Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 1493290, 10 pages
http://dx.doi.org/10.1155/2016/1493290

Research Article

Conditional Ciphertext-Policy Attribute-Based Encryption Scheme in Vehicular Cloud Computing

Zhitao Guan,[1] Jing Li,[1] Zijian Zhang,[2] and Liehuang Zhu[2]

[1]*School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China*
[2]*School of Computer, Beijing Institute of Technology, Beijing 100081, China*

Correspondence should be addressed to Zijian Zhang; zhangzijian@bit.edu.cn

Received 24 June 2016; Revised 22 September 2016; Accepted 16 October 2016

Academic Editor: Hui Zhu

VCC (Vehicular Cloud Computing) is an emerging and promising paradigm, due to its significance in traffic management and road safety. However, it is difficult to maintain both data security and system efficiency in Vehicular Cloud, because the traffic and vehicular related data is large and complicated. In this paper, we propose a conditional ciphertext-policy attribute-based encryption (C-CP-ABE) scheme to solve this problem. Comparing with CP-ABE, this scheme enables data owner to add extra access trees and the corresponding conditions. Experimental analysis shows that our system brings a trivial amount of storage overhead and a lower amount of computation compared with CP-ABE.

1. Introduction

VANETs (vehicular ad hoc networks) gained great attention in recent years, which can not only improve road safety but also enhance traffic management [1, 2]. In VANET, the vehicles, V2V (Vehicle to Vehicle) and V2I (Vehicle to Infrastructure), generate an enormous amount of data. In order to enhance the scalability of the VANETs, some studies focus on VCC (Vehicular Cloud Computing), which combines cloud computing [3, 4] and VANETs together [5, 6]. The illustration of VCC is shown in Figure 1. In VCC, mass data from different VANET nodes are collected and stored in cloud servers efficiently.

Meanwhile, VCC is faced with serious security and privacy challenges. For instance, the intruders can intrude the onboard infrastructure or the cloud server to get the sensitive data, which may leak the privacy of the data owner (DO), or even endanger the lives of passengers. To solve this problem, we introduce the CP-ABE [7], which also realizes the fine-grained access control on the encrypted data. However, the vehicular situation and surroundings are quite complicated; a one-off encryption under one access tree may be no longer adequate for the needs. In addition, the attributes consisted in access structure may denote more abundant information, not just descriptive information about users' identities. According to these attributes' features, we extract them from the access structure and take them as "conditions." When inserting other access trees, the conditions extracted from them can be used for identifying the corresponding tree.

Main contributions of this paper can be summarized as follows:

(1) In this paper, we propose a conditional ciphertext-policy attribute-based encryption (C-CP-ABE) scheme to allow users to add extra access trees based on the original ciphertext to their own ciphertexts.

(2) All the trees multiply by a parameter except only one tree multiplies by the message, which extends the expression compared with original CP-ABE in [7] without bringing a heavy computation and storage overhead.

(3) We further give the security analysis and performance evaluation, which prove that security and performance of our scheme are no weaker than those of traditional scheme.

The rest of this paper is organized as follows. Section 2 introduces the related work. In Section 3, some preliminaries are given. In Section 4, the definition of the condition is given. In Section 5, the proposed scheme is stated. In Section 6,

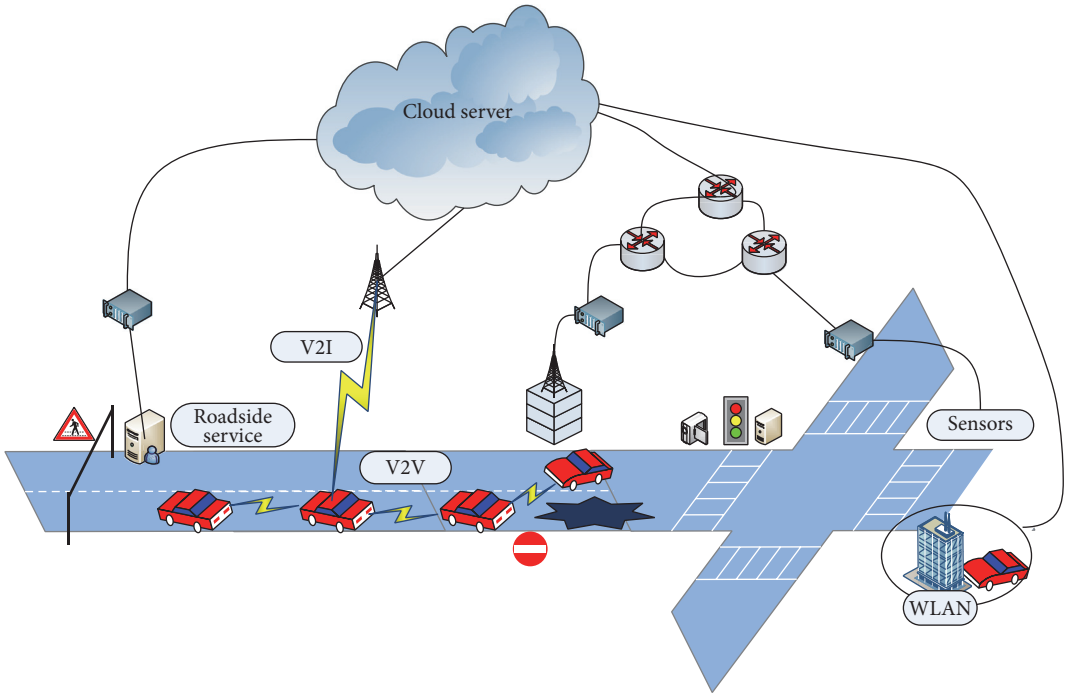

Figure 1: The illustration of VCC (Vehicular Cloud Computing).

security analysis is given. In Section 7, the performance of our scheme is evaluated. In Section 8, the paper is concluded.

2. Related Work

Cloud security has been a hot topic recently [8, 9]. It had developed a new field in the application and research of public key encryption since Shamir's proposed Identity Based Encryption in [10]. Based on IBE, Sahai and Waters proposed a new encryption scheme called Fuzzy IBE; they thought that the users' attributes did not have to be precisely matched to the attributes that are specified by data owners. The ciphertext could be decrypted provided the threshold value of attributes was achieved [11]. Chase constructed a multiauthority attribute-based encryption to compensate for the weaknesses of single authority in [12], allowing each authority to be in charge of a domain of attributes. Soon, they improved the privacy and security of their previous system by removing the central trusted authority, which had usability in practice [13]. In [14], Kapadia et al. proposed a scheme to hide the policies and plaintexts from the servers. Along with the increasing number of attributes, the ciphertext size grew as well. To solve this problem, Herranz et al. gave a method to keep the size constant. In the situation of the attribute universe was certain [15].

The access structure adopted in the schemes before was monotonic, so Ostrovsky et al. tried to construct an attribute-based encryption scheme which is nonmonotonic, which was proven to be secure based on Decisional Bilinear Diffie-Hellman (DBDH) assumption [16]. Compared with the previous work, Lewko et al. constructed a scheme that had been proven to be fully secure rather than selectively secure in [17].

In [7, 18], KP-ABE and CP-ABE are proposed, respectively. In KP-ABE, the ciphertext's encryption policy is also associated with a set of attributes, but the attributes are organized into a tree structure (named access tree). In CP-ABE, the data owner constructs the access tree using visitors' identity information. The user can decrypt the ciphertext only if attributes in his private key match the access tree. Both became the important branches of attribute-based encryption. In [19, 20] Attrapadung et al. adopted the non-monotonic access structure to realize key-policy attribute-based encryption; what is more, the size of ciphertext was designed to be constant. In [21–24], the ciphertext-policy attribute-based encryption (CP-ABE) scheme with constant size ciphertexts for threshold predicates is proposed.

CP-ABE is a promising research area that attracts more and more attention from lots of researchers. Ibraimi et al. constructed a mediated CP-ABE that provided attributes revocation in [25]. Similar to KP-ABE, the researches on multiauthority in CP-ABE are quite a lot. Li et al. proposed a multiauthority CP-ABE that allowed tracing the misbehaving users; although only the AND gates were supported, it extended the application of CP-ABE to some extent [26]. Further improvements were made in this respect [27, 28].

In [29], with the case of the personal health record, Li et al. realized the security and scalable and fine-grained access control, supporting the modification of access policies and revocation of attributes. Liu et al. assumed that each attribute had different importance and constructed schemes that supported the access structure with different weights in CP-ABE [30] and KP-ABE [31]. Liu et al. added traceability to an existing expressive, efficient, and secure CP-ABE scheme without weakening its security, and change in the length of

the ciphertext and decryption key does not cause too much overhead [32]. Then they realized the White-Box Traceable CP-ABE in a large universe and the storage for traitor tracing is constant [33]. In [34], Goyal et al. allowed the access structure to be represented by an access tree with a bounded size access tree with threshold gates as its nodes. All of these existing schemes enhance the function of the original CP-ABE [7] to adapt to different scenarios. In this paper, we will introduce a new scheme to raise the adaptability of original CP-ABE in VCC.

3. Preliminaries

3.1. Bilinear Maps. Let G_0 and G_1 be two cyclic multiplicative groups of prime order p and g be the generator of G_0.

The bilinear map $e, e : G_0 \times G_0 \to G_1$, for all $a, b \in \mathbb{Z}_p$ is as follows:

(1) Bilinearity: $\forall u, v \in G_1, e(u^a, v^b) = e(u, v)^{ab}$

(2) Nondegeneracy: $e(g, g) \neq 1$

(3) Symmetry: $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$

3.2. Complexity Assumption. The Discrete Logarithm (DL) problem is defined as follows.

Let G be a multiplicative cyclic group of prime order p and g be its generator. DL problem is to compute $x \in \mathbb{Z}_p$ such that $y = g^x$, given $y \in_R G$ as input.

The DL hardness assumption holds, if no probabilistic polynomial time algorithm can solve the DL problem.

3.3. Access Structure. Let $P = \{P_1, P_2, \ldots, P_n\}$ be a set of participants; let $U = 2^{\{P_1, P_2, \ldots, P_n\}}$ be the universal set. If $\exists \text{AS} \subseteq U \setminus \{\varnothing\}$, then AS can be viewed as an access structure.

If $A \in \text{AS}$, $\forall B \in U$, $A \subseteq B$, and $B \in \text{AS}$, then AS is monotonic. Then sets in AS are defined as authorized sets, while the other sets are regarded as unauthorized sets.

In this paper, we construct an access tree $\mathcal{T}$ to represent the access structure. All the leaves represent the attributes, while the interior nodes represent the threshold gates. Before encrypting the data, we randomly choose a secret s and generate a polynomial for each interior node from top to bottom to share this secret.

To retrieve the secret, we define the Lagrange coefficient $\Delta_{i,S}$.

For $i \in \mathbb{Z}_p$ and for $\forall x \in S$,

$$\Delta_{i,S(x)} = \prod_{j \in S, j=i} \frac{x - j}{i - j}. \quad (1)$$

Only the authorized sets can succeed in decryption with polynomial interpolation.

4. The Condition

Definition 1 (condition primitives c). The condition primitives refer to the attributes in the access tree that are not closely related to the users' identities.

Definition 2 (condition C). The condition is a set of condition primitives, by which a specific access tree can be identified.

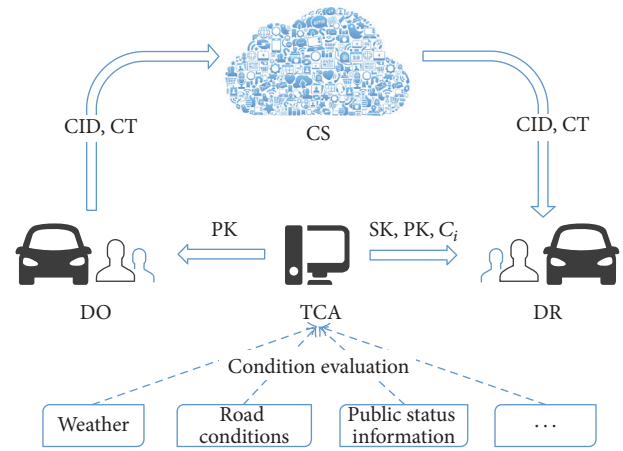

Figure 2: System model.

In VCC, condition primitives can be the external objective factors, such as the weather, the traffic, and the status information released by traffic control department. Comparing with the attributes related to the user's identities, we do not have to be concerned that the condition primitives may lead to user privacy disclosure. In our proposed scheme, the condition primitives will be extracted from each access tree to form the corresponding condition. In other words, a condition corresponds to a specific access tree.

Trust Center Authority (TCA) is in charge of evaluating the current conditions and sends them to data users. Condition C_i may consist of several components: $C_i : \{c_a, c_b, c_c, \ldots\}$. Each element is a condition primitive, which corresponds to a specific value that is randomly generated when the system is set up. Once generated, all the values are fixed and different from each other. In our scheme, we consider the relation of these conditions that belong to the same access structure to be AND. We multiply these corresponding values to denote the current condition.

With the support of the condition, when a data owner encrypts the data, more than one access tree can be added to the original ciphertext. When a data user requests data decryption, the current condition should be checked firstly to get the corresponding access tree, and then data decryption can be continued.

5. The Proposed System

5.1. System Model. In our system, four entities are included, namely, cloud servers (CS), data owners (DOs), data receivers (DRs), and Trust Center Authority (TCA), as shown in Figure 2.

Generally, cloud servers are assumed to be semitrusted. We employ them to be in charge of storing our encrypted data. Data owners and data receivers are either vehicle or nonmobile users. The former ones decide the access policies and the corresponding conditions, outsourcing their storage to CS after encrypting. The latter one submits the requests to CS and obtains their secret keys that are related to the attributes from TCA. Only when their attributes satisfy the access policies of data can they correctly decrypt the

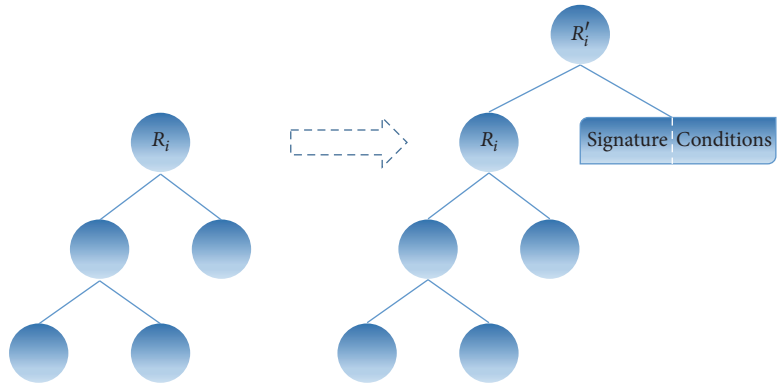

Figure 3: Structure of access tree in this paper.

ciphertext. TCA takes responsibility for evaluating DRs and assigns a set of attributes to DRs. In addition, conditions will be managed, determined, and finally transmitted to DRs by TCA.

In this paper, we allow DOs to add extra access trees and conditions to their own ciphertexts. Each access tree is related to one or several conditions. Only when its corresponding condition is satisfied can it be valid.

5.2. Security Model. The following is a security game between adversary $\mathcal{A}$ and challenger $\mathcal{C}$.

Init. $\mathcal{A}$ first chooses a challenge access structure AS^* and sends it to $\mathcal{C}$.

Setup. $\mathcal{C}$ runs this algorithm and gives the public parameters PK to the adversary $\mathcal{A}$.

Phase 1. $\mathcal{A}$ issues queries for repeated private keys corresponding to sets of attributes $S_1, \ldots, S_{q_1}$ (q and q_1 are integers that are randomly chosen by $\mathcal{A}$ and $1 < q_1 < q$).

(i) If any of the sets $S_1, \ldots, S_{q_1}$ satisfies the access structure AS^*, then it is aborted.

(ii) Else, $\mathcal{C}$ generates the corresponding secret keys to the sets for $\mathcal{A}$.

Challenge. $\mathcal{A}$ submits two equal length messages M_0 and M_1 to $\mathcal{C}$. $\mathcal{C}$ randomly flips a coin b and encrypts M_b under the challenge access structure AS^*. Then the generated ciphertext CT^* will be given to $\mathcal{A}$.

Phase 2. Repeat Phase 1, and the sets are turned from $S_1, \ldots, S_{q_1}$ to $S_{q_1+1}, \ldots, S_q$.

Guess. The adversary $\mathcal{A}$ outputs a guess b' of b.

The advantage of an adversary $\mathcal{A}$ in this game is defined as $\mathrm{Adv}(\mathcal{A}) = |\Pr[b' = b] - 1/2|$.

5.3. Our Construction. Our construction is based on the ciphertext-policy attribute-based encryption in [7]. In this section, we will describe the details of each algorithm.

5.3.1. Setup. This setup algorithm will choose a bilinear group G_0 of prime order p with generator g. Then, it will choose several random exponents: $\alpha, \beta, k, q \in \mathbb{Z}_p$. Let U denote the total number of attributes. Let V denote the total number of conditions. The algorithm randomly generates $u_1, u_2, \ldots, u_V \in \mathbb{Z}_p$ for each condition. We introduce hash functions H_{att}, $H_{\check{C}}$, and enough H_i for the access trees in ciphertext. The public key and the master key are published as

$$\begin{aligned} \mathrm{PK} &= \left\{G_0, g, h = g^{\beta}, e(g,g)^{\alpha}, g^{k\alpha}, g^{u_1}, g^{u_2}, \ldots, g^{u_V}\right\}, \\ \mathrm{MK} &= \left\{\beta, g^{\alpha}, k, q\right\}. \end{aligned} \tag{2}$$

5.3.2. Encryption. As shown in Figure 3, the original access tree is rooted at node R_i (i denotes the id of the access tree), and the new root node is R_i'.

At first, DO calls the data encryption subroutine to encrypt the plaintext into ciphertexts under access policies expressed in access tree structure. Compared with [7], we add a new root node and insert an extra node that denotes condition and its signature.

(i) Enc$(PK, M, \widetilde{T}_i, C_i) \rightarrow CID, CT$. The encryption subroutine takes as inputs public key PK, message M, access structure $\widetilde{T}_i$, symmetric encryption keys K_1 and K_2, and condition C_i and then outputs ciphertext CT and its CID. In this algorithm, it requires several steps to get CT being generated properly.

① *Encrypt Data*. DO receives message M, condition C_i, and one access policy. The encryption algorithm encrypts M under the access structure $\widetilde{T}_i$. From root node R_i' to the subtree rooted at R_i, this encryption algorithm is similar to that Bethencourt et al. described in [7].

First, the algorithm chooses a random number $s_i \in \mathbb{Z}_p$ for root node R_i', which means $q_{R_i'}(0) = s_i$. Then it generates polynomial for each interior node and computes (let Y be the set of leaf nodes in $\widetilde{T}_i$)

$$\begin{aligned} \widetilde{C}_i &= Me(g,g)^{\alpha s_i}, \\ C_i &= h^{s_i}, \\ \widehat{C}_{i,y} &= g^{q_{i,y}(0)}, \\ \widehat{C}'_{i,y} &= H_{\mathrm{att}}(\mathrm{att}(y))^{q_{i,y}(0)}, \\ &\qquad \forall y \in Y_i. \end{aligned} \tag{3}$$

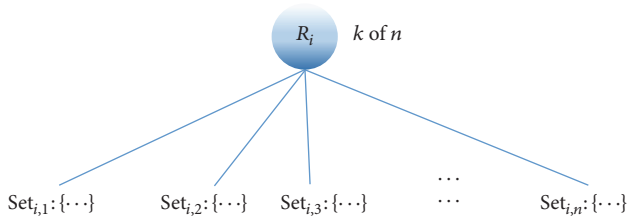

Figure 4: Partition of the set.

② *Generate CID.* Simultaneously,

$$\begin{aligned} \text{Set } P_1 &= g^{s_i/k}, \\ P_2 &= \text{Enc}_{K_1}(P_1), \\ \text{Enc}_{K_2}(P_2) &= \text{CID}. \end{aligned} \tag{4}$$

When DO would like to add an extra access structure $\widetilde{T}_{i+1}$ to his ciphertext, he first searches for the ciphertext from CS according to the CID and reencrypts it rather than reencrypting its original plaintext. What is different from before is that the part of the ciphertext associated with M is no longer involved in encryption. Thus, the computation burden is reduced.

Equally, the algorithm selects a random number $s_{i+1} \in \mathbb{Z}_p$ for the new access tree $\widetilde{T}_{i+1}$ and computes according to CID:

$$\widetilde{C}_{i+1} = \text{Dec}_{K_2}(\text{CID})\, e(g,g)^{\alpha s_{i+1}} = P_2 e(g,g)^{\alpha s_{i+1}}. \tag{5}$$

M is encrypted under only one of the access trees; we generally set $\widetilde{C}_1 = Me(g,g)^{\alpha s_1}$.

③ *Generate a Signature and Share It.* Obviously, the threshold relation of R_i' is AND. Suppose that the polynomial of R_i' is $q_{i,R'} = ax + s_i$ (a is randomly chosen by the algorithm). ESP randomly chooses a number $t_i \in \mathbb{Z}_p$ and makes the following calculation:

$$\begin{aligned} \text{Set } w_i &= \frac{t_i}{q}, \\ \text{Sig}_i &= \frac{g^{(2a+s_i)/t_i}}{g^{(u_a+u_b+u_c\cdots)/t_i}} = g^{(2a+s_i-(u_a+u_b+u_c\cdots))/t_i}. \end{aligned} \tag{6}$$

Sig_i can be viewed as a signature from DO. Assume the threshold relation of the node R_i is k of n; we first share Sig_i with (k,n) secret sharing scheme in [35] and then generate n pairs of $(x_{i,j}, y_{i,j})_{1\le j\le n}$.

As we know, the node R_i holds n branches, and each branch holds several attributes. Thus, all the attributes can be divided into n disjoint sets that are shown in Figure 4.

We distribute $(x_{i,j}, y_{i,j})_{1\le j\le n}$ to the sets that are shown as follows.

Distribution of the shares:

$$\begin{aligned} (x_{i,1}, y_{i,1}) &\longleftarrow \text{Set}_{i,1} : \{\cdots\} \\ (x_{i,2}, y_{i,2}) &\longleftarrow \text{Set}_{i,2} : \{\cdots\} \\ (x_{i,3}, y_{i,3}) &\longleftarrow \text{Set}_{i,3} : \{\cdots\} \\ &\vdots \quad\longleftarrow\quad \vdots \\ (x_{i,n}, y_{i,n}) &\longleftarrow \text{Set}_{i,n} : \{\cdots\}. \end{aligned} \tag{7}$$

We introduce a one-dimensional array $A_i[u]_{1\le u\le U}$ for each access tree $\widetilde{T}_i$ and assign each attribute a unique number from 1 to U. According to the shares and sets, we set

$$\begin{aligned} A_i[1] &= H_i(\text{att}_1) \oplus (x_{i,1} \parallel y_{i,1}) \quad (\text{att}_1 \in \text{Set}_{i,1}), \\ A_i[2] &= H_i(\text{att}_2) \oplus (x_{i,1} \parallel y_{i,1}) \quad (\text{att}_2 \in \text{Set}_{i,1}), \\ &\vdots \\ A_i[m] &= H_i(\text{att}_m) \oplus (x_{i,2} \parallel y_{i,2}) \quad (\text{att}_m \in \text{Set}_{i,2}), \\ A_i[m+1] &= H_i(\text{att}_{m+1}) \oplus (x_{i,2} \parallel y_{i,2}) \\ &\qquad (\text{att}_{m+1} \in \text{Set}_{i,2}), \\ &\vdots \\ A_i[U] &= H_i(\text{att}_U) \oplus (x_{i,n} \parallel y_{i,n}) \quad (\text{att}_U \in \text{Set}_{i,n}). \end{aligned} \tag{8}$$

④ *Record the Corresponding Condition.* Condition C_i associated with this tree is $\{c_a, c_b, c_c, \ldots\}$. In our system, we consider that one condition term only relates to one access tree. For each $c \in C_i$, the algorithm computes

$$\begin{aligned} \widehat{C}_{i,c} &= g^{u_c/t_i}, \\ \check{C} &= c_a \parallel c_b \parallel c_c \cdots, \\ \check{C} \oplus (i \parallel w_i) &\longleftarrow H_{\check{c}}(\check{C} \oplus \text{CID}). \end{aligned} \tag{9}$$

Let Y_i be the set of leaf nodes in $\mathcal{T}$ and C be the set of conditions. Finally, the complete ciphertext is as follows:

$$\begin{aligned} \text{CT} = \Big(& \widetilde{T}_1, \widetilde{C}_1 = Me(g,g)^{\alpha s_1},\ C_1 = h^{s_1}, A_1,\ \widehat{C}_{1,y} \\ & = g^{q_{1,y}(0)},\ \widehat{C}'_{1,y} = H(\text{att}(y))^{q_{1,y}(0)},\ \forall y \in Y_1,\ \widehat{C}_{1,c} \\ & = g^{u_c/t_1},\ \forall c \in C,\ \widetilde{T}_2, \widetilde{C}_2 = P_2 e(g,g)^{\alpha s_2},\ C_2 \\ & = h^{s_2}, A_2,\ \widehat{C}_{2,y} = g^{q_{2,y}(0)},\ \widehat{C}'_{2,y} \\ & = H(\text{att}(y))^{q_{2,y}(0)},\ \forall y \in Y_2,\ \widehat{C}_{2,c} = g^{u_c/t_2},\ \forall c \\ & \in C, \ldots \Big). \end{aligned} \tag{10}$$

5.3.3. Key Generation. DR should legally register to the attribute authorities, which will assign some attributes to this DR. Before decryption, the authorities will generate a corresponding SK for DR based on his attributes. The algorithm is as follows.

(i) KeyGen$(PK, MSK, S) \to SK$. This algorithm will take a set of attributes S as input and output a key that identifies with that set. First, a random $r \in \mathbb{Z}_p$ was chosen for the key and random $r_j \in \mathbb{Z}_p$ for each attribute $j \in S$. Then it computes the key as

$$\begin{aligned} \mathrm{SK} = \Big(D = g^{(\alpha+r)/\beta},\ \widehat{D} = g^{rq},\ D_j = g^r H(j)^{rj},\ D_j' \\ = g^{rj},\ \forall j \in S\Big). \end{aligned} \tag{11}$$

5.3.4. Decryption. When there is a DR requirement to decrypt a ciphertext, TCA will first evaluate the current condition and send the current condition term $C_0 = \{c_a, c_b, c_c, \ldots\}$ to DR.

(a) Query_Tree_id$(C_0, CID) \to i, w_i$. DR gets condition term from TCA, according to the current conditions and CID that is requested by DR, computing as follows:

$$\begin{aligned} \check{C} &= c_a \parallel c_b \parallel c_c \cdots \\ H_{\check{c}}\left(\check{C} \oplus \mathrm{CID}\right) &\longrightarrow \check{C} \oplus (i \parallel w_i) \\ \check{C} \oplus H_{\check{c}}\left(\check{C} \oplus \mathrm{CID}\right) &\longrightarrow (i \parallel w_i). \end{aligned} \tag{12}$$

This algorithm gets the corresponding access tree id i and a parameter w_i.

(b) Compute_Condition$(i, C_0, SK, CT) \to Con_i$. For each $c_j \in C_0$,

$$\begin{aligned} &\mathrm{Set}\ e\left(\widehat{C}_{i,j}, \widehat{D}\right) = e\left(g^{u_j/t_i}, g^{rq}\right) = e(g,g)^{rqu_j/t_i}, \\ &\mathrm{Con}_i = \left(e(g,g)^{rqu_a/t_i}\, e(g,g)^{rqu_b/t_i}\, e(g,g)^{rqu_c/t_i} \cdots\right) \\ &\quad = e(g,g)^{rq\sum u_j/t_i}. \end{aligned} \tag{13}$$

(c) Retrieve_Sig$(S, CT, PK) \to Sig_i$ *or* $\perp$. For each $\mathrm{att}_u \in S$, it computes

$$A_i[u] \oplus H_i(\mathrm{att}_u) = x_i \parallel y_i. \tag{14}$$

For each computation, the algorithm gets a pair of (x, y) that is shown as follows.

Retrieve the shares:

$$S \begin{cases} \left.\begin{array}{l} \mathrm{att}_1 \\ \mathrm{att}_2 \\ \mathrm{att}_3 \end{array}\right\} & \{\cdots\} : \mathrm{Set}_{i,1} \longrightarrow (x_{i,1}, y_{i,1}) \\ & \{\cdots\} : \mathrm{Set}_{i,2} \\ \mathrm{att}_4 & \{\cdots\} : \mathrm{Set}_{i,3} \longrightarrow (x_{i,3}, y_{i,3}) \\ \mathrm{att}_5 & \vdots \longrightarrow \vdots \\ \left.\begin{array}{l} \vdots \\ \vdots \\ \mathrm{att}_u \end{array}\right\} & \{\cdots\} : \mathrm{Set}_{i,n} \longrightarrow (x_{i,n}, y_{i,n}). \end{cases} \tag{15}$$

As we first share Sig_i with (k, n) secret sharing scheme, DR performs the polynomial interpolation and retrieves Sig_i only based on no less than k different pairs of (x, y). Otherwise, it outputs $\perp$.

(d) Dec$(S, CT, i, w_i, Con_i, Sig_i) \to M$ *or* $\perp$. Once w_i, Sig_i, and Con_i are retrieved, DR computes

$$\begin{aligned} &e\left(\mathrm{Sig}_i, \widehat{D}\right)^{w_i} \mathrm{Con}_i^{w_i} = e\left(g^{(2a+s_i-(u_a+u_b+u_c\cdots))/t_i}, g^{rq}\right)^{t_i/q} \\ &\cdot e(g,g)^{(rq\sum u_j/t_i)(t_i/q)} = e(g,g)^{r(2a+s_i)}. \end{aligned} \tag{16}$$

For the access tree rooted at node R_i', the algorithm can leverage the Lagrange interpolation and get $A = e(g,g)^{rs_i}$. Then it computes

$$B = \frac{\widetilde{C}_i A}{e(C_i, D)}. \tag{17}$$

If $i = 1$,

$$\begin{aligned} B = \frac{\widetilde{C}_1 A}{e(C_1, D)} &= \frac{Me(g,g)^{\alpha s_1}\, e(g,g)^{rs_1}}{e\left(h^{s_1}, g^{(\alpha+r)/\beta}\right)} \\ &= \frac{Me(g,g)^{\alpha s_1}\, e(g,g)^{rs_1}}{e(g,g)^{s_1(a+r)}} = M. \end{aligned} \tag{18}$$

Else,

$$\begin{aligned} B = \frac{\widetilde{C}_i A}{e(C_i, D)} &= \frac{P_2 e(g,g)^{\alpha s_i}\, e(g,g)^{rs_i}}{e\left(h^{s_i}, g^{(\alpha+r)/\beta}\right)} \\ &= \frac{P_2 e(g,g)^{\alpha s_i}\, e(g,g)^{rs_i}}{e(g,g)^{s_i(a+r)}} = P_2. \end{aligned} \tag{19}$$

The algorithm decrypts P_2 with K_1

$$P_1 = \mathrm{Dec}_{K_1}(P_2) = g^{s_1/k} \tag{20}$$

and recovers M by computing

$$M = \frac{\widetilde{C}_1}{e\left(g^{k\alpha}, g^{s_1/k}\right)} = \frac{Me(g,g)^{\alpha s_1}}{e(g,g)^{\alpha s_1}}. \tag{21}$$

Otherwise, it outputs $\perp$.

6. Security Analysis

We make the traditional CP-ABE more expressive by allowing DOs to add extra access trees and conditions as they like. To reduce the computation cost and storage overhead, we replace M with CID within the reencryption. And before decryption, condition values are to be computed first.

The modification of the ciphertexts and the way of decryption may affect the security of the system.

Theorem 3. *If hash function H is collision resistant, our system is secure against chosen plaintext attack in random oracle model.*

Proof. Since ciphertexts are CID : CT, data that is exposed to the adversary is

$$\begin{aligned} \widetilde{T}_i, \widetilde{C}_i &= P_m e(g,g)^{\alpha s_1}, \\ C_i &= h^{s_i}, A_i, \\ \widehat{C}_{1,i} &= g^{q_{i,y}(0)}, \\ \widehat{C}'_{i,y} &= H(\mathrm{att}(y))^{q_{i,y}(0)}, \\ &\qquad \forall y \in Y_i, \end{aligned} \tag{22}$$

where $P_0 = M$.

We then use a random function f to replace the hash function H. Therefore, the adversary $\mathscr{A}$ can obtain

$$\begin{aligned} \widetilde{T}_i, \widetilde{C}_i &= P_m e(g,g)^{\alpha s_1}, \\ C_i &= h^{s_i}, A_i, \\ \widehat{C}_{1,i} &= g^{q_{i,y_j}(0)}, \\ \widehat{C}'_{i,y} &= f\left(\mathrm{att}\left(y_j\right)\right)^{q_{i,y_j}(0)}, \\ &\qquad \forall y_j \in Y_i, \end{aligned} \tag{23}$$

where $P_0 = M$. This scheme is named as the alternative scheme.

Finally, we construct an experiment to simulate the chosen plaintext attack here.

(1) $\mathscr{A}$ calls the encryption oracle to query a cipher for plaintext y_j in the probabilistic polynomial time; we run the alternative scheme and return $f(\mathrm{att}(y_j))^{q_{i,y_j}(0)}$, as the cipher, to the adversary.

(2) $\mathscr{A}$ chooses two messages y_j^1 and y_j^2, and we return $f(\mathrm{att}(y_j^1))^{q_{i,y_j^1}(0)}, f(\mathrm{att}(y_j^2))^{q_{i,y_j^2}(0)}$ to the adversary.

(3) $\mathscr{A}$ asks a challenge plaintext; we flip a coin to generate a random number y_j^b, $b \in \{0,1\}$, and send $f(\mathrm{att}(y_j^b))^{q_{y_j^b}(0)}$ to the adversary.

(4) $\mathscr{A}$ continues to query some plaintexts which is the same as the first step.

(5) Finally, adversary $\mathscr{A}$ outputs b'; it wins if $b = b'$. Otherwise, it fails.

There are two ways to win the experiment for the adversary.

(1) s_j can be recovered. This contradicts DL problem.

(2) There must exist two y and y', such that $f(\mathrm{att}(y))^{q_y(0)} = f(\mathrm{att}(y'))^{q_{y'}(0)}$. The probability is negligible as f is a random function.

Above all, the adversary is negligible to win in the simulation experiment.

Furthermore, since the hash function cannot be distinguished with a random function in the random oracle model, the proposed system is secure to resist chosen plaintext attack.

The advantage of an adversary $\mathscr{A}$ is defined as

$$\begin{aligned} \mathrm{Adv}(\mathscr{A}) = {} & \Pr[H \text{ Collision}] + \Pr\left[\text{Recover } s_j\right] \\ & + \Pr[D \text{ wins}]. \end{aligned} \tag{24}$$

We define $\mathrm{negl}(k)$ as a function that is negligible with a security parameter k. Assume that the probability that H collision occurs is $\Pr[H \text{ Collision}]$. Since H is collision resistant, $\Pr[H \text{ Collision}]$ is $\mathrm{negl}_1(k)$. D is used to distinguish f with a random function. Since f is a pseudorandom function, therefore $\Pr[D \text{ wins}]$ is $\mathrm{negl}_2(k)$. Since it is computationally infeasible to solve DL problem, $\Pr[\text{Recover } s_j]$ is $\mathrm{negl}_3(k)$. In sum, $\mathrm{Adv}(\mathscr{A}) \le \mathrm{negl}_1[k] + \mathrm{negl}_2[k] + \mathrm{negl}_3[k]$.

We complete the proof. □

7. Performance Evaluation

In this section, we analyze our proposed scheme numerically, mainly discussing the computation and storage overhead.

7.1. Computation Overhead

7.1.1. Setup. This algorithm is used to define a cyclic multiplicative group and to generate a PK and MK that will be used in encryption and key generation especially. The number of random numbers is fixed, which means that there is no relationship between computing time and number of attributes. Time complexity of the algorithm is $O(1)$.

7.1.2. Encrypt. DO first encrypts M under an access tree, and the computation cost is proportional to the number of attributes in this tree. If the universal attributes set in $\mathscr{T}$ is I ($|I|$ denotes the total number of attributes in set I), for each element in I, we need 2 exponentiation operations. Hence, total computation complexity is $O(|I|)$. Additionally, in C-CP-ABE, one ciphertext is allowed to be associated with more than one access tree; if there are m access trees, for each tree, computation cost is proportional to the number of attributes, and the total computation complexity is $mO(|I|)$.

7.1.3. KeyGen. This algorithm is used to generate SK for DR. Computing cost is proportional to the number of attributes in SK. For each attribute, the algorithm requires 2 pairing operations and 1 multiplication operation. If the universal attributes set is S ($|S|$ is the total number of attributes in set, $|S| \le |I|$), then the time complexity of SK computation is $O(|S|)$.

7.1.4. Decrypt. Computing and checking the conditions are the first step within decryption. TCA evaluates the current conditions and sends them to DR. The computation cost is fixed, and the computation complexity is $O(1)$. Once the valid

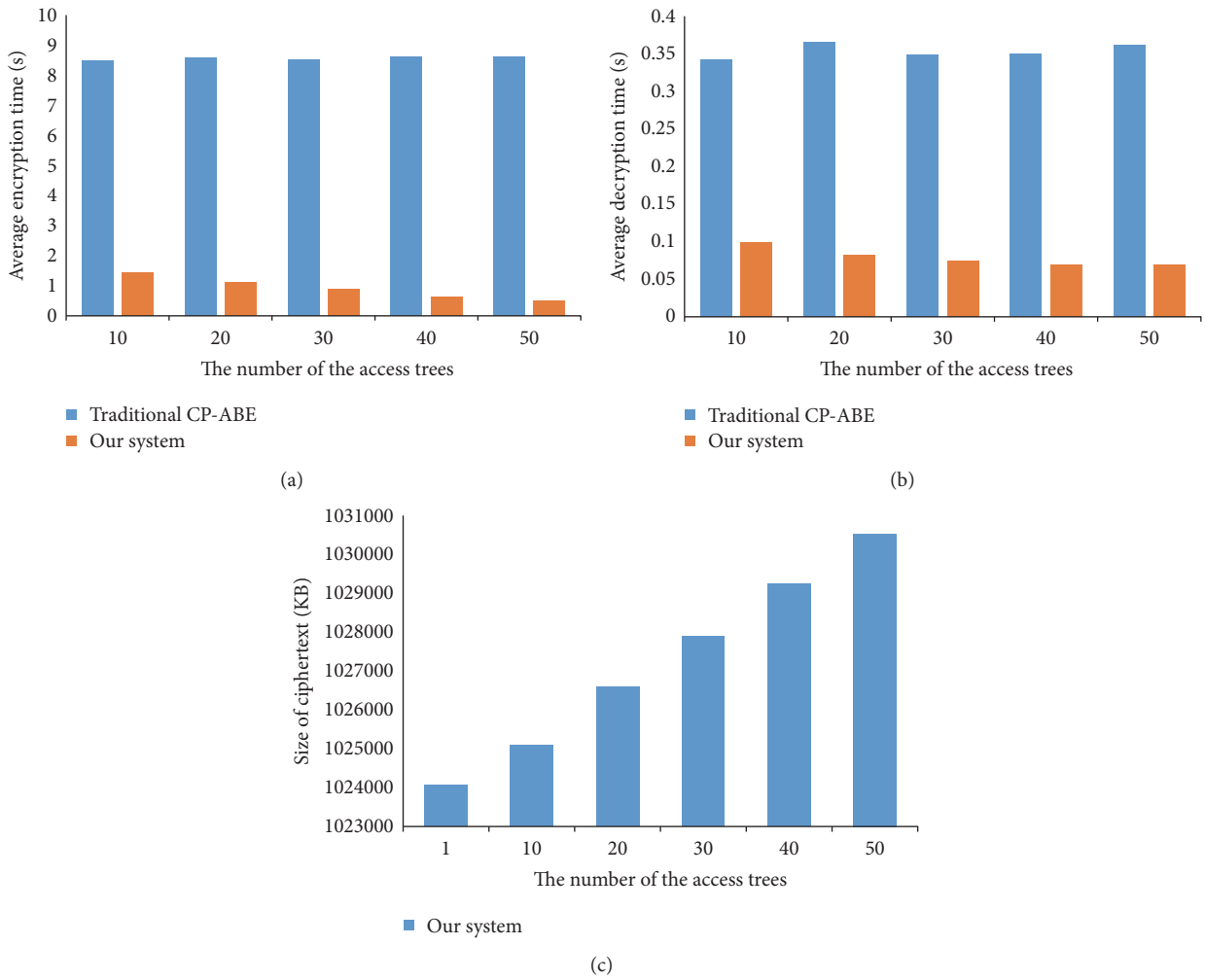

Figure 5: (a) Comparison of the average encryption times between CPABE and our system, when the number of access trees is different. (b) Comparison of decryption times between CPABE and our system, when the number of access trees is different. (c) Changes in the ciphertext size of our system, when the number of access trees is different.

tree is found, the following steps are similar to CP-ABE; the overhead mainly occurs during computing each attribute of the tree. Total cost is proportional to the number of attributes in the tree. Thus, the complexity is $O(|I|)$.

7.2. Storage Overhead. To realize more expressive access control, more storage cost has been inevitably introduced. One ciphertext is associated with more than one access tree. Only one tree's secret multiplies by M, while others multiply by a parameter. Along with the increasing number of access trees, the size of ciphertext grows. However, it is of low storage overhead compared with the component $\widetilde{C} = Me(g, g)^{\alpha s}$ in CT.

In addition, we create a one-dimensional array for each access tree in CT, whose length is equal to the number of the tree's leaf nodes. The total storage cost would not be a significant burden for CS.

7.3. Experimental Results. With the help of the CPABE-toolkit [36], we evaluate the performance of our system and compare it with CP-ABE [7].

In order to strengthen the expression of access structure, we make the ciphertext to be associated with more than one access tree. Compared with only one access tree in the traditional CPABE, introducing more trees may affect the time of encryption and decryption. However, we encrypt the message only once when it has more than one access tree; the other trees are multiplied with a parameter that includes decryption information. We set the size of the message as 1 G and the number of attributes in each tree as 100.

For the purposes of comparison, all the access trees have the same number of leaves nodes. Figure 5(a) shows the average time cost of [7] and our system within encryption. Figure 5(b) shows the average time cost of [7] and our system within decryption. From Figures 5(a) and 5(b), we can conclude that the average computation overheads in our system are lower than that in [7].

When considering that there is only one ciphertext, introducing extra access trees is bound to cause the storage overhead. In traditional CP-ABE scheme, more trees mean more ciphertexts, while in our system one ciphertext is

associated with a few access trees. In this case, it is obvious that our system reduces the storage overhead since the other trees are multiplied by a parameter rather than a message. Figure 5(c) shows that along with the increasing number of access trees the ciphertext size of our system grows.

Obviously, there exists a rough linear rise for the ciphertext size with the number of access trees, but even then it does not bring a high storage overhead when compared with the original ciphertext.

8. Conclusion

In this paper, we propose an expressive and fine-grained access control scheme C-CP-ABE in Vehicle Cloud Computing, making it possible that one ciphertext can be able to be associated with more than one access tree, different access tree under different conditions. DOs are allowed to add new access trees and new conditions to their ciphertexts flexibly. And a parameter that is calculated by ESP using s replaces M, which can reduce the computation and storage overhead when adding other access trees. The detailed security and performance analysis have been stated. There are some failings in our system, such as conditions not being flexible enough. All of them will be our future work.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was jointly supported by the National Natural Science Foundation of China (nos. 61402171, 61271512, and 61300177) and the Fundamental Research Funds for the Central Universities (2016MS29).

References

[1] M. Wang, H. Shan, R. Lu, R. Zhang, X. Shen, and F. Bai, "Real-Time path planning based on hybrid-VANET-enhanced transportation system," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 5, pp. 1664–1678, 2015.

[2] J. Shao, X. Lin, R. Lu, and C. Zuo, "A threshold anonymous authentication protocol for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1711–1720, 2016.

[3] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.

[4] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2706–2716, 2016.

[5] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *Journal of Network and Computer Applications*, vol. 40, no. 1, pp. 325–344, 2014.

[6] E. Lee, E.-K. Lee, M. Gerla, and S. Y. Oh, "Vehicular cloud networking: architecture and design principles," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 148–155, 2014.

[7] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on Security and Privacy (SP '07)*, pp. 321–334, IEEE, Berkeley, Calif, USA, 2007.

[8] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel & Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[9] Z. Zhou, Y. Wang, Q. M. J. Wu, C.-N. Yang, and X. Sun, "Effective and efficient global context verification for image copy detection," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 48–63, 2017.

[10] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, pp. 47–53, Springer, Berlin, Germany, 1984.

[11] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22–26, 2005. Proceedings*, vol. 3494 of *Lecture Notes in Computer Science*, pp. 457–473, Springer, Berlin, Germany, 2005.

[12] M. Chase, "Multi-authority attribute based encryption," in *Proceedings of the Theory of Cryptography, Theory of Cryptography Conference (TCC '07)*, pp. 515–534, Amsterdam, The Netherlands, February 2007.

[13] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, pp. 121–130, ACM, Chicago, Ill, USA, November 2009.

[14] A. Kapadia, P. P. Tsang, and S. W. Smith, "Attribute-based publishing with hidden credentials and hidden policies," in *Proceedings of the Network and Distributed System Security Symposium (NDSS '07)*, pp. 179–192, San Diego, Calif, USA, February-March 2007.

[15] J. Herranz, F. Laguillaumie, and C. Ràfols, "Constant size ciphertexts in threshold attribute-based encryption," in *Public Key Cryptography—PKC 2010*, P. Q. Nguyen and D. Pointcheval, Eds., vol. 6056 of *Lecture Notes in Computer Science*, pp. 19–34, Springer, Berlin, Germany, 2010.

[16] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures ," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 195–203, Alexandria, Va, USA, November 2007.

[17] A. Lewko, T. Okamoto, A. Sahai et al., "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology—EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings*, vol. 6110 of *Lecture Notes in Computer Science*, pp. 62–91, Springer, Berlin, Germany, 2010.

[18] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 89–98, November 2006.

[19] N. Attrapadung, B. Libert, and E. D. Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *Public Key Cryptography—PKC 2011: 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6–9, 2011. Proceedings*, vol. 6571 of *Lecture Notes in Computer Science*, pp. 90–108, Springer, Berlin, Germany, 2011.

[20] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. de Panafieu, and C. Ràfols, "Attribute-based encryption schemes with constant-size ciphertexts," *Theoretical Computer Science*, vol. 422, pp. 15–38, 2012.

[21] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in *Information Security Practice and Experience*, F. Bao, H. Li, and G. Wang, Eds., vol. 5451 of *Lecture Notes in Computer Science*, pp. 13–23, Springer, Berlin, Germany, 2009.

[22] Y. Zhang, D. Zheng, X. Chen et al., "Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts," in *Provable Security: 8th International Conference, ProvSec 2014, Hong Kong, China, October 9-10, 2014. Proceedings*, vol. 8782 of *Lecture Notes in Computer Science*, pp. 259–273, Springer, Berlin, Germany, 2014.

[23] C. Chen, Z. Zhang, and D. Feng, "Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost," in *Provable Security*, X. Boyen and X. Chen, Eds., vol. 6980 of *Lecture Notes in Computer Science*, pp. 84–101, Springer, Berlin, Germany, 2011.

[24] A. Ge, R. Zhang, C. Chen et al., "Threshold ciphertext policy attribute-based encryption with constant size ciphertexts," in *Proceedings of the Australasian Conference on Information Security and Privacy*, pp. 336–349, Wollongong, Australia, July 2012.

[25] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application," in *Information Security Applications*, H. Y. Youm and M. Yung, Eds., vol. 5932 of *Lecture Notes in Computer Science*, pp. 309–323, Springer, Berlin, Germany, 2009.

[26] J. Li, Q. Huang, X. Chen, S. S. M. Chow, D. S. Wong, and D. Xie, "Multi-authority ciphertext-policy attribute-based encryption with accountability," in *Proceedings of the ACM Symposium on Information, Computer and Communications Security (ASI-ACCS '11)*, pp. 386–390, ACM, Hong Kong, March 2011.

[27] Z. Liu, Z. Cao, Q. Huang et al., "Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles," in *Computer Security—ESORICS 2011: 16th European Symposium on Research in Computer Security, Leuven, Belgium, September 12–14,2011. Proceedings*, vol. 6879 of *Lecture Notes in Computer Science*, pp. 278–297, Springer, Berlin, Germany, 2011.

[28] H. Wang, Z. Zheng, L. Wu, and D. He, "New large-universe multi-authority ciphertext-policy ABE scheme and its application in cloud storage systems," *Journal of High Speed Networks*, vol. 22, no. 2, pp. 153–167, 2016.

[29] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel & Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2013.

[30] X. Liu, J. Ma, J. Xiong, Q. Li, and J. Ma, "Ciphertext-policy weighted attribute based encryption for fine-grained access control," in *Proceedings of the 5th IEEE International Conference on Intelligent Networking and Collaborative Systems (INCoS '13)*, pp. 51–57, September 2013.

[31] X. Liu, H. Zhu, J. Ma, J. Ma, and S. Ma, "Key-Policy Weighted Attribute based Encryption for fine-grained access control," in *Proceedings of the IEEE International Conference on Communications Workshops (ICC '14)*, pp. 694–699, Sydney, Australia, June 2014.

[32] Z. Liu, Z. Cao, and D. S. Wong, "White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 76–88, 2013.

[33] Z. Liu, Z. Cao, and D. S. Wong, "Traceable CP-ABE: how to trace decryption devices found in the wild," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 55–68, 2015.

[34] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Automata, Languages and Programming*, pp. 579–591, Springer, 2008.

[35] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[36] J. Bethencourt, A. Sahai, and B. Waters, "The cpabe toolkit," http://acsc.csl.sri.com/cpabe/.