A Traceable and Revocable Ciphertext-Policy Attribute-based Encryption Scheme Based on Privacy Protection

Dezhi Han, *Member, IEEE*, Nannan Pan, and Kuan-Ching Li, *Senior Member, IEEE*

Abstract—Considered as a promising fine-grained access control mechanism for data sharing without a centralized trusted third-party, the access policy in a plaintext form may reveal sensitive information in the traditional CP-ABE method. To address this issue, a hidden policy needs to be applied to the CP-ABE scheme, as the identity of a user cannot be accurately confirmed when the decryption key is leaked, so the malicious user is traced and revoked as demanded. In this article, a CP-ABE scheme that realizes revocation, white-box traceability, and the application of hidden policy is proposed, and such ciphertext is composed of two parts. One is related to the access policy encrypted by the attribute value, and only the attribute name is evident in the access policy. Another is related to the revocation information and updated when revoking, where the revocation information is generated by the binary tree related to users. The leaf node value of a binary tree in the decryption key is used to trace the malicious user. From experimental results, it is shown that the proposed scheme is proven to be IND-CPA secure under the chosen plaintext attacks and selective access policy based on the decisional q-BDHE assumption in the standard model, efficient, and promising.

Index Terms—Binary tree, CP-ABE, hidden policy, revocation, traceability

1 Introduction

With the development of cloud computing technology, cloud storage systems provide users data storage and web services conveniently, tend to offer flexibility, scalability, and pay-as-you-go pricing. Seagate predicts that data creation will grow to an enormous 163 zettabytes (ZB) worldwide by 2025 [1], so thus, an increasing amount of sensitive data that needs to be encrypted is outsourced into the cloud. Many encryption schemes [2], [3], [4], [5] have been proposed. However, encrypted data is always shared as a "one-to-many" mode, so messages with fine-grained access control need to be encrypted.

Sahai and Waters [6] proposed Attribute-Based Encryption (ABE) that provides an encryption scheme with fine-grained access control. Types of attribute-based encryption are twofold, being the former one key-policy attribute-based encryption (KP-ABE) that is proposed by Goyal *et al.* [7], while the latter is ciphertext attribute-based encryption (CP-ABE), as proposed by Bethencourt *et al.* [8]. In CP-ABE, the ciphertext is associated with the access policy, and the decryption key is related to attributes, whereas the ciphertext of KP-ABE is associated with attributes and decryption key bounded to the access policy. ABE achieves rich attribute operations and supports more flexible access policies, so several ABE schemes are proposed for better expressivity, efficiency, and security [9], [10], [11], [12], [13].

In recent years, the access policy in traditional ABE schemes proposed is stored in the cloud server together with the ciphertext, so relevant access policy is available to anyone who can retrieve the ciphertext. Nevertheless, the access policy may also contain sensitive information. For example, a medical record is revealed by the access policy "neurology AND (doctor OR nurse)". The patient with a medical record may also have a neurological problem, so it may be necessary to hide such a policy. Hidden policies are classified in partially hidden policy and fully hidden policy, where the attributes are always split into attribute names and attribute values, as only the attribute names are in clear text without sensitive information in the former one, while none of the attributes are related in the latter.

However, there may have malicious users who leak the decryption key to third parties in ABE systems. Since the decryption key is associated with the attribute, users who leaked the decryption key cannot be determined. For example, Alice and Emily have attributes {neuropathy AND Nurses}, and they can access to the medical records encrypted by "neurology AND (doctor OR nurse)". Though, if the decryption key is leaked, it is not known whether Alice or Emily did it. To solve the problem of decryption key leakage and trace the malicious users, the CP-ABE scheme [14], [15], [16] is proposed. There are white-box traceability and black-box traceability, in which white-box traceability can trace the malicious user by a well-formed decryption key that binds the user's identity. And comparing to white-box traceability, black-box traceability is given a decryption device, while the

- *D. Han and N. Pan are with the College of Information Engineering, Shanghai Maritime University, Pudong 201306, Shanghai, China. E-mail: dzhan@shmtu.edu.cn, nnp215@foxmail.com.*
- *K.-C. Li is with the Department of Computer Science and Information Engineering (CSIE), Providence University, Taichung 43301, Taiwan. E-mail: kuancli@pu.edu.tw.*

Manuscript received 24 June 2019; revised 21 Feb. 2020; accepted 26 Feb. 2020. Date of publication 2 Mar. 2020; date of current version 17 Jan. 2022. (Corresponding author: Kuan-Ching Li.)
Digital Object Identifier no. 10.1109/TDSC.2020.2977646

1545-5971 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Tsinghua University. Downloaded on September 27,2022 at 03:46:20 UTC from IEEE Xplore. Restrictions apply.

decryption key and even the decryption algorithm could be hidden in the device. In black-box traceability, the tracing algorithm is able to trace the malicious user of the decryption device built from user's decryption key. As any user is traced, he cannot be revoked from the system, and thus, it is highly demanded a revocation mechanism based on traceability.

The revocation takes two forms: direct revocation [17], [18], [19] and indirect revocation [20], [21], [22]. In direct revocation ABE systems, data owners directly specify the revocation list when encrypting the ciphertext and only update the revocation list when revoking. It is unnecessary to communicate with the authority in the direct revocation. On the other hand, the authority needs to update the information and communicate with users in the indirect revocation, which will result in more cost more if the number of users is larger.

1.1 Related Work

With the emergence of identity-based encryption, Sahai and Waters constitute an alternative method, namely attribute-based encryption (ABE) [6]. Later, several KP-ABE [23], [24] and CP-ABE [25], [26] schemes have been proposed to enhance further and improve the performance of ABE.

To trace malicious users, Liu proposed a white-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures [27]. Ning proposed a white-box traceable ciphertext-policy attribute-based encryption [28], that supporting flexible attributes. He also proposed a large universe ciphertext-policy attribute-based encryption with white-Box [29], that can support the tracing of malicious users. In [30], Ning *et al.* also designed a white-box traceable CP-ABE scheme able to trace the users who maliciously leak their decryption key effectively. For such, the revocation mechanism was proposed to revoke the illegal and invalidate users in the system. The CP-ABE scheme with user revocation was proposed by Yasumura [31] to reduce the communication cost of proxy re-encryption.

Even though some ABE schemes support a traceability mechanism, the user cannot be revoked after tracing. Liu [32] proposed a traceable-then-revocable ciphertext-policy attribute-based encryption scheme, where the ciphertext and decryption key of the scheme contains two parts, in which one is associated with the revocation list. However, it cannot resist to anti-collusion attacks. The CP-ABE scheme proposed by Wang *et al.* [33] utilizes a binary tree associated with user information to implement attribute revocation and user tracing. This scheme proved to be safe under the chosen plaintext attack and selective access policy. Lian *et al.* [34] proposed a fully secure traceable and revocable-storage ABE scheme that only needs to update part of the key after revocation.

Although variations of ABE schemes that can trace yet revoke are proposed, the access policy contains sensitive information that would still expose users' privacy. Song *et al.* [35] proposed attribute-based encryption with hidden policies in the access tree. It needs to verify the user's attributes. However, from the verification system, the attacker can directly obtain the decrypted content without performing attribute matching, which results in lower security. Xu *et al.* [36] proposed a tree-based CP-ABE scheme with a hidden policy that marks each node in the access tree through a threshold secret sharing scheme. The hidden policy CP-ABE schemes proposed by Lai *et al.* [37] and Zhang *et al.* [38] indicate the attributes as the attribute name and the attribute value, and the access policy related to the ciphertext, only includes the attribute name, realizing thus partially hidden policy. The hidden policy CP-ABE proposed by Sun *et al.* [39] applies And-Gates to represent the access policy. As each attribute can have multiple values, the access policy is still relatively simple. Fan *et al.* [40] proposed a CP-ABE scheme with the hidden policy in multi-authority that partitions secret by threshold secret sharing scheme with And-Or-Gates access policy.

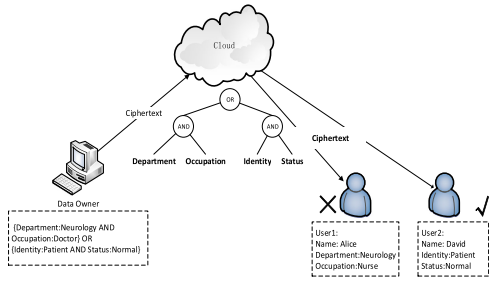

Fig. 1. Example of patially hidden policy.

All the above schemes, they only realize one or two among the hidden policy, traceability, and revocation. In the schemes of traceability and revocation, there exists the risk of privacy leaked, while the scheme with the hidden policy cannot trace and revoke malicious users. Thereafter, it is proposed in this paper a traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection, implemented the traceability and revocation with partially hidden policy.

1.2 Our Contribution

In this paper, a traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection (TR-AP-CPABE) is proposed, which realizes the partially hidden policy to protect privacy as well as the tracking and revoking of malicious users. The main contributions of this paper are as follows:

1) Partially hidden policy: the linear secret sharing scheme (LSSS) in [38] is used to represent the access policy. As shown in Fig. 1, the attribute is divided into the attribute name and the attribute value, that is used for encrypting, and the access policy related to the ciphertext only contains the attribute name. The user cannot know the specific attribute value, so the policy is partially hidden,
2) Traceability: the white-box traceability is used, which can trace the malicious user by given a well-formed decryption key bound user's identity. The user's identity information is denoted by the leaf node value of the binary tree that is created by all users in the system. Then the leaf node value is encrypted and contained in the decryption key. In the case the decryption key is leaked, the user's identity can be revealed from the decryption key. Compared with scheme [28], the user information can be encrypted directly without initializing user information list,

Authorized licensed use limited to: Tsinghua University. Downloaded on September 27,2022 at 03:46:20 UTC from IEEE Xplore. Restrictions apply.

3) Revocation: the binary tree is created by all users, in which the leaf nodes are associated with user information. The revocation information is derived from the binary tree and revocation list and then encrypted during the encryption stage which is an independent part of the ciphertext. After tracing the user, the user revocation list is updated and only the ciphertext associated with the revocation information needs to be updated. Compared to key updating, the ciphertext only needs to be updated once while the keys of all unrevoked users need to be updated which will cause lots of overhead in key updating.
4) Security: the TR-AP-CPABE scheme is proven to be IND-CPA secure and efficient based on the decisional q-BDHE assumption, which means to be indistinguishable under chosen-plaintext attacks and selective access policy based on the decisional q-BDHE assumption in the standard model.

1.3 Organization

The remaining of this paper is organized as follows. Section 2 introduces the background knowledge, to be applied to the proposed scheme, the scheme model is formally defined and the security model is well described in Section 3. Details of the TR-AP-CPABE scheme are depicted in Section 4 and proof in Section 5. In Section 6, experimental analysis and theoretical analysis between the proposed scheme and related works are provided, and finally, concluding remarks and directions for future work are summarized in Section 7.

2 Preliminaries

This section describes the background information for this proposed research, including access structures, secret sharing schemes, binary trees, and assumptions.

2.1 Bilinear Maps

Let G and G_T be two multiplication cyclic groups of prime order$p$, and $g$is a generator of G, A map $e : G \times G \rightarrow G_T$ is a bilinear map [41] and has the following properties:

- Bilinearity: for $\forall P, Q \in G$ and $a, b \in Z_p^*$, $e(P^a, Q^b) = e(P, Q)^{ab}$,
- Non-degeneracy: $e(g, g) \neq 1$,
- Computability: for $\forall P, Q \in G$, there is an efficient algorithm to compute $e(P,Q)$

2.2 Access Structure

Let $P = \{P_1, P_2, \ldots, P_n\}$ be a set of parties, a collection $\mathbb{A} \subseteq 2^P$ is monotone for $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. A monotone access structure [38] is a monotone collection $\mathbb{A}$ of non-empty subsets of P, i.e., $\mathbb{A} \subseteq 2^{\{P_1, P_2, \ldots, P_n\}} \backslash \{\emptyset\}$. The sets in $\mathbb{A}$ are called authorized sets; otherwise, the sets are called unauthorized sets.

2.3 Linear Secret Sharing Scheme

Let $A = \{A_1, A_2, \ldots, A_n\}$ be the attribute name universe, and for $\forall A_i \in A$, the set of the attribute value is $A_i = \{a_{i,1}, a_{i,2}, \ldots, a_{i,n_i}\}$, where n_i is the order of A_i. A Linear Secret Sharing Scheme (LSSS) is used to represent an access policy [38] by (M, ρ), where M is a matrix $l \times n$ and ρ maps a row of M into an attribute name in A, consisting of two algorithms:

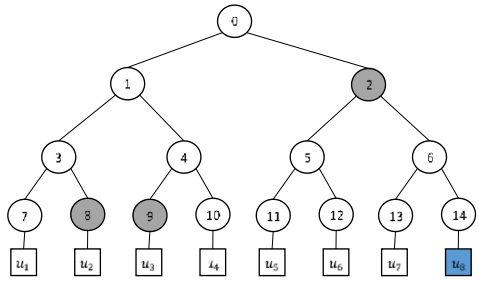

Fig. 2. Binary tree T.

- $Share((M, \rho), s)$: This algorithm is used to share a secret value s based on A. Consider a vector $v = (s, v_2, \ldots, v_n)^T$, where $s \in Z_p$ is the secret of being shared and $v_2, \ldots, v_n \in Z_p$ is randomly chosen, then $\lambda_i = M_i \cdot v$ is a share of the secret s corresponding to the attribute name by $\rho(i)$, where M_i is the *i-th* row of M.
- $Reconstruction(\lambda_i, \ldots, \lambda_l, (M, \rho))$: This algorithm is used to reconstruct s from secret shares. Let $S \in \mathbb{A}$ be any authorized set and $I = \{i : \rho(i) \in S\} \subseteq \{1, 2, \ldots, l\}$. Then there exist coefficients $\{c_i | i \in I\}$ such that $\sum_{i \in I} c_i M_i = (1, 0, \ldots, 0)$, and thus, we have $\sum_{i \in I} c_i \lambda_i = s$.

Let $\mathcal{S} = (I_S, S)$ be a set of user attribute, where $I_S \subseteq A$ is the set of user attribute name and $S = \{s_i\}_{i \in I_S}$ is the set of the user attribute value. We denote access policy by $W = (M, \rho, \mathcal{T})$, ρ maps a row of M into an attribute name in A, and each attribute name can only appear once, $\mathcal{T} = \{t_{p(i)}\}_{i \in [1,l]}$ is the attribute value associated with (M, ρ). If there exists an $I = \{i : \rho(i) \in S\} \subseteq \{1, 2, \ldots, l\}$ satisfying (M, ρ) and $s_{p(i)} = t_{p(i)}$ for $\forall i \in I$, we say that $\mathcal{S}$ matches W and record as $\mathcal{S} \models W$; otherwise, $\mathcal{S}$ does not match W and record as $\mathcal{S} \not\models W$. Let $\overline{W} = (M, \rho)$ be an incomplete access policy that removes the attribute value set.

2.4 Binary Tree

Let U be a set of users in the system, R be the revocation list. The binary tree T [32] is depicted as:

- The leaf node of the binary tree is associated with a user u. Let $|U|$ be the number of users, and the number of nodes in the tree is $2|U| - 1$. Nodes are numbered by the breadth-first search. For instance, the root is 0, and the last node is $2|U| - 2$,
- $path(i)$ is defined as a path from the root to the node i,
- The minimum cover set $cover(R)$ is a minimum set of nodes that can cover all users not listed in the revocation list R,
- According to $cover(R) \cap path(u)$, a user u who is not in the revocation list, there is only one node $j = cover(R) \cap path(u)$;

From the binary tree shown in Fig. 2, the revocation list is given as $R = \{u_1, u_4\} = \{7, 10\}$, so $cover(R) = \{2, 8, 9\}$. In the tree, the path of u_8: $path(u_8) = path(14) = \{0, 2, 6, 14\}$ is known. Thus, the only node is $j = cover(R) \cap path(u_8) = \{2\}$.

Authorized licensed use limited to: Tsinghua University. Downloaded on September 27,2022 at 03:46:20 UTC from IEEE Xplore. Restrictions apply.

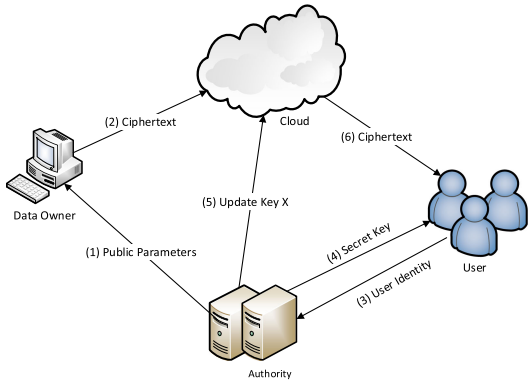

Fig. 3. System architecture of TR-AP-CPABE.

The features of the binary tree is used to implement the revocation.

2.5 Complexity Assumptions

In this section, the q-Bilinear Diffie-Hellman Exponent (q-BDHE) assumption and the l-Strong Diffie-Hellman (l-SDH) assumption are described, and they are used to prove the security of the proposed scheme.

Definition 1 (q-BDHE assumption [33]). *A decisional q-BDHE hardness assumption can be described as follows: let* G *and* G_T *be two multiplication cyclic groups of prime order* p*, and* g *is a generator of* G*, A map* $e : G \times G \rightarrow G_T$ *is a bilinear map. Randomly choose* $s, \alpha \in Z_p^*$ *and give* $Y = (g, g^s, g^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^q}, g^{\alpha^{q+2}}, \ldots, g^{\alpha^{2q}})$*. The q-BDHE hardness assumption holds if no polynomial-time algorithm can distinguish the* $e(g,g)^{\alpha^{q+1} \cdot s}$ *from a random element in* G_T *with non-negligible advantage* ε*.*

Definition 2 (l-SDH assumption [28]). *The l-Strong Diffie-Hellman assumption can be described as follows: let* G *and* G_T *be two multiplication cyclic groups of prime order* p *and* g *is a generator of* G*, A map* $e : G \times G \rightarrow G_T$ *is a bilinear map. Randomly choose* $x \in Z_p^*$ *and give a* $l+1$ *tuple* $(g, g^x, g^{x^2}, \ldots, g^{x^l})$*. The l-SDH hardness assumption holds if no polynomial-time algorithm can compute the* $(c, g^{1/(x+c)})$ *with non-negligible advantage* ε*.*

3 System and Security Models

In this section, the system architecture of TR-AP-CPABE is defined, and followed by the security models of TR-AP-CPABE

3.1 System Architecture

The system architecture of TR-AP-CPABE with four entities included is depicted in Fig. 3 as:

- Data owner: Data owner specifies an access policy and encrypts the message according to the access policy, then sends the ciphertext to the cloud storage,
- Cloud: Cloud stores the ciphertext and updates it by using the updated key sent from the authority through the secret channel,
- Authority: The authority publishes the public parameter and retains the system master key, also generates a decryption key for the user. It is fully trusted and realizes the traceability and revocation,
- User: The user always accesses the encrypted data, he can decrypt the plaintext if and only if his attributes satisfy the access policy and not in the revocation list.

3.2 Formal Definition of TR-AP-CPABE

The TR-AP-CPABE scheme mainly consists of the following seven algorithms and formally defined as:

- $Setup(\lambda, T) \rightarrow (PP, MSK)$: the authority runs the algorithm and inputs a binary tree T and a security parameter λ. It outputs the public parameter PP that will be published and system master key MSK, retained by the authority. The authority also maintains the revocation list R,
- $Encrypt(PP, m, W, R) \rightarrow CT$: the data owner runs the algorithm and inputs the public parameter PP, an access policy W described in Section 2.3, a message m, and the revocation list R. The encryption algorithm outputs a ciphertext CT of m, then sends CT with the incomplete access policy $\overline{W}$ to the cloud. The specific attribute values cannot be known,
- $KeyGen(MSK, \mathcal{S}, u) \rightarrow SK$: the authority runs the algorithm and inputs the user identity u and an attribute set $\mathcal{S} = (I_S, S)$ as well as a master key MSK, then generates a decryption key SK of u and $\mathcal{S} = (I_S, S)$, then sends it to the user,
- $Decrypt(SK, CT) \rightarrow m$: the user inputs his/her decryption key SK and a ciphertext CT with an incomplete access policy $\overline{W}$. If and only if his attributes satisfy the access policy and he is not in the revocation list, he can decrypt and recover the message m,
- $KeySanityCheck(PP, SK) \rightarrow 1$ or $\perp$: the authority runs the algorithm, inputs the public parameter PP, and a user's decryption key SK. Next, the algorithm checks whether the decryption key SK is required to trace. If it passes the key sanity check, then the algorithm outputs 1; otherwise, the algorithm outputs $\perp$,
- $Trace(PP, R, SK) \rightarrow u$ or $\perp$: the authority runs the algorithm. First, the authority runs $KeySanityCheck$ with the public parameter PP and a user's decryption key SK. If SK passes the $KeySanityCheck$, then the algorithm outputs the user's identity u and updates the user revocation list $R' = R \cup \{u\}$; otherwise, the algorithm terminates and outputs $\perp$. The algorithm is performed privately by the authority, so the privacy of malicious users could not be leaked.
- $CTUpdate(CT, R', X') \rightarrow CT'$: The ciphertext updated algorithm is the same as the one presented in [32], called by the cloud. The algorithm inputs the ciphertext CT, updated revocation list R', the updated key sent by the authority through the secret channel, and outputs the updated ciphertext and keeps it in the cloud.

3.3 Traceability Security Model

The traceability model [28] of TR-AP-CPABE can be described by a security game between a challenger B and an adversary $\mathrm{A}dv$. The specific steps are as follows:

Initialization. The challenger B executes the $Setup$ algorithm and sends the public parameter PP to the adversary $\mathrm{A}dv$;

Key Query. $\mathrm{A}dv$ asks B about the decryption keys of attribute sets $(u_1, \mathcal{S}_1)(u_2, \mathcal{S}_2) \ldots (u_q, \mathcal{S}_q)$, which u_i is in the revocation list

Authorized licensed use limited to: Tsinghua University. Downloaded on September 27,2022 at 03:46:20 UTC from IEEE Xplore. Restrictions apply.

R or $\mathcal{S}_i$ does not satisfy the access policy W, where $i = 1, 2, \ldots, q$. For each i,B executes the $KeyGen$ algorithm and sends the decryption key SK_i to Adv;

Key Forgery. Adv outputs a decryption key $SK_{\mathrm{A}dv}$. The adversary Adv wins the game if $Trace(PP, R, SK_{\mathrm{A}dv}) \neq \perp$ and $Trace(PP, R, SK_{\mathrm{A}dv}) \notin \{u_1, u_2, \ldots, u_q\}$.

The advantage of Adv winning the game is defined as:

$$\varepsilon = \Pr\left[Trace(PP, R, SK_{\mathrm{A}dv}) \notin \left\{\perp, u_1, u_2, \ldots, u_q\right\}\right]$$

Definition 3. *A TR-AP-CPABE scheme is traceable if all polynomial-time adversaries have at most negligible advantage in the game.*

3.4 IND-CPA Security Model

The IND-CPA security model [33] of TR-AP-CPABE can be described by a security game between a challenger B and an adversary Adv. The specific steps are as follows:

Init. The adversary Adv selects an access policy $W = (M^*, \rho^*, \mathcal{T})$ and a revocation list R^*, where M^* is a $l^* \times n^*$ matrix and ρ^* maps a row of M^* into an attribute name in A in which each attribute name can only appear once. $\mathcal{T} = \{t_{\rho^*(i)}\}_{i\in[1,l^*]}$ is the attribute value associated with (M^*, ρ^*);

Setup. The challenger B executes the $Setup$ algorithm and sends the public parameter PP to Adv;

Phase1. Adv asks B about the decryption keys of attribute sets $(u_1, \mathcal{S}_1)(u_2, \mathcal{S}_2) \ldots (u_q, \mathcal{S}_q)$.

- If $\mathcal{S} \models W$, which means that $\mathcal{S}$ matches W, and $u \notin R^*$, then abortion.
- If $\mathcal{S} \not\models W$, which means that $\mathcal{S}$ does not match W, or $u \in R^*$, B generates a decryption key of $(u_i, \mathcal{S}_i)$ for $i = 1, 2, \ldots, q$ and returns it to Adv.

Challenge. Adv submits two equal length messages m_0 and m_1 to B that randomly picks a message $m_b (b \in \{0, 1\})$, encrypt it under the access policy $W = (M^*, \rho^*, \mathcal{T})$ and the revocation list R^*. Lastly, B sends the challenge ciphertext CT_b to Adv.

Phase2. Phase2 is the same as Phase1.

Guess. Adv outputs a guess b' of b. If $b = b'$, then the adversary Adv wins the game.

The advantage of Adv winning the game is defined as: $\varepsilon = |\Pr[b = b'] - 1/2|$.

Definition 4. *A TR-AP-CPABE scheme is indistinguishable from chosen-plaintext attacks under a selective access policy if all polynomial-time adversaries have at most negligible advantage in the game.*

4 Construction of TR-AP-CPABE

In this section, the components and specific steps of the TR-AP-CPABE scheme are described. As the design of the proposed scheme, the access control policy in [38] is adopted, given that the ciphertext does not contain specific attribute values to realize the privacy protection. A binary tree is used to realize traceability and revocation. The ciphertext consists of two parts. One is associated with the revocation list that only needs to update this part of ciphertext when revoking. Another is associated with the access policy. If and only if his/her attributes satisfy the access policy and he/she is not in the revocation list, he/she can decrypt and recover the message.

4.1 Setup

The algorithm is initialized and called by the authority, which generates public parameters and system master key. In this algorithm, the security parameter λ, the system attribute universe A and a binary tree T generated by all users are the input. Let G and G_T are two multiplication cyclic groups of prime order p, g is a generator of G, and A map $e : G \times G \rightarrow G_T$ is a bilinear map. T is a binary tree associated with u in a user set U. The algorithm is executed as follows:

1. Randomly pick $a, \alpha \in Z_p, h, u \in G$,
2. For each node in T, randomly select $\{x_i\}_{i=0}^{2|U|-2} \in Z_p$ and compute $\{y_i = g^{x_i}\}_{i=0}^{2|U|-2}$,
3. A probabilistic encryption scheme (Enc, Dec) [42] is selected, which is symmetric encryption from $\{0,1\}^*$ to Z_p with secret key $k \in Z_p$ and encrypts the same message to yield different ciphertext each time.

The public parameter is published as: $PP = \langle p, G, G_T, e, g, h, u, e(g,g)^{\alpha}, g^a, \{y_i\}_{i=0}^{2|U|-2}\rangle$, and the master key is kept as: $MSK = \langle a, \alpha, \{x_i\}_{i=0}^{2|U|-2}, k\rangle$.

4.2 Encrypt

The data owner executes the algorithm and will give the ciphertext of the message. The encryption algorithm inputs the public parameter PP, a message m, the revocation list R, and access policy $W = (M, \rho, \mathcal{T})$ described in Section 2.3, where M is a $l \times n$ matrix and ρ maps a row of M into an attribute name in A that each attribute name can only appear once. $\mathcal{T} = \{t_{\rho(i)}\}_{i\in[1,l]}$ is the attribute value associated with (M, ρ). The algorithm randomly selects a vector $v = (s, v_2, \ldots, v_n)^T$, where $v_2, \ldots, v_n \in Z_p$ are applied to share the secret $s \in Z_p$. The algorithm computes $\lambda_i = M_i \cdot v$ for each $i \in [1, l]$, where M_i is the i-th row of M. The algorithm is executed as follows:

1. For each $i \in [1, l]$, the algorithm randomly selects $t_i \in Z_p$ and computes the ciphertext component associated with the access policy: $\left\langle \begin{array}{l} C = me(g,g)^{\alpha s}, C_0 = g^s, C_0' = g^{as}, \\ \{C_{i,1} = h^{\lambda_i} u^{t_i}, C_{i,2} = g^{-t_i \cdot t_{\rho(i)} + \lambda_i}, C_{i,3} = g^{t_i}\}_{i\in[1,l]} \end{array} \right\rangle$,
2. Let $cover(R)$ be the minimum cover set associated with the revocation list R, for each $j \in cover(R)$, the algorithm computes the ciphertext component $\langle \{T_j = y_j^s\}_{j\in cover(R)} \rangle$ that is associated with the revocation list R,
3. Let $\overline{W} = (M, \rho)$ be an incomplete access policy that removes the attribute value set. The data owner sends the final ciphertext $CT = \langle C, C_0, C_0', \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i\in[1,l]}, \{T_j\}_{j\in cover(R)}, R, \overline{W}\rangle$ that includes $\overline{W}$ and the revocation list R to the cloud. The cloud and users cannot access the specific attribute values, thus realizes the partially hidden policy.

4.3 KeyGen

The authority runs the algorithm that is used to generate the user decryption key. The algorithm inputs the user identity u in U, the master key MSK and user attribute set $\mathcal{S} = (I_S, S)$, where $I_S \subseteq A$ is the set of user attribute name, and $S = \{s_i\}_{i\in I_S}$ is the set of user attribute values. Next, $c = Enc_k(i_d)$ is calculated, where i_d is the value of the leaf node in the binary

Authorized licensed use limited to: Tsinghua University. Downloaded on September 27,2022 at 03:46:20 UTC from IEEE Xplore. Restrictions apply.

tree associated with the user u. The algorithm is executed as follows:

1. The algorithm randomly selects $r \in Z_p$, and for $\forall \tau \in I_S$, the decryption key component $\langle K' = c, K = g^{\frac{\alpha}{a+c}} h^r, L = g^r, L' = g^{ar}, \{K_\tau = g^{s_\tau \cdot r} u^{-(a+c)r}\}_{\tau \in I_S} \rangle$ is computed, as it is associated with user attributes,
2. Suppose $path(i_d) = \{i_0, \ldots, i_d\}$, where $i_0 = root$ and i_d is the value of the leaf node in the binary tree that is associated with the user u. The algorithm computes the decryption key component $\langle K_u = g^{r/x_{i_d}} \rangle$ that is associated with the user u,
3. Finally, the algorithm outputs the decryption key $SK = \langle K', K, L, L', K_u \{K_\tau\}_{\tau \in I_S}, \{x_i\}_{i \in path(i_d)}, \mathcal{S} \rangle$ and sends it to the user.

4.4 Decrypt

The decryption algorithm is called by the user. The algorithm inputs ciphertext CT and a decryption key SK next. There exist two cases:

Case1: If the attribute set of user $S \notin (M, \rho)$ or the user identity $u \in R$, then the algorithm aborts,

Case2: If $u \notin R$ and $S \in (M, \rho)$, then the algorithm is executed as follows:

1. For $u \notin R$, there exists a node j such that $j \in cover(R) \cap path(u)$. Suppose that $path(u) = \{i_0, \ldots, i_{dept(j)}, \ldots, i_d\}$, where $i_{dept(j)} = j$ and i_d is the value of the leaf node in the binary tree associated with the user u, the algorithm computes $\theta = x_{i_d}/x_j$, and then calculates $B = e(K_u, T_j)^\theta = e(g^{r/x_{i_d}}, y_j^s)^\theta = e(g,g)^{rs}$.
2. Let $I = \{i : \rho(i) \in I_S\} \subseteq [1, 2, \ldots, l]$, which is the matrix row number set that user's attribute names satisfy the access policy (M, ρ), there exist coefficients $\{c_i | i \in I\}$ such that $\sum_{i \in I} c_i M_i = (1, 0, \ldots, 0)$, and thus, $\sum_{i \in I} c_i \lambda_i = s$. Next, the algorithm computes:

$$
\begin{aligned}
E &= e\Big(L^{K'} \cdot L', C_{i,1}\Big) e\big(L, C_{i,2}\big) e\big(K_{\rho(i)}, C_{i,3}\big) \\
&= e\Big(g^{(a+c)r}, h^{\lambda_i} u^{t_i}\Big) e\Big(g^r, g^{-t_i \cdot t_{\rho(i)} + \lambda_i}\Big) \cdot e\Big(g^{s_{\rho(i)} \cdot r} u^{-(a+c)r}, g^{t_i}\Big) \\
&= e(g,h)^{(a+c)rs} e(g,g)^{rs} \\
F &= \prod_{i \in I} (E)^{c_i} = \prod_{i \in I} \Big(e(g,h)^{(a+c)r\lambda_i} e(g,g)^{r\lambda_i} \Big)^{c_i} \\
&= e(g,h)^{(a+c)r\lambda_i} e(g,g)^{r\lambda_i}
\end{aligned}
$$

3. Next, the algorithm computes:

$$
\begin{aligned}
D &= e\Big(K, C_0^{K'} \cdot C_0'\Big) = e\Big(g^{\frac{\alpha}{a+c}} h^r, g^{(a+c)s}\Big) \\
&= e(g,g)^{\alpha s} e(g,h)^{(a+c)rs}.
\end{aligned}
$$

4. Finally, the algorithm recovers the message as $m = \frac{C \cdot F}{D \cdot B}$.

4.5 KeySanityCheck

The authority runs the algorithm. The KeySanityCheck algorithm is used to evaluate whether the decryption key needs to be tracked. If the decryption key SK is suspected, the algorithm will check whether the decryption key satisfies the KeySanityCheck that consists of four parts:

$$K' \in Z_p, K, L, L', K_\tau, K_u \in G \tag{1}$$

$$e(g, L') = e(g^a, L) \neq 1 \tag{2}$$

$$e\Big(K, g^a g^{K'}\Big) = e(g,g)^\alpha e\Big(L^{K'} \cdot L', h\Big) \neq 1 \tag{3}$$

$$\exists \tau \in I_S,\ s.t.\ e(K_\tau, g) e\Big(L^{K'} \cdot L', u\Big) = e(L,g)^{s_\tau} \neq 1. \tag{4}$$

If the decryption key SK satisfies the equations (1), (2), (3), (4), it can pass the key sanity check, and the algorithm outputs 1; otherwise, the algorithm outputs $\perp$.

4.6 Trace

The algorithm is performed by the authority. If the decryption key SK cannot pass the key sanity check, the algorithm aborts, and outputs $\perp$. Otherwise, the algorithm is executed as follows:

1. It computes $i_d = Dec_k(K')$ to recover the leaf node value i_d associated with the user u,
2. Search the leaf node in the binary tree which value is i_d and then output the user u associated with i_d. If there not exists such node, the algorithm aborts, and outputs $\perp$,
3. If $u \notin R$ then u is added to the revocation list R, so the new revocation list is $R' = R \cup \{u\}$.

4.7 CTUpdate

The cloud runs the ciphertext updated algorithm. The ciphertext updated algorithm considered in this research is the same scheme as presented in [32]. The authority randomly chooses $\eta \in Z_p$ and computes $X' = \{\eta \cdot x_i\}_{i=0}^{2|T|-2}$, then send it to the cloud via a secret channel. The cloud inputs the updated key X', the latest revocation list R', and the ciphertext CT. Next, the algorithm will output the updated ciphertext CT' related to the latest R'. Let $cover(R')$ be the minimum cover set associated with the latest revocation list R'. Given $j' \in cover(R)$, there are two cases:

1. If there exists $j \in cover(R)$ such that $j = j'$, then set $T_j = T_j$,
2. If there exists $j \in cover(R)$ such that j is an ancestor of j', suppose that $path(j') = path(j) \cup \{i_{dept(j)+1}, \ldots, i_{dept(j')}\}$, where $i_{dept(j)} = j$ and $i_{dept(j')} = j'$. Let $Y_j = T_j$ and compute iteratively $Y_{i_{k+1}} = (Y_{i_k})^{\frac{x'_{i_{k+1}}}{x'_{i_k}}} = y^s_{i_{k+1}}$, where $k = dept(j), \ldots, dept(j')$. Next, $T_j = Y_{j'}$ is set.

The ciphertext component associated with access policy is unchanged, and finally, the updated ciphertext is $CT = \langle C, C_0, C_0', \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i \in [1,l]}, \{T_j\}_{j' \in cover(R')}, R', \overline{W} \rangle$.

5 Security Analysis

In this section, the traceability of TR-AP-CPABE scheme based on l-SDH hardness assumption and the IND-CPA security of TR-AP-CPABE based on q-BDHE hardness assumption are shown.

Authorized licensed use limited to: Tsinghua University. Downloaded on September 27,2022 at 03:46:20 UTC from IEEE Xplore. Restrictions apply.

5.1 Traceability

Theorem 1. *If l-SDH hardness assumption holds, the TR-AP-CPABE scheme is traceable based on $q < l$, where q is the number of key queries from the adversary* $\mathrm{A}dv$.

Proof. If there exists a polynomial-time adversary $\mathrm{A}dv$ that can win the traceability game with the advantage ε after q key queries. Suppose that $l = q + 1$, a challenger B that can break the l-SDH hardness assumption with a non-negligible advantage can be constructed. Let G and G_T be two multiplication cyclic groups of prime order p and g is a generator of G, A map $e : G \times G \rightarrow G_T$ is a bilinear map. B receives an l-SDH problem $(p, G, G_T, e, g_1, g_1^a, g_1^{a^2}, \ldots, g_1^{a^l})$, where $g_1 \in G, a \in Z_p$ and the goal is to output a tuple $\left(c_r, w_r = g_1^{\frac{a}{a+c_r}}\right)$. □

Let $A_i = g_1^{a^i}$, where $i = 0, 1, \ldots, l$. B can be able to play the role of a challenger for the adversary $\mathrm{A}dv$ and solve the l-SDH challenge problem. The simulation is executed as follows:

Initialization: B executes *Setup* algorithm and randomly chooses q different values $c_1, c_2, \ldots, c_q \in Z_p^*$ and also randomly selects $\alpha, \theta \in Z_p, u \in G$. Suppose $f(y) = \prod_{i=1}^{q}(y + c_i) = \sum_{i=0}^{q} \alpha_i y^i$, where $\alpha_i \in Z_p, i = 0, 1, \ldots, q$ are the coefficients of $f(y)$. B is executed as follows:

1. Let $g = \prod_{i=0}^{q}(A_i)^{\alpha_i} = g_1^{f(a)}$ and $g^a = \prod_{i=1}^{q+1}(A_i)^{\alpha_{i-1}} = g_1^{f(a)\cdot a}$,
2. For each node in the binary tree T, randomly chooses $\{x_i\}_{i=0}^{2|U|-2} \in Z_p$, computes $\{y_i = g^{x_i}\}_{i=0}^{2|U|-2}$ and publishes the public parameter as: $PP = \langle p, G, G_T, e, g, h = g^\theta, u, e(g,g)^\alpha, g^a, \{y_i\}_{i=0}^{2|U|-2}\rangle$.

Key Query. The adversary $\mathrm{A}dv$ submits $(u_i, \mathcal{S}_i)$ to B for the related decryption keys of $(u_i, \mathcal{S}_i)$ where $\mathcal{S}_i = (I_S, S)$. As it comes to the i-th query, $i < q$, set $f_i(y) = \frac{f(y)}{y+c_i} = \prod_{j=1, j\neq i}^{q}(y + c_j) = \sum_{j=0, j\neq i}^{q-1} \beta_j y^j$, where $\beta_j \in Z_p, j = 0, 1, \ldots, q-1$ are the coefficients of $f_i(y)$. B computes $\sigma_i = \prod_{j=0}^{q-1}(A_j)^{\beta_j} = g_1^{f_i(a)} = g_1^{\frac{f(a)}{a+c_i}} = g^{\frac{1}{a+c_i}}$.

B randomly selects $r \in Z_p$ and computes the decryption key component $\left\langle \begin{array}{l} K' = c_i, K = (\sigma_i)^\alpha h^r = g^{\frac{\alpha}{a+c_i}} h^r, L' = (g^a)^r = g^{ar}, \\ L = g^r, \{K_\tau = g^{s_\tau \cdot r}(u^a \cdot u^{c_i})^{-r} = g^{s_\tau \cdot r} u^{-(a+c_i)r}\}_{\tau \in I_S} \end{array} \right\rangle$ that is associated with $(u_i, \mathcal{S}_i)$.

Suppose $path(i_d) = \{i_0, \ldots, i_d\}$, where $i_0 = root$ and i_d is the value of the leaf node in the tree associated with the user u_i. B computes the decryption key component $\langle K_{u_i} = g^{r/x_{i_d}}\rangle$ that is associated with the user u_i. Finally, B sends the decryption key $SK_i = \langle K', K, L, L', K_u\{K_\tau\}_{\tau \in I_S}, \{x_i\}_{i \in path(i_d)}, \mathcal{S}_i\rangle$ to $\mathrm{A}dv$.

Key Forgery: $\mathrm{A}dv$ submits a forged key $SK_{\mathrm{A}dv}$ to B and $\varepsilon_{\mathrm{A}dv}$ denotes the event of $\mathrm{A}dv$ winning the game. That is, the key $SK_{\mathrm{A}dv}$ satisfies the key sanity check and $K' \notin \{c_1, c_2, \ldots, c_q\}$. There exist two cases:

1. If the event $\varepsilon_{\mathrm{A}dv}$ does not happen, B randomly selects $(c_r, w_r) \in Z_p \times G$ as a solution to l-SDH problem.
2. If the event $\varepsilon_{\mathrm{A}dv}$ happens, B sets a polynomial $f(y) = \varphi(y)(y + K') + \varphi - 1$, where $\varphi(y) = \sum_{i=0}^{q-1} \varphi_i y^i$ and $\varphi - 1 \in Z_p^*$. Since $f(y) = \prod_{i=1}^{q}(y + c_i)$, $c_i \in Z_p^*$ and $K' \notin \{c_1, c_2, \ldots, c_q\}$, then $(y + K')$ cannot divide $f(y)$. Suppose $L = g^r$ and $r \in Z_p$ is unknown. We can get $L' = g^{ar}$ according to $e(g, L') = e(g^a, L)$ and then get $K = g^{\frac{\alpha}{a+K'}} h^r$ according to $e(K, g^a g^{K'}) = e(g,g)^\alpha e(L^{K'} \cdot L', h)$.

Let $\eta = (K/L^\theta)^{\alpha^{-1}} = g^{\frac{1}{a+K'}} = g_1^{\frac{f(a)}{a+K'}} = g_1^{\varphi(a)} g_1^{\frac{\varphi-1}{a+K'}}$, and B computes (c_r, w_r), where $c_r = K' \bmod \mathrm{p} \in \mathrm{Z}_p$ and $w_r = (\eta \cdot \prod_{i=0}^{q-1} A_i^{-\varphi_i})^{\frac{1}{\varphi-1}} = g_1^{\frac{1}{a+K'}}$.

Since $e(g_1^a \cdot g_1^{c_r}, w_r) = e(g_1^a \cdot g_1^{K'}, g_1^{\frac{1}{a+K'}}) = e(g_1, g_1)$, then the tuple (c_r, w_r) is the solution to l-SDH challenge problem.

The advantage of B solving the l-SDH hardness assumption is verified and follows next. Let $\varepsilon_{\mathrm{A}dv}$ denote the event (c_r, w_r) as the solution to l-SDH challenge problem, and verify whether the $e(g_1^a \cdot g_1^{c_r}, w_r) = e(g_1, g_1)$ holds. For simplicity, $\varepsilon_{\mathrm{A}dv}$ is denoted as 0 since the $\varepsilon_{\mathrm{A}dv}$ can happen with a negligible advantage when B randomly select the tuple (c_r, w_r). As B outputs (c_r, w_r), the probability of (c_r, w_r) that satisfies the equality $e(g_1^a \cdot g_1^{c_r}, w_r) = e(g_1, g_1)$ is 1 in the case of $(\mathrm{A}dv \text{ wins } \wedge \gcd(\gamma - 1, p) = 1)$. Suppose the advantage of $\mathrm{A}dv$ winning the game is ε, then the probability of B solving l-SDH challenge problem is:

$$\begin{aligned}
\Pr[\varepsilon_{SDH}] &= \Pr\left[\varepsilon_{SDH} | \overline{\mathrm{A}dv \text{ wins}}\right] \cdot \Pr\left[\overline{\mathrm{A}dv \text{ wins}}\right] \\
&\quad + \Pr[\varepsilon_{SDH} | \mathrm{A}dv \text{ wins} \wedge \gcd(\gamma - 1, p) \neq 1] \\
&\quad \cdot \Pr[\mathrm{A}dv \text{ wins} \wedge \gcd(\gamma - 1, p) \neq 1] \\
&\quad + \Pr[\varepsilon_{SDH} | \mathrm{A}dv \text{ wins} \wedge \gcd(\gamma - 1, p) = 1] \\
&\quad \cdot \Pr[\mathrm{A}dv \text{ wins} \wedge \gcd(\gamma - 1, p) = 1] \\
&= 0 + 0 + 1 \cdot \Pr[\mathrm{A}dv \text{ wins} \wedge \gcd(\gamma - 1, p) = 1] \\
&= \Pr[\mathrm{A}dv \text{ wins}] \cdot \Pr[\gcd(\gamma - 1, p) = 1] \\
&= \varepsilon
\end{aligned}$$

5.2 IND-CPA Security Analysis of TR-AP-CPABE

In the TR-AP-CPABE scheme, the updated ciphertext is the same as the original ciphertext, so the security associated with the original ciphertext needs to be verified.

Theorem 2. *If the $q - BDHE$ hardness assumption holds, then there are no polynomial-time adversaries that can break the TR-AP-CPABE scheme with the non-negligible advantage under the selective access policy and chosen plaintext attacks, where $q > 2|U| - 2$ and $|U|$ is the number of users in the system.*

Proof. If there exists an adversary $\mathrm{A}dv$ that can break the TR-AP-CPABE scheme with a non-negligible advantage ε, then a challenger B that can solve the $q - BDHE$ hardness assumption with the advantage $\frac{\varepsilon}{2}$ can be constructed. B is executed as follows: □

Let G and G_T are two multiplication cyclic groups of prime order p and g is a generator of G, A map $e : G \times G \rightarrow G_T$ is a bilinear map. B flips a fair coin $\mu = \{0, 1\}$. Give $Y' = (g, g^s, g^\alpha, g^{\alpha^2}, \ldots, g^{\alpha^q}, g^{\alpha^{q+2}}, \ldots, g^{\alpha^{2q}})$ and if $\mu = 0$, then B computes $Z = e(g,g)^{\alpha^{q+1}s}$; if $\mu = 1$, B randomly selects $Z \in G_T$.

Init: $\mathrm{A}dv$ chooses a challenge access policy $W^* = (M^*, \rho^*, \mathcal{T})$ and a revocation list R^*, where M^* is a $l^* \times n^*$ matrix and $n^* \leq q$, ρ^* maps a row of M^* into an attribute name that can only appear once. $\mathcal{T} = \{t_{\rho^*(i)}\}_{i \in [1, l^*]}$ is the attribute value associated with (M^*, ρ^*).

Setup: To generate the public parameter, B is processed as follows:

1. Choose $\alpha' \in Z_p$ randomly, and set $e(g,g)^\alpha = e(g,g)^{\alpha'} e(g^d, g^{d^q})$, then $\alpha = \alpha' + d^{q+1}$,
2. Choose $a \in Z_p$ and compute g^a, $h = g^d$, $u = g^{d^q}$,

Authorized licensed use limited to: Tsinghua University. Downloaded on September 27,2022 at 03:46:20 UTC from IEEE Xplore. Restrictions apply.

3. For the revocation list R^*, set $I_{R^*} = \{i \in path(u) | u \in R^*\}$, $v_i \in Z_p\ \forall i = 0, 1, \ldots, 2|U| - 2$ is selected. If $i \in I_{R^*}$, set $y_i = g^{v_i} g^{d^i}$, then $x_i = v_i + d^i$; otherwise, set $y_i = g^{v_i} g^{d^q}$, then $x_i = v_i + d^q$

Publish the public parameter as $PP = \langle p, G, G_T, e, g, h, u, e(g,g)^{\alpha}, g^a, \{y_i\}_{i=0}^{2|U|-2} \rangle$.

Phase 1: Adv submits a sequence of user attribute sets $(u, \mathcal{S} = (I_S, S))$ to request the related decryption keys, where I_S is the attribute name of the user, and $S = \{s_i\}_{i \in I_S}$ is the attribute value of the user. For each attribute value s_τ in S and $\forall i \in \{1, 2, \ldots, l^*\}$, if $s_\tau = t_{\rho^*(i)}$, then set $u_\tau = s_\tau + \sum_{n=1}^{n^*} d^n M^*_{k,n}$, otherwise set $u_\tau = s_\tau$. There exist four cases, where the $\mathcal{S} \models W^*$ means that the $\mathcal{S}$ matches the access policy W^*, and the $\mathcal{S} \not\models W^*$ means that the $\mathcal{S}$ does not match the access policy W^*:

Case 1: If $\mathcal{S} \models W^*$ and $u \notin R^*$, then abortion.

Case 2: If $\mathcal{S} \models W^*$ and $u \in R^*$, B is processed as follows:

1. Select $c \in Z_p$ randomly. Let $r = -\frac{d^q}{a+c} + \frac{d^{q-1}}{a+c} \cdot \frac{M^*_{i,1}}{M^*_{i,2}}$ and compute $K', K, L, L', \{K_\tau\}_{\tau \in I_S}$: $K' = c$,

$$K = g^{\frac{\alpha'}{a+c}} \left(g^{\frac{d^q}{a+c}}\right)^{\frac{M^*_{i,1}}{M^*_{i,2}}} = g^{\frac{\alpha}{a+c}} h^r,$$

$$L = \left[\left(g^{d^q}\right)^{\frac{1}{a+c}}\right]^{-1} \left[\left(g^{d^{q-1}}\right)^{\frac{1}{a+c}}\right]^{\frac{M^*_{i,1}}{M^*_{i,2}}} = g^r,$$

$$L' = (L)^a = g^{ar},$$

$$K_\tau = \left[\left(g^{d^q}\right)^{\frac{s_\tau}{a+c}}\right]^{-1} \cdot \left[\left(g^{d^{q-1}}\right)^{\frac{s_\tau M^*_{i,1}}{a+c M^*_{i,2}}}\right]^{\frac{M^*_{i,1}}{M^*_{i,2}}} \cdot g^{d^{2q}} \cdot \left[\left(g^{d^{2q-1}}\right)^{\frac{M^*_{i,1}}{M^*_{i,2}}}\right]^{-1} \cdot \left[\left(\prod_{k=1}^{n^*} g^{d^{q+k} \cdot M^*_{i,k}}\right)^{-1} \cdot \left(\prod_{k=1}^{n^*} g^{d^{q+k-1} \cdot M^*_{i,k}}\right)^{\frac{M^*_{i,1}}{M^*_{i,2}}}\right]^{\frac{1}{a+c}} = g^{u_\tau \cdot r} u^{-(a+c)r},$$

where $\tau \in I_S$.

2. Suppose that $path(i_d) = \{i_0, \ldots, i_d\}$, where $i_0 = root$ and i_d is the value of the leaf node in the tree associated with the user u. Since $u \in R^*$, then $i_d \in I_{R^*}$, $x_{i_d} = v_{i_d} + d^{i_d}$ is concluded. B computes $K_u = \left[(g^{d^q})^{-1} \cdot (g^{d^{q-1}})^{\frac{M^*_{i,1}}{M^*_{i,2}}}\right]^{\frac{1}{(v_{i_d} + d^{i_d}) \cdot (a+c)}} = g^{r/x_{i_d}}$.

Case 3: If $\mathcal{S} \not\models W^*$ and $u \in R^*$, B is processed as follows:

1. Select a vector $\overrightarrow{\omega} = (\omega_1, \omega_2, \ldots, \omega_{n^*}) \in Z_p^{n^*}$ randomly, where $\omega_1 = -1$ and $M_i^* \cdot \overrightarrow{\omega} = 0$ for all i where $\rho^*(i) \in I_S$. Such a vector exists as per the definition of $LSSS$,
2. Randomly choose $c \in Z_p$ and set $K' = c$,
3. Randomly choose $t \in Z_p$ and implicitly define $r = \frac{1}{a+c}(t + \omega_1 d^q + \omega_2 d^{q-1} \cdots + \omega_{n^*} d^{q-n^*+1})$,
4. Compute K, L, L': $L = g^{\frac{t}{a+c}} \prod_{i=1}^{n^*} (g^{\omega_i d^{q+1-i}})^{\frac{1}{a+c}} = g^r$, $L' = g^{\frac{at}{a+c}} \prod_{i=1}^{n^*} (g^{\omega_i d^{q+1-i}})^{\frac{a}{a+c}} = g^{ar}$, $K = (g^{\alpha' + dt} \prod_{i=2}^{n^*} g^{\omega_i d^{q+2-i}})^{\frac{1}{a+c}} = g^{\frac{\alpha}{a+c}} h^r$,
5. For $\forall \tau \in I_S$, if there exists i such that $\rho^*(i) = \tau$ and $s_\tau = t_{\rho^*(i)}$, then B computes $K_\tau = L^{s_\tau} \left[\prod_{j=1}^{n^*} \left(g^{t \cdot d^j} \cdot \prod_{k=1}^{n^*} g^{\omega_k d^{q+1+j-k}}\right)^{M^*_{i,j}}\right]^{\frac{1}{a+c}} \cdot (g^{t \cdot d^q} \prod_{i=1}^{n^*} g^{\omega_i d^{2q+1-i}})^{-1}$. Otherwise, B computes $K_\tau = L^{s_\tau} (g^{t \cdot d^q} \prod_{i=1}^{n^*} g^{\omega_i d^{2q+1-i}})^{-1}$,
6. Suppose that $path(i_d) = \{i_0, \ldots, i_d\}$, where $i_0 = root$ and i_d is the value of the leaf node in the tree associated with the user u. Since $u \in R^*$, then $i_d \in I_{R^*}$, $x_{i_d} = v_{i_d} + d^{i_d}$ is obtained. Next, B computes $K_u = (g^t \prod_{i=1}^{n^*} g^{\omega_i d^{q+1-i}})^{\frac{1}{(v_{i_d} + d^{i_d}) \cdot (a+c)}} = g^{r/x_{i_d}}$.

Case 4: If $\mathcal{S} \not\models W^*$ and $u \notin R^*$, B computes $K', K, L, L', \{K_\tau\}_{\tau \in I_S}$, following to Case 3. Suppose that $path(i_d) = \{i_0, \ldots, i_d\}$, where $i_0 = root$ and i_d is the value of the leaf node in the tree associated with the user u. Since $u \notin R^*$, then $i_d \notin I_{R^*}$, we know $x_{i_d} = v_{i_d} + d^q$. Next, B computes $K_u = (g^t \prod_{i=1}^{n^*} g^{\omega_i d^{q+1-i}})^{\frac{1}{(v_{i_d} + d^q) \cdot (a+c)}} = g^{r/x_{i_d}}$.

Challenge: Adv submits two equal length messages m_0, m_1 to B. B is processed as follows:

1. B casts a fair coin $b \in \{0, 1\}$ and computes $C = m_u e(g,g)^{\alpha s}$, $C_0 = g^s$, $C_0' = (g^s)^a$,
2. B Randomly selects $r_2, \ldots, r_n \in Z_p^*$ and sets $\overrightarrow{v} = (s, sd + r_2, sd^2 + r_3, \ldots, sd^{n^*-1} + r_{n^*})^T \in Z_p^*$, then computes: $C_{i,1} = \prod_{j=2}^{n^*} (g^{dr_j})^{M^*_{i,j}} \prod_{j=1}^{n^*} (g^{sd^j})^{M^*_{i,j}} g^{-ad^{q+i}}$,
3. $C_{i,2} = (g^{t_{\rho^*(i)}})^{-ad^i} \prod_{j=2}^{n^*} (g^{d^j M^*_{i,j}})^{-ad^i} \cdot \prod_{j=2}^{n^*} (g^{r_j})^{M^*_{i,j}} \prod_{j=1}^{n^*} (g^{sd^{j-1}})^{M^*_{i,j}}$, $C_{i,3} = g^{-ad^i}$,
4. For $\forall j \in cover(R^*)$, since $x_j = v_j + d^q$ and $y_j = g^{v_j + d^q}$, then B sets $T_j = (g^s)^{v_j + d^q} = y_j^s$.

Finally, B sets the ciphertext as $CT = \langle C, C_0, C_0', \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i \in [1, l^*]}, \{T_j\}_{j \in cover(R^*)} \rangle$ and sends it to Adv.

Phase 2: Same as Phase 1.

Guess: Adv outputs a guess b' of b.

If $b = b'$, B outputs a guess $\mu' = 0$ of μ.

If $b \neq b'$, B outputs a guess $\mu' = 1$ of μ.

1. With $\mu = 0$, $Z = e(g,g)^{\alpha^{q+1} s}$ and Adv obtain a legal ciphertext. Assuming that the advantage of Adv is $\varepsilon = \Pr[b = b' | \mu = 0] - \frac{1}{2}$, $\Pr[b = b' | \mu = 0] = \Pr[\mu = \mu' | \mu = 0]$ is concluded. Thus, the probability that B wins the game is $\Pr[\mu = \mu' | \mu = 0] = \varepsilon + \frac{1}{2}$.
2. With $\mu = 1$, Z is a random element in G_T. Thus, Adv cannot obtain any information about b. In this case, Adv loses the attack advantage, so that $\Pr[b \neq b' | \mu = 1] = \frac{1}{2}$. $\Pr[b \neq b' | \mu = 1] = \Pr[\mu = \mu' | \mu = 1]$ is easily known, and therefore, the probability that B wins the game is $\Pr[\mu = \mu' | \mu = 1] = \frac{1}{2}$.

Finally, the advantage that B solves $q - BDHE$ hardness assumption is defined as:

$$\begin{aligned} \Pr[\mu = \mu'] &= \Pr[\mu = \mu' | \mu = 0] \cdot \Pr[\mu = 0] \\ &\quad + \Pr[\mu = \mu' | \mu = 1] \cdot \Pr[\mu = 1] - \frac{1}{2} \\ &= \left(\varepsilon + \frac{1}{2}\right) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} = \frac{1}{2}\varepsilon \end{aligned}$$

Therefore, Theorem 2 is proved.

Authorized licensed use limited to: Tsinghua University. Downloaded on September 27,2022 at 03:46:20 UTC from IEEE Xplore. Restrictions apply.

TABLE 1
Functionality Comparison

Scheme	Revocation	Access Structure	Security	Update	Traceability	Hidden Policy
UR-CPABE	User	Tree	Selective	Key	×	×
AR-CPABE	Attribute	Tree	Selective	Key	×	×
[21]	Attribute	LSSS	Selective	Key	×	×
[28]	×	LSSS	Selective	×	√	×
[33]	Attribute	LSSS	Selective	Ciphertext	√	×
[38]	×	LSSS	Full	×	×	√
Ours	User	LSSS	Selective	Ciphertext	√	√

√ denotes that there exists this functionality. × denotes that there not exists this functionality.

6 Performance Analysis

In this section, the functionality and efficiency of the proposed scheme and other schemes [19], [21], [28], [33], [38] are evaluated and compared. In Table 1, the functionality comparisons are discussed, where the scheme [19] proposed two revocation schemes that are user revocation scheme UR-CPABE and attribute revocation scheme AR-CPABE. However, the key of unrevoked users needs to be updated when the revocation occurs, which will cause more overhead. The scheme [21] only implements the attribute revocation with the key updating. The schemes [19] and [21] do not implement traceability and hidden policy. The scheme [38] supports the hidden policy even if the revocation traceability mechanism is not implemented, so its function is relatively straightforward. The scheme [28] implements only the traceability mechanism, while the scheme [33] can revoke the malicious user based on the implementation of the traceability mechanism without hidden the access policy. The access policy is hidden in the proposed scheme, and thus, the privacy leakage due to the access policy is reduced to a minimum, while the traceability and revocation are supported. Also, the proposed scheme supports the ciphertext updated. The proposed scheme and scheme [33] are proved to be safe under the selective access policy and chosen-plaintext attack.

Table 2 discusses the efficiency performance of the proposed scheme against related schemes, where $t_1 = \sum_{j' \in cover(R')} (dept(j') - 1 - dept(j))$. Only the efficiency of key updating is compared in the scheme [19]. Based on the prime order group, scheme [38] is compared, which is observed that the proposed scheme is more efficient than other related ones. In the KeyGen and Encrypt algorithms, the exponent operation and multiplication operation of the proposed scheme are of lower complexity than other schemes, so higher the efficiency. In the Decrypt algorithm, since the scheme [21] only implements the attribute revocation and scheme [28] only implements the traceability mechanism, the decryption efficiency is slightly better than the proposed scheme. In scheme introduced in [33], the user can decrypt the plaintext based only on the pairing operation of the ciphertext component $C_{i,1}$ and the decryption key, which can abandon the pairing operation between the user attributes and the access policy and reduces the security of encryption.

In comparison among the proposed scheme and previously published schemes [38] and [33], the proposed scheme is more efficient in decryption. Comparing to the scheme [28] and [33], the proposed scheme is significantly more efficient in tracing of the Trace algorithm, due to additional multiplication operations of the scheme [28] and more pairing operation in the scheme [33]. Also, the proposed scheme does not need to initialize the traceable list that reduces the space complexity. The AR-CPABE in schemes [19] and [21] are all attribute revocation and update the key of unrevoked users when revoking. Therefore, the exponent operation and multiplication operation of key updating are based on the user's attributes that result in significant performance overhead. Compared to the UR-CPABE and AR-CPABE in the scheme [19] with the proposed scheme, the proposed scheme is less efficient than scheme [19], in the encryption and decryption algorithms. However, compared with the scheme [19], the advantages of the proposed scheme are reflected in the following aspects: 1. After a revocation, the scheme [19] needs to update the keys for all unrevoked users, which will cause extra overhead. The proposed scheme only needs to update the ciphertext related to the user's revocation list. 2.

TABLE 2
Efficiency Comparison

Scheme	KeyGen	Encrypt	Decrypt	Trace	Update
UR-CPABE	$(2+2s)E+sM$	$(6+2l)E+3P+5M$	$(3+2n)P+(4+n)M$	-	$tE+(t-1)M$
AR-CPABE	$(2+2s)E+sM$	$(2+3l)E+2M$	$(1+2n)P+nE+(2+n)M$	-	$stE+s(t-1)M$
[21]	$(s+sj)E+sjM$	$(2+5l)E+(2l+1)M$	$(1+3n)P+nE+(2+2n)M$	-	$(s+2sr)E+(s+sr)M$
[28]	$(5+3s)E+(1+2s)M$	$(3+5l)E+(2l+1)M$	$(1+3n)P+(2+n)E+(4+2n)M$	$(5+3s)P+(2+s)E+(3+2s)M$	-
[33]	$(4+4s)E+M$	$(3+4l+lr)E+(l+1)M$	$(1+4n)P+(3+n)E+(3+3n)M$	$(4+2s)P+4E+3M$	$(4+s+lr)E+(3+2s)M$
[38]	$(3+2s)E+(1+s)M$	$(5+6l)E+(4l+1)M$	$(2+4n)P+2nE+(3+2n)M$	-	-
Ours	$(6+s)E+(1+s)M$	$(3+4l+r)E+(l+1)M$	$(2+3n)P+(3+n)E+(5+2n)M$	$(6+s)P+(2+s)E+(3+s)M$	t_1E

E denotes an exponent operation in G, G_T. P denotes a bilinear pairing operation. M denotes a multiplication operation in G, G_T. l denotes the number of attributes in access policy. s denotes the number of user's attributes. n denotes the number of attributes in the decryption key that satisfies the access policy. r denotes the length of cover(R). j denotes the length of path(u). t denotes the maximal number of revoked users.

Authorized licensed use limited to: Tsinghua University. Downloaded on September 27,2022 at 03:46:20 UTC from IEEE Xplore. Restrictions apply.

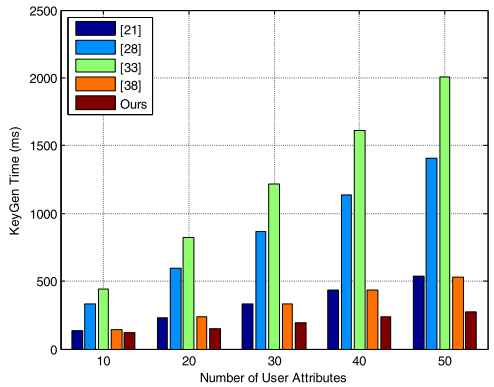

Fig. 4. KeyGen time analysis.

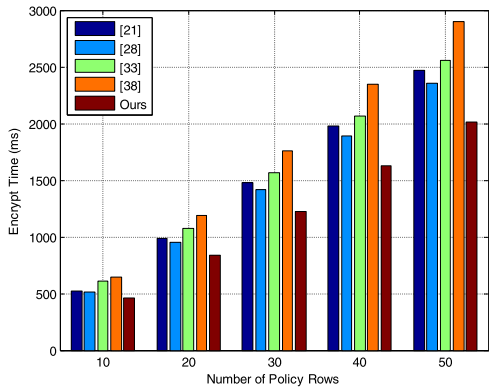

Fig. 5. Encrypt time analysis.

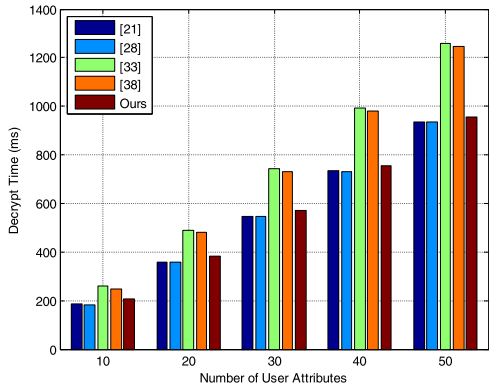

Fig. 6. Decrypt time analysis.

The proposed scheme not only implements the user's revocation, but also partially hidden policy and the traceability, while the scheme [19] only implements the user's revocation and attribute revocation. Both schemes [33] and the proposed one can support the ciphertext updated, but the proposed scheme only needs to update the ciphertext related to the revocation list, which is independent of the access policy. Therefore, based on the same number of users, the ciphertext updated efficiency of the proposed scheme is almost unchanged, and the ciphertext update efficiency of the scheme [33] varies with the size of the access policy.

Figs. 4, 5, 6, 7 and 8 depict the performance and time cost comparison on KeyGen, Encrypt, Decrypt, Trace and Update algorithms among the proposed scheme and schemes [19],

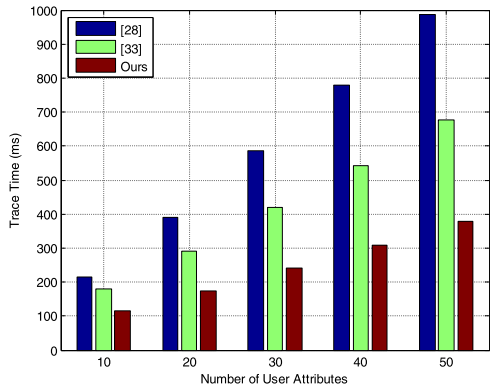

Fig. 7. Trace time analysis.

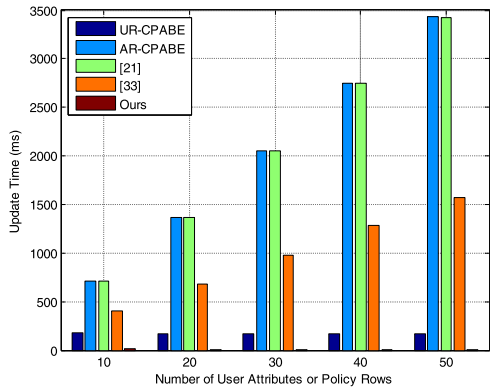

Fig. 8. Update time analysis.

[21], [28], [33], [38]. To carry out the experimentation, the JPBC2.0.0 library is used for the simulation. The hardware environment is Intel Core i5-6500 CPU @ 3.20 GHz, 8 GB memory, the software environment is JDK1.8.0 and Intellij IDEA 2017.1.3. In the Setup stage, the prime-order bilinear pairing of Type A is used, which is constructed on the curve $y^2 = x^3 + x$ over the field F_q for some prime $q = 3 \bmod 4$. In the simulation experiment, the number of users is $10 \sim 50$ range, also in the same range the number of matrix rows of the access policy. The user set is $U = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\}$, the revocation list is $R = \{u_4\}$ and the traceable user is u_1. The final result of the experiment is the average of 50 experiments.

Compared with the mentioned schemes, the proposed scheme is the most efficient that realizes the traceability and revocation as well as the hidden policy applied. In Fig. 4, we show the time cost comparison between the proposed scheme and the schemes above mentioned on KeyGen algorithm. The proposed scheme takes the least time yet the most efficient while the proposed scheme associates the decryption key with the traceable information. Next, it is compared in Fig. 5 the time cost of the Encrypt algorithm. We can see that the proposed scheme is most efficient due to the ciphertext related to the revocation list separated from the ciphertext associated with the hidden policy. In Fig. 6, we compared the time cost of the Decrypt algorithm. Even though the efficiency of the proposed scheme is approximate to the one proposed in [21] and [28], the proposed scheme implements both the user's traceability mechanism and revocation mechanism. The scheme [21] only implements the

Authorized licensed use limited to: Tsinghua University. Downloaded on September 27,2022 at 03:46:20 UTC from IEEE Xplore. Restrictions apply.

attribute revocation and scheme [28] can only implement the user's traceability mechanism. The efficiency of the proposed scheme is much better than the scheme [38] even though it only realized the hidden policy. Both the proposed scheme and scheme [33] realized the white-box traceability and revocation. However, the proposed is faster than [33] at the time of decryption from Fig. 6. Moreover, in the performance of traceability, the proposed scheme has distinct advantages from the comparison of the tracing time between the proposed scheme and the scheme [28], [33] in Fig. 7. In Fig. 8 we show the time comparison in the Update algorithm. We compare the key updating time of UR-CPABE, AR-CPABE in schemes [19] and [21]. Furthermore, compared with the key updating time of schemes [19] and [21], the ciphertext updated time of the proposed scheme is the least. Compared with the scheme [33], the ciphertext updated time of the proposed scheme is constant, and the ciphertext updated time of scheme [33] increases significantly with the matrix size of the access policy. In summary, the experimental results are consistent with the above theoretical analysis.

7 Conclusion and Future Work

In this paper, a traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection (TR-AP-CPABE) is proposed. The LSSS is used to be access policy, and user attributes are represented as attribute names and attribute values, in which the attribute values are used for encryption, thereby the ciphertext-related access policy only contains the attribute name to fulfill the partially hidden policy. The TR-AP-CPABE scheme can trace the user based on the user's decryption key and then revoke the user by using the leaf node values of the binary tree associated with the user information. Since the ciphertext is composed of two parts. One is related to the revocation list, and thus only this part needs to be updated when revoking. Another is associated with the access policy that contains only the attribute names to realize the partially hidden policy. The proposed scheme is efficient, proven to be secure under the chosen plaintext attacks and selective access policy based on the decisional q-BDHE assumption in the standard model.

As future work, we aim at building a traceable and revocable ciphertext-policy attribute-based encryption based on the full hidden policy that realizes the more secure privacy protection and the same attribute name can appear any multiple numbers of times [43], [44]. Also, we will apply the black-box traceability to trace malicious users [45], [46], [47]. In the proposed scheme, we will research the efficiency and performance of the application of the flexible multi-tree, where the leaf nodes are associated with users.

Acknowledgments

The authors wish to thank the editors and reviewers for their constructive feedback and insightful suggestions. This work was supported by the National Natural Science Foundation of China under Grant No. 61672338 and No. 61873160.

References

[1] Seagate Blog, 2019. Accessed: May 10, 2019. Online [Available]: https://blog.seagate.com/business/enormous-growth-in-data-is-coming-how-to-prepare-for-it-and-prosper-from-it/

[2] L. Chen *et al.*, "DMRS: An efficient dynamic multi-keyword ranked search over encrypted cloud data," *Soft Comput.*, 21, pp. 4829–4841, 2017, doi: 10.1007/s00500-017-2684-6.

[3] L. Chen *et al.*, "Improving file locality in multi-keyword top-k search based on clustering," *Soft Comput.*, vol. 22, pp. 3111–3121, 2018, doi: 10.1007/s00500-018-3145-6.

[4] G. Song and Y. Q. Deng, "Security-level switchable attribute-based encryption under the strictly weaker assumption family," *Inf. Sci.*, vol. 482, pp. 47–62, May. 2019, doi: 10.1016/j.ins.2018.12.062.

[5] H. Li and D. Han, "EduRSS: A blockchain-based educational records secure storage and sharing scheme," *IEEE Access*, vol. 7, pp. 179273–179289, 2019.

[6] A. Sahai and B. Waters, "Fuzzy identity based encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2005, pp. 457–473, doi: 10.1007/11426639_27.

[7] V. Goyal, O. Pandey, and A. Sahai, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98, doi: 10.1145/1180405.1180418.

[8] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Security Privacy*, 2007, pp. 321–334.

[9] M. Joshi, K. Joshi, and T. Finin, "Attribute-based encryption for secure access to cloud based EHR systems," in *Proc. IEEE 11th Int. Conf. Cloud Comput.*, 2018, pp. 932–935.

[10] Z. Liu *et al.*, "Practical attribute-based encryption: Outsourcing decryption, attribute revocation and policy updating," *J. Netw. Comput. Appl.*, vol. 108, pp. 112–123, Feb. 2018, doi: 10.1016/j.jnca.2018.01.016.

[11] W. Fan *et al.*, "Deploying parallelized ciphertext-policy attributed based encryption in clouds," *Int. J. Comput. Sci. Eng.*, vol. 16, no. 5, pp. 321–333, 2018.

[12] M. Cui, D. Han, and J. Wang, "An efficient and safe road condition monitoring authentication scheme based on fog computing," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9076–9084, Oct. 2019.

[13] X. Li, K. Liang, and Z. Liu, "Attribute based encryption: Traitor tracing, revocation and fully security on prime order groups," in *Proc. 7th Int. Conf. Cloud Comput. Serv. Sci.*, 2017, pp. 309–320, doi: 10.5220/0006220203090320.

[14] Z. Liu, Z. Cao, and D. S. Wong, "Blackbox traceable CP-ABE: How to catch people leaking their keys by selling decryption devices on ebay," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 475–486, doi: 10.1145/2508859.2516683.

[15] X. Yang, P. Yang, F. An, Q. Zhou, and M. Yang, "Traceable multi-authority attribute-based encryption scheme for cloud computing," in *Proc. 14th Int. Comput. Conf. Wavelet Active Media Technol. Inf. Process.*, 2017, pp. 263–267, doi: 10.1109/ICCWAMTIP.2017.8301492.

[16] Z. Liu *et al.*, "White-box traceable dynamic attribute based encryption," in *Proc. Int. Conf. Secur. Pattern Anal. Cybern.*, 2017, pp. 526–530, doi: 10.1109/SPAC.2017.8304334.

[17] Y. Shi *et al.*, "Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation," *Inf. Sci.*, vol. 295, pp. 221–231, 2015, doi: 10.1016/j.ins.2014.10.020.

[18] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," in *Proc. IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 7, pp. 1214–1221, 2011.

[19] V. Hoang, E. Lehtihet, and Y. Ghamri-Doudane, "Forward-secure data outsourcing based on revocable attribute-based encryption," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf.*, 2019, pp. 1839–1846, doi: 10.1109/IWCMC.2019.8766674.

[20] S. Xu, G. Yang, and Y. Mu, "Revocable attribute-based encryption with decryption key exposure resistance and ciphertext delegation," *Inf. Sci.*, vol. 479, pp. 116–134, Apr. 2019, doi: 10.1016/j.ins.2018.11.031.

[21] G. Xiang, B. Li, X. Fu, M. Xia, and W. Ke, "An attribute revocable CP-ABE scheme," in *Proc. 7th Int. Conf. Adv. Cloud Big Data*, 2019, pp. 198–203.doi: 10.1109/CBD.2019.00044.

[22] C. Fan, V. S. Huang, and H. Ruan, "Arbitrary-state attribute-based encryption with dynamic membership," *IEEE Trans. Comput.*, vol. 63, no. 8, pp. 1951–1961, Aug. 2014.

[23] J. Li, Q. Yu, Y. Zhang, and J. Shen, "Key-policy attribute-based encryption against continual auxiliary input leakage," *Inf. Sci.*, vol. 470, pp. 175–188, Aug. 2018, doi: 10.1016/j.ins.2018.07.077.

[24] W. Dai *et al.*, "Implementation and evaluation of a lattice-based key-policy ABE scheme," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1169–1184, May 2018.

Authorized licensed use limited to: Tsinghua University. Downloaded on September 27,2022 at 03:46:20 UTC from IEEE Xplore. Restrictions apply.

[25] H. Y. Ma, Z. J. Wang, and Z. J. Guan, "Efficient ciphertext-policy attribute-based online/offline encryption with user revocation," *Secur. Commun. Netw.*, vol. 6, pp. 1–11, Feb. 2019, doi: 10.1155/2019/8093578.
[26] Y. Zhang, J. Li, and H. Yan, "Constant size ciphertext distributed CP-ABE scheme with privacy protection and fully hiding access structure," *IEEE Access*, vol. 7, pp. 47982–47990, 2019.
[27] Z. Liu, Z. Cao, and D. S. Wong, "White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 76–88, Jan. 2013.
[28] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin, "White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 6, pp. 1274–1288, Jun. 2015.
[29] J. Ning *et al.*, "Large universe ciphertext-policy attribute-based encryption with white-box traceability," in *Proc. ESORICS 19th Eur. Symp. Res. Comput. Secur.*, 2014, vol. 8713, pp. 55–72, doi: 10.1007/978-3-319-11212-1_4.
[30] J. Ning, Z. Cao, X. Dong, and L. Wei, "White-box traceable CP-ABE for cloud storage service: How to catch people leaking their access credentials effectively," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 883–897, Sep./Oct. 1, 2018.
[31] G. Chen, Z. Xu, H. Jiang, and K.-C. Li, "Generic user revocation systems for attribute-based encryption in cloud storage," *Front. Inf. Technol. Electron. Eng.*, vol. 19, no. 11, pp. 1362–1384, 2018, doi: 10.1631/FITEE.1800405.
[32] Z. Liu, S. Duan, P. Zhou, and B. Wang, "Traceable-then-revocable ciphertext-policy attribute-based encryption scheme," *Future Gener. Comput. Syst.*, vol. 93, pp. 903–913, Oct. 2017, doi: 10.1016/j.future.2017.09.045.
[33] S. Wang, K. Guo, and Y. Zhang, "Traceable ciphertext-policy attribute-based encryption scheme with attribute level user revocation for cloud storage," *Plos One*, vol. 13, no. 9, Sep. 2018, doi: 10.1371/journal.pone.0203225.
[34] H. Lian, G. Wang, and Q. Wang, "Fully secure traceable and revocable-storage attribute-based encryption with short update keys via subset difference method," in *Proc. 3rd Int. Conf. Secur. Smart Cities Ind. Control Syst. Commun.*, 2018, pp. 1–8, doi: 10.1109/SSIC.2018.8556734.
[35] Y. Song *et al.*, "Attribute-based encryption with hidden policies in the access tree," *J. Commun.*, vol. 36, no. 9, pp. 119–126, 2015, doi: 10.11959/j.issn.1000-436x.2015135.
[36] R. Xu, Y. Wang, and B. Lang, "A tree-based CP-ABE scheme with hidden policy supporting secure data sharing in cloud computing," in *Proc. Int. Conf. Adv. Cloud Big Data*, 2013, pp. 51–57, doi: 10.1109/CBD.2013.9.
[37] J. Lai, R. H. Deng, and Y. Li, "Expressive CP-ABE with partially hidden access structures," in *Proc. 7th ACM Symp. Inf. Comput. Commun. Secur.*, 2012, pp. 18–19, doi: 10.1145/2414456.2414465.
[38] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: efficient policy-hiding attribute-based access control," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2130–2145, Jun. 2018.
[39] L. Sun and C. Xu, "Hidden policy ciphertext-policy attribute based encryption with conjunctive keyword search," in *Proc. 3rd IEEE Int. Conf. Comput. Commun.*, 2017, pp. 1439–1443.
[40] F. Yundong, W. Xiaoping, and W. Jiasheng, "Multi-authority attribute-based encryption access control scheme with hidden policy and constant length ciphertext for cloud storage," in *Proc. IEEE 2 nd Int. Conf. Data Sci. Cyberspace*, 2017, pp. 205–212.
[41] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Proc. 21st Annu. Int. Cryptol. Conf.*, 2001, vol. 32, pp. 213–229, doi: 10.1007/3-540-44647-8_13.
[42] L. M. Surhone *et al.*, "Probabilistic encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, 1984, doi: 10.1016/0022-0000(84)90070-9.
[43] W. Liang, K.-C. Li, J. Long, X. Kui, and A. Y. Zomaya, "An industrial network intrusion detection algorithm based on multi-feature data clustering optimization model," *IEEE Trans. Ind. Inf.*, vol. 16, no. 3, pp. 2063–2071, Mar. 2020.
[44] W. Zhang *et al.*, "Wireless sensor network intrusion detection system based on MK-ELM," *Soft Comput.*, to be published, doi: 10.1007/s00500-020-04678-1.
[45] W. Liang, M. Tang, J. Long, X. Peng, J. Xu, and K.-C. Li, "A Secure fabric blockchain-based data transmission technique for industrial internet-of-things," *IEEE Trans. Ind. Inf.*, vol. 15, no. 6, pp. 3582–3592, Jun. 2019.
[46] W. Liang, Y. Fan, K. C. Li, D. Zhang, and J.-L. Gaudiot, "Secure data storage and recovery in industrial blockchain network environments," *IEEE Trans. Ind. Inf.*, to be published, doi: 10.1109/TII.2020.2966069.
[47] W. Liang, W. Huang, J. Long, K. Zhang, K.-C. Li, and D. Zhang, "Deep reinforcement learning for resource protection and real-time detection in IoT environment," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2020.2974281.

Dezhi Han (Member, IEEE) received the PhD degree from the Huazhong University of Science and Technology. He is currently a professor of computer science and engineering with Shanghai Maritime University. His research interests include network security, cloud computing, mobile networking, wireless communication, and cloud security.

Nannan Pan received the BS degree in network engineering from Shanghai Maritime University, and she is currently working toward the the MS degree in computer application technology. Her main research interests include cloud computing, distributed computing and sensor networks, cloud security, and attribute-based encryption.

Kuan-Ching Li (Senior Member, IEEE) is currently a distinguished professor at Providence University, Taiwan. He has received awards and funding support from several agencies and industrial companies, has also received distinguished chair professorships from universities in several countries. He has been actively involved in many major conferences and workshops in program/general/steering conference chairman positions and member of the program committee and has organized numerous conferences related to high-performance computing and computational science and engineering. Besides publication in refereed journals and top conferences papers, he is co-author/co-editor of several technical professional books published by CRC Press/Taylor & Francis, Springer, and McGraw-Hill. His research interests include parallel and distributed computing, big data, and emerging technologies. He is a fellow of the IET.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.

Authorized licensed use limited to: Tsinghua University. Downloaded on September 27,2022 at 03:46:20 UTC from IEEE Xplore. Restrictions apply.