Reducing Trust in the PKG in Identity Based Cryptosystems

Vipul Goyal

Department of Computer Science
University of California, Los Angeles
vipul@cs.ucla.edu

Abstract. One day, you suddenly find that a private key corresponding to your Identity is up for sale at e-Bay. Since you do not suspect a key compromise, perhaps it must be the PKG who is acting dishonestly and trying to make money by selling your key. How do you find out for sure and even prove it in a court of law?
This paper introduces the concept of Traceable Identity based Encryption which is a new approach to mitigate the (inherent) key escrow problem in identity based encryption schemes. Our main goal is to restrict the ways in which the PKG can misbehave. In our system, if the PKG ever maliciously generates and distributes a decryption key for an Identity, it runs the risk of being caught and prosecuted.
In contrast to other mitigation approaches, our approach does not require multiple key generation authorities.

1 Introduction

The notion of identity based encryption (IBE) was introduced by Shamir [Sha84] as an approach to simplify public key and certificate management in a public key infrastructure (PKI). Although the concept was proposed in 1984 [Sha84], it was only in 2001 that a practical and fully functional IBE scheme was proposed by Boneh and Franklin [BF01]. Their construction used bilinear maps and could be proven secure in the random oracle model. Following that work, a rapid development of identity based PKI has taken place. A series of papers [CHK03, BB04a, BB04b, Wat05, Gen06] striving to achieve stronger notions of security led to efficient IBE schemes in the standard model. There now exist hierarchical IBE schemes [GS02, HL02, BBG05], identity based signatures and authentication schemes [CC03, FS86, FFS88] and a host of other identity based primitive.

In an IBE system, the public key of a user may be an arbitrary string like an e-mail address or other identifier. This eliminates certificates altogether; the sender could just encrypt the message with the identity of the recipient without having to first obtain his public key (and make sure that the obtained public key is the right one). Of course, users are not capable of generating a private key for an identity themselves. For this reason, there is a trusted party called the private key generator (PKG) who does the system setup. To obtain a private key for his identity, a user would go to the PKG and prove his identity. The PKG would then generate the appropriate private key and pass it on to the user.

Since the PKG is able to compute the private key corresponding to any identity, it has to be completely trusted. The PKG is free to engage in malicious activities without any risk of being confronted in a court of law. The malicious activities could include: decrypting and reading messages meant for any user, or worse still: generating and distributing private keys for any identity. This, in fact, has been cited as a reason for the slow adoption of IBE despite its nice properties in terms of usability. It has been argued that due to the inherent key escrow problem, the use of IBE is restricted to small and closed groups where a central trusted authority is available [ARP03, LBD$^+$04, Gen03].

One approach to mitigate the key escrow problem problem is to employ multiple PKGs [BF01]. In this approach, the master key for the IBE system is distributed to multiple PKGs; that is, no single PKG has the knowledge of the master key. The private key generation for an identity is done in a threshold manner. This is an attractive solution and successfully avoids placing trust in a single entity by making the system distributed. However, this solution comes at the cost of introducing extra infrastructure and communication. It is burdensome for a user to go to several key authorities, prove his identity to each of them and get a private key component (which has to be done over a secure channel). Further, maintaining multiple independent entities for managing a single PKI might be difficult in a commercial setting (e.g., the PKI has to be jointly managed by several companies).

To the best of our knowledge, without making the PKG distributed, there is no known solution to mitigate the problem of having to place trust in the PKG.

A New Approach. In this paper, we explore a new approach to mitigate the above trust problem. Very informally, the simplest form of our approach is as follows:

- In the IBE scheme, there will be an exponential (or super-polynomial) number of possible decryption keys corresponding to every identity ID.
- Given one decryption key for an identity, it is intractable to find any other.
- A users gets the decryption key corresponding to his identity from the PKG using a secure *key generation protocol*. The protocol allows the user to obtain a single decryption key d_{ID} for his identity *without letting the PKG know which key he obtained*.
- Now if the PKG generates a decryption key d'_{ID} for that identity for malicious usage, with all but negligible probability, it will be different from the key d_{ID} which the user obtained. Hence the key pair $(d_{\mathsf{ID}}, d'_{\mathsf{ID}})$ is a cryptographic proof of malicious behavior of the PKG (since in normal circumstances, only one key per identity should be in circulation).

The PKG can surely decrypt all the user message *passively*. However, the PKG is severely restricted as far as the distribution of the private key d'_{ID} is concerned. The knowledge of the key d'_{ID} enables an entity E to go to the honest user U (with identity ID and having key d_{ID}) and together with him, sue the PKG by presenting the pair $(d'_{\mathsf{ID}}, d_{\mathsf{ID}})$ as a proof of fraud (thus potentially closing down its business or getting some hefty money as a compensation which can be happily shared by E and U). This means that if the PKG ever generates a decryption key for an identity for malicious purposes, it runs the risk that the key could fall into "right hands" which could be fatal.

The above approach can be compared to a regular (i.e., not identity based) PKIs. In a regular PKI, a user will go to a CA and get a certificate binding his identity with his public key. The CA could surely generate one more certificate binding a malicious public key to his identity. However, two certificates corresponding to the same identity constitute a cryptographic proof of fraud. Similarly in our setting, the PKG is free to generate one more decryption key for his identity. However, two decryption keys corresponding to the same identity constitute a proof of fraud. Of course, there are important differences. In a regular PKI, the CA has to actively send the fraudulent certificate to potential encrypters (which is risky for the CA) while in our setting, the PKG could just decrypt the user messages *passively*. However, we believe that the IBE setting is more demanding and ours is nonetheless a step in the right direction.

We call an identity based encryption scheme of the type discussed above as a *traceable identity based encryption* (T-IBE) scheme. This is to reflect the fact that if a malicious decryption key is discovered, it can be traced back either to the corresponding user (if his decryption key is the same as the one found) or to the PKG (if the user has a different decryption key). We formalize this notion later on in the paper. We remark that what we discussed above is a slight simplification of our T-IBE concept. Given a decryption key for an identity, we allow a user to compute certain other decryption keys for the same identity as long as all the decryption keys computable belong to the *same family* (a family can be seen as a subspace of decryption keys). Thus in this case, two decryption keys belonging to different families is a cryptographic proof of malicious behavior of the PKG.

Although the concept of T-IBE is interesting, we do not expect it to be usable on its own. We see this concept more as a stepping stone to achieving what we call a *black-box traceable identity based encryption* discussed later in this section.

Our Constructions. We formalize the notion of traceable identity based encryption and present two construction for it; one very efficient but based on a strong assumption, the other somewhat inefficient but based on the standard decisional BDH assumption.

Our first construction is based on the identity based encryption scheme recently proposed by Gentry [Gen06]. The scheme is the most efficient IBE construction known to date without random oracle. Apart from computational efficiency, it enjoys properties such as short public parameters and a tight security reduction (albeit at the cost of using a strong assumption). Remarkably, we are able to convert Gentry's scheme to a T-IBE scheme without any changes whatsoever to the basic cryptosystem. We are able to construct a secure key generation protocol as per our requirement for the basic cryptosystem and then present new proofs of security to show that the resulting system is a T-IBE system.

Our second construction of traceable identity based encryption is based on the decisional BDH assumption and uses the IBE scheme of Waters [Wat05] and the Fuzzy IBE scheme of Sahai and Waters [SW05] as building blocks. We remark that the construction is not very efficient and requires several pairing operations per decryption.

Black-Box Traceable Identity based Encryption. In traceable identity based encryption, as explained we consider the scenario when a PKG generates and tries to distribute a decryption key corresponding to an identity. T-IBE specifically assumes that the key is a *well-formed* decryption key. However, one can imagine a scenario where the PKG constructs a malformed decryption key which, when used in conjunction with some other decryption process, is still able to decrypt the ciphertexts. In the extreme case, there could a black box (using an unknown key and algorithm) which is able to decrypt the ciphertexts. Given such a box, a third party (such as the court of law), possibly with the cooperation of the PKG and the user, should be able to trace the box back to its source. That is, it should be able to determine whether it was the PKG or the user who was involved in the creation of this black box. We call such a system as *black-box traceable identity based encryption* system. This is a natural extension of the T-IBE concept and is closely related to the concept of black-box traitor tracing in broadcast encryption [CFN94, BSW06]. We leave the construction of a black-box T-IBE scheme as an important open problem.

We stress that black-box T-IBE is really what one would like to use in practice since in our current T-IBE constructions, it is perfectly plausible that the PKG could constructs a malformed decryption key and use it in conjunction with some other decryption process as remarked before. We intend the current work only to serve as an indication of what might be possible, and as motivation for further work in this direction.

Related Work. To our knowledge, T-IBE is the first approach for any kind of mitigation to the problem of trust in the PKG without using multiple PKGs. On the multiple PKGs side, Boneh and Franklin [BF01] proposed an efficient approach to make the PKG distributed in their scheme using techniques from threshold cryptography. Lee *et al* [LBD$^+$04] proposed a variant of this approach using multiple key privacy agents (KPAs). Also relevant are the new cryptosystems proposed by Gentry [Gen03] and Al-Riyami and Paterson [ARP03]. Although their main motivation was to overcome the key escrow problem, these works are somewhat orthogonal to ours since these cryptosystems are not identity based.

Other Remarks. We only consider identity based *encryption* in this paper. The analogue of T-IBE for identity based signatures appears straightforward to achieve. We also note that it may be possible to profitably combine our approach with the multiple PKG approach and exploit the mutual distrust between the PKGs. For example if two (or more) PKGs collude together to generate a decryption key for an identity, each PKG knows that it has left a cryptographic proof of its fraud with others. We can have a system where the PKG who presents this proof of fraud to the court is not penalized. Now since a PKG has the power to sue another (to close down its business and have one less competitor), this seems to be an effective fraud prevention idea. We do not explore this approach in this paper and defer it to future work.

2 Preliminaries

2.1 Bilinear Maps

We present a few facts related to groups with efficiently computable bilinear maps. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two multiplicative cyclic groups of prime order p. Let g be a generator of $\mathbb{G}_1$ and e be a bilinear map, $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. The bilinear map e has the following properties:

1. Bilinearity: for all $u, v \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$.

We say that $\mathbb{G}_1$ is a bilinear group if the group operation in $\mathbb{G}_1$ and the bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ are both efficiently computable. Notice that the map e is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

2.2 Complexity Assumptions

Decisional Bilinear Diffie-Hellman (BDH) Assumption Let $a, b, c, z \in \mathbb{Z}_p$ be chosen at random and g be a generator of $\mathbb{G}_1$. The decisional BDH assumption [BB04a, SW05] is that no probabilistic polynomial-time algorithm $\mathcal{B}$ can distinguish the tuple $(A = g^a, B = g^b, C = g^c, e(g, g)^{abc})$ from the tuple $(A = g^a, B = g^b, C = g^c, e(g, g)^z)$ with more than a negligible advantage. The advantage of $\mathcal{B}$ is

$$\left|\Pr[\mathcal{B}(A, B, C, e(g, g)^{abc}) = 0] - \Pr[\mathcal{B}(A, B, C, e(g, g)^z) = 0]\right|$$

where the probability is taken over the random choice of the generator g, the random choice of a, b, c, z in $\mathbb{Z}_p$, and the random bits consumed by $\mathcal{B}$.

Decisional Truncated q-ABDHE Assumption The truncated augmented bilinear Diffie-Hellman exponent assumption (truncated q-ABDHE assumption) was introduced by Gentry [Gen06] and is very closely related to the q-BDHE problem [BBG05] and the q-BDHI problem [BB04a]. Let g be a generator of $\mathbb{G}_1$. The decisional truncated q-ABDHE assumption is: given a vector of $q + 3$ elements

$$(g', g'^{(\alpha^{q+2})}, g, g^\alpha, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)})$$

no PPT algorithm $\mathcal{B}$ can distinguish $e(g,g')^{(\alpha^{q+1})}$ from a random element $Z \in \mathbb{G}_2$ with more than a negligible advantage. The advantage of $\mathcal{B}$ is defined as

$$\Big| \Pr[\mathcal{B}(g', g'^{(\alpha^{q+2})}, g, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}, e(g,g')^{(\alpha^{q+1})}) = 0] \\ - \Pr[\mathcal{B}(g', g'^{(\alpha^{q+2})}, g, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}, Z) = 0] \Big|$$

where the probability is taken over the random choice of generator $g, g' \in \mathbb{G}_1$, the random choice of $\alpha \in \mathbb{Z}_p$, the random choice of $Z \in \mathbb{G}_2$, and the random bits consumed by $\mathcal{B}$.

Computational q-BSDH Assumption The q-Strong Diffie-Hellman assumption (q-SDH assumption) was introduced by Boneh and Boyen [BB04c] for the construction of short signatures where it was also proven to be secure in the generic group model. This assumption was also later used in the construction of short group signatures [BBS04]. Let g be a generator of $\mathbb{G}_1$. The q-SDH assumption is defined in $(\mathbb{G}, \mathbb{G})$ as follows. Given a vector of $q+1$ elements

$$(g, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)})$$

no PPT algorithm $\mathcal{A}$ can compute a pair $(r, g^{1/(\alpha+r)})$ where $r \in \mathbb{Z}_p$ with more than a negligible advantage. The advantage of $\mathcal{A}$ is defined as

$$\Big| \Pr[\mathcal{A}(g, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}) = (r, g^{1/(\alpha+r)})] \Big|$$

The q-BSDH assumption is defined identically to q-SDH except that now $\mathcal{A}$ is challenged to compute $(r, e(g,g)^{1/(\alpha+r)})$. Note that the q-BSDH assumption is already implied by the q-SDH assumption.

2.3 Miscellaneous Primitives

Zero-knowledge Proof of Knowledge of Discrete Log Informally, a zero-knowledge proof of knowledge (ZK-POK) of discrete log protocol enables a prover to prove to a verifier that it possesses the discrete log r of a given group element R in question. Schnorr [Sch89] constructed an efficient number theoretic protocol to give a ZK-POK of discrete log.

A ZK-POK protocol has two distinct properties: the *zero-knowledge* property and the *proof of knowledge* properties. The former implies the existence of a simulator $\mathcal{S}$ which is able to simulate the view of a verifier in the protocol from scratch (i.e., without being given the witness as input). The latter implies the existence of a *knowledge-extractor* Ext which interacts with the prover and extracts the witness using rewinding techniques. For more details on ZK-POK systems, we refer the reader to [BG92].

1-out-of-2 Oblivious Transfer Informally speaking, a 1-out-of-2 oblivious transfer protocol allows a receiver to choose and receive exactly one of the two string from the sender, such that the other string is computationally hidden from the receiver and the choice of the receiver is computationally hidden from the sender.

Oblivious transfer was first introduced by [Rab81] while the 1-out-of-2 variant was introduced by [EGL85]. Various efficient constructions of 1-out-of-2 oblivious transfer are known based on specific assumptions such factoring or Diffie-Hellman [NP01, Kal05].

3 The Definitions and the Model

A Traceable Identity Based Encryption (T-IBE) scheme consists of five components.

Setup This is a randomized algorithm that takes no input other than the implicit security parameter. It outputs the public parameters PK and a master key MK.
Key Generation Protocol This is an interactive protocol between the public parameter generator PKG and the user U. The common input to PKG and U are: the public parameters PK and the identity ID (of U) for which the decryption key has to be generated. The private input to PKG is the master key MK. Additionally, PKG and U may use a sequence of random coin tosses as private inputs. At the end of the protocol, U receives a decryption key d_{ID} as its private output.
Encryption This is a randomized algorithm that takes as input: a message m, an identity ID, and the public parameters PK. It outputs the ciphertext C.
Decryption This algorithm takes as input: the ciphertext C that was encrypted under the identity ID, the decryption key d_{ID} for ID and the public parameters PK. It outputs the message m.
Trace This algorithm associates each decryption key to a family of decryption keys. That is, the algorithm takes as input a well-formed decryption key d_{ID} and outputs a decryption key family number n_F.

Some additional intuition about the relevance of trace algorithm is as follows. In a T-IBE system, there are a super-polynomial number of families of decryption keys. Each decryption key d_{ID} for an identity ID will belong to a unique decryption key family (denoted by the number n_F). Roughly speaking, in the definitions of security we will require that: given a decryption key belonging to a family, it should be intractable to find a decryption key belonging to a different family (although it may be possible to find another decryption key belonging to the same family).

To define security for a traceable identity based encryption system, we first define the following games.

The $\mathsf{IND\text{-}ID\text{-}CPA}$ game The $\mathsf{IND\text{-}ID\text{-}CPA}$ game for T-IBE is very similar to the $\mathsf{IND\text{-}ID\text{-}CPA}$ for standard IBE [BF01].

Setup The challenger runs the Setup algorithm of T-IBE and gives the public parameters PK to the adversary.
Phase 1 The adversary runs the Key Generation protocol with the challenger for several adaptively chosen identities $\mathsf{ID}^1, \ldots, \mathsf{ID}^q$ and gets the decryption keys $d_{\mathsf{ID}^1}, \ldots, d_{\mathsf{ID}^q}$.
Challenge The adversary submits two equal length messages m_0 and m_1 and an identity ID not equal to any of the identities quries in Phase 1. The challenger flips a random coin b and encrypts m_b with ID. The ciphertext C is passed on to the adversary.
Phase 2 This is identical to Phase 1 except that adversary is not allowed to ask for a decryption key for ID.
Guess The adversary outputs a guess b' of b.

The advantage of an adversary $\mathcal{A}$ in this game is defined as $\Pr[b' = b] - \frac{1}{2}$.

We note that the above game can extended to handle chosen-ciphertext attacks in the natural way by allowing for decryption queries in Phase 1 and Phase 2. We call such a game to be the $\mathsf{IND\text{-}ID\text{-}CCA}$ game.

The FindKey game The FindKey game for T-IBE is defined as follows.
Setup The adversary (acting as an adversarial PKG) generates and passes the public parameters PK and an identity ID on to the challenger. The challenger runs a sanity check on PK and aborts if the check fails.
Key Generation The challenger and the adversary then engage in the key generation protocol to generate a decryption key for the identity ID. The challenger gets the decryption d_{ID} as output and runs a sanity check on it to ensure that it is well-formed. It aborts if the check fails.
Find Key The adversary outputs a decryption key d'_{ID}. The challenger runs a sanity check on d'_{ID} and aborts if the check fails.

Let SF denote the event that $\mathsf{trace}(d'_{\mathsf{ID}}) = \mathsf{trace}(d_{\mathsf{ID}})$, i.e., d'_{ID} and d_{ID} belong to the same decryption key family. The advantage of an adversary $\mathcal{A}$ in this game is defined as $\Pr[SF]$.

We note that the above game can be extended to include a *decryption phase* where the adversary adaptively queries the challenger with a sequence of ciphertexts $C_1, \ldots, C_m$. The challenger decrypts C_i with its key d_{ID} and sends the resulting message m_i. This phase could potentially help the adversary deduce information about the decryption key family of d_{ID} if it is able to present a *maliciously formed* ciphertext and get the challenger try to decrypt it.

However, if the adversary was somehow restricted to presenting only well-formed ciphertexts, the decrypted message is guaranteed to contain *no information* about the decryption key family (since decryption using every well-formed key would lead to the same message). This can be achieved by adding a ciphertext sanity check phase during decryption. In both of our constructions, we take this route instead of adding a decryption phase to the FindKey game.

The ComputeNewKey game The ComputeNewKey game for T-IBE is defined as follows.

Setup The challenger runs the Setup algorithm of T-IBE and gives the public parameters PK to the adversary.

Key Generation The adversary runs the Key Generation protocol with the challenger for several adaptively chosen identities $\mathsf{ID}^1, \ldots, \mathsf{ID}^q$ and gets the decryption keys $d_{\mathsf{ID}^1}, \ldots, d_{\mathsf{ID}^q}$.

New Key Computation The adversary outputs two decryption keys d^1_{ID} and d^2_{ID} for an identity ID. The challenger runs a key sanity check on both of them and aborts if the check fails.

Let DF denote the event that $\mathsf{trace}(d^1_{\mathsf{ID}}) \neq \mathsf{trace}(d^2_{\mathsf{ID}})$, i.e., d^1_{ID} and d^2_{ID} belong to different decryption key families. The advantage of an adversary $\mathcal{A}$ in this game is defined as $\Pr[DF]$.

We also define a Selective-ID ComputeNewKey game where the adversary has to declare in advance (i.e., before the setup phase) the identity ID for which it will do the new key computation. The advantage of the adversary is similarly defined to be the probability of the event that it is able to output two decryption keys from different decryption key families *for the pre-declared identity* ID. This weakening of the game can be seen as similar to weakening of the IND-ID-CPA game by some previously published papers [CHK03, BB04a, SW05, GPSW06].

Definition 1 *A traceable identity based encryption scheme is* IND-ID-CPA *secure if all polynomial time adversaries have at most a negligible advantage in the* IND-ID-CPA *game, the* FindKey *game and the* ComputeNewKey *game.* IND-ID-CCA *security for T-IBE is defined similarly.*

4 Construction based on Gentry's Scheme

Our first construction of traceable identity based encryption is based on the identity based encryption scheme recently proposed by Gentry [Gen06]. The scheme is the most efficient IBE construction known to date without random oracle. Apart from computational efficiency, it enjoys properties such as short public parameters and a tight security reduction. This comes at the cost of using a stronger assumption known as the truncated q-ABDHE which is a variant of an assumption called q-BDHE introduced by Boneh, Boyen and Goh [BBG05].

Remarkably, we are able to convert Gentry's scheme to a T-IBE scheme without any changes whatsoever to the basic cryptosystem. We are able to construct a secure key generation protocol as per our requirement for the basic cryptosystem and then present new proofs of security to show the negligible advantage of the adversary in the three games of our T-IBE model. Our proofs are based on the truncated q-ABDHE and the q-BSDH assumption (see Section 2). The end result is a T-IBE scheme which is as efficient as the best known IBE scheme without random oracle. We view this fact as evidence that the additional traceability property does not necessarily come at the cost of a performance penalty.

4.1 The Construction

Let $\mathbb{G}_1$ be a bilinear group of prime order p, and let g be a generator of $\mathbb{G}_1$. In addition, let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ denote a bilinear map. A security parameter, κ, will determine the size of the groups.

As discussed before, the basic cryptosystem (i.e., Setup, Encryption and Decryption) is identical to Gentry's [Gen06]. For completeness, we describe the whole T-IBE scheme. Parts of this section are taken almost verbatim from [Gen06].

Setup The PKG picks random generators $g, h_1, h_2, h_3 \in \mathbb{G}_1$ and a random $\alpha \in \mathbb{Z}_p$. It sets $g_1 = g^\alpha$ and then chooses a hash function H from a family of universal one-way hash function. The published public parameters PK and the master key MK are given by

$$\mathrm{PK} = g, g_1, h_1, h_2, h_3, H \qquad \mathrm{MK} = \alpha$$

Key Generation Protocol This is the protocol through which a user U with an identity ID can securely get a decryption key d_{ID} from PKG . As in [Gen06], PKG aborts if $\mathsf{ID} = \alpha$. The key generation protocol proceeds as follows.

1. The user U selects a random $r \in \mathbb{Z}_p$ and sends $R = h_1^r$ to the PKG .
2. U gives to PKG a zero-knowledge proof of knowledge of the discrete log (as in Section 2) of R with respect to h_1.
3. The PKG now chooses three random numbers $r', r_{\mathsf{ID},2}, r_{\mathsf{ID},3} \in \mathbb{Z}_p$. It then computes the following values
$$(r', h'_{\mathsf{ID},1}), (r_{\mathsf{ID},2}, h_{\mathsf{ID},2}), (r_{\mathsf{ID},3}, h_{\mathsf{ID},3})$$
$$\text{where} \quad h'_{\mathsf{ID},1} = (Rg^{-r'})^{1/(\alpha-\mathsf{ID})} \quad \text{and} \quad h_{\mathsf{ID},i} = (h_i g^{-r_{\mathsf{ID},i}})^{1/(\alpha-\mathsf{ID})}, i \in \{2, 3\}$$
and sends them to the user U.
4. U computes $r_{\mathsf{ID},1} = r'/r$ and $h_{\mathsf{ID},1} = (h'_{\mathsf{ID},1})^{1/r}$. Note that since $h'_{\mathsf{ID},1} = (h_1^r g^{-r'})^{1/(\alpha-\mathsf{ID})}$, $h_{\mathsf{ID},1} = ((h_1^r)^{1/r}(g^{-r'})^{1/r})^{1/(\alpha-\mathsf{ID})} = (h_1 g^{-r_{\mathsf{ID},1}})^{1/(\alpha-\mathsf{ID})}$. It sets the decryption key $d_{\mathsf{ID}} = \{(r_{\mathsf{ID},i}, h_{\mathsf{ID},i}) : i \in \{1, 2, 3\}\}$.
5. U now runs a key sanity check on d_{ID} as follows. It computes $g^{\alpha-\mathsf{ID}} = g_1/g^{\mathsf{ID}}$ and checks if $e(h_{\mathsf{ID},i}, g^{\alpha-\mathsf{ID}}) \stackrel{?}{=} e(h_i g^{-r_{\mathsf{ID},i}}, g)$ for $i \in \{1, 2, 3\}$. U aborts if the check fails for any i.

At the end of this protocol, U has a well-formed decryption key d_{ID} for the identity ID.

Encryption To encrypt a message $m \in \mathbb{G}_2$ using identity $\mathsf{ID} \in \mathbb{Z}_p$, generate a random $s \in \mathbb{Z}_p$ and compute the ciphertext C as follows

$$C = (g_1^s g^{-s.ID},\ e(g, g)^s,\ m{\cdot}e(g, h_1)^{-s},\ e(g, h_2)^s e(g, h_3)^{s\beta})$$

where for $C = (u, v, w, y)$, we set $\beta = H(u, v, w)$.

Decryption To decrypt a ciphertext $C = (u, v, w, y)$ with identity ID, set $\beta = H(u, v, w)$ and test whether

$$y = e(u, h_{\mathsf{ID},2} h_{\mathsf{ID},3}{}^{\beta}) v^{r_{\mathsf{ID},2} + r_{\mathsf{ID},3}\beta}$$

If the above check fails, output $\perp$, else output

$$m = w \cdot e(u, h_{\mathsf{ID},1}) v^{r_{\mathsf{ID},1}}$$

For additional intuition about the system and its correctness, we refer the reader to [Gen06]. We also note that the ciphertext sanity check in the decryption algorithm rejects *all* invalid ciphertexts as shown in [Gen06].

Trace This algorithm takes a well-formed decryption key $d_{\mathsf{ID}} = \{(r_{\mathsf{ID},i}, h_{\mathsf{ID},i}) : i \in \{1, 2, 3\}\}$ and outputs the decryption key family number $n_F = r_{\mathsf{ID},1}$. Hence if $r_{\mathsf{ID},1} = r'_{\mathsf{ID},1}$ for two decryption keys d_{ID} and d'_{ID}, then $\mathsf{trace}(d'_{\mathsf{ID}}) = \mathsf{trace}(d_{\mathsf{ID}})$ (i.e., the two keys belong to the same decryption key family).

4.2 Security Proofs

Theorem 1 *The advantage of an adversary in the* IND-ID-CCA *game is negligible for the above traceable identity based encryption scheme under the decisional truncated q-ABDHE assumption.*

Proof Sketch: The proof in our setting very much falls along the lines of the proof of IND-ID-CCA security of Gentry's scheme [Gen06]. Here we just give a sketch highlighting the only difference from the one in [Gen06].

The only difference between [Gen06] and our setting is how a decryption key d_{ID} is issued for an identity ID. In the proof of [Gen06], PKG was free to choose a decryption key d_{ID} on its own and pass it on to the user. PKG in fact chose $r_{\mathsf{ID},i}$ using a specific technique depending upon the truncated q-ABDHE problem instance given. In our setting, however, PKG and the user U engage in a key generation protocol where $r_{\mathsf{ID},1}$ is jointly determined by both of them (via the choice of numbers r and r'). Hence PKG does not have complete control over $r_{\mathsf{ID},1}$.

The above problem can be solved as follows. PKG generates a decryption key $d_{\mathsf{ID}} = \{(r_{\mathsf{ID},i}, h_{\mathsf{ID},i}) : i \in \{1, 2, 3\}\}$ on its own exactly as in [Gen06] and then "forces" the output of U to be the above key during key generation. Recall that during the key generation protocol, U first chooses a random $r \in \mathbb{Z}_p$ and sends $R = h_1^r$ to the PKG . U then gives to PKG a zero-knowledge *proof of knowledge* of the discrete log of R. The proof of knowledge property of the proof system implies the existence of a *knowledge extractor* Ext (see Section 2). Using Ext on U during the proof of knowledge protocol, PKG can extract the discrete log r (by rewinding U during protocol execution) with all but negligible probability. Now PKG sets $r' = r r_{\mathsf{ID},1}$. It then sends $(r', h'_{\mathsf{ID},1} = h^r_{\mathsf{ID},1}), (r_{\mathsf{ID},2}, h_{\mathsf{ID},2}), (r_{\mathsf{ID},3}, h_{\mathsf{ID},3})$ to U.

The user U will compute $r_{\mathsf{ID},1} = r'/r, h_{\mathsf{ID},1} = (h'_{\mathsf{ID},1})^{1/r}$. Hence, PKG has successfully forced the decryption key d_{ID} to be the key chosen by it in advance. ■

Theorem 2 *Assuming that computing discrete log is hard in $\mathbb{G}_1$, the advantage of an adversary in the* FindKey *game is negligible for the above traceable identity based encryption scheme.*

Proof: Let there be a PPT algorithm $\mathcal{A}$ that has an advantage ϵ in the FindKey game in the above T-IBE construction. We show how to build a simulator $\mathcal{B}$ that is able to solve discrete log in $\mathbb{G}_1$ with the same advantage ϵ. $\mathcal{B}$ proceeds as follows.

$\mathcal{B}$ runs the algorithm $\mathcal{A}$ and gets the public parameters $\mathrm{PK} = (g, g_1, h_1, h_2, h_3)$ and the identity ID from $\mathcal{A}$. It then invokes the challenger, passes on h_1 to it and gets a challenge $R \in \mathbb{G}_1$. The goal of $\mathcal{B}$ would be to find the discrete log r of R w.r.t. h_1.

$\mathcal{B}$ engages in the key generation protocol with $\mathcal{A}$ to get a decryption key for ID as follows. It sends R to $\mathcal{A}$ and now has to give a *zero-knowledge* proof of knowledge of the discrete log of R. The zero-knowledge property of the proof system implies the existence of a simulator $\mathcal{S}$ which is able to successfully simulate the view of $\mathcal{A}$ in the protocol (by rewinding $\mathcal{A}$), with all but negligible probability. $\mathcal{B}$ uses the simulator $\mathcal{S}$ to simulate the required proof even without of knowledge of r. $\mathcal{B}$ then receives the string $(r', h'_{\mathsf{ID},1}), (r_{\mathsf{ID},2}, h_{\mathsf{ID},2}), (r_{\mathsf{ID},3}, h_{\mathsf{ID},3})$ from $\mathcal{A}$. As before, $\mathcal{B}$ runs a key sanity check by testing if $e(h_{\mathsf{ID},i}, g^{\alpha-\mathsf{ID}}) \stackrel{?}{=} e(h_i g^{-r_{\mathsf{ID},i}}, g)$ for $i \in \{2, 3\}$. For $i = 1$, $\mathcal{B}$ tests if $e(h'_{\mathsf{ID},i}, g^{\alpha-\mathsf{ID}}) \stackrel{?}{=} e(Rg^{-r'}, g)$. If any of these tests fail, $\mathcal{B}$ aborts as would an honest user in the key generation protocol.

Now with probability at least ϵ, $\mathcal{A}$ outputs a decryption key (passing the key sanity check and hence well-formed) d'_{ID} such that its decryption key family number n'_F equals the decryption key family number of the key d_{ID}, where d_{ID} is defined (but unknown to $\mathcal{B}$) as $(r'/r, (h'_{\mathsf{ID},1})^{1/r}, (r_{\mathsf{ID},2}, h_{\mathsf{ID},2}), (r_{\mathsf{ID},3}, h_{\mathsf{ID},3}))$. After computing n'_F from d'_{ID} (by running trace on it), $\mathcal{B}$ computes $r = r'/n'_F$. $\mathcal{B}$ outputs r as the discrete log (w.r.t. h_1) of the challenge R and halts. ■

Theorem 3 *The advantage of an adversary in the* ComputeNewKey *game is negligible for the above traceable identity based encryption scheme under the computational q-BSDH assumption*

Proof: Let there be a PPT algorithm $\mathcal{A}$ that has an advantage ϵ in the ComputeNewKey game in the above T-IBE construction. We show how to build a simulator $\mathcal{B}$ that is able to solve the computational q-BSDH assumption with the same advantage ϵ. $\mathcal{B}$ proceeds as follows.

The functioning of $\mathcal{B}$ in this proof is very similar to that of the simulator in the $\mathsf{IND\text{-}ID\text{-}CPA}$ proof of Gentry's scheme [Gen06]. $\mathcal{B}$ invokes the challenger and gets as input the q-BSDH problem instance $(g, g_1, g_2, \ldots, g_q)$, where $g_i = g^{(\alpha^i)}$.

$\mathcal{B}$ generates random polynomials $f_i(x) \in \mathbb{Z}_p[x]$ of degree q for $i \in \{1, 2, 3\}$. It computes $h_i = g^{f_i(\alpha)}$ using $(g, g_1, g_2, \ldots, g_q)$ and sends the public parameters $\mathrm{PK} = (g, g_1, h_1, h_2, h_3)$ to the algorithm $\mathcal{A}$.

$\mathcal{B}$ now runs the key generation protocol with $\mathcal{A}$ (possibly multiple times) to pass on the decryption keys $d_{\mathsf{ID}^1}, \ldots, d_{\mathsf{ID}^q}$ for the identities $\mathsf{ID}^1, \ldots, \mathsf{ID}^q$ chosen adaptively by $\mathcal{A}$. For an identity ID, $\mathcal{B}$ runs the key generation protocol as follows. If $\mathsf{ID} = \alpha$, $\mathcal{B}$ uses α to solve the q-BSDH problem immediately. Otherwise, let $F_{\mathsf{ID},i}(x)$ denote the $(q-1)$ degree polynomial $F_{\mathsf{ID},i}(x) = (f_i(x) - f_i(\mathsf{ID}))/(x - \mathsf{ID})$. $\mathcal{B}$ computes the decryption key $d_{\mathsf{ID}} = \{(r_{\mathsf{ID},i}, h_{\mathsf{ID},i}) : i \in \{1,2,3\}\}$ where $r_{\mathsf{ID},i} = f_i(\mathsf{ID})$ and $h_{\mathsf{ID},i} = g^{F_{\mathsf{ID},i}(\alpha)}$. Note that this is a valid private key since $h_{\mathsf{ID},i} = g^{(f_i(\alpha)-f_i(\mathsf{ID}))/(\alpha-\mathsf{ID})} = (h_i g^{-f_i(\mathsf{ID})})^{1/(\alpha-\mathsf{ID})}$. Now $\mathcal{B}$ *forces* the output of $\mathcal{A}$ to be the key d_{ID} during the key generation protocol (see proof of Theorem 1). For more details on why this decryption key appears to be correctly distributed to $\mathcal{A}$, we refer the reader to [Gen06].

Now with probability at least ϵ, $\mathcal{A}$ outputs two decryption keys (passing the key sanity check and hence well-formed) $d^1_{\mathsf{ID}} = \{(r^1_{\mathsf{ID},i}, h^1_{\mathsf{ID},i})\}$ and $d^2_{\mathsf{ID}} = \{(r^2_{\mathsf{ID},i}, h^2_{\mathsf{ID},i})\}$ for $i \in \{1,2,3\}$ for an identity ID such that $\mathsf{trace}(d^1_{\mathsf{ID}}) \neq \mathsf{trace}(d^2_{\mathsf{ID}})$. This means that $r^1_{\mathsf{ID},1} \neq r^2_{\mathsf{ID},1}$. $\mathcal{B}$ then computes

$$\begin{aligned}
&(h^1_{\mathsf{ID},1}/h^2_{\mathsf{ID},1})^{1/(r^2_{\mathsf{ID},1}-r^1_{\mathsf{ID},1})} \\
&= (h_1 g^{-r^1_{\mathsf{ID},1}}/h_1 g^{-r^2_{\mathsf{ID},1}})^{1/(r^2_{\mathsf{ID},1}-r^1_{\mathsf{ID},1})(\alpha-\mathsf{ID})} \\
&= g^{1/(\alpha-\mathsf{ID})}
\end{aligned}$$

Finally, $\mathcal{B}$ outputs $-\mathsf{ID}, g^{1/(\alpha-\mathsf{ID})}$ as a solution to the q-BSDH problem instance given and halts. ■

5 Construction based on Decisional BDH Assumption

Our second construction of traceable identity based encryption is based on the decisional BDH assumption which is considered to be relatively standard in the groups with bilinear maps. However, the construction is not very efficient and requires several pairing operations per decryption.

We use two cryptosystems as building blocks in this construction: the identity based encryption scheme proposed by Waters [Wat05] and the fuzzy identity based encryption (FIBE) scheme proposed by Sahai and Waters [SW05]. Our first idea is to use an IBE scheme derived from the FIBE construction of Sahai and Waters [SW05]. In FIBE, the encryption is done with a set of attributes which will be defined by the identity in our setting. Additionally, we add a set of *dummy attributes* in the ciphertext. During the key generation protocol, the user gets the set of attributes as defined by his identity as well as a certain subset of the dummy attributes. Very roughly, the subset is such that it can be used to decrypt the ciphertext part encrypted with dummy attributes.

The main properties we need (to add traceability) are derived from the above IBE scheme constructed using FIBE [SW05]. However, as is the case with FIBE, the IBE scheme is only secure in the selective-ID model. To achieve full security, we use the Waters cryptosystem [Wat05] *in parallel* with the FIBE scheme. We remark that Waters cryptosystem is only used to achieve full security and any other fully secure IBE scheme (e.g., [BB04b]) based on a standard assumption could be used. We treat the Waters cryptosystem as a black box as we do not require any specific properties from it. Although we are able to achieve full security in the IND-ID-CCA game, we do need to use the selective-ID model for the ComputeNewKey game.

5.1 The Construction

As before, $\mathbb{G}_1$ is a bilinear group of prime order p, and let g be a generator of $\mathbb{G}_1$. In addition, let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ denote a bilinear map.

We represent the identities as strings of a fixed length ℓ_{ID} (since an identity $\mathsf{ID} \in \mathbb{Z}_p$, ℓ_{ID} is the number of bits required to represent an element in $\mathbb{Z}_p$). Let ℓ_{sp} be a number which is decided by a statistical security parameter κ_s. Let $\ell = \ell_{ID} + \ell_{sp}$. We define the following sets: $S = \{1, \ldots, \ell\}, S_{ID} = \{1, \ldots, \ell_{ID}\}, S_{sp} = \{\ell_{ID} + 1, \ldots, \ell\}$. We shall denote the ith bit of the identity ID with ID_i. Our construction follows.

Setup Run the setup algorithm of the Waters cryptosystem [Wat05] and get the public parameters PK_w and master key MK_w. Now, for each $i \in S$, choose two numbers $t_{i,0}$ and $t_{i,1}$ uniformly at random from $\mathbb{Z}_p$ such that all 2ℓ numbers are different. Also choose a number y uniformly at random in $\mathbb{Z}_p$.

The published public parameters are $\mathrm{PK} = (\mathrm{PK}_w, \mathrm{PK}_{sw})$, where:

$$\mathrm{PK}_{sw} = (\{(T_{i,j} = g^{t_{i,j}}) : i \in S, j \in \{0,1\}\}, Y = e(g,g)^y, g)$$

The master key $\mathrm{MK} = (\mathrm{MK}_w, \mathrm{MK}_{sw})$, where:

$$\mathrm{MK}_{sw} = (\{(t_{i,j}) : i \in S, j \in \{0,1\}\}, y)$$

Key Generation Protocol The key generation protocol between PKG and a user U (with the identity ID) proceeds as follows.

1. U aborts if the published values in the set $\{T_{i,j} : i \in S, j \in \{0,1\}\}$ are not all different.
2. PKG generates a decryption key d_w for identity ID using MK_w as per the key generation algorithm of the Waters cryptosystem. It sends d_w to U.
3. PKG generates ℓ random numbers $r_1, \ldots, r_\ell$ from $\mathbb{Z}_p$ such that $r_1 + \cdots + r_\ell = y$. It computes $R_1 = g^{r_1}, \ldots, R_\ell = g^{r_\ell}$ and sends them to the user.
4. PKG computes the key components $d_{sw,i} = g^{r_i/t_{i,\mathsf{ID}_i}}$ for all $i \in S_{ID}$ and sends them to U. It also computes key components $d_{sw,i,j} = g^{r_i/t_{i,j}}$ for all $i \in S_{sp}, j \in \{0,1\}$ and stores them.
5. PKG and U then engage in ℓ_{sp} executions of a 1-out-of-2 oblivious transfer protocol where PKG acts as the sender and U acts as the receiver. In the ith execution (where $i \in S_{sp}$), the private input of PKG is the key components $d_{sw,i,0}, d_{sw,i,1}$ and the private input of U is a randomly selected bit b_i. The

private output of U is the key component d_{sw,i,b_i}. For each $i \in S_{sp}$, U now runs the following check:

$$e(R_i, g) \stackrel{?}{=} e(T_{i,b_i}, d_{sw,i,b_i})$$

U aborts if any of the above checks fails. This check roughly ensures that for a majority of indices $i \in S_{sp}$, the correct value r_i was used in the creation of $d_{sw,i,0}$ and $d_{sw,i,1}$. Thus for a majority of indices $i \in S_{sp}$, $d_{sw,i,0}$ and $d_{sw,i,1}$ were different from each other.

6. U sets $d_{sw} = (\{d_{sw,i}\}_{i \in S_{ID}}, \{b_i, d_{sw,i,b_i}\}_{i \in S_{sp}})$ and runs a key sanity check on d_{sw} by checking if:

$$Y \stackrel{?}{=} \prod_{i \in S_{ID}} e(T_{i,\mathsf{ID}_i}, d_{sw,i}) \prod_{i \in S_{sp}} e(T_{i,b_i}, d_{sw,i,b_i})$$

U aborts if the above check fails. Finally, U sets its decryption key $d_{\mathsf{ID}} = (d_w, d_{sw})$.

Encryption To encrypt a message $m \in \mathbb{G}_2$ under an identity ID, break the message into two random shares m_1 and m_2 such that $m_1 \oplus m_2 = m$. Now choose a random value $s \in \mathbb{Z}_p$ and compute the ciphertext $C = (C_w, C_{sw})$. C_w is the encryption of m_1 with ID using the public parameters PK_w as per the encryption algorithm of the Waters cryptosystem and C_{sw} is given by the following tuple.

$$(C' = m_2 Y^s, C'' = g^s, \{(C_i = T_{i,\mathsf{ID}_i}{}^s) : i \in S_{ID}\}, \{(C_{i,j} = T_{i,j}{}^s) : i \in S_{sp}, j \in \{0,1\}\})$$

Decryption To decrypt the ciphertext $C = (C_w, C_{sw})$ using the decryption key $d_{\mathsf{ID}} = (d_w, d_{sw})$, first run a ciphertext sanity check on C_{sw} by checking if:

$$\begin{aligned} &e(C_i, g) \stackrel{?}{=} e(T_{i,\mathsf{ID}_i}, C''), \quad i \in S_{ID}, \quad \text{and} \\ &e(C_{i,j}, g) \stackrel{?}{=} e(T_{i,j}, C''), \quad i \in S_{sp}, j \in \{0,1\} \end{aligned}$$

If any of the above check fails, output $\perp$. It is easy to see that this check ensures that $\{(C_i = T_{i,\mathsf{ID}_i}{}^s) : i \in S_{ID}\}, \{(C_{i,j} = T_{i,j}{}^s) : i \in S_{sp}, j \in \{0,1\}\}$ where s is the discrete log of C'' w.r.t. g. This ensure that *all* invalid ciphertexts are rejected. In the appendix, we sketch an alternative technique of doing ciphertext sanity check which requires only two pairing operations.

If the ciphertext sanity check succeeds, recover the share m_1 by running the decrypt algorithm of Waters cryptosystem on C_w using d_w. The share m_2 is recovered by the following computations:

$$\begin{aligned} &C' / \prod_{i \in S_{ID}} e(C_i, d_{sw,i}) \prod_{i \in S_{sp}} e(C_{i,b_i}, d_{sw,i,b_i}) \\ &= m_2 e(g,g)^{sy} / \prod_{i \in S_{ID}} e(g^{st_{i,\mathsf{ID}_i}}, g^{r_i/t_{i,\mathsf{ID}_i}}) \prod_{i \in S_{sp}} e(g^{st_{i,b_i}}, g^{r_i/t_{i,b_i}}) \\ &= m_2 e(g,g)^{sy} / \prod_{i \in S} e(g,g)^{sr_i} = m_2 \end{aligned}$$

Finally, output $m = m_1 \oplus m_2$.

Trace This algorithm takes a well-formed decryption key $d_{\mathsf{ID}} = (d_w, d_{sw})$ where the component $d_{sw} = (\{d_{sw,i}\}_{i \in S_{ID}}, \{b_i, d_{sw,i,b_i}\}_{i \in S_{sp}})$ and outputs the decryption key family number $n_F = b_{\ell_{ID}+1} \circ b_{\ell_{ID}+2} \circ \ldots \circ b_\ell$, where $\circ$ denotes concatenation.

Security proofs are omitted for the lack of space. They can be found in the full version.

6 Future Work

This work motivates several interesting open problems. The most important one of course is the construction of a *black-box* traceable identity based encryption as discussed in Section 1.

Our second construction based on the decisional BDH assumption is not very efficient and requires several pairing operations per decryption. It is an open problem to design more efficient T-IBE schemes based on an standard assumption. Further, the second construction used the Selective-ID ComputeNewKey game to prove security. It would be interesting to see if this restriction can be removed. Combining the T-IBE approach with the multiple PKG approach also seems to be a promising direction.

Finally, it remains to be seen if the same approach to mitigate the key escrow problem can be profitably used in other related setting like attribute based encryption [SW05, GPSW06].

7 Acknowledgements

We wish to thank the anonymous reviewers for many useful comments and suggestions. Thanks to Yevgeniy Dodis and Omkant Pandey for helpful discussions. Finally, thanks to Rafail Ostrovsky and Amit Sahai for their feedback and encouragement.

References

[ARP03] Sattam S. Al-Riyami and Kenneth G. Paterson. Certificateless public key cryptography. In Chi-Sung Laih, editor, *ASIACRYPT*, volume 2894 of *Lecture Notes in Computer Science*, pages 452–473. Springer, 2003.

[BB04a] Dan Boneh. and Xavier Boyen. Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles. In *Advances in Cryptology – Eurocrypt*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.

[BB04b] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459. Springer, 2004.

[BB04c] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer, 2004.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Cramer [Cra05], pages 440–456.

[BBS04] D. Boneh, X. Boyen, and H. Shacham. Short Group Signatures. In *Advances in Cryptology - CRYPTO 2004*, Lecture Notes in Computer Science. Springer, 2004.

[BF01] D. Boneh and M. Franklin. Identity Based Encryption from the Weil Pairing. In *Advances in Cryptology – CRYPTO*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.

[BG92] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In Ernest F. Brickell, editor, *CRYPTO*, volume 740 of *Lecture Notes in Computer Science*, pages 390–420. Springer, 1992.

[Bra90] Gilles Brassard, editor. *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*. Springer, 1990.

[BSW06] Dan Boneh, Amit Sahai, and Brent Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In Vaudenay [Vau06], pages 573–592.

[CC03] Jae Choon Cha and Jung Hee Cheon. An identity-based signature from gap diffie-hellman groups. In Yvo Desmedt, editor, *Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 18–30. Springer, 2003.

[CFN94] B. Chor, A. Fiat, and M. Naor. Tracing traitor. In Y. Desmedt, editor, *Advances in Cryptology – Crypto'94*, volume 839 of *LNCS*, pages 257–270. Springer, 1994.

[CHK03] R. Canetti, S. Halevi, and J. Katz. A Forward-Secure Public-Key Encryption Scheme. In *Advances in Cryptology – Eurocrypt*, volume 2656 of *LNCS*. Springer, 2003.

[Cra05] Ronald Cramer, editor. *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*. Springer, 2005.

[EGL85] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.

[FFS88] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *J. Cryptology*, 1(2):77–94, 1988.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[Gen03] Craig Gentry. Certificate-based encryption and the certificate revocation problem. In Eli Biham, editor, *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 272–293. Springer, 2003.

[Gen06] Craig Gentry. Practical identity-based encryption without random oracles. In Vaudenay [Vau06], pages 445–464.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM Conference on Computer and Communications Security*, pages 89–98. ACM, 2006.

[GS02] Craig Gentry and Alice Silverberg. Hierarchical id-based cryptography. In *ASIACRYPT*, pages 548–566, 2002.

[HL02] Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 466–481. Springer, 2002.

[Kal05] Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. In Cramer [Cra05], pages 78–95.

[LBD+04] Byoungcheon Lee, Colin Boyd, Ed Dawson, Kwangjo Kim, Jeongmo Yang, and Seungjae Yoo. Secure key issuing in id-based cryptography. In James M. Hogan, Paul Montague, Martin K. Purvis, and Chris Steketee, editors, *ACSW Frontiers*, volume 32 of *CRPIT*, pages 69–74. Australian Computer Society, 2004.

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *SODA*, pages 448–457, 2001.

[Rab81] M. O. Rabin. How to exchange secrets by oblivious transfer. In *TR-81, Harvard*, 1981.

[Sch89] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Brassard [Bra90], pages 239–252.

[Sha84] A. Shamir. Identity Based Cryptosystems and Signature Schemes. In *Advances in Cryptology – CRYPTO*, volume 196 of *LNCS*, pages 37–53. Springer, 1984.

[SW05] A. Sahai and B. Waters. Fuzzy Identity Based Encryption. In *Advances in Cryptology – Eurocrypt*, volume 3494 of *LNCS*, pages 457–473. Springer, 2005.

[Vau06] Serge Vaudenay, editor. *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*. Springer, 2006.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In Cramer [Cra05], pages 114–127.

Appendix

A Efficient Ciphertext Sanity Check in the Second Construction

To decrypt the ciphertext $C = (C_w, C_{sw})$ using the decryption key $d_{\mathsf{ID}} = (d_w, d_{sw})$, the efficient ciphertext sanity check on C_{sw} is run as follows. First choose $\ell_{ID} + 2\ell_{sp}$ random numbers $s_{i,\mathsf{ID}_i}, i \in S_{ID}$ and $s_{i,j}, i \in S_{sp}, j \in \{0,1\}$. Now check if:

$$e\left(g, \prod_{i \in S_{ID}} C_i^{s_{i,\mathsf{ID}_i}} \prod_{i \in S_{sp}, j \in \{0,1\}} C_{i,j}^{s_{i,j}}\right) \stackrel{?}{=} e\left(C'', \prod_{i \in S_{ID}} T_{i,\mathsf{ID}_i}^{s_{i,\mathsf{ID}_i}} \prod_{i \in S_{sp}, j \in \{0,1\}} T_{i,j}^{s_{i,j}}\right)$$

If the above check fails, output $\perp$. It can be shown that the above check rejects an invalid ciphertext with all but negligible probability (while the previous check was *perfect*).