Identity-Based Traitor Tracing with Short Private Key and Short Ciphertext

Fuchun Guo, Yi Mu, and Willy Susilo

Centre for Computer and Information Security Research
School of Computer Science and Software Engineering
University of Wollongong, Wollongong, Australia
{fuchun,ymu,wsusilo}@uow.edu.au

Abstract. Identity-based traitor tracing (IBTT) scheme can be utilized to identify a decryption key of any identity that is illegally used in an identity-based broadcast encryption scheme. In *PKC'07*, Abdalla *et al.* proposed the first IBTT construction with short private key. In *CCS'08*, Boneh and Naor proposed a public-key traitor tracing, which can be extended to IBTT with short ciphertext. With a further exploration, in this paper, we propose the first IBTT with short private key *and* short ciphertext. Private key and ciphertext are both order of $O(l_1 + l_2)$, where l_1 is the bit length of codeword of fingerprint codes and l_2 is the bit length of group element. To present our IBTT scheme, we introduce a new primitive called *identity-based set encryption* (IBSE), and then describe our IBTT scheme from IBSE and fingerprint codes based on the Boneh-Naor paradigm. Our IBSE scheme is provably secure in the random oracle model under the variant of q-BDHE assumption.

Key words: traitor tracing, identity-based, short private key, short ciphertext

1 Introduction

1.1 Traitor Tracing

The concept of traitor tracing was introduced by Chor, Fiat, and Naor in [13]. One of the applicable scenarios of traitor tracing is to provide copyright protection in a Pay-TV setting. A copyrighted TV program is encrypted using a secure encryption scheme, where only legitimate subscribers are assigned with a decryption key for decrypting the program. An obvious problem in this scenario is that a Pay-TV subscriber could sell its decryption key to non-subscribers so that they can receive the program illegally and can even produce pirate decoders. Traitor tracing was proposed to identify the traitors who violate the copyright restrictions. A traitor tracing scheme comprises an encryption key, a tracing key and n decryption keys, where n is the number of users. Each legitimate user (subscriber) is given a unique decryption key, and any of the decryption keys can decrypt the encrypted item. More importantly, the tracing key can trace at least one decryption key used to create pirate decoders. A traitor tracing is said to be t-collusion resistant if the tracing is still successful against t colluded users (traitors).

The concept of identity-based traitor tracing (IBTT) was introduced by Abdalla *et al.* [2]. IBTT provides the tracing capability for identity-based encryption, where the private key of each identity is possessed by a group user. The ID-based traitor tracing exhibits broader applications. To motivate this, let us consider a more complex Pay-TV scheme. Subscribers could subscribe to multiple channels, which are sold separately. Hence, if each channel requires a distinct encryption key, many keys will be required. There is also an implication of key expiry. If a decryption key is expired, the entire scheme must be reset and re-encryptions are required. The non-ID-based schemes are inapplicable to this scenario, while the IBTT scheme is desirable. In an IBTT scheme, the encryption key can be the channel name along with an expiry date. The Pay-TV dealer only needs to manage the master secret key of the IBTT scheme and can easily handle the key management and revocation.

1.2 State of the Art

IBTT constructions are built from identity-based encryptions and fingerprint codes. The first approach proposed by Abdalla *et al.* [2] is based on the identity-based encryption with wildcards (WIBE) [1] and fingerprint codes. This IBTT construction provides a short private key, consisting of one codeword and three group elements. The ciphertext has to be sufficiently long and it consists of $O(l_1)$ number of group elements, where l_1 is the bit length of codeword. The second approach introduced by Boneh and Naor [8] enables IBTT construction from any IBE and fingerprint codes. This generic construction is short in ciphertext consisting of one index and two constant-size ciphertexts of IBE. The private key has to be sufficiently long and consists of one codeword and $O(l_1)$ number of private keys of IBE.

The existing IBTT schemes can only offer *either* a short private key *or* a short ciphertext, *but not both.* Since long private key increases the hardware cost of secure storage and long ciphertext requires a big bandwidth in communiation, our goal is to achieve both short private key and short ciphertext. In this paper, we propose an IBTT scheme based on a new encryption approach and fingerprint codes. Our IBTT construction captures both features of short private key and short ciphertext.

Table 1. Comparison of identity-based traitor tracing. Here, l_1 denotes the bit length of codeword and l_2 denotes the bit length of group element.

IBTT Schemes	Private Key Size	Ciphertext Size
[2]	$O(l_1 + l_2)$	$O(l_1 l_2)$
[8]	$O(l_1 l_2)$	$O(l_1 + l_2)$
Ours	$O(l_1 + l_2)$	$O(l_1 + l_2)$

1.3 Our Contributions

We propose the first IBTT with short private key and short ciphertext. Intuitively, our IBTT scheme can be outlined as follows. Let n be the number bound of users for each identity, t be the collusion bound, and l_1 be the corresponding codeword length of fingerprint codes [35, 10]. Our t-collusion resistant IBTT scheme generates both private key and ciphertext of size $O(l_1 + l_2)$, where l_2 denotes the length of group element. Precisely, our private key consists of one codeword and two group elements; our ciphertext is composed of one index and two constant-size ciphertexts. Our IBTT scheme utilizes the fingerprint codes and it gives the black-box tracing capability [26]. It provides the same properties as other code-based traitor tracing schemes, where it is applicable for stateless pirate decoders and the tracing key is secret.

We construct our IBTT from fingerprint codes and a new cryptographic primitive: *identity-based set encryption* (IBSE). Roughly speaking, in an IBSE scheme, an aggregated private key of identities $\mathbb{ID} = \{ID_1, ID_2, \cdots, ID_L\}$ can decrypt all ciphertexts for any identity $ID \in \mathbb{ID}$

as long as the encryption for identity ID takes input an additional identity set $\mathbb{S}_{ID}$ satisfying $\mathbb{ID} \subseteq \mathbb{S}_{ID}$. For example, let $\mathbb{ID} = \{ID_1, ID_2\}$ and $\mathbb{S}_{ID} = \{ID_1, ID_2, ID_3, ID_4\}$. If a message is encrypted using ID_1 (or ID_2) and $\mathbb{S}_{ID}$, the private key of $\mathbb{ID}$ enables to decrypt the message. Our generic IBTT construction shows that the private key of IBTT is composed of one codeword of fingerprint codes and two private keys of IBSE. The ciphertext of IBTT consists of an index and two ciphertexts of IBSE. Therefore, the private key size and the ciphertext size of IBTT are heavily dependent on its original IBSE scheme. In the remainder of this paper, we focus on constructing a secure IBSE scheme with a short private key and a short ciphertext, where both sizes are constant independent of the cardinality of $\mathbb{ID}$ and $\mathbb{S}_{ID}$. The IBSE scheme instantiated in this paper is provably secure in random oracles based on the hardness of the variant of q-BDHE assumption [5, 7].

1.4 Related Work

Since its seminal introduction in [13], many schemes in developing traitor tracing have been produced. A summary of traitor tracing categories can be found in [9, 8, 3]. Notably, Kiayias and Yung [26] and other researchers [12, 16, 8, 3] introduced a black-box tracing scheme, where the tracing procedure is only allowed to have black-box access to pirate decoders. Naor and Pinkas [29] and others [28, 23, 20, 15, 11, 18] proposed a trace-and-revoke scheme, where decryption keys in pirate decoders can be traced and then revoked without affecting any other legitimate decoders. Pfitzmann [30] and other researchers [36, 25, 12] achieved public traceability in which the tracing key can be public. Kiayias and Yung [24] and others [27, 31, 34] explored stateful pirate decoders, which can keep the state between decryptions.

Since the seminal work of fingerprint codes introduced by Boneh and Shaw [10], many code-based traitor tracing schemes have been proposed [26, 32, 31, 34, 16, 8, 3]. These schemes exhibit black-box tracing capability, and the schemes in [8, 3] even offer constant-size ciphertext. The main drawback of code-based traitor tracing schemes is the large private key size, which is significantly dependent on the length of codewords. The imperfect decoders further increase the private key length. We refer the readers to [8, 3] for further discussions.

Traitor tracing schemes associated with short ciphertext have been studied in [26, 16, 9, 8, 3]. Some of them [26, 16] achieved a constant rate for long messages but not a constant size. Boneh, Sahai and Waters [9] proposed a scheme with a ciphertext size $O(\sqrt{n})$ and a constant-size private key, where n is the number of users. Using fingerprint codes, it is able to achieve constant-size ciphertext [8, 3], but the private-key size is large. To the best of our knowledge, there exists *no* traitor tracing schemes where both ciphertext and private key are short or have a constant size.

Identity-based traitor tracing was first introduced by Abdalla *et al.* [2]. They managed to achieve a short private key from the IBE scheme with wildcards [1, 37], where the private key is composed of one codeword and three group elements. However, the ciphertext is not constant and composed of $O(l_1)$ number of group elements for an l_1-bit codeword.

It seems not hard to construct identity-based traitor tracing schemes with short ciphertext by extending the code-based traitor tracing scheme [8, 3, 4] into code-based identity-based traitor tracing using an identity-based encryption. This type of construction, however, is not more efficient than code-based public key traitor tracing in terms of private-key size, which requires $O(l_1)$ number of group elements for an l_1-bit codeword.

A potential approach for reducing the private-key size of code-based IBTT could be by building the traitor tracing scheme from another variant of identity-based encryption scheme.

For example, we can replace an IBE scheme with a multi-identity and a single-key decryption scheme (MISKD) [21, 22], where many private keys of distinct identities can be aggregated into a single one. This single private key decrypts all ciphertexts for any identity mapped to this key. Unfortunately, the current MISKD schemes are accompanied with a linear-size ciphertext, which is determined by the aggregated number of private keys. It is a tradeoff between utilizing IBE scheme and MISKD scheme for IBTT construction. The IBE-based IBTT gives a long private key, while the MISKD-based IBTT gives a long ciphertext. We will present a detailed comparison of IBE, MISKD and our IBSE schemes in later sections.

2 Identity-Based Set Encryption and Identity-Based Traitor Tracing

In Appendix A and B we review the definition of fingerprint codes [35, 10] and identity-based traitor tracing (IBTT)[2]. Instead of directly proposing our IBTT, we first define the new primitive of identity-based set encryption (IBSE) and give a generic construction of IBTT from IBSE and fingerprint codes. Then in the rest of this paper we propose a concrete IBSE that enables the IBTT construction with short private key and short ciphertext.

2.1 Definition of Identity-Based Set Encryption

In identity-based set encryption (IBSE), messages are encrypted to a single recipient identity. This is the common feature among the encryption notions of IBE, MISKD and our IBSE.

In comparison with IBE, IBSE produces three differences as follows.

- The key generation algorithm of IBSE enables to compute a single private key for multi-identity $\mathbb{ID} = \{ID_1, ID_2, \cdots, ID_L\}$. Normally, this private key $d_{\mathbb{ID}}$ is shorter in length than the sum of all separated private keys from a traditional IBE.
- The encryption algorithm of IBSE requires the recipient's identity ID along with an identity set $\mathbb{S}_{ID}$, if the private key of recipient is $d_{\mathbb{ID}}$ for multi-identity $\mathbb{ID}$ including ID. The encryption algorithm allows to pick any identity set $\mathbb{S}_{ID}$ satisfying $\mathbb{ID} \subseteq \mathbb{S}_{ID}$.
- The decryption algorithm of IBSE requires the private key $d_{\mathbb{ID}}$ of $\mathbb{ID}$ along with the recipient's identity ID, the multi-identity $\mathbb{ID}$ and the identity set $\mathbb{S}_{ID}$. Successful decryption on a ciphertext for ID requires $ID \in \mathbb{ID}$ and $\mathbb{ID} \subseteq \mathbb{S}_{ID}$.

In comparison with MISKD [21, 22], IBSE requires an identity set $\mathbb{S}_{ID}$ satisfying $\mathbb{ID} \subseteq \mathbb{S}_{ID}$ in both encryption and decryption. IBSE can be seemed as a variant of MISKD by setting $\mathbb{S}_{ID}$ as the universe. We propose the IBSE notion as there exists more efficient IBSE construction compared to MISKD schemes in the litearature. Comparison is given in Table 2.

Table 2. Comparison of IBE, MISKD and IBSE.

Schemes	Key Generation	Encryption	Decryption	Decryption Condition
IBE	ID	ID	d_{ID}	–
MISKD	$\mathbb{ID}$	ID	$d_{\mathbb{ID}}, ID, \mathbb{ID}$	$ID \in \mathbb{ID}$
IBSE	$\mathbb{ID}$	ID,$\mathbb{S}_{ID}$	$d_{\mathbb{ID}}, ID, \mathbb{ID}, \mathbb{S}_{ID}$	$ID \in \mathbb{ID}$ & $\mathbb{ID} \subseteq \mathbb{S}_{ID}$

An IBSE scheme consists of four algorithms as follows.

$\mathbf{Setup}_S(N, \lambda)$. The setup algorithm takes as input N, the cardinality of identity set (i.e., $|\mathbb{S}_{ID}| = N$), and a security parameter λ, and returns a master public key MPK and a master secret key MSK.

$\mathbf{KGen}_S(\mathbb{ID}, MSK)$. The key generation algorithm takes as input identities $\mathbb{ID} = \{ID_1, ID_2, \cdots, ID_L\}$ with $L \leq N$ and the master secret key MSK, and returns a private key $d_{\mathbb{ID}}$ for $\{ID_1, ID_2, \cdots, ID_L\}$.

$\mathbf{Enc}_S(ID, \mathbb{S}_{ID}, M, MPK)$. The encryption algorithm takes as input an identity ID, the identity set $\mathbb{S}_{ID}$ containing N distinct identities (including ID) and the message M, and returns a ciphertext $C \leftarrow \mathsf{Enc}_S(ID, \mathbb{S}_{ID}, M, MPK)$.

$\mathbf{Dec}_S(C, d_{\mathbb{ID}}, ID, \mathbb{ID}, \mathbb{S}_{ID})$. The decryption algorithm takes as input the ciphertext C, the private key $d_{\mathbb{ID}}$, identity ID, identities $\mathbb{ID}$ and the identity set $\mathbb{S}_{ID}$. The algorithm returns a message M or $\perp$.

The correctness requires that for all (MPK, MSK), $ID, \mathbb{ID}, \mathbb{S}_{ID}$, and $d_{\mathbb{ID}}$ if $ID \in \mathbb{ID}$ and $\mathbb{ID} \subseteq \mathbb{S}_{ID}$, we have

$$\mathsf{Dec}_S\big(\mathit{Enc}_S(ID, \mathbb{S}_{ID}, M, MPK), d_{\mathbb{ID}}, ID, \mathbb{ID}, \mathbb{S}_{ID}\big) = M.$$

Security. The full security notion for IBSE scheme is similar to the IND-ID-CCA notion for IBE scheme. We name it IND-ID-Set-CCA, which is secure against chosen-ciphertext attacks. It is stated as follows:

Setup. The challenger runs the $\mathsf{Setup}_S(N, \lambda)$ algorithm to generate (MPK, MSK) and gives the adversary MPK.

Phase 1. The adversary makes private key queries and decryption queries in this phase.

- For a private key query on $\mathbb{ID}$ ($|\mathbb{ID}| \leq N$) from the adversary, the challenger runs the $\mathsf{KGen}_S(\mathbb{ID},\ MSK)$ algorithm and returning the private key $d_{\mathbb{ID}}$ to the adversary.
- For a decryption query on $(ID, \mathbb{S}_{ID}, \mathbb{ID}, C)$ from the adversary, the challenger runs the $\mathsf{KGen}_S(\mathbb{ID},\ MSK)$ algorithm to compute $d_{\mathbb{ID}}$, runs the decryption algorithm $\mathsf{Dec}_S(C, d_{\mathbb{ID}}, ID, \mathbb{ID}, \mathbb{S}_{ID})$, and returns the decryption result to the adversary.

Challenge. The adversary outputs $(ID^*, \mathbb{S}_{ID^*}, M_0, M_1)$ to be challenged, where $ID^* \in \mathbb{S}_{ID^*}$. This challenge identity must be different from other identities for private key query. The challenger responds by flipping a coin $c \in \{0, 1\}$, running the $\mathsf{Enc}_S(ID^*, \mathbb{S}_{ID^*}, M_c)$ algorithm, and returning the challenge ciphertext C^* to the adversary.

Phase 2. The adversary can make further private key queries and decryption queries in this phase, except a private key query on any $\mathbb{ID}$ satisfying $ID^* \in \mathbb{ID}$ and all decryption queries on C^* for ID^*.

Remark 1. In this security model, the adversary submits both ID^* and $\mathbb{S}^*_{ID}$ for challenge. Let $\mathbb{ID}$ be the identities queried in the security model. There are two different restriction definitions on $\mathbb{ID}_R$ with regard to $(ID^*, \mathbb{S}_{ID^*})$.

- ID^* cannot be one of identities in $\mathbb{ID}$.
- ID^* can be one of identities in $\mathbb{ID}$, but $\mathbb{ID} \nsubseteq \mathbb{S}^*_{ID}$.

We adopt the first definition for our IBSE scheme. Notice that the second definition is more stronger but it does not fit for those schemes with dynamic key aggregation. For example, let $d_{\mathbb{ID}_1}$ be the private key of $\mathbb{ID}_1 = \{ID_1, ID_2, ID_3, ID_4\}$, and $d_{\mathbb{ID}_2}$ be the private key of $\mathbb{ID}_2 = \{ID_1, ID_2\}$. If the private key $d_{\mathbb{ID}_3}$ of $\mathbb{ID}_3 = \{ID_3, ID_4\}$ is computable from $d_{\mathbb{ID}_1}$ and $d_{\mathbb{ID}_2}$, it is easy to verify the second definition does not work when $ID^* = ID_3$, $\mathbb{S}_{ID^*} = \mathbb{ID}_3$, and private key queries on $\mathbb{ID}_1$ and $\mathbb{ID}_2$ are allowed.

Guess. The adversary returns a guess $c' \in \{0,1\}$ wins the game if $c' = c$.

We let the number of private key query be q_1 and let the number of decryption query be q_2. We define the advantage of the adversary in the above game as $\mathsf{Adv}_S = \left|\Pr[c' = c] - \frac{1}{2}\right|$.

Definition 1. *An IBSE scheme is (T, q_1, q_2, ϵ)-secure against IND-ID-Set-CCA attacks if for all T-polynomial time adversaries who make q_1 private key queries at most and q_2 decryption queries at most, we have $\epsilon = \mathsf{Adv}_S$ is a negligible function of λ.*

Definition 2. *An IBSE scheme is $(T, q_1, 0, \epsilon)$-secure against IND-ID-Set-CPA attacks if for all T-polynomial time adversaries who make q_1 private key queries at most and 0 decryption queries at most, we have $\epsilon = \mathsf{Adv}_S$ is a negligible function of λ. In this case, we write (T, q_1, ϵ) the shorthand of $(T, q_1, 0, \epsilon)$.*

2.2 Generic Construction of IBTT

Let $(\mathsf{Setup_S}, \mathsf{KGen_S}, \mathsf{Enc_S}, \mathsf{Dec_S})$ be an identity-based set encryption scheme and $(\mathsf{Gen_{FC}}, \mathsf{Tra_{FC}})$ be a fingerprint code. Our identity-based traitor tracing scheme is described as follows:

Setup$_T(\lambda)$. Let $l_1 = l_1(\lambda)$ be the length of codeword in the fingerprint codes. The setup algorithm of IBTT scheme sets $N = l_1$, and runs the Setup_S algorithm two times to generate two key pairs (MPK_{S_0}, MSK_{S_0}) and (MPK_{S_1}, MSK_{S_1}). The master public key MPK and the master secret key MSK of the IBTT scheme are

$$MPK = \big(MPK_{S_0}, MPK_{S_1}\big), \quad MSK = \big(MSK_{S_0}, MSK_{S_1}\big).$$

KGen$_T(ID, MSK)$. The algorithm works as follows:

– Run the Gen_{FC} algorithm to generate (Γ_{ID}, tk_{ID}) for ID, where $\Gamma_{ID} = \{\overline{w}^{(1)}, \overline{w}^{(2)}, \cdots, \overline{w}^{(n)}\}$ and tk_{ID} is the tracing key. We require that the Gen_{FC} algorithm always computes the same (Γ_{ID}, tk_{ID}) for ID. This can be accomplished, for example, using a pseudo-random function.
– Let $\mathbb{ID}_{ID,i,0}$ and $\mathbb{ID}_{ID,i,1}$ be two identity sets defined as

$$\mathbb{ID}_{ID,i,0} = \big\{ID|k|0: \ k = 1, 2, \cdots, l_1, \ s.t. \ w_k^{(i)} = 0\big\}$$
$$\mathbb{ID}_{ID,i,1} = \big\{ID|k|1: \ k = 1, 2, \cdots, l_1, \ s.t. \ w_k^{(i)} = 1\big\}.$$

Compute the private keys

$$d_{\mathbb{ID}_{ID,i,0}} \leftarrow \mathsf{KGen}_S\big(\mathbb{ID}_{ID,i,0}, MSK_{S_0}\big), d_{\mathbb{ID}_{ID,i,1}} \leftarrow \mathsf{KGen}_S\big(\mathbb{ID}_{ID,i,1}, MSK_{S_1}\big).$$

The private key of ID for the ith user is $d_{ID,i} = (\overline{w}^{(i)}, \ d_{\mathbb{ID}_{ID,i,0}}, \ d_{\mathbb{ID}_{ID,i,1}})$.

Enc$_T(ID, M, MPK)$. The algorithms works as follows:

– Choose $j \in \{1, 2, \cdots, l_1\}$ at random.

– Let $\mathbb{S}_{ID,0}$ and $\mathbb{S}_{ID,1}$ be two identity sets defined as

$$\mathbb{S}_{ID,0} = \big\{ID|k|0 : \ k = 1, 2, \cdots, l_1\big\}, \ \mathbb{S}_{ID,1} = \big\{ID|k|1 : \ k = 1, 2, \cdots, l_1\big\}.$$

Compute the ciphertexts

$$C_{ID,0} \leftarrow \mathsf{Enc}_S\big(ID|j|0, \ \mathbb{S}_{ID,0}, \ M, \ MPK_{S_0}\big)$$
$$C_{ID,1} \leftarrow \mathsf{Enc}_S\big(ID|j|1, \ \mathbb{S}_{ID,1}, \ M, \ MPK_{S_1}\big).$$

The ciphertext is $C = \big(j, \ C_{ID,0}, \ C_{ID,1}\big)$.

Dec$_T(C, d_{ID,i})$. For the ith user with the private key $d_{ID,i}$, the decryption algorithm works as follows:

– If $w_j^{(i)} = 0$, compute $\mathbb{ID}_{ID,i,0}$ and $\mathbb{S}_{ID,0}$ from ID and $\overline{w}^{(i)}$, and output

$$\mathsf{Dec}_S\big(C_{ID,0}, \ d_{\mathbb{ID}_{ID,i,0}}, \ ID|j|0, \ \mathbb{ID}_{ID,i,0}, \ \mathbb{S}_{ID,0}\big);$$

otherwise, compute $\mathbb{ID}_{ID,i,1}$ and $\mathbb{S}_{ID,1}$ from ID and $\overline{w}^{(i)}$, and output

$$\mathsf{Dec}_S\big(C_{ID,1}, \ d_{\mathbb{ID}_{ID,i,1}}, \ ID|j|1, \ \mathbb{ID}_{ID,i,1}, \ \mathbb{S}_{ID,1}\big).$$

Trace$_T(\mathcal{PD}_{ID}, ID, MSK)$. The tracing algorithm works as follows:

– For $j = 1, 2, \cdots, l_1$, randomly choose a message $M_j \neq 0$ and does as follows:
  • Compute the ciphertexts

$$C_{ID,0} \leftarrow \mathsf{Enc}_S\big(ID|j|0, \mathbb{S}_{ID,0}, M_j, MPK_{S_0}\big)$$
$$C'_{ID,1} \leftarrow \mathsf{Enc}_S\big(ID|j|1, \mathbb{S}_{ID,1}, \ 0 \ , MPK_{S_1}\big).$$

  • Send $C_j = (j, C_{ID,0}, C'_{ID,1})$ to the pirate decryption box $\mathcal{PD}_{ID}$.
  • Let the return from $\mathcal{PD}_{ID}$ be M'_j. Define the bit w_j as

$$w_j = \begin{cases} 0 & \text{if } M'_j = M_j, \text{ and} \\ 1 & \text{otherwise.} \end{cases}$$

  Output the l_1-bit codeword $\overline{w}^* = w_1 w_2 \cdots w_{l_1}$.
– Compute the tracing key tk_{ID} for ID from Gen_{FC}. Run the $\mathsf{Tra}_{FC}(\overline{w}^*, tk_{ID})$ algorithm to output the set of traitors $\mathbb{T}_{ID} \subseteq \{1, 2, \cdots, n\}$.

Our IBTT scheme above is extended from the Boneh-Naor public-key traitor tracing scheme [8]. We do not change their paradigm, but replace the public-key encryption scheme with the IBSE scheme. The following theorem shows that our IBTT scheme is t-collusion resistant.

Theorem 1 *Given an identity-based set encryption scheme* $(\mathsf{Setup_S}, \mathsf{KGen_S}, \mathsf{Enc_S}, \mathsf{Dec_S})$*, which is IND-ID-Set-CPA secure and fingerprint codes* $(\mathsf{Gen_{FC}}, \mathsf{Tra_{FC}})$*, which is t-collusion resistant, our IBTT scheme is a t-collusion resistant identity-based traitor tracing scheme.*

Particularly, using the notation of Appendix A and B, for all $t > 0, n > t$, and all polynomial time adversaries attacking IBTT, there exist polynomial time adversaries attacking IBSE such that

$$\mathsf{Adv}_T^s \leq (2l_1) \cdot \mathsf{Adv}_S, \quad \mathsf{Adv}_T^c \leq l_1 \cdot \mathsf{Adv}_S + \mathsf{Adv}_{FC} + \frac{l_1}{|\mathcal{M}|},$$

where l_1 denotes the bit length of codeword and $\mathcal{M}$ denotes the message space.

The proof of Theorem 1 is very similar to the proof of Theorem 1 in [8]. For completeness, the sketch of the proof is provided below.

Proof. We bound the adversary's advantage Adv^s_T of game 1 in distinguishing the encrypted message. The adversary attacking IBTT can win the game 1 by breaking the indistinguishability of the ciphertext of IBSE. Notice that the IBTT scheme's ciphertext for ID^* will be generated by IBSE using potential identities of $\{ID^*|j|0, ID^*|j|1 : j = 1, 2, \cdots, l_1\}$, and each IBSE ciphertext will be broken with advantage Adv_S at most. Hence, the upper bound of breaking IBSE is $(2l_1) \cdot \mathsf{Adv}_S$.

We bound the adversary's advantage Adv^c_T of game 2 in creating a codeword that cannot be traced. Let $\mathbb{W}$ be the codewords corresponding to the set of private keys for ID^* in the adversary's possession. In game 2, if we can produce a codeword $\overline{w}^* \in F(\mathbb{W})$ based on the pirate decryption box $\mathcal{PD}_{ID^*}$ for ID^*, we immediately have $\mathsf{Adv}^c_T \leq \mathsf{Adv}_{FC}$. The remaining analysis is the probability analysis of $\overline{w}^* \notin F(\mathbb{W})$.

To analyze the probability of $\overline{w}^* \notin F(\mathbb{W})$, we consider a modified tracing algorithm that produces a codeword $\overline{q} = q_1 q_2 \cdots q_{l_1}$ as follows.

- For $j = 1, 2, \cdots, l_1$, randomly select a message $M_j \neq 0$ and do the following.
  - If all codewords in $\mathbb{W}$ have a 1 or 0 in position j, compute the ciphertexts

$$C_{ID^*,0} \leftarrow \mathsf{Enc}_S\big(ID^*|j|0, \mathbb{S}_{ID^*,0},\ 0, MPK_{S_0}\big)$$
$$C_{ID^*,1} \leftarrow \mathsf{Enc}_S\big(ID^*|j|1, \mathbb{S}_{ID^*,1},\ 0, MPK_{S_1}\big).$$

    Otherwise, compute the ciphertexts

$$C_{ID^*,0} \leftarrow \mathsf{Enc}_S\big(ID^*|j|0, \mathbb{S}_{ID^*,0},\ M_j, MPK_{S_0}\big)$$
$$C_{ID^*,1} \leftarrow \mathsf{Enc}_S\big(ID^*|j|1, \mathbb{S}_{ID^*,1},\ M_j, MPK_{S_1}\big).$$

  - Send the ciphertext $C_j = (j, C_{ID^*,0}, C_{ID^*,1})$ to the pirate decryption box $\mathcal{PD}_{ID^*}$.
  - Let the return from $\mathcal{PD}_{ID}$ be M'_j. If all codewords in $\mathbb{W}$ have a 1 in position j, define the bit q_j as

$$q_j = \begin{cases} 0 & \text{if } M'_j = M_j, \text{ and} \\ 1 & \text{otherwise.} \end{cases}$$

    Otherwise all codewords in $\mathbb{W}$ have a 0 in position j, define the bit q_j as

$$q_j = \begin{cases} 1 & \text{if } M'_j = M_j, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

- Set $\overline{q} = q_1 q_2 \cdots q_{l_1}$ as the traced codeword.

We argue that $\Pr[\overline{w}^* \notin F(\mathbb{W})] \leq l_1/|\mathcal{M}|$. The reason is provided as follows. Without loss of generality, we analyze the case of all codewords in $\mathbb{W}$ have a 1 in position j. According to the modified tracing algorithm, the pirate box will return 0 by following the decryption, or randomly pick a message M'_j instead of 0. The probability of $M'_j = M_j$ for a random M_j is $1/|\mathcal{M}|$ at most. Therefore, the modified tracing algorithm will output $q_i = 1$ except with $1/|\mathcal{M}|$ probability. We therefore obtain the upper bound probability of $l_1/|\mathcal{M}|$.

The tracing algorithm and the modified tracing algorithm are different in terms of the encryption of $C_{ID^*,0}$ or $C_{ID^*,1}$. It requires that the adversary cannot distinguish the modified tracing algorithm from the tracing algorithm and the encryption algorithm, especially when all codewords contain the same symbol (1 or 0) at the same position. If the symbol is 1 for j and the adversary does not have the private key of $ID^*|j|0$, this is equivalent to distinguishing the encryption $\mathsf{Enc}_S\big(ID^*|j|0, \mathbb{S}_{ID^*,0},\ 0, MPK_{S_0}\big)$ from $\mathsf{Enc}_S\big(ID^*|j|0, \mathbb{S}_{ID^*,0},\ M_j, MPK_{S_0}\big)$. The probability is bounded by Adv_S and the upper bound probability is $l_1 \cdot \mathsf{Adv}_S$.

The adversary wins the game 2 if $\overline{w}^* \in F(\mathbb{W})$ but $\overline{w}$ cannot be traced by the tracing algorithm Tra_{FC}, or $\overline{w}^* \notin F(\mathbb{W})$, or the adversary distinguishes the modified tracing algorithm. With the above separated analysis, we obtain the result of $\mathsf{Adv}_T^c \leq l_1 \cdot \mathsf{Adv}_S + \mathsf{Adv}_{FC} + \frac{l_1}{|\mathcal{M}|}$.

2.3 Comparison of IBTT Constructions

We give an IBTT construction from IBSE scheme in subsection 2.2 by following the Boneh-Naor paradigm. Notice that the IBSE scheme used to construct the IBTT scheme can be replaced with an IBE scheme (e.g. [6, 37, 19]) or a MISKD scheme (e.g.[21, 22]). The difference is the representation of private key and ciphertext. In the above IBTT scheme, each private key is associated with one codeword $\overline{w}^{(i)}$ and l_1 distinct identities $\{ID|k|w_k^{(i)} : k = 1, 2, \cdots, l_1\}$. And each ciphertext is composed of one index j and two ciphertexts of its original encryption scheme. If the encryption scheme is the MISKD or IBSE, according to our above construction, the private keys associated with l_1 identities can be aggregated into two private keys. Otherwise, it will produce l_1 private keys using the IBE scheme.

Let K_X be the private key and C_X be the ciphertext of X encryption scheme. Let X→ IBTT be the IBTT construction from X encryption scheme. We give a summary of private key length and ciphertext length in the following table.

Table 3. Comparison of IBTT Systems.

Constructions	Private Key Size	Ciphertext Size
IBE→ IBTT	$\lvert\overline{w}\rvert + l_1 \cdot \lvert K_{IBE}\rvert$	$\lvert j\rvert + 2 \cdot \lvert C_{IBE}\rvert$
MISKD→ IBTT	$\lvert\overline{w}\rvert + 2 \cdot \lvert K_{MISKD}\rvert$	$\lvert j\rvert + 2 \cdot \lvert C_{MISKD}\rvert$
IBSE→ IBTT	$\lvert\overline{w}\rvert + 2 \cdot \lvert K_{IBSE}\rvert$	$\lvert j\rvert + 2 \cdot \lvert C_{IBSE}\rvert$

The table exhibits that only MISKD→ IBTT or IBSE→ IBTT could capture both short ciphertext and short private key. However, the ciphertext of current MISKD schemes [21, 22] has a linear size, and we cannot achieve IBTT scheme with short private key and short ciphertext from the existing MISKD schemes. The remaining candidate is IBSE→ IBTT. In the next section, we show how to construct an IBSE scheme with short private key and short ciphertext, where both size are constant independent of $\mathbb{ID}$ and $\mathbb{S}_{ID}$. It will enable an IBTT construction with short private key and short ciphertext.

3 IBSE with Short Private key and Short Ciphertext

3.1 Definitions

Let $\mathcal{G}_B$ be a generator of bilinear groups. Taking as input a security parameter λ, it outputs bilinear groups $(g, p, \mathbb{G}, \mathbb{G}_T, e)$. Here, $\mathbb{G}, \mathbb{G}_T$ are two (multiplicative) cyclic groups of prime order p, g is a generator of $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is the bilinear map. The bilinear map e is a map with the following three properties:

– For all $u, v \in \mathbb{G}, a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$.
– $e(g, g)$ is a generator of $\mathbb{G}_T$.
– It is efficient to compute the bilinear map e.

The security of our scheme is based on the variant of q-bilinear Diffie-Hellman exponent assumption (q-BDHE), which has been used in [5, 7, 19]. We modify the BDHE assumption by using one group generator instead of two. The modified q-BDHE assumption is defined as follows, which can be justified in the generic group model by the result proved in [5].

Modified q-Bilinear Diffie-Hellman Exponent Problem:
Input: $g,\ g^{(a)},\ g^{(a^2)},\ \cdots,\ g^{(a^q)},\ g^{(a^{2q+2})},\ g^{(a^{2q+3})}, \cdots,\ g^{(a^{3q+1})} \in \mathbb{G}^{2q+1}$.
Output: $e(g, g)^{(a^{2q+1})}$.

Definition 3. *The (T, q, ϵ)-BDHE assumption holds in $\mathbb{G}$ if for all T-polynomial time adversaries, the advantage of solving the modified q-BDHE problem is ϵ at most, which is a negligible function of λ.*

3.2 Our Construction

In this construction, the private key structure for an individual identity is similar to the identity-based broadcast encryption in [33], and the encryption structure is modified from the identity-based broadcast encryption in [14] to achieve constant-size ciphertext. Our IBSE scheme will be provably secure in the random oracles under the modified q-BDHE assumption with the IND-ID-Set-CPA security. We can naturally extend it to CCA security using the technique due to Fujisaki-Okamoto [17] in the random oracle model.

Setup$_S(N, \lambda)$. The setup algorithm takes as input N and a security parameter λ. It first generates the bilinear groups $(g, p, \mathbb{G}, \mathbb{G}_T, e)$ by running $\mathcal{G}_B(\lambda)$. The algorithm randomly chooses $h \in \mathbb{G}$ and $\alpha \in \mathbb{Z}_p$. It picks two collision-resistant hash functions at random $H_1 : \{0, 1\}^* \to \mathbb{Z}_p$ and $H_2 : \{0, 1\}^* \to \{0, 1\}^{l_m}$. Here, l_m denotes the length of messages to be encrypted. The algorithm computes $h_1 = h^\alpha$ and $g_i = g^{(\alpha^i)}$ for $i = 1, 2, \cdots, N$. The master secret key MSK is α and the master public key MPK is

$$MPK = \Big(h, h_1, g, g_1, g_2, \cdots, g_N, p, \mathbb{G}, \mathbb{G}_T, e, H_1, H_2\Big).$$

KGen$_S(\mathbb{ID}, MSK)$. The key generation algorithm takes as input identities $\mathbb{ID} = \{ID_1, ID_2, \cdots, ID_L\}$ with $L \leq N$ and the master secret key α. It computes the private key $d_{\mathbb{ID}}$ as

$$d_{\mathbb{ID}} = h^{\frac{1}{\alpha - H_1(ID_1)} + \frac{1}{\alpha - H_1(ID_2)} + \cdots + \frac{1}{\alpha - H_1(ID_L)}} \in \mathbb{G}.$$

Enc$_S(ID, \mathbb{S}_{ID}, M, MPK)$. The encryption algorithm takes as input an identity ID, an identity set $\mathbb{S}_{ID} = \{ID'_1, ID'_2, \cdots, ID'_N\}$ $(ID \in \mathbb{S}_{ID})$, a message $M \in \{0, 1\}^{l_m}$ and the master public key MPK. Let

$$(\alpha - \mathbb{S}_{ID}) = \prod_{i=1}^{N} \Big(\alpha - H_1(ID'_i)\Big).$$

The algorithm picks a random $r \in \mathbb{Z}_p$ and outputs the ciphertext $C = (c_1, c_2, c_3) \in \mathbb{G}^2 \times \{0, 1\}^{l_m}$ as

$$c_1 = \Big(g^{(\alpha - \mathbb{S}_{ID})}\Big)^r, \quad c_2 = \Big(h^{\alpha - H_1(ID)}\Big)^r, \quad c_3 = H_2\Big(\ e\Big(g^{\frac{(\alpha - \mathbb{S}_{ID})}{\alpha - H_1(ID)}}, h\Big)^r\ \Big) \oplus M.$$

$\mathbf{Dec}_S(C, d_{\mathbb{ID}}, ID, \mathbb{ID}, \mathbb{S}_{ID})$. The decryption algorithm takes as input the ciphertext C, the private key $d_{\mathbb{ID}}$, the identity ID, the identities $\mathbb{ID}$ and the identity set $\mathbb{S}_{ID}$. If $ID \in \mathbb{ID}$ and $\mathbb{ID} \subseteq \mathbb{S}_{ID}$, we let the polynomial function $f(x)$ be

$$\begin{aligned} f(x) &= \big(x - \mathbb{S}_{ID}\big) \cdot \Big(\sum_{i=1}^{L} \frac{1}{x - H_1(ID_i)}\Big) \\ &= \frac{(x - \mathbb{S}_{ID})}{x - H_1(ID)} + \Big(x - H_1(ID)\Big) \cdot \Big(\sum_{i=0}^{N-2} f_i x^i\Big), \end{aligned}$$

where f_i is the coefficient of x^i. The algorithm outputs the message M by computing

$$c_3 \oplus H_2\Bigg(e(c_1, d_{\mathbb{ID}}) \cdot e\bigg(c_2, \prod_{i=1}^{N-2} g_i^{f_i} \cdot g^{f_0}\bigg)^{-1}\Bigg).$$

3.3 Correctness

In the encryption algorithm, $(\alpha - \mathbb{S}_{ID})$ and $\frac{(\alpha - \mathbb{S}_{ID})}{\alpha - H_1(ID)}$ are two polynomial functions in α, $g^{(\alpha - \mathbb{S}_{ID})}$ and $g^{\frac{(\alpha - \mathbb{S}_{ID})}{\alpha - H_1(ID)}}$ can be computed from the coefficients of polynomial functions and $r, g, g_1, g_2, \cdots, g_N$.

In the decryption algorithm, we have

$$e(c_1, d_{\mathbb{ID}}) = e\Bigg(\Big(g^{(\alpha - \mathbb{S}_{ID})}\Big)^r, h^{\sum_{i=1}^{L} \frac{1}{\alpha - H_1(ID_i)}}\Bigg) = e\big(g^{f(\alpha)}, h\big)^r$$

$$\begin{aligned} e\bigg(c_2, \prod_{i=1}^{N-2} g_i^{f_i} \cdot g^{f_0}\bigg)^{-1} &= e\Bigg(\Big(h^{\alpha - H_1(ID)}\Big)^r, g^{\sum_{i=0}^{N-2} f_i \alpha^i}\Bigg)^{-1} \\ &= e\Bigg(g^{-\big(\alpha - H_1(ID)\big) \cdot \big(\sum_{i=0}^{N-2} f_i \alpha^i\big)}, h\Bigg)^r \end{aligned}$$

$$e(c_1, d_{\mathbb{ID}}) \;\cdot\; e\bigg(c_2, \prod_{i=1}^{N-2} g_i^{f_i} \cdot g^{f_0}\bigg)^{-1} = e\Big(g^{\frac{(\alpha - \mathbb{S}_{ID})}{\alpha - H_1(ID)}}, h\Big)^r.$$

3.4 Comparison of IBE, MISKD and IBSE

We provide the comparison of IBE, MISKD and IBSE in Table 4 under the assumption that a user has to manage L distinct identities. The IBE scheme (e.g. [6, 37, 19]) has a very simple structure in encryption and decryption, but it cannot aggregate private keys into a short one. The MISKD scheme (e.g. [21, 22]) enables private key aggregation into a single one but the ciphertext size is not constant. In comparison with the MISKD, IBSE is able to aggregate private keys without expanding ciphertext size for decryption. Our IBSE scheme is short in both private key and ciphertext.

We realize the short private key and short ciphertext, at the price of complex encryption and decryption. An identity set $\mathbb{S}_{ID}$ such that $\mathbb{ID} \subseteq \mathbb{S}_{ID}$ must be known by the encryptor and the decryptor; otherwise, a ciphertext cannot be decrypted using the aggregated private key $d_{\mathbb{ID}}$. However, this provide a negligible implication on our IBTT construction since $\mathbb{S}_{ID}$ is computable from ID. Other applications using the IBSE primitive should be carefully checked.

Table 4. Comparison of IBE, MISKD and IBSE with L identities.

Schemes	Private Key Size	Ciphertext Size
IBE	$O(L)$	$O(1)$
MISKD	$O(1)$	$O(L)$
IBSE	$O(1)$	$O(1)$

3.5 Security Proof

Theorem 2 *Suppose the hash functions H_1, H_2 are two random oracles. Let q_{H_1} and q_{H_2} be the query number to the oracles H_1 and H_2 respectively. Let $q = \{q_{H_1},\ N\}_{max}$. Assuming the q-BDHE assumption is (T', ϵ')-hard, our IBSE scheme is (T, q_1, ϵ)-secure under IND-ID-Set-CPA attacks.*

$$T = T' - O(q_{H_1} t_e),\ q_1 \le q_{H_1},\ \epsilon = q_{H_1} q_{H_2} \epsilon',$$

where t_e denotes the average time of an exponentiation in $\mathbb{G}$.

Proof. Suppose there exists an adversary who can break the IBSE scheme with advantage (t, q_1, ϵ). We construct an algorithm $\mathcal{B}$ that solves the q-BDHE assumption with advantage (t', ϵ') at least. The algorithm $\mathcal{B}$ is given

$$\left(g, g^{(a)}, g^{(a^2)}, \cdots, g^{(a^q)}, g^{(a^{2q+2})}, g^{(a^{2q+3})}, \cdots, g^{(a^{3q+1})}\right),$$

and the aim of $\mathcal{B}$ is to output $e(g, g)^{(a^{2q+1})} \in \mathbb{G}_T$. The algorithm $\mathcal{B}$ interacts with the adversary $\mathcal{A}$ as below.

Setup. The algorithm $\mathcal{B}$ randomly chooses $\{I_1, I_2, \cdots, I_{q_{H_1}}, b\}$ from $\mathbb{Z}_p$, and picks a random $i^* \in \{1, 2, \cdots, q_{H_1}\}$. Let $F(x) \in \mathbb{Z}_p[x]$ be a $(q_{H_1} - 1)$-degree polynomial function as

$$F(x) = b \prod_{i=1, i \ne i^*}^{q_{H_1}} (x - I_i) = F_{q_{H_1}\text{-}1} x^{q_{H_1}\text{-}1} + \cdots + F_2 x^2 + F_1 x + F_0.$$

It sets $g_i = g^{(a^i)}$ for all $i = 1, 2, \cdots, N$ and computes $h = g^{F(a)}, h_1 = g^{aF(a)}$ from the challenge input and $F(x)$. The algorithm $\mathcal{B}$ forwards $MPK = (h, h_1, g, g_1, g_2, \cdots, g_N, p, \mathbb{G}, \mathbb{G}_T, e)$ except the two hash functions to the adversary and sets H_1, H_2 as random oracles.

Hash Queries. At any time, the adversary can query the random oracles H_1, H_2.

- For an identity query on ID to the random oracle H_1, the algorithm $\mathcal{B}$ maintains a list $\mathcal{L}_{H_1}$ and responds as follows. If there has been already a tuple (ID, I) in the list $\mathcal{L}_{H_1}$, the algorithm responds with $H_1(ID) = I$. Otherwise, let ID be the ith distinct query to H_1. The algorithm $\mathcal{B}$ responds by returning $H_1(ID) = I_i$ to the adversary, and adding (ID, I_i) to $\mathcal{L}_{H_1}$.
- For a random query on R to the random oracle H_2, the algorithm $\mathcal{B}$ maintains a list $\mathcal{L}_{H_2}$ and responds as follows. If R is not in the list, the algorithm responds by randomly choosing a different $Y \in \mathbb{Z}_p$, returning $H_2(R) = Y$ to the adversary, and adding (R, Y) to $\mathcal{L}_{H_2}$. Otherwise, there has been already a tuple (R, Y) in the list and the algorithm responds with $H_2(R) = Y$.

Phase 1. For a key query on $\mathbb{ID} = \{ID_1, ID_2, \cdots, ID_L\}$ from the adversary, the challenger responds as follows.

- Let the response for ID_i in the list $\mathcal{L}_{H_1}$ be (ID_i, I_i) for all $i = 1, 2, \cdots, L$. If $I_i = I_{i^*}$ holds for any $i \in \{1, 2, \cdots, L\}$, the algorithm aborts the simulation.
- When $I_i \neq I_{i^*}$ holds for all $i = 1, 2, \cdots, L$, we have that $H_1(ID_1), H_1(ID_2), \cdots, H_1(ID_L)$ are all the roots of $F(x)$. Then, we deduce that

$$F_{\mathbb{ID}}(x) = F(x) \cdot \Big(\frac{1}{x - H_1(ID_1)} + \frac{1}{x - H_1(ID_2)} + \cdots + \frac{1}{x - H_1(ID_L)}\Big)$$

is a $(q_{H_1} - 2)$-degree at most polynomial function. The algorithm $\mathcal{B}$ can compute

$$\begin{aligned} d_{\mathbb{ID}} &= h^{\frac{1}{\alpha - H_1(ID_1)} + \frac{1}{\alpha - H_1(ID_2)} + \cdots + \frac{1}{\alpha - H_1(ID_L)}} \\ &= g^{F(\alpha) \cdot \left(\frac{1}{\alpha - H_1(ID_1)} + \frac{1}{\alpha - H_1(ID_2)} + \cdots + \frac{1}{\alpha - H_1(ID_L)}\right)} = g^{F_{\mathbb{ID}}(a)} \end{aligned}$$

from $F_{\mathbb{ID}}(x)$ and $g, g^{(a)}, \cdots, g^{(a^q)}$, and $d_{\mathbb{ID}}$ is a valid private key of $\mathbb{ID}$.

Challenge. The adversary outputs $(ID^*, \mathbb{S}_{ID^*}, M_0, M_1)$ to be challenged. If the tuple (ID^*, I^*) in the list $\mathcal{L}_{H_1}$ satisfies $I^* \neq I_{i^*}$, abort; otherwise, the algorithm randomly chooses $c_3^* \in \{0,1\}^{l_m}$. Since $ID^* \in \mathbb{S}_{ID^*}$, we let

$$F'(x) = \frac{(x - \mathbb{S}_{ID^*})}{x - I^*}$$

be an $(N-1)$-degree polynomial function. The algorithm randomly chooses $r' \in \mathbb{Z}_p$ and computes the challenge ciphertext (c_1, c_2, c_3) by

$$c_1 = g^{r'\left(a^{2q+2} - I^{*2q+2}\right)F'(a)}, \quad c_2 = g^{r'\left(a^{2q+2} - I^{*2q+2}\right)F(a)}, \quad c_3 = c_3',$$

where both c_1 and c_2 are computable from $F'(x), F(x)$ and the challenge input.

Let the randomness r be

$$r = r' \cdot \frac{a^{2q+2} - I^{*2q+2}}{a - I^*},$$

which is also universally random in $\mathbb{Z}_p$. We have

$$\begin{aligned} g^{r'\left(a^{2q+2} - I^{*2q+2}\right)F'(a)} &= g^{r' \cdot \frac{\left(a^{2q+2} - I^{*2q+2}\right)}{\left(a - I^*\right)} \cdot \left(a - \mathbb{S}_{ID^*}\right)} = \left(g^{(\alpha - \mathbb{S}_{ID^*})}\right)^r, \\ g^{r'\left(a^{2q+2} - I^{*2q+2}\right)F(a)} &= g^{r' \frac{\left(a^{2q+2} - I^{*2q+2}\right)}{\left(a - I^*\right)} \cdot F(a)(a - I^*)} = \left(h^{\alpha - H_1(ID^*)}\right)^r, \end{aligned}$$

and the challenge ciphertext is equivalent to

$$\left(\left(g^{(\alpha - \mathbb{S}_{ID^*})}\right)^r, \ \left(h^{\alpha - H_1(ID^*)}\right)^r, \ c_3^*\right).$$

According to our setting, there must exist a hash query on $e\Big(g^{\frac{(\alpha - \mathbb{S}_{ID^*})}{\alpha - H_1(ID^*)}}, h\Big)^r$ to the random oracle H_2 in order to decrypt the message in the challenge ciphertext.

$$M = H_2\left(e\Big(g^{\frac{(\alpha - \mathbb{S}_{ID^*})}{\alpha - H_1(ID^*)}}, h\Big)^r\right) \oplus c_3^*.$$

Guess. The adversary returns a guess $c' \in \{0,1\}$ of c. Let $F''(x)$ be the $(2q+N+q_{H_1}-1)$-degree polynomial function

$$F''(x) = r' \cdot \frac{x^{2q+2} - I^{*2q+2}}{x - I^*} \cdot F'(x) \cdot F(x),$$

and F''_i be the coefficient of x^i in $F''(x)$. We have that $e\Big(g^{\frac{(\alpha-\mathbb{S}_{ID^*})}{\alpha-H_1(ID^*)}}, h\Big)^r = e(g,g)^{F''(a)}$.

It is easy to verify that F''_{2q+1} is equal to $r'F'(I^*)F(I^*)$ which is nonzero, and that $e(g,g)^{F''_i \cdot a^i}$ for all $i \neq 2q+1$ are computable from the challenge input. The algorithm $\mathcal{B}$ picks a random tuple (R, Y) from the list $\mathcal{L}_{H_2}$ and computes

$$\left(R \cdot \prod_{i=1, i\neq 2q+1}^{2q+N+q_{H_1}-1} e(g,g)^{-F''_i \cdot a^i}\right)^{\frac{1}{r'F'(I^*)F(I^*)}} = e(g,g)^{a^{2q+1}}$$

as the solution to the q-BDHE assumption.

We have completed the simulation proof of our IBSE scheme. To complete the proof, it remains to analyze the probability of successful simulation. We define the three types of events A_i, A^*, A_s:

- A_i is the event that the algorithm $\mathcal{B}$ can generate the ith private key query on ID_i. Let $\big(ID_i, I_i\big)$ be the response for ID_i in the list $\mathcal{L}_{H_1}$. This indicates that $I_i \neq I_{i^*}$ holds for ID_i.
- A^* is the event that the algorithm $\mathcal{B}$ does not abort in the challenge phase. Let $\big(ID^*, I^*\big)$ be the response for ID^* in the list $\mathcal{L}_{H_1}$. This indicates $I^* = I_{i^*}$.
- A_s is the event that what the algorithm $\mathcal{B}$ randomly picks from the list $\mathcal{L}_{H_2}$ is equal to $e\Big(g^{\frac{(\alpha-\mathbb{S}_{ID^*})}{\alpha-H_1(ID^*)}}, h\Big)^r$. Let q_{H_2} be the number of queries to the random oracle H_2. If the adversary ever made a query on $e\Big(g^{\frac{(\alpha-\mathbb{S}_{ID^*})}{\alpha-H_1(ID^*)}}, h\Big)^r$ to the random oracle, the probability of choosing a correct R_i is $1/q_{H_2}$.

According to the definition of security model, the adversary cannot query the private key of the challenge identity. With $1/q_{H_1}$ probability, the simulation does not abort till the guess phase. Therefore, if the adversary can break the IBSE scheme, the probability of successfully reducing the attack to solving the q-BDHE assumption is

$$\Pr\Big[\bigwedge_{i=1}^{q_1} A_i \bigwedge A^* \bigwedge A_s\Big] = \frac{1}{q_{H_1} q_{H_2}}.$$

Hence, if the adversary can break the scheme with probability ϵ, we can reduce the proof to solve the q-BDHE assumption with probability $\epsilon/(q_{H_1} q_{H_2})$.

The time complexity of our simulation is mainly dominated by the private key generation, where each private key computation takes $O(q_{H_1})$ exponentiations. The above analysis yields the theorem and we complete the proof. □

4 IBTT with Short Private Key and Short Ciphertext

In Section 2, we gave a generic IBTT construction from IBSE and fingerprint codes. In Section 3, we presented our IBSE scheme with short private key and short ciphertext. Putting our concrete

IBSE scheme into the generic IBTT construction, we yield an identity-based traitor tracing with short private key and short ciphertext.

The private key of our IBTT scheme is $d_{ID,i} = \big(\overline{w}^{(i)},\ d_{\mathbb{ID}_{ID,i,0}},\ d_{\mathbb{ID}_{ID,i,1}}\big)$, where $\overline{w}^{(i)}$ is the l-bit length of codeword, and $d_{\mathbb{ID}_{ID,i,0}}, d_{\mathbb{ID}_{ID,i,1}}$ are private keys of an IBSE scheme. We have $d_{\mathbb{ID}_{ID,i,0}}, d_{\mathbb{ID}_{ID,i,1}} \in \mathbb{G}$ from our IBSE scheme, and therefore our private key is short and composed of one codeword and two group elements.

The ciphertext of our IBTT scheme is denoted by $C = \big(j,\ C_{ID,0},\ C_{ID,1}\big)$, where j is the index from $[1, l]$, and $C_{ID,0}, C_{ID,1}$ are ciphertexts of an IBSE scheme. We have $C_{ID,0}, C_{ID,1} \in \mathbb{G}^2 \times \{0,1\}^{l_m}$ from our IBSE scheme, and therefore our ciphertext is short composed of one index, four group elements and two encrypted messages. The hybrid encryption technique will further reduce the two encrypted long messages into two encrypted short-random keys and one encrypted long message with the short-random key.

Computational Efficiency. We note that our IBTT scheme gives a tradeoff in private key size and computational efficiency. Our encryption/decryption requires to perform linear number of exponentiations, while the generic construction [8] only fulfils constant-number exponentiations for the same task. This tradeoff seems hard to be solved especially for decryption. This is because the decryption on a ciphertext for an identity with a private key of multi-identity must produce redundancy. It requires additional computations to remove them for decryption. Nevertheless, it is still interesting to explore more efficient IBTT schemes with short private key and short ciphertext.

Imperfect Decoders. The above traitor tracing assumes that the adversary produces a perfect pirate decoder that is able to decrypt all well-formed ciphertexts. Boneh and Naor also considered imperfect pirate decoders in their work. The countermeasure is by utilizing a powerful fingerprint code, which has to increase the length of codewords. Fortunately, we are able to use their fingerprint codes to construct our IBTT scheme against imperfect decoders. As the private key of IBSE is constant, the private key of our IBTT scheme only increases the length of codeword. The private key is still short. We observe that another solution for imperfect decoders is given in [3]. It requires a shorter codeword but a longer ciphertext compared to [8]. We refer the reader to [3] for the detail.

5 Conclusion

We introduced the first identity-based traitor tracing with short private key and short ciphertext. The private key consists of one codeword and two group elements; the ciphertext is composed of one index and two constant-size ciphertexts. It saves both secure storage and bandwidth for IBTT applications. We also introduced the new primitive of identity-based set encryption for multi-identity scenarios. Our proposed IBSE scheme is short in both private key and ciphertext, and is provably secure in the random oracles under the q-BSDH assumption.

Acknowledgement. We would like to thank the anonymous reviewers for their helpful comments and suggestions. This work has been supported by ARC Discovery Grant DP110101951.

References

1. Abdalla, M., Catalano, D., Dent, A.W., Malone-Lee, J., Neven, G., Smart, N.P.: Identity-based encryption gone wild. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 300–311. Springer, Heidelberg (2006)

2. Abdalla, M., Dent, A.W., Malone-Lee, J., Neven, G., Phan, D.H., Smart, N.P.: Identity-based traitor tracing. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 361–376. Springer, Heidelberg (2007)
3. Billet, O., Phan, D.H.: Efficient traitor tracing from collusion secure codes. In: Safavi-Naini, R. (ed.) ICITS 2008. LNCS, vol. 5155, pp. 171–182. Springer, Heidelberg (2008)
4. Billet, O., Phan, D.H.: Traitors collaborating in public: Pirates 2.0. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 189–205. Springer, Heidelberg (2009)
5. Boneh, D., Boyen, X., Goh, E.J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
6. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
7. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
8. Boneh, D., Naor, M.: Traitor tracing with constant size ciphertext. In: Ning, P., Syverson, P.F., Jha, S. (eds.) ACM CCS 2008. pp. 501–510. ACM (2008)
9. Boneh, D., Sahai, A., Waters, B.: Fully collusion resistant traitor tracing with short ciphertexts and private keys. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 573–592. Springer, Heidelberg (2006)
10. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE Transactions on Information Theory 44(5), 1897–1905 (1998)
11. Boneh, D., Waters, B.: A fully collusion resistant broadcast, trace, and revoke system. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) ACM CCS 2006. pp. 211–220. ACM (2006)
12. Chabanne, H., Phan, D.H., Pointcheval, D.: Public traceability in traitor tracing schemes. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 542–558. Springer, Heidelberg (2005)
13. Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: Desmedt, Y. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 257–270. Springer, Heidelberg (1994)
14. Delerablée, C.: Identity-based broadcast encryption with constant size ciphertexts and private keys. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 200–215. Springer, Heidelberg (2007)
15. Dodis, Y., Fazio, N.: Public key trace and revoke scheme secure against adaptive chosen ciphertext attack. In: Desmedt, Y. (ed.) PKC 2003. LNCS, vol. 2567, pp. 100–115. Springer, Heidelberg (2003)
16. Fazio, N., Nicolosi, A., Phan, D.H.: Traitor tracing with optimal transmission rate. In: Garay, J.A., Lenstra, A.K., Mambo, M., Peralta, R. (eds.) ISC 2007. LNCS, vol. 4779, pp. 71–88. Springer, Heidelberg (2007)
17. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999)
18. Garg, S., Kumarasubramanian, A., Sahai, A., Waters, B.: Building efficient fully collusion-resilient traitor tracing and revocation schemes. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM CCS 2010. pp. 121–130. ACM (2010)
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. Goodrich, M.T., Sun, J.Z., Tamassia, R.: Efficient tree-based revocation in groups of low-state devices. In: Franklin, M.K. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 511–527. Springer, Heidelberg (2004)
21. Guo, F., Mu, Y., Chen, Z.: Identity-based encryption: How to decrypt multiple ciphertexts using a single decryption key. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 392–406. Springer, Heidelberg (2007)
22. Guo, F., Mu, Y., Chen, Z., Xu, L.: Multi-identity single-key decryption without random oracles. In: Pei, D., Yung, M., Lin, D., Wu, C. (eds.) Inscrypt 2007. LNCS, vol. 4990, pp. 384–398. Springer, Heidelberg (2007)
23. Halevy, D., Shamir, A.: The lsd broadcast encryption scheme. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 47–60. Springer, Heidelberg (2002)
24. Kiayias, A., Yung, M.: On crafty pirates and foxy tracers. In: Sander, T. (ed.) ACM DRM 2001. LNCS, vol. 2320, pp. 22–39. Springer, Heidelberg (2001)
25. Kiayias, A., Yung, M.: Breaking and repairing asymmetric public-key traitor tracing. In: Feigenbaum, J. (ed.) ACM DRM 2002. LNCS, vol. 2696, pp. 32–50. Springer, Heidelberg (2002)
26. Kiayias, A., Yung, M.: Traitor tracing with constant transmission rate. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 450–465. Springer, Heidelberg (2002)
27. Matsushita, T., Imai, H.: A public-key black-box traitor tracing scheme with sublinear ciphertext size against self-defensive pirates. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 260–275. Springer, Heidelberg (2004)
28. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)

29. Naor, M., Pinkas, B.: Efficient trace and revoke schemes. In: Frankel, Y. (ed.) FC 2000. LNCS, vol. 1962, pp. 1–20. Springer, Heidelberg (2000)
30. Pfitzmann, B.: Trials of traced traitors. In: Anderson, R.J. (ed.) Information Hiding 1996. LNCS, vol. 1174, pp. 49–64. Springer, Heidelberg (1996)
31. Phan, D.H.: Traitor tracing for stateful pirate decoders with constant ciphertext rate. In: Nguyen, P.Q. (ed.) VIETCRYPT 2006. LNCS, vol. 4341, pp. 354–365. Springer, Heidelberg (2006)
32. Phan, D.H., Safavi-Naini, R., Tonien, D.: Generic construction of hybrid public key traitor tracing with full-public-traceability. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 264–275. Springer, Heidelberg (2006)
33. Sakai, R., Furukawa, J.: Identity-based broadcast encryption. IACR Cryptology ePrint Archive 2007, 217 (2007)
34. Sirvent, T.: Traitor tracing scheme with constant ciphertext rate against powerful pirates. Tech. rep., http://eprint.iacr.org/2006/383.pdf (2006)
35. Tardos, G.: Optimal probabilistic fingerprint codes. In: STOC 2003. pp. 116–125. ACM (2003)
36. Watanabe, Y., Hanaoka, G., Imai, H.: Efficient asymmetric public-key traitor tracing without trusted agents. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 392–407. Springer, Heidelberg (2001)
37. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)

A Definition of Fingerprint Codes

The fingerprint codes [8] are defined as follows.

- Let $\overline{w} \in \{0,1\}^l$ be an l_1-bit codeword. We write $\overline{w} = w_1 w_2 \cdots w_{l_1}$ and assume w_i is the ith bit of $\overline{w}$.
- Let $\mathbb{W} = \{\overline{w}^{(1)}, \overline{w}^{(2)}, \cdots, \overline{w}^{(t)}\}$ be a set containing t codewords in $\{0,1\}^{l_1}$. We say that a codeword $\overline{w} = w_1 w_2 \cdots w_{l_1}$ is feasible for the set $\mathbb{W}$, if for all $i = 1, 2, \cdots, l_1$ there exists a $j \in \{1, 2 \cdots, t\}$ such that the ith bit of $\overline{w}^{(j)}$, denoted by $w_i^{(j)}$, is equal to w_i.
- Let $F(\mathbb{W})$ be a feasible set of $\mathbb{W}$, if it includes all codewords that are feasible for $\mathbb{W}$.

A fingerprint code consists of two algorithms defined as follows.

Gen$_{FC}(n, t, \lambda)$. On input the number of codewords n, the collusion bound t and a security parameter λ, the generation algorithm outputs a set Γ containing n codewords $\{\overline{w}^{(1)}, \overline{w}^{(2)}, \cdots, \overline{w}^{(n)}\}$ in $\{0,1\}^{l_1}$ with length $l_1 = l_1(n, t, \lambda)$ and a tracing key tk.
Tra$_{FC}(\overline{w}^*, tk)$. On input a codeword $\overline{w}^* \in \{0,1\}^{l_1}$ and the tracing key tk, the tracing algorithm outputs a subset of $\{1, 2, \cdots, n\}$. Informally, let $\mathbb{W}$ be a subset of Γ, if $\overline{w}^* \in F(\mathbb{W})$, we have that the output of the tracing algorithm is a subset of $\mathbb{W}$.

The security definition of a fingerprint code from a game is stated as follows:
Setup. The challenger runs the $\mathsf{Gen}_{FC}(n, t, \lambda)$ algorithm to generate $\Gamma = \{\overline{w}_1, \overline{w}_2, \cdots, \overline{w}_n\}$ and the tracing key tk.
Query. For the query on the indices $\mathbb{I} \subseteq \{1, 2, \cdots, n\}$ with $|\mathbb{I}| \leq t$ from the adversary, the challenger responds by returning the codewords $\mathbb{W} = \{\overline{w}_i\}_{i \in \mathbb{I}}$ to the adversary.
Challenge. The adversary outputs a codeword $\overline{w}^* \in F(\mathbb{W})$ to be challenged.
Trace. The challenger runs the $\mathsf{Tra}_{FC}(\overline{w}^*, tk)$ algorithm and outputs a set $\mathbb{T} \subseteq \{1, 2, \cdots, n\}$. The adversary wins the game if $\mathbb{T}$ is empty, or not a subset of $\mathbb{I}$.

We define the advantage of the adversary in the above game as Adv_{FC}.

Definition 4. *A fingerprint code is t-collusion resistant if for all adversaries, all n, t satisfying $n \geq t$, all $\mathbb{I}$ satisfying $\mathbb{I} \subseteq \{1, 2, \cdots, n\}$ and $|\mathbb{I}| \leq t$, we have that Adv_{FC} is a negligible function of λ.*

B Identity-Based Traitor Tracing

An IBTT scheme consists of the following five algorithms.

Setup$_T(\lambda)$. The setup algorithm takes as input a security parameter λ and returns a key pair (MPK, MSK), where MPK denotes master public key and MSK denotes master secret key.
KGen$_T(ID, n, t, MSK)$. The key generation algorithm takes as input an identity ID, the number bound of users n, the collusion bound of traitors t, and the master secret key MSK. The algorithm returns n private keys $\{d_{ID,1}, d_{ID,2}, \cdots, d_{ID,n}\}$, where $d_{ID,i}$ is given to the ith user.
Enc$_T(ID, M, MPK)$. The encryption algorithm takes as input an identity ID, a message M and the master public key MPK and returns a ciphertext C denoted by $C = \mathsf{Enc}_T(ID, M, MPK)$.
Dec$_T(C, d_{ID,i})$. The decryption algorithm takes as input the ciphertext C and a private key $d_{ID,i}$ and outputs $\mathsf{Dec}_T(C, d_{ID,i}) \in \{M, \perp\}$.
Trace$_T(\mathcal{PD}_{ID}, ID, MSK)$. The tracing algorithm takes as input $\mathcal{PD}_{ID}$, a pirate decryption box for ID, the identity ID and the master secret key MSK and returns a set of traitors $\mathbb{T} \subseteq \{1, 2, \cdots, n\}$.

Let $\mathcal{PD}_{\mathbb{T}}$ be all traitors used to create $\mathcal{PD}_{ID}$. For correctness, it requires

$$\mathsf{Dec}_T\Big(\mathsf{Enc}_T(ID, M), d_{ID,i}\Big) = M, \quad \text{and} \quad \mathbb{T} \subseteq \mathcal{PD}_{\mathbb{T}}$$

hold for all (MPK, MSK), ID, M, i and $d_{ID,i}$.

We define the IBTT scheme with n total number bound of users and t collusion bound of traitors for each identity ID. The concrete parameter (n, t) for each identity can be different. For simplicity, we assume that all identities are set with the same collusion parameter (n, t). The algorithm $\mathsf{KGen}(ID, n, t, MSK)$ is written as $\mathsf{KGen}(ID, MSK)$ for shorthand. Our above IBTT definition is similar to the definition in [2], but ours has a more flexible (n, t) parameter for each identity.

Security. Boneh and Naor [8] defined semantic security and t-collusion resistance for traitor tracing. Following their security definitions, we define the security of an IBTT scheme in terms of the following two games.

Game 1. The first game is semantically secure against chosen-plaintext attacks. It is stated as follows:
Setup. The challenger runs the $\mathsf{Setup}_T(\lambda)$ algorithm to generate (MPK, MSK) and gives the adversary MPK.
Phase 1. For a key query on ID from the adversary, the challenger responds by running the $\mathsf{KGen}_T(ID, MSK)$ algorithm and returning the private keys $\{d_{ID,1}, d_{ID,2}, \cdots, d_{ID,n}\}$ to the adversary.
Challenge. The adversary outputs an identity ID^* and two different messages M_0, M_1 for challenge. This challenge identity must be different from other identities in the query phase. The challenger responds by flipping a coin $c \in \{0, 1\}$, running the algorithm $\mathsf{Enc}_T(ID^*, M_c, MPK)$ and returning the challenge ciphertext $C^* = \mathsf{Enc}_T(ID^*, M_c, MPK)$ to the adversary.
Phase 2. For a private key query on $ID \neq ID^*$ from the adversary, the challenger responds the same as the phase 1.
Guess. The adversary outputs the guess c' as to the bit c and wins the game if $c = c'$.

We define the advantage of the adversary in the above game as $\mathsf{Adv}^s_T = \big|\Pr[c = c'] - \frac{1}{2}\big|$.

Definition 5. *We say that an IBTT scheme is* (T, q_1, ϵ)*-semantically secure against chosen-plaintext attacks if for all polynomial time adversaries who makes* q_1 *private key queries, we have that* $\epsilon = \mathsf{Adv}_T^s$ *is a negligible function of* λ*.*

Game 2. The second game is traceable against t-collusion attacks. It is stated as follows:
Setup. The challenger runs the $\mathsf{Setup}_T(\lambda)$ algorithm to generate (MPK, MSK) and gives MPK to the adversary.
Query. For a key query on (ID, i) from the adversary, the challenger responds by running the $\mathsf{KGen}_T(ID, MSK)$ algorithm and returning the private keys $d_{ID,i}$ to the adversary.
Challenge. The adversary outputs a pirate decryption box $\mathcal{PD}_{ID^*}$ for ID^*.
Trace. The challenger runs the $\mathsf{Trace}_T(\mathcal{PD}_{ID^*}, ID^*, MSK)$ algorithm and outputs a set $\mathbb{T} \subseteq \{1, 2, \cdots, n\}$. Let $\mathbb{I}_{ID^*}$ be the set of indices that the adversary ever made private key query on (ID^*, i) for all $i \in \mathbb{I}_{ID^*}$. The adversary wins the game if

- The pirate decryption box $\mathcal{PD}_{ID^*}$ is perfect with

$$\Pr\left[\mathcal{PD}_{ID^*}\Big(\mathsf{Enc}_T(ID^*, M, MPK)\Big) = M\right] = 1.$$

- The traitor set $\mathbb{T}$ is empty, or not a subset of $\mathbb{I}_{ID^*}$.
- There are t private key queries on ID^* at most, i.e., $|\mathbb{I}_{ID^*}| \leq t$.

We define the advantage of the adversary in the above game as Adv_T^c.

Definition 6. *An IBTT scheme is* (T, n, t, ϵ)*-collusion resistant if for all* T *polynomial time adversaries who makes at most* t *private keys on the challenge identity, we have that* $\epsilon = \mathsf{Adv}_T^c$ *is a negligible function of* λ*.*