IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 29, 2018, accepted October 16, 2018, date of publication October 22, 2018,
date of current version November 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2877177

A Compact Ciphertext-Policy Attribute-Based Encryption Scheme for the Information-Centric Internet of Things

JING WANG[1,2], NEAL NAIXUE XIONG[3], JINHAI WANG[4], AND WEI-CHANG YEH[5]

[1]School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510275, China
[2]School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
[3]Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA
[4]College of Electronic and Information Engineering, Foshan University, Foshan 528000, China
[5]Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 30013, Taiwan

Corresponding author: Jing Wang (wangj478@mail.sysu.edu.cn)

This work was supported by the National Natural Science Foundation of China under grant No. 61802083.

ABSTRACT The information-centric Internet of things (IC-IoT) is different from the traditional Internet of things (IoT) in that the device-to-device pattern is generalized to a device-to-network pattern. Furthermore, in an IC-IoT environment, there is a demand for protecting the security of all data generated from IC-IoT devices. A cryptography scheme named attribute-based encryption (ABE) represents a smart method of providing the fine-grained access control that can sufficiently protect data security. The most attractive advantage of ABE is its expressive access policy, which makes the access control of data flexible and manageable. However, there is a serious problem caused by such an access policy; it incurs a greater ciphertext redundancy and computational overhead. This implies that the current ABE scheme is hard to implement in the thin client devices of IC-IoT. In this paper, we propose a universalized policy-compacting method via sharing public parts of the policy. Compared with the original policy, the compacted policy applies a more compact ciphertext and requires less computation, communication, and storage cost. However, the policy-compacting problem is proved to be a non-deterministic polynomial complete (NPC) problem. Thus, a greedy algorithm is provided to obtain an approximate minimum compacted policy scale. Finally, we propose a compact ciphertext-policy attribute-based encryption (CCP-ABE) scheme with the policy-compacting method. A security proof and performance evaluation show that the proposed CCP-ABE scheme provides a comprehensive performance improvement.

INDEX TERMS Information-centric internet of things (IC-IoT), access control, attribute-based encryption (ABE), policy compacting.

I. INTRODUCTION

The rise of the information-centric Internet of things (IC-IoT) indicates a pattern change of information exchange and content sharing. Furthermore, the data security demand of IC-IoT is different from traditional IoT [1], [2]. It requires protecting the privacy of the shared content in an IC-IoT network [3]–[5]. Data access control is an effective way to support secure data sharing. The traditional access control mechanism requires a delegation administrator to manage access privilege, and the security of such a mechanism depends entirely on the administrator. However, the highly pervasive and distributed IC-IoT environment presses for a more scalable and flexible access control mechanism [6]. Fortunately, attribute-based encryption (ABE), as a security cryptosystem, can provide fine-grained ciphertext access control for IC-IoT. Different from other kinds of cryptography, such as symmetric and asymmetric cryptography, ABE supports a one-to-many encryption pattern. This implies that a ciphertext of ABE can be decrypted by a set of different secret keys, which improves the scalability and flexibility of ABE. In brief, the core properties of an ABE ciphertext access control mechanism are: (1) the access control of data is maintained by a data owner (i.e., a device of the IC-IoT) instead of the storage service provider (i.e., another device of the IC-IoT); (2) the access privilege of users (i.e., all devices of IC-IoT) is described by an access policy, which is more

intuitive and readable; and (3) the security property of ABE is derived from cryptography. Therefore, the ABE ciphertext access control mechanism provides several attractive advantages over other access control mechanisms, and it is more suitable for IC-IoT.

The most attractive advantage of ABE is its flexible and expressive access policy, which is used to describe fine-grained data access privilege. Furthermore, the access policy can be managed in a scalable way [33]. However, the complex policy also incurs large ciphertext redundancy. This implies that the large-scale ciphertext of ABE always results in a large overhead: (1) a large computational overhead during encryption and decryption; (2) a high communication overhead during ciphertext uploading and downloading; and (3) a massive storage overhead. As is well known, in IC-IoT, there are a lot of thin client devices with limited resources, and the overhead of ABE is too heavy for these devices. Thus, an effective way of reducing ciphertext redundancy is necessary to improve the existing ABE scheme.

The most significant challenge of low-overhead ABE research is reducing ciphertext redundancy without sacrificing additional performance. In order to reduce ciphertext redundancy, Herranz *et al*. [17] and Chen *et al*. [18] proposed constant ciphertext-policy ABE (CP-ABE) schemes, but these schemes were only provided with an expression-limited policy (i.e., AND gate access structure or a threshold function). In order to reduce the computation of client cost by large-scale ciphertext, Hohenberger and Waters [25] and Lai *et al*. [26] provided outsourced computation ABE schemes. In these schemes, most of the computation of encryption or decryption is outsourced to the service providers of the network. Thus, such schemes have a high communication overhead. In order to reduce the policy scale, Zhou and Huang [28] and Song *et al*. [29] provided a minimum sum of product expression (minimum SOPE) and minimum linear code, respectively, to minimize the policy scale. However, although a small-scale policy has less ciphertext redundancy, the reduction of redundancy is limited and unstable. Thus, it is hard to propose an ideal method to reduce the ciphertext redundancy of ABE without sacrificing performance.

In this work, we propose a compact ciphertext-policy ABE (CCP-ABE) scheme to compact the policy scale and reduce ciphertext redundancy. As is well known, there are two kinds of ciphertexts in the CP-ABE scheme: *data ciphertext* and *attribute ciphertext*, which are associated with data and attributes, respectively. Furthermore, the number of *attribute ciphertexts* increases with the scale of the access policy. Thus, policy-compacting, which decreases the policy scale, is an effective way to reduce ciphertext redundancy. In our proposed CCP-ABE, all *attribute ciphertexts* are divided into two categories: public and private attribute ciphertext units. Different from the private unit, the public unit is shared by multiple parties of the access policy. This implies that multiple private units can be merged as one public unit, and the multiple policy parties associated with the same public unit can be compacted as one. As a result, ciphertext redundancy is reduced by merging ciphertext and compacting policy. Although a cross-utilization public unit could trigger a risk of data leak, our CCP-ABE scheme provides effective protection to avoid such a risk, with only little additional storage overhead.

The main contributions of this paper are summarized as follows:

(1) We propose a CCP-ABE scheme to reduce the ciphertext redundancy by sharing public parties of the access policy and public attribute ciphertext units.

(2) Two metrics, the flexible factor and overlap factor, are provided to evaluate the policy-compacting efficiency and compact ratio. Thus, the reduction of ciphertext redundancy is more intuitional and measurable.

(3) The policy-compacting problem is proven to be an non-deterministic polynomial complete (NPC) problem, and thus, a greedy compacting algorithm is provided to obtain the approximate minimum compact-policy and ciphertext scales.

The remaining of this paper is organized as follows. Section II gives the related work. Then, we propose the policy compacting method in Section III and present the CCP-ABE scheme in Section IV, respectively. Thirdly, we analyse the performance of the proposed compacted policy and CCP-ABE scheme in Section V. Finally, the conclusions is given in Section VI.

II. RELATED WORK

Substantial changes have occurred in information technology (IT), which have also brought various challenges to information security [9]–[11]. For instance, IC-IoT provides a novel data sharing method which is different from traditional IoT. However, data security and privacy become critical issues that restrict IC-IoT development [12], [13]. This is because, in the open access environment, it is hard for the data owner to prevent sensitive data leakage, which incurs a serious security risk to IC-IoT [14]–[16].

An access control mechanism provides an efficient way to protect the data security of IC-IoT. Significantly, scalability and flexibility are two important properties for an IC-IoT data access control mechanism [6]. In the traditional access control mechanism, there is a central organization responsible for managing data access privilege [6]–[8]. This implies that such mechanisms are always restricted in scalability. Fortunately, ABE provides a novel data access control mechanism, where its security depends on cryptography instead of a central privileged organization. Thus, its scalability is effectively improved. Furthermore, ABE provides a novel one-to-many encrypting pattern, which is different from other cryptographies [19]. It is well-suited for the device-to-network pattern in the IC-IoT network. Specifically, ABE can provide a fined-grained ciphertext access control mechanism for IC-IoT by using an expressive access policy. Such an access policy results in high flexibility of the ABE access control mechanism. Furthermore, in a CP-ABE scheme, the access privilege of data is described by an access policy which is derived from

TABLE 1. Comparison of existing low-cost schemes.

Scheme		Text Overhead			Access Policy	
		Computation	Communication	Storage	Form	Expression
Constant Ciphertext	Constant-Ciphertext [23]	reduced	reduced	reduced	AND Gate	limited
	Threshold CP-ABE [24]	reduced	reduced	reduced	Threshold function	limited
Computation Outsourcing	ABOOE[1][25]	reduced	increased	-	LSSS[4]	monotone
	Outsourced Decrypting [26]	reduced	increased	-	LSSS	monotone
	DAC-MACS[2][27]	reduced	increased	-	LSSS	monotone
Policy Minimizing	Minimum SOPE[3][28]	reduced	reduced	reduced	Boolean formula	monotone
	Minimum linear code [29]	reduced	reduced	reduced	LSSS	monotone
Proposed CCP-ABE		reduce	reduce	reduce	access tree/LSSS/Boolean formula[5]	monotone

[1] ABOOE: Attribute-Based Online/Offline Encryption.
[2] DAC-MACS: Effective Data Access Control for Multi-authority Cloud Storage Systems
[3] Minimum SOPE: Minimum Sum-of-Product Expression.
[4] LSSS: Linear Secret Sharing Scheme.
[5] For simplicity, only the compated access tree is given in this work. However, other compacted policy forms can be obtained by the policy transforming method of reference [33].

a secret sharing scheme [34]. A wide range of studies have been conducted to design secret sharing schemes, such as the schemes proposed in references [35]–[43] The access policy is often expressed in various forms, such as the monotonic Boolean formula, an access tree, or a linear secret sharing scheme (LSSS). In brief, the access policy of ABE is diverse, which makes the ABE access control mechanism more scalable and flexible [33].

However, the access policy incurs larger costs of computation, communication, and storage, which limits its commercial applications [30]–[32]. Thus, many researchers focus on the low-cost ABE schemes and low-cost applications of ABE [20]–[22]. As shown in Table 1, there are three ways to achieve low-cost ABE schemes, as recently reported: constant-ciphertext setting, computation outsourcing, and policy minimizing. In a constant-ciphertext ABE scheme [23], [24], the access policy is expressed as an AND gate or a threshold function. Although this simple access policy reduces the resource costs of clients, it is also limited at the expression. In order to reduce client computation cost, some ABE schemes provide outsourcing of the decrypting function [25]–[27]. Although the computation cost of the client is reduced, the communication overhead is increased. The other efficient way to reduce resources of the ABE scheme is by minimizing the access policy. Minimal sum-of-product expression (minimum SOPE) [28] and minimum linear code [29] schemes are provided to minimize policy size without breaking policy logic. This implies that the system overhead and policy performance of these schemes are all optimized. It seems that minimum policy ABE is optimal, as the system overheads are all reduced and the access policy is strong at expression. However, the performance of minimum policy ABE is limited and unstable.

Considering the insufficiencies of the above low-cost ABE schemes, we propose CCP-ABE, which has the advantage of comprehensive performance. It provides an efficient and stable way to reduce the ciphertext redundancy of ABE without any additional restrictions or costs. Furthermore, the policy-compacting method can also be universally used to reduce the overhead of various existing ABE schemes.

III. COMPACTED POLICY FOR CIPHERTEXT-POLICY ATTRIBUTE-BASED ENCRYPTION

ABE can be viewed as a tuple $\{E_M, E_A, \mathcal{A}, Share\}$, where E_M and E_A are encryption algorithms, $\mathcal{A}$ denotes the attribute set, and *Share* is a secret sharing scheme (SSS). In most CP-ABE schemes, plaintext M is encrypted with a random secret s as $\mathbf{C_0} = E_M(M, s)$. Then, s is divided into a set of shares $S = \{s_1, s_2, \ldots, s_n\}$ by *Share*. Finally, each share is encrypted with an attribute public key (PK) as $\mathbf{C_i} = E_A(s_i, PK_{\rho(i)})$, where ρ denotes a map from the labels $\{1, 2, \ldots, n\}$ to $\mathcal{A}$, and $PK_{\rho(i)}$ denotes the PK of $\rho(i) \in \mathcal{A}$. Thus, the whole ciphertext is expressed as $CT = \{\mathbf{C_0}, \mathbf{C_1}, \ldots, \mathbf{C_n}\}$. Significantly, the size of CT is linearly increased with the size of S.

Assume that a data owner encrypts a set of data and generates multiple attribute ciphertext units associated with the same attribute. If these units are also assigned with the same share, they can be compacted as one unit, which is called the *public attribute ciphertext* in this work. In this vein, we present a method to compact the share set and reduce the ciphertext redundancy of ABE.

A. ACCESS POLICY

The access policy, also called the access structure, is a core concept of ABE. The formal definition of the access structure is given as follows.

Definition 1 (Access Structure [19]): Let $\{P_1, P_2, \ldots, P_n\}$ be a set of parties.[1] An access structure is a collection $\mathcal{A}$ of non-empty subsets of $\{P_1, P_2, \ldots, P_n\}$. The sets in $\mathcal{A}$ are called the authorized set, and the sets not in $\mathcal{A}$ are called the unauthorized sets. Furthermore, an access structure $\mathcal{A}$ is monotonic if $\forall B, C$: $B \in \mathcal{A} \wedge B \subset C \rightarrow C \in \mathcal{A}$.

There are three common access policy forms: access tree, LSSS matrix, and monotonous Boolean expression [33]. For simplicity, we only discuss the access tree in this work. An access tree is special kind of tree structure. Let $\mathcal{T}$ be an access tree. Each non-leaf node $\mathfrak{n}$ of $\mathcal{T}$ is a $t_\mathfrak{n}$-out-of-$n_\mathfrak{n}$ node (i.e., threshold), where $n_\mathfrak{n}$ denotes the number of its children, $t_\mathfrak{n}$ denotes its threshold, and $0 \leq t_\mathfrak{n} \leq n_\mathfrak{n}$. Each leaf $\mathfrak{n}$ of $\mathcal{T}$

[1] In the ABE context, the role of the parties is taken by the attributes.

is described by an attribute $\rho(\mathfrak{n})$ and the threshold $t_{\mathfrak{n}} = 1$. Then, according to Shamir's secret sharing scheme [35], each node $\mathfrak{n}$ is assigned with a node polynomial $f_{\mathfrak{n}}(x)$, such that:

$$f_{\mathfrak{n}}(0) = \begin{cases} s, & \mathfrak{n} \text{ is the root of } \mathcal{T} \\ f_{\mathfrak{p}}(x_{\mathfrak{n}}), & \text{Otherwise,} \end{cases} \tag{1}$$

$$d_{\mathfrak{n}} = t_{\mathfrak{n}} - 1, \tag{2}$$

where s denotes the secret, $\mathfrak{p}$ denotes the parent of $\mathfrak{n}$, $x_{\mathfrak{n}}$ denotes the interpolation of $\mathfrak{n}$, and $d_{\mathfrak{n}}$ denotes the degree of $f_{\mathfrak{n}}(x)$. Finally, $s_{\mathfrak{n}} = f_{\mathfrak{n}}(0)$ is called the node share of $\mathfrak{n}$.

Furthermore, the secret s (i.e., node share of the root of $\mathcal{T}$) can be reconstructed by an authorized attribute set $\mathcal{A}$. The reconstruction is processed recursively as follows. If $\mathfrak{n}$ is a leaf, $s_{\mathfrak{n}}$ can be reconstructed if and only if $\rho(\mathfrak{n}) \in \mathcal{A}$. Otherwise, $\mathfrak{n}$ is a non-leaf, and the reconstruction of $s_{\mathfrak{n}}$ is successful if and only if at least $t_{\mathfrak{n}}$ node shares of its children are successfully reconstructed. Let $N_{\mathfrak{n}}$ be a $t_{\mathfrak{n}}$-sized child set of $\mathfrak{n}$, where all $\mathfrak{c}_i \in N_{\mathfrak{n}}$ have successfully been reconstructed with node share $s_{\mathfrak{c}_i}$. The share of $\mathfrak{n}$ can be calculated as follows:

$$s_{\mathfrak{n}} = \sum_{\mathfrak{c} \in N_{\mathfrak{n}}} s_{\mathfrak{c}_i} \Delta_{i,X_{\mathfrak{n}}}(0), \tag{3}$$

$$\Delta_{i,X_{\mathfrak{n}}}(x) = \prod_{j \in X_{\mathfrak{n}}, j \neq i} \frac{x-j}{i-j}, \tag{4}$$

where $f_{\mathfrak{n}}(i) = s_{\mathfrak{c}_i}$, $X_{\mathfrak{n}} = \{i | \mathfrak{c}_i \in N_{\mathfrak{n}}\}$, and $\Delta_{i,X_{\mathfrak{n}}}(x)$ is called the Lagrange coefficient. The correctness proof is shown as follows:

$$\begin{aligned} \sum_{\mathfrak{c}_i \in N_{\mathfrak{n}}} s_{\mathfrak{c}_i} \Delta_{i,X_{\mathfrak{n}}}(0) &= \sum_{\mathfrak{c}_i \in N_{\mathfrak{n}}} f_{\mathfrak{n}}(i) \Delta_{i,X_{\mathfrak{n}}}(0) \\ &= f_{\mathfrak{n}}(0) \\ &= s_{\mathfrak{n}}, \end{aligned} \tag{5}$$

B. POLICY-COMPACTING PROBLEM

As is well known, a leaf of an access policy can be describe by a attribute and a share. Thus, multiple leaves can be compacted to one when such leaves are assigned with the same attribute and share. Figure 1 shows two examples of policy compacting. Multiple ordinary leaves of access trees can be compacted as one public leaf. In the single policy case, there are two leaves assigned with the attribute $\mathfrak{a}_2$. The two leaves can be compacted as one public leaf when s_3 equals s_2. Similarly, in the multi-policy case, four leaves of two policies can be compacted as two public leaves when $s_2 = s_4$ and $s_3 = s_5$. Furthermore, the scale of ABE ciphertexts is dependent on the number of leaf nodes. This implies that the multiple *private attribute ciphertext units* associated with the ordinary leaves can also be compacted as one *public attribute ciphertext unit* associated with the public leaf. Thus, the ciphertext scale can be effectively reduced by using the compacted access policies and the public ciphertext units.

However, there may be a risk of information leaking incurred by public ciphertext units. Suppose that an owner uploads his data M_1 and M_2, as shown in Figure 2. Let s_1 and

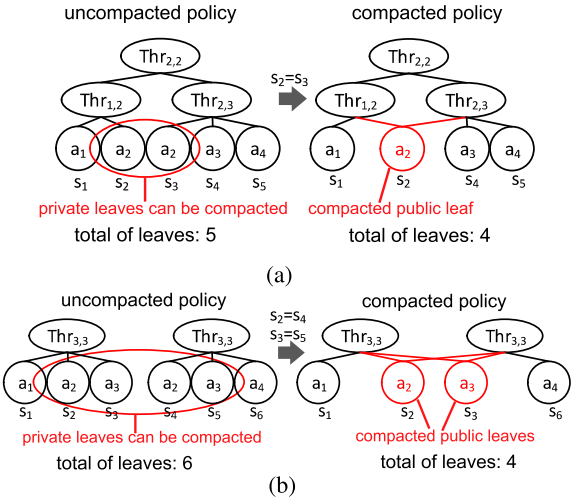

FIGURE 1. Examples of access policy compacting. ($Thr_{t,n}$: a t-out-of-n gate; a_j: leaf attribute index; s_i: leaf share.). (a) Single Policy Case. (b) Multi-Policy Case.

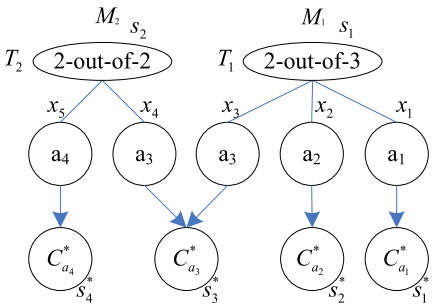

FIGURE 2. An example of public share leaking. (M_i: data; s_i secret of date; T_i: access policy; x_j node interpolation; a_j: leaf attribute index; s_j^*: leaf share; $C_{a_j}^*$: attribute ciphertext.)

s_2 be the secrets of M_1 and M_2, respectively; $\mathbb{T}_1$ and $\mathbb{T}_2$ are the access trees assigned to M_1 and M_2, respectively; and a user gets the attribute set $S_u = \{a_1, a_2, a_4\}$. It is clear that the user is not allowed to access M_2, because S_u is an unauthorized set of $\mathbb{T}_2$. However, in this case, the user can illegally access M_2 as follows. First, they recover shares s_1^*, s_2^*, and s_4^* via S_u. Then, they calculate:

$$s_3^* = \frac{x_2 - x_3}{x_2 - x_1} s_1^* + \frac{x_1 - x_3}{x_1 - x_2} s_2^*. \tag{6}$$

Finally,

$$s_2 = \frac{x_5}{x_5 - x_4} s_4^* + \frac{x_4}{x_4 - x_5} s_3^*. \tag{7}$$

As a result, they can successfully recover M_2 via the unauthorized S_u. In order to prevent the information from leaking, the interpolations x_3 and x_4 of public unit $C_{\mathfrak{a}_3}^*$ must be encoded by the associated attribute a_3. Thus, the user cannot get x_3 and x_4 unless the attribute $\mathfrak{a}_3 \in S_u$. Furthermore, s_3^* cannot be calculated by Equation (6), s_2 cannot be recovered by Equation (7), and M_2 cannot be illegally accessed.

The other challenge of policy compacting is constructing the optimal compacted policies, which correspond to the minimum ciphertext scale. The formal definition of the *optimal policy-compacting problem* is given as follows:
Input: $\langle \mathcal{T}, k \rangle$, where $\mathcal{T}$ is an access policy set, and $k \in \mathbb{Z}^+$.
Question: Does $\mathcal{T}$ have a valid share set $\mathcal{S}$ with size k?

Claim 1: The problem is non-deterministic polynomial (NP) hard.

Proof:

Let π be a function that assigns node shares for the access tree, $\mathcal{L}$ be a set, and element $\langle a, s \rangle \in \mathcal{L}$ be described by an attribute a and a share s. The following verifier of the problem runs in polynomial time of $|\mathcal{L}|$:

Verifier $V(\langle \mathcal{T}, k \rangle, \langle \pi, \mathcal{L} \rangle)$.

The verifier output is true if and only if all the following conditions are true:

- $|\mathcal{L}| \leq k$
- $\forall \mathfrak{n} \in \mathbb{T}$ and $\mathbb{T} \in \mathcal{T}$, $\pi(\mathfrak{n})$ must be calculated efficiently.
- $\forall$ leaf $\mathfrak{n} \in \mathbb{T}$ and $\mathbb{T} \in \mathcal{T}$, $\langle \rho(\mathfrak{n}), \pi(\mathfrak{n}) \rangle \in \mathcal{L}$, where $\rho(\mathfrak{n})$ denotes the attribute associated with $\mathfrak{n}$.

Claim 2: 3-satisfiability(3-SAT)$\leq_p$policy compacting.

Proof:

Define a function f with input φ and outputs $\langle \mathcal{T}, k \rangle$, where φ is an instance of 3-SAT and $\langle \mathcal{T}, k \rangle$ is an instance of policy-compacting. We now show that f is a polynomial-time function which converts the policy-compacting problem into a 3-SAT problem.

Let $X = \{x_1, \ldots, x_n\}$ denote the literals of φ, and $C = \{c_1, \ldots, c_m\}$ denote the clauses of φ. To justify this claim, suppose $k = 2n + 3$ and $A = \{y_1, z_1, \ldots, y_n, z_n\} \cup \{\omega, \varpi\}$ is the attribute set, where y_i represents x_i, z_i represents $\neg x$, and ω, ϖ denote the adding attributes.

For each $c_j = \alpha \vee \beta \vee \gamma \in C$, we get an access tree $\bar{c}_j$, as shown in Figure 3. Thus, $\mathcal{T} = \{\bar{c}_1, \ldots, \bar{c}_m\}$. Let $F \in \{0, 1\}^n$ denote an assignment of literal set X, and let F_i denote the value of x_i. If φ is satisfied, $\exists F$ that makes φ true. We choose $a, b \in Z$ at random and define a function as follows:

$$\tau(y_i) = \begin{cases} a, & F_i = 1 \\ b, & \text{otherwise}, \end{cases} \tag{8}$$

$$\tau(z_i) = \begin{cases} b, & F_i = 1 \\ a, & \text{otherwise}. \end{cases} \tag{9}$$

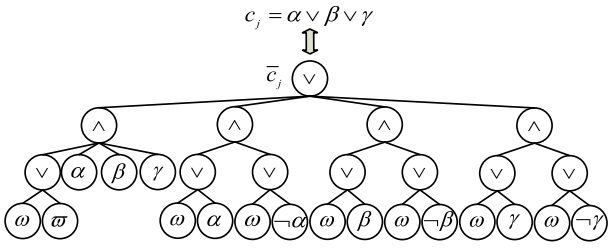

FIGURE 3. Clauses c_j transform into access policy $\bar{c}_j$. ($\wedge$: AND gate; $\vee$: OR gate; $\alpha, \beta, \gamma, \in X$; ω, ϖ: adding attributes.)

Let $f_1(x), f_2(x), f_3(x)$, and $f_4(x)$ be four non-constant functions assigned as node polynomials of $\bar{c}_j$. Then, an interpolation set $S_x = \{x_1, x_\alpha, x_\beta, x_\gamma\} \in Z_p^*$ is chosen at random, and $f_1(x)$ is subject to:

$$\begin{cases} f_1(x_1) = b \\ f_1(x_\alpha) = \tau(\alpha) \\ f_1(x_\beta) = \tau(\beta) \\ f_1(x_\gamma) = \tau(\gamma). \end{cases} \tag{10}$$

Then, $f_1(x)$ can be constructed as follows:

$$f_1(x) = b\Delta_{S_x,x_1}(x) + \sum_{i \in \{\alpha,\beta,\gamma\}} \tau(i)\Delta_{S_x,x_i}(x). \tag{11}$$

Let $f_2(x) = c_2 x + s$, where $c_2 \in Z_p$ and $s = f_1(0)$. There are two solutions, $x_{2,a} = (a - s)/c_2 \in Z_p$ and $x_{2,b} = (b - s)/c_2 \in Z_p$, of the equations $f_2(x_a) = a$ and $f_2(x_b) = b$, respectively. The node polynomials $f_3(x)$ and $f_4(x)$ are constructed in the same manner. The node shares of $\bar{c}_j$ are chosen as shown in Figure 4, and at most nine elements are added to $\mathcal{L}$:

$$\mathcal{L} \leftarrow \mathcal{L} \cup \{\langle \alpha, \tau(\alpha) \rangle, \langle \neg\alpha, \tau(\neg\alpha) \rangle, \langle \beta, \tau(\beta) \rangle, \langle \neg\beta, \tau(\neg\beta) \rangle, \langle \gamma, \tau(\gamma) \rangle, \langle \neg\gamma, \tau(\neg\gamma) \rangle, \langle \omega, a \rangle, \langle \omega, b \rangle, \langle \varpi, b \rangle\}.$$

There must be three solutions of the equation $f_1(x) = b$, otherwise the third-order polynomial function $f_1(x)$ degenerates to a constant function $f(x) = b$. Thus, the node shares shown in Figure 4 are valid if and only if $\bar{c}_j$ is true (i.e., at most two of the variables $\tau(\alpha), \tau(\beta), \tau(\gamma)$ are set to be b). Furthermore, all $\bar{c}_j, 1 \leq j \leq m$ can be assigned node shares as shown in Figure 4 if and only if F makes φ true. Finally, we find:

$$\mathcal{L} = \{\langle \omega, a \rangle, \langle \omega, b \rangle, \langle \varpi, b \rangle, \langle y_i, \tau(y_i) \rangle, \langle z_i, \tau(z_i) \rangle \, 1 \leq i \leq n\}.$$

Thus, $|\mathcal{L}| = 2n + 3$ in this case.

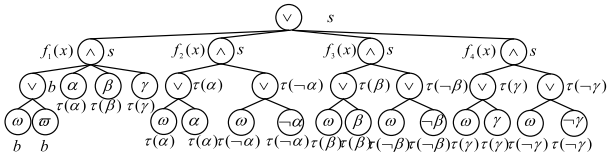

FIGURE 4. Choosing node shares of $\bar{c}_j$. (s: the secret; $f_i(x)$: node polynomial; $\tau(*)$: assignment function; $\wedge$: AND gate; $\vee$: OR gate; $\alpha, \beta, \gamma, \in X$; $\neg*$: negation of $*$; ω, ϖ: adding attributes.)

If φ is unsatisfied, $\forall F$, $\exists c_{k_1} = \alpha_{k_1} \vee \beta_{k_1} \vee \gamma_{k_1}$ and $c_{k_2} = \alpha_{k_2} \vee \beta_{k_2} \vee \gamma_{k_2}$, where $\alpha_{k_1}, \beta_{k_1}, \gamma_{k_1}$ are all false and $\alpha_{k_2}, \beta_{k_2}, \gamma_{k_2}$ are all true. Because φ is unsatisfied, $\exists c_{k_1}$ that is unsatisfied (which implies that $\alpha_{k_1}, \beta_{k_1}, \gamma_{k_1}$ are all false). If $\nexists c_{k_2}$, where $\alpha_{k_2}, \beta_{k_2}, \gamma_{k_2}$ are all true, then this implies $\forall c_j$ there is at least one false literal. Thus, $\bar{F}$ (the negation of F) is a satisfied assignment for φ.

Furthermore, $\bar{c}_{k_1}$ and $\bar{c}_{k_2}$ must be assigned node shares, as shown in Figure 5. Because the third-order polynomial function $f_{k_1,1}(x)$ has three solutions of $f_{k_1,1}(x) = b$ at most, the share $\langle a, \varpi \rangle$ must be added into $\mathcal{L}$. Similarly, $f_{k_2,1}(x)$ has

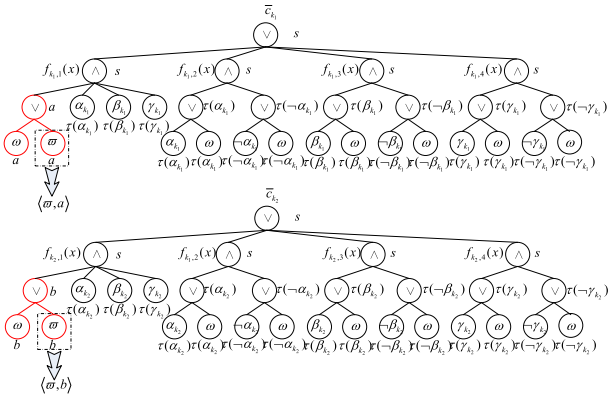

FIGURE 5. Choosing node shares of $\bar{c}_{k_1}$ and $\bar{c}_{k_2}$. (s: the secret; $f_{k_i,j}(x)$: node polynomial; $\tau(*)$: assignment function; $\wedge$: AND gate; $\vee$: OR gate; $\alpha_{k_i}, \beta_{k_i}, \gamma_{k_i} \in X$; $\neg*$: negation of $*$; ω, ϖ: adding attributes.)

TABLE 2. Notations.

Notation	Description
A	The attribute set.
$\mathfrak{a}_i$	The i^{th} element of A.
L_*	The leaf node set of access tree $*$ or access tree set $*$.
$\mathbb{C}$	The attribute ciphertext set.
ρ	$\rho : L \to A$.
ϕ	$\phi : L \to \mathbb{C}$.
$C^*_{\mathfrak{a}_i}$	The public attribute ciphertext unit associated with attribute $\mathfrak{a}_i$, there are two or more leaves $\mathfrak{n} \in L$ such that $C^*_i = \phi(\mathfrak{n})$.
$C_\mathfrak{n}$	The private attribute ciphertext associated with leaf node $\mathfrak{n}$.
$\lvert * \rvert$	The total of elements of set $*$.

only three solutions of $f_{k_2,1}(x) = a$, and share $\langle \varpi, b \rangle$ must be added into $\mathcal{L}$. Thus,

$$\mathcal{L} = \{\langle \omega, a \rangle, \langle \omega, b \rangle, \langle \omega, b \rangle, \langle \varpi, b \rangle, \langle y_i, \tau(y_i) \rangle, \langle z_i, \tau(z_i) \rangle \; 1 \le i \le n\}.$$

As a result, $|\mathcal{C}| = 2n + 4 > k$ when φ is unsatisfied.

Finally, we can get the result: 3-SAT$\le_p$ policy-compacting.

C. NOTATIONS AND DEFINITIONS

The notations used in describing CCP-ABE are given in Table 2. There are two important metrics immediately given, which are named the flexibility factor and overlap factor. Assume that $\mathbb{T}$ denotes an access tree and $\mathfrak{n} \in \mathbb{T}$ is a t-out-of-n node (non-leaf), where $t \le n$. Let set $N_\mathfrak{n} = \{\mathfrak{c}_i, 0 \le i \le n\}$, where $\mathfrak{c}_i$ for $i \ne 0$ denotes a child of $\mathfrak{n}$, and $\mathfrak{c}_0$ denotes $\mathfrak{n}$. The flexibility factor is defined as follows.

Definition 2 (Flexibility Factor $\gamma_\mathfrak{n}$ for Node $\mathfrak{n}$): The flexibility factor $\gamma_\mathfrak{n}$ denotes the total number of nodes in $N_\mathfrak{n}$ that can be assigned with random shares.

The upper limit of $\gamma_\mathfrak{n}$ is given as follows:

$$\bar{\gamma}_\mathfrak{n} = \begin{cases} t_\mathfrak{n}, & t_\mathfrak{n} \ne 2 \\ |N_\mathfrak{n}|, & t_\mathfrak{n} = 2. \end{cases} \tag{12}$$

Following the definition, for all subtrees $\mathbb{T}' \subset \mathbb{T}$, the following inequality must hold:

$$\sum_{\mathfrak{n} \in \mathbb{T}'} \gamma_\mathfrak{n} \le |\bigcup_{\mathfrak{n} \in \mathbb{T}'} N_\mathfrak{n}|. \tag{13}$$

Thus, $\gamma_\mathfrak{n} = \min\{\bar{\gamma}_\mathfrak{n}, |N_\mathfrak{n} - N'_\mathfrak{n}|\}$, where $N'_\mathfrak{n} = \{\mathfrak{c}_i | \gamma_{\mathfrak{c}_i} = |N_{\mathfrak{c}_i} - N'_{\mathfrak{c}_i}|, 1 \le i \le n_\mathfrak{n}\}$. The detailed proof of this equation is given in Appendix A.

Similarly, the overlap factor is defined as follows.

Definition 3 (Overlap Factor $\delta_\mathbb{T}$ for Access Tree $\mathbb{T}$): The overlap factor $\delta_\mathbb{T}$ of $\mathbb{T}$ denotes the total number of leaves of $\mathbb{T}$ that can be associated with public units.

We then get:

$$\delta_\mathbb{T} = \sum_{\mathfrak{n} \in \mathbb{T}} (\gamma_\mathfrak{n} - 1). \tag{14}$$

The validity of Equation (14) is proved in Appendix B.

Furthermore, the bilinear map plays a crucial role in ABE. The definition of a bilinear map is described as follows.

Definition 4 (Bilinear Map): Assume that G_0, G_T are two multiplicative cyclic groups with prime order p, and g is a generator of G_0. Function $e : G_0 \times G_0 \to G_T$ is a bilinear map if and only if it satisfies three criteria:

1) Bilinearity: $\forall u, v \in G_0$ and $a, b \in Z_p$, $e(u^a, v^b) = e(u, v)^{ab}$;

2) Non-degeneracy: $\forall u, v \ne g^0$,$e(u, v) \ne 1$;

3) Computability: e must be computed efficiently.

Additionally, there are some *hard problems* which support the security of the ABE mechanism. We introduce one of the *hard problems*–the decisional q-parallel bilinear Diffie-Hellman exponent assumption–to guarantee the security of our CCP-ABE scheme.

Assumption 1 (Decisional q-Parallel Bilinear Diffie-Hellman Exponent Assumption, q-Parallel BDHE): Assume G_0, G_T are two group with prime order p, and $e : G_0 \times G_0 \to G_T$ is a bilinear map. Let $\alpha, b_1, \ldots, b_q, s \in Z_p$ be chosen at random, and let g be a generator of G_0. If a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ is given:

$$\vec{y} = \{g, g^s, g^\alpha, \ldots, g^{\alpha^q}, g^{\alpha^{q+2}}, \ldots, g^{\alpha^{2q}}; \\ \forall 1 \le j \le q, g^{sb_j}, g^{\alpha/b_j}, \ldots, g^{\alpha^q/b_j}, g^{\alpha^{q+2}/b_j}, \ldots, g^{\alpha^{2q}/b_j}; \\ \forall 1 \le j, l \le q, l \ne j, g^{\alpha s b_l/b_j}, \ldots, g^{\alpha^q s b_l/b_j}\}.$$

then it must be hard to distinguish a valid element $T_0 = e(g, g)^{a^{q+1}s} \in G_T$ from a random element $T_1 = R \in G_T$.

Assume that $\mathcal{B}$ is a PPT algorithm with output $z \in \{0, 1\}$. We say that $\mathcal{B}$ gets the advantage ϵ in solving q-parallel BDHE, if:

$$|Pr[\mathcal{B}(\vec{y}, T = T_0) = 0] - Pr[\mathcal{B}(\vec{y}, T = T_1) = 0]| \ge \epsilon, \tag{15}$$

where $Pr[*]$ denotes the probability of event $*$. The decisional q-parallel BDHE assumption holds if and only if there is no PPT algorithm $\mathcal{B}$ that gets a non-negligible advantage ϵ in distinguishing the q-parallel BDHE tuple $\{\vec{y}, T\}$.

D. GREEDY POLICY-COMPACTING ALGORITHM

Before compacting access polices, we need to initialize the flexibility factor of each node via Algorithm 1. Let $\mathbb{T}$ be an access tree, and let $\mathfrak{r}$ be the root of $\mathbb{T}$. $Init_\gamma(\mathfrak{r})$ is run as a depth-first traversal of $\mathbb{T}$ and initializes the flexibility factor of all nodes of $\mathbb{T}$.

Algorithm 1 $Init_\gamma(\mathfrak{n})$

Require: node: $\mathfrak{n}$
Ensure: State flag: $\mu_\mathfrak{n} \in \{0, 1\}$

if $\mathfrak{n}$ is a leaf **then**
  return 0
else
  //Let $\mathfrak{n}$ be a $t_\mathfrak{n}$-out-of-$n_\mathfrak{n}$ node and $N_\mathfrak{n}$ be the child set of $\mathfrak{n}$
  $k_\mathfrak{n} \leftarrow 0$
  for $\mathfrak{c}_i \in N_\mathfrak{n} - \{\mathfrak{n}\}$ **do**
    $k_\mathfrak{n} \leftarrow k_\mathfrak{n} + Init(\mathfrak{c}_i)$
  end for
  if $t_\mathfrak{n} = 2$ **then**
    $\gamma_\mathfrak{n} \leftarrow \min\{n_\mathfrak{n}, |N_\mathfrak{n}| - k_\mathfrak{n}\}$
  else
    $\gamma_\mathfrak{n} \leftarrow \min\{t_\mathfrak{n}, |N_\mathfrak{n}| - k_\mathfrak{n}\}$
  end if
  if $\gamma_\mathfrak{n} = |N_\mathfrak{n}| - k_\mathfrak{n}$ **then**
    return 1
  else
    return 0
  end if
end if

Then, the algorithm $Update(\mathfrak{n})$ is called to update the flexibility factor of the access tree $\mathbb{T}$ when a node $\mathfrak{n} \in \mathbb{T}$ is assigned with node share. In this algorithm, three arrays $Count$, $\mathbb{C}$, and $\mathcal{S}$ are given to describe the public attribute ciphertext. For each attribute $\mathfrak{a}_i$, there are three related parameters $count_i \in Count$, $C^*_{\mathfrak{a}_i} \in \mathbb{C}$, and $s^*_i \in \mathcal{S}$. $count_i$ denotes the number of leaves $\mathfrak{n} \in \mathcal{T}$ associated with $\mathfrak{a}_i$. $C^*_{\mathfrak{a}_i}$ denotes the public ciphertext unit of $\mathfrak{a}_i$. s^*_i denotes the public share of $\mathfrak{a}_i$. Clearly, $C^*_{\mathfrak{a}_i}$ and s^*_i are all initialized to be *null* at the beginning. Then, $count_i$, $C^*_{\mathfrak{a}_i}$, $s^*_i \in \mathcal{S}$ are all updated when a leaf $\mathfrak{n} \in \mathbb{T}$ with attribute $\mathfrak{a}_i$ is assigned with a node share $s_\mathfrak{n}$. Finally, $Update(\mathfrak{n})$ is called iteratively to assign the share for $\mathfrak{n}$ and update the flexibility factor of part of the node in $\mathbb{T}$.

Suppose that $\mathcal{T} = \{\mathbb{T}_1, \mathbb{T}_2, \ldots, \mathbb{T}_m\}$ denotes an access tree set, and $Compact(\mathcal{T})$ is a greedy algorithm proposed for compact-policy set $\mathbb{T}$. First, array $Count$ is initialized. Then, $Compact(\mathcal{T})$ calls $Init_\gamma(\mathfrak{r}_j)$ for each root $\mathfrak{r}_j$ of $\mathbb{T}_j \in \mathcal{T}$ to initialize its flexibility factor. Third, each secret s_j of $\mathfrak{r}_j$ is assigned, and $Update_\gamma \mathfrak{r}_j$ is called to update the flexibility factor again. Finally, the attribute $\mathfrak{a}_i$ with max $count_i$ is chosen each time, and a share s^*_i is assigned for all nodes $\mathfrak{n}$ where $\rho(\mathfrak{n}) = \mathfrak{a}_i$ and $s_\mathfrak{n} = null$. In this step, for all leaves $\mathfrak{n}$, $\rho(\mathfrak{n}) = \mathfrak{a}_i$, $Update_\gamma(\mathfrak{n})$ is called to update their flexibility factors and assign their node shares.

Algorithm 2 $Update_\gamma(\mathfrak{n})$

Require: Node: $\mathfrak{n}$
Ensure: Node share: $s_\mathfrak{n}$ Updated node flexibility factor: $\gamma_\mathfrak{n}$

if $\mathfrak{n}$ is a leaf **then**
  if $s^*_{\rho(\mathfrak{n})} = null$ **then**
    if $s_\mathfrak{n} = null$ **then**
      $s_\mathfrak{n} \xleftarrow{R} Z_p$
      $\gamma_\mathfrak{n} \leftarrow \gamma_\mathfrak{n} - 1$
      $\gamma_\mathfrak{p} \leftarrow \gamma_\mathfrak{p} - 1$
      $Update_\gamma(\mathfrak{p})$
    end if
    $s^*_{\rho(\mathfrak{n})} \leftarrow s_\mathfrak{n}$
    $\phi(\mathfrak{n}) \leftarrow \&C^*_{\rho(\mathfrak{n})}$
  else
    if $s_\mathfrak{n} \neq null$ **then**
      $s_\mathfrak{n} \leftarrow s^*_{\rho(\mathfrak{n})}$
      $\phi(\mathfrak{n}) \leftarrow \&C^*_{\rho(\mathfrak{n})}$
      $\gamma_\mathfrak{n} \leftarrow \gamma_\mathfrak{n} - 1$
      $\gamma_\mathfrak{p} \leftarrow \gamma_\mathfrak{p} - 1$
      $Update_\gamma(\mathfrak{p})$
    else
      assign the storage unit for $C_\mathfrak{n}$
      $\phi(\mathfrak{n}) \leftarrow \&C_{\rho(\mathfrak{n})}$
      $Count_{\rho(\mathfrak{n})} \leftarrow Count_{\rho(\mathfrak{n})} - 1$
    end if
  end if
else
  if $\mathfrak{n}$ is root and $s_\mathfrak{n} = null$ **then**
    $s_\mathfrak{n} \xleftarrow{R} Z_p$
    $\gamma_\mathfrak{n} \leftarrow \gamma_\mathfrak{n} - 1$
  end if
  if $\gamma_\mathfrak{n} = 0$ and $f_\mathfrak{n} = null$ **then**
    //Let $N_\mathfrak{n}$ be the child set of $\mathfrak{n}$
    Calculate node polynomial function $f_\mathfrak{n}(x)$
    Assign interpolation for all $\mathfrak{c}_i \in N_\mathfrak{n}$
    if $s_\mathfrak{n} = null$ **then**
      $s_\mathfrak{n} \leftarrow f_\mathfrak{n}(0)$
      $\gamma_\mathfrak{p} \leftarrow \gamma_\mathfrak{p} - 1$
      $Update_\gamma(\mathfrak{p})$
    end if
    for $\mathfrak{c}_i \in N_\mathfrak{n}$ **do**
      if $s_{\mathfrak{c}_i} = null$ **then**
        $s_{\mathfrak{c}_i} \leftarrow f_\mathfrak{n}(s_{\mathfrak{c}_i})$
        $\gamma_{\mathfrak{c}_i} \leftarrow \gamma_{\mathfrak{c}_i} - 1$
        $Uptate_\gamma(\mathfrak{c}_i)$
      end if
    end for
  end if
end if

IV. COMPACT CIPHERTEXT-POLICY ATTRIBUTE-BASED ENCRYPTION SCHEME

As shown in Figure 6, the system model of the compact ciphertext-policy attribute-based encryption (CCP-ABE) scheme consists of four types of entities:

Algorithm 3 *Compact*($\mathcal{T}$)

Require: Access policy set: $\mathcal{T}$
Ensure: Node share set: $\mathcal{S}$
for leaf $\mathfrak{n} \in \mathcal{T}$ **do**
  $count_{\rho(\mathfrak{n})} \leftarrow count_{\rho(\mathfrak{n})} + 1$
end for
for $\mathbb{T}_i \in \mathcal{T}$ **do**
  $Init_\gamma(\mathfrak{r}_i)$ //$\mathfrak{r}_i$ denotes the root of $\mathbb{T}$
  $Uptate_\gamma(\mathfrak{r}_i)$
end for
while $\exists \mathfrak{a}_i, count_{\mathfrak{a}_i} \neq 0$ **do**
  //Let $N_{\mathfrak{a}_i}$ be the set of leaves assigned with attribute $\mathfrak{a}_i$
  while $N_{\mathfrak{a}_i} \neq \varnothing$ **do**
    Choose a node $\mathfrak{n} \in N_{\mathfrak{a}_i}$
    $Update_\gamma(\mathfrak{n})$
    $N_{\mathfrak{a}_i} \leftarrow N_{\mathfrak{a}_i} - \{\mathfrak{n}\}$
  end while
end while

Authority. The authority is responsible for generating the public key (PK), secret key (SK), and master key (MK).
Information center. The information center can be viewed as an abstract distributed cluster of the network. It is responsible for data storage in the IC-IoT network.
Owner. An owner denotes a device of IC-IoT that generates and uploads data to the information center. Note that data uploaded to the information center are all encrypted as ciphertexts by using the PK.
User. A user denotes a device that downloads ciphertext from the information center and recovers the according plaintext by its SK. Note that a user can also be an owner in this system.

Furthermore, the CCP-ABE scheme includes the following four functional modules:

Setup: The authority chooses $g_1, g_2 \in G_0, w, x \in Z_p$ and calculates $P = g_1^x$ and $\Upsilon = e(g_1, g_2)^w$, where G_0, G_T are two multiplicative cyclic groups with prime order p, and $e : G_0 \times G_0 \rightarrow G_T$ denotes a bilinear map. For each attribute $\mathfrak{a}_i \in A, 1 \leq i \leq N$, a random number $a_i \in Z_p$ is chosen and the public key $P_i = g_1^{xa_i}$ is computed. Then, PK and MK are shown as follows:

$$PK = \{g_1, g_2, \Upsilon, P, P_i\}_{1 \leq i \leq N},$$
$$MK = \{w, x\}.$$

KeyGen: A user sends their attribute set U_s to the authority to request their SK. First, the authority picks $u \in Z_p$ randomly. Then, $D'_u = g_2^{w-xu}$, $\bar{D}_u = g_2^u$, and $D_i = g_2^{uxa_i}$ are calculated, where $\mathfrak{a}_i \in U_s$. Finally, the SK $SK_u = \{D'_u, \bar{D}_u, D_i\}_{\mathfrak{a}_i \in U_s}$ is sent to the user.

Encryption: Let $\mathcal{M} = \{M_1, M_2, \ldots, M_n\}$ be a plaintext set and let $\mathcal{T} = \{\mathbb{T}_1, \mathbb{T}_2, \ldots, \mathbb{T}_n\}$ denote the according access tree set. The owner calls *Compact*($\mathcal{T}$) to compact access policies and assign shares. Then, the ciphertext is calculated in three steps. First, for each $M_j \in \mathcal{M}$, the data ciphertext is calculated:

$$C_{M_j} =< C_j = M_j \Upsilon^{s_j}, C'_j = g_1^{s_j} >, \tag{16}$$

where $s_j \in Z_p$. Second, for each private node $\mathfrak{n}$, the private ciphertext unit is calculated as follows:

$$C_\mathfrak{n} =< \bar{C}_\mathfrak{n} = g_1^{r_\mathfrak{n}}, \quad \hat{C}_\mathfrak{n} = P_{\rho(\mathfrak{n})}^{r_\mathfrak{n}} P^{s_\mathfrak{n}} >, \tag{17}$$

where $r_\mathfrak{n}, s_\mathfrak{n} \in Z_p$. Finally, for each public ciphertext $C^*_{\mathfrak{a}_i}$, we calculate:

$$C^*_{\mathfrak{a}_i} = < \bar{C}^*_i = g_1^{r_{\mathfrak{a}_i}}, \quad \hat{C}^*_i = P_{\mathfrak{a}_i}^{r_{\mathfrak{a}_i}} P^{s^*_{\mathfrak{a}_i}},$$
$$C'_i = g_1^{s'_{\mathfrak{a}_i}}, \quad \bar{C}'_i = g_1^{r'_{\mathfrak{a}_i}}, \ \hat{C}'_i = P_{\mathfrak{a}_i}^{r'_{\mathfrak{a}_i}} P^{s'_{\mathfrak{a}_i}} >, \tag{18}$$

where $r_{\mathfrak{a}_i}, s^*_{\mathfrak{a}_i}, r'_{\mathfrak{a}_i}, s'_{\mathfrak{a}_i} \in Z$. Additionally, for all nodes $\mathfrak{n}, \phi(\mathfrak{n}) = C^*_{\mathfrak{a}_i}$, the node interpolation $x_\mathfrak{n}$ is encoded as $\bar{x}_\mathfrak{n} = x_\mathfrak{n} \Upsilon^{s'_{\mathfrak{a}_i}}$.

Decryption: The user determines an authorized set $U' \subset U_s$ of $\mathbb{T}_j$, where U_s is their attribute set. For each private leaf node $\mathfrak{n} \in \mathbb{T}_j$, they calculate:

$$TK_\mathfrak{n} = \frac{e(\hat{C}_\mathfrak{n}, \bar{D}_u)}{e(\bar{C}_\mathfrak{n}, D_{\rho(\mathfrak{n})})} = e(g_1, g_2)^{xus_\mathfrak{n}}, \tag{19}$$

where $\rho(\mathfrak{n}) \in U'$. For each public leaf node $\mathfrak{n} \in \mathbb{T}_j$, the user calculates:

$$TK_\mathfrak{n} = \frac{e(\hat{C}^*_{\rho(\mathfrak{n})}, \bar{D}_u)}{e(\bar{C}^*_{\rho(\mathfrak{n})}, D_{\rho(\mathfrak{n})})} = e(g_1, g_2)^{xus^*_{\rho(\mathfrak{n})}} \tag{20}$$

$$x_\mathfrak{n} = \frac{\bar{x}_\mathfrak{n} e(\hat{C}'_{\rho(\mathfrak{n})}, \bar{D}_u)}{e(D'_u, C'_{\rho(\mathfrak{n})}) e(\bar{C}'_{\rho(\mathfrak{n})}, D_{\rho(\mathfrak{n})})}, \tag{21}$$

where $\phi(\mathfrak{n}) = C^*_{\mathfrak{a}_i}$. Then, all non-leaf nodes are processed as follows. Let $\mathfrak{n}$ be a t-out-of-n node, let $S_\mathfrak{n}$ be a child set of $\mathfrak{n}$, and $|S_\mathfrak{n}| = t$. Assume that $\forall \mu \in S_\mathfrak{n}$, TK_μ is obtained. The user computes $TK_\mathfrak{n}$ as follows:

$$\begin{aligned} TK_\mathfrak{n} &= \prod_{\mu \in S'_\mathfrak{n}} TK_\mu^{\Delta_{\mu, S_\mathfrak{n}}(0)} \\ &= \prod_{\mu \in S'_\mathfrak{n}} (e(g_1, g_2)^{\mu x s_\mu})^{\Delta_{i, S_\mathfrak{n}}(0)} \\ &= e(g_1, g_2)^{\mu x s_\mathfrak{n}}, \end{aligned} \tag{22}$$

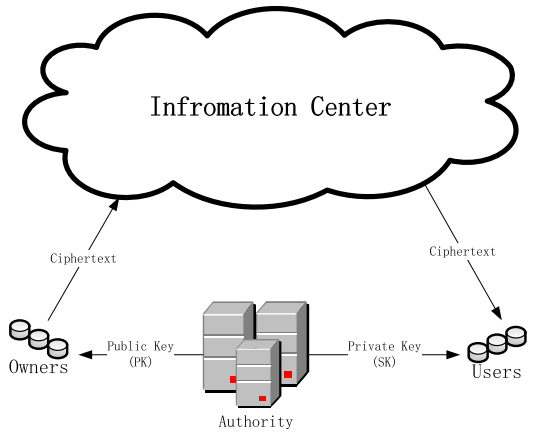

FIGURE 6. System model.

where $\Delta_{\mu,S_n}(0)$ is the Lagrange coefficient[2] of μ. The user recovers the plaintext when $TK_{\mathfrak{r}_j}$ of the access tree root $\mathfrak{r}_j$ is obtained:

$$\frac{C_j}{e(C'_j, D'_u)TK_{\mathfrak{r}_j}} = \frac{M_j e(g_1, g_2)^{ws_{\mathfrak{r}_j}}}{e(g_1, g_2)^{ws_{\mathfrak{r}_j} - xus_{\mathfrak{r}_j}} e(g_1, g_2)^{xus_{\mathfrak{r}_j}}} = M_j. \tag{23}$$

V. PERFORMANCE ANALYSIS

A. SECURITY PROOF

We prove that the security of our CCP-ABE scheme in the selective security model[3] reduces to the hardness of the q-parallel BDHE assumption. Suppose that there exists a polynomial-time adversary $\mathcal{A}$ which can attack our scheme in the selective security model with advantage ϵ. Then, we can build a simulator $\mathcal{B}$ which distinguishes the q-parallel BDHE tuple $\{\vec{y}, T\}$ with advantage ϵ. The simulation proceeds as follows.

Init: First, $\mathcal{B}$ gets a challenge q-parallel BDHE tuple $\{\vec{y}, T\}$. Note that we only consider the simplest case of a compact access policy in this proof; the proof of other cases is similar. Thus, $\mathcal{A}$ can present a challenge policy set as $\mathcal{T} = \{\mathbb{T}_1, \mathbb{T}_2\}$, where $|L_{\mathbb{T}_1}| = l_1$, $|L_{\mathbb{T}_2}| = l_2$, and $l_1 + l_2 \leq q$. Let $\mathfrak{n}^*$ be the only public node of $\mathcal{T}$, which only associates with the i_1^{th} leaf of $\mathbb{T}_1$ and the i_2^{th} leaf of $\mathbb{T}_2$. Following the method proposed in Reference [33], $\mathbb{T}_1$ and $\mathbb{T}_2$ can be equivalently expressed as two LSSS matrices denoted by (X, ρ_1) and (Y, ρ_2). Assume that $X = (x_{i,j})_{l_1 \times m_1}$, $Y = (y_{i,j})_{l_2 \times m_2}$, X_{i_1}, Y_{i_2} denote the rows assigned to public node $\mathfrak{n}^*$, $\rho_1(i_1) = \rho_2(i_2) = \mathfrak{a}^*$, and $m_1 \leq l_1, m_2 \leq l_2$. Following the definition of X and Y, there exist $j_1 \in \mathbb{Z}_{m_1}, j_2 \in \mathbb{Z}_{m_2}$ where $\forall i \neq i_1, x_{i,j_1} \neq 0$ and $\forall i \neq i_2, y_{i,j_2} \neq 0$. For simplicity, let $j_1 = j_2 = 1$. Then, we define the following matrix M, and mapping function ρ^*:

$$M = (m_{i,j})_{(l_1+l_2)\times(m_1+m_2)} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,m_1} & \frac{y_{i_2,2}x_{1,1}}{x_{i_1,1}} & \cdots & \frac{y_{i_2,m_2}x_{1,1}}{x_{i_1,1}} \\ \vdots & & \vdots & \vdots & & \vdots \\ x_{l_1,1} & \cdots & x_{l_1,m_1} & \frac{y_{i_2,2}x_{l_1,1}}{x_{i_1,1}} & \cdots & \frac{y_{i_2,m_2}x_{l_1,1}}{x_{i_1,1}} \\ \frac{x_{i_1,1}y_{1,1}}{y_{i_2,1}} & \cdots & \frac{x_{i_1,m_1}y_{1,1}}{y_{i_2,1}} & y_{1,2} & \cdots & y_{1,m_2} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{x_{i_1,1}y_{l_2,1}}{y_{i_2,1}} & \cdots & \frac{x_{i_1,m_1}y_{l_2,1}}{y_{i_2,1}} & y_{l_2,2} & \cdots & y_{l_2,m_2} \end{pmatrix}$$

$$\rho^*(l) = \begin{cases} \rho_1(l), & 1 \leq l \leq l_1 \\ \rho_2(l - l_1), & l_1 < l \leq l_2. \end{cases}$$

Setup: $\mathcal{B}$ chooses $w', \eta \in Z_p$ and sets $x = \alpha\eta$, $w = w' + \alpha^{q+1}$, $g_1 = g_2 = g$, $P = (g^\alpha)^\eta \Upsilon = e(g, g)^w = e(g, g^{w'})e(g^\alpha, g^{\alpha^q})$. Then, it chooses $a_i = (a'_i + \sum_{\rho^*(l)=\mathfrak{a}_i} \sum_{j=1}^{m_1+m_2-1} \alpha^j m_{l,j}/b_l)/x$. This implies that:

$$P_i = g^{xa_i} = g^{a'_i} \prod_{\rho^*(l)=\mathfrak{a}_i} \prod_{j=1}^{m_1+m_2-1} (g^{\alpha^j/b_l})^{m_{l,j}}.$$

Significantly, $P_i = g^{a'_i}$ if and only if $S_i = \varnothing$.

Phase 1: $\mathcal{B}$ responds to the SK queries. Assume that $\mathcal{B}$ is given an SK query with a set U_s which does not satisfy policy (X, ρ_1). Then, $\mathcal{B}$ picks $u' \in Z_p$ at random. Furthermore, it finds a vector $\varpi = (\varpi_1, \ldots, \varpi_{m_1+m_2-1}) \in Z_p^{m_1}$ such that $\varpi_1 = 1$, and $\forall i$, $\rho^*(i) \in U_s$ gets $\varpi M_i = 0$, where M_i denotes the i^{th} row of M. By the definition of M, such a vector must exist. $\mathcal{B}$ sets $u = u' + (\sum_{j=1}^{m_1+m_2-1} \alpha^{q+1-j}\varpi_j)/\eta$. Furthermore, it sets:

$$\bar{D}_u = g^u = g^{u'} \prod_{j=1}^{m_1+m_2-1} g^{\alpha^{q+1-j}\varpi_j/\eta},$$

$$D'_u = g^{\omega+\alpha^{q+1}} p_u^{-x} = g^\omega \prod_{j=2}^{m_1+m_2-1} g^{-\alpha^{q+2-j}\varpi_j}.$$

Additionally, $\mathcal{B}$ calculates the attribute SK $D_i = p_u^{xa_i} = g^{uxa_i}$. We consider the calculation in two case:

1) For a given $\mathfrak{a}_i$, $\forall l$, $\rho^*(l) \neq \mathfrak{a}_i$. In this case, $a_i = a'_i/x$ and $D_i = p_u^{xa_i} = g^{a'_i u' + a'_i/\eta \sum_{j=1}^{m_1+m_2-1} \varpi_j \alpha^{q+1-j}}$.

2) For a given $\mathfrak{a}_i$, $\exists l$ such that $\rho^*(l) = \mathfrak{a}_i$. We find $a_i = (a'_i + \sum_{\rho^*(l)=\mathfrak{a}_i, l\neq i_1} \sum_{j=1}^{m_1+m_2-1} \alpha^j m_{l,j}/b_l)/x$. This implies:

$$\begin{aligned} D_i &= g^{uxa_i} \\ &= g^{u'a'_i} \prod_{j=1}^{m} (g^{\alpha^{q+1-j}})^{a'_i\varpi_j/\eta} \prod_{\rho^*(l)=\mathfrak{a}_i} \prod_{k=1}^{m_1+m_2-1} (g^{\alpha^k/b_l})^{u'm_{l,k}} \\ &\quad \cdot \prod_{\rho^*(l)=\mathfrak{a}_i} \prod_{j=1}^{m_1+m_2-1} \prod_{k=1}^{m_1+m_2-1} g^{\alpha^{q+1-j+k}\varpi_j m_{l,k}/(b_l\eta)}, \end{aligned}$$

where:

$$\begin{aligned} &\sum_{\rho^*(l)=\mathfrak{a}_i} \sum_{j=k} \alpha^{q+1-j+k}\varpi_j x_{l,k}/(b_l\eta) \\ &= \alpha^{q+1}/\eta \sum_{\rho^*(l)=\mathfrak{a}_i} \sum_{j=1}^{m_1+m_2-1} \varpi_j x_{l,j}/b_l \\ &= 0. \end{aligned}$$

Thus:

$$\begin{aligned} D_i = g^{u'a'_i} &\prod_{j=1}^{m_1+m_2-1} (g^{\alpha^{q+1-j}})^{a'_i\varpi_j/\eta} \\ &\times \prod_{\rho_1(l)=\mathfrak{a}_i} \prod_{k=1}^{m_1+m_2-1} (g^{\alpha^k/b_l})^{u'm_{l,k}} \\ &\cdot \prod_{\rho_1(l)=\mathfrak{a}_i} \prod_{j=1}^{m_1+m_2-1} \prod_{k\neq j} (g^{\alpha^{q+1-j+k}/b_l})^{\omega_j m_{l,k}/\eta}. \end{aligned}$$

Since the unknown term $g^{\alpha^{q+1}}$ is canceled, we can calculate D'_u and D_i easily.

[2]Calculation of the Lagrange coefficient is given in Equation (4).

[3]The selective security model is given in [19].

Challenge: $\mathcal{B}$ builds the challenge ciphertext. $\mathcal{A}$ gives two challenge messages M_0 and M_1 and an appended message M_2 to $\mathcal{B}$. It creates $C_1 = M_\upsilon T\Upsilon^{\gamma_1}, C_2 = M_2\Upsilon^{\gamma_2}$, and $C_1' = g^s g^{\gamma_1}, C_2' = g^{\gamma_2}$, where $\gamma_1, \gamma_2 \in Z_p$, and $\upsilon \in \{0, 1\}$ are all chosen at random. Then, $\mathcal{B}$ chooses $v_2, \ldots, v_{m_1+m_2} \in Z_p$ randomly and generates the vector:

$$V_1 = (s + \gamma_1 + \frac{y_{i_2,1}\gamma_2}{x_{i_1,1}} + s\sum_{j=1}^{m_2}\frac{y_{i_2,j}\alpha^{m_1+j-1}}{x_{i_1,1}}, + \sum_{j=2}^{m_2}\frac{y_{i_2,j}v_{m_1+j}}{x_{i_1,1}}s\alpha + v_2, \ldots, s\alpha^{m_1-1} + v_{m_1}).$$

For each row X_l of X, $\mathcal{B}$ chooses $r_l' \in Z_p$ randomly, sets $r_l = r_l' - sb_l\eta$, and calculates:

$$\begin{aligned}\lambda_l &= X_l V_1 \\ &= x_{l,1}\gamma_1 + \frac{y_{i_2,1}\gamma_2 x_{l,1}}{x_{i_1,1}} + s\sum_{j=1}^{m_1}\alpha^{j-1}x_{l,j} + \sum_{j=2}^{m_1} v_j x_{l,j} \\ &\quad + s\sum_{j=2}^{m_2}\alpha^{j-1}\frac{y_{i_2,j}x_{l,1}}{x_{i_1,1}} + \sum_{j=2}^{m_2} v_{m_1+j-1}\frac{y_{i_2,j}x_{l,1}}{x_{i_1,1}} \\ &= m_{l,1}\gamma_1 + \frac{y_{i_2,1}\gamma_2 x_{l,1}}{x_{i_1,1}} + s\sum_{j=1}^{m_1+m_2-1}\alpha^{j-1}m_{l,j} \\ &\quad + \sum_{j=2}^{m_1+m_2-1} v_j m_{l,j}.\end{aligned}$$

Assume that $\rho_1(l) = \mathfrak{a}_i$. We find:

$$\begin{aligned}\bar{C}_{1,l} &= g^{r_l} = g^{r_l'}(g^{sb_l})^{-\eta}, \\ \hat{C}_{1,l} &= g^{x\lambda_l} g^{r_l x a_i} \\ &= g^{t_l' a_i'}(g^{-sb_l})^{\eta a_i'}(g^\alpha)^{\eta(m_{l_1+l,1}\gamma_1 + y_{i_2,1}\gamma_2 x_{l,1}/x_{i_1,1})} \\ &\quad \times \prod_{j=1}^{m_1+m_2-1} g^{s\alpha^j \eta m_{l,j}} \prod_{j=2}^{m_1+m_2-1}(g^\alpha)^{\eta v_j m_{l,j}} \\ &\quad \times \prod_{\rho_1(k)=\mathfrak{a}_i}\prod_{j=1}^{m_1+m_2-1}(g^{\alpha^j/b_k})^{r_l' m_{k,j}} \\ &\quad \times \prod_{\rho_1(k)=\mathfrak{a}_i}\prod_{j=1}^{m_1+m_2-1}(g^{s\alpha^j b_l/b_k})^{\eta m_{k,j}} \\ &= g^{r_l' a_i'}(g^{-sb_l})^{\eta a_i'}(g^\alpha)^{\eta(m_{l,1}\gamma_1 + y_{i_2,1}\gamma_2 x_{l,1}/x_{i_1,1})} \\ &\quad \times \prod_{j=2}^{m_1+m_2-1}(g^\alpha)^{\eta v_j m_{l,j}} \prod_{\rho_1(k)=\mathfrak{a}_i}\prod_{j=1}^{m_1+m_2-1}(g^{\alpha^j/b_k})^{r_l' m_{k,j}} \\ &\quad \times \prod_{\rho_1(k)=\mathfrak{a}_i, k\neq l}\prod_{j=1}^{m_1+m_2-1}(g^{s\alpha^j b_l/b_k})^{\eta m_{k,j}}.\end{aligned}$$

For message M_2, $\mathcal{B}$ generates the following vector:

$$V_2 = (\gamma_2 + \frac{\gamma_1 x_{i_1,1}}{y_{i_2,1}} + s\sum_{j=2}^{m_1}\frac{x_{i_1,j}\alpha^{j_1}}{y_{i_2,1}}, s\alpha^{m_1} + v_{m_1+1}, \ldots, s\alpha^{m_1+m_2-2} + v_{m_1+m_2-1}).$$

Similarly, for each row Y_l of Y, $\mathcal{B}$ calculates:

$$\begin{aligned}\Lambda_l &= Y_l V_2 \\ &= m_{l,1}\gamma_1 + y_{l,1}\gamma_2 \\ &\quad + s\sum_{j=1}^{m_1+m_2-1}\alpha^{j-1}m_{l,j} + \sum_{j=2}^{m_1+m_2-1} v_j m_{l,j},\end{aligned}$$

and $\Lambda_{i_2} = \lambda_{i_1}$. Furthermore, $\mathcal{B}$ chooses $t_l' \in Z_p, l \neq i_2$ at random and sets:

$$t_l = \begin{cases} t_l' - sb_{l_1+l}\eta, & l \neq i_2 \\ r_{i_1}' - sb_{i_1}\eta, & l = i_2 \end{cases}. \tag{24}$$

Thus, it can easily calculate $\bar{C}_{2,l} = g^{t_l}$. Furthermore, for $\forall l \neq i_2$, assume that $\rho_2(l) = \mathfrak{a}_i$. Thus:

$$\begin{aligned}\hat{C}_{2,l} &= g^{x\Lambda_l} g^{t_l x a_i} \\ &= g^{t_l' a_i'}(g^{-sb_{l_1+l}})^{\eta a_i'}(g^\alpha)^{\eta(m_{l_1+l,1}\gamma_1 + y_{l,1}\gamma_2)} \\ &\quad \times \prod_{j=1}^{m_1+m_2-1} g^{s\alpha^j \eta m_{l_1+l,j}} \prod_{j=2}^{m_1+m_2-1}(g^\alpha)^{\eta v_j m_{l_1+l,j}} \\ &\quad \times \prod_{\rho^*(k)=\mathfrak{a}_i}\prod_{j=1}^{m_1+m_2-1}(g^{\alpha^j/b_k})^{t_l' m_{k,j}} \\ &\quad \times \prod_{\rho^*(k)=\mathfrak{a}_i}\prod_{j=1}^{m_1+m_2-1}(g^{s\alpha^j b_{l_1+l}/b_k})^{\eta m_{k,j}} \\ &= g^{t_l' a_i'}(g^{-sb_{m_1+l}})^{\eta a_i'}(g^\alpha)^{\eta(m_{l_1+l,1}\gamma_1 + y_{l,1}\gamma_2)} \\ &\quad \times \prod_{j=2}^{m_1+m_2-1}(g^\alpha)^{\eta v_j m_{l_1+l,j}} \\ &\quad \times \prod_{\rho^*(k)=\mathfrak{a}_i}\prod_{j=1}^{m_1+m_2-1}(g^{\alpha^j/b_k})^{t_l' m_{k,j}} \\ &\quad \times \prod_{\rho(k)=\mathfrak{a}_i, k\neq l_1+l}\prod_{j=1}^{m_1+m_2-1}(g^{s\alpha^j b_{l_1+l}/b_k})^{\eta m_{k,j}}\end{aligned}$$

Significantly, for $l = i_2$, $\mathcal{B}$ gets $\bar{C}_{2,i_2} = \bar{C}_{1,i_1}$ and $\hat{C}_{2,i_2} = \hat{C}_{1,i_1}$. Finally, $\mathcal{B}$ sets $C'_{\mathfrak{a}^*} = g_1^{s'_{\mathfrak{a}^*}}, \bar{C}'_{\mathfrak{a}^*} = g_1^{r'_{\mathfrak{a}^*}}, \hat{C}'_{\mathfrak{a}^*} = P_{\mathfrak{a}^*}^{r'_{\mathfrak{a}^*}} P^{s'_{\mathfrak{a}^*}}$, where $s'_{\mathfrak{a}^*}, r'_{\mathfrak{a}^*} \in Z_p$ are chosen at random. The public attribute ciphertext of public node $\mathfrak{n}$ is shown as follows:

$$C^*_{\rho_1(i_1)} = < \bar{C}^*_{\mathfrak{a}^*} = \bar{C}_{1,i_1}, \quad \hat{C}^*_{\mathfrak{a}^*} = \hat{C}_{1,i_1}, C'_{\mathfrak{a}^*}, \bar{C}'_{\mathfrak{a}^*}, \hat{C}'_{\mathfrak{a}^*} >$$

The above components are easy to calculate because there is no unknown term $g^{\alpha^{q+1}}$. Additionally, X_{i_1}, Y_{i_2} are encrypted as $\bar{X}_{i_1} = \Upsilon^{s'_{\mathfrak{a}^*}} X_{i_1}, \bar{Y}_{i_2} = \Upsilon^{s'_{\mathfrak{a}^*}} Y_{i_2}$.

Phase 2: Phase 1 is repeated.

Guess: $\mathcal{A}$ outputs a guess υ' of υ. $\mathcal{B}$ outputs 0 to guess $T = e(g, g)^{s\alpha^{q+1}}$ when $\upsilon' = \upsilon$; otherwise, it outputs 1 to indicate that T is a random element in G_T. Assume that $\mathcal{A}$ has the non-negligible advantage $Adv_{\mathcal{A}} = \epsilon$ in breaking the CCP-ABE scheme. It is clear that $\mathcal{B}$ gives a perfect simulation when $T = e(g, g)^{s\alpha^{q+1}}$. Thus,

$$Pr[\mathcal{B}(\vec{y}, T = e(g, g)^{s\alpha^{q+1}}) = 0] = \frac{1}{2} + Adv_{\mathcal{A}}.$$

Otherwise, if T is a random element of G_T, M_b is completely hidden from $\mathcal{A}$. We find:

$$Pr[\mathcal{B}(\vec{y}, T = R) = 0] = \frac{1}{2}.$$

Therefore, $\mathcal{B}$ can play the decisional q-BDHE game with non-negligible advantage ϵ:

$$\begin{aligned} Adv_{\mathcal{B}} &= |Pr[\mathcal{B}(\vec{y}, T = e(g,g)^{s\alpha^{q+1}}) = 0] \\ &\quad - Pr[\mathcal{B}(\vec{y}, T = R) = 0]| \\ &= Adv_{\mathcal{A}} \\ &= \epsilon. \end{aligned}$$

As a result, $\mathcal{B}$ has the non-ignorable advantage ϵ in distinguishing the q-parallel BDHE tuple $\{\vec{y}, T\}$ when the PPT attacker $\mathcal{A}$ has the advantage ϵ in breaking our CCP-ABE scheme.

B. PERFORMANCE EVALUATION

Scalability and flexibility are two important properties for the IC-IoT data access control mechanism. Different from the traditional access control mechanism, the ABE access control mechanism does not require that a central organization be responsible for managing data access privilege. This implies that ABE has the significant improvement of having access control mechanism scalability. Furthermore, compared with other cryptography schemes, ABE provides a novel one-to-many encrypting pattern, which provides improved flexibility of the IC-IoT data access control mechanism. In brief, the ABE ciphertext access control mechanism is more suitable for the highly pervasive and distributed IC-IoT.

However, the existing ABE scheme has greater ciphertext redundancy, which incurs heavy computation, communication, and storage costs for IC-IoT devices. Thus, low-cost ABE schemes have been extensively researched in recent years. Significantly, in this respect, the proposed CCP-ABE scheme shows comprehensive performance improvement. For performance evaluation, we simulated our CCP-ABE scheme using a Linux virtual machine with 2.83 GHz CPU and 1.00 GB RAM. The result of the simulation is given as follows.

Assume that a set of data $\{M_1, M_2, \ldots, M_m\}$ is given with the access tree set $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m\}$. Let l_i denote the number of leaves of $\mathcal{T}_i$, and let δ_i denote the overlap factor of $\mathcal{T}_i$, where $1 \leq i \leq m$. Furthermore, assume that $l_i \sim N(\mu_l, \delta_l)$ and $\delta_i \sim N(\mu_{\mathbb{T}}, \delta_{\mathbb{T}})$ are independent and identically distributed. Figure 7 shows the compacted ciphertext size as a function of the number m for $\mu_l = 20$, $\delta_l = \delta_{\mathbb{T}} = 5$, and $\mu_{\mathbb{T}} = 0, 5, 10, 15$. Note that, the curve $\mu_{\mathbb{T}} = 0$ approximates to the uncompacted case, it can be viewed as the reference curve. However, in this case, the ciphertext size increases with the total number of access policies m, and its increase rate depends on the expected overlap factor $\mu_{\mathbb{T}}$ of each data access policy.

Similarly, assume a ciphertext compacting ratio $R = 1 - L'/L$, where L' is the number of leaves in the compacted access tree sets and L is the number of leaves in the uncompacted access tree sets. Figure 8 shows the compacting ratio of such a ciphertext set as a function of the number m when $\mu_l = 20$, $\delta_l = \delta_{\mathbb{T}} = 5$, and $\mu_{\mathbb{T}} = 0, 5, 10, 15$. Note that, the curve $\mu_{\mathbb{T}} = 0$ approximates to the case of the uncompacted scheme, it can be viewed as the reference curve. In this case, the compacting ratio R increases quickly with m when m is small and approaches the constant $\mu_{\mathbb{T}}/\mu_l$ when m is larger than a certain threshold.

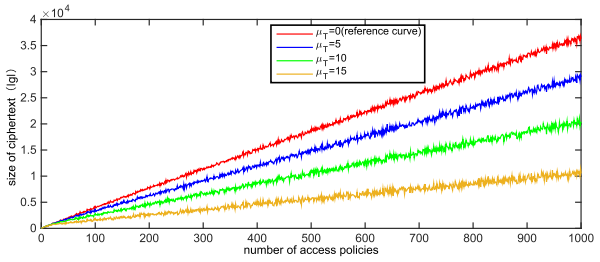

FIGURE 7. Size of the attribute ciphertext set. (Leaf number variance of each access tree $\delta_l = 5$; Leaf number expectation of each access tree $\mu_l = 20$; overlap factor variance of each access tree $\delta_{\mathbb{T}} = 5$; overlap factor expectation of each access tree $\mu_{\mathbb{T}} = 0, 5, 10, 15$).

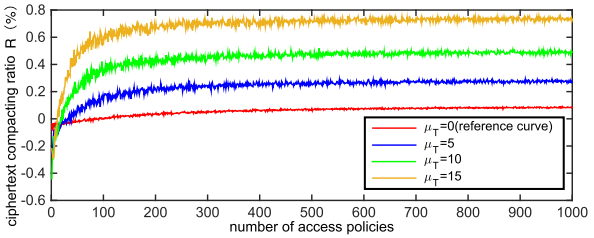

FIGURE 8. Compacting ratio of ciphertexts. (Leaf number variance of each access tree $\delta_l = 5$; Leaf number expectation of each access tree $\mu_l = 20$; overlap factor variance of each access tree $\delta_{\mathbb{T}} = 5$; overlap factor expectation of each access tree $\mu_{\mathbb{T}} = 0, 5, 10, 15$).

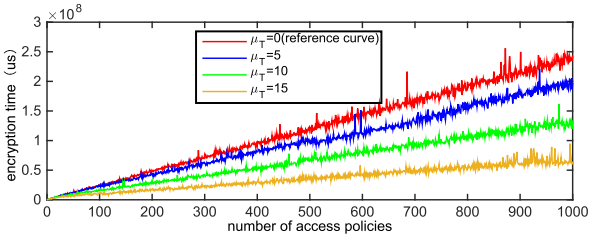

FIGURE 9. Encryption time. (Leaf number variance of each access tree $\delta_l = 5$; Leaf number expectation of each access tree $\mu_l = 20$; overlap factor variance of each access tree $\delta_{\mathbb{T}} = 5$; overlap factor expectation of each access tree $\mu_{\mathbb{T}} = 0, 5, 10, 15$).

Furthermore, Figure 9 shows the encryption time of such compacted ciphertexts as a function of the number m when $\mu_l = 20$, $\delta_l = \delta_{\mathbb{T}} = 5$, and $\mu_{\mathbb{T}} = 0, 5, 10, 15$. Note that, the curve $\mu_{\mathbb{T}} = 0$ approximates to the case of uncompacted ABE scheme, it can be viewed as the reference curve. In this case, the encryption time linearly grows with the number of access policies, but the larger $\mu_{\mathbb{T}}$ incurs a lower growth rate.

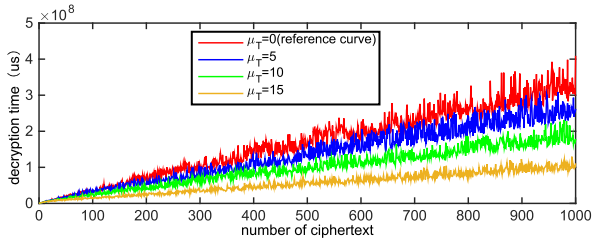

FIGURE 10. Decryption time. (Leaf number variance of each decrypting subtree $\delta_l = 5$; Leaf number expectation of each decrypting subtree $\mu_l = 20$; overlap factor variance of each decrypting subtree $\delta_{\mathbb{T}} = 5$; overlap factor expectation of each decrypting subtree $\mu_{\mathbb{T}} = 0, 5, 10, 15$).

Finally, let $\mathcal{T}_i, 1 \le i \le m$ denote a decrypting subtree of M_i, and let l_i denote the number of leaves of $\mathcal{T}_i$, where $l_i \sim N(\mu_l, \delta_l)$ and $\delta_i \sim N(\mu_{\mathbb{T}}, \delta_{\mathbb{T}})$ are independently identically distributed. Figure 10 shows the corresponding decryption time of compacted ciphertexts as a function of the ciphertext number m when $\mu_l = 20$, $\delta_l = \delta_{\mathbb{T}} = 5$, and $\mu_{\mathbb{T}} = 0, 5, 10, 15$. Note that, the curve $\mu_{\mathbb{T}} = 0$ approximates to the case of uncompacted ABE scheme, it can be viewed as the reference curve. Similarly, in this case, the decryption time has approximately linear growth with the number of ciphertexts, but a larger μ_l incurs a lower growth rate.

In brief, as shown in the simulation, the comprehensive performance of the proposed CCP-ABE scheme has been greatly improved.

VI. CONCLUSION

In order to reduce ciphertext redundancy, we provide a policy-compacting method for ABE. The method can reduce various overheads of the ABE scheme without sacrificing any additional performance. However, the policy-compacting problem is an NPC problem, and a greedy compacting algorithm is provided to achieve an approximate minimum ciphertext scale. The detailed security proof and performance evaluation of the scheme are also given in this work. These demonstrate that the scheme obtains comprehensive performance improvement, with its storage and computation overhead all significantly reduced.

APPENDIX A
FLEXIBILITY FACTOR

Let $\mathfrak{n}$ be a non-leaf ($t_{\mathfrak{n}}$-out-of-$n_{\mathfrak{n}}$ node) of access tree $\mathbb{T}$, $\mathfrak{p}$ be the parent of $\mathfrak{n}$, $\gamma_{\mathfrak{n}}$ be the flexibility factor of $\mathfrak{n}$, $\bar{\gamma}_{\mathfrak{n}}$ denote the upper limit of $\gamma_{\mathfrak{n}}$, and p be a prime. Here we prove that the inequality in Equation (13) holds. First, $\bar{\gamma}_{\mathfrak{n}}$ is discussed in two cases:

1) For $t_{\mathfrak{n}} \neq 2$. This case is presented in two further subcases: $t_{\mathfrak{n}} = 1$ and $t_{\mathfrak{n}} \ge 3$. If $t_{\mathfrak{n}} = 1$, $f_{\mathfrak{n}}(x)$ must be a constant function, and it can be affirmed with only one interpolating node $(x_{\mathfrak{c}}, s_{\mathfrak{c}}), \mathfrak{c} \in N_{\mathfrak{n}}$. Thus, $\bar{\gamma}_{\mathfrak{n}} = t_{\mathfrak{n}} = 1$. If $t_{\mathfrak{n}} \ge 3$, $f_{\mathfrak{n}}(x)$ is a $(t_{\mathfrak{n}} - 1)$-order polynomial function. It is clear that it is highly possible that there is a non-feasible solution $x \in Z_p$ for $f(x) \equiv s \mod p$, $s \in Z_p$ when $t_{\mathfrak{n}} \ge 3$. For feasibility and efficiency, $f(x)$ is constructed as a Lagrange interpolation polynomial:

$$f(x) = \sum_{\mathfrak{c}_i \in N'} \Delta_{x_{\mathfrak{c}_i}, X_{\bar{N}}}(x) s_i,$$

where $s_i \in Z_p$ is chosen at random, $\bar{N} \subset N_{\mathfrak{n}}$ and $|\bar{N}| = t$, $\bar{X} = \{x_{\mathfrak{c}_i} | \mathfrak{c}_i \in \bar{N}\}$, and $\Delta_{x_i, \bar{X}}$ is the Lagrange coefficient of $\mathfrak{c}_i$. Note that the shares of $\mathfrak{c}_i \in \bar{N}$ are not all equal, otherwise $f(x)$ is degenerate and hence a constant function. For the remaining nodes $\mathfrak{c}_i \in N_{\mathfrak{n}} - \bar{N}$, the node share s_i is set to $f(x_{\mathfrak{c}_i})$, where $x_{\mathfrak{c}_i}$ is chosen at random. Thus, $\bar{\gamma}_{\mathfrak{n}}$ is also equal to $t_{\mathfrak{n}}$ in this subcase.

2) For $t_{\mathfrak{n}} = 2$. Let $s_{\mathfrak{c}_i} \in Z_p$ be the share of $\mathfrak{c}_i \in N$, and the node polynomial can be assigned as $f_{\mathfrak{n}}(x) = ax + s_{\mathfrak{n}}$, where $a \in Z_p$ and $s_{\mathfrak{n}}$ is the share of $\mathfrak{n}$. However, all shares of $s_{\mathfrak{c}_i}, 0 \le i \le n_{\mathfrak{n}}$ do not equally guarantee that $f_{\mathfrak{n}}(x)$ is not degenerate. $\forall \mathfrak{c}_i, 1 \le i \le n_{\mathfrak{n}}$, the valid interpolation in Z_p can be calculated as $x_i = a^{-1}(s_{\mathfrak{c}_i} - s_{\mathfrak{n}}) \mod p$. Thus, $\bar{\gamma}_{\mathfrak{n}}$ is generally set to be $|N_{\mathfrak{n}}|$ when $t_{\mathfrak{n}} = 2$.

Next, the inequality $\sum_{\mathfrak{n} \in \mathbb{T}'} \gamma_{\mathfrak{n}} \le |\bigcup_{\mathfrak{n} \in \mathbb{T}'} N_{\mathfrak{n}}|$ is considered. Let $\mathfrak{n}$ be a non-leaf of $\mathbb{T}$, $h_{\mathfrak{n}}$ be the height of $\mathfrak{n}$, and $N'_{\mathfrak{n}} = \{\mathfrak{c} \in N_{\mathfrak{n}} | \bar{\gamma}_{\mathfrak{c}} \ge |N'_{\mathfrak{c}}|\}$. The equation $\gamma_{\mathfrak{n}} = \min\{\bar{\gamma}_{\mathfrak{n}}, |N'_{\mathfrak{n}}|\}$ can be inductively proved. If $h_{\mathfrak{n}} = 2$, then $N_{\mathfrak{n}} = N'_{\mathfrak{n}}$. Thus, $\gamma_{\mathfrak{n}} = \min\{\bar{\gamma}_{\mathfrak{n}}, |N'_{\mathfrak{n}}|\}$ holds.

Assume that $\gamma_{\mathfrak{n}} = \min\{\bar{\gamma}_{\mathfrak{n}}, |N'_{\mathfrak{n}}|\}$ when $h_{\mathfrak{n}} \le k$. Then, we prove that $\gamma_{\mathfrak{n}} = \min\{\bar{\gamma}_{\mathfrak{n}}, |N'_{\mathfrak{n}}|\}$ when $h_{\mathfrak{n}} = k+1$. $\forall \mathbb{T}' \subset \mathbb{T}$, $\sum_{\mathfrak{n}' \in \mathbb{T}'} \gamma_{\mathfrak{n}'} \le |\cup_{\mathfrak{n}' \in \mathbb{T}'_{\mathfrak{n}}} N_{\mathfrak{n}'}|$. This implies that:

$$\gamma_{\mathfrak{n}} \le |\cup_{\mathfrak{n}' \in \mathbb{T}'_{\mathfrak{n}}} N_{\mathfrak{n}'}| - \sum_{\mathfrak{n}' \in \mathbb{T}'_{\mathfrak{n}} - \{\mathfrak{n}\}} \gamma_{\mathfrak{n}'},$$

where $\mathbb{T}'_{\mathfrak{n}}$ denotes that a subtree of $\mathbb{T}$ includes $\mathfrak{n}$. Thus, we select a subtree $\mathbb{T}_{\mathfrak{n}} \subset \mathbb{T}$ that satisfies the following conditions:

1) $\mathfrak{n}$ is the root of $\mathbb{T}_{\mathfrak{n}}$; and

2) $\forall \mathfrak{c} \in N_{\mathfrak{n}'}, \mathfrak{n}' \in \mathbb{T}_{\mathfrak{n}}$, if $\bar{\gamma}_{\mathfrak{c}} \ge |N'_{\mathfrak{c}}|$, then $\mathfrak{c} \in \mathbb{T}_{\mathfrak{n}}$.

$\forall \mathfrak{n}' \in \mathbb{T}_{\mathfrak{n}} - \{\mathfrak{n}\}$, we have $h_{\mathfrak{n}'} \le k$ and $\gamma_{\mathfrak{n}'} = \min\{\bar{\gamma}_{\mathfrak{n}'}, |N'_{\mathfrak{n}'}|\} \ge |N'_{\mathfrak{n}'}|$. As a result:

$$\begin{aligned}
\gamma_{\mathfrak{n}} &= \min_{\mathbb{T}'_{\mathfrak{n}}}\{|\cup_{\mathfrak{n}' \in \mathbb{T}'_{\mathfrak{n}}} N_{\mathfrak{n}'}| - \sum_{\mathfrak{n}' \in \mathbb{T}'_{\mathfrak{n}} - \{\mathfrak{n}\}} \gamma_{\mathfrak{n}'}\} \\
&\le |\cup_{\mathfrak{n}' \in \mathbb{T}_{\mathfrak{n}}} N_{\mathfrak{n}'}| - \sum_{\mathfrak{n}' \in \mathbb{T}_{\mathfrak{n}} - \{\mathfrak{n}\}} \gamma_{\mathfrak{n}'} \\
&= |\cup_{\mathfrak{n}' \in \mathbb{T}_{\mathfrak{n}}} N_{\mathfrak{n}'}| - \sum_{\mathfrak{n}' \in \mathbb{T}_{\mathfrak{n}} - \{\mathfrak{n}\}} |N'_{\mathfrak{n}'}| \\
&= |N'_{\mathfrak{n}}|.
\end{aligned}$$

Thus, $\gamma_{\mathfrak{n}} = \min\{\bar{\gamma}_{\mathfrak{n}}, |N'_{\mathfrak{n}}|\}$ when $h_{\mathfrak{n}} = k + 1$.

APPENDIX B
OVERLAP FACTOR

Let $\mathbb{T}$ be an access tree, $N_{\mathbb{T}}$ be the non-leaf set of $\mathbb{T}$, $\mathfrak{r}$ be the root of $\mathbb{T}$, and $d_{\mathbb{T}}$ describe the depth of $\mathbb{T}$. Here, we prove that Equation (14) holds. Considering security, the root $\mathfrak{r}$ of $\mathbb{T}$ must be assigned with a random secret. Thus, the overlap factor $\delta_{\mathbb{T}}$ is analyzed as follows.

If $k = 2$, then $\mathfrak{r}$ is the only non-leaf of $\mathbb{T}$. We describe $\delta_{\mathbb{T}}$ in two cases:

1) $s_{\mathfrak{r}}$ is assigned. Thus, $\delta_{\mathbb{T}} = \gamma_{\mathfrak{n}} - 1$. As a result:

$$\delta_{\mathbb{T}} = \sum_{\mathfrak{n}' \in N_{\mathbb{T}}} (\gamma_{\mathfrak{n}'} - 1). \tag{25}$$

2) $s_{\mathfrak{r}}$ is unassigned. Then $\delta_{\mathbb{T}}$ is calculated as follows:

$$\delta_{\mathbb{T}} = \begin{cases} \gamma_{\mathfrak{n}}, & \gamma_{\mathfrak{n}} < N'_{\mathfrak{n}} \\ \gamma_{\mathfrak{n}} - 1, & \gamma_{\mathfrak{n}} = N'_{\mathfrak{n}} \end{cases}. \tag{26}$$

This implies that:

$$\delta_{\mathbb{T}} = \begin{cases} 1 + \sum_{\mathfrak{n}' \in N_{\mathbb{T}}} (\gamma_{\mathfrak{n}'} - 1), & \gamma_{\mathfrak{n}} < N'_{\mathfrak{n}} \\ \sum_{\mathfrak{n}' \in N_{\mathbb{T}}} (\gamma_{\mathfrak{n}'} - 1), & \gamma_{\mathfrak{n}} = N'_{\mathfrak{n}} \end{cases}. \tag{27}$$

Assume Equation (14) always holds when $d_{\mathbb{T}} \leq k$. Then, we prove that the equation hold when $d_{\mathbb{T}} = k + 1$. Let $\mathbb{T}_{\mathfrak{c}_i}$ be the i^{th} subtree of $\mathfrak{r}$, where $\mathfrak{c}_i \in N_{\mathfrak{r}} - \{\mathfrak{r}\}$ and $d_{\mathbb{T}_{\mathfrak{c}_i}} \leq k$. $\delta_{\mathbb{T}}$ is also discussed in two cases:

1) $s_{\mathfrak{r}}$ is assigned. For an arbitrary subset $\bar{N} \subset N'_{\mathfrak{r}} - \{\mathfrak{r}\}$, $|\bar{N}| = \gamma_{\mathfrak{r}} - 1$, $\forall \mathfrak{c}_i \in \bar{N}$ can be assigned with random node share (i.e., $s_{\mathfrak{c}_i}$ is unassigned). At the same time, $\forall \mathfrak{c}_i \in N_{\mathfrak{r}} - \bar{N} - \{\mathfrak{r}\}$, $s_{\mathfrak{c}_i}$ must be assigned. Thus:

$$\delta_{\mathbb{T}_{\mathfrak{c}_i}} = \begin{cases} 1 + \sum_{\mathfrak{n}' \in N_{\mathbb{T}_{\mathfrak{c}_i}}} (\gamma_{\mathfrak{n}'} - 1), & \mathfrak{c}_i \in \bar{N} \\ \sum_{\mathfrak{n}' \in N_{\mathbb{T}_{\mathfrak{c}_i}}} (\gamma_{\mathfrak{n}'} - 1), & \mathfrak{c}_i \in N'_{\mathfrak{c}_i} - \bar{N} - \{\mathfrak{r}\} \\ \sum_{\mathfrak{n}' \in N_{\mathbb{T}_{\mathfrak{c}_i}}} (\gamma_{\mathfrak{n}'} - 1), & \mathfrak{c}_i \in N_{\mathfrak{c}_i} - N'_{\mathfrak{c}_i} \end{cases}.$$

As a result, we calculate:

$$\begin{aligned} \delta_{\mathbb{T}} &= \sum_{\mathfrak{c}_i \in \bar{N}} \delta_{\mathbb{T}_{\mathfrak{c}_i}} + \sum_{\mathfrak{c}_i \in N'_{\mathfrak{c}_i} - \bar{N} - \{\mathfrak{r}\}} \delta_{\mathbb{T}_{\mathfrak{c}_i}} + \sum_{\mathfrak{c}_i \in N_{\mathfrak{c}_i} - N'_{\mathfrak{c}_i}} \delta_{\mathbb{T}_{\mathfrak{c}_i}} \\ &= \bar{N} + \sum_{\mathfrak{c}_i \in N'_{\mathfrak{r}}} \delta_{\mathbb{T}_{\mathfrak{c}_i}} \\ &= \sum_{\mathfrak{n}' \in N_{\mathbb{T}}} (\gamma_{\mathfrak{n}'} - 1). \end{aligned}$$

2) $s_{\mathfrak{r}}$ is unassigned. For an arbitrary subset $\bar{N} \subset N'_{\mathfrak{r}} - \{\mathfrak{r}\}$, $|\bar{N}| = \min\{\gamma_{\mathfrak{r}}, |N'_{\mathfrak{r}} - \{\mathfrak{r}\}|\}$, $\forall s_{\mathfrak{c}_i}, \mathfrak{c}_i \in \bar{N}$ is unassigned and $\forall \mathfrak{c}_i \in N_{\mathfrak{r}} - \bar{N} - \{\mathfrak{r}\}$, $s_{\mathfrak{c}_i}$ is assigned. Then, $\delta_{\mathfrak{c}_i}$ is also calculated as the equation shows. Thus, we find:

$$\begin{aligned} \delta_{\mathbb{T}} &= \bar{N} + \sum_{\mathfrak{c}_i \in N'_{\mathfrak{r}}} \delta_{\mathbb{T}_{\mathfrak{c}_i}} \\ &= \min\{\gamma_{\mathfrak{r}}, |N'_{\mathfrak{r}} - \{\mathfrak{r}\}|\} + \sum_{\mathfrak{n}' \in N_{\mathbb{T}} - \{\mathfrak{r}\}} (\gamma_{\mathfrak{n}'} - 1), \end{aligned}$$

and:

$$\delta_{\mathbb{T}} = \begin{cases} 1 + \sum_{\mathfrak{n}' \in N_{\mathbb{T}}} (\gamma_{\mathfrak{n}'} - 1), & \gamma_{\mathfrak{r}} < N'_{\mathfrak{n}} \\ \sum_{\mathfrak{n}' \in N_{\mathbb{T}}} (\gamma_{\mathfrak{n}'} - 1), & \gamma_{\mathfrak{r}} = N'_{\mathfrak{n}} \end{cases}$$

REFERENCES

[1] A. Shahzad *et al.*, "The protocol design and New approach for SCADA security enhancement during sensors broadcasting system," *Multimedia Tools Appl.*, vol. 75, no. 22, pp. 14641–14668, 2016.

[2] A. Shahzad, R. Landry, M. Lee, N. Xiong, J. Lee, and C. Lee, "A new cellular architecture for information retrieval from sensor networks through embedded service and security protocols," Sensors, vol. 16, no. 6, p. 821, 2016.

[3] A. Shahzad, M. Lee, H. D. Kim, S.-M. Woo, and N. Xiong, "New security development and trends to secure the SCADA sensors automated transmission during critical sessions," *Symmetry*, vol. 7, no. 4, pp. 1945–1980, 2015.

[4] Z. Wu, N. Xiong, Y. Huang, and Q. Gu, "Optimal service distribution in WSN service system subject to data security constraints," *Sensors*, vol. 14, no. 8, pp. 14180–14209, 2014.

[5] N. Xiong, Z. Wu, Y. Huang, and D. Xu, "Analyzing comprehensive QoS with security constraints for services composition applications in wireless sensor networks," *Sensors*, vol. 14, no. 12, pp. 22706–22736, 2014.

[6] B. Anggorojati, P. N. Mahalle, N. R. Prasad, and R. Prasad, "Capability-based access control delegation model on the federated IoT network," in *Proc. IEEE Int. Symp. Wireless Pers. Multimedia Commun.*, Sep. 2012, pp. 604–608.

[7] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the Internet of Things," *Math. Comput. Model.*, vol. 58, nos. 5–6, pp. 1189–1205, 2013.

[8] J. Liu, Y. Xiao, and C. L. P. Chen, "Authentication and access control in the Internet of Things," in *Proc. Int. Conf. Distrib. Comput. Syst. Workshops*, Jun. 2012, pp. 588–592.

[9] A. Shahzad *et al.*, "Real time MODBUS transmissions and cryptography security designs and enhancements of protocol sensitive information," *Symmetry*, vol. 7, no. 3, pp. 1176–1210, 2015.

[10] Q. Zhang, C. Zhou, N. Xiong, Y. Qin, X. Li, and S. Huang, "Multimodel-based incident prediction and risk assessment in dynamic cybersecurity protection for industrial control systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 10, pp. 1429–1444, Oct. 2017.

[11] Q. Zhang, C. Zhou, Y.-C. Tian, N. Xiong, Y. Qin, and B. Hu, "A fuzzy probability Bayesian network approach for dynamic cybersecurity risk assessment in industrial control systems," *IEEE Trans. Ind. Inform.*, vol. 14, no. 6, pp. 2497–2506, Jun. 2018.

[12] B. Lin, W. Guo, N. Xiong, G. Chen, A. V. Vasilakos, and H. Zhang, "A pretreatment workflow scheduling approach for big data applications in multicloud environments," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 581–594, Sep. 2016.

[13] H. Zheng, W. Guo, and N. Xiong, "A kernel-based compressive sensing approach for mobile data gathering in wireless sensor network systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, to be published. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8013710&isnumber=6376248

[14] L. Zhou, A. V. Vasilakos, N. Xiong, Y. Zhang, and S. Lian, "Scheduling security-critical multimedia applications in heterogeneous networks," *Comput. Commun.*, vol. 34, no. 3, pp. 429–435, 2011.

[15] N. Xiong, J. H. Park, L. T. Yang, B.-S. Koh, and Y. Li, "A security management scheme for failure detector distributed systems based on self-tuning control theory," *J. Intell. Manuf.*, vol. 22, no. 2, pp. 333–342, 2011.

[16] N. Xiong, F. Yang, H.-Y. Li, J. H. Park, Y.-S. Dai, and Y. Pan, "Security analysis and improvements of IEEE standard 802.16 in next generation wireless metropolitan access network," *Wireless Commun. Mobile Comput.*, vol. 11, no. 2, pp. 163–175, 2011.

[17] J. Herranz, F. Laguillaumie, and C. Ràfols, "Constant size ciphertexts in threshold attribute-based encryption," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2010. 19-34.

[18] C. Chen *et al.*, "Fully secure attribute-based systems with short ciphertexts/signatures and threshold access structures," in *Proc. Cryptographers' Track RSA Conf.* Berlin, Germany: Springer, 2013, pp. 50–67.

[19] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography—PKC* (Lecture Notes in Computer Science), vol. 6571, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds. Heidelberg, Germany: Springer, 2011, pp. 53–70.

[20] J. Han, W. Susilo, Y. Mu, and J. Yan, "Attribute-based oblivious access control," *Comput. J.*, vol. 55, no. 10, pp. 1202–1215, 2012.

[21] Y. Zhang, D. Zheng, X. Chen, J. Li, and H. Li, "Efficient attribute-based data sharing in mobile clouds," *Pervasive Mobile Comput.*, vol. 28, pp. 135–149, Jun. 2015.

[22] Y. Mao, J. Li, M.-R. Chen, J. Liu, C. Xie, and Y. Zhan, "Fully secure fuzzy identity-based encryption for secure IoT communications," *Comput. Standards Interfaces*, vol. 44, pp. 117–121, Feb. 2015.
[23] Y. Zhang, D. Zheng, X. Chen, J. Li, and H. Li, "Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts," in *Proc. Int. Conf. Provable Secur.*, 2014, pp. 259–273.
[24] A. Ge *et al.*, "Threshold ciphertext policy attribute-based encryption with constant size ciphertexts," in *Proc. Australas. Conf. Inf. Secur. Privacy*. New York, NY, USA: Springer-Verlag, 2012, pp. 336–349.
[25] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2014, pp. 293–310.
[26] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 8, pp. 1343–1354, Aug. 2013.
[27] K. Yang, X. Jia, K. Ren, B. Zhang, and R. Xie, "DAC-MACS: Effective data access control for multiauthority cloud storage systems," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1790–1801, Nov. 2013.
[28] Z. Zhou and D. Huang, "On efficient ciphertext-policy attribute based encryption and broadcast encryption: Extended abstract," in *Proc. 17th ACM Conf. Comput. Commun. Secur.*, 2010, pp. 753–755.
[29] Y. Song, Z. Li, Y. Li, and J. Li, "A new multi-use multi-secret sharing scheme based on the duals of minimal linear codes," *Secur. Commun. Netw.*, vol. 8, no. 2, pp. 202–211, 2015.
[30] J. Ning, Z. Cao, X. Dong, K. Liang, H. Ma, and L. Wei, "Auditable σ-time outsourced attribute-based encryption for access control in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 94–105, Jan. 2018.
[31] J. Borgh, E. Ngai, B. Ohlman, and A. M. Malik, "Employing attribute-based encryption in systems with resource constrained devices in an information-centric networking context," in *Proc. IEEE Global Internet Things Summit (GIoTS)*, Jun. 2017, pp. 1–6.
[32] J. Li, F. Sha, Y. Zhang, X. Huang, and J. Shen, "Verifiable outsourced decryption of attribute-based encryption with constant ciphertext length," *Secur. Commun. Netw.*, vol. 2017, Jan. 2017, Art. no. 3596205.
[33] J. Wang, C. Huang, N. N. Xiong, and J. Wang, "Blocked linear secret sharing scheme for scalable attribute based encryption in manageable cloud storage system," *Inf. Sci.*, vol. 424, pp. 1–26, Jan. 2018.
[34] M. Stadler, "Publicly verifiable secret sharing," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 1996, pp. 190–199.
[35] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.
[36] J. Benaloh and J. Leichter, "Generalized secret sharing and monotone functions," in *Proc. Conf. Theory Appl. Cryptogr.* Berlin, Germany: Springer, 1990, pp. 27–35.
[37] M. Bertilsson and I. Ingemarsson, "A construction of practical secret sharing schemes using linear block codes," in *Proc. Int. Workshop Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 1992, pp. 67–79.
[38] F. E. Brickell, "Some ideal secret sharing schemes," in *Proc. Workshop Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 1990, pp. 468–475.
[39] J. Simonis and A. Ashikhmin, "Almost affine codes," *Des., Codes Cryptogr.*, vol. 14, no. 2, pp. 179–197, 1998.
[40] V. Nikov and N. Svetla, "New monotone span programs from old," IACR Cryptol. ePrint Archive, Bellevue, WA, USA, Tech. Rep. 282, 2004.
[41] M. Karchmer and A. Wigderson, "On span programs," in *Proc. Struct. Complex. Theory Conf.*, May 1993, pp. 102–111.
[42] M. H. Dehkordi and R. Ghasemi, "A lightweight public verifiable multi secret sharing scheme using short integer solution," *Wireless Pers. Commun.*, vol. 91, no. 3, pp. 1459–1469, 2016.
[43] Y. Sun, G. Li, Z. Lin, F. Xiao, and X. Yang, "A completely fair secret sharing scheme without dealer," in *Proc. IEEE Int. Conf. Consumer Electron.-Taiwan (ICCE-TW)*, May 2016, pp. 35–36.
[44] J. L. Massey, "Minimal codewords and secret sharing," in *Proc. 6th Joint Swedish-Russian Int. Workshop Inf. Theory*, 1993, pp. 276–279.
[45] G. R. Blakley and G. A. Kabatianskii, "Linear algebra approach to secret sharing schemes," in *Proc. Error Control, Cryptol., Speech Compress.* Berlin, Germany: Springer, 1994, pp. 33–40.

JING WANG is currently pursuing the Ph.D. degree with Wuhan University. She holds a post-doctoral position at Sun Yat-sen University. Her research interests include security and privacy of cloud.

NEAL N. XIONG received the dual Ph.D. degrees in software engineering from Wuhan University and in dependable networks from the Japan Advanced Institute of Science and Technology. He is currently an Associate Professor at the Department of Mathematics and Computer Science, Northeastern State University. Before he attends Northeastern State University, he was with the Wentworth Technology Institution, Georgia State University, for many years. His research interests include cloud computing, security and dependability, parallel and distributed computing, networks, and optimization theory.

WANG JINHAI is currently pursuing the Ph.D. degree with Wuhan University. He is also an Associate Professor with Foshan University. His research interests include cloud computing and high performance computing.

WEI-CHANG YEH (SM'09) received the M.S. and Ph.D. degrees from the Department of Industrial Engineering, The University of Texas at Arlington. He is currently a Distinguished Professor with the Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Taiwan. His research interests include network reliability theory, graph theory, the deadlock problem, linear programming, and scheduling. He is a member of INFORMS.