (I_1, \ldots, I_j)$  obtains the corresponding secret key  $sk_{ID|j}$ from the parent identity  $ID_{|j-1} = (I_1, \ldots, I_{j-1})$  or outputs an error message. The parent identity output is **the identity**  $ID_{|j}$  or an error message.

- In the Encrypt $(params, ID_{|j}, m) \to C$  algorithm, on input identity  $ID_{|j} \in \mathcal{I}^{j}$  and a message  $m \in \mathcal{M}$ , any party can output a ciphertext C.

- In the  $\mathsf{Decrypt}(sk_{ID|j}, C) \to m$ , on input a ciphertext C, the user with  $sk_{ID|j}$  can output a message  $m \in \mathcal{M}$  or an error message.

**Definition 1** (IND-sID-CPA security for HIBE [11], [12]). Let  $\kappa$ , l be the security parameter and maximum depth of HIBE, and  $\mathcal{M}$  the message space. The HIBE is IND-sID-CPA-secure if every p.p.t adversary  $\mathcal{A}$  has an advantage negligible in  $\kappa$ , l for the following game with a challenger: (1)  $\mathcal{A}$  outputs a target identity ID<sup>\*</sup>. (2) The challenger runs Setup( $\kappa$ , l) to obtain (params, msk) and gives params to  $\mathcal{A}$ . (3)  $\mathcal{A}$ may make polynomially-many key extraction queries ID. In response, the challenger runs the Extract algorithm on input ID to obtain the corresponding private key  $sk_{ID}$ and gives it to  $\mathcal{A}$ . The only restrictions are: ID is not ID<sup>\*</sup> and is not a prefix of ID<sup>\*</sup>. (4)  $\mathcal{A}$  outputs two equal-length messages  $m_0$ ,  $m_1$ . The challenger chooses a random bit b, and gives  $\mathcal{A}$  the challenge ciphertext  $C^* \leftarrow \text{Encrypt}(\text{params}, ID^*, m_b)$ . (5)  $\mathcal{A}$  may continue to make key extraction queries under the same conditions as before. (6)  $\mathcal{A}$ outputs  $b' \in \{0,1\}$ . The advantage of  $\mathcal{A}$  in the game is defined as  $|\Pr[b'=b] - 1/2|$ .

**On IND-ID-CPA security for HIBE.** This notion of security is stronger than the above notion, allowing the adversary to choose a target identity at Step 4 in the above game.

<sup>&</sup>lt;sup>4</sup>We will use the words "algorithm" and "protocol" interchangeably in some contexts.

**Blind HIBE.** A blind HIBE scheme consists the same algorithms Setup, Encrypt, Decrypt as in a traditional HIBE one, but the protocol Extract is replaced by a new protocol BlindExtract:

- In the BlindExtract( $\mathcal{P}_{ID|j-1}(sk_{ID|j-1}), \mathcal{U}(ID_{|j})) \rightarrow (nothing, sk_{ID|j})$  protocol, an honest user  $\mathcal{U}$  with identity  $ID_{|j} = (I_1, \ldots, I_j)$  obtains the corresponding secret key  $sk_{ID|j}$  from the parent identity  $ID_{|j-1} = (I_1, \ldots, I_{j-1})$  or outputs an error message. The parent identity output is **nothing** or an error message.

We now define security for blind HIBE, which is informally any IND-sID-CPA HIBE scheme with a BlindExtract protocol satisfying two below properties:

1. Leak-free Extract [20]: a potentially malicious user cannot learn anything by executing the BlindExtract protocol with a parent identity which she could not have learned by executing the Extract protocol with the parent identity; moreover, as in Extract the user must know the identity for which she is extracting the key.

2. Selective-failure Blindness [15]: a potentially malicious parent identity cannot learn anything about the user's choice of identity during the BlindExtract protocol (more than what it has already known before BlindExtract); moreover, the parent identity cannot cause the BlindExtract protocol to fail in a manner dependent on the user's choice of identity.

The formal definitions of leak-freeness and selective-failure blindness are in A. We finally arrive at the following definition.

**Definition 2** (Secure Blind HIBE). A blind HIBE  $\Pi = (Setup, BlindExtract, Encrypt, Decrypt)$  is IND-sID-CPA-secure if and only if: (1) its corresponding HIBE is IND-sID-CPA secure, and (2) BlindExtract is leak-free and selective-failure blind.

#### 3.2 HIBE Schemes with Efficient BlindExtract Protocols

In this section, we describe efficient BlindExtract protocols for: (1) the IND-sID-CPAsecure HIBE due to Boneh and Boyen [2] and (2) the IND-ID-CPA-secure HIBE proposed by Chatterjee and Sakar [17]. Since these schemes share a similar structure, we begin by describing their common parts.

- Setup $(\kappa, l)$ : Let  $\gamma = (q, g, G, G_T, e)$  be the output of BMsetup $(\kappa)$ . Choose  $\alpha \stackrel{\$}{\leftarrow} Z_q$ , and set  $g_1 \leftarrow g^{\alpha}$ . Choose  $g_2, h_1, \ldots, h_l \stackrel{\$}{\leftarrow} G$ . Select functions  $F_k : \mathcal{I} \to G$  for  $1 \leq k \leq l$ . (The descriptions of  $F_k$  will be defined specific to the schemes below.) Output params =  $(\gamma, g_1, g_2, h_1, \ldots, h_l, F_k)$  and  $msk = g_2^{\alpha}$ .

- Extract( $\mathcal{P}_{j-1}(sk_{ID|j-1}), \mathcal{U}(ID_{|j})$ )  $\rightarrow (ID_{|j}, sk_{ID|j})$ : The private key of identity  $ID_{|j} = (I_1, \ldots, I_j)$  is of the form  $sk_{ID|j} = (g_2^{\alpha}\Pi_{k=1}^j F_k(I_k)^{r_k}, g^{r_1}, \ldots, g^{r_j})$ . Such a private key can be generated by  $\mathcal{P}_{j-1}$  as follows: let  $sk_{ID|j-1} = (d_0, \ldots, d_{j-1})$ .  $\mathcal{P}_{j-1}$  picks random  $r_j \stackrel{\$}{\leftarrow} Z_q$  and sets

$$sk_{ID|j} = (d_0F_j(I_j)^{r_j}, d_1, \dots, d_{j-1}, g^{r_j}).$$

 $-\operatorname{\mathsf{Encrypt}}(params, ID = (I_1, \dots, I_j), m) \to C: t \xleftarrow{\$} Z_q, C \leftarrow (C_0 = m \times e(g_1, g_2)^t, C_1 = g^t, B_1 = F_1(I_1)^t, \dots, B_j = F_j(I_j)^t)$ 

$$\frac{\mathcal{P}_{j-1}(sk_{ID|j-1} = (d_0, \dots, d_{j-1}))}{1. \ y \stackrel{\$}{\leftarrow} Z_q.} \\
\frac{\mathcal{U}(ID = (I_1, \dots, I_j))}{1. \ y \stackrel{\$}{\leftarrow} Z_q.} \\
2. \text{ Compute } h' \leftarrow g^y g_1^{I_j} \text{ and send } h' \text{ to } \mathcal{P}_{j-1}. \\
3. \text{ Execute } PoK\{(y, I_j): h' = g^y g_1^{I_j}\}. \\
4. \text{ If the proof fails to verify, abort.} \\
5. \text{ Choose } r_j \stackrel{\$}{\leftarrow} Z_q. \\
6. \text{ Compute } d'_0 \leftarrow d_0(h'h_j)^{r_j} \text{ and } d'_j \leftarrow g^{r_j} \\
7. \text{ Send } (d'_0, d_1, \dots, d_{j-1}, d'_j) \text{ to } \mathcal{U}. \\
8. \text{ Check} \\
e(d'_0, g) = e(g_2, g_1)e(h'h_j, d'_j)\Pi_{k=1}^{j-1}e(F_k(I_k), d_k). \\
9. \text{ If the check passes, choose } z \stackrel{\clubsuit}{\leftarrow} Z_q; \\
\text{ otherwise output } \bot \text{ and abort.} \\
10. \text{ Output} \\
sk_{ID|i} \leftarrow (d'_0(d'_i)^{-y}F_i(I_j)^z, d_1, \dots, d_{j-1}, d'_i g^z).
\end{aligned}$$

Figure 1: BlindExtract protocol for Boneh-Boyen HIBE. For Chatterjee-Sakar HIBE, modify as follows: Parse  $I_j$  as  $I_j = I_j[1] \cdots I_j[n]$ . In line 2, compute  $h' \leftarrow g^y \prod_{i=1}^n u_i^{I_j[i]}$ . In line 3, execute  $PoK\{(y, I_j = I_j[1] \dots I_j[n]) : h' = g^y \prod_{i=1}^n u_i^{I_j[i]} \land 0 \le I_j[i] < 2^{n'} \forall 1 \le i \le n\}$ .

-  $\mathsf{Decrypt}(sk_{ID|j}, C) \to m$ : To decrypt a given ciphertext  $C = (C_0, C_1, B_1, \ldots, B_j)$ using the private key  $sk_{ID|j} = (d_0, \ldots, d_j)$ , output

$$m = C_0 \frac{\prod_{k=1}^{j} e(B_k, d_k)}{e(d_0, C_1)}$$

We proceed to describe the precise format of the private keys and corresponding BlindExtract protocols for particular HIBEs.

#### 3.2.1 A BlindExtract Protocol for Boneh-Boyen HIBE

In the Boneh-Boyen HIBE [2],  $\mathcal{I} = Z_q$  and the function  $F_k : \mathcal{I} \to G$  is defined as  $F_k(I) = h_k \cdot g_1^I$  for  $I \in \mathcal{I}$  and  $1 \leq k \leq l$ . The private key for identity  $ID_{|j}$  is thus

$$sk_{ID|j} = (g_2^{\alpha} \cdot \prod_{k=1}^{j} (h_k \cdot g_1^{I_k})^{r_k}, g^{r_1}, \dots, g^{r_j}).$$

The protocol BlindExtract for this HIBE is described in Figure 1.

Let  $\Pi_1$  be the blind HIBE that combines the algorithms Setup, Encrypt, Decrypt and the protocol BlindExtract in Figure 1.

**Theorem 3.** The protocol BlindExtract in Figure 1 is both leak-free and selective-failure blind. As a result, the blind HIBE  $\Pi_1$  is IND-sID-CPA-secure (according to Definition 2) under the DBDH assumption.

The proof of this theorem, which is similar to its IBE counterpart in [20], is given in Appendix B.1.

#### 3.2.2 A BlindExtract Protocol for Chatterjee-Sakar HIBE

In the HIBE proposed by Chatterjee and Sakar [17], the set  $\mathcal{I}$  is the set of bit strings of length N, where  $N(=n \cdot n')$  is polynomial in  $\kappa$ , represented by n blocks of n' bits. Define the function  $F_k(I) = h_k \cdot \prod_{i=1}^n u_i^{I[i]}$  for  $1 \le k \le l$ , where  $I[1], \ldots, I[n]$  are n'-bit segments of  $I \in \mathcal{I}$ , and  $u_1, \ldots, u_n \in G$  are randomly chosen by the master authority in **Setup** algorithm. The private key for identity  $ID_{|i|}$  is thus

$$sk_{ID|j} = (g_2^{\alpha} \cdot \prod_{k=1}^{j} (h_k \cdot \prod_{i=1}^{n} u_i^{I_k[i]})^{r_k}, g^{r_1}, \dots, g^{r_j}).$$

The protocol BlindExtract is described in Figure 1, with the following alterations. Parse  $I_j$  as  $I_j = I_j[1] \cdots I_j[n]$ . In line 2, compute  $h' \leftarrow g^y \prod_{i=1}^n u_i^{I_j[i]}$ . In line 3, execute  $PoK\{(y, I_j = I_j[1] \dots I_j[n]) : h' = g^y \prod_{i=1}^n u_i^{I_j[i]} \land 0 \leq I_j[i] < 2^{n'} \forall 1 \leq i \leq n\}$ . Follows the rest of the protocol as it is.

Let  $\Pi_2$  the blind HIBE that combines Setup, Encrypt, Decrypt with the BlindExtract protocol described above.

**Theorem 4.** The protocol BlindExtract in this subsection is both leak-free and selectivefailure blind. As a result, the blind HIBE  $\Pi_2$  is IND-ID-CPA-secure (according to Definition 2) under the DBDH assumption.

The proof of this theorem, which is similar to its IBE counterpart in [20], is given in Appendix B.2.

# 4 The first application: Shorter IBBS scheme in the standard model

We first recap syntax and security notions for IBBS, and then move to our construction.

#### 4.1 Syntax and security notions

**Syntax.** An IBBS scheme  $\mathsf{IBBS} = (\mathsf{IBBS}.\mathsf{Setup}, \mathsf{IBBS}.\mathsf{Extract}, \mathsf{IBBS}.\mathsf{Sign}, \mathsf{IBBS}.\mathsf{Vrf})$  is described as follows.

- IBBS.Setup( $\kappa$ )  $\rightarrow$  (params, msk): The master authority uses this algorithm to produce system-wide parameters params and master-secret key msk. params is also the input of all algorithms below, but we omit writing it for the sake of clarity.

- IBBS.Extract( $\mathcal{P}(msk), \mathcal{S}(I \in \mathcal{I})$ )  $\rightarrow (I, sk_I)$ : The master authority  $\mathcal{P}$  and the signer  $\mathcal{S}$  of identity I interact so that, at the end,  $\mathcal{S}$  receives the private key  $sk_I$ , while  $\mathcal{P}$  knows  $\mathcal{S}$  identity.

- IBBS.Sign $(\mathcal{S}(sk_I), \mathcal{U}(I, m)) \to (\text{nothing}, \sigma)$ : the signer  $\mathcal{S}$  with identity I interacts with the user  $\mathcal{U}$  who wants to obtain a blind signature on the message m. At the end,  $\mathcal{S}$  knows nothing about m, while  $\mathcal{U}$  gets the blind signature  $\sigma$ .

- IBBS.Vrf $(I, m, \sigma) \rightarrow 0/1$ : Everyone can check a message-signature pair  $(m, \sigma)$  under identity I is valid or not, using this algorithm.

**Security notions.** There are two security notions for an IBBS scheme: *blindness* and *one-more unforgeability*, which are informally discussed as follows.

Blindness ensures that the signer know nothing about the message of the user even after the signing process. We will consider selective-failure blindness for IBBS, which furthermore captures the idea that the signer cannot cause IBBS.Sign to fail in a manner dependent on the message. This notion is stronger than that of [21]. The precise formulation of selective-failure blindness is very similar to that of blind HIBE in A. We omit that formulation, since the selective-failure of our below construction comes directly from that of the corresponding blind HIBE.

One-more unforgeability ensures that the forger  $\mathcal{F}$  (malicious user), after  $q_s$  signing queries, cannot obtain strictly more than  $q_s$  valid signatures (say,  $q_s + 1$  signatures). In the identity-based setting,  $\mathcal{F}$  can have access for different  $q_e$  times to a key extract oracle. We also omit the details, and refer the readers to [21] for the formal definition, since we will not directly use it.

#### 4.2 Our construction

**Intuition.** A well-known fact, attributed to Naor by Boneh and Franklin<sup>5</sup>, is that one can construct a standard signature scheme from an IBE. By the same token, from level-2 (IND-ID-CPA) HIBE, identity-based signature (IBS) can be built [22]. This idea was used by Paterson and Schuldt in [30] to construct an efficient cdh-based IBS in the standard model from Chatterjee-Sakar HIBE of level 2. Our observation is that from level-2 (IND-ID-CPA) *blind* HIBE, we can have a secure IBBS scheme. We will use the IND-ID-CPA blind HIBE in subsubsection 3.2.2.<sup>6</sup>

**Our construction.** Our IBBS = (IBBS.Setup, IBBS.Extract, IBBS.Sign, IBBS.Vrf) is construct as follows.

- IBBS.Setup( $\kappa$ ): We run  $\Pi_2$ .Setup( $\kappa$ , 2). Note that we take the level l = 2. The outputs are system-wide parameter  $params = (\gamma, g_1, g_2, h_1, h_2, F_1, F_2)$  and the master-secret key  $msk = g_2^{\alpha}$ . Recall that  $F_1, F_2 : \mathcal{I} \to G$ , and  $\mathcal{I} = \{0, 1\}^{n \cdot n'}$ . The signing space of our IBBS is also  $\mathcal{I}$ .

- IBBS.Extract( $\mathcal{P}(msk)$ ,  $\mathcal{S}(I \in \mathcal{I})$ ): We run the Extract( $\mathcal{P}(msk)$ ,  $\mathcal{S}(I)$ ) algorithm related to  $\Pi_2$  described in Subsection 3.2. The signer  $\mathcal{S}$  with the identity I will receive  $sk_I = (g_2^{\alpha} F_1(I)^r, g^r)$  as his private key.

- IBBS.Sign( $S(sk_I), U(I, m)$ ): We run the interactive protocol  $\Pi_2$ .BlindExtract( $S(sk_I), U(ID = (I, m) \in \mathcal{I}^2)$ ). Note that m is treated as a part of the identity ID. As the results of the protocol, S outputs nothing, while  $\mathcal{U}$  receives a value of the form  $(g_2^{\alpha}F_1(I)^rF_2(m)^s, g^r, g^s)$ , which is his signature on m.

- IBBS.Vrf $(I, m, \sigma = (\sigma_1, \sigma_2, \sigma_3))$ : Using the pairing e, check that  $e(\sigma_1, g) = e(g_1, g_2) \cdot e(F_1(I), \sigma_2) \cdot e(F_2(m), \sigma_3)$ . Return 1 if the check passes, else return 0.

We proceed to examine securities of the above construction. Selective-failure blindness of our IBBS comes directly from that of  $\Pi_2$ .BlindExtract, since in the IBBS.Sign protocol, only BlindExtract is executed. We formally state this fact in the following theorem.

**Theorem 5.** The above construction of IBBS satisfies selective-failure blindness.

We next consider one-more unforgeability, which is ensured by the following theorem.

<sup>&</sup>lt;sup>5</sup>See [9] for a careful treatment.

<sup>&</sup>lt;sup>6</sup>Note that the blind HIBE in subsubsection 3.2.1 is irrelevant in this case since we need IND-ID-CPA security.

IBBS scheme	sig. size	fac. loss, assum.	com. cost
[21] + [26] + [5]	$6 \cdot  q $	$O(q_e q_s), \operatorname{cdh}$	161 var.
Ours	$3 \cdot  q $	$O((q_e + q_s)q_s), \operatorname{cdh}$	21 var.

Figure 2: Comparisons between our IBBS and the cdh-based IBBS derived from [21], using concrete constructions in [26], [5]. In the figure, sig.= signature, fac.= factor, assum.= assumption, com.= communication, var.= variables.

**Theorem 6.** The IBBS given above satisfies one-more unforgeability. Specifically, given a forger  $\mathcal{F}$  against the IBBS, then there exist adversaries  $\mathcal{A}_1$  (against leak-freeness) and  $\mathcal{A}_2$  (against the CDH assumption) whose running times are essentially the same as that of  $\mathcal{F}$ , and

 $\mathbf{Adv}_{\textit{IBBS}}^{om-forge}(\mathcal{F}) \leq q_s \mathbf{Adv}_{\Pi_2}^{leak-free}(\mathcal{A}_1) + 16(2^{n'-1}n+1)^2(q_e+q_s)q_s \mathbf{Adv}_G^{cdh}(\mathcal{A}_2),$ 

where  $q_e$  is the number of extract queries and  $q_s$  is the number of signing queries  $\mathcal{F}$  makes.

The proof is intuitively as follows. From the viewpoint of  $\mathcal{F}$ , the protocol BlindExtract in IBBS.Sign is essentially identical to the corresponding Extract protocol, thank to the leak-freeness of  $\Pi_2$ . Consequently, in simulating the environment for  $\mathcal{F}$ , we can safely replace the protocol BlindExtract in IBBS.Sign by the protocol Extract. This change induces the first term on the right hand-side of the above inequality. Now that  $\mathcal{F}$  can be seen as a forger of an IBS (not blind), the rest of the simulation for  $\mathcal{F}$  will be the same as that of Paterson and Schuldt [30], who in turn use Waters' technique [34]. We omit further details.

The comparison between our IBBS and the (only known) cdh-based one in the standard model is depicted in Fig. 2. Below is the explanation for that figure.

**Concrete parameters.** Typically, the signing and identity space  $\mathcal{I} = \{0, 1\}^{160}$ , which is also the output range of SHA-1. Thus we take  $n \cdot n' = 160$ . Due to the term  $2^{n'-1}$  in security reduction, we suggest to take  $n' - 1 \approx \log_2(n \cdot n')$  so that  $2^{n'-1}(=n \cdot n')$  is polynomial. Thus we recommend n' = 8 and n = 20. The reduction loss in the above theorem is considered as  $O((q_e + q_s)q_s)$ .

The communication cost between signer and user in IBBS.Sign is mainly incurred by the following zero-knowledge proof of knowledge:  $PoK\{(y, m = m[1] \dots m[20]) :$  $(h' = g^y \prod_{i=1}^{20} u_i^{m[i]}) \land (0 \le m[i] < 2^8) \forall 1 \le i \le 20\}$ . Namely, the user has to make a proof of knowledge for 21 variables  $(y, m[1] \dots m[20])$ .

The signature size of our IBBS is three elements of G, so the size is  $3 \cdot |q|$ . (Here, |q| is the bit length of q.) Typical value for |q| is 170. The discussion thus far illustrates our IBBS in Fig. 2.

We now explain the second line of Fig. 2, which is about the cdh-based IBBS resulted from [21], [26], [5]. The paper [21] gave a generic construction in the standard model of IBBS from any ordinary blind signature BS and strongly-secure signature S. Thus, to build a cdh-based IBBS, both BS and S must be cdh-based secure. The only known cdh-based S strongly secure in the standard model is in [5], and the cdh-based secure BS is in [26] (Section 10). Both results in [5] and [26] are based on the cdh-based Waters' signature scheme [34]. The signature size of S in [5] is  $3 \cdot |q|$ , and that of BS

in [26] is  $2 \cdot |q|$ . The signature size of the generic construction [21] is that of S, and BS, and one group element of length |q|, so that  $6 \cdot |q|$  is the total signature size of the IBBS. The loss factors in security reduction to the CDH assumption of S and BS are  $O(q_s)$ . The loss in the generic construction is of  $O(q_e)$ , and hence the total loss factor for the IBBS is  $O(q_eq_s)$ , which is a bit better than ours. The communication cost of BS is about 161 variables<sup>7</sup>, so is the derived IBBS.

# 5 The second application: Hierarchical Blind Decryption Protocol

In this section, we first formalize the syntax and security for HBDP, and then present our construction (from blind HIBE) of HBDP. This section is considered as the second main contribution of this paper.

#### 5.1 Syntax and Security Definitions

**Syntax.** A HBDP is specified by three algorithms **Setup**, **Extract**, **Enc**, and a protocol BlindDec. The description and functionality of **Setup** and **Extract** are the same as those of HIBE. Namely, **Setup** is used to create system-wide parameters *params* and master secret key *msk*; while **Extract** is used by a parent identity to create private key for its child identities. The algorithm **Extract** supports all identities in the HBDP. The formal syntax descriptions of **Setup** and **Extract** are the same as those of HIBE, so we omit them here.

Denote Leaves the set of leaf identities, which depends on the (current) structure of the HBDP. The protocol BlindDec supports only leaf identities of the HBDP. The algorithm Enc requires as one of its input a leaf identity. The formal description and functionality of the two new algorithms are as follows:

 $- \operatorname{Enc}(params, ID \in \operatorname{Leaves}, m) \to C$ : On inputs *params*, a leaf identity *ID*, and a message *m*, Enc produces a ciphertext *C*. The purpose of this algorithm is to create ciphertexts which the retailers (leaf identities) will decrypt. We note that not only the retailers but also any party can use this algorithm.

- BlindDec( $\mathcal{P}_{ID}(sk_{ID}), \mathcal{U}(C)$ )  $\rightarrow$  (nothing, m): When an honest buyer  $\mathcal{U}$  wants to buy the content inside a ciphertext C which is available from  $\mathcal{P}_{ID}$  where  $ID \in \text{Leaves}$ , she engages  $\mathcal{P}_{ID}$  in a BlindDec protocol. At the end,  $\mathcal{U}$  receives the content m while  $\mathcal{P}_{ID}$  knows nothing about the choice of the ciphertext. We emphasize that only leaf identities of the HBDP supports BlindDec (with buyers).

In the Introduction section, we stated that we also solved the problem of blind decryption with one decrypter. We now elaborate on that statement. Consider a special case of HBDP where the set Leaves contains only one element – the root identity. (Note that the algorithm Extract is not used at all in this case, or otherwise the root identity will not be a leaf identity any more.) Recall that a leaf identity represents a decrypter (retailer), so that we have only one decrypter in this case. This is exactly the case that previous works on blind decryption considered.

<sup>&</sup>lt;sup>7</sup>Note that here we consider the message length of BS [26] is 160 bits, which is typical.

We proceed to define security notions for HBDP. There are two issues to be considered: the privacy of buyer (namely, blindness), and the security of retailers.

1. Blindness. Blindness of HBDP ensures the privacy of the buyer. Intuitively, it ensures that the retailers cannot know more about the buyer's choice of ciphertexts after the BlindDec protocol than what it has already known before the BlindDec protocol. We will not provide the formal definition for blindness here, since the blindness of our construction of HBDP comes directly and obviously from the blindness of the underlying blind HIBE. We believe that doing so makes our presentation clearer.

2. One-More Indistinguishability (IND-OM). IND-OM security intuitively ensures that it is impossible for a (potentially malicious) buyer to get more than what he/she paid. We will use the one-more security approach, and yet, in an indistinguishability style. This notion itself is new to this paper. Let us first present some intuitions behind the notion, and then move to the precise description. Recall that a malicious user, an adversary, can collect ciphertexts available from the websites of the retailers. We capture this ability by providing the adversary an encryption algorithm, in a left-or-right style. The adversary can also engage the retailers in BlindDec protocols, so that BlindDec oracles are given. At the end, the adversary wins if after vtimes of querying the BlindDec oracles, it knows one-more bit used in the left-or-right encryption oracles; namely, it knows non-negligible information about the content of some ciphertext for which it did not pay. Conversely, a HBDP is ind-om-secure if no p.p.t adversary can win. Note that ind-om security also implies that the adversary cannot decrypt a (v + 1)-th ciphertext after v times of accessing to the BlindDec oracles. In other words, ind-om security implies one-way one-more (OW-OM) security, which was considered in [24] under the name "existential unforgeability<sup>8</sup>". We do not give a formal definition of OW-OM security here because our construction meets the stronger security notion, IND-OM. The formal definition of IND-OM security is as follows.

**Definition 7** (IND-OM security for HBDP). Let  $\Pi_{bdp} = (Setup, Extract, Enc, Blind-Dec)$  be a HBDP. Consider the following game between a challenger and an adversary  $\mathcal{A}$ .

**Setup:** The challenger runs  $Setup(\kappa, l)$  to obtain params and msk, and gives params to A.

Denote Leaves =  $\{ID_1, \ldots, ID_k\}(k \ge 1)$  the set of leaf identities of the HBDP, which is given to A.

**Queries:** the adversary is allowed to make the following types of queries:

- Encryption query<sup>9</sup>  $(ID_j \in Leaves, m_0^{(i)}, m_1^{(i)})$ : the challenger chooses a random bit, denoted as  $b_{(j,i)} \stackrel{*}{\leftarrow} \{0,1\}$ , computes  $C_{(j,i)} \leftarrow Enc(params, ID_j, m_{b_{(j,i)}}^{(i)})$ , and then returns  $C_{(j,i)}$  to  $\mathcal{A}$ . The adversary is permitted to make  $u_j$  encryption queries to  $ID_j$ , thus  $1 \leq i \leq u_j$ .

- Blind decryption query: the adversary  $\mathcal{A}$  engages  $\mathcal{P}_{ID}(sk_{ID})$  in a BlindDec( $\mathcal{P}_{ID}(sk_{ID})$ ,  $\mathcal{A}(C)$ ) protocol for some  $ID \in Leaves$  and some ciphertext C.  $\mathcal{A}$  can engage many BlindDec protocols with the leaf nodes in an arbitrary manner. The total number of permitted blind decryption queries is denoted as  $v \ (< u_1 + \cdots + u_k)$ .

<sup>&</sup>lt;sup>8</sup>This is originally used for signature security. We think that OW-OM is a more proper name.

<sup>&</sup>lt;sup>9</sup>Note that the adversary  $\mathcal{A}$  can also obtain the ciphertext of a message m under a leaf identity  $ID_j$  by making an encryption query of the form  $(ID_j, m, m)$ .

**Output:** A outputs v + 1 bits  $b'_1, \ldots, b'_{v+1}$  and an injective map  $\pi : \{1, \ldots, v+1\} \rightarrow \{1, \ldots, k\} \times \{1, \ldots, u\}$ , where  $u = \max\{u_1, \ldots, u_k\}$ . Denote the above adversary as  $\mathcal{A}(u_1, \ldots, u_k, v)$ .

We say that  $\mathcal{A}$  wins if  $b'_i$  is the right guess of  $b_{\pi(i)}$  for all  $1 \leq i \leq v+1$ . Define the advantage of  $\mathcal{A}$  as

$$\mathbf{Adv}_{\mathsf{HBDP}}^{ind-om}(\mathcal{A}) = \Pr[b'_i = b_{\pi(i)} \forall 1 \le i \le v+1] - \frac{1}{2}.$$

If the advantage is either negligible or negative for all p.p.t adversaries  $\mathcal{A}$ , we say that the HBDP is ind-om-secure.

**Remarks and Discussions about IND-OM.** Firstly, we emphasize that, unusually, the absolute value cannot be taken when defining the above advantage function, and one has to consider adversaries with *negative* advantage. In fact, if the absolute value is taken, then an adversary just returning random bits has advantage  $|1/2^{v+1}-1/2| \approx 1/2$  for large v, which is not negligible at all.

Secondly, consider again the situation between the retailers and the buyer as in Section 1. We note that, in practice, some summaries and/or keywords of the contents inside the ciphertexts should also be available on the websites so that the buyer can make choices. The IND-OM security, like other ind-securities (e.g., ind-cca), ensures that even so, the adversary still has negligible knowledge about the other parts of the contents. Furthermore, our IND-OM notion is stronger than the one-more decryption security considered in [31]. In fact, our notion allows the adversary to adaptively encrypt messages of its own choice, while that of [31] does not. All messages in the notion of [31] are randomly chosen, and the corresponding ciphertexts are non-adaptively given to the adversary at the beginning.

Thirdly, as in HIBE schemes, one may want to think of a situation where there is a coalition among some identities in HBDP in order to compromise the privacy of the other. This issue can be addressed by additionally giving the above adversary  $\mathcal{A}$  oracle calls to Extract with a natural restriction that the target leaf identity (and its parent identities) is not queried to that oracle. However, for the time being, we will not consider this security model, although our construction of HBDP (in the next subsection) would also remain secure. We believe that the current security model, as formally described above, is sufficient enough to reflect situations as mentioned in the motivation of this work.

#### 5.2 The construction of HBDP

In this section, we show how to turn any HIBE supporting blind key extract (namely, blind HIBE), into a HBDP. Let  $\Pi = (Setup, Extract, Encrypt, Decrypt)$  be a HIBE and  $\Pi' = (Setup, BlindExtract, Encrypt, Decrypt)$  its corresponding blind HIBE. Our construction of HBDP reuses the algorithms Setup and Extract of  $\Pi$  without any change. Thus to build the HBDP  $\Pi_{bdp} = (Setup, Extract, Enc, BlindDec)$ , we just have to construct the Enc algorithm and the BlindDec protocol. The constructions of the algorithms are as follows.

#### The Constructions of Enc and BlindDec

- $\operatorname{Enc}(params, ID \in \operatorname{Leaves}, m): I \stackrel{\$}{\leftarrow} \mathcal{I}, \stackrel{10}{ID'} \leftarrow (ID, I), \hat{C} \stackrel{\$}{\leftarrow} \operatorname{Encrypt}(params, ID', m). \text{ Output } C \leftarrow (ID', \hat{C}).$
- BlindDec( $\mathcal{P}_{ID}(sk_{ID}), \mathcal{U}(C)$ ):  $\mathcal{U}$  first parses C as  $(ID', \hat{C})$ .  $\mathcal{U}$  then runs (with  $\mathcal{P}_{ID}$ ) the protocol BlindExtract( $\mathcal{P}_{ID}(sk_{ID}), \mathcal{U}(ID')$ )  $\rightarrow$  (nothing,  $sk_{ID'}$ ).  $\mathcal{U}$  finally outputs Decrypt( $sk_{ID'}, \hat{C}$ ).

In the construction, one can intuitively imagine a leaf identity (a retailer) as a "parent identity" of its ciphertexts, and a user bought a ciphertext (namely, bought a private key as seen in our construction) becomes a "child identity" of the leaf identity. The privacy of the buyer is ensured via the blindness of BlindExtract protocol. The IND-OM security for the retailers comes from the security of the underlying HIBE and blind HIBE schemes: the coalition of the "child identities" (the private keys the user bought) cannot compromise the privacy of the others.

We proceed to formally consider the securities of the above construction. We first consider blindness. A user only engages a retailer in BlindExtract protocols, so that blindness of  $\Pi_{bdp}$  is obvious from the blindness property of BlindExtract. We formally state this fact in the following theorem.

**Theorem 8.**  $\Pi_{bdp}$  satisfies selective-failure blindness.

We now move to the ind-om security, which is ensured by the following theorem.

**Theorem 9.** Let  $\mathcal{A}(u_1, \ldots, u_k, v)$  be an ind-om adversary against the HBDP  $\Pi_{bdp}$ . Then there exist a distinguisher  $\mathcal{A}_1$  (against the leak-freeness of  $\Pi'$ ) and an adversary  $\mathcal{A}_2$  (against the IND-sID-CPA security of  $\Pi$ ), whose resources are essentially the same as that of  $\mathcal{A}$  such that

$$\mathbf{Adv}_{\Pi_{bdp}}^{ind-om}(\mathcal{A}) \leq \mathbf{Adv}_{\Pi'}^{leak-free}(\mathcal{A}_1) + ku\mathbf{Adv}_{\Pi}^{ind-sid-cpa}(\mathcal{A}_2),$$

where  $u = \max\{u_1, ..., u_k\}.$ 

As mentioned, the intuition behind its proof is as follows: the coalition of some child identities of the leaf identities cannot affect other child identities. The formal proof, which is quite technical, is in C.

## 6 Conclusion

We have considered blind HIBE schemes and used them to construct IBBS and HBDP. Our IBBS scheme enjoys many merits over existing proposals such as: cdh-based security in the standard model, shorter signature, and lesser communication cost. Our second contribution, which is the HBDP schemes, is useful in the situations where a party holds some ciphertexts which will be decrypted by some decrypters in a blind manner. An example of its use is trading encrypted information anonymously, but HBDP can serve as a building block for other applications as well, as in [33], [28], [29]. We expect further applications of HBDP as well as blind HIBE in the future.

<sup>&</sup>lt;sup>10</sup>We assume that  $\mathcal{I}$  is big enough so that collision probability in choosing the value I is negligible.

# References

- F. Boudot. Efficient Proofs that a Committed Number Lies in an Interval. EU-ROCRYPT 2000, 431–444, 2000.
- [2] D. Boneh and X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In Advances in Cryptology (EUROCRYPT 2004), Springer LNCS 3027, 223–238, 2004.
- [3] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. In J. Kilian, editor, Proceedings of Crypto 2001, volume 2139 of LNCS, 213–229. Springer-Verlag, 2001.
- [4] N. Baric and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In Proc. EUROCRYPT '97, volume 1233 of Lecture Notes in Computer Science, 480–494. Springer–Verlag, 1997.
- [5] D. Boneh, E. Shen, and B. Waters. Strongly Unforgeable Signatures Based on Computational Diffie-Hellman. Public Key Cryptography 2006, 229-240, 2006.
- [6] D. Chaum. Blind signatures for untraceable payments. Advances in Cryptology -Crypto '82, Springer-Verlag (1983), 199-203.
- [7] R. Cramer, I. Damgard, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Y. G. Desmedt, editor, Advances in Cryptology — CRYPTO '94, volume 839 of Lecture Notes in Computer Science, 174–187. Springer Verlag, 1994.
- [8] A. Chan, Y. Frankel, and Y. Tsiounis. Easy come easy go divisible cash. In K. Nyberg, editor, Advances in Cryptology — EUROCRYPT '98, volume 1403 of Lecture Notes in Computer Science, 561–575. Springer Verlag, 1998.
- [9] Y. Cui, E. Fujisaki, G. Hanaoka, H. Imai, R. Zhang. Formal Security Treatments for IBE-to-Signature Transformation: Relations among Security Notions. Cryptology ePrint Archive, Report 2007/030. Available at http://eprint.iacr.org/2007/030.
- [10] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In 30th Annual ACM Symposium on Theory of Computing, pages 209-218, Dallas, Texas, USA, May 23-26, 1998. ACM Press.
- [11] R. Canetti, S. Halevi, and J. Katz. A Forward-Secure Public-Key Encryption Scheme. Adv. in Cryptology – Eurocrypt 2003, LNCS vol. 2656, Springer-Verlag, 255–271, 2003.
- [12] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identitybased encryption. In Advances in Cryptology—EUROCRYPT '04, volume 3027 of LNCS, 207–222. Springer-Verlag, 2004.
- [13] S. S. M. Chow, L. C.K. Hui, S. M. Yiu, and K. P. Chow. Two improved partially blind signature schemes from bilinear pairings. In Proceedings of ACISP 2005, pages 316-325, 2005.
- [14] J. Camenisch and M. Michels. Proving in Zero-Knowledge That a Number Is the Product of Two Safe Primes. In Eurocrypt '99, LNCS 1592, 107–122 Springer-Verlag, 1999.

- [15] Jan Camenisch, Gregory Neven, and abhi shelat. Simulatable Adaptive Oblivious Transfer. In M. Naor, editor, Advances in Cryptology - EUROCRYPT 2007, volume 4515 of Lecture Notes in Computer Science, 573–590 Springer-Verlag, 2007.
- [16] J. Camenisch and M. Stadler. Efficient Group Signature Schemes for Large Groups In Advances in Cryptology–CRYPTO '97, 410–424, 1997.
- [17] S. Chatterjee and P. Sarkar. HIBE With Short Public Parameters Without Random Oracle. ASIACRYPT 2006, 145–160, 2006.
- [18] E. Fujisaki and T. Okamoto. Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations. CRYPTO 1997, 16–30, 1997.
- [19] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In CRYPTO '86, volume 263 of LNCS, 186–194, 1986.
- [20] M. Green and S. Hohenberger. Blind Identity-Based Encryption and Simulatable Oblivious Transfer. ASIACRYPT 2007, 265–282. Available at http://eprint.iacr.org/2007/235.
- [21] David Galindo, Javier Herranz, Eike Kiltz. On the Generic Construction of Identity-Based Signatures with Additional Properties. ASIACRYPT 2006, 178-193.
- [22] C. Gentry, A. Silverberg. Hierarchical ID-based cryptography, Advances in Cryptology – Asiacrypt'02, LNCS 2501, Springer-Verlag, 548–566, 2002.
- [23] J. Horwitz and B. Lynn. Toward Hierarchical Identity-Based Encryption. Advances in Cryptology – Eurocrypt 2002, LNCS vol. 2332, Springer-Verlag, 466– 481, 2002.
- [24] M. Mambo, K. Sakurai, E. Okamoto. How to Utilize the Transformability of Digital Signatures for Solving the Oracle Problem. ASIACRYPT 1996, 322–333, 1996.
- [25] T. Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In Crypto '92, LNCS 740, 31–53. Springer-Verlag, 1992.
- [26] T. Okamoto. Efficient Blind and Partially Blind Signatures Without Random Oracles. In TCC 2006, 80-99, 2006.
- [27] K. Ohta. Remarks on Transformable Digital Signatures. IEICE TRANSAC-TIONS on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E81-A, No.5, 814–817, 1998.
- [28] H. Ota and F. Sugaya. Proposal of an anonymous auction scheme using hierarchical blind decryption based on the RSA encryption. Proceedings of the Society Conference of IEICE, The Institute of Electronics, Information and Communication Engineers, 2003.
- Perlman. The Ephemerizer: [29] R. Making Data Disappear. Sun Mi-Technical Report No. TR-2005-140, 2005.Available crosystems at http://research.sun.com/techrep/2005/abstract-140.html
- [30] K. Paterson and J. Schuldt. Efficient Identity-based Signatures Secure in the Standard Model. In ACISP 2006, pages 207-222, 2006.
- [31] C. P. Schnorr and M. Jakobsson. Security of Signed ElGamal Encryption. In Asiacrypt 2000, LNCS 1976, 458–469, Springer-Verlag, 2000.

- [32] K. Sakurai, and Y. Yamane. Blind decoding, blind undeniable signatures, and their applications to privacy protection. Proc. 1st Information Hiding Workshop, Cambridge, U.K. Springer LNCS 1174, 257–264, 1996.
- [33] K. Sakurai, Y. Yamani, S. Miyazaki, T. Inoue. A Key Escrow System with Protecting User's Privacy by Blind Decoding. ISW 1997, LNCS 1396, 147–157, 1997.
- [34] B. Waters. Efficient identity-based encryption without random oracles. Advances in CryptologyEurocrypt'05, LNCS 1666, pp.114127, 2005.
- [35] F. Zhang and K. Kim. ID-Based Blind Signature and Ring Signature from Pairings. Advances in Cryptology - ASIACRYPT'02, LNCS 2501, SpringerVerlag, pp. 533-547, 2002.
- [36] F. Zhang and K. Kim. Ecient ID-based blind signature and proxy signature from bilinear pairings. In ACISP'03, pages 312-323, 2003.

# A Formal definitions of leak-freeness and selectivefailure blindness

**Definition 10** (Leak-Free Extract [20]). A protocol BlindExtract( $\mathcal{P}, \mathcal{U}$ ) associated with an HIBE scheme  $\Pi = ($ Setup, Extract, Encrypt, Decrypt) is leak-free if for all efficient adversaries  $\mathcal{A}$ , there exists an efficient simulator  $\mathcal{S}$  such that for every value  $\kappa$  and l, no efficient distinguisher  $\mathcal{D}$  can distinguish whether  $\mathcal{A}$  is playing with Game Real or Game Ideal with non-negligible advantage:

**Game Real:** Run (params, msk)  $\leftarrow$  Setup( $\kappa$ , l). As many time as  $\mathcal{D}$  wants,  $\mathcal{A}$  chooses an identity  $ID_{|j} = (I_1, \ldots, I_j)$  and executes the BlindExtract protocol with  $\mathcal{P}_{ID|j-1}$ :

 $BlindExtract(\mathcal{P}_{ID|j-1}(sk_{ID|j-1}), \mathcal{A}(ID_{|j})).$ 

**Game Ideal:** Run (params, msk)  $\leftarrow$  Setup( $\kappa$ , l). As many time as  $\mathcal{D}$  wants,  $\mathcal{S}$  chooses an identity  $ID_{|j} = (I_1, \ldots, I_j)$  and queries the trusted party  $\mathcal{P}_{ID|j-1}$  to obtain its output in

 $Extract(\mathcal{P}_{ID|j-1}(sk_{ID|j-1}), \mathcal{S}(ID_{|j})).$ 

Here,  $\mathcal{D}$  and  $\mathcal{A}$  (or  $\mathcal{S}$ ) may communicate at any time.

This definition implies that the identity  $ID_{|j}$  for the key being extracted must be known by  $\mathcal{S}$ . The simulator must be able to interact with  $\mathcal{A}$  to learn which identities to submit to corresponding parent identities. This observation will be used in proving leak-freeness later.

Another nice feature of the above definition is that any key extraction protocol with leak-freeness (regardless of whether blindness holds or not) composes into the existing security definitions for HIBE, which is formally stated in the following lemma, which is an obvious generalization of its counterpart for IBE in [20].

**Lemma 11.** If  $\Pi = (Setup, Extract, Encrypt, Decrypt)$  is an IND-sID-CPA-secure (resp., IND-ID-CPA) HIBE scheme and BlindExtract protocol associated with  $\Pi$  is leak-free, then  $\Pi' = (Setup, BlindExtract, Encrypt, Decrypt)$  is also an IND-sID-CPA-secure (resp., IND-ID-CPA) HIBE scheme.

We next define *selective-failure blindness*. This is the strongest notion of blindness, recently proposed by Camenisch et al. [15], ensuring that even a malicious parent identity is unable to induce BlindExtract protocol failures that are dependent on the identity being extracted.

**Definition 12** (Blindness of Blind HIBE [15]). A protocol  $P(\mathcal{A}_{ID|j-1}(\cdot), \mathcal{U}(\cdot, \cdot))$  is said to be selective-failure blind if every p.p.t adversary  $\mathcal{A}_{ID|j-1}$  has a negligible advantage in the following game: First, (params, msk)  $\leftarrow$  Setup $(\kappa, l)$ . The master key msk is used to generate the private key  $sk_{ID|j-1}$  for  $\mathcal{A}_{ID|j-1}$  while params is given to  $\mathcal{U}$ .  $\mathcal{A}_{ID|j-1}$ first outputs two values  $I_0, I_1 \in \mathcal{I}$ . A random  $b \in \{0, 1\}$  is chosen and let  $ID_{b|j} =$  $(ID_{|j-1}, I_b)$ .  $\mathcal{A}_{ID|j-1}$  is given a black-box access to two oracles  $\mathcal{U}(params, ID_{b|j})$  and  $\mathcal{U}(params, ID_{1-b|j})$ . The  $\mathcal{U}$  algorithms produce local outputs  $sk_b$  and  $sk_{1-b}$  respectively. If  $sk_b \neq \bot$  and  $sk_{1-b} \neq \bot$ , then  $\mathcal{A}_{ID|j-1}$  is given  $(sk_0, sk_1)$ . If  $sk_b = \bot$  and  $sk_{1-b} \neq \bot$ , then  $\mathcal{A}_{ID|j-1}$  is given  $(\bot, \epsilon)$ . If  $sk_b \neq \bot$  and  $sk_{1-b} = \bot$ , then  $\mathcal{A}_{ID|j-1}$  is given  $(\epsilon, \bot)$ . If  $sk_b = \bot$  and  $sk_{1-b} = \bot$ , then  $\mathcal{A}_{ID|j-1}$  is given  $(\bot, \bot)$ . Finally,  $\mathcal{A}_{ID|j-1}$  outputs its guess b'. We define the advantage of  $\mathcal{A}_{ID|j-1}$  in this game as  $|\Pr[b' = b] - 1/2|$ .

## **B** Security Proof for blind HIBE schemes

#### B.1 Proof of Theorem 3

We will show that the BlindExtract protocol for Boneh-Boyen HIBE is both leak-free and selective-failure blind.

We first begin with leak-freeness. Let  $\mathcal{A}$  be a p.p.t adversary interacting with  $\mathcal{P}_{ID|j-1}$ in a BlindExtract protocol. We will show that there exists a simulator  $\mathcal{S}$  interacting with  $\mathcal{P}_{ID|j-1}$  in an Extract protocol such that Game Real and Game Ideal as in Definition 10 are indistinguishable. The simulator  $\mathcal{S}$  is built as follows:

1. S hands its input *params* to A, which is run internally by S.

2. S simulates the BlindExtract protocol for A as follows: in the first message of the protocol, A must send to S a value h' and prove knowledge of values  $(y, I_j)$  such that  $h' = g^y g_1^{I_j}$ . If the proof fails to verify, S aborts. Otherwise, notice that S runs A so that S can efficiently extract y and  $I_j$  by rewinding A. (This is the well-known rewind technique.)

S queries (ID<sub>|j-1</sub>, I<sub>j</sub>) to its own Extract oracle to obtain (d<sub>0</sub>F<sub>j</sub>(I<sub>j</sub>)<sup>r<sub>j</sub></sup>, d<sub>1</sub>, ..., d<sub>j-1</sub>, g<sup>r<sub>j</sub></sup>) where r<sub>j</sub> is random chosen and the private key of P<sub>ID|j-1</sub> is (d<sub>0</sub>,..., d<sub>j-1</sub>).
 4. Finally, S returns to A the following values:

$$(d_0F_j(I_j)^{r_j} \cdot (g^{r_j})^y, d_1, \dots, d_{j-1}, g^{r_j}).$$

Note that S knows y, so the above values are either known or computable by S. Also note  $d_0F_j(I_j)^{r_j} \cdot (g^{r_j})^y = d_0(F_j(I_j)g^y)^{r_j} = d_0(h'h_j)^{r_j}$ , so that what S returns to A has exactly the same distribution as what A can obtain from the BlindExtract protocol in Game Real. Thus Game Real and Game Ideal are indistinguishable and hence the BlindExtract protocol for Boneh-Boyen HIBE is leak-free.

We proceed to prove selective-failure blindness. Recall that  $\mathcal{A}_{ID|j-1}$  first outputs two values  $I_0, I_1 \in \mathcal{I}$ . Then a random bit b is chosen and let  $ID_{b|j} = (ID_{|j-1}, I_b)$ . Next,  $\mathcal{A}_{ID|j-1}$  is given a black-box access to two oracles  $\mathcal{U}(params, ID_{b|j})$  and  $\mathcal{U}(params, ID_{1-b|j})$ . The  $\mathcal{U}$  algorithms produce local outputs  $sk_b$  and  $sk_{1-b}$  respectively. If  $sk_b \neq \bot$  and  $sk_{1-b} \neq \bot$ , then  $\mathcal{A}_{ID|j-1}$  is given  $(sk_0, sk_1)$ . If  $sk_b = \bot$  and  $sk_{1-b} \neq \bot$ , then  $\mathcal{A}_{ID|j-1}$  is given  $(\bot, \epsilon)$ . If  $sk_b \neq \bot$  and  $sk_{1-b} = \bot$ , then  $\mathcal{A}_{ID|j-1}$  is given  $(\epsilon, \bot)$ . If  $sk_b = \bot$  and  $sk_{1-b} = \bot$ , then  $\mathcal{A}_{ID|j-1}$  is given  $(\epsilon, \bot)$ .

We observe that in the BlindExtract protocol, the user (and hence the oracles  $\mathcal{U}(params, ID_{0|j})$ ,  $\mathcal{U}(params, ID_{1|j})$ ) speaks first, sending to  $\mathcal{A}_{ID|j-1}$  a uniformlydistributed value h' in G, and then performs a zero-knowledge proof of representation  $PoK\{(y, I_b): h' = g^y g_1^{I_b}\}$ . Now it is  $\mathcal{A}_{ID|j-1}$ 's turn to speak, and at this point, his views so far are computationally indistinguishable, since  $\mathcal{A}_{ID|j-1}$  only received uniformlydistributed values in G and proofs of representation in zero-knowledge.  $\mathcal{A}_{ID|j-1}$  must now returns  $(d'_0, d_1, \ldots, d_{j-1}, d'_j) \in G^{j+1}$  to the first oracle.  $\mathcal{A}_{ID|j-1}$  can choose these values in an arbitrary manner. We will show that accessing to oracles  $\mathcal{U}(params, ID_{0|j})$ and  $\mathcal{U}(params, ID_{1|j})$ ) is useless to  $\mathcal{A}_{ID|j-1}$ . In other words, we will show that  $\mathcal{A}_{ID|j-1}$ can predict the outputs of these oracles without accessing to them. To achieve that,  $\mathcal{A}_{ID|j-1}$  does the following:

1.  $\mathcal{A}_{ID|j-1}$  checks if

$$e(d'_0,g) = e(g_2,g_1)e(h'h_j,d'_j)\Pi_{k=1}^{j-1}e(F_k(id_k),d_k),$$

where  $ID_{|j-1} = (id_1, \ldots, id_{j-1})$ . If the check fails,  $\mathcal{A}_{ID|j-1}$  temporarily records  $sk_0 = \bot$ . Otherwise, it (owning  $sk_{ID|j-1}$ ) records  $sk_0 = \mathsf{Extract}(sk_{ID|j-1}, ID_{0|j})$ .

2.  $\mathcal{A}_{ID|j-1}$  again chooses values  $(d'_0, d_1, \ldots, d_{j-1}, d'_j) \in G^{j+1}$  for the second oracle, performs the same check and temporarily recording  $sk_1$  for  $ID_{1|j}$  as before.

3. Finally, if both checks failed or both checks succeeded,  $\mathcal{A}_{ID|j-1}$  takes  $(sk_0, sk_1)$ . If  $sk_0 = \bot$  and  $sk_1 \neq \bot$ , take  $(\bot, \epsilon)$ . If  $sk_0 \neq \bot$  and  $sk_1 = \bot$ , take  $(\epsilon, \bot)$ .

By inspection, we can see that  $\mathcal{A}_{ID|j-1}$  is performing what the two oracles do, namely  $\mathcal{A}_{ID|j-1}$  makes the same checks and finally taking values which have the same distributions as the outputs of the oracles. As a consequence,  $\mathcal{A}_{ID|j-1}$  does not need the oracles to improve its advantage in guessing the bit b. Thus, all of  $\mathcal{A}_{ID|j-1}$ 's advantage must come from distinguishing the earlier messages from the oracles, which are only randomly-distributed values  $h' \in G$  and zero-knowledge proofs of representation of h'. From the security of the zero-knowledge proofs, we know that  $\mathcal{A}_{ID|j-1}$ 's advantage is negligible, so that selective-failure blindness is satisfied.

#### **B.2** Proof of Theorem 4

This proof follows the outline of the proof of Theorem 3 almost identically. To make leak-freeness satisfied, the simulator S is built exactly as before: (1) running the adversary A internally; (2) extracting the values  $(y, I_j[1], \dots, I_j[n])$  from the corresponding proof of knowledge; (3) querying  $(ID_{|j-1}, I_j = I_j[1] \cdots I_j[n])$  to the Extract oracle to get  $(d_0F_j(I_j)^{r_j}, d_1, \dots, d_{j-1}, g^{r_j})$ ; (4) and finally returning  $(d_0F_j(I_j)^{r_j} \cdot (g^{r_j})^y, d_1, \dots, d_{j-1}, g^{r_j})$  to A.

To satisfy selective-failure blindness, we observe that the prediction of  $\mathcal{U}$ 's final output is done exactly as before. Thus  $\mathcal{A}_{ID|j-1}$  does not need the  $\mathcal{U}$  oracles, and hence all of its advantage come from seeing the randomly-distributed values  $h' \in G$  and zero-knowledge proofs of representation of h' from the oracles. We conclude that this advantage must be negligible.

## C Proof of Theorem 9

*Proof.* Let Game 0 be the attack game as in Definition 7. Note that in Game 0, every time the adversary  $\mathcal{A}$  makes a blind decryption query to  $\mathcal{P}_{ID}$  with a ciphertext C, where  $ID \in \mathsf{Leaves}$  and  $C = ((ID, I), \hat{C})$ , the protocol  $\mathsf{BlindExtract}(\mathcal{P}_{ID}(sk_{ID}), \mathcal{A}(ID, I))$  is executed.

Game 1 is the same as Game 0, except that the part of  $\mathcal{A}$  making blind decryption queries is replaced by the simulator  $\mathcal{S}$ , and the protocol BlindExtract is replaced by Extract. That means, BlindExtract( $\mathcal{P}_{ID}(sk_{ID}), \mathcal{A}(ID, I)$ ) is replaced by Extract( $\mathcal{P}_{ID}(sk_{ID}), \mathcal{S}(ID, I)$ ).

By the leak-free property of the underlying blind HIBE, Game 0 and Game 1 are indistinguishable. Indeed, every p.p.t algorithm distinguishing the two games can be turned into the distinguisher  $\mathcal{A}_1$  against leak-freeness. Thus,

$$\Pr[b'_{i} = b_{\pi(i)} \forall 1 \leq i \leq v + 1 \text{ in Game } 0] - \Pr[b'_{i} = b_{\pi(i)} \forall 1 \leq i \leq v + 1 \text{ in Game } 1]$$
  
 
$$\leq \mathbf{Adv}_{\mathsf{B}-\mathsf{HIBE}}^{leak-free}(\mathcal{A}_{1}),$$

which leads to

$$\Pr[b'_{i} = b_{\pi(i)} \forall 1 \le i \le v + 1 \text{ in Game } 0] - \frac{1}{2} \le \mathbf{Adv}_{\mathsf{B}\text{-HIBE}}^{leak-free}(\mathcal{A}_{1}) + \left(\Pr[b'_{i} = b_{\pi(i)} \forall 1 \le i \le v + 1 \text{ in Game } 1] - \frac{1}{2}\right)$$

and hence,

$$\begin{aligned} \mathbf{Adv}_{\mathsf{HBDP}}^{ind-om}(\mathcal{A}) &\leq & \mathbf{Adv}_{\mathsf{B}-\mathsf{HIBE}}^{leak-free}(\mathcal{A}_1) \\ &+ & \Big(\Pr[b'_i = b_{\pi(i)} \forall 1 \leq i \leq v+1 \text{ in Game } 1] - \frac{1}{2}\Big). \end{aligned}$$

It now suffices to build the ind-sid-cpa adversary  $\mathcal{A}_2$  who utilizes the adversary  $\mathcal{A}$  and its corresponding simulator  $\mathcal{S}$  in Game 1 such that

$$\Pr[b'_i = b_{\pi(i)} \forall 1 \le i \le v + 1 \text{ in Game } 1] - \frac{1}{2} \le ku \mathbf{Adv}_{\mathsf{B}-\mathsf{HIBE}}^{ind-sid-cpa}(\mathcal{A}_2).$$

Recall that, as in IND-OM security definition, Leaves =  $\{ID_1, \ldots, ID_k\}(k \ge 1)$ . We now build  $\mathcal{A}_2$  as follows.  $\mathcal{A}_2$  chooses  $j^* \stackrel{\$}{\leftarrow} \{1, \ldots, k\}$  and  $i^* \stackrel{\$}{\leftarrow} \{1, \ldots, u_{j^*}\}$ . It furthermore prepares  $I_{(i,1)}, \ldots, I_{(i,u_i)} \stackrel{\$}{\leftarrow} \mathcal{I}$  for all  $1 \le i \le k$ . (These values represents items owned by retailer  $ID_i$ .)  $\mathcal{A}_2$  outputs the target identity  $ID^* = (ID_{j^*}, I_{(j^*, i^*)})$ , receiving *params* from its challenger. It makes key extraction queries  $ID_j$  for all  $1 \le j \ne j^* \le k$ to get  $sk_{ID_j}$  from its own oracle. It feeds *params* to  $(\mathcal{A}, \mathcal{S})$  and begins to simulate the environment for  $(\mathcal{A}, \mathcal{S})$  as follows:

- Encryption query  $(ID_j \in \text{Leaves}, m_0^{(i)}, m_1^{(i)})$   $(1 \leq j \leq k \text{ and } 1 \leq i \leq u_j)$  by  $\mathcal{A}$ : If  $j = j^*$  and  $i = i^*$  then  $\mathcal{A}_2$  forwards  $(m_0^{(i)}, m_1^{(i)})$  to its own encryption oracle to obtain  $\hat{C}^* = \text{Encrypt}(params, ID^*, m_b^{(i)})$  for some challenge bit b, denoted as  $b_{(j^*,i^*)}$ .  $\mathcal{A}_2$  gives  $(ID^*, \hat{C}^*) = ((ID_{j^*}, I_{(j^*,i^*)}), \hat{C}^*)$  to  $\mathcal{A}$ . In the other cases, namely  $j \neq j^*$  or  $i \neq i^*$ ,  $\mathcal{A}_2$  chooses a bit  $b_{(j,i)}$  at random and compute  $\hat{C}_{(j,i)} \leftarrow$ Encrypt $(params, (ID_j, I_{(j,i)}), m_{b_{(j,i)}}^{(i)})$ , and gives  $C_{(j,i)} \leftarrow ((ID_j, I_{(j,i)}), \hat{C}_{(j,i)})$  to  $\mathcal{A}$ . - Key extract query  $(ID_j, SI)$   $(1 \le j \le k \text{ and } SI \in \mathcal{I})$  by  $\mathcal{S}$ : If  $j \ne j^*$ , then  $\mathcal{A}_2$  uses its own key  $sk_{ID_j}$  to answer the query. If  $j = j^*$  and  $SI \ne I_{(j^*,i^*)}$ , then  $\mathcal{A}_2$  forwards the query to its own key extraction oracle and gives  $\mathcal{S}$  the private key it gets. If  $j = j^*$ and  $SI = I_{(j^*,i^*)}$ ,  $\mathcal{A}_2$  halts and outputs a random bit. Denote this event Halt<sub>1</sub>.

If  $\operatorname{Halt}_1$  does not occur,  $(\mathcal{A}, \mathcal{S})$  outputs v + 1 bits  $b'_1, \ldots, b'_{v+1}$  and an injective map  $\pi : \{1, \ldots, v+1\} \to \{1, \ldots, k\} \times \{1, \ldots, u\}$ . Consider v + 1 corresponding ciphertexts of the form  $((ID_j, I_{(j,i)}), \hat{C}_{(j,i)})$  where  $(j, i) \in \pi(\{1, \ldots, v+1\})$ . Since there are v + 1 corresponding identities, and only v key extraction queries are made, there exists one identity which was not a key extraction query, called  $((ID_r, I_{(r,s)})$  where  $(r, s) = \pi(t)$  for some  $1 \leq t \leq v + 1$ . Recall that  $b'_t$  is the guess of  $b_{\pi(t)}$ . If  $\pi(t) = (j^*, i^*)$  then  $\mathcal{A}_2$  outputs  $b'_t$ ; otherwise it halts and outputs a random bit. Denote this event  $\operatorname{Halt}_2$ 

Let  $b_{\mathcal{A}_2}$  be the bit finally output by  $\mathcal{A}_2$  and R the event that  $\mathcal{A}_2$  outputs a random bit. We have

$$\begin{aligned} \mathbf{Adv}_{\mathsf{B}\text{-HIBE}}^{ind-sid-cpa}(\mathcal{A}_2) \\ &= \left| \Pr[b_{\mathcal{A}_2} = b_{(j^*,i^*)}] - \frac{1}{2} \right| \\ &= \left| \Pr[b_{\mathcal{A}_2} = b_{(j^*,i^*)} | \mathbf{R}] \Pr[\mathbf{R}] + \Pr[b_{\mathcal{A}_2} = b_{(j^*,i^*)}] \overline{\mathbf{R}}] \Pr[\overline{\mathbf{R}}] - \frac{1}{2} \\ &= \left| \frac{1}{2} (1 - \Pr[\overline{\mathbf{R}}]) + \Pr[b'_t = b_{\pi(t)}] \Pr[\overline{\mathbf{R}}] - \frac{1}{2} \right| \\ &= \Pr[\overline{\mathbf{R}}] \left( \left| \Pr[b'_t = b_{\pi(t)}] - \frac{1}{2} \right| \right) \\ &\geq \Pr[\overline{\mathbf{R}}] \left( \Pr[b'_t = b_{\pi(i)}] \forall 1 \le i \le v + 1 \text{ in Game } 1] - \frac{1}{2} \right). \end{aligned}$$

It is now sufficient to prove that  $\Pr[\overline{R}] \ge 1/ku$ . Since  $R = \operatorname{Halt}_1 \lor \operatorname{Halt}_2$ , we have  $\overline{R} = \operatorname{Halt}_1 \land \operatorname{Halt}_2$ , and hence  $\Pr[\overline{R}] = \Pr[\operatorname{Halt}_2] \Pr[\operatorname{Halt}_1|\operatorname{Halt}_2]$ . Note that if  $\operatorname{Halt}_2$  happened, then we know that  $\operatorname{Halt}_1$  had previously occurred, so that  $\Pr[\operatorname{Halt}_1|\operatorname{Halt}_2] = 1$ . We also have  $\Pr[\operatorname{Halt}_2] = \Pr[\pi(t) = (j^*, i^*)] \ge 1/ku$ , so  $\Pr[\overline{R}] \ge 1/ku$ , which concludes the proof.