10-2018

Efficient attribute-based encryption with blackbox traceability

Shengmin XU
Singapore Management University, smxu@smu.edu.sg

Guomin YANG

Yi MU

Ximeng LIU

Efficient Attribute-Based Encryption with Blackbox Traceability

Shengmin Xu[1,2(✉)], Guomin Yang[1], Yi Mu[4], and Ximeng Liu[2,3]

[1] Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong, Australia
{sx914,gyang}@uow.edu.au
[2] School of Information Systems, Singapore Management University, Singapore, Singapore
snbnix@gmail.com
[3] College of Mathematics and Computer Science, Fuzhou University, Fuzhou, Fujian, China
[4] Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou, China
ymu.ieee@gmail.com

Abstract. Traitor tracing scheme can be used to identify a decryption key is illegally used in public-key encryption. In CCS'13, Liu et al. proposed an attribute-based traitor tracing (ABTT) scheme with blackbox traceability which can trace decryption keys embedded in a decryption blackbox/device rather than tracing a well-formed decryption key. However, the existing ABTT schemes with blackbox traceability are based on composite order group and the size of the decryption key depends on the policies and the number of system users. In this paper, we revisit blackbox ABTT and introduce a new primitive called attribute-based set encryption (ABSE) based on key-policy ABE (KP-ABE) and identity-based set encryption (IBSE), which allows aggregation of multiple related policies and reduce the decryption key size in ABTT to be irrelevant to the number of system users. We present a generic construction of the ABTT scheme from our proposed ABSE scheme and fingerprint code based on the Boneh-Naor paradigm in CCS'08. We then give a concrete construction of the ABSE scheme which can be proven secure in the random oracle model under the decisional BDH assumption and a variant of q-BDHE assumption.

Keywords: Public-key cryptosystems · Attribute-based encryption
Blackbox traceability

1 Introduction

Public-key encryption is the most fundamental primitive of public-key cryptography. However, the traditional public-key infrastructure (PKI) suffers from the certificate management problem. To overcome this drawback, identity-based

J. Baek et al. (Eds.): ProvSec 2018, LNCS 11192, pp. 182–200, 2018.
https://doi.org/10.1007/978-3-030-01446-9_11

encryption (IBE) has been proposed, and it provides a new paradigm for public-key encryption [3]. IBE uses the identity string (e.g., email or IP address) of a user as the public key of that user. The sender using an IBE does not need to look up the public keys and the corresponding certificates of the receiver. However, IBE cannot efficiently handle data sharing among multiple users. To address this issue, attribute-based encryption (ABE) was introduced [3] to provide fine-grained access control. However, encryption schemes supporting multiple valid decryptors suffer the problem of decryption key re-distribution. A malicious user might have an intention to leak the decryption key or some decryption privileges by giving the decryption key or decryption blackbox/device to other unauthorized users for financial gain or for some other incentives.

To address this problem, traitor tracing scheme [4] was proposed to identify the traitor who violates the copyright restrictions. A traitor tracing scheme comprises an encryption key, a tracing key and n decryption keys, where n is the number of system users. Each legitimate user is given a unique decryption key that can decrypt any properly encrypted message. The tracing key can trace at least one user decryption used to construct the decryption blackbox/device. A traitor tracing scheme is said to be t-collusion resistant if the tracing is still successful against t colluded users. In this paper, we investigate the traitor tracing scheme in the ABE setting.

ABE with traitor tracing (ABTT) has been studied in the literature [9–11,14]. There are two levels of traceability depending on the way of tracing traitors. Level one is whitebox traceability [10,14], by which given a well-formed decryption key as input, a tracing algorithm can find out user who owns this decryption key. Level two is blackbox traceability [9,11], by which given a decryption blackbox/device, which the decryption key and even decryption algorithm could be hidden, the tracing algorithm, which treats the decryption blackbox as an oracle, can still find out the malicious user whose key has been used in constructing the decryption blackbox.

In this paper, we present a new construction of ABTT based on a new primitive called attribute-based set encryption (ABSE) inspired by KP-ABE [16] and IBSE [7]. We then describe our ABTT scheme from our proposed ABSE scheme and fingerprint code [4] to provide the efficient traitor tracing mechanism in the ABE setting. Our ABSE scheme is provably secure in the random oracle model under the decisional BDH assumption and a variant of q-BDHE assumption. Compared with the previous ABTT schemes, our ABTT scheme only requires the prime order group and the size of the decryption key only depends on the access policies as traditional KP-ABE rather than both the access policies and the number of system users.

1.1 Related Work

Sahai and Waters [16] introduced ABE that allows users to selectively share their encrypted data at a fine-grained level. To enrich expressiveness of access control policies, Goyal et al. [6] and Bethencourt et al. [2] then proposed key-policy and ciphertext-policy ABE schemes, respectively. In KP-ABE schemes,

attribute sets are used to annotate ciphertexts, and private keys are associated with access structures that specify which ciphertexts the user will be entitled to decrypt. Ciphertext-policy ABE (CP-ABE) proceeds in a dual way, by assigning attribute sets to private keys and letting senders specify an access policy that receivers' attribute sets should comply with. However, the seminal works [2,6,16] of ABE schemes suffer some common problems, such as the size of the key and the ciphertext are linear to the attribute set and security proofs are under the selective model. Attrapadung et al. [1] proposed the first constant-size ABE and Lewko et al. [8] provided first fully secure ABE with dual encryption system [17], respectively. Unfortunately, the above schemes must define the attribute universe at setup phase or have to sacrifice the security by deploying the random oracle to scale up the attribute universe. Rouselakis and Waters [15] proposed large universe ABE schemes with selective security that can overcome this problem.

The concept of whitebox ABTT was introduced by Liu et al. [10] to identify the traitors who violate the copyright restrictions in the ABE setting. However, Liu et al.'s work must define the attribute universe at setup phase and cannot support the large attribute universe. To overcome this drawback, several ABTT [14,18] schemes were proposed to support the large universe. Liu et al. [9,11] introduced blackbox ABTT to solve a practical problem that the decryption key may not be a well-formed key and it may be embedded in a decryption blackbox/device. However, the decryption key in the proposed scheme is in the order of $O(|\mathcal{S}| + \sqrt{n})$, where $|\mathcal{S}|$ represents the number of attributes in the attribute set $\mathcal{S}$ and n is the number of system users. After that, some other works [12,13,19,20] have been proposed to improve efficiency, functionality or security. Unfortunately, the above schemes require the composite order group or large decryption key size depending on the number of system users.

1.2 Contribution

In this paper, we proposed an efficient blackbox ABTT scheme. Compared to previous ABTT schemes, our scheme provides the blackbox traceability based on prime order group and decryption key only relates to the access policies as the traditional ABE rather than both the access policies and the number of system users. Note that most of the previous blackbox ABTT schemes are based on Boneh et al.' traitor tracing scheme [5], which requires the decryption key in the order of $O(\sqrt{n})$, where n is the number of system users.

Our approach utilizes fingerprint codes to realize the traitor tracing mechanism. However, the trivial solution needs $O(n)$ private keys by appending a unique index from fingerprint codes as the user identifier at the end of each access policy. Suppose the i^{th} user has an access structure $\mathbb{A} = (\mathbb{M}, \rho)$ with the matrix $\mathbb{M}$ of size $d \times l$ and mapping function ρ mapping each row in the matrix to the attribute universe. The trivial solution requires to extend each row j in the matrix to a set of policies for tracing traitors in blackbox, e.g., the j^{th} row policy $(\mathbb{M}_j, \rho(j))$ extends to a set of policies

$$(\mathbb{M}_j, \rho(j)\|1\|w_1^{(i)}), (\mathbb{M}_j, \rho(j)\|2\|w_2^{(i)}), \ldots, (\mathbb{M}_j, \rho(j)\|\ell\|w_\ell^{(i)}), \tag{1}$$

where ℓ denotes the size of codeword in fingerprint codes and $w_k^{(i)}$ represents the k^{th} position codeword for the i^{th} user. It is obvious that the trivial solution requires the policy size of $\ell \times d \times l$ eventually.

To reduce the size of the decryption key, we introduce a new cryptographic primitive called attribute-based set encryption (ABSE). Roughly speaking, our ABSE compresses the decryption key for a set of policies as shown in Eq. (1) to two policies

$$(\mathbb{M}_j, \rho(j) \| \mathcal{S}_0 \| 0) \text{ and } (\mathbb{M}_j, \rho(j) \| \mathcal{S}_1 \| 1)$$

with $O(1)$ size for each row in the access policy, where $\mathcal{S}_b$ represents a set of indices recording all positions $j \in [\ell]$ *s.t.* $w_j^{(i)} = b$ ($w_j^{(i)}$ representing j^{th} position in the codeword for the i^{th} user). Finally, the decryption key has the policy size of $d \times l$ as the traditional ABE system.

We provide a generic construction of ABTT from fingerprint codes and ABSE under the prime order group, and it is provably secure based on the underlying fingerprint code and ABSE. The ABSE scheme instantiated in this paper is provably secure in random oracle model based on the decisional BDH assumption and a variant of q-BDHE assumption.

1.3 Outline

We introduce some preliminaries in Sect. 2 and provide the generic construction of the ABTT scheme and its proof in Sect. 3. In Sect. 4, we provide the concrete construction of ABSE scheme and its formal proof. We then summarize this paper in Sect. 5.

2 Preliminaries

2.1 Notations

Let $\mathbb{N}$ denote the set of all natural numbers, and for $n \in \mathbb{N}$, we define $[n] := \{1, \ldots, n\}$. If a and b are strings, then $|a|$ denotes the bit-length of a, $a \| b$ denotes the concatenation of a and b. Let $\vec{u} := (u_1, u_2, \ldots, u_\ell)$ denote a vector of dimension ℓ in $\mathbb{Z}_p$. Let the Greek character λ denote a security parameter. A function $\epsilon(\lambda) : \mathbb{N} \to [0, 1]$ is said to be negligible if for all positive polynomials $p(\lambda)$ and all sufficiently large $\lambda \in \mathbb{N}$, we have $\epsilon(\lambda) < 1/p(\lambda)$. To simplify, ϵ is used to represent negligible.

2.2 Bilinear Map

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic multiplicative groups of prime order p and g be a generator of $\mathbb{G}$. The map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is said to be an admissible bilinear pairing if the following properties hold true.

- Bilinearity: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$.
- Non-degeneration: $e(g, g) \neq 1$.
- Computability: it is efficient to compute $e(u, v)$ for any $u, v \in \mathbb{G}$.

2.3 Decisional Bilinear Diffie-Hellman Assumption

Let $a, b, c, z \in \mathbb{Z}_p$ be chosen at random and g be a generator of $\mathbb{G}$. The decisional Bilinear Diffie-Hellman (BDH) assumption [16] is that no probabilistic time algorithm can distinguish the tuple $(g^a, g^b, g^c, e(g,g)^{abc})$ from the tuple $(g^a, g^b, g^c, e(g,g)^z)$ with a non-negligible advantage over random guess.

2.4 Modified *q*-Biliner Diffie-Hellman Exponent Assumption

Let $a \in \mathbb{Z}_p$ be chosen at random and g be a generator of $\mathbb{G}$. The modified q-Bilinear Diffie-Hellman Exponent (q-BDHE) [7] is that giving the terms $g, g^{(a)}, g^{(a^2)}, \ldots, g^{a^q}, g^{(a^{2q+2})}, g^{(a^{2q+3})}, \ldots, g^{(a^{3q+1})} \in \mathbb{G}^{2q+1}$, no probabilistic time algorithm can output the term $e(g,g)^{a^{2q+1}}$ with a non-negligible advantage.

2.5 Access Structure and Monotone Span Program

We recall the definition of access structures and monotone span program, as defined in [6].

Definition 1 (Access Structure). *Let $\{P_1, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \subseteq \mathbb{A}$. A monotone access structure is a monotone collection $\mathbb{A}$ of non-empty subsets of $\{P_1, \ldots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called authorized sets, and the sets not in $\mathbb{A}$ are called unauthorized sets.*

Definition 2 (Monotone Span Program (MSP)). *Let $\mathcal{K}$ be a field and $\{x_1, \ldots, x_n\}$ be a set of variables. A MSP over $\mathcal{K}$ is labeled matrix $\tilde{M}(\mathbb{M}, \rho)$ where $\mathbb{M}$ is a matrix over $\mathcal{K}$, and ρ is a labeling of the rows of $\mathbb{M}$ by literals from $\{x_1, \ldots, x_n\}$ (every row is labeled by one literal). A MSP accepts or rejects an input by the following criterion. For every input set $\mathcal{S}$ if literals, define the submatrix $\mathbb{M}_{\mathcal{S}}$ of $\mathbb{M}$ consisting of those rows whose labels are in S, i.e., rows labeled by some i such that $i \in \mathcal{S}$. The MSP $\tilde{M}$ accepts $\mathcal{S}$ of and only if $\vec{1} \in \mathsf{span}(\mathbb{M}_S)$, i.e., some linear combination of the rows of $\mathbb{M}_{\mathcal{S}}$ given the all-one vector $\vec{1}$. The MSP $\tilde{M}$ computes a boolean function f_M if it accepts exactly those input $\mathcal{S}$ where $f_{\mathbb{M}}(\mathcal{S}) = 1$. The size of $\tilde{M}$ is the number of rows in $\mathbb{M}$.*

In the rest of paper, we define $\mathbb{M}$ as a matrix with $d \times l$ elements, where d is a dynamic value depending on the access policy $\mathbb{A}$. $\mathbb{M}_i$ stands for the i^{th} row of the matrix $\mathbb{M}$ and is a vector size of l. In our proposed scheme, each row of the matrix $\mathbb{M}$ maps to different attributes. For simply the notation, let $\mathbb{A}(\mathcal{S}) = 1$ indicate the attribute set $\mathcal{S}$ satisfies the access policy $\mathbb{A}$ and $\mathbb{A}(\mathcal{S}) = 0$ denote the attribute set $\mathcal{S}$ does not satisfy the access policy $\mathbb{A}$.

2.6 Fingerprint Code

The fingerprint code [4] is defined as follows.

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

Definition 3 (Fingerprint Code). *Let $\mathcal{FC}$ denote a fingerprint code and it consists of two algorithms defined as follows.*

$\mathcal{FC}.\mathsf{Gen}(n,t,\lambda) \rightarrow (\Gamma, tk)$. *On input the number of codewords n, the collusion bound t and a security parameter λ, the generation algorithm outputs a codebook Γ containing n codewords $\{\bar{w}^{(1)}, \bar{w}^{(2)}, \cdots, \bar{w}^{(n)}\}$ in $\{0,1\}^\ell$ with length $\ell = \ell(n,t,\lambda)$ and a tracing key tk.*

$\mathcal{FC}.\mathsf{Trace}(\bar{w}^*, tk) \rightarrow \mathbb{S}$. *On input a codeword $\bar{w}^* \in \{0,1\}^\ell$ and the tracing key tk, the tracing algorithm outputs a subset $\mathbb{S} \subseteq [n]$. Informally, let $\mathbb{W}$ be a subset of Γ, if $\bar{w}^* \in F(\mathbb{W})$, we have that the output set $\mathbb{S}$ is a subset of $\mathbb{W}$.*

Definition 4 (Security Model of Fingerprint Code). *The security definition of a fingerprint code from the following experiment:*

$$\begin{aligned}&\mathbf{Exp}_{\mathcal{FC},\mathcal{A}}(n,t,\lambda)\\ &\quad (\Gamma, tk) \leftarrow \mathcal{FC}.\mathsf{Gen}(n,t,\lambda);\\ &\quad \bar{w}^* \leftarrow \mathcal{A}^{\mathcal{O}(\cdot)}(n,t);\\ &\quad \textit{If } \mathcal{FC}.\mathsf{Trace}(\bar{w}^*, tk) \not\subseteq \emptyset \textit{ return } 1 \textit{ else return } 0.\end{aligned}$$

$\mathcal{O}(\cdot)$ is a oracle that allows the adversary queries the index $\mathbb{I} \subseteq [n]$ with $|\mathbb{I}| \leq t$, the challenger responds by returning the codewords $\mathbb{W} = \{\bar{w}_i\}_{i \in \mathbb{I}}$ to the adversary. Note that the challenge codeword $\bar{w}^*$ is not belongs to the returning codeword set $\mathbb{W}$, such that $\bar{w}^* \notin \mathbb{W}$.

A fingerprint code is t-collusion resistant if for all adversaries, all n, t satisfying $n \geq t$, all $\mathbb{I}$ satisfying $\mathbb{I} \subseteq [n]$ and $|\mathbb{I}| \leq t$, we have that the advantage of the adversary in the above game $\mathbf{Adv}_{\mathcal{FC},\mathcal{A}}$ is negligible:

$$\mathbf{Adv}_{\mathcal{FC},\mathcal{A}}(n,t,\lambda) = \Big| \Pr[\mathbf{Exp}_{\mathcal{FC},\mathcal{A}}(n,t,\lambda) = 1] \Big|.$$

2.7 Attribute-Based Encryption with Traitor Tracing

We refine the definition and security model in [9,11]. It is worth to notice that the augmented ABTT scheme is considered in the previous works since the encryption algorithm needs an additional index for labeling users, which works as an identifier that allows another user to identify the malicious users. In our proposed scheme, we use fingerprint codes as a different tracing method, thus our scheme does not need to consider augmented ABTT.

Definition 5 (Attribute-Based Encryption with Traitor Tracing). *Let $\mathcal{ABTT}$ denote an ABTT scheme and an ABTT scheme with the attribute set Ω that supports policies $\mathcal{P}$ with the message space $\mathcal{M}$ consists of five algorithms as follows.*

$\mathcal{ABTT}.\mathsf{Setup}(n, t, \lambda) \rightarrow (pp, msk, tk)$. *The probabilistic setup algorithm takes the number of system users n, the collusion bound t and a security parameter λ as input, and outputs the public parameter pp, the master secret key msk and the tracing key tk.*

$\mathcal{ABTT}.\mathsf{KeyGen}(msk, \mathbb{A}) \rightarrow sk_{\mathbb{A}}$. *The probabilistic key generation algorithm takes the master secret key msk and the an access structure $\mathbb{A} \in \mathcal{P}$ as input, and outputs the secret key $sk_{\mathbb{A}}$.*

$\mathcal{ABTT}.\mathsf{Enc}(pp, m, \mathcal{S}) \rightarrow ct_{\mathcal{S}}$. *The probabilistic encryption algorithm takes the public parameter pp, a message $m \in \mathcal{M}$ and an attribute set $\mathcal{S} \subseteq \Omega$ as input, and outputs the ciphertext $ct_{\mathcal{S}}$.*

$\mathcal{ABTT}.\mathsf{Dec}(sk_{\mathbb{A}}, ct_{\mathcal{S}}) \rightarrow m$. *The deterministic decryption algorithm takes the secret key $sk_{\mathbb{A}}$ and the ciphertext $ct_{\mathcal{S}}$ as input, and outputs a message $m \in \mathcal{M}$.*

$\mathcal{ABTT}.\mathsf{Trace}^{\mathcal{PD}}(tk) \rightarrow \mathbb{S}$. *The deterministic tracing algorithm is an oracle algorithm takes is given as input the tracing key tk. The tracing algorithm queries the pirate decoders $\mathcal{PD}$ as a blackbox oracle. It outputs a set of traitors $\mathbb{S}$ which is a subset of $[n]$.*

Next, we define the security of the traitor tracing scheme in terms of the following games, called selective indistinguishability under chosen plaintext attack (sIND-CPA) and traceability against t-collusion attack.

Definition 6 (sIND-CPA **in Attribute-Based Encryption with Traitor Tracing**). *The security definition of an ABTT scheme for message hiding is based on the following experiment:*

$$
\begin{aligned}
&\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{ABTT},\mathcal{A}}(n, t, \lambda)\\
&\quad \mathcal{S}^* \leftarrow \mathcal{A}(n, t, \lambda);\\
&\quad (pp, msk, tk) \leftarrow \mathcal{ABTT}.\mathsf{Setup}(n, t, \lambda);\\
&\quad (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{ABTT}.\mathsf{KeyGen}}(\cdot)}(pp);\\
&\quad b \leftarrow \{0, 1\};\\
&\quad ct_{\mathcal{S}^*} \leftarrow \mathcal{ABTT}.\mathsf{Enc}(pp, m_b, \mathcal{S}^*);\\
&\quad b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{ABTT}.\mathsf{KeyGen}}(\cdot)}(ct_{\mathcal{S}^*});\\
&\quad \textit{If } b = b' \textit{ return } 1 \textit{ else return } 0.
\end{aligned}
$$

$\mathcal{O}_{\mathcal{ABTT}.\mathsf{KeyGen}}(\cdot)$ *represents the key generation oracle that allows the adversary to query an access structure $\mathbb{A} \in \mathcal{P}$ except $\mathbb{A}(\mathcal{S}^*) = 1$, and it returns the secret key $sk_{\mathbb{A}}$ by running $\mathcal{ABTT}.\mathsf{KeyGen}(msk, \mathbb{A})$.*

An ABTT scheme is said to be sIND-CPA *secure if for any probabilistic polynomial time adversary* $\mathcal{A}$*, the following advantage is negligible:*

$$\mathbf{Adv}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{ABTT},\mathcal{A}}(n,t,\lambda) = \Big|\Pr[\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{ABTT},\mathcal{A}}(n,t,\lambda)=1] - 1/2\Big|.$$

Definition 7 (Traceability against t-collusion Attack in Attribute-Based Encryption with Traitor Tracing). *The security definition of an ABTT scheme for traceability is based on the following experiment:*

$\mathbf{Exp}^{\mathsf{Trace}}_{\mathcal{ABTT},\mathcal{A}}(n,t,\lambda)$
$\mathcal{S}^* \leftarrow \mathcal{A}(n,t,\lambda)$;
$(pp, msk, tk) \leftarrow \mathcal{ABTT}.\mathsf{Setup}(n,t,\lambda)$;
$\mathcal{PD} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{ABTT}.\mathsf{KeyGen}}(\cdot)}(pp)$;
$\mathbb{S} \leftarrow \mathcal{ABTT}.\mathsf{Trace}^{\mathcal{PD}}(tk)$;
If $\Pr[\mathcal{PD}(\mathcal{ABTT}.\mathsf{Enc}(pp,m,\mathcal{S}^*))=m]=1$ *and* $\mathbb{S}\subseteq\emptyset$ *or* $\mathbb{S}\not\subseteq\mathbb{I}$ *return* 1 *else return* 0.

$\mathcal{O}_{\mathcal{ABTT}.\mathsf{KeyGen}}(\cdot)$ *represents the key generation oracle that allows the adversary to query a set of the indices* $\mathbb{I}\subseteq[n]$ *(*$|\mathbb{I}|\leq t$*), and it runs* $\mathcal{ABTT}.\mathsf{KeyGen}(msk,\mathbb{A})$ *to all* $i\in\mathbb{I}$ *and* $\mathbb{A}(\mathcal{S}^*)=0$*, and then returns the secret key* $\{sk_i\}_{i\in\mathbb{I}}$*. Notice that the adversary cannot adaptively query this oracle since this oracle only runs once before the challenge phase.*

An ABTT scheme is said to be t-collusion resistant if for any probabilistic polynomial time adversary $\mathcal{A}$*, the following advantage is negligible:*

$$\mathbf{Adv}^{\mathsf{Trace}}_{\mathcal{ABTT},\mathcal{A}}(n,t,\lambda) = \Pr[\mathbf{Exp}^{\mathsf{Trace}}_{\mathcal{ABTT},\mathcal{A}}(n,t,\lambda)=1].$$

2.8 Attribute-Based Set Encryption

An IBSE scheme [7] was introduced to improve the efficiency of identity-based traitor tracing scheme by reducing the size of private key and ciphertext. We refined the definition and security model in the IBE setting to the ABE setting. It is worth to notice that the following algorithms have some elements in the definition of fingerprint code as given Sect. 2.6.

Definition 8 (Attribute-Based Set Encryption). *Let* $\mathcal{ABSE}$ *be an ABSE scheme and an ABSE scheme with the attribute set* Ω *that supports the policies* $\mathcal{P}$ *and the message space* $\mathcal{M}$ *consists of four algorithms as follows.*

$\mathcal{ABSE}.\mathsf{Setup}(n,\lambda)\rightarrow(pp,msk)$*. The probabilistic setup algorithm takes the number* n *and a security parameter* λ *as input, and outputs the public parameter* pp *and the master secret key* msk*.*

$\mathcal{ABSE}.\mathsf{KeyGen}(msk,\mathbb{A},b,\mathcal{L})\rightarrow sk_{\mathbb{A}}$*. The probabilistic key generation algorithm takes the master secret key* msk *and the access structure* $\mathbb{A}\in\mathcal{P}$*, a bit* $b\in\{0,1\}$ *and a list of indices* $\mathcal{L}$ *(*$|\mathcal{L}|\leq\ell$ *and* $\mathcal{L}$ *represents all indices* $j\in[\ell]$ *s.t.* $w_j^{(i)}=b$*,*

where $w_j^{(i)}$ representing j^{th} position in the codeword for the i^{th} user) as input, and outputs the private key $sk_{\mathbb{A}}$.

$\mathcal{ABSE}.\mathsf{Enc}(pp, m, \mathcal{S}, b, \ell, \tau) \rightarrow ct_{\mathcal{S}}$. *The probabilistic encryption algorithm takes the public parameter pp, the message $m \in \mathcal{M}$, an attribute set $\mathcal{S}$, a bit $b \in \{0,1\}$, a number ℓ representing the size of each codeword and a number τ ($\tau \leq \ell$ which represents the position in the codeword and will be used to form the attribute $A\|\tau\|b$ for all $A \in \mathcal{S}$) as input, and outputs a ciphertext $ct_{\mathcal{S}}$.*

$\mathcal{ABSE}.\mathsf{Dec}(sk_{\mathbb{A}}, ct_{\mathcal{S}}, b) \rightarrow m$. *The deterministic decryption algorithm takes the secret key $sk_{\mathbb{A}}$, the ciphertext $ct_{\mathcal{S}}$ and a bit $b \in \{0,1\}$ as input, and outputs a message $m \in \mathcal{M}$.*

Next, we describe the security of selective indistinguishability under chosen plaintext attack in the random oracle model (sIND-CPA security) for the ABSE setting.

Definition 9 (sIND-CPA **in Attribute-Based Set Encryption**). *The security definition of an ABSE scheme is based on the following experiment:*

$$
\begin{aligned}
&\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{ABSE},\mathcal{A}}(n,\lambda)\\
&\quad \mathcal{S}^* \leftarrow \mathcal{A}(n,\lambda);\\
&\quad (pp, msk) \leftarrow \mathcal{ABSE}.\mathsf{Setup}(n,\lambda);\\
&\quad (m_0, m_1, b, \tau) \leftarrow \mathcal{A}^{\mathcal{O}}(pp);\\
&\quad c \leftarrow \{0,1\};\\
&\quad ct_{\mathcal{S}^*} \leftarrow \mathcal{ABSE}.\mathsf{Enc}(pp, m_c, \mathcal{S}^*, b, \ell, \tau);\\
&\quad c' \leftarrow \mathcal{A}^{\mathcal{O}}(ct_{\mathcal{S}^*});\\
&\quad \textit{If } c = c' \textit{ return } 1 \textit{ else return } 0.
\end{aligned}
$$

In the random oracle setting $\mathcal{O}$ represent a set of oracles, $\{\mathcal{O}_{\mathcal{ABSE}.\mathsf{KeyGen}}(\cdot,\cdot,\cdot), \mathcal{O}_H(\cdot)\}$, and the details are given in below.

- $\mathcal{O}_{\mathcal{ABSE}.\mathsf{KeyGen}}(\cdot,\cdot,\cdot)$ *is the key generation oracle that allows the adversary to query on the access structure $\mathbb{A}$ (expect $\mathbb{A}(\mathcal{S}^*) = 1$), a bit b and a set of indices $\mathcal{L}$, and the challenger runs the $\mathcal{ABSE}.\mathsf{KeyGen}(msk, \mathbb{A}, b, \mathcal{L})$ algorithm and returns the secret key $sk_{\mathbb{A}}$ to the adversary.*
- *In random oracle model, we provide the oracle $\mathcal{O}_H(\cdot)$ that allows the adversary to query on the message s, if s has been queried, it will output the same has output; otherwise, it outputs a random hash output. Note that, we may provide multiple hash oracles in the random oracle model.*

An ABSE scheme is said to be sIND-CPA *secure if for any probabilistic polynomial time adversary $\mathcal{A}$, the following advantage is negligible:*

$$
\mathbf{Adv}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{ABSE},\mathcal{A}}(n,\lambda) = \Big|\Pr[\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{ABSE},\mathcal{A}}(n,\lambda) = 1] - 1/2\Big|.
$$

3 Attribute-Based Encryption with Traitor Tracing

3.1 Generic Construction

Let $\mathcal{FC} = (\mathsf{Gen}, \mathsf{Trace})$ be an fingerprint code and $\mathcal{ABSE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an ABSE scheme. Our ABTT is described as follows.

$\mathcal{ABTT}.\mathsf{Setup}(n, t, \lambda) \rightarrow (pp, msk, tk)$. Let $\ell = \ell(n, t, \lambda)$ be the length of codeword in the fingerprint code. The setup algorithm runs

$$\mathcal{FC}.\mathsf{Gen}(n, t, \lambda) \rightarrow (\Gamma, tk),$$
$$\mathcal{ABSE}.\mathsf{Setup}(n, \lambda) \rightarrow (pp_0, msk_0).$$
$$\mathcal{ABSE}.\mathsf{Setup}(n, \lambda) \rightarrow (pp_1, msk_1).$$

The public parameter is $pp = (\Gamma, pp_0, pp_1)$ and the master secret key is $msk = (msk_0, msk_1)$.

$\mathcal{ABTT}.\mathsf{KeyGen}(msk, \mathbb{A}) \rightarrow sk_{\mathbb{A}}$. For the i^{th} user, this algorithm assigns the i^{th} codeword $\bar{w}^{(i)}$ to this user and initializes two empty lists $\mathcal{L}_0$ and $\mathcal{L}_1$. For $j \in [\ell]$, the algorithm derives the ciphertext $sk_{\mathbb{A}}$ as: If $w_j^{(i)} = 0$,

$$\mathcal{L}_0 \leftarrow \mathcal{L}_0 \cup \{j\};$$

otherwise,

$$\mathcal{L}_1 \leftarrow \mathcal{L}_1 \cup \{j\}.$$

The key generation algorithm runs

$$\mathcal{ABSE}.\mathsf{KeyGen}(msk_0, \mathbb{A}, 0, \mathcal{L}_0) \rightarrow sk_{\mathbb{A}}^{(0)},$$
$$\mathcal{ABSE}.\mathsf{KeyGen}(msk_1, \mathbb{A}, 1, \mathcal{L}_1) \rightarrow sk_{\mathbb{A}}^{(1)}.$$

Finally, it returns the secret key $sk_{\mathbb{A}} = (sk_{\mathbb{A}}^{(0)}, sk_{\mathbb{A}}^{(1)})$ for the access structure $\mathbb{A}$.

$\mathcal{ABTT}.\mathsf{Enc}(pp, m, \mathcal{S}) \rightarrow ct_{\mathcal{S}}$. The encryption algorithm randomly pick $\tau \in \mathbb{Z}_p$ $(\tau \leq \ell)$. Then, it runs

$$\mathcal{ABSE}.\mathsf{Enc}(pp_0, m, \mathcal{S}, 0, \tau) \rightarrow ct_{\mathcal{S}}^{(0)},$$
$$\mathcal{ABSE}.\mathsf{Enc}(pp_1, m, \mathcal{S}, 1, \tau) \rightarrow ct_{\mathcal{S}}^{(1)}.$$

The ciphertext is $ct_{\mathcal{S}} = (\tau, ct_{\mathcal{S}}^{(0)}, ct_{\mathcal{S}}^{(1)})$.

$\mathcal{ABTT}.\mathsf{Dec}(sk_{\mathbb{A}}, ct_{\mathcal{S}}) \rightarrow m$. For the i^{th} user, the decryption algorithm runs as follows. If $w_{\tau}^{(i)} = 0$, it runs

$$\mathcal{ABSE}.\mathsf{Dec}(sk_{\mathbb{A}}^{(0)}, ct_{\mathcal{S}}^{(0)}, 0) \rightarrow m;$$

otherwise, it runs

$$\mathcal{ABSE}.\mathsf{Dec}(sk_{\mathbb{A}}^{(1)}, ct_{\mathcal{S}}^{(1)}, 1) \rightarrow m.$$

$\mathcal{ABTT}.\mathsf{Trace}^{\mathcal{PD}}(tk) \to \mathbb{S}$. Suppose the pirate decoder $\mathcal{PD}$ claims to be able to decrypt any message $m \in \mathcal{M}$ under the access structure $\mathbb{A}$: For all $j \in [\ell]$, the tracing algorithm randomly chooses a message $m_j \neq 0$, and derives the ciphertext under the attribute set $\mathcal{S}$ with $\mathbb{A}(\mathcal{S}) = 1$ by running

$$\mathcal{ABSE}.\mathsf{Enc}(pp_0, m_j, \mathcal{S}, 0, \ell, j) \to ct_{\mathcal{S}}^{(0)},$$
$$\mathcal{ABSE}.\mathsf{Enc}(pp_1, 0, \mathcal{S}, 1, \ell, j) \to ct_{\mathcal{S}}^{(1)}.$$

It then sends the ciphertext $ct_{\mathbb{A}} = (j, ct_{\mathcal{S}}^{(0)}, ct_{\mathcal{S}}^{(1)})$ to $\mathcal{PD}$. Let the return from $\mathcal{PD}$ be m'_j. Define the bit w_j^* as

$$w_j^* = \begin{cases} 0 & \text{if } m'_j = m_j, \text{ and} \\ 1 & \text{otherwise.} \end{cases}$$

It outputs the ℓ-bit codeword $\bar{w}^* = w_1^* w_2^* \cdots w_\ell^*$ and returns a set of traitors $\mathbb{S} \subseteq [n]$ by running

$$\mathcal{FC}.\mathsf{Trace}(\bar{w}^*, tk) \to \mathbb{S}.$$

3.2 Security Analysis

Our ABTT scheme above is extended from the public-key traitor tracing scheme [4]. We do not change their paradigm, but replace the public-key encryption scheme in [4] with ABSE. The following theorem shows that our ABTT scheme is secure.

Theorem 1. *Given an attribute-based set encryption scheme* $\mathcal{ABSE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$*, which is* sIND-CPA *secure and fingerprint codes* $\mathcal{FC} = (\mathsf{Gen}, \mathsf{Trace})$*, which is t-collusion resistant, our* $\mathcal{ABTT} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Trace})$ *is a t-collusion resistant attribute-based traitor tracing scheme. Particularly, using the notion in Sect. 2, for all* $t > 0, n > t$*, and all polynomial time adversaries attacking ABTT, there exist polynomial adversaries attacking ABSE or fingerprint code such that*

$$\mathbf{Adv}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{ABTT},\mathcal{A}}(n,t,\lambda) \leq 2\ell \cdot \mathbf{Adv}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{ABSE},\mathcal{A}}(n,t),$$
$$\mathbf{Adv}^{\mathsf{Trace}}_{\mathcal{ABTT},\mathcal{A}}(n,t,\lambda) \leq \mathbf{Adv}_{\mathcal{FC},\mathcal{A}}(n,t,\lambda) + \ell \cdot (\mathbf{Adv}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{ABSE},\mathcal{A}}(n,t) + 1/|\mathcal{M}|).$$

where ℓ *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 [4]. We detail the proof in the full version of this paper[1].

[1] Please contact the authors for it.

4 The Proposed Attribute-Based Set Encryption

4.1 Our Construction

An ABSE scheme with the attribute set Ω that supports policies $\mathcal{P}$ with message space Ω is described as follows.

$\mathcal{ABSE}.\mathsf{Setup}(n, \lambda) \rightarrow (pp, msk)$. The setup algorithm takes the number of system users n and the security parameter λ as input. It first generates the bilinear groups $(g, p, \mathbb{G}, \mathbb{G}_T, e)$ by running the bilinear group generator $\mathcal{G}(\lambda)$. The algorithm randomly chooses the terms $\alpha, \beta \in \mathbb{Z}_p$ and $h \in \mathbb{G}$, then the algorithm computes the terms $h_1, h_2, g_1, g_2, \ldots, g_n$ as:

$$h_1 = g^{\alpha}, h_2 = h^{\beta}, g_1 = g^{(\beta)}, g_2 = g^{(\beta^2)}, \ldots, g_n = g^{(\beta^n)}.$$

It picks three collusion-resistant hash functions H_1, H_2 and H_3 at random:

$$H_1 : \Omega \rightarrow \mathbb{G}, \quad H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_p, \quad H_3 : \mathbb{G}_T \rightarrow \mathcal{M}.$$

The public parameter pp and the master secret key msk are

$$pp = (p, \mathbb{G}, \mathbb{G}_T, e, g, g_1, \ldots, g_n, h, h_1, h_2, H_1, H_2, H_3), \quad msk = (\alpha, \beta).$$

$\mathcal{ABSE}.\mathsf{KeyGen}(msk, \mathbb{A}, b, \mathcal{L}) \rightarrow sk_{\mathbb{A}}$. The key generation algorithm takes the master secret key msk, an access structure $\mathbb{A} = (\mathbb{M}, \rho) \in \mathcal{P}$, a bit $b \in \{0,1\}$ and a index list $\mathcal{L}$ ($|\mathcal{L}| \leq n$) as input, where $\mathbb{M}$ is a matrix of the size $d \times l$ in $\mathbb{Z}_p$ and $\rho : [d] \rightarrow \Omega$ is a mapping function. Let $\vec{u}$ be a random l dimensional vector over $\mathbb{Z}_p$ and $\vec{1} \cdot \vec{u} = \alpha$. For each row i in the matrix $\mathbb{M}$, it randomly chooses $r_i \in \mathbb{Z}_p$ and computes the terms $K_i^{(0)}, K_i^{(1)}$ and $K_i^{(2)}$ as:

$$K_i^{(0)} = h^{\mathbb{M}_i \vec{u}_i} H_1(\rho(i))^{r_i}, \quad K_i^{(1)} = g^{r_i}, \quad K_i^{(2)} = h^{\sum_{j \in \mathcal{L}} \frac{1}{\beta - H_2(\rho(i) \| j \| b)}}.$$

The secret key $sk_{\mathbb{A}}$ is

$$sk_{\mathbb{A}} = \{K_i^{(0)}, K_i^{(1)}, K_i^{(2)}\}_{i \in [d]}.$$

$\mathcal{ABSE}.\mathsf{Enc}(pp, m, \mathcal{S}, b, \ell, \tau) \rightarrow ct_{\mathcal{S}}$. The encryption algorithm takes the public parameter pp, the message $m \in \mathcal{M}$, an attribute set $\mathcal{S} = (A_1, A_2, \ldots, A_k)$, a random bit $b \in \{0,1\}$, and a number $\ell \in \mathbb{Z}_p$ and a number $\tau \in \mathbb{Z}_p$ ($\tau \leq \ell$) as input. It randomly chooses a message $m' \in \mathcal{M}$, and derives the message m'' as:

$$m'' = m \oplus m'.$$

It chooses a random exponent $s \in \mathbb{Z}_p$ and computes the terms $C^{(0)}$ and $C^{(1)}$ as:

$$C^{(0)} = m' \cdot e(h, h_1)^s, \quad C^{(1)} = g^s.$$

For all $i \in [k]$, it computes the terms $C_i^{(2)}, C_i^{(3)}, C_i^{(4)}$ and $C_i^{(5)}$ as:

$$\begin{aligned}
C_i^{(2)} &= H_1(A_i)^s, \\
C_i^{(3)} &= \left(g^{\prod_{j=1}^{\ell} \beta - H_2(A_i \| j \| b)}\right)^{s'}, \\
C_i^{(4)} &= \left(h^{\beta - H_2(A_i \| \tau \| b)}\right)^{s'}, \\
C_i^{(5)} &= m'' \oplus H_3\left(e\left(g^{\frac{\prod_{j=1}^{\ell} \beta - H_2(A_i \| j \| b)}{\beta - H_2(A_i \| \tau \| b)}}, h\right)^{s'}\right).
\end{aligned}$$

The ciphertext $ct_{\mathcal{S}}$ is

$$ct_{\mathcal{S}} = (\tau, C^{(0)}, C^{(1)}, \{C_i^{(2)}, C_i^{(3)}, C_i^{(4)}, C_i^{(5)}\}_{i \in [k]}).$$

$\mathcal{ABSE}.\mathsf{Dec}(sk_{\mathbb{A}}, ct_{\mathcal{S}}, b) \rightarrow m$. The decryption algorithm takes the secret key $sk_{\mathbb{A}}$, the ciphertext $ct_{\mathcal{S}}$ and a bit $b \in \{0,1\}$ as input. It takes the vector $\vec{w}$ *s.t.* $\sum_{\rho(i) \in \mathcal{S}} \mathbb{M}_i w_i = \vec{1}$ and recovers the message m' by computing:

$$\begin{aligned}
& C^{(0)} \cdot \prod_{\rho(i) \in \mathcal{S}} \left(\frac{e(K_i^{(1)}, C_i^{(2)})}{e(K_i^{(0)}, C^{(1)})}\right) \\
&= m' \cdot e(h, h_1)^s \cdot \prod_{\rho(i) \in \mathcal{S}} \left(\frac{e(g^{r_i}, H_1(\rho(i))^s)}{e(h^{\mathbb{M}_i \vec{u}_i} H_1(\rho(i))^{r_i}, g^s)}\right) \\
&= m'.
\end{aligned}$$

It randomly chooses $A_i \in \mathcal{S}$ and recovers the message m'' as: Let the polynomial function $f(a)$ be

$$\begin{aligned}
f(a) &= \prod_{j=1}^{\ell} (a - H_2(A_i \| j \| b)) \cdot \left(\sum_{j \in \mathcal{L}} \frac{1}{a - H_2(A_i \| \tau \| b)}\right) \\
&= \frac{\prod_{j=1}^{\ell} (a - H_2(A_i \| j \| b))}{a - H_2(A_i \| \tau \| b)} + (a - H_2(A_i \| \tau \| b)) \cdot \left(\sum_{j=0}^{\ell-2} f_j a^j\right),
\end{aligned}$$

where f_j is the coefficient of a^j. The algorithm derives the message m'' by computing:

$$\begin{aligned}
& C_i^{(5)} \oplus H_3\left(e(C_i^{(3)}, K_i^{(2)}) \cdot e\left(C_i^{(4)}, \prod_{j=1}^{\ell-2} g_j^{f_j} \cdot g^{f_0}\right)^{-1}\right) \\
&= m'' \oplus H_3\left(e\left(g^{\frac{\prod_{j=1}^{\ell} \beta - H_2(A_i \| j \| b)}{\beta - H_2(A_i \| \tau \| b)}}, h\right)^{s'}\right) \oplus H_3\left(e\left(g^{\frac{\prod_{j=1}^{\ell} \beta - H_2(A_i \| j \| b)}{\beta - H_2(A_i \| \tau \| b)}}, h\right)^{s'}\right) \\
&= m''.
\end{aligned}$$

The returning message m is $m = m' \oplus m''$.

4.2 Security Proof

We prove the security of our ABSE in the selective security model based on the decisional BDH assumption and the modified q-BDHE assumption.

Theorem 2. *Suppose the hash functions H_1, H_2, H_3 are three random oracles. Let q_{H_1}, q_{H_2} and q_{H_3} be the query number to the oracle H_1, H_2 and H_3, respectively. Assuming the decisional BDH assumption is ϵ_{BDH}-hard, and the modified q-BDHE is $\epsilon_{\mathsf{q\text{-}BDHE}}$-hard, our ABSE scheme is $(q_{H_2}, q_{H_3}, \epsilon)$-secure under selective* IND-CPA *model under the ABSE setting. We have*

$$\mathbf{Adv}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{ABSE},\mathcal{A}} \leq 1/2 \cdot (\epsilon_{\mathsf{BDH}} + 1/(q_{H_2} q_{H_3}) \cdot \epsilon_{\mathsf{q\text{-}BDHE}}) .$$

Proof. Suppose there exist a probabilistic polynomial time adversary $\mathcal{A}$ that can break our ABSE scheme in the selective security model with a non-negligible advantage. We can build an algorithm $\mathcal{B}$ that can have a non-negligible advantage to break the decisional BDH problem or the modified q-BDHE problem.

Init. $\mathcal{B}$ runs $\mathcal{A}$. $\mathcal{A}$ chooses the challenge attribute set $\mathcal{S}^*$ and sends $\mathcal{S}^*$ to $\mathcal{B}$. $\mathcal{B}$ randomly chooses a bit $\hat{c} \in \{0, 1\}$.

If $\hat{c} = 0$, $\mathcal{B}$ is giving the terms $(A = g^a, B = g^b, C = g^c, Z)$ and the aim of $\mathcal{B}$ is to distinguish Z is $e(g, g)^{abc}$ or a random value.

If $\hat{c} = 1$, $\mathcal{B}$ is giving the terms $g, g^{(a)}, g^{(a^2)}, \ldots, g^{a^q}, g^{(a^{2q+2})}, g^{(a^{2q+3})}, \ldots, g^{(a^{3q+1})} \in \mathbb{G}^{2q+1}$ and the aim of $\mathcal{B}$ is to output $e(g, g)^{(a^{2q+1})}$.

Setup. $\mathcal{B}$ generates the public parameters pp to $\mathcal{A}$.

If $\hat{c} = 0$, $\mathcal{B}$ assigns the public parameter $h = B$ and $h_1 = A$ and chooses random value $\beta \in \mathbb{Z}_p$ to derive the rest of public elements:

$$h_2 = h^\beta, g_1 = g^{(\beta)}, g_2 = g^{(\beta^2)}, \ldots, g_n = g^{(\beta^n)}.$$

$\mathcal{B}$ then randomly chooses two collusion-resistant function $H_2 : \{0, 1\}^* \to \mathbb{Z}_p$ and $H_3 : \mathbb{G}_T \to \mathcal{M}$, and forwards the public parameter as:

$$pp = (p, \mathbb{G}, \mathbb{G}_T, e, g, g_1, \ldots, g_n, h, h_1, h_2, H_2, H_3)$$

except the hash function H_1 to $\mathcal{A}$, where H_1 works as a random oracle in the rest of reduction.

If $\hat{c} = 1$, $\mathcal{B}$ randomly picks a random value $\alpha \in \mathbb{Z}_p$ and sets $h_1 = g^\alpha$. Next, $\mathcal{B}$ randomly chooses $\{I_1, I_2, \ldots, I_{q_{H_2}}\}$ from $\mathbb{Z}_p$, and picks a random $i^* \in [q_{H_2}]$. Let $F(x) \in \mathbb{Z}_p[x]$ be a $(q_{H_2} - 1)$-degree polynomial function as:

$$F(x) = b \prod_{i=1, i \neq i^*}^{q_{H_2}} (x - I_i) = F_{q_{H_2}-1} x^{q_{H_2}-1} + \cdots + F_2 x^2 + F_1 x + F_0.$$

It sets $g_i = g^{(a^i)}$ for all $i \in [\ell]$ and computes $h = g^{F(a)}$ and $h_2 = g^{aF(a)}$ from the challenge input and $F(x)$. $\mathcal{B}$ then randomly chooses a collusion-resistant hash function $H_1 : \Omega \to \mathbb{G}$, and forwards the public parameter pp as

$$pp = (p, \mathbb{G}, \mathbb{G}_T, e, g, g_1, \ldots, g_n, h, h_1, h_2, H_1)$$

except the two hash functions to $\mathcal{A}$, and sets H_1 and H_2 as random oracles.

Hash Queries. If $\hat{c} = 0$, $\mathcal{A}$ can query the random oracle H_1 at any time; otherwise, $\mathcal{A}$ can query the random oracles H_2 and H_3 at any time.

$\mathcal{O}_{H_1}(\cdot)$. For any query on A to the random oracle H_1, $\mathcal{B}$ maintains a list $\mathcal{L}_{H_1}$ and responds as follows. If A is not in the list, the algorithm responds depended on $\mathcal{S}^*$. If $A \in \mathcal{S}^*$, the algorithm sets $r = 0$ and randomly picks $r' \in \mathbb{Z}_p$. If $A \notin \mathcal{S}^*$, the algorithm randomly chooses $r, r' \in \mathbb{Z}_p$. The algorithm returns $H_1(A) = h^r g^{r'}$ to $\mathcal{A}$, and adding (A, r, r') to $\mathcal{L}_{H_1}$. Otherwise, there has been already a tuple (A, r, r') in the list and the algorithm responds with $H_1(A) = h^r g^{r'}$.

$\mathcal{O}_{H_2}(\cdot)$. For any query on A to the random oracle H_2, $\mathcal{B}$ maintains a list $\mathcal{L}_{H_2}$ and responds as follows. If there has been already a tuple (A, I) in the list $\mathcal{L}_{H_2}$, the algorithm responds with $H_2(A) = I$. Otherwise, let A be the i^{th} distinct query. $\mathcal{B}$ responds by returning $H_2(A) = I_i$ to $\mathcal{A}$, and adding (A, I_i) to $\mathcal{L}_{H_2}$.

$\mathcal{O}_{H_3}$. For a random query on R to the random oracle H_3, $\mathcal{B}$ maintains a list $\mathcal{L}_{H_3}$ 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 $H_3(R) = Y$ to $\mathcal{A}$, and adding (R, Y) to $\mathcal{L}_{H_3}$. Otherwise, there has been already a tuple (R, Y) in the list and the algorithm responds with $H_2(R) = Y$.

Phase 1. $\mathcal{A}$ queries the key generation oracle $\mathcal{O}_{\mathcal{ABSE}.\mathsf{KeyGen}}(\cdot, \cdot, \cdot)$. For the query on access structure $\mathbb{A} = (\mathbb{M}, \rho)$, a bit b and a number ℓ from $\mathcal{A}$, $\mathcal{B}$ responds as: If $\hat{c} = 0$, according to the proposition 1 in [6], we have

$$\mathbb{M}_i \vec{u} = \vec{v} + \frac{ab - \vec{v}}{h} \cdot \vec{w} = \alpha\mu_1 + \mu_2,$$

where the coefficients $\mu_1 = \mathbb{M}_i \vec{w} \cdot h^{-1}$ and $\mu_2 = \mathbb{M}_i(h\vec{v} - \vec{v}\vec{w})$ are computable. For all $i \in [d]$, the algorithm fetches $(\rho(i), r, r')$ and computes the terms $K_i^{(0)}$ and $K_i^{(1)}$ as:

If $\rho(i) \in \mathcal{S}^*$, the algorithm randomly chooses $r_i \in \mathbb{Z}_p$ and sets

$$K_i^{(0)} = h^{\mathbb{M}_i \vec{u}_i} H_3(\rho(i))^{r_i}, \quad K_i^{(1)} = g^{r_i}, \quad K_i^{(2)} = h^{\sum_{j=1}^{\ell} \frac{1}{\beta - H_2(\rho(i)\|j\|b)}}.$$

If $\rho(i) \notin \mathcal{S}^*$, the algorithm randomly chooses $r_i' \in \mathbb{Z}_p$ and sets

$$K_i^{(0)} = h_1^{\frac{-\mu_1 \cdot r'}{r}} g_2^{\mu_2} H_3(\rho(i))^{r_i'}, \quad K_i^{(1)} = g^{r_i'} h_1^{\frac{-\mu_1}{r}}, \quad K_i^{(2)} = h^{\sum_{j=1}^{\ell} \frac{1}{\beta - H_2(\rho(i)\|j\|b)}}.$$

If $\hat{c} = 1$, it computes the terms $\{K_i^{(0)}, K_i^{(1)}\}_{i \in [d]}$ as our proposed scheme. For all $i \in [d]$, let the response for $\rho(i)\|j\|b$ in the list $\mathcal{L}_{H_2}$ be $(\rho(i)\|j\|b, I_j)$ for all $j \in [\ell]$. If $I_j = I^*$ holds for any $i \in [\ell]$, the algorithm aborts the simulation. When $I_j \neq I^*$ holds for $j \in [\ell]$, we have that $H_2(\rho(i)\|1\|b), H_2(\rho(i)\|2\|b), \ldots, H_2(\rho(i)\|\ell\|b)$ are all the roots of $F(x)$. Then, we deduce that

$$F_{\rho(i)}(x) = F(x) \cdot \left(\sum_{j=1}^{\ell} \frac{1}{x - H_2(\rho(i)\|j\|b)} \right)$$

is a $(q_{H_2}-2)$-degree at most polynomial function, and $\mathcal{B}$ can compute

$$K_i^{(2)} = h^{\sum_{j=1}^{\ell} \frac{1}{\beta - H_2(\rho(i)\|j\|b)}} = g^{F(\beta)\cdot\left(\sum_{j=1}^{\ell} \frac{1}{\beta - H_2(\rho(i)\|j\|b)}\right)} = g^{F_{\rho(i)}(a)}$$

from $F_{\rho(i)}(x)$ and $g, g^{(a)}, \ldots, g^{(a^q)}$, and $K_i^{(2)}$ is a valid secret key component.

Finally, $\mathcal{B}$ returns the secret key $sk_{\mathbb{A}} = \{K_i^{(0)}, K_i^{(1)}, K_i^{(2)}\}_{i\in[d]}$ to $\mathcal{A}$.

Challenge. $\mathcal{A}$ will submit two challenge message (m_0, m_1, b, τ) to $\mathcal{B}$. $\mathcal{B}$ flips a fair binary coin $\bar{c}$.

If $\hat{c} = 0$, $\mathcal{B}$ randomly chooses $m' \in \mathbb{M}$ and sets $m'' = m_{\bar{c}} \oplus m'$, then computes the terms $C^{(0)}, C^{(1)}, C^{(2)}$ as:

$$C = m' \cdot Z, \quad C^{(1)} = C, \quad \forall A_i \in \mathcal{S}^* : C_i^{(2)} = C^{r'}.$$

If $Z = e(g,g)^{abc}$. Then the ciphertext is:

$$C = m' \cdot e(g,g)^{abc}, \quad C^{(1)} = g^c, \quad \forall A_i \in \mathcal{S}^* : C_i^{(2)} = H_3(A_i)^c.$$

The rest of ciphertext $(\tau, \{C_i^{(3)}, C_i^{(4)}, C_i^{(5)}\}_{A_i\in\mathcal{S}^*})$ are generated as our proposed scheme.

If $\hat{c} = 1$, $\mathcal{B}$ randomly chooses $m' \in \mathbb{M}$ and sets $m'' = m_{\bar{c}} \oplus m'$, then computes the terms $C^{(0)}, C^{(1)}, C_i^{(2)}$ as our proposed scheme. For each attribute $\rho(i)$ in $\mathcal{S}$, $\mathcal{B}$ works as follows:

If $\mathcal{B}$ cannot find the tuple $(\rho(i)\|\tau\|b, I^*) \in \mathcal{L}_{H_2}$ satisfies $I^* \neq I_\tau$, abort; otherwise, the algorithm randomly chooses $C_i^{(5)*} \in \{0,1\}^\ell$. Let

$$F'(x) = \frac{\sum_{j=1}^{\ell} \frac{1}{\beta - H_2(\rho(i)\|j\|b)}}{x - I^*}$$

be an $(n-1)$-degree polynomial function. The algorithm randomly chooses $r' \in \mathbb{Z}_p$ and computes the challenge ciphertext $(C_i^{(3)}, C_i^{(4)}, C_i^{(5)})$ by

$$C_i^{(3)} = g^{r'(a^{2q+2} - I^{*2q+2})F'(a)}, \quad C_i^{(4)} = g^{r'(a^{2q+2} - I^{*2q+2})F(a)}, \quad C_i^{(5)} = C_i^{(5)*}.$$

where $C_i^{(3)}$ and $C_i^{(4)}$ are computable from $F'(x)$ and $F(x)$ 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 the challenge ciphertext is equivalent to

$$C_i^{(3)} = \left(g^{\sum_{j=1}^{\ell} \frac{1}{\beta - H_2(\rho(i)\|j\|b)}}\right)^r, \quad C_i^{(4)} = \left(h^{\beta - H_2(\rho(i)\|\tau\|b)}\right)^r, \quad C_i^{(5)} = C_i^{(5)*}.$$

According to our setting, there must exist a hash query on $e\left(g^{\frac{\sum_{j=1}^{\ell} \frac{1}{\beta - H_2(\rho(i)\|j\|b)}}{\beta - H_2(\rho(i)\|\tau\|b)}}, h\right)^r$ to the random oracle H_3 in order to decrypt the

message in the challenge ciphertext.

$$m'' = H_3\left(e\left(g^{\frac{\sum_{j=1}^{\ell}\frac{1}{\beta - H_2(\rho(i)\|j\|b)}}{\beta - H_2(\rho(i)\|\tau\|b)}}, h\right)^r\right) \cdot C_i^{(5)*}.$$

Phase 2. Phase 2 is same as Phase 1.

Guess. If $\hat{c} = 0$, $\mathcal{A}$ will submit a guess $\bar{c}'$. If $\bar{c} = \bar{c}'$, $\mathcal{B}$ will output 0 to indicate that is was given a valid BDH-tuple otherwise it will output 1 to indicate it was given a random 4-tuple.

If $\hat{c} = 1$, $\mathcal{A}$ returns a guess $\bar{c}'$. 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 $e\left(g^{\frac{\sum_{j=1}^{\ell}\frac{1}{\beta - H_2(\rho(i)\|j\|b)}}{\beta - H_2(\rho(i)\|\tau\|b)}}, h\right)^r$ $= e(g, g)^{F''(a)}$. It is easy to verify that F''_{2q+1} is equal to $r'F'(I^*)F(I^*)$ which is non-zero, and that $e(g, g)^{F''\cdot a^i}$ for all $i \neq 2q + 1$ are computable from the challenge input. $\mathcal{B}$ picks a random tuple (R, Y) from the list $\mathcal{L}_{H_3}$ and computes

$$\left(R \cdot \prod_{i=1, i\neq 2q+1}^{2q+n+q_{H_2}-1} e(g, g)^{-F_i''\cdot a^i}\right)^{\frac{1}{r'F'(I^*)F(I^*)}} = e(g, g)^{a^{2q+1}}$$

as the solution to the modified q-BDHE problem.

When $\hat{c} = 0$, if $\mathcal{B}$ output 0 ($\bar{c} = \bar{c}'$), the generation of public parameters and secret keys is identical to that of the actual scheme. In the case where outputs 1 ($\bar{c} \neq \bar{c}'$), $\mathcal{A}$ gains no information about $\bar{c}$. Therefore, the probability of guessing successful is 1/2. In the case where outputs 0, $\mathcal{A}$ sees an encryption of $m_{\bar{c}}$. The advantage in this situation is ϵ_{BDH} by definition. Hence, the advantage is ϵ_{BDH}.

When $\hat{c} = 1$, we need to consider of three events. The first event is $\mathcal{B}$ can generate i^{th} key generation query on the challenge attribute. The second event is $\mathcal{B}$ does not abort in the challenge phase. Hence, we have the overall abort in the guess phase $1/q_{H_2}$. The last one is $\mathcal{B}$ may not query $e(g, g)^{F''(a)}$ to the random oracle $\mathcal{O}_{H_3}$, and the probability of choosing a correct randomness R_i is $1/q_{H_3}$. The advantage in this situation is $(1/(q_{H_2}q_{H_3}))\epsilon_{\mathsf{q\text{-}BDHE}}$. Therefore, the advantage of $\mathcal{A}$ breaking the game is $(1/(q_{H_2}q_{H_3}))\epsilon_{\mathsf{q\text{-}BDHE}}$

We have the advantage when $\hat{c} = 0$ and $\hat{c} = 1$, respectively. Hence, the above probability analysis does not consider $\mathcal{B}$ guessing $\hat{c}$ correctly, and the probability of $\mathcal{B}$ guessing $\hat{c}$ successful is 1/2. Therefore, the overall advantage is

$$\mathbf{Adv}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{ABSE},\mathcal{A}} = 1/2 \cdot (\epsilon_{\mathsf{BDH}} + 1/(q_{H_2}q_{H_3}) \cdot \epsilon_{\mathsf{q\text{-}BDHE}}).$$

5 Conclusion

We introduced the attribute-based traitor tracing scheme based on the fingerprint code in blackbox setting. The size of the secret key relates to the size of policies as the normal attribute-based encryption scheme rather than the previous blackbox attribute-based traitor tracing schemes depend on both the number of the user in the system and the size of policies. It saves both secure storage and bandwidth for ABTT applications. We also introduced a new primitive of attribute-based set encryption for reducing the multi-attribute scenarios. Our proposed ABSE scheme is provably secure in the random oracle under the decisional BDH assumption and the modified q-BDHE assumption.

References

1. Attrapadung, N., Herranz, J., Laguillaumie, F., Libert, B., de Panafieu, E., Ràfols, C.: Attribute-based encryption schemes with constant-size ciphertexts. Theor. Comput. Sci. **422**, 15–38 (2012)
2. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334 (2007)
3. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
4. Boneh, D., Naor, M.: Traitor tracing with constant size ciphertext. In: CCS, pp. 501–510 (2008)
5. 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). https://doi.org/10.1007/11761679_34
6. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: CCS, pp. 89–98 (2006)
7. Guo, F., Mu, Y., Susilo, W.: Identity-based traitor tracing with short private key and short ciphertext. In: Foresti, S., Yung, M., Martinelli, F. (eds.) ESORICS 2012. LNCS, vol. 7459, pp. 609–626. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33167-1_35
8. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_4
9. Liu, Z., Cao, Z., Wong, D.S.: Blackbox traceable CP-ABE: how to catch people leaking their keys by selling decryption devices on eBay. In: Sadeghi, A.-R., Gligor, V.D., Yung, M. (eds.) 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS 2013, Berlin, Germany, 4–8 November 2013, pp. 475–486. ACM (2013)
10. Liu, Z., Cao, Z., Wong, D.S.: White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures. IEEE Trans. Inf. Forensics Secur. **8**(1), 76–88 (2013)
11. Liu, Z., Cao, Z., Wong, D.S.: Traceable CP-ABE: how to trace decryption devices found in the wild. IEEE Trans. Inf. Forensics Secur. **10**(1), 55–68 (2015)

12. Liu, Z., Wong, D.S.: Practical attribute-based encryption: traitor tracing, revocation and large universe. Comput. J. **59**(7), 983–1004 (2016)
13. Ning, J., Dong, X., Cao, Z., Wei, L.: Accountable authority ciphertext-policy attribute-based encryption with white-box traceability and public auditing in the cloud. In: Pernul, G., Ryan, P.Y.A., Weippl, E. (eds.) ESORICS 2015. LNCS, vol. 9327, pp. 270–289. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24177-7_14
14. Ning, J., Dong, X., Cao, Z., Wei, L., Lin, X.: White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes. IEEE Trans. Inf. Forensics Secur. **10**(6), 1274–1288 (2015)
15. Rouselakis, Y., Waters, B.: Practical constructions and new proof methods for large universe attribute-based encryption. In: CCS, pp. 463–474 (2013)
16. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_27
17. 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). https://doi.org/10.1007/978-3-642-03356-8_36
18. Yu, G., Cao, Z., Zeng, G., Han, W.: Accountable ciphertext-policy attribute-based encryption scheme supporting public verifiability and nonrepudiation. In: Chen, L., Han, J. (eds.) ProvSec 2016. LNCS, vol. 10005, pp. 3–18. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47422-9_1
19. Yu, G., Ma, X., Cao, Z., Zhu, W., Zeng, J.: Accountable multi-authority ciphertext-policy attribute-based encryption without key escrow and key abuse. In: CSS, pp. 337–351 (2017)
20. Zhang, Y., Li, J., Zheng, D., Chen, X., Li, H.: Accountable large-universe attribute-based encryption supporting any monotone access structures. In: Liu, J.K.K., Steinfeld, R. (eds.) ACISP 2016. LNCS, vol. 9722, pp. 509–524. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40253-6_31