Identity-Based Encryption Secure against Selective Opening Attack

Mihir Bellare[1], Brent Waters[2], and Scott Yilek[3]

[1] Department of Computer Science and Engineering,
University of California San Diego, La Jolla CA, USA
http://cseweb.ucsd.edu/~mihir/
[2] Department of Computer Science,
University of Texas at Austin, Austin TX, USA
http://www.cs.utexas.edu/~bwaters/
[3] Department of Computer and Information Sciences,
University of St. Thomas, St. Paul, MN, USA
http://personal.stthomas.edu/yile5901/

Abstract. We present the first IBE schemes that are proven secure against selective opening attack (SOA). This means that if an adversary, given a vector of ciphertexts, adaptively corrupts some fraction of the senders, exposing not only their messages *but also their coins*, the privacy of the unopened messages is guaranteed. Achieving security against such attacks is well-known to be challenging and was only recently done in the PKE case. We show that IBE schemes having a property we call 1-sided public openability (1SPO) yield SOA secure IBE schemes and then provide two 1SPO IBE schemes, the first based on the Boyen-Waters anonymous IBE and the second on Waters' dual-system approach.

1 Introduction

Security against selective-opening attack (SOA) is arguably the most paradoxical and vexing open question in the theory of encryption. Recently (and 10 years after the problem was identified), we have seen solutions [2]. These and followups [24,22], however, have been for the case of Public-Key Encryption (PKE). Another domain where the problem arises, and is important for applications, is Identity-Based Encryption (IBE). The techniques used for PKE do not yield solutions here.This paper initiates a treatment of IBE secure under SOA, providing definitions of security and the first schemes achieving them. Our schemes do not use random oracles.

Background. A selective-opening attack on a PKE scheme imagines n senders and receivers. Sender i encrypts a message $\mathbf{m}[i]$ under fresh, random coins $\mathbf{r}[i]$ and the public key $\mathbf{pk}[i]$ of the i-th receiver to get a ciphertext $\mathbf{c}[i]$. An adversary given the vector $\mathbf{c}$ corrupts some subset of the senders and learns not only their messages but also their coins. SOA-security requires that the remaining, unopened messages retain their privacy. SOA-security is required when implementing the

Y. Ishai (Ed.): TCC 2011, LNCS 6597, pp. 235–252, 2011.

assumed secure channels in an adaptively-secure multi-party computation protocol. More pragmatically, it would be required to distribute shares in a distributed file-system that is using secret-sharing for privacy.

IND-CPA and IND-CCA, widely-accepted as the "right" notions of encryption privacy, are not known to imply security under SOA. The difficulty of establishing SOA-security stems from the fact that the adversary gets the coins and also that the messages $\mathbf{m}[1], \ldots, \mathbf{m}[n]$ may be related. Constructions of SOA secure schemes also remained elusive, the area colored by negative results for commitment schemes [21,2,28]. Finally, Bellare, Hofheinz, and Yilek (BHY) [2] showed a large class of encryption schemes, which they call *lossy* [2,25,30], are SOA secure. Schemes they show to be lossy include variants of El Gamal [27], the IND-CPA scheme built from lossy trapdoor functions by Peikert and Waters [31], and even the original Goldwasser-Micali encryption scheme [23]. Hemenway, Libert, Ostrovsky and Vergnaud [24] showed that re-randomizable encryption and statistically hiding, two-round oblivious transfer imply lossy encryption, yielding still more examples of SOA secure PKE schemes via the lossy-implies-SOA-secure connection of BHY. Fehr, Hofheinz, Kiltz, and Wee (FHKW) [22] use a deniable encryption [13] approach to achieve CC-SOA (Chosen-Ciphertext SOA) secure PKE.

SOA FOR IBE. We can adapt the SOA framework to IBE in a natural way. A vector **id** of adversarially-chosen target receiver identities replaces the vector **pk** of public receiver keys. Sender i encrypts message $\mathbf{m}[i]$ under coins $\mathbf{r}[i]$ for identity $\mathbf{id}[i]$ to get a ciphertext $\mathbf{c}[i]$. As before the adversary, given $\mathbf{c}$, corrupts a subset of the senders and learns their messages and coins, and SOA-security requires that the unopened messages are secure. At any time, the adversary can query **Extract** with any identity not in the vector **id** and obtain its decryption key.

There are two elements here, new compared to PKE, that will be central to the technical challenges in achieving the goal. The first is the **Extract** oracle, a feature of IBE security formalizations since the pioneering work of Boneh and Franklin [9], that allows the adversary to obtain the decryption key of any (non-target) receiver of its choice. The second is that the target identities are chosen by the adversary. (We will achieve full, rather than selective-id security [15].)

IBE can conveniently replace PKE in applications such as those mentioned above, making its SOA-security important. Beyond this, we feel that determining whether SOA-secure IBE is possible is a question of both foundational and technical interest.

CONTRIBUTIONS IN BRIEF. We provide a simulation-based, semantic security formalization of SOA-secure IBE. (This means our results do not need to assume conditional re-samplability of message spaces, in contrast to some of the results of [2] for IND-style notions.) We provide a general paradigm to achieve SOA-secure IBE based on IBE schemes that are IND-CPA and have a property we call 1-Sided Public Openability (1SPO). We discuss why obtaining 1SPO IND-CPA IBE schemes without random oracles is not immediate and then illustrate two ways to do it.

Scheme	Pars	Ctxt	Keys	Enc	Dec	F/S	Assumption
LoR	$n+6$	5	5	5 exp	5 pr	F	DLIN
BBoR	4	2	2	2 exp	2 pr	F	GSD

Fig. 1. Our 1SPO IND-CPA IBE schemes. These encrypt 1-bit messages. Bit-by-bit encryption yields SOA-secure IBE schemes encrypting full messages. "Pars" is the size of the public parameters, "Ctxt" of the ciphertext and "Keys" of the decryption keys, all in group elements, with n the length of identities. (In practice $n = 160$ by hashing identities.) "Enc" and "Dec" are the encryption and decryption costs with "exp" standing for an exponentiation or multi-exponentiation and "pr" for a pairing. "F/S" indicates whether we get Full or Selective-id security. "GSD" stands for the General Subgroup Decision assumption.

The first, adapting the anonymous IBE scheme of Boyen and Waters [12], yields a SOA-secure IBE scheme based on the DLIN (Decision Linear) assumption of [7]. The second, using the dual-system approach of [32], yields a SOA-secure IBE scheme in the Boneh-Boyen style [6] based on a subgroup decision assumption in composite order groups. Attributes of the schemes are summarized in Figure 1. We now expand on these contributions.

1SPO IBE IMPLIES SOA-SECURE IBE. There are fundamental obstacles to extending BHY's lossy-implies-SOA-secure approach, that worked for SOA-secure PKE, to the IBE setting. (Briefly, we cannot make the encryption undetectably lossy on all challenge identities because the adversary has an **Extract** oracle and we wish to achieve full, not selective-id [15] security.) Instead we return to ideas from non-committing [14] and deniable [13] encryption. We define IBE schemes that have a property we call *one-sided public openability* (1SPO) and is an IBE-analogue of a weak form of deniable PKE [13]. In short, an IBE scheme for 1-bit messages is 1SPO if it is possible, given the public parameters par, an identity id, and the encryption c of message 1 under par and id, to efficiently open the encryption, meaning find correctly-distributed randomness r such that encrypting a 1 using $\mathsf{par}, \mathsf{id}, r$ results in the ciphertext c. We emphasize that this opening must be done without the aid of any secret information. Bit-by-bit encryption then results in a scheme that can encrypt long messages. We show in Theorem 1 that if the starting 1-bit 1SPO scheme is also IND-CPA secure then the constructed IBE scheme is SOA-secure. This reduces the task of obtaining SOA-secure IBE schemes to obtaining IND-CPA secure 1SPO schemes.

FHKY [22] develop a similar approach in the PKE setting. Their work and ours are concurrent and independent. (Both were submitted to Eurocrypt 2010 but only theirs was accepted.)

FINDING 1SPO IBE SCHEMES. Known Random Oracle (RO) model IBE schemes [9,19] can be adapted to be 1SPO secure, yielding SOA-secure IBE in the RO model. Achieving it without ROs, however, turns out not to be straightforward. The natural approach, extending that used for PKE [13,22], is to build IBE

schemes that are what we call 1SIS (1-sided invertibly samplable). Here, encryptions of 0 to a certain identity would have a certain structure. This structure should be detectable with the secret key associated to the identity, but *not* without it, and thus not by an attacker. On the other hand, encryptions of 1 would be random, but in a special way, namely there is a public procedure that given an encryption c of a 1 can compute randomness (coins) under which the encryption algorithm applied to 1 would produce c. Any such scheme is 1SPO. The challenge that emerges is to find 1SIS IND-CPA IBE schemes. Existing IBE schemes do not have the property, and nor do direct adaptations work. The Boneh-Boyen approach [6] is probably the most widely used in IBE design. (Waters' IBE scheme [33] is one instance.) However, ciphertexts in BB-schemes contain group elements that obey relations an attacker can test and thus cannot be undetectably replaced with random group elements. We will obtain our first solution by a different approach. Then, however, we will go back to show how the dual-system approach can be used to make a BB-style scheme work if we use composite order groups.

The linear scheme. In our "Linear or Random" (LoR) scheme, an encryption of 0 to a given identity, id, is done using (a modification of) the Boyen-Waters (BW) encryption algorithm [12]. This output of the encryption will be five group elements that share a certain structure that is only detectable to a user with the private key for id. To encrypt a 1 we simply choose five random group elements. This, however, must be done using what we call a publicly invertible process (see below). The main feature of this encryption scheme is that an encryption of 0 can always be claimed as just five random group elements, and thus as an encryption of 1. This reveals why we choose to build of the BW anonymous IBE scheme as opposed to other simpler IBE systems without random oracles. The main feature of the BW ciphertexts is that they have no detectable structure from an attacker that does not have a private key for id. In contrast, in BB-style IBE systems [6,33] the attacker can test for structure between two group elements in well formed ciphertexts. Therefore we cannot create a secure encryption system simply by replacing these with random group elements.

We prove LoR is 1SPO directly. We must also, however, prove it is IND-CPA. We adapt techniques from [12,3] to do this under the DLIN assumption. The proof technique of [3] allows us to avoid Waters' artificial abort step [33] thereby resulting in a more efficient reduction.

The dual-system scheme. Waters introduced a new approach to IBE called the dual system approach in which both the challenge ciphertext and keys are replaced in the proof by "semi-functional" versions [32]. We adapt this approach to get a 1SIS (and thus 1SPO) IND-CPA IBE scheme and thus a SOA-secure IBE scheme. An interesting feature of the scheme is that ciphertexts have a BB-form, showing that the dual-system approach can surmount the above-mentioned difficulties in making BB-style systems 1SIS. We accordingly call the scheme BBoR (BB or Random). In addition this is interesting because it illustrates a quite different technique and yields a scheme based on a different assumption

(subgroup decision in a composite group, not known to imply or be implied by DLIN in a prime-order group). As Figure 1 shows, the main pragmatic difference compared to LoR is short public parameters. (Those of LoR are long due to the Waters' hash function [33] which is required to get full security.) Others costs have dropped as well (from 5 to 2) but the group is larger so a closer analysis would be needed to determine whether this translates to actual efficiency gains.

Our starting points are the dual-system based Lewko-Waters (LW) IBE scheme [26] and its anonymous extension by De Caro, Iovino and Persiano (DIP) [17]. We modify these to get a 1SIS scheme where an encryption of a 0 is BB-ciphertext but in a subgroup while an encryption of a 1 is a pair of random points in the full group. While extending these schemes we manage simultaneously to make the assumptions simpler, more natural and fewer. Specifically, all these schemes rely on a pairing $\mathrm{e}\colon\ \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ where $\mathbb{G}, \mathbb{G}_T$ have composite order N. LW make three different assumptions (numbered 1,2,3), the first two being about subgroup decision in $\mathbb{G}$ and the third in $\mathbb{G}_T$. DIP also make three assumptions, with the third being quite ad hoc and tailored to the scheme. We eliminate the third assumption in both cases and unify the rest, formulating what we call the general subgroup decision assumption, which is only in $\mathbb{G}$, and basing the proof solely on this single assumption.

Publicly invertible sampling. We have said that encryptions of a 1 in our 1SIS schemes are random group elements. This, however, is not enough. They have to be invertibly sampled. As we explained above, this means there is a public procedure that given an encryption c of a 1 can compute randomness (coins) under which the encryption algorithm applied to 1 would produce c. To illustrate the subtleties of the notion, consider a scheme in which the encryption c of a 1 is computed by picking an exponent x at random and returning g^x where g is a generator of a group $\mathbb{G}$. Although the ciphertext is random, this is not invertibly samplable since we cannot recover x from c. Instead, a ciphertext must be sampled "directly" as $c \leftarrow\!\!\$\ \mathbb{G}$. The difficulty is that whether or not this is possible depends on the group. In the PKE case, it is possible to stay within simple groups such as $\mathbb{Z}_p^*$ for prime p, where such sampling is easy. (Pick a a random integer in the range $1, \ldots, p-1$.) In our case, however, $\mathbb{G}$ is a complex group, namely a subgroup of the points on an elliptic curve. We show how to sample invertibly nonetheless, relying on the structure of the elliptic curve groups in question. Specifically, we modify some methods used to implement the hash function of the BLS signature scheme [11].

Extensions and open problems. After seeing a preliminary version of our work, Peikert [29] has said that the lattice-based IBE schemes of [18,1] adapt to yield 1SPO schemes, whence, by our results, SOA-secure IBE. One interesting direction for further work is to define SOA-secure HIBE and then extend our schemes (the second in particular) as well as the lattice ones to achieve it. Another direction is to define deniable IBE and then fuse our approaches with those of [13] to achieve it. We remark that even given SOA-secure HIBE, we do not directly achieve CC-SOA (Chosen-ciphertext SOA) secure IBE because

the BCHK transform [8] does not work in the SOA setting. (The problem is opening the randomness used in creating the one-time-signature key.) Achieving CC-SOA secure IBE is another interesting open question.

Related work. Canetti, Feige, Goldreich and Naor [14] introduced non committing encryption (NCE) to achieve adaptively secure multi-party computation in the computational (as opposed to secure channels) setting without erasures. In their treatment, NCE is an interactive protocol, and their definition of security is in the MPC framework. The model allows corruption of both senders and receivers. They show how to achieve NCE but, viewed as a public-key system, they would have keys larger than the total number of message bits that may be securely encrypted. Damgård and Nielsen [20] introduced more efficient schemes but this restriction remained, and Nielsen [28] showed it was necessary. With partial erasures, more efficient solutions were provided by Canetti, Halevi and Katz [16].

Dwork, Naor, Reingold and Stockmeyer [21] extracted out a stand-alone notion of commitment secure against selective opening defined directly by a game rather than via the MPC framework. Corruptions allow the adversary to obtain the committer's coins along with its message. This was adapted to public-key encryption in [2], who focused on sender (as opposed to receiver) corruptions and were then able to obtain solutions based on lossy encryption.

Canetti, Dwork, Naor and Ostrovsky [13] introduced deniable encryption, where a sender may open a ciphertext to an arbitrary message by providing coins produced by a faking algorithm. The authors explain that this is stronger than NCE because in the latter only a simulator can open in this way. A weak form of their requirement is that encryptions of 1 can be opened as encryptions of 0 even if not vice versa. 1SPO IBE is an IBE analogue of this notion.

Sender versus receiver corruptions. We clarify that our model and results are for adaptive sender corruptions, not adaptive receiver corruptions. (The latter would correspond to being allowed to query to **Extract** identities in the challenge vector **id**.) Security against adaptive receiver corruptions seems out of reach of current techniques for PKE let alone for IBE. (Without either erasures or keys as long as the total number of messages bits ever encrypted.) We do allow receiver corruptions via the **Extract** oracle but these are non-adaptive. We view this as retaining (meaning neither weakening nor strengthening) the guarantees against receiver corruption already provided by the basic definition of IND-CPA-secure IBE [9]. Our notion, security against adaptive sender and non-adaptive receiver corruptions, is still very strong.

2 Preliminaries

Notation. We use boldface to denote vectors, i.e., $\mathbf{m}$. For vector $\mathbf{m}$, we let $|\mathbf{m}|$ denote the number of components in the vector. When $\mathbf{m}[i] \in \{0,1\}^*$, we denote by $\mathbf{m}[i][j]$ the jth bit of the ith component of $\mathbf{m}$, i.e., the jth bit of $\mathbf{m}[i]$. On the other hand, when $\mathbf{c}[i]$ is a sequence, we let $\mathbf{c}[i][j]$ denote the jth value

in the sequence $\mathbf{c}[i]$. We sometimes abuse notation and treat vectors as sets. Specifically, if S is a set we may write $S \cup \mathbf{m}$ to denote $S \cup \{\mathbf{m}[1]\} \cup \{\mathbf{m}[2]\} \ldots$. If two adversaries $\mathcal{A}$ and $\mathcal{B}$ have access to different oracles with the same name (e.g., **NewMesg**) we sometimes write $\mathbf{NewMesg}_{\mathcal{B}}$ to mean $\mathcal{B}$'s version of the oracle. For $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$.

We fix pairing parameters $\mathsf{GP} = (\mathbb{G}, \mathbb{G}_T, p, \mathrm{e})$ where $\mathbb{G}, \mathbb{G}_T$ are groups of order prime p and the map $\mathrm{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficiently computable non-degenerate bilinear map. We let $\mathsf{T}_{\mathrm{exp}}(\mathbb{G})$ be the time to compute an exponentiation in the group $\mathbb{G}$. We let $\mathsf{T}_{\mathrm{op}}(\mathbb{G})$ be the time to compute a group operation in $\mathbb{G}$. For any group $\mathbb{G}$, let $\mathbb{G}^*$ denote the generators of $\mathbb{G}$.

Code-Based Games. We use code based games [4] for our security definitions. A game consists of numerous procedures including an **Initialize** procedure and a **Finalize** procedure. When an adversary $\mathcal{A}$ executes with the game, the **Initialize** procedure is executed first and its outputs are the initial inputs to adversary $\mathcal{A}$. Then $\mathcal{A}$ executes and its oracle queries are answered by the corresponding procedures of the game. When the adversary halts with some final output, this output is given as input to the **Finalize** procedure. The output of the **Finalize** procedure is then considered the output of the game. We let $G^{\mathcal{A}} \Rightarrow y$ be the event that game G, when executed with adversary $\mathcal{A}$, has output y. We abbreviate "$G^{\mathcal{A}} \Rightarrow \mathsf{true}$" by "$G^{\mathcal{A}}$". The running time of the adversary while playing the game is considered to be the running time of the adversary while playing the game plus the time to execute all of the game procedures during the execution.

Randomized Algorithms and Sampling from Groups. We have to model randomized algorithms carefully and in a particular way to define invertible sampling. We assume that all algorithms have access to a RNG Rand that is the only source of randomness in the system. On input a positive integer n, function Rand returns a value uniformly distributed in $\mathbb{Z}_n$. We stress that Rand is not viewed as having an underlying source of coins in the form of bits as in complexity-theoretic/Turing machine models. Rather, its operation is atomic and its outputs *are* the coins.

When we write $a \xleftarrow{\$} \mathbb{G}$ we mean that we run $i \xleftarrow{\$} \mathsf{Rand}(p)$, where $p = |\mathbb{G}|$, and let $a = g^i$ where g is a generator of $\mathbb{G}$. However, we also want to use publicly reversible sampling. A publicly reversible (PR) sampler Samp takes no input and, via access to Rand, outputs a point in $\mathbb{G}$ or the failure symbol $\perp$. It has sampling failure probability ζ if the probability that it outputs $\perp$ is at most ζ. We require that $\Pr[a' = a \mid a' \neq \perp] = 1/|\mathbb{G}|$ for all $a \in \mathbb{G}$, where the probability is over $a' \xleftarrow{\$} \mathsf{Samp}$.

If $(r_1, \ldots, r_s)$ is a sequence of non-negative integers, we let $\mathsf{Samp}[r_1, \ldots, r_s]$ be the result of running Samp with Rand replaced by the subroutine that returns r_i in response to the i-th query made to it, for $1 \leq i \leq s$. We require that there is an algorithm Samp^{-1} which on input $a \in \mathbb{G}$ outputs a sequence $(r_1, \ldots, r_s)$ such that $\mathsf{Samp}[r_1, \ldots, r_s] = a$. ($\mathsf{Samp}^{-1}$, as with any other algorithm, has access to

Rand.) Samp^{-1} also might fail (and output $\perp$). We call this the reverse sampling failure probability and denote it with θ.

Identity-Based Encryption. An Identity-based encryption scheme (IBE) is a tuple of algorithms $\Pi = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ with identity space IdSp, message space MsgSp, and the following properties. The parameter generation algorithm Pg takes no input and outputs a public parameter string par and a master secret key msk. The identity key generation algorithm Kg takes as input the public parameter string par, the master secret key msk, and an identity id, and outputs a secret key sk for identity id. The encryption algorithm Enc takes as input the public parameters par, an identity id, and a message M, and outputs a ciphertext C. Lastly, the decryption algorithm Dec takes as input the public parameters par, an identity secret key sk, and a ciphertext C, and outputs either a message M or a failure symbol $\perp$. We say that an IBE scheme has completeness error ϵ if the probability that $\mathsf{Dec}(\mathsf{par}, sk, \mathsf{id}, \mathsf{Enc}(\mathsf{par}, \mathsf{id}, M)) = M$ is $\geq 1 - \epsilon$ for all $\mathsf{id} \in \mathsf{IdSp}$, all $M \in \mathsf{MsgSp}$, all $(\mathsf{par}, \mathsf{msk}) \in [\mathsf{Pg}]$, and all $sk \in [\mathsf{Kg}(\mathsf{par}, \mathsf{msk}, \mathsf{id})]$, where the probability is taken over the coins used in encryption.

A one-bit IBE scheme $\Pi = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ is one with $\mathsf{MsgSp} = \{0,1\}$, while an ℓ-bit IBE scheme has $\mathsf{MsgSp} = \{0,1\}^\ell$. We will build ℓ-bit IBE schemes from one-bit IBE schemes as follows. Given one-bit IBE scheme Π as above, let $\Pi^\ell = (\mathsf{Pg}^\ell, \mathsf{Kg}^\ell, \mathsf{Enc}^\ell, \mathsf{Dec}^\ell)$ be an ℓ-bit IBE scheme defined as follows: parameter and key generation are unchanged, i.e., $\mathsf{Pg}^\ell = \mathsf{Pg}$ and $\mathsf{Kg}^\ell = \mathsf{Kg}$. The encryption algorithm Enc^ℓ, on input par, id, $M \in \{0,1\}^\ell$, outputs $\mathsf{Enc}(\mathsf{par}, \mathsf{id}, M[1]) \parallel \ldots \parallel \mathsf{Enc}(\mathsf{par}, \mathsf{id}, M[\ell])$, where $M[i]$ is the ith bit of M. In other words, encryption encrypts each bit separately and concatenates the resulting ciphertexts. Decryption works in the obvious way: decrypt each ciphertext component separately to learn individual bits. It is easy to see that if Π has ϵ completeness error, then the resulting ℓ-bit scheme has completeness error at most $\ell \cdot \epsilon$.

The standard notion of security for IBE schemes is indistinguishability under chosen plaintext attack (IND-CPA) [9]. We define the IND-CPA advantage of an IND-CPA adversary A against IBE scheme Π to be $\mathbf{Adv}^{\text{ind-cpa}}_{\Pi}(A) = 2 \cdot \Pr\left[\,\text{INDCPA}^A_\Pi \Rightarrow \mathsf{true}\,\right] - 1$, where game INDCPA can be found in Figure 2. An IND-CPA adversary interacts with game INDCPA, querying **LR** only once and on an identity $\mathsf{id}^* \in \mathsf{IdSp}$ that is never queried to **Extract** and on equal length messages $M_0, M_1 \in \mathsf{MsgSp}$. We note that adversaries may query the same identity id to **Extract** multiple times, since key generation is randomized.

We associate to encryption algorithm Enc the set $\mathsf{Coins}(\mathsf{par}, m)$. This is the set from which Enc draws its coins when encrypting message m using parameters par. Similarly, we let $\mathsf{Coins}(\mathsf{par}, \mathsf{id}, c, 1)$ be the set of coins $\{r \mid c = \mathsf{Enc}(\mathsf{par}, \mathsf{id}, 1; r)\}$.

3 Security against Selective Opening Attacks

In this section we formalize SOA security for IBE, closely following the formalizations from [2]. Before proceeding, we need two definitions. A (k, ℓ)-message sampler is a randomized algorithm $\mathcal{M}$ that on input string $\alpha \in \{0,1\}^*$ outputs

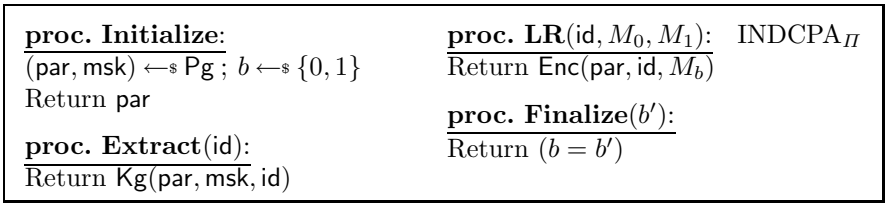

Fig. 2. The IBE IND-CPA Game

a vector of messages $\mathbf{m}$ such that $|\mathbf{m}| = k$ and each $\mathbf{m}[i] \in \{0,1\}^\ell$. A relation $\mathcal{R}$ is any randomized algorithm that outputs a single bit.

An soa-adversary is one that runs with game REAL making one query to **NewMesg** before making one query to **Corrupt**; it may make one or more queries to **Extract** at any time during the game. An soa-simulator is an adversary that runs with game SIM, makes one query to **NewMesg** and later makes one query to **Corrupt**. It makes no **Extract** queries. We define the soa-advantage of soa-adversary $\mathcal{A}$ against an IBE scheme Π with respect to a (k,ℓ)-message sampler $\mathcal{M}$, relation $\mathcal{R}$, and soa-simulator $\mathcal{S}$ as

$$\mathbf{Adv}^{\mathrm{soa}}_{\Pi,k,\mathcal{S},\mathcal{M},\mathcal{R}}(\mathcal{A}) = \Pr\left[\,\mathrm{REAL}^{\mathcal{A}}_{\Pi,k,\mathcal{M},\mathcal{R}} \Rightarrow 1\,\right] - \Pr\left[\,\mathrm{SIM}^{\mathcal{S}}_{\Pi,k,\mathcal{M},\mathcal{R}} \Rightarrow 1\,\right] .$$

Discussion. In game REAL (shown in Figure 3), the **Initialize** procedure runs the parameter generation algorithm and returns the scheme parameters to the adversary. The adversary then runs with oracles **NewMesg**, **Corrupt**, and **Extract**. The adversary may never query an identity to **Extract** that appears in a query to **NewMesg**.

The adversary may query the **NewMesg** oracle once with a vector of identities **id** and a string α that is meant to capture state to pass on to the message sampler. Procedure **NewMesg**, on input **id** and α, samples a vector of messages from the message sampling algorithm $\mathcal{M}$ and encrypts the entire vector using independent coins to the identities specified in **id**. This means that the ith component of the resulting ciphertext vector $\mathbf{c}$ is $\mathsf{Enc}(\mathsf{par}, \mathbf{id}[i], \mathbf{m}[i]; \mathbf{r}[i])$, the encryption of the ith message to the ith identity with the ith coins.

After querying the **NewMesg** oracle, the adversary may make one query to **Corrupt** with a set of indices $I \subseteq [k]$. These indices specify which ciphertexts from the vector $\mathbf{c}$ returned by **NewMesg** the adversary would like opened. The **Corrupt** procedure returns the messages and randomness used in **NewMesg** corresponding to indices in I. Additionally, at any time the adversary may query the **Extract** oracle on an identity of its choice and learn a secret key for that identity. We do not allow the adversary to query **Extract** on any identity appearing in the vector id queried to **NewMesg**.

Finally, the adversary halts with output *out* and the output of the game is the relation $\mathcal{R}$ applied to the message vector $\mathbf{m}$, the set of challenge IDs ChID, the corrupt set I, and the output *out*.

In game SIM (shown in Figure 4), the **Initialize** procedure does nothing and returns $\perp$ to the simulator. The simulator then runs with two oracles, **NewMesg**

proc. Initialize:
$(\mathsf{par}, \mathsf{msk}) \leftarrow\!\!\$\ \mathsf{Pg}$
Return par

proc. NewMesg$(\mathbf{id}, \alpha)$:
If $\mathbf{id} \cap \mathrm{ExID} \neq \emptyset$ then return $\perp$
$\mathrm{ChID} \leftarrow \mathrm{ChID} \cup \mathbf{id}$; $\mathbf{m} \leftarrow\!\!\$\ \mathcal{M}(\alpha)$
For i in 1 to k
  $\mathbf{r}[i] \leftarrow\!\!\$\ \mathsf{Coins}(\mathsf{par}, \mathbf{m}[i])$
  $\mathbf{c}[i] \leftarrow \mathsf{Enc}(\mathsf{par}, \mathbf{id}[i], \mathbf{m}[i]; \mathbf{r}[i])$
Return $\mathbf{c}$

proc. Extract(id):
If $\mathsf{id} \in \mathrm{ChID}$ then return $\perp$
$\mathrm{ExID} \leftarrow \mathrm{ExID} \cup \{\mathsf{id}\}$
$sk \leftarrow\!\!\$\ \mathsf{Kg}(\mathsf{par}, \mathsf{msk}, \mathsf{id})$
Return sk

proc. Corrupt(I):
Return $\mathbf{r}[I]$, $\mathbf{m}[I]$

proc. Finalize(out):
Return $\mathcal{R}(\mathbf{m}, \mathrm{ChID}, I, out)$

Fig. 3. Game $\mathrm{REAL}_{\Pi,k,\mathcal{M},\mathcal{R}}$

proc. Initialize:
Return $\perp$

proc. NewMesg$(\mathbf{id}, \alpha)$:
$\mathrm{ChID} \leftarrow \mathrm{ChID} \cup \mathbf{id}$; $\mathbf{m} \leftarrow\!\!\$\ \mathcal{M}(\alpha)$
Return $\perp$

proc. Corrupt(I):
Return $\mathbf{m}[I]$

proc. Finalize(out):
Return $\mathcal{R}(\mathbf{m}, \mathrm{ChID}, I, out)$

Fig. 4. Game $\mathrm{SIM}_{\Pi,k,\mathcal{M},\mathcal{R}}$

and **Corrupt**. On input an identity vector **id** and a string α, oracle **NewMesg** samples a vector $\mathbf{m}$ of messages using the message sampling algorithm $\mathcal{M}$ applied to the state string α. Nothing is returned to the simulator. The simulator is only allowed one **NewMesg** query. At a later time, the simulator may then make a single query to oracle **Corrupt** with a set of indices I and as a result will learn the messages in $\mathbf{m}$ corresponding to I. Finally, the simulator halts with output *out* and the output of the game is the relation $\mathcal{R}$ applied to the message vector $\mathbf{m}$, the set of challenge IDs ChID, the corrupt set I, and the output *out*.

As noted at the end of Section 1, we model adaptive sender corruptions while retaining standard IBE security against non-adaptive receiver corruptions. (Adaptive receiver corruptions would correspond to removing the restriction that **Extract** return $\perp$ when queried on a challenge identity.) Security against adaptive receiver corruptions seems out of reach of current techniques for PKE let alone IBE.

4 SOA Secure IBE from 1SPO IBE

A perfect one-sided public (1SP) opener for one-bit IBE scheme $\Pi = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ is an algorithm $\mathsf{OpToOne}$ that takes input parameters par, identity id, and ciphertext c, and has the following property: for all $\mathsf{par} \in [\mathsf{Pg}]$, all $\mathsf{id} \in \mathsf{IdSp}$, every $c \in [\mathsf{Enc}(\mathsf{par}, \mathsf{id}, 1)]$, and every $\bar{r} \in \mathsf{Coins}(\mathsf{par}, \mathsf{id}, c, 1)$,

$$\Pr[\, r \leftarrow\!\!\$\ \mathsf{OpToOne}(\mathsf{par}, \mathsf{id}, c)\ :\ r = \bar{r}\,] = \frac{1}{|\mathsf{Coins}(\mathsf{par}, \mathsf{id}, c, 1)|}\ .$$

We can weaken this definition slightly by considering opening algorithms that can fail with some probability δ, but in the case of success their output distribution is identical to the actual coin distribution. This is reflected as for all $\mathsf{par} \in [\mathsf{Pg}]$, all $\mathsf{id} \in \mathsf{IdSp}$, every $c \in [\mathsf{Enc}(\mathsf{par}, \mathsf{id}, 1)]$, and every $\bar{r} \in \mathsf{Coins}(\mathsf{par}, \mathsf{id}, c, 1)$,

$$\Pr[\, r \leftarrow\!\!\$\ \mathsf{OpToOne}(\mathsf{par}, \mathsf{id}, c)\ :\ r = \bar{r} \mid r \neq \bot\,] = \frac{1}{|\mathsf{Coins}(\mathsf{par}, \mathsf{id}, c, 1)|}\ .$$

Notice that the probability is only over the coins used by $\mathsf{OpToOne}$. We call such an $\mathsf{OpToOne}$ algorithm a δ-1SP opener and we also call an IBE scheme with a δ-1SP opener δ-one-sided publicly openable (δ-1SPO).

The idea of constructing encryption schemes with such one-sided opening originates with Canetti, Dwork, Naor, and Ostrovsky [13], who used PKE schemes with this property to build deniable PKE schemes. From translucent sets they get PKE schemes where an encryption of a 1 is pseudorandom while an encryption of a 0 is random. It is then possible to claim the encryption of a 1 was random and thus open it to a 0. More recently, in independent work, Fehr, Hofheinz, Kiltz, and Wee [22] used PKE schemes with a 1SPO property as a building block to achieve CC-SOA public-key encryption security. Of course, both of these works focus on PKE while we focus on IBE.

From 1SPO to SOA. We now state our main result: IND-CPA 1SPO one-bit IBE schemes lead to many-bit SOA-secure IBE schemes. Let $\Pi = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ denote a one-bit IBE scheme that is δ-one sided openable and let $\Pi^\ell = (\mathsf{Pg}^\ell, \mathsf{Kg}^\ell, \mathsf{Enc}^\ell, \mathsf{Dec}^\ell)$ the ℓ-bit IBE scheme built from Π as described in Section 2.

Theorem 1. *Let Π be a one-bit IBE scheme with a δ one-sided opener $\mathsf{OpToOne}$, and let Π^ℓ be the ℓ-bit scheme built from it. Let k be an integer, $\mathcal{A}$ an soa-adversary making at most q queries to **Extract**, $\mathcal{R}$ a relation, and $\mathcal{M}$ a (k, ℓ)-message sampler. Then there exists an soa-simulator $\mathcal{S}$ and an IND-CPA adversary $\mathcal{B}$ such that*

$$\mathbf{Adv}^{\mathrm{soa}}_{\Pi^\ell, k, \mathcal{M}, \mathcal{R}, \mathcal{S}}(\mathcal{A}) \leq k\ell \cdot \mathbf{Adv}^{\text{ind-cpa}}_{\Pi}(\mathcal{B}) + k\ell \cdot \delta\ ,$$

where $\mathsf{T}(\mathcal{S}) = \mathcal{O}(\mathsf{T}(\mathcal{A}) + k\ell \cdot \mathsf{T}(\mathsf{OpToOne}) + q \cdot \mathsf{T}(\mathsf{Kg}^\ell) + k \cdot \mathsf{T}(\mathsf{Enc}^\ell) + \mathsf{T}(\mathsf{Pg}^\ell))$ *and* $\mathsf{T}(\mathcal{B}) = \mathcal{O}(\mathsf{T}(\mathcal{A}) + \mathsf{T}(\mathcal{M}) + k\ell \cdot \mathsf{T}(\mathsf{Enc}) + k\ell \cdot \mathsf{T}(\mathsf{OpToOne}) + \mathsf{T}(\mathcal{R}))$. □

The full proof is in [5]. We briefly sketch the ideas here. Simulator $\mathcal{S}$ runs $\mathcal{A}$ and gives it encryptions of all 0s. When $\mathcal{A}$ asks for some of the ciphertexts to be opened, $\mathcal{S}$ queries its own **Corrupt** oracle, learns the messages it needs to open the ciphertexts to, and then opens bit-by-bit. If it needs to open a ciphertext component to a 0, it simply gives $\mathcal{A}$ the coins it used when originally creating the ciphertext. If it needs to open a ciphertext to a 1, it uses the scheme's $\mathsf{OpToOne}$ algorithm to find the coins. The simulator then outputs the same output as $\mathcal{A}$. The IND-CPA security of the scheme will allow us to argue that the simulator is successful. We will do a hybrid over the ℓ individual components of the k messages sampled from $\mathcal{M}$, where in the ith hybrid game the first i bits sampled

from $\mathcal{M}$ are ignored and 0s are encrypted in their place. Thus, in the first hybrid game all bits sampled from $\mathcal{M}$ are accurately encrypted, while in the last hybrid game only 0s are encrypted. This hybrid causes the loss of a factor $k \cdot \ell$ in the theorem.

A natural question is why not prove Theorem 1 for $\ell = 1$ (meaning, show that any 1-bit 1SPO IBE scheme is SOA-secure) and then prove the general result that if 1-bit Π is SOA-secure then so is Π^ℓ for any ℓ? The answer is that we do not know how to prove this general result.

5 A First Attempt

As a first attempt at constructing a 1SPO IBE scheme, we try to adapt techniques from deniable PKE [13], in particular the idea that an encryption of a 0 should be "pseudorandom" while the encryption of a 1 is "random". A typical IBE scheme has ciphertexts consisting of a tuple of group elements with some structure. To make such a scheme 1SPO, a natural idea is to make the honestly-generated ciphertext tuple the encryption of a 0, and a tuple of random group elements an encryption of a 1. The secret key for an identity then contains information which helps test for this structure. Let us see what happens if we apply this idea to the IBE scheme of Boneh and Boyen (BB) [6], which has been the basis for many other IBE schemes including the Waters (W) IBE [33].

Recall in the BB scheme (and its variants) a ciphertext has the form $(C_1, C_2, C_3) = (\mathrm{e}(g_1, g_2)^s \cdot M, g^s, H(\mathbf{u}, \mathsf{id})^s)$, where different variants define the hash function H differently, $g, g_1, g_2, \mathbf{u}$ are part of the parameters, and s is chosen randomly by the encryptor. Now, if we follow the ideas described above for making the scheme 1SPO, encryptions of 0 would be $(C, C') = (g^s, H(\mathbf{u}, \mathsf{id})^s)$, while encryptions of 1 would be a pair of group elements chosen uniformly at random from $\mathbb{G} \times \mathbb{G}$.

Syntactically the scheme works, but it is unfortunately not IND-CPA secure. The reason is that distinguishing an encryption of a 0 from an encryption of a 1 is exactly the DDH problem in $\mathbb{G}$ and hence easy given the pairing. Given a ciphertext (C, C'), we can output 0 if $\mathrm{e}(g, C') = \mathrm{e}(C, H(\mathbf{u}, \mathsf{id}))$ and 1 otherwise. This is (with high probability) a correct decryption.

The fundamental issue is that in the BB scheme, the structure of the ciphertexts can be detected given only public parameters. The key idea in our two schemes is to destroy any structure that is publicly detectable.

6 A 1SPO IBE Scheme Based on the DLIN Assumption

The security of our first scheme will rely on the Decisional Linear Assumption [7]. The decisional linear game DLIN is found in Figure 5. We say the DLIN-advantage of an adversary A against GP is

$$\mathbf{Adv}^{\mathrm{dlin}}_{\mathsf{GP}}(A) = 2 \cdot \Pr\left[\, \mathrm{DLIN}^{A}_{\mathsf{GP}} \Rightarrow \mathsf{true} \,\right] - 1 \,.$$

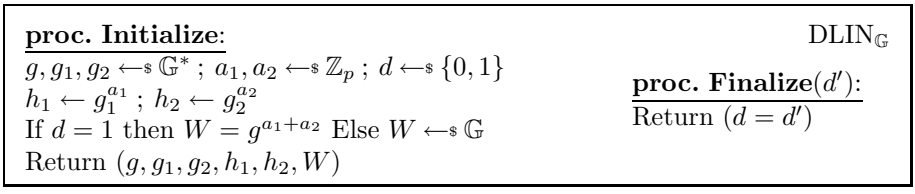

Fig. 5. The DLIN game for the decisional linear assumption

Scheme Description. Our IBE scheme $\mathsf{LoR} = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ is a one-bit version of the anonymous IBE scheme from Boyen and Waters [12] but using the Waters' hash function [33] for adaptive security. The name is short for "Linear or Random" to represent the fact that the encryption of a 0 consists of five group elements whose relationship is similar to that of the group elements in the decisional linear assumption, while an encryption of a 1 consists of five random group elements. The scheme will use a cyclic group $\mathbb{G}$ of prime order p with an efficiently computable pairing $\mathrm{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. We also require the group has a PR sampler Samp with failure probability ζ and corresponding inverse sampler Samp^{-1} with reverse failure probability θ. (The full version [5] gives details on how to instantiate such groups.) Let $\mathbb{G}^*$ denote the generators of $\mathbb{G}$. Let $1_{\mathbb{G}_T}$ be the identity element of the target group $\mathbb{G}_T$. Define hash function $H : \mathbb{G}^{n+1} \times \{0,1\}^n \to \mathbb{G}$ as $H(\mathbf{u}, \mathsf{id}) = \mathbf{u}[0] \prod_{i=1}^{n} \mathbf{u}[i]^{\mathsf{id}[i]}$, where $\mathsf{id}[i]$ is the ith bit of string id. This is the Waters' hash function [33]. The scheme LoR, shown in Figure 6, has message space $\{0,1\}$ and identity space $\{0,1\}^n$. The scheme has completeness error $1/p^2$.

We claim the scheme is δ-1SPO where $\delta \leq 5\theta$. The algorithm $\mathsf{OpToOne}$ simply runs Samp^{-1} on each of the five ciphertext components with independent failure probabilities θ.

The following says LoR is IND-CPA-secure based on DLIN. The proof combines techniques from [12,33,3] and can be found in the full version [5].

Theorem 2. Fix pairing parameters $\mathsf{GP} = (\mathbb{G}, \mathbb{G}_T, p, \mathrm{e})$ and an integer $n \geq 1$, and let $\mathsf{LoR} = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ be the one-bit IBE scheme associated to GP and $\mathsf{IdSp} = \{0,1\}^n$. Assume $\mathbb{G}$ is PR-samplable with sampling failure probability ζ. Let $\mathcal{A}$ be an IND-CPA adversary against LoR which has advantage $\epsilon = \mathbf{Adv}^{\text{ind-cpa}}_{\mathsf{LoR}}(\mathcal{A}) > 2^{n+1}/p + 5\zeta$ and makes at most $q \in [1\,..\,p\epsilon/9n]$ queries to its **Extract** oracle. Let

$$\delta = \frac{1}{2}\left(\frac{\epsilon}{2} - \frac{2^n}{p} - 5\zeta\right) .$$

Then there is a DLIN-adversary $\mathcal{B}$ such that

$$\mathbf{Adv}^{\text{dlin}}_{\mathsf{GP}}(\mathcal{B}) \geq \frac{\delta^2}{9qn + 3\delta} \quad \text{and} \quad \mathsf{T}(\mathcal{B}) = \mathsf{T}(\mathcal{A}) + \mathsf{T}_{\text{sim}}(n, q) \tag{1}$$

where $\mathsf{T}_{\text{sim}}(n, q) = \mathcal{O}(qn + (n + q)\mathsf{T}_{\text{exp}}(\mathbb{G}))$. □

Alg. Pg:

$g \leftarrow\!\!\$\ \mathbb{G}^*$; $\mathbf{u} \leftarrow\!\!\$\ \mathbb{G}^{n+1}$
$t_1, t_2, t_3, t_4 \leftarrow\!\!\$\ \mathbb{Z}_p^*$
$v_1 \leftarrow g^{t_1}$; $v_2 \leftarrow g^{t_2}$
$v_3 \leftarrow g^{t_3}$; $v_4 \leftarrow g^{t_4}$
$\mathsf{par} \leftarrow (g, \mathbf{u}, v_1, v_2, v_3, v_4)$
$\mathsf{msk} \leftarrow (t_1, t_2, t_3, t_4)$
Return $(\mathsf{par}, \mathsf{msk})$

Alg. $\mathsf{Kg}(\mathsf{par}, \mathsf{msk}, \mathsf{id})$:

$(g, \mathbf{u}, v_1, v_2, v_3, v_4) \leftarrow \mathsf{par}$
$(t_1, t_2, t_3, t_4) \leftarrow \mathsf{msk}$
$r_1, r_2 \leftarrow\!\!\$\ \mathbb{Z}_p$; $d_0 \leftarrow g^{r_1 t_1 t_2 + r_2 t_3 t_4}$
$d_1 \leftarrow H(\mathbf{u}, \mathsf{id})^{-r_1 t_2}$
$d_2 \leftarrow H(\mathbf{u}, \mathsf{id})^{-r_1 t_1}$
$d_3 \leftarrow H(\mathbf{u}, \mathsf{id})^{-r_2 t_4}$
$d_4 \leftarrow H(\mathbf{u}, \mathsf{id})^{-r_2 t_3}$
Return $(d_0, d_1, d_2, d_3, d_4)$

Alg. $\mathsf{Enc}(\mathsf{par}, \mathsf{id}, M)$:

$(g, \mathbf{u}, v_1, v_2, v_3, v_4) \leftarrow \mathsf{par}$
If $M = 0$ then
  $s, s_1, s_2 \leftarrow\!\!\$\ \mathbb{Z}_p$; $C_0 \leftarrow H(\mathbf{u}, \mathsf{id})^s$
  $C_1 \leftarrow v_1^{s-s_1}$; $C_2 \leftarrow v_2^{s_1}$
  $C_3 \leftarrow v_3^{s-s_2}$; $C_4 \leftarrow v_4^{s_2}$
Else
  For $i = 0$ to 4 do $C_i \leftarrow\!\!\$\ \mathsf{Samp}_{\mathbb{G}}()$
Return $(C_0, C_1, C_2, C_3, C_4)$

Alg. $\mathsf{Dec}(\mathsf{par}, sk, C)$:

$(g, \mathbf{u}, v_1, v_2, v_3, v_4) \leftarrow \mathsf{par}$
$(d_0, d_1, d_2, d_3, d_4) \leftarrow sk$
$(C_0, C_1, C_2, C_3, C_4) \leftarrow C$
If $\prod_{i=0}^{4} \mathrm{e}(C_i, d_i) = 1_{\mathbb{G}_T}$ then
  Return 0
Else return 1

Fig. 6. Scheme LoR based on Boyen-Waters IBE

7 A Scheme Based on Dual System IBE

General Subgroup Decision. We introduce the general subgroup decision problem and assumption as a generalization of several assumptions in the literature. An order-n group generator with security parameter k is an algorithm Gen that returns a pair $(\pi, \bar{\pi})$, where $\pi = (\langle\mathbb{G}\rangle, \langle\mathbb{G}_T\rangle, \langle\mathrm{e}\rangle, N)$ and $\bar{\pi} = (p_1, \ldots, p_n)$ with $p_1 < \ldots < p_n$ primes; $\mathbb{G}, \mathbb{G}_T$ groups and $\mathrm{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ a non-degenerate bilinear map; $p_i \in \{2^{k-1}, \ldots, 2^k - 1\}$ for $1 \le i \le n$; $N = p_1 \cdots p_n = |\mathbb{G}| = |\mathbb{G}_T|$. For $S \subseteq [n]$ we let $\mathbb{G}(S)$ denote the unique subgroup of $\mathbb{G}$ of order $\prod_{i \in S} p_i$. By $\mathbb{H}^*$ we denote the set of generators of a cyclic group $\mathbb{H}$.

The orthogonality property is that if $S_1, S_2 \subseteq [n]$ are disjoint and $g_i \in \mathbb{G}(S_i)$ $(i = 1, 2)$, then $\mathrm{e}(g_1, g_2) = 1_{\mathbb{G}_T}$. Now suppose $S_0, S_1 \subseteq [n]$ and given $T \in \mathbb{G}(S_b)$ we wish to determine $b \in \{0, 1\}$. Orthogonality makes this easy if we possess $g \in \mathbb{G}(S)$ where one of $S \cap S_0, S \cap S_1$ is empty and the other is not. The general subgroup decision assumption is that it is hard without such a g, even if we possess elements of any $G(S)$ for which $S \cap S_0, S \cap S_1$ are both empty or both not empty. Our formalization uses game $\mathrm{GSD}_{\mathsf{Gen}}$ of Figure 7. Adversary $\mathcal{A}$ must make exactly one **Ch** query, consisting of a pair $S_0, S_1 \subseteq [n]$, and this must be its first oracle query. Subsequently it can query $\mathbf{Gen}(S)$ on any $S \subseteq [n]$ and is allowed multiple queries to this oracle. It terminates by outputting a bit b' and its advantage is

$$\mathbf{Adv}^{\mathrm{gsd}}_{\mathsf{Gen}}(\mathcal{A}) = 2 \cdot \Pr\left[\, \mathrm{GSD}^{\mathcal{A}}_{\mathsf{Gen}} \Rightarrow \mathsf{true} \,\right] - 1 \,.$$

proc. Initialize:
$(\pi, \bar{\pi}) \leftarrow\!\!\$\ \mathsf{Gen}$; $b \leftarrow\!\!\$\ \{0,1\}$
$(\langle\mathbb{G}\rangle, \langle\mathbb{G}_T\rangle, \langle \mathrm{e}\rangle, N) \leftarrow \pi$
$(p_1, \ldots, p_n) \leftarrow \bar{\pi}$
Return π

proc. Ch(S_0, S_1):
If $(S_0 = \emptyset$ or $S_1 = \emptyset)$ then return $\perp$
$T \leftarrow\!\!\$\ \mathbb{G}(S_b)$
Return T

proc. Gen(S):
If $(S \cap S_0 = \emptyset) \wedge (S \cap S_1 \neq \emptyset)$
  then return $\perp$
If $(S \cap S_0 \neq \emptyset) \wedge (S \cap S_1 = \emptyset)$
  then return $\perp$
$g \leftarrow\!\!\$\ \mathbb{G}(S)$
Return g

proc. Finalize(b'):
Return $(b = b')$

Fig. 7. Game $\mathrm{GSD}_{\mathsf{Gen}}$

Discussion. Lewko and Waters [26] make several different subgroup decision assumptions about order-n group generators with $n = 3$, and [17] do the same with $n = 4$. Each of these corresponds to a particular choice of S_0, S_1 queried to **Ch**, and particular queries to **Gen**, in our game. (And hence can be formulated without these oracles. We note these papers also make other assumptions, some pertaining to $\mathbb{G}_T$, that we will not need or consider.) Although the authors make only a few specific assumptions, it is apparent that they would be willing to make any "allowed" one in the family, where "allowed" means that the adversary can get elements of $\mathbb{G}(S)$ only as long as $S \cap S_0, S \cap S_1$ are both empty or both not empty. Our aim in formulating GSD has been to make this more transparent, namely, to make the full family of potential choices explicit, thereby generalizing, unifying and explaining subgroup decisions assumptions from [10,26,17].

GSD may at first glance look like an "interactive" assumption. It isn't. The value n will be a fixed constant, eg. $n = 3$ for [26] and $n = 4$ for us. The GSD assumption is then just a compact way of stating a constant number —one for each subset $\{S_0, S_1\}$ of $2^{[n]}$ with $S_0, S_1 \neq \emptyset$— of non-interactive assumptions. (By non-interactive we mean the game has only **Initialize** and **Finalize** procedures, no oracles.)

We don't really need the full strength of GSD. As in previous works, we only need a few special cases, namely a few particular choices of queries S_0, S_1 to **Ch** and queries S to **Gen**. But we feel that stating GSD better elucidates the source of the assumptions, and it will allow more compact assumption and theorem statements.

Scheme Description. For our scheme we require a 4 group generator Gen with the property that the group $\mathbb{G}$ described by the first output of Gen has a PR sampler Samp with failure probability ζ and corresponding inverse sampler Samp^{-1} with reverse failure probability θ. (The full version [5] describes how we can instantiate such groups.) The scheme $\mathsf{BBoR} = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ associated to a order 4 group generator Gen is shown in Figure 8, where $\mathsf{IdSp} = \mathbb{Z}_{2^{4k-4}}$ and $\mathsf{MsgSp} = \{0,1\}$. (We use identity space $\mathbb{Z}_{2^{4k-4}}$ since N will vary but will always be at least 2^{4k-4}.) If $(C, C') \in [\mathsf{Enc}(\mathsf{par}, \mathsf{id}, 0)]$ and $(K, K') \in [\mathsf{Kg}(\mathsf{par}, \mathsf{msk}, \mathsf{id})]$,

Alg. Pg:
$(\pi, \bar{\pi}) \leftarrow\!\!\$\ \mathsf{Gen}$
$(\langle\mathbb{G}\rangle, \langle\mathbb{G}_T\rangle, \langle \mathrm{e}\rangle, N) \leftarrow \pi$
$(p_1, p_2, p_3, p_4) \leftarrow \bar{\pi}$
$g_1 \leftarrow\!\!\$\ \mathbb{G}(\{1\})^*$
$g_3 \leftarrow\!\!\$\ \mathbb{G}(\{3\})^*$; $g_4 \leftarrow\!\!\$\ \mathbb{G}(\{4\})^*$
$u_1 \leftarrow\!\!\$\ \mathbb{Z}_N$; $U_1 \leftarrow g_1^{u_1}$
$u_4 \leftarrow\!\!\$\ \mathbb{Z}_N$; $U_4 \leftarrow g_4^{u_4}$
$x_1 \leftarrow\!\!\$\ \mathbb{Z}_N$; $X_1 \leftarrow g_1^{x_1}$
$x_4 \leftarrow\!\!\$\ \mathbb{Z}_N$; $X_4 \leftarrow g_4^{x_4}$
$w_4 \leftarrow\!\!\$\ \mathbb{Z}_N$; $W_4 \leftarrow g_4^{w_4}$; $U_{14} \leftarrow U_1 U_4$
$W_{14} \leftarrow g_1 W_4$; $X_{14} \leftarrow X_1 X_4$
$\mathsf{par} \leftarrow (\pi, U_{14}, X_{14}, W_{14}, g_4)$
$\mathsf{msk} \leftarrow (g_1, U_1, X_1, g_3)$

Alg. $\mathsf{Dec}(\mathsf{par}, (K, K'), (C, C'))$:
$(\pi, U_{14}, X_{14}, W_{14}, g_4) \leftarrow \mathsf{par}$
If $\mathrm{e}(C, K) = \mathrm{e}(C', K')$ then return 0
Else return 1

Alg. $\mathsf{Kg}(\mathsf{par}, \mathsf{msk}, \mathsf{id})$:
// $\mathsf{id} \in \mathbb{Z}_{2^{4k-4}} \subseteq Z_N$
$(\pi, U_{14}, X_{14}, W_{14}, g_4) \leftarrow \mathsf{par}$
$(g_1, U_1, X_1, g_3) \leftarrow \mathsf{msk}$
$r, r_3, r_3' \leftarrow\!\!\$\ \mathbb{Z}_N$
$K \leftarrow g_1^r g_3^{r_3}$; $K' \leftarrow (U_1^{\mathsf{id}} X_1)^r g_3^{r_3'}$
Return (K, K')

Alg. $\mathsf{Enc}(\mathsf{par}, \mathsf{id}, M)$:
$(\pi, U_{14}, X_{14}, W_{14}, g_4) \leftarrow \mathsf{par}$
If $M = 0$ then
  $s \leftarrow\!\!\$\ \mathbb{Z}_N$; $t_4, t_4' \leftarrow\!\!\$\ \mathbb{Z}_N$
  $C \leftarrow (U_{14}^{\mathsf{id}} X_{14})^s g_4^{t_4}$
  $C' \leftarrow W_{14}^s g_4^{t_4'}$
Else $C, C' \leftarrow\!\!\$\ \mathsf{Samp}_{\mathbb{G}}()$
Return (C, C')

Fig. 8. Scheme BBoR based on composite order pairing groups

then decryption always succeeds. On the other hand, if $(C, C') \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{par}, \mathsf{id}, 1)$ and $(K, K') \leftarrow\!\!\$\ \mathsf{Kg}(\mathsf{par}, \mathsf{msk}, \mathsf{id})$ then $\Pr[\,\mathrm{e}(C, K) = \mathrm{e}(C', K')\,] \leq 8 \cdot 2^{-2k}$ where k is the security parameter associated to Gen.

We claim the scheme is δ-1SPO with $\delta \leq 2\theta$. The algorithm $\mathsf{OpToOne}$ runs Samp^{-1} on each of the two ciphertext components. Each component will give independent reverse sample failure probability of θ. The IND-CPA security of the scheme is captured by the following theorem, proven in our full version [5].

Theorem 3. *Let* Gen *be an order 4 group generator and let the resulting group* $\mathbb{G}$ *be PR-samplable with sampling failure probability* ζ*. Let* $\mathsf{BBoR} = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ *the associated IBE scheme defined above. For all adversaries* $\mathcal{A}'$ *making* q ***Extract*** *queries there exists an adversary* $\mathcal{B}$ *such that*

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathsf{BBoR}}(\mathcal{A}') \leq (9 + 2q) \cdot \mathbf{Adv}^{\text{gsd}}_{\mathsf{Gen}}(\mathcal{B}) + 4 \cdot \zeta\ .$$

Adversary $\mathcal{B}$ *makes at most 5 queries to* ***Gen*** *and runs in time at most* $\mathsf{T}(\mathcal{B}) = \mathsf{T}(\mathcal{A}') + \mathcal{O}(q \cdot \mathsf{T}_{\exp}(\mathbb{G}) + q \cdot \mathsf{T}(\gcd))$. □

Acknowledgements

Bellare is supported in part by NSF grants CNS-0627779, CCF-0915675 and CCF-0904380. Waters is supported in part by NSF grants CNS-0915361 and CNS-0952692, an AFOSR MURI award for "Collaborative policies and assured information sharing" (Project PRESIDIO), Department of Homeland Security

Grant 2006-CS-001-000001-02 (subaward 641), a Google Faculty Research award, and the Alfred P. Sloan Foundation. Part of Yilek's work was done at UCSD, supported in part by NSF grants CNS-0831536 and CNS-0627779.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)
2. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009)
3. 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)
4. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (2006)
5. Bellare, M., Waters, B., Yilek, S.: Identity-based encryption secure against selective opening attack. IACR ePrint Archive Report 2010/159
6. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
7. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
8. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM Journal on Computing 36(5), 915–942 (2006)
9. Boneh, D., Franklin, M.K.: Identity based encryption from the Weil pairing. SIAM Journal on Computing 32(3), 586–615 (2003)
10. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
11. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. Journal of Cryptology 17(4), 297–319 (2004)
12. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (Without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
13. Canetti, R., Dwork, C., Naor, M., Ostrovsky, R.: Deniable encryption. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 90–104. Springer, Heidelberg (1997)
14. Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multi-party computation. In: 28th ACM STOC, pp. 639–648. ACM Press, New York (May 1996)
15. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg (2003)
16. Canetti, R., Halevi, S., Katz, J.: Adaptively-secure, non-interactive public-key encryption. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 150–168. Springer, Heidelberg (2005)

17. Caro, A.D., Iovino, V., Persiano, G.: Fully secure anonymous HIBE with short ciphertexts. IACR ePrint Archive Report 2010/197
18. 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)
19. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001)
20. Damgård, I., Nielsen, J.B.: Improved non-committing encryption schemes based on a general complexity assumption. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 432–450. Springer, Heidelberg (2000)
21. Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.: Magic functions. Journal of the ACM 50(6), 852–921 (2003)
22. Fehr, S., Hofheinz, D., Kiltz, E., Wee, H.: Encryption schemes secure against chosen-ciphertext selective opening attacks. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 381–402. Springer, Heidelberg (2010)
23. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)
24. Hemenway, B., Libert, B., Ostrovsky, R., Vergnaud, D.: Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. IACR ePrint Archive Report 2009/088
25. Kol, G., Naor, M.: Cryptography and game theory: Designing protocols for exchanging information. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 320–339. Springer, Heidelberg (2008)
26. Lewko, A.B., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010)
27. Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: 12th SODA, pp. 448–457. ACM-SIAM (January 2001)
28. Nielsen, J.B.: Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 111–126. Springer, Heidelberg (2002)
29. Peikert, C.: Private Communication (May 2010)
30. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008)
31. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC, pp. 187–196. ACM Press, New York (May 2008)
32. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)
33. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)