Making *Any* Identity-Based Encryption Accountable, Efficiently

Aggelos Kiayias[1] and Qiang Tang[2](✉)

[1] National and Kapodistrian University of Athens, Athens, Greece
aggelos@di.uoa.gr
[2] University of Connecticut, Storrs, USA
qtang84@gmail.com

Abstract. Identity-Based Encryption (IBE) provides a compelling solution to the PKI management problem, however it comes with the serious privacy consideration that a trusted party (called the PKG) is required to generate (and hence also know) the secret keys of all users. This inherent key escrow problem is considered to be one of the major reasons hindering the wider utilization of IBE systems. In order to address this problem, Goyal [20] introduced the notion of accountable authority IBE (A-IBE), in which a judge can differentiate the PKG from the user as the source of a decryption software. Via this "tracing" mechanism, A-IBE deters the PKG from leaking the user's secret key and hence offers a defense mechanism for IBE users against a malicious PKG.

All previous works on A-IBE focused on specialized constructions trying to achieve different properties and efficiency enhancements. In this paper for the first time we show how to add accountability to *any* IBE scheme using oblivious transfer (OT), with almost the same ciphertext efficiency as the underlying IBE. Furthermore, we extend our generic construction to support identity reuse without losing efficiency. This property is desirable in practice as users may accidentally lose their secret keys and they -naturally- prefer not to abandon their identities. How to achieve this property was open until our work. Along the way, we first modify the generic construction and develop a new technique to provide public traceability generically.

1 Introduction

Identity-Based Encryption (IBE) was introduced by Shamir [31], to remove the need for maintaining a certificate based public-key infrastructure (PKI). Long time after the concept was proposed, Boneh and Franklin constructed the first practical IBE [8] in the random oracle model [4]. Since then, IBE has gotten more attention and a lot of alternative schemes have emerged with an extended set of properties, cf. [5,6,11,19,22,29,32,33].

Although significant progress has been made in constructing secure and efficient IBE schemes, a critical problem of IBE is that a trusted authority, called PKG, is required to generate secret keys for all users. The possibility of the corruption of this authority (or just her temporary misbehavior due to an insider

G. Pernul et al. (Eds.): ESORICS 2015, Part I, LNCS 9326, pp. 326–346, 2015.
DOI: 10.1007/978-3-319-24174-6_17

attack) is considered one of the most important reasons hindering the deployment of IBE systems in practice [1,18,21]. The problem is inherent since there is no user-side secret that is used when generating the secret key corresponding to an arbitrarily formed identity; it follows that there is no *built-in* incentive for the PKG in a standard IBE system to protect the users' secret information.

Beyond the obvious privacy problem (the unavoidable fact that the PKG can decrypt all users' ciphertexts) there is also a more serious attack that can take place: the PKG may share the users' secret keys. One may address this by arguing that such malicious behavior can be detectable by the user: for instance, a decryption program B leaked to the public (e.g., uploaded on a public forum) can be noticed by the user. In such case, the user could conceivably bring the program to court and sue the PKG, thus the PKG would be deterred from such behavior. However, notice that both user and PKG are capable of producing B thus the device itself can not be used as conclusive proof about who is at fault.

In order to make the above detect-then-punish mechanism effective, Goyal introduced the concept of accountable authority IBE, (A-IBE in short) [20], where a convincing proof can be provided from which a judge can make a decision about who is at fault. In order to achieve this characteristic, every identity must be corresponded with super-polynomially many secret keys, and the PKG and the user jointly generate a secret key for the user so that the PKG does not know which key is chosen by the user. Using the secret key received by the user, any third party, a judge for example, can tell whether the decryption device is made from the user secret key or not, thus the judge (and the public) can identify unequivocally the creator of the device. A number of works followed up this seminal result, [21,25,26,28,34], further refining the notion of A-IBE.

Still, the adoption of A-IBE in practice is hindered by a couple of facts. First, many constructions are inefficient (in the sense that they require linear in the security parameter number of group elements, cf. Fig. 1) or that the designs are incompatible with existing practical deployments such as RFC 5091 [12]. Second, when a user accidentally loses his key, in all existing A-IBE schemes, the user and the PKG have to discard this identity and generate a new key for the user using a different identity (otherwise, it enables malicious users to frame the PKG). This artifact brings users annoying inconvenience. These put forth the main motivations in our work: is it possible to add accountability to *any* existing (that is potentially already deployed, e.g., RFC 5091) IBE system, with a minimum cost? furthermore, we ask whether such generic transformation can be extended to allow identity reuse, without losing efficiency? If such transformation exists, users may choose to "upgrade" their IBE scheme to be accountable without requiring a modification to the basic algorithms of the underlying IBE.

Our Contributions. In this work, we address both problems listed above. First, we propose a generic construction of an A-IBE (in the so-called weak black-box model with full security against malicious users, see definition in Appendix A) that uses any existing IBE in a black-box way. And this generic construction has ciphertext size only 2 times the underlying IBE ciphertext size. (we call this construction S-I). The key observation behind our construction is that users can choose from a set of secret-keys that are based on an extended form of their identity. When encrypting messages it is possible for the sender to use only two ciphertexts to guarantee an honest user to decrypt. However, it is also possible to generate a set of tracing ciphertexts that can reveal part of the "fingerprint" of the secret-key that was assigned obliviously to the user by the PKG. The presence of the partial fingerprint in a user decoder that is found publicly incriminates the user, otherwise, incriminates the PKG.

We then consider how to allow identity reuse. This property is not known whether achievable before, even with specifically tailored constructions. We achieve it while maintaining the generic nature and the small size ciphertext. The main challenge for achieving identity reuse in A-IBE setting is that a malicious user can obtain multiple secret keys corresponding to the same identity by claiming to the PKG that she lost the key. Such malicious user could then implement a pirate box B using one key, and reveal another key to the judge. A secret key tracing algorithm may erroneously accuse the PKG, as, by definition, the key used to implement B is different to the key that the user is currently using.

Our strategy is to add public traceability to our generic construction that will enable the judge to differentiate among all the secret keys that were ever obtained by a user for the same identity. Note that in S-I, part of the user fingerprint is recovered, if there is a public reference for the user fingerprint, it might be possible for the judge to check whether the recovered string matches. In order to implement this idea, we improve the generic construction S-I to allow the tracing algorithm to recover the *whole* "fingerprint" while maintaining the ciphertext size still to be small (at most logarithmic overhead, and we call it S-II). With this new feature of S-II we developed, it is possible to deposit the fingerprint (using a one way function) that the user chooses for selecting the secret key to the PKG in a secure way so that: (i) it enables the judge to use a public tracing key T to determine whether a recovered fingerprint matches the fingerprint, and (ii) it prevents a malicious PKG from producing a pirate box without being traced with the help of T. The main technical part is to design a proper one way function for the secure deposit of the bitstring, together with an efficient zero-knowledge proof for the consistency between the privately deposited fingerprint and that used in the OT protocol, bit by bit.

The intuition for S-II follows from the observation that if the "fingerprint" is generated from an error correcting code, a linear fraction of it could be enough to reveal the whole string. With a careful probabilistic analysis, we see that with slightly longer ciphertexts, one is able to retrieve a larger fraction of the fingerprint from a pirate box. (this new mechanism also allows the length of the fingerprint to be reduced asymptotically, so as the secret key size). This feature of S-II makes it a steppingstone for further allowing identity re-use and public traceability. Our A-IBE tracing mechanisms are inspired by previous works related to traitor tracing [10] and leakage-deterring cryptosystems [24].

With such public traceability, the scheme can be further extended to support identity reuse. Each identity now will have multiple extended forms (instead of one in S-II), and for each extended form indexed by a state, the user can use

	G-I	G-II	GLSW08	LV09	SS11	LDZW13	YCZY14	S-I	S-III
Generic	no	no	no	no	no	no	no	yes	yes
Ciphertext size	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$	O(1)	$O(\lambda)$	O(1)	O(1)	O(1)	O(1)
Malicious User	s	s	s	a	a	a	a	a	a
Malicious PKG	w	bb_0	bb_1	bb_0	bb_1	bb_0	bb_1	bb_0	bb_0
Public Traceable	no	no	no	no	no	yes	no	no	yes
ID Reuse	no	no	no	no	no	no	no	no	yes

Fig. 1. Comparisons of all existing A-IBEs, ciphertext size means the number of group elements; ‘s’ means selective, ‘a’ means adaptive; w, bb_0, bb_1 mean white box, weak black-box and full black-box traceability respectively; S-I, S-III are our constructions.

an independent string as a fingerprint to request one secret key. During the i-th key generation protocol for an identity, the PKG will store a public tracing key T_i and the updated state about the current version of the extended form for each identity in a public directory. The encryption algorithm will use the current version of the extended form of identity. The tracing algorithm will run on all versions of the extended form of the identity, extract (potentially multiple) fingerprints; subsequently, it will check whether they match the public tracing keys. In this way, the tracing algorithm can decide that the key inside the pirate box is the one the user is currently using or whether it is one of the keys claimed to be lost, or is a key originating from the PKG. A malicious user can never frame the PKG using a key claimed to be lost, and a PKG can not evade the tracing algorithm if she ever leaks a decryption box for the user identity (even for previous versions of extended form of identity).

Note that after adding public traceability and id-reuse to our generic construction, the ciphertext efficiency and the generic nature are still the same as in S-II. The model that T has to be stored for each user is the same as the only existing paper [25] (that was based on Gentry IBE [19]) providing public traceability.[1] Finally it is worth to point out that our construction allows these two properties to be optional services by the PKG and the user may opt-in or opt-out to such properties at will when she requests a key from the PKG.

We remark that our generic transformations can go beyond IBE and can be easily adapted to apply to more advanced systems like attribute based encryption [22,29]. The performance comparison of all A-IBE schemes (including ours) is summarized in Fig. 1.

Related Work. In [20], Goyal proposed the notion of A-IBE and gave two constructions. The first one is traceable only in the white-box model (requires the key material of the pirate box) while the second one is in the weak black-box model. We call those constructions G-I, G-II and both have ciphertext size that includes a linear number of group elements. In the following work of [21], Goyal et al. proposed a construction having traceability in the full black-box model, but at the price of having (i) secret key and ciphertext size that has linear in the security parameter

[1] In fact, it is not hard to see (explained in Sect. 3.1) that the size of the public tracing key has to grow linearly with the number of users.

number of group elements, (ii) security against malicious users only in a selective model (where the adversary needs to commit to its move ahead at the beginning of the game). Libert and Vergnaud [26] made an improvement on G-I, and they gave an A-IBE with constant group elements in the ciphertext that is proven traceable in the weak black box model. Sahai and Seyalioglu [28] improved the security against dishonest users, and achieved full security against dishonest users, but their construction still has a linear size ciphertext. Lai et al. [25] proposed the first scheme with public traceability that the authority is required to store a public tracing key for each user which is later used to generate tracing ciphertext, and it is also traceable in the weak black-box model. Our public traceability can be based on any IBE and uses a different tracing technique that we can directly compare whether a recovered fingerprint matches the one contained in the public tracing key. Concurrent to our work an E-print technical report [34] proposed an A-IBE with traceability in the full black-box model, adaptive security against malicious user and constant size ciphertext under non-standard assumptions. All these works rely on a highly specific structure, specifically, Gentry IBE [19] as in [20,25,26,34] or fuzzy IBE [29] as in [20,21,28]; their techniques for accountability do not adapt in other settings straightforwardly (and specifically none can be applied to current real world IBE's such as those of RFC5091 directly). Also, none of those works allows public traceability (except [25]) or identity reuse.

There are also other proposals to deal with the key escrow problem in IBE. In [8], Boneh and Franklin gave a simple solution where multiple authorities distributively produce the PKG master secret key. However, in principle, those PKGs still may collude to leak user's secret leaving the user defenseless; Al-Riyami and Paterson proposed the concept of certificateless public key cryptography [1], and attempted to combine both the advantages of certificate-based PKI and IBE. The authority only has a partial secret key k_1, and it jointly generates secret key together with the user who has her own secret k_2. However, part of the public key must be in a specific form corresponding to k_2 and thus it can not be as expressive as IBE. Hence such systems may be of more narrow applicability compared to proper IBE schemes. Au et al. [2] proposed the notion of retrievability that from two secret keys of a user, one can compute the master secret key. The notion of retrievability is interesting but it is achieved only in the white box model. Chow [14] considers the notion of anonymous ciphertext indistinguishability, which requires that the PKG cannot learn the recipient identity from a ciphertext, thus hoping that the authority is not able to figure out which secret key to use to decrypt. This is an interesting notion as well, but only meaningful in an IBE system with an extremely large number of users; furthermore it does not protect against a PKG that targets a specific user and publishes the decryption algorithm (which is the main defense objective of A-IBE).

2 Generic Construction of A-IBE with Constant Size Ciphertext

Due to space limit, we refer the definitions and security models for A-IBE to Appendix A. In this section, we give a generic construction of A-IBE secure in the weak dishonest-PKG model from any IBE scheme using 1-out-of-2 OT, and it only has ciphertext size two times the underlying IBE scheme.

The intuition behind this generic transformation is that for each identity ID, there are exponentially many secret keys, each of which has a unique "fingerprint". Each user will select his key with a random "fingerprint" using an OT protocol. Given only an oracle access to a decryption box B implemented using one key, part of the fingerprint can be retrieved. When a decryption box is found, the recovered partial "fingerprint" is able to reveal the source of the box.

Specifically, 2ℓ identities $(ID||1||0, ID||1||1), \ldots, (ID||\ell||0, ID||\ell||1)$ are all considered as the same user with identity ID.[2] During **KeyGen**, for each pair of secret keys corresponding to identities $ID||i||0$, and $ID||i||1$, user randomly selects one of them using a 1-out-of-2 OT.[3] The "fingerprint" of the user selected key corresponds to the bit string of length ℓ he uses in the OT protocols. **Enc** randomly selects an index r, and simply encrypts the same message under both $ID||r||0, ID||r||1$, thus sender does not need to know the fingerprint of the user with ID. Note the user has one key per location, i.e., one key corresponding to the identity $ID||r||0$ or $ID||r||1$ for each r, thus he can decrypt. Also **Trace** can attempt to recover each bit of the fingerprint from a decryption box by feeding ciphertexts containing different messages for the location, i.e., for an index i, c_0, c_1 are fed, where $c_b = \mathbf{Enc}(ID||i||b, mpk, m_b)$. The semantic security of the underlying IBE suggests that the box will not distinguish these tracing ciphertexts from regular ciphertexts, and the answer m_b reveals the $i-$th bit of the user fingerprint. Whenever λ bits are recovered, and all of them equal to the corresponding bits in the user "fingerprint", the user will be accused, otherwise the PKG will be accused. Essentially, a malicious PKG can evade the tracing algorithm only if she guesses correctly λ random bits.

One may notice that a malicious user may put as few keys as possible, e.g., only one key corresponding to $ID||t||b_t$ for some t, into a pirate box B and thus for the other indices, there is no hope to recover the fingerprint bits. However, since B has to provide some minimum functionality, i.e., answering correctly with some noticeable probability δ, (formally, $\Pr[B(\mathbf{Enc}(m, ID, mpk)) = m] \geq \delta$), if we choose ℓ large enough (λ/δ through our probabilistic analysis), there must be at least λ keys contained in the pirate box to maintain the δ-correctness. In particular, we can argue that there exist at least λ indices, the box decrypts ciphertext generated using those indices, with probability at least δ/λ. Then as elaborated above, once a key is used, we can recover the corresponding bit.

2.1 Detailed Construction

We call this generic construction S-I, for an IBE scheme (`Setup, KeyGen, Enc, Dec`), the details of S-I are as follows:

[2] Doing above may reduce the original identity space, however, this problem can be easily addressed by extending the identity string $O(\log \ell)$ bits longer.

[3] Unlike ABE schemes, our generic construction does not have to provide collusion-resistance, as for each index, a user can obtain only one key.

- **Setup**(λ, δ): This algorithm inputs the security parameter λ and the correctness parameter δ, it runs the $\mathtt{Setup}$ algorithm of the underlying IBE and outputs master key pair (mpk, msk), and a parameter $\ell = \lambda/\delta$.
- **KeyGen** This is a protocol between PKG and a user A with identity ID,
  1. PKG generates 2ℓ secret keys $\{k_{i,b}\}_{i=1,\dots,\ell,b=0,1}$, using $\mathtt{KeyGen}$ of the underlying IBE, where $k_{i,b} = \mathtt{KeyGen}(msk, ID||i||b)$.
  2. User A randomly chooses a bit string $\bar{b} = b_1, \dots, b_\ell$ with length ℓ.
  3. A executes ℓ (1,2)-OT protocols with the PKG in parallel. In the i-th execution, A inputs b_i, PKG inputs $k_{i,0}, k_{i,1}$ and A receives k_{i,b_i}.

  The protocol ends with A outputting $sk_{ID} = \{sk_i = (b_i, k_{i,b_i})\}_{i=1,\dots,\ell}$.
- **Enc**(ID, mpk, m): To encrypt a message m for user A, the algorithm randomly chooses an index $r \in \{1, \dots, \ell\}$ and outputs ciphertext $C = (r, c_{r,0}, c_{r,1})$, where for $b \in \{0, 1\}$, $c_{r,b} = \mathtt{Enc}(ID||r||b, mpk, m)$.
- **Dec**(C, sk_{ID}): On input ciphertext C and the secret keys of user A, the decryption algorithm parses the ciphertext and runs the underlying IBE decryption algorithm, it returns $m = \mathtt{Dec}(c_{r,b_r}, sk_r)$.
- **Trace**$^B(ID, \delta, \{b_i\})$ This is a two stage protocol. In the first stage, the judge $\mathcal{J}$ interacts with user A[4] to get his secret string and verify its validity.
  1. A sends $\bar{b}$ and a pirate decryption box B to $\mathcal{J}$.
  2. $\mathcal{J}$ parses $\bar{b}$, and then randomly selects 2ℓ messages $\{r_{i,0}, r_{i,1}\}_{i=1,\dots,\ell}$, and asks A to decrypt one of the ciphertext $\{c_{i,0}, c_{i,1}\}$, where $c_{i,b} = \mathtt{Enc}(ID||i||b, mpk, r_{i,b})$ for $i = 1, \dots, \ell$. A decrypts $\{c_{i,b_i}\}$ and sends back $\{r'_{i,b_i}\}$, $\mathcal{J}$ then checks $r_{i,b_i} \overset{?}{=} r'_{i,b_i}$ for all $i \in \{1, \dots, \ell\}$.

  If not, $\mathcal{J}$ outputs "user"; otherwise, $\mathcal{J}$ runs the following algorithm:
  1. For each $i \in \{1, \dots, \ell\}$, $\mathcal{J}$ repeats the following N times (the exact number of N will be specified in the analysis) to define a bit s_i. In each run, $\mathcal{J}$ randomly selects m_0, m_1, and feeds B with $(i, c_{i,0}, c_{i,1})$, where $c_{i,b} = \mathtt{Enc}(ID||i||b, mpk, m_b)$ for $b = 0, 1$. $\mathcal{J}$ records a b for s_i if B returns m_b, otherwise, $\mathcal{J}$ records a $\perp$.
  2. After the repetitions for each i, $\mathcal{J}$ takes the majority of the non-$\perp$ records as the value for s_i; if all records are $\perp$, then s_i is undefined.
  3. Suppose $s_{i_1}, \dots, s_{i_t}$ are the defined bits. If $s_{i_j} = b_{i_j}$ for all $j \in \{1, \dots, t\}$ and $t \geq \lambda$, $\mathcal{J}$ returns "user"; otherwise, $\mathcal{J}$ returns "PKG".

Remark. Our tracing algorithm is conditioned on the fact that the box has a noticeable correctness δ for random messages, and the box is resettable.

A Note about Fully Black-Box Traceability. We can see from the tracing algorithm of S-I that given access to a decryption oracle, the PKG learns the bit string that the user chose to select the secret keys, thus further learns the chosen secret keys of the user. One possible remedy is to introduce a mechanism that only the judge can create a valid tracing ciphertext, i.e., regular ciphertext pair is augmented with a ZK proofs of the statement that "either they contain equal plaintexts or I am the judge". This prevents the PKG from learning any

[4] It can be easily made non-interactive if the user proves that he has the right keys.

information about the user fingerprint via access to a decryption oracle, but also at the same time enables the judge to trace. One downside of this mechanism is that the judge needs to keep some private state thus we will have to work on a slightly weaker model. Due to lack of space, we defer the details of achieving fully black-box recoverability in this model to the full version. We will focus on the other advanced properties, e.g., identity reuse, which is not known whether achievable before in the standard model of A-IBE in the rest of the paper.

2.2 Security Analysis

We will give intuitions about the security properties of S-I and for the proof, we mainly focused on the most involved part dealing with malicious users.

IND-ID-CPA Security. $ID||i||0, ID||i||1$ are considered two different identities and thus our generic construction S-I is simply a double encryption of a same message using two different identities. It follows easily that a double encryption is as secure as the underlying IBE.

Security in the Weak Dishonest-PKG Game. Note that the **Trace** algorithm does not outputs "PKG" only when the recovered string is composed of two parts: an all-$\perp$ part, and a bitstring which is at least λ bits long and matches the corresponding substring of the user secret string. All other cases, including an all-$\perp$ string is recovered, or any single bit recovered is different with the corresponding bit of the user "fingerprint", the PKG is accused.

The receiver security of the OT protocol executed in **KeyGen** guarantees that a malicious PKG can only guess each bit of the secret string, thus she can fools the **Trace** algorithm with probability negligibly close to $2^{-\lambda}$. Specifically, in the execution of the i-th OT protocol, the malicious PKG can not distinguish the transcript created by an user inputting a random bit r from the transcript created using the selected bit b_i. We can do a sequence of game changes and end up with a game that all OT transcripts are created using independently selected random bits $\bar{r} = r_1, \ldots, r_\ell$. In the last game, since $\bar{b} = b_1, \ldots, b_\ell$ are independent of the transcripts, we can let the malicious PKG output a box and the judge recovers a substring with length at least λ first, and then select $\bar{b}$. It follows easily that the corresponding substring of $\bar{b}$ matches the recovered substring of $\bar{r}$, with probability at most $2^{-\lambda}$.

Security in the Adaptive Dishonest-User Game. Our main observation that if the box is leaked by a user, the judge will always be able to accuse her, relies on the following reasons. First, since the user has only one key for each location, due to the semantic security of the underlying IBE (and the OT sender security), the user has to report to the judge honestly her secret string. Furthermore, the box B is not able to tell a tracing ciphertext (the pair of the ciphertext encrypting different messages) from a normal ciphertext, thus B will have δ-correctness during tracing. We will analyze that the box has to decrypt using the keys with probability δ/λ for at least λ indices to maintain such correctness. Again, for each index i, B can never succeed in decrypting

m_{1-b} if only $k_{i,b}$ is inside, thus for the indices it responds, it has to reveal the correct bits after enough repetitions.

Theorem 1. *(1). S-I is IND-ID-CPA secure if the underlying IBE scheme is IND-ID-CPA secure; (2). S-I is secure in the* `weak dishonest-PKG` *game if the underlying 1-out-of-2-OT protocol satisfies the receiver security; (3). S-I is secure in the adaptive* `dishonest-user` *game if the underlying IBE is IND-ID-CPA secure, and the 1-out-of-2-OT protocol satisfies the (simulatable) sender security.*

Proof. The security properties (1) and (2) follow easily from the explanation above, we will focus on property (3).

First, it is not hard to see that in the first phase of the **Trace** protocol, the user has to submit the same string she selected. This can be shown via a sequence of game changes. In the original game, the adversary $\mathcal{A}$ runs the OT protocols one by one for ℓ times (or in parallel), during the **KeyGen** protocol, and answers the decryption queries during the first phase of the **Trace** algorithm. In the modified ℓ games, the OT protocols are replaced with an oracle (one by one) that on inputting a bit, outputting the corresponding secret key. The indistinguishability of these game changes are ensured by the (simulatable) sender security of the OT protocol (see the composition lemma of Canetti [13]).

In the last game, during **KeyGen** $\mathcal{A}$ has only oracle access to the OT instances, which can be "controlled" by a simulator. Now suppose the adversary answers correctly for the decryption request $c_{i,1-b_i}$ at some index i with probability Δ_i, there exists a simulator $\mathcal{S}$ playing the role of PKG with $\mathcal{A}$ as a user, can break the IND-ID-CPA security of the underlying IBE. $\mathcal{S}$ can answer all the OT queries perfectly with the corresponding secret keys, (which can be asked to the IND-ID-CPA game challenger directly). $\mathcal{S}$ simply uses $ID||i||1-b_i$ as the challenge identity. $\mathcal{S}$ selects m_0, m_1 as the challenge message, and forwards the challenge ciphertexts to the adversary. If $\mathcal{A}$ answers m_b, $\mathcal{S}$ answers b, otherwise, a random bit. It is straightforward that $\mathcal{S}$ breaks the IND-ID-CPA security with advantage $\frac{\varDelta_i}{2}$ (which can be derived as follows: $\varDelta_i \cdot 1 + (1-\varDelta_i)\frac{1}{2} - \frac{1}{2}$).

Let $\delta_i = \Pr[B \text{ decrypts correctly} \mid i \text{ is selected}]$. We divide the indices $i \in \{1, \ldots, \ell\}$ in two sets, Bad and Good, we define $i \in \mathsf{Good}$ if and only if $\delta_i \geq \delta_0$, where $\delta_0 = \delta/\lambda$. Next, we lower bound $n = |\mathsf{Good}|$. If $n < \lambda$, then:

$$\Pr[B \text{ works correctly}] = \sum_{i=1}^{\ell} \Pr[B \text{ works correctly}|i \text{ is selected}] \ \Pr[i \text{ is selected}]$$

$$\leq [1 \cdot (\lambda - 1) + \delta_0 \cdot (\ell - n + 1)]\frac{1}{\ell} = \frac{\lambda - 1}{\ell} + \frac{\delta(\ell - n + 1)}{\ell\lambda} \leq \frac{\lambda - 1}{\ell} + \frac{\delta}{\lambda} = \frac{\lambda}{\ell} = \delta$$

thus, we can conclude that for at least λ indices, the box will answer correctly with probability at least δ/λ.

Next, similar to the analysis for the first stage of the protocol, we can show that the probability that B decrypts to the other message selected in the **Trace** algorithm (m_{1-b_i}, which is with high entropy) will be a negligible function. Following the standard Chernoff bound, we can see that if we run the **Trace** algorithm with the a number of $N = O(\delta_0^{-2}\log^2\lambda)$ repetitions, the correct value of b_i would form a majority of the non-$\perp$ records for s_i.

Summarizing the above facts, if a box B implemented using one key from the user and it has δ-correctness, there will be at least λ indices that the **Trace** algorithm recovers the correct bits, ($\perp$ for all other indices), it returns "user". □

3 Generic Construction of A-IBE Allowing Public Traceability and Identity Reuse

In this section, we consider how to add advanced properties of A-IBE generically, without influencing the ciphertext efficiency much. And for a general definition and security models capturing the advanced properties, we refer to Appendix A.

3.1 A General Framework Allowing Identity Re-use

As elaborated in the introduction, a user may accidentally lose his secret key, in all previous works, the user has to change a different identity to request a new key. Allowing identity re-use in such cases is highly desirable. The main difficulty for achieving id reuse lies in the fact that a malicious user can obtain multiple keys (for a same ID) by claiming to the PKG that she lost her key. Then she will implement a pirate box using one key and reveal a different key to the judge for the tracing algorithm, trying to frame the PKG.

Necessity of Public Traceability and Linear Size Tracing Key. To defend against the above attack, a correct tracing algorithm on inputting two keys requested using the same identity should not always output "PKG". It follows that the judge has to be able to identify a "lost" key using some public information, which in turn "implies" public traceability.

Note that in S-I, each user chooses a "fingerprint" $b_1 \ldots b_\ell$ when requesting a key. If the **Trace** algorithm is able to recover the whole "fingerprint" from the pirate box, and there is a public reference, e.g., a value $T = f(b_1 \ldots b_\ell)$ for a one way function f, then the judge can publicly check whether the pirate box is from the user or not. In particular, T is generated by the user during the key generation, and he proved in zero-knowledge that the bits of the pre-image of T are consistent with those used in the OT protocols. We will first revise S-I to enable the tracing algorithm to recover the whole fingerprint, and explain in detail in the next section about the one way function and the ZK proofs.

Before we go into technical details of constructions, we first argue that the public tracing key has to grow linear to the number of the identities. To see this, suppose there are N different identities, d_i is the binary random variable that denotes the judge output when seeing a key k_{ID_i} for identity ID_i, and T is the public tracing key. It is obvious that without the tracing key, each d_i is a uniformly random bit (and they are mutually independent), thus $H(d_1, \ldots, d_N) = N$; while given T, all $\{d_i\}$ will be determined, thus $H(d_1, \ldots, d_N|T) = 0$, from the chain rule, we can see $H(T) = H(d_1, \ldots, d_N, T) \geq H(d_1, \ldots, d_N) = N$. Thus the length of T grows linearly to the number of identities used in the system.

Recovering All Bits of Each User Fingerprint. As one may notice, the **Trace** algorithm of S-I can recover only λ bits, thus for the above public tracing strategy to work, we have to improve the construction of S-I so that one can publicly recover the user "fingerprint" perfectly. A simple observation is that if one can recover a larger fraction of bits, e.g., a linear fraction of ℓ, one may use an error correcting code to generate the fingerprint and recover the whole string by decoding a string having a linear fraction of correct bits. However, the probabilistic analysis of S-I will not hold if we set $n = |\mathsf{Good}|$ to be $O(\ell)$. We further observe that if we use slightly more indices for encryption, (splitting the message, and using the S-I encryption algorithm at each index for the shares), the pirate box has to contain more keys to maintain the δ-correctness. Through a careful analysis, if we use $t = 5\ln\frac{2}{\delta}$ pairs of identities for encryption, B has to include at least $\frac{4}{5}$ fraction of the keys to maintain δ-correctness. Interestingly, the secret key length of is reduced to $O(\log\frac{1}{\delta})$. We present here the modified generic construction, (named S-II) with only the difference with S-I. We will show how to augment S-II to allow id-reuse and analyze the security in the next sections.

- **Setup**(λ, δ): Same as S-I, except $\ell = O(\log\frac{1}{\delta})$, and it also generates an error correcting code ECC : $\{0,1\}^{\ell_0} \to \{0,1\}^{\ell}$, (e.g., [23].) which corrects at least $\frac{\ell}{5}$-bit errors.
- **KeyGen**: Same as S-I, except that the bitstring of user A is generated by first selecting a random bitstring $\bar{r}$ with length ℓ_0, then applying the ECC to $\bar{r}$ and produces $\bar{b} = b_1, \ldots, b_\ell$.
- **Enc**(ID, mpk, m, δ): To encrypt a message m for user A, the algorithm first randomly chooses a subset $S = \{s_1, \ldots, s_t\} \subset \{1, \ldots, \ell\}$ with size $t(\delta) = 5\ln\frac{2}{\delta}$. It then chooses $t-1$ random messages $m_2, \ldots, m_t$ and computes $m_1 = m - \sum_{i=2}^{t} m_i$ and uses the $\mathtt{Enc}$ algorithm of the underlying IBE to encrypt each m_i. The algorithm outputs ciphertext $C = \{(s_i, c_{i,0}, c_{i,1})\}_{i=1,\ldots,t}$, where for $b \in \{0,1\}$, $c_{i,b} = \mathtt{Enc}(ID||s_i||b, mpk, m_i)$.
- **Dec**(C, sk_{ID}): On input ciphertext C and the secret key of user A, the decryption algorithm parses the ciphertext and then runs the underlying IBE decryption algorithm, and it selects the secrect keys corresponding to s_i and returns $m = \sum_{i=1}^{t} m_i$, where $m_i = \mathtt{Dec}(sk_{s_i}, c_{i,b_i})$.
- **Trace**$^B(ID, \delta, \{b_i\})$ The first stage is the same as that of S-I except that the user submits $\bar{r}$ and the judge $\mathcal{J}$ applies the ECC to get $\bar{b}$ himself. If $\mathcal{J}$ does not output "user" in the first stage, it runs the following:
    1. For each $i \in \{1, \ldots, \ell\}$, $\mathcal{J}$ randomly selects a subset $S \subset \{1, \ldots, \ell\}$ of size t until $i \in S$, and let us denote $S = \{s_1, \ldots, s_t\}$ and $i = s_k$; $\mathcal{J}$ randomly samples m, m' and other $t-1$ messages $m_1, \ldots, m_{k-1}, m_{k+1}, \ldots, m_t$ uniformly, and he computes $m_{k,0} = m - \sum_{j \neq k} m_j, m_{k,1} = m' - \sum_{j \neq k} m_j$. For $j = 1, \ldots, t$, $\mathcal{J}$ feeds the box B with $\{(s_j, c_{j,0}, c_j^1)\}$, where for $j \neq k$, $c_{j,b} = \mathtt{Enc}(ID||s_j||b, m_j)$, and $c_{k,b}$ is encryption of $m_{k,b}$, i.e., $c_{k,b} = \mathtt{Enc}(ID||s_k||b, m_{k,b})$ for both $b = 0, 1$. $\mathcal{J}$ records a 0 for b_i if the box returns m, 1 if the box returns m' and $\perp$ otherwise.

2. After repeating the above N times (the exact number of N will be specified in the analysis), $\mathcal{J}$ takes the majority of the non-$\perp$ symbols in the records as the value for b_i. If b_i is not defined, let $b_i = 0$.
3. $\mathcal{J}$ runs the decoding algorithm of ECC on $\bar{b}$, and gets a bitstring $\bar{r}'$ or $\perp$. If $\bar{r} = \bar{r}'$, $\mathcal{J}$ returns "user", otherwise, it returns "PKG".

Allowing Identity Re-Use. Now with the above briefly explained intuition of public traceability, a user can use different secret string $\{b_1^k, \ldots, b_\ell^k\}$ to choose the $k-$th secret key. The PKG keeps different public tracing key for each string, and the judge can indeed differentiate among the keys of the same identity and the PKG as long as he can extract the "fingerprints" correctly. (For detailed construction, see Sect. 3.3). To provide some collision resilience to the generic construction S-II, we extend it further to keep a state st_{ID} for each identity, so that each secret key request for a same identity can actually correspond to different extended identities. In more detail, in S-II, an identity ID is represented using a group of identities $\{ID||i||b_i\}_{i=1,\ldots,\ell,b_i=0,1}$. With a state st_{ID} denoting the number of key requested for ID, the modified extended identities would be $\{ID||st_{ID}||i||b_i^k\}_{i=1,\ldots,\ell;b_i^k=0,1;k=st_{ID}}$.

For the k-th time the user requests a key using $b_1^k, \ldots, b_\ell^k$, the PKG adds a new public tracing key $T_k = f_k(b_1^k, \ldots, b_\ell^k)$ to the public directory, and also updates st_{ID} to be $k+1$.[5] The sender first figures out the state, then he can simply run **Enc** of S-II using $ID||st_{ID}$ as identity. The **Trace** algorithm runs the S-II tracing algorithm on all $ID||1, \ldots, ID||st_{ID}$, with a smaller correctness parameter δ/st_{ID}, and extracts fingerprints (potentially more than one). If all of the fingerprints match the corresponding public tracing keys (except the $st_{ID}-$th one), they are considered as lost keys then no one will be accused; If the one that the user is using (the st_{ID}-th key) matches $T_{st_{ID}}$, the user would be accused, otherwise the PKG will be accused.[6]

We can see that we use the underlying IBE as a black-box, thus this improved construction (named S-III) is still a general transformation from IBE to A-IBE.

3.2 Building Blocks for Public Traceability

OT Instantiation. We choose the Bellare-Micali OT [3] as an example, and construct efficent zero-knowledge proofs for the consistency. (In principle any OT is applicable if we do not insist on efficient ZK proofs). The sender S (the PKG in our setting) sets up the system parameters (including a prime q, group G_q with a random generator g, and a random value $C \in Z_q$). The receiver R(with input b) randomly chooses $PK_b = g^x$ and computes $PK_{1-b} = C/PK_b$, then R sends PK_0 to S; the sender computes $PK_1 = C/PK_0$ and encrypts the messages m_0, m_1 to be transmitted, using ElGamal encryption [17] with PK_0, PK_1 as

[5] A malicious PKG may put different public tracing keys, however this is trivially detectable by the user and proves to the judge.

[6] Note that if the recovered fingerprint corresponds to one of the lost keys, it is impossible to decide whether it is from the user or from someone else who gets the lost key, not erroneously accusing the PKG is the best possible security in this case.

public keys respectively, i.e., $\{(g^{r_b}, H(PK_b^{r_b}) \oplus m_b)\}_{b=0,1}$ are returned to R, where H is modeled as a random oracle. It is well-known that this OT protocol satisfies information theoretic receiver security, and simulatable sender security under the CDH assumption [27].

Public Tracing Key Generation. We first describe the one way function tailored for our A-IBE scheme. Suppose $\bar{g} = (g_{1,0}, g_{1,1}), \ldots, (g_{\ell,0}, g_{\ell,1}) \in G^{2\ell}$, for each i, $g_{i,0} \cdot g_{i,1} = C$ for a random group element C, and $\bar{b} = b_1 \ldots b_\ell \in \{0,1\}^\ell$, we define $f_{\bar{g}}(b_1 \ldots b_\ell) = \prod_{i=1}^{\ell} g_{i,b_i}$. We will show that $f_{\bar{g}}(\cdot)$ is one way. Let us first look at a related one way function, suppose $\tilde{g} = (g_1, \ldots, g_\ell) \in G^\ell$ and for $b_1 \ldots b_\ell \in \{0,1\}^\ell$, $\tilde{f}_{\tilde{g}}(b_1 \ldots b_\ell)$ is defined by $\prod_{i=1}^{\ell} g_i^{b_i}$. It is implicit that $\tilde{f}_{\tilde{g}}(\cdot)$ is one way in a couple of papers, e.g., in [9], $b_1 \ldots b_\ell$ is the secret key and $\tilde{g}, h = \tilde{f}_{\tilde{g}}(b_1 \ldots b_\ell)$ are the public keys for their circular secure encryption scheme. We will omit the proof of one-wayness for $\tilde{f}$, and we prove the one-wayness of our function f in the following lemma.

Lemma 1. *If there exists a PPT adversary $\mathcal{A}$ breaks the one way security of f with advantage δ, then there exists another PPT adversary $\mathcal{B}$ breaks the one way security of $\tilde{f}$ with advantage δ/ℓ.*

Proof. When $\mathcal{B}$ receives the public keys $\tilde{g} = g_1, \ldots, g_\ell$ from the $\tilde{f}$ challenger $\mathcal{C}$, $\mathcal{B}$ selects a random C and prepares $g_{1,0}, \ldots, g_{\ell,0}$ such that for each i, $g_{i,0} = g_i$, he also prepares $g_{1,1}, \ldots, g_{\ell,1}$ in a way that $g_{i,1} = C/g_{i,0}$ for all i. $\mathcal{B}$ sends $\mathcal{A}$ $C, \bar{g} = (g_{1,0}, g_{1,1}), \ldots, (g_{\ell,0}, g_{\ell,1})$ as public keys.

Once $\mathcal{B}$ receives the challenge $X = \tilde{f}_{\tilde{g}}(b_1 \ldots b_\ell)$ for some $b_1 \ldots b_\ell$, $\mathcal{B}$ selects a random $t \in \{1, \ldots, \ell\}$, computes $Y = C^{\ell-t} \cdot \prod_{i=1}^{\ell} g_i \cdot X^{-2}$ and sends Y to $\mathcal{A}$. $\mathcal{B}$ forwards the bit string $b'_1 \ldots b'_\ell$ returned by $\mathcal{A}$ as her answer to the challenger $\mathcal{C}$.

Note that if the bitstring $b_1 \ldots b_\ell$ has Hamming weight $\ell - t$, i.e., t of them are 0, then $Y = \prod_{i=1}^{\ell} g_{i,b_i}$. To see this, suppose $S = \{i | b_i = 0\}$, and $|S| = t$, $Y = C^{\ell-t} \cdot \prod_{i=1}^{\ell} g_i / \prod_{i=1}^{\ell} g_i^{2b_i} = \prod_{i \in S} g_i \cdot \prod_{i \notin S}(C/g_i)$. Thus with probability $1/\ell$, $\mathcal{B}$ guesses t correctly, and in turn, $\mathcal{B}$ produces a valid value of $f_{\bar{g}}(b_1 \ldots b_\ell)$. In this case under our assumption, $\mathcal{A}$ will invert correctly with probability δ. We can conclude that $\mathcal{B}$ breaks the one way security of $\tilde{f}$ with probability δ/ℓ. □

The public tracing key T will be $h = f_{\overline{PK}}(b_1 \ldots b_\ell)$, together with $\overline{PK}$ which are $\{(PK_{1,0}, PK_{1,1}) \ldots, (PK_{\ell,0}, PK_{\ell,1})\}$ used in the OT protocols.

Efficient Zero-Knowledge Proof for Consistency. Next, we provide an efficient zero-knowledge proof protocol for the consistency between the public tracing key and the bit string selected by the user in the OT protocol. Essentially, we need to prove that each bit of the pre-image of the public tracing key is used for selecting one secret key in each call of the OT protocol. For the public tracing key h, the user first commits $\{PK_{i,b_i}\}$ to be $\{c_i\}$, and proves in zero-knowledge for the following statements, $\exists g_1, \ldots, g_\ell$:

$$h = \prod_{i=1}^{\ell} g_i \wedge_{i=1}^{\ell} [c_i \,\text{opens to}\, g_i \wedge (g_i = PK_{i,0} \vee g_i = PK_{i,1})] \wedge \text{PoK for} \log_g h.$$

Before we describe the detailed ZK proofs, we first explain how we can prove a commitment opens to a value. We will use a homomorphic commitment scheme from the BBS encryption [7]. It has the public keys in the form of (g, u, v, w), where $u^x = v^y = w$, and x, y are private keys. The ciphertext (which is a commitment as well) for m is $\bar{C} = (C_1, C_2, C_3)$ where $C_1 = u^{r_1}, C_2 = v^{r_2}, C_3 = w^{r_1+r_2}m$. One can easily prove a BBS commitment $\bar{C}$ opens to a message m in zero-knowledge using the following $\Sigma-$protocol: the proof is in the form of (a_1, a_2, c, z_1, z_2), where $a_1 = C_1^{t_1}, a_2 = C_2^{t_2}$ are the first round messages sent by the prover, a random value c is returned by the verifier and $z_1 = t_1 + cx, z_2 = t_2+cy$ are calculated by the prover, The verifier checks $C_1^{z_1} \cdot C_2^{z_2} = a_1 \cdot a_2 \cdot (C_3/m)^c$.

Now we are ready to construct the efficient ZK proofs. (1). Prove the first clause, which is equivalent to prove $\prod_{i=1}^{\ell} c_i$ opens to h. (2). Prove c_i opens to either $PK_{i,0}$ or $PK_{i,1}$. This can be done easily using the OR proof [15] of the two $\Sigma-$protocol. More specifically, suppose $b_i = 1$, the proof is in the form of $(a_{1,0}, a_{2,0}, a_{1,1}, a_{2,1}, c, z_{1,0}, z_{2,0}, z_{1,1}, z_{2,1})$, where $(a_{1,0}, a_{2,0}, c_0, z_{1,0}, z_{2,0})$ is simulated and using $c_1 = c - c_0$, and generates the proof of $(a_{1,1}, a_{2,1}, c_1, z_{1,1}, z_{2,1})$. The verifier checks $C_1^{z_{1,0}+z_{1,1}} \cdot C_2^{z_{2,0}+z_{2,1}} = a_{1,0} \cdot a_{1,1} \cdot a_{2,0} \cdot a_{2,1} \cdot (C_3/m)^c$. (3) Repeat step (2) for each commitment c_i to do an "And" proof. (4) Do a regular proof of knowledge about the exponent of h using e.g., Schnorr proof [30].

All these $\Sigma-$protocols can be made zero-knowledge following the standard technique, e.g., let the verifier commits to the challenge value c first, and they can be made non-interactive by applying the FS heuristic [16].

Finally, let us check whether the soundness is enough for ensuring h is generated in the honest way, i.e. $h = \prod_{i=1}^{\ell} PK_{i,b_i}$. Suppose there is an adversary $\mathcal{A}$ convinces the verifier and uses one $PK_{i,1-b_i}$ when generating h. We can see that $\mathcal{A}$ can be separated into two independent parts $(\mathcal{A}_1, \mathcal{A}_2)$. $\mathcal{A}_1$ prepares $\{PK_{i,0}, PK_{i,1}\}$ and the corresponding exponents, and $\mathcal{A}_2$ finishes the ZK proofs. It follows that if we replace $\mathcal{A}_1$ with another algorithm $\mathcal{A}_1'$ which simply receives $\{PK_{i,0}, PK_{i,1}\}$ and the corresponding exponents from an oracle, the modified adversarial algorithm $\mathcal{A}' = (\mathcal{A}_1', \mathcal{A}_2)$ behaves identically as $\mathcal{A}$.

According to the special soundness of the proof of knowledge part, a simulator can run $\mathcal{A}'$ ($\mathcal{A}_2$ part) to extract $\log_g h = \sum_{j \neq i} \alpha_{j,b_j} + \alpha_{i,1-b_i}$, where $\alpha_{j,b} = \log_g PK_{j,b}$ for $j \in \{1, \ldots, \ell\}$ and $b = 0, 1$. As the simulator can "control" the oracle of $\mathcal{A}_1'$, and prepare $\{PK_j\}_{j \neq i}$ accordingly for $\mathcal{A}_1'$, thus he knows the exponents $\{\alpha_{j,b_j}\}$ and recovers $\alpha_{i,1-b_i}$ and further $\log_g C = \alpha_{i,b_i} + \alpha_{i,1-b_i}$ thus breaks the discrete log assumption, where C is the system parameter in the OT protocol. (for the case that more than one PK_{i,b_i} are used in generating h, a similar argument can be made to recover $\log_g C$).

3.3 Concrete Construction and Security Analysis

With the building blocks we developed above, we now describe the concrete algorithms of our generic A-IBE construction allowing public traceability and identity reuse (named S-III). We only describe the difference with S-II here.

- **Setup**(λ, δ): Same as S-II.

- **KeyGen**: For the k-th key requests from user A for an identity ID, the **KeyGen** protocol of S-II is run for identity $ID||k$, and user returns $sk_{ID,k}$. During the **KeyGen**, the OT described above [3] is utilized to transmit secret keys. Suppose $\overline{PK}_k = \{(PK^k_{i,0}, PK^k_{i,1})\}$ are the first round messages of the user. After the OT protocols are done, A sends the PKG his public tracing key $h_k = \prod_{i=1}^{\ell} PK^k_{i,b_i}$ and proves in zero-knowledge (we call this proof π_k) for the consistency using protocol described in Sect. 3.2. The PKG outputs a new public tracing key $T_k = (h_k, \overline{PK}_k)$, adds them to the list of public tracing keys T_{ID} for ID and updates the st_{ID} to be k. The PKG outputs (T_{ID}, st_{ID}) and the user outputs secret key $sk_{ID,k}$.
- **Enc**$(ID, mpk, m, st_{ID}, \delta)$: It runs the **Enc** of S-II with identity $ID||st_{ID}$.
- **Dec**$(C, sk_{ID,st_{ID}})$: It runs the **Dec** of S-II with identity $ID||st_{ID}$.
- **Trace**$^B(ID, \delta/st_{ID}, T_{ID})$: The first stage is the same as S-II using $ID||st_{ID}$. If the judge does not output "user", the following is run. The second stage of the **Trace** algorithm of S-II is repeated for all identities from $ID||1$ to $ID||st_{ID}$. For $ID||k$, the algorithm recovers a bitstring $b^k_1, \ldots, b^k_\ell$ or $\perp$, and it records a flag t_k for this run. For $k = 1, \ldots, st_{ID} - 1$, if the recovered string is $\perp$ or $f_{\overline{PK_k}}(b^k_1, \ldots, b^k_\ell){=}h_k$, where $h_k, \overline{PK_k}$ are from T_k, then $t_k = 0$; otherwise $t_k = 2$. For $k = st_{ID}$, if no string is extracted, $t_{st_{ID}} = 0$; if $f_{\overline{PK_{st_{ID}}}}(b^{st_{ID}}_1, \ldots, b^{st_{ID}}_\ell){=}h_{st_{ID}}$, then $t_{st_{ID}} = 1$; otherwise, $t_{st_{ID}} = 2$.
  The algorithm returns $\perp$ if for all $k = 1, \ldots, st_{ID}$, $t_k = 0$; it returns "user" if $t_{st_{ID}} = 1$; it returns "PKG", otherwise.

Remark that using δ/st_{ID} for tracing is necessary, as from our definition of $\delta-$correctness in this case is only for a random state (see Appendix A).

Security Analysis of S-III. Due to lack of space, we provide here only some high-level intuition for S-III, and mainly on the difference with S-I.

IND-ID-CPA Security. This is very similar to that of S-I, except that there are extra public tracing keys T_{ID}, while they are only related with the bit strings for selecting the keys, thus independent with the real secret keys. Also S-III uses multiple extended form of identities, but all of them can be seen as different identities of the underlying IBE scheme. The semantic security is not influenced.

Security in the Weak Dishonest-PKG Game. Note that a malicious PKG can evade the **Trace** algorithm only when the recovered string matches one of the fingerprints contained in the public tracing key. The difference with S-I is that the malicious PKG receives extra public tracing keys $\{T_i = (h_i, \overline{PK_i})\}$, and ZK proof transcripts $\{\pi_i\}$. If an adversary $\mathcal{A}$ (malicious PKG) is able to produce a pirate box which fools the **Trace** algorithm, it can be easily turned to an algorithm that breaks the OT receiver security or the one-wayness of f.

In more detail, we can argue the security via a sequence of game changes by first replacing each OT transcript with one generated using a random bit r. The indistinguishability can be guaranteed by the information theoretic receiver security of the Bellare-Micali OT. In the next game changes, the ZK proofs will be replaced with simulated transcripts, and the indistinguishability can be guaranteed by the zero-knowledge property of the proofs. Now in the last game, what the adversary sees are only simulated transcripts (OT and ZK proofs) which are independent with the actual fingerprints, there exists a simulator $\mathcal{S}$ who can use $\mathcal{A}$ to break the one-way security of f. In particular, $\mathcal{S}$ randomly picks an one way function instance, i.e., $\mathcal{S}$ embeds the public keys and a value h received from the one way security challenger and sets it to be the i-th public tracing key, and sends them (together with a simulated proof) to $\mathcal{A}$. Then from the pirate box outputted by $\mathcal{A}$, with probability $1/st_{ID}$, the recovered string is the pre-image of h, thus $\mathcal{S}$ breaks the one way security.

Security in the Adaptive Dishonest User Game. A malicious user may try to frame the PKG by outputting a box with recovered fingerprint not matching any of the public tracing keys for the target identity, and it is possible unless one of the following events happens: for at least one index i, the adversary $\mathcal{A}$, (1). learns the secret key of $ID||i||1-b_i$ during the OT protocol; (2). is able to decrypt ciphertext under $ID||i||1-b_i$ for which she does not have the secret key; (3). cheats in the ZK proof of consistency during **KeyGen**. We can similarly do a sequence of game changes that first replace the OT instance to be oracle, the indistinguishability is guaranteed by the simulatable sender security of OT. We then argue from a box, the tracing algorithm must extract one of the whole fingerprints of the keys. This is similar to the proof of Theorem 1, we will focused on the main difference about the probabilistic argument. We can see that if the sender splits the message into $t(\delta) = 5\ln\frac{2}{\delta}$ pieces, the user has to put at least $\frac{4}{5}$ fraction of keys for each state into the box B to ensure δ-correctness, and this fraction is enough for the ECC decoding to recover the whole original fingerprint. *The probabilistic argument.* Let $\delta_i = \Pr[B \text{ decrypts correctly} \mid i \in S]$. We divide the indices $i \in \{1,\ldots,\ell\}$ in two sets, Bad and Good, we define $i \in \mathsf{Good}$ if and only if $\delta_i \geq \delta_0$, where $\delta_0 = \delta/\ell^2$. In order to upper bound the size of Bad consider the following. Let D be the event of correct decryption,

$$\Pr[\mathsf{D}] = \Pr[\mathsf{D} \mid S\cap\mathsf{Bad} = \emptyset]\cdot\Pr[S\cap\mathsf{Bad} = \emptyset]+\Pr[\mathsf{D} \mid S\cap\mathsf{Bad} \neq \emptyset]\cdot\Pr[S\cap\mathsf{Bad} \neq \emptyset],$$

Regarding $\Pr[S \cap \mathsf{Bad} = \emptyset]$ observe that if $k = |\mathsf{Bad}|$, this probability is bounded by $p(k,t) = C^t_{\ell-k}/C^t_\ell = \prod_{i=0}^{t-1}(1-\frac{k}{\ell-i}) \leq (1-\frac{k}{\ell})^t$. From inequality $e^x \geq 1+x$, we can get $p(k,t) \leq e^{-kt\ell}$. Regarding $\Pr[\mathsf{D} \mid S \cap \mathsf{Bad} \neq \emptyset]$, note that it is bounded by $\sum_{i\in\mathsf{Bad}}\delta_i \leq \ell\delta_0 = \delta/\ell$ (This follows from the fact that $\Pr[F| \cup_{i=1}^n A_i] \leq \sum_{i=1}^n \Pr[F|A_i]$, for any event F, A_i). We can now derive the following, $\delta \leq \Pr[\mathsf{D}] \leq e^{-tk/\ell}+\delta/\ell$, from which we obtain the upper bound $k \leq \frac{\ell}{t}\cdot\ln(\delta-\delta/\ell)^{-1}$, since $\delta - \delta/\ell \geq \delta/2$, when we set $t = 5\ln(2\delta^{-1})$ into the above bound for k, and in this case $k \leq \ell/5$.

Now in the last game, the adversary has only oracle access to OT which can be controlled by the simulator if from the outputted box, the simulator recovers a different fingerprint, the simulator can break the one way security using the extractor as explained at the end of Sect. 3.2.

Public traceability is obvious, and identity reuse follows also straightforwardly as for each state, the identities are considered as independent "user", the above argument implicitly captures this property. We summarize the security properties of S-III in the following theorem:

Theorem 2. *(1). S-III is secure in the* `IND-ID-CPA` *model if the underlying IBE is IND-ID-CPA secure. (2). S-III is secure in the* `weak dishonest PKG` *game if the proof π is zero-knowledge, f is one way and the OT has receiver security. (3). S-III is secure in the* `adaptive dishonest user` *game, if the underlying IBE is IND-ID-CPA secure, the proof π is sound, and the CDH assumption holds.*

4 Conclusions and Open Problems

We presented a generic transformation from IBE to A-IBE, with ciphertext size to be only twice large as the underlying IBE. We further refine the generic construction, and for the first time achieve identity reuse. We believe that the efficient generic transformations with preferable advanced properties can be an important step towards a wider deployment of A-IBE thus may potentially stimulate the adoption of IBE schemes in practice.

There are still several interesting open problems relating to the authority accountability in IBE schemes. One is to consider efficient generic construction of A-IBE with fully blackbox traceability directly, the other is to do a systematic study about proactive deterring mechanisms for IBE schemes.

Acknowledgements.. We thank Hong-Sheng Zhou for the early discussions. We thank the anonymous reviewer to point out the simplification for S-I. The authors were supported by the ERC project CODAMODA.

A Preliminaries

1-out-of-2 Oblivious Transfer Protocol. Briefly speaking, a 1-out-of-2 OT protocol [27] is between a sender S and a receiver R. S has two messages (m_0, m_1) as input, and R chooses one of them according to a bit b. S should not know b, while R should not have any knowledge of m_{1-b}.

We only provide a half simulation type of definition (cf. [27]). For `sender security`, we make a comparison to the ideal implementation in which there is a trusted party receiving m_0, m_1 from S and b from R, and sends R the message m_b. We require that $\forall m_0, m_1$ and any efficient adversary $\mathcal{A}$ as the receiver, there is a simulator plays as the receiver in the ideal world that, the output distribution of the simulator and $\mathcal{A}$ are computationally indistinguishable. For `receiver security`, suppose t_0, t_1 represent the trascript sent by the receiver w.r.t input 0 and 1 respectively, we require that the sender can not distinguish the distribution of t_0 and t_1.

Accountable authority identity-based encryption. Here we provide a general definition for an A-IBE scheme, it is composed of the following algorithms:

- **Setup**(λ, δ) This algorithm takes the security parameter λ and the correctness parameter δ as input and outputs master key pair (mpk, msk) and the system parameters $t(\delta), \ell(\delta)$.
- **KeyGen** This is a stateful protocol between a user and the PKG in which the user has an identity ID and mpk, and the PKG has mpk, msk, ID as inputs respectively. It ends with the user outputting her secret key sk_{ID} or $\perp$ if the secret-keys are malformed, and the PKG output a tracing key T_{ID} and a current state st_{ID}.
- **Enc**(ID, mpk, m, st_{ID}) This algorithm inputs a receiver identity ID, master public key mpk, the message m and potentially a public state st_{ID}, and outputs the ciphertext C.
- **Dec**(C, sk_{ID}) This algorithm takes ciphertext C and user secret key sk_{ID} as input, and outputs the plaintext m.
- **Trace**$^B(ID, \delta, T_{ID})$. This algorithm inputs a pirate decryption box B for ID, correctness parameter δ and a tracing key T_{ID} as input, it outputs "user", "PKG" or "$\perp$".

Note that the algorithms can be stateless as usual if identity reuse is not required. When T_{ID} is public, then the A-IBE scheme has public traceability.

δ**-correctness** of a decryption device B, for regular A-IBE schemes, it is defined as $\Pr[B(C) = m : C = \mathbf{Enc}(ID, mpk, m)] \geq \delta$; while for A-IBE schemes allowing identity re-use, the box might contain a couple of keys for one identity corresponding to different states, we require that for a randomly selected state, it works with δ correctness, thus the δ-correctness in this case is defined as $\Pr[B(C) = m : C = \mathbf{Enc}(ID, mpk, m, i) \wedge i \leftarrow \{1, \ldots, st_{ID}\}] \geq \delta$. Note that according to the pigeonhole principle, there exists at least one state j, $\Pr[B(C) = m : C = \mathbf{Enc}(ID, mpk, m, j)] \geq \delta / st_{ID}$, and this is important for the tracing algorithm of S-III.

IND-ID-CPA security. This is similar to the standard semantic security for IBE schemes. Consider the following game between the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$:

- **Setup** $\mathcal{C}$ runs **Setup**, and sends $\mathcal{A}$ the system public key: mpk.
- **Phase 1** $\mathcal{A}$ runs the **KeyGen** protocol with the challenger for several distinct adaptively chosen identities $ID_1, .., ID_q$ and gets the decryption keys $sk_{ID_1}, .., sk_{ID_q}$.
- **Challenge** $\mathcal{A}$ submits two equal length messages m_0, m_1 and an identity ID that is not appearing in the queries of Phase 1. $\mathcal{C}$ flips a random coin b and encrypts m_b with ID. The ciphertext C is passed on to $\mathcal{A}$.
- **Phase 2** This is identical to Phase 1 and $\mathcal{A}$ is not allowed to query for ID.
- **Guess** The adversary outputs a guess b' of b.

The advantage of the adversary $\mathcal{A}$ is defined as $|\Pr[b' = b] - 1/2|$; we say an A-IBE is IND-ID-CPA secure if $\mathcal{A}$'s advantage is negligible.

Note that for A-IBE schemes with public traceability, the adversary also gets the public tracing key T.

Besides standard semantic security, for an A-IBE scheme, there are two additional security properties that have to be considered. The first is security against a malicious PKG. Any A-IBE scheme, should prevent the PKG from learning useful information which can help her to leak a decryption program B (we will also call it a decryption "box") on behalf of a certain identity and evade the tracing algorithm. The second is security against malicious users. In this perspective, a group of colluding users should not be able to make a working box B that frames the PKG. Depending of the form of B, one may consider various models for the tracing algorithm. Specifically, if the tracing algorithm only needs oracle access to B, we call it traceable in the black-box model. Common variants of the black-box model exist depending on whether the PKG is given access to the decryption oracle that corresponds to the secret key the user gets (called the "fully black-box model" if yes, and the "weak black-box model" otherwise).

Weak (Black-Box) Dishonest-PKG Game. Consider the following game between a PPT adversary $\mathcal{A}$ and a PPT challenger $\mathcal{C}$:

- **Setup**: The adversary acts as a malicious PKG, generates system public keys and sends $\mathcal{C}$ mpk. Also $\mathcal{A}$ specifies an identity ID.
- **KeyGen**: $\mathcal{C}$ and $\mathcal{A}$ then engage in the **KeyGen** protocols of A-IBE acting as a user and PKG respectively. In each run, they jointly generate a decryption key and a tracing key T_{ID} and state st_{ID} for the identity ID. If neither party aborts, then $\mathcal{C}$ gets a decryption key sk_{ID} for user ID as output.
- **Create Decryption Box**: The adversary outputs a decryption box B.

The adversary $\mathcal{A}$ wins the game if the following conditions hold true:

$$B \text{ has } \delta\text{-correctness} \wedge \mathbf{Trace}^{B}(ID, sk_{ID}) \neq \text{“PKG”}.$$

In a *full* dishonest-PKG game, $\mathcal{A}$ is also allowed to ask decryption queries. In other weaker (non-black-box) models, the tracing algorithm might have non-black-box access to the pirate box B.

Adaptive Dishonest-User Game. In this game, a set of malicious users collude to create a decoder box, trying to frame the PKG.

- **Setup** $\mathcal{C}$ runs the A-IBE **Setup** algorithm, and sends $\mathcal{A}$ mpk;
- **Secret Key Queries** The adversary runs the **KeyGen** protocols with $\mathcal{C}$, playing the role of different users and PKG respectively, for adaptively chosen identities $ID_1, .., ID_q$ for different times. $\mathcal{A}$ gets the corresponding secret keys $\{sk_{ID_1}\}, .., \{sk_{ID_q}\}$ and $\mathcal{C}$ outputs the corresponding tracing keys $T_{ID_1}, \ldots, T_{ID_q}$ and the states $st_{ID_1}, \ldots, st_{ID_q}$.
- **Create Decryption Box** The adversary outputs an identity ID together with a decryption box B for ID.

The adversary wins if the followings hold true:

$$B \text{ has } \delta\text{-correctness} \wedge \mathbf{Trace}^{B}(ID, sk_{ID}) = \text{“PKG”}.$$

Weaker model also exists, i.e., in the selective dishonest-user game, the adversary is required to declare the ID to be attacked at the beginning.

References

1. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003)
2. Au, M.H., Huang, Q., Liu, J.K., Susilo, W., Wong, D.S., Yang, G.: Traceable and retrievable identity-based encryption. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 94–110. Springer, Heidelberg (2008)
3. Bellare, M., Micali, S.: Non-interactive oblivious transfer and applications. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 547–557. Springer, Heidelberg (1989)
4. Bellare, M., Rogaway, P.: Random oracles are practical: Aa paradigm for designing efficient protocols. In: ACM Conference on Computer and Communications Security, pp. 62–73 (1993)
5. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles-. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
6. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)
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., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, p. 213. Springer, Heidelberg (2001)
9. Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-secure encryption from decision Diffie-Hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer, Heidelberg (2008)
10. Boneh, D., Naor, M.: Traitor tracing with constant size ciphertext. In: ACM Conference on Computer and Communications Security, pp. 501–510 (2008)
11. Boneh, D., Sahai, A., Waters, B.: Functional encryption: definitions and challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer, Heidelberg (2011)
12. Boyen, X., Martin, L.: Identity-Based Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems. RFC 5091 (Informational), December (2007)
13. Canetti, R.: Security and composition of multiparty cryptographic protocols. J. Cryptol. **13**(1), 143–202 (2000)
14. Chow, S.S.M.: Removing escrow from identity-based encryption. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 256–276. Springer, Heidelberg (2009)
15. Cramer, R., Damgåard, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (1994)
16. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
17. Gamal, T.E.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory **31**(4), 469–472 (1985)

18. Gentry, C.: Certificate-based encryption and the certificate revocation problem. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 272–293. Springer, Heidelberg (2003)
19. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
20. Goyal, V.: Reducing trust in the PKG in identity based cryptosystems. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 430–447. Springer, Heidelberg (2007)
21. Goyal, V., Lu, S., Sahai, A., Waters, B.: Black-box accountable authority identity-based encryption. In: ACM Conference on Computer and Communications Security, pp. 427–436 (2008)
22. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security, pp. 89–98 (2006)
23. Guruswami, V., Indyk, P.: Expander-based constructions of efficiently decodable codes. FOCS **2001**, 658–667 (2001)
24. Kiayias, A., Tang, Q.: How to keep a secret: leakage deterring public-key cryptosystems. In: ACM Conference on Computer and Communications Security, pp. 943–954 (2013)
25. Lai, J., Deng, R.H., Zhao, Y., Weng, J.: Accountable authority identity-based encryption with public traceability. In: Dawson, E. (ed.) CT-RSA 2013. LNCS, vol. 7779, pp. 326–342. Springer, Heidelberg (2013)
26. Libert, B., Vergnaud, D.: Towards black-box accountable authority ibe with short ciphertexts and private keys. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 235–255. Springer, Springer (2009)
27. Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: SODA, pp. 448–457 (2001)
28. Sahai, A., Seyalioglu, H.: Fully secure accountable-authority identity-based encryption. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 296–316. Springer, Heidelberg (2011)
29. Sahai, A., Waters, B.: Fuzzy Identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
30. Schnorr, C.-P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 239–252. Springer, Heidelberg (1990)
31. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
32. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
33. 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)
34. Yuen, T.H., Chow, S.S.M., Zhang, C., Yiu, S.-M.: Exponent-inversion signatures and ibe under static assumptions. IACR Cryptol. ePrint Arch. **2014**, 311 (2014)