IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON SECURITY AND TRUSTED COMPUTING FOR INDUSTRIAL INTERNET OF THINGS

Received March 28, 2018, accepted May 10, 2018, date of publication May 15, 2018, date of current version June 5, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2836350

A Novel Efficient Pairing-Free CP-ABE Based on Elliptic Curve Cryptography for IoT

SHENG DING[1], CHEN LI[2], AND HUI LI[1]
[1]School of Cyber Engineering, Xidian University, Xi'an 710071, China
[2]School of Telecommunications Engineering, Xidian University, Xi'an 710071, China

Corresponding author: Chen Li (lichen@xidian.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0802203 and in part by the National Natural Science Foundation of China under Grants 61672411, 61772404, and U1401251.

ABSTRACT Ciphertext-policy attribute-based encryption (CP-ABE) is a promising cryptographic technique that integrates data encryption with access control for ensuring data security in IoT systems. However, the efficiency problem of CP-ABE is still a bottleneck limiting its development and application. A widespread consensus is that the computation overhead of bilinear pairing is excessive in the practical application of ABE, especially for the devices or the processors with limited computational resources and power supply. In this paper, we proposed a novel pairing-free data access control scheme based on CP-ABE using elliptic curve cryptography, abbreviated PF-CP-ABE. We replace complicated bilinear pairing with simple scalar multiplication on elliptic curves, thereby reducing the overall computation overhead. And we designed a new way of key distribution that it can directly revoke a user or an attribute without updating other users' keys during the attribute revocation phase. Besides, our scheme use linear secret sharing scheme access structure to enhance the expressiveness of the access policy. The security and performance analysis show that our scheme significantly improved the overall efficiency as well as ensured the security.

INDEX TERMS Access control, internet of things, CP-ABE, elliptic curve, pairing-free.

I. INTRODUCTION

The advent of Internet of Things (IoT) brings a revolutionary transformation to data management. Billions of devices, such as wearable devices, smartphones, smartcars, are connected to the Internet that they can share data with each other in this novel paradigm. As most of these devices are resource-constrained, data are generally stored in the cloud that people and devices can conveniently upload and download data anytime and anywhere as long as they can access the Internet. However, this causes a knotty problem that how to ensure the security of data. As data management is out of data owner's direct control, it is not only important to enforce strict control of the data access, but also hide it from the cloud service provider (CSP), which can not be fully trusted. To securely share data in an open distributed computing environment, common practice is to encrypt the data before storing it into the cloud. As we know, either symmetric or asymmetric key encryption can realize the function of encryption. However, symmetric key encryption needs to share a common session key in advance between data owner and data user. Sharing data in IoT systems makes it impossible to know every possible data user. Even if we have a list of all the data users, we have to repeatedly encrypt the data by each session key shared with them. It requires complicated key management and will definitely incur high computation and storage overhead. The same goes for asymmetric key encryption, as we also need to get every data user's public key, repeatedly encrypt the data and store multiple encrypted copies of same data into the cloud.

Sahai and Waters [1] solved the above problem by proposing a new cryptographic technique called attribute-based encryption (ABE). A data owner can specify access to the data as a boolean formula over a set of attributes. Everyone in the ABE system will be issued a private key that represents her attributes from an authority. No one can decrypt the ciphertext unless the attributes associated with her private key satisfy the boolean formula ascribed to the ciphertext. Bethencourt *et al.* [2] then proposed a new type of ABE where user private keys are specified by a set of attributes and the data owner can specify a more expressive access policy over these attributes, called CP-ABE. For example, suppose that a user intends to share her biomedical data and medical record with relevant doctors. The user may specify an access policy: (Chief Physician $\wedge$ Internal

Medicine ∧ (Hospital X ∨ Hospital Y)). In this way, the user could mean that her privacy data should only be seen by chief physicians of internal medicine from hospital X or hospital Y.

ABE perfectly combines data encryption and access control, but the efficiency problem is still a bottle neck limiting its development and application. Bilinear pairing has been regarded as the most expensive operation and more time-consuming compared with scalar multiplication in pairing-based cryptographic protocols. As we have experimented, the computation overhead of bilinear pairing is two or three times higher than that of scalar multiplication under a same elliptic curve. Hence, to reduce the calculation times of bilinear pairing as far as possible is a way to essentially improve the efficiency of ABE.

Our main contributions are summarized as follows:

1) We greatly improve the efficiency of ABE algorithms. Based on CP-ABE, we proposed a novel access control scheme for IoT systems, which no longer needs any complicated bilinear pairing. In this way, our scheme can be more effective and practical, especially for entities with limited computing capability and energy supply.
2) We designed a new way of key distribution that each data user's secret keys are generated and implicitly maintained by the attribute authority along with an attribute list. In this way, we can directly revoke a user or an attribute without updating other users' keys. It greatly decreases the computation and communication overhead caused by attribute revocation.
3) We use LSSS access structure to enhance the expressiveness of the access policy. We provided a elaborate security and performance analysis of our scheme, and the experimental results prove the efficiency of our scheme.

The paper is organized as follows: related work is summarized in Section 2, followed by preliminaries in Section 3. We propose the detailed construction of our data access control scheme for IoT systems in Section 4. Section 5 and section 6 presented the security and performance analysis respectively. The paper ends with conclusion in Section 7.

II. RELATED WORK

In the last decade, the arise of bilinear pairings helped solving many problems which were unrealistic in the field of cryptography [3]–[5]. Based on bilinear pairings, ABE has been proposed to realize the combination of data encryption and access control. In 2007, Ling and Newport [6] presented a CP-ABE scheme supporting AND gate access structure on both positive and negative attributes. Lewko and Waters [7] proposed a multi-authority CP-ABE scheme without the need of the collaboration between the attribute authorities. Considering the flexibility of the system, Horvath [8] proposed a multi-authority CP-ABE scheme, which can realize identity-based revocation. Hur [9] proposed a CP-ABE scheme which supports direct revocation on the attribute set of each user. The similar technologies have also been used in other researchers' work, such as [10]–[13], to ensure the information security in the cloud. Wang *et al.* [14] solved the key escrow problem in CP-ABE schemes as well as enhanced the attribute expressiveness. Guo *et al.* [15] proposed a CP-ABE scheme with constant size keys, and the number of the decryption key is independent of attribute number. Nonetheless, CP-ABE schemes are computationally intensive, which include a number of pairing operations and exponentiations. This greatly limits their uses on the resource-constrained devices in the IoT systems.

As we know, the efficiency of pairing-based cryptographic protocols depends on the speed of the computation of pairings. Hence, lots of research work has been done for enhancing its efficiency [16]–[20]. To optimize the ECC protocols, Freeman *et al.* [21] classified some pairing-friendly elliptic curves and introduced the constructions of them as well as some relevant optimization techniques. In [22], Scott analyzed how to select the pairings and the curves for improving the efficiency of ABE schemes. Rivain [23] also talked about how to implement scalar multiplication faster in ECC schemes in detail.

One way to decrease the computation overhead for the data users is to delegate the complicated computation of pairings to other entities with more computing power. Chevallier-Mames *et al.* [24] first presented a scheme for outsourcing the complicated computation of pairings under an untrusted server model. But the computation overhead is still higher compared with Chen *et al.* [25]. In [25], they proposed a novel computation outsourcing algorithm under one-malicious version of two untrusted program model. It is more efficient but the security model is unrealistic in practical application.

A more direct way is to delegate part of decryption to the cloud. In 2011, Green *et al.* [26] proposed an ABE scheme with the decryption outsourced. In their scheme, the attribute secret keys are composed of two parts: El Gamal type keys and transformation keys. The proxy can partial decrypt the ciphertexts with the help of the transformation keys, leaving only a simple El Gamal ciphertext to be decrypted for the data user. Li *et al.* [27] also improved this way to make it support outsourcing both the key distribution and the decryption. However, the computation overhead is just shifted to the proxy or the cloud server. For the whole system, the overhead has not been effectively reduced.

To essentially optimize ABE algorithms, our way is to replace complicated bilinear pairings with other more efficient arithmetic operations. Odelu and Das [28] proposed a CP-ABE scheme with constant size keys based on elliptic curve cryptography, but it only supports AND gate access structure, which limits its flexibility. Subsequently, based on RSA, they also proposed a novel CP-ABE scheme [29] with constant size keys and ciphertexts. Although the time complexity of the encryption and decryption are both $\mathcal{O}(1)$, it only supports AND gate access structure.

III. PRELIMINARY

A. ACCESS STRUCTURE

Definition 1 (Access Structure [30]): Let $\{A_1, A_2, \cdots, A_n\}$ be a set of attributes. A collection $\mathbb{S} \subseteq 2^{\{A_1, A_2, \cdots, A_n\}}$ is monotone only if $\forall B, C$: if $B \in \mathbb{S}$ and $B \subseteq C$ then $C \in \mathbb{S}$. An access structure is a collection $\mathbb{S}$ of nonempty subsets of $\{A_1, A_2, \cdots, A_n\}$, i.e. $\mathbb{S} \subseteq 2^{\{A_1, A_2, \cdots, A_n\}} \backslash \{\emptyset\}$. The sets in $\mathbb{S}$ refer to the authorized sets. Otherwise, they are called unauthorized sets.

In ABE system, the access structure stipulates that an eligible user should have the corresponding attributes in it. For example, a boolean formula $A \wedge B \wedge (C \vee D)$ represents that the one who can decrypt the ciphertext must have attributes A, B, C or A, B, D. It can also be expressed in a more comprehensible way, like an access tree, as shown in Fig.1. And [30] indicated that any monotonic access structure can be converted into an LSSS representation by standard techniques.

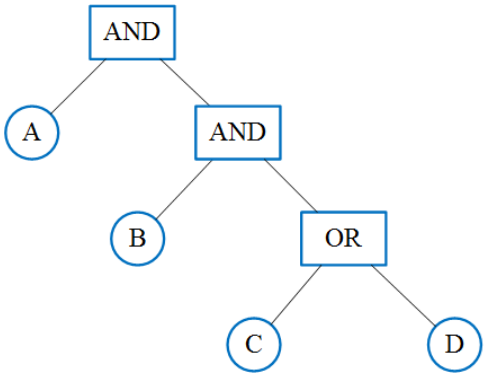


FIGURE 1. Access tree.

B. LSSS

LSSS is designed for applying highly expressive monotone access structures in CP-ABE schemes.

Definition 2 (LSSS [30]): A secret sharing scheme over a set of parties is called linear if

1) A vector over $\mathbb{Z}_p$ is generated by the shares for each party.
2) A matrix A with n rows and l columns is designed to generate the shares. For $i \in \{1, \cdots, n\}$, each row i is labeled by the function ρ to make it associate with one of the parties. Let $s \in \mathbb{Z}_p$ be the secret to be shared. A column vector $v = (s, r_2, \cdots, r_l)$ chooses s as its first element, and randomly chooses the rest $r_2, \cdots, r_l \in \mathbb{Z}_p$. Then $A \cdot v$ becomes the vector of n shares of the secret s. The share $(A \cdot v)_i$ belongs to party $\rho(i)$.

As illustrated in [30], if a linear secret sharing scheme is defined as above, it also satisfies the linear reconstruction property. To be specific, let $S \in \mathbb{A}$ be any authorized set, where $\mathbb{A}$ is the access structure, and let I be the corresponding set of row number $\{i : \rho(i) \in S\}$. Then, there must exist constants $\{c_i \in \mathbb{Z}_p\}_{i \in I}$, such that if $\{\lambda_i\}$ are the shares of the secret s, the secret can be recovered by computing $\Sigma_{i \in I} c_i \lambda_i = s$.

The matrix is generated by algorithm [7], whose input is an access tree representing a monotone boolean formula. The non-leaf nodes on the tree are either **AND** or **OR** gate and the leaf nodes are attributes. The output of the algorithm is the a LSSS matrix and the number of rows of the matrix is equal to the number of leaf nodes on the input access tree. The algorithm goes down the tree, and labels the nodes as follows:

- if the parent node is an **OR** gate labeled with a vector v, the algorithm labels both child nodes as v and keeps the counter c unchanged;
- if the parent node is an **AND** gate labeled with a vector v, the algorithm pads v with 0 at the end to change its length to c, then the algorithm labels its right child node with the vector $v \| 1$, $\|$ denotes concatenation here and the left child node with the vector $(0, \cdots, 0) \| -1$, where the length of $(0, \cdots, 0)$ is c, and increases the value of c by 1.

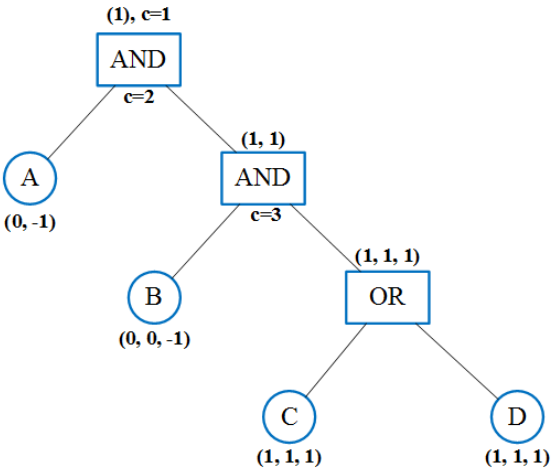


FIGURE 2. Label the access tree to generate an LSSS matrix.

The algorithm labels the tree as described in Fig.2. The vectors labeled on the leaf nodes make up the rows of an LSSS matrix. If the vectors are different in length, the algorithm will pad the shorter ones with 0 at the end to make them have the same length as the longest one.

With the help of algorithm [7], the access tree in Fig.1 can generate an LSSS matrix as below.

$$A = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{matrix} \rho(1) = A \\ \rho(2) = B \\ \rho(3) = C \\ \rho(4) = D \end{matrix}$$

ρ maps each row of the matrix to attributes A, B, C and D respectively. Given an attribute set S, the LSSS is said to be satisfied by S only if the rows of the matrix labeled by the attributes in S include the vector $(1, 0, \cdots, 0)$ in their span.

C. ELLIPTIC CURVE CRYPTOGRAPHY

In 1985, Neal Koblitz and Victor Miller first proposed the definition of ECC. The curves in ECC is defined by an equation:

$$y^2 = x^3 + ax + b, \ where\ 4a^3 + 27b^2 \neq 0.$$

The security of ECC is based on the elliptic curve discrete logarithm problem (ECDLP), which is defined as follows.

Given a base point, it is difficult to compute the discrete logarithm of a random elliptic curve element. To be specific, let E be an elliptic curve defined over a finite field $GF(q)$, G be a generator with order r. Given a $Q = kG, k \in \mathbb{Z}_r$, to find the integer k in polynomial time is almost infeasible.

Compared with RSA, ECC is able to ensure the same security with a smaller key size, as solving the ECDLP is more difficult than factoring an integer. An ECC encryption protocol can be generally divided into three steps. The plaintext message should be first mapped to a point Q on the elliptic curve. And then the encryption protocol between two parties, e.g. Alice and Bob, is executed as follows.

1) Key generation.
   a) Alice and Bob first agree on a same elliptic curve $y^2 = x^3 + ax + b(mod\ p)$, with a generator point G.
   b) Alice selects an integer $n_a \in \mathbb{Z}_p$ as the private key and computes a point $P_a = n_aG$ as the corresponding public key.
   c) Bob selects an integer $n_b \in \mathbb{Z}_p$ as the private key and computes a point $P_b = n_bG$ as the corresponding public key.
2) Encryption.
   To encrypt Q, Alice first randomly selects an integer $k \in \mathbb{Z}_p$ and computes the two parts of the ciphertext, $C_1 = kG$ and $C_2 = Q + kP_b$. And then Alice sends C_1 together with C_2 to Bob.
3) Decryption.
   After receiving the ciphertext, Bob may multiply C_1 by his private key n_b and subtracts it from C_2. That is,

$$\begin{aligned} C_2 - n_bC_1 &= (Q + kP_b) - n_b(kG) \\ &= (Q + kn_bG) - n_bkG \\ &= Q \end{aligned}$$

Finally, Bob can map Q back to the plaintext message.

D. SYSTEM MODEL AND SECURITY MODEL

1) SYSTEM MODEL

In order to have a general view of our PF-CP-ABE access control scheme for IoT systems, we first review the classical CP-ABE protocol and give the system model of our scheme.

A CP-ABE system generally consists of five algorithms: System Setup, Authority Setup, Key Generation, Encryption and Decryption, as defined below [7].

System Setup $(k) \rightarrow PP$. The system setup algorithm takes a security parameter k as input and then outputs all of the necessary public parameters (PP) for the system.

Authority Setup $(PP) \rightarrow PK, SK$. Based on the PP generated in the first step, the attribute authority creates the public keys (PK) and secret keys (SK) for itself and each attribute in the system.

Key Generation $(PP, i, GID, SK) \rightarrow SK_{i,GID}$. The key generation algorithm takes the public parameters, an attribute i, an identity GID, and the SK of the attribute authority as input. It outputs an attribute secret key $SK_{i,GID}$ corresponding to an GID and issues it to eligible users.

Encryption $(PP, M, (A, \rho), \{PK_i\}) \rightarrow CT$. Given a message M, an access matrix (A, ρ) and the public keys of all of the attributes used in the access policy, the encryption algorithm outputs a ciphertext CT.

Decryption $(PP, CT, \{SK_{i,GID}\}) \rightarrow M$. If a set of attribute secret keys owned by a certain user satisfy the access matrix of the ciphertext, the decryption algorithm can successfully recover the message M. Otherwise the decryption fails.

Our PF-CP-ABE mainly consists of four entities: CSP, attribute authority, data owner and data user, as described in Fig.3.

Attribute Authority. The attribute authority is the only fully trusted entity in the system besides data user. It is in charge of issuing and revoking users' attributes according to their roles or identities in the system. The secret key of each attribute is generated by it and the corresponding public key is published to all of the users in the system. Each user was bound with a global identity (GID), with which to register in the system. An attribute list of each user is also maintained by the attribute authority to record their owned attributes. In the phase of decryption, the attribute authority assists the data user with part of the decryption.

Cloud Service Provider. The CSP is an honest but curious entity. As same as defined in other schemes, it will work in strict accordance with the protocol but may be curious about the content of the ciphertexts. The CSP can store the encrypted data instead of the data owner and provide data access service later.

Data Owner. The data owner can define access control policy over attributes in the system and under which encrypt the data before outsourcing it to the cloud. Only the user, with enough attributes satisfying the access policy, can decrypt the ciphertexts. The access control happens inside the cryptography.

Data User. The data user can ask for the access to the encrypted data stored in the CSP. Only if there is an attribute match between the data user and the access policy can the ciphertext be successfully decrypted. The data user are not fully trusted as they may collude with each other, driven by interests, to decrypt the ciphertext which none of them can decrypt independently.

2) SECURITY MODEL

We now give the security model of our PF-CP-ABE access control scheme. The model is defined by a game between a challenger and an adversary, as described below.

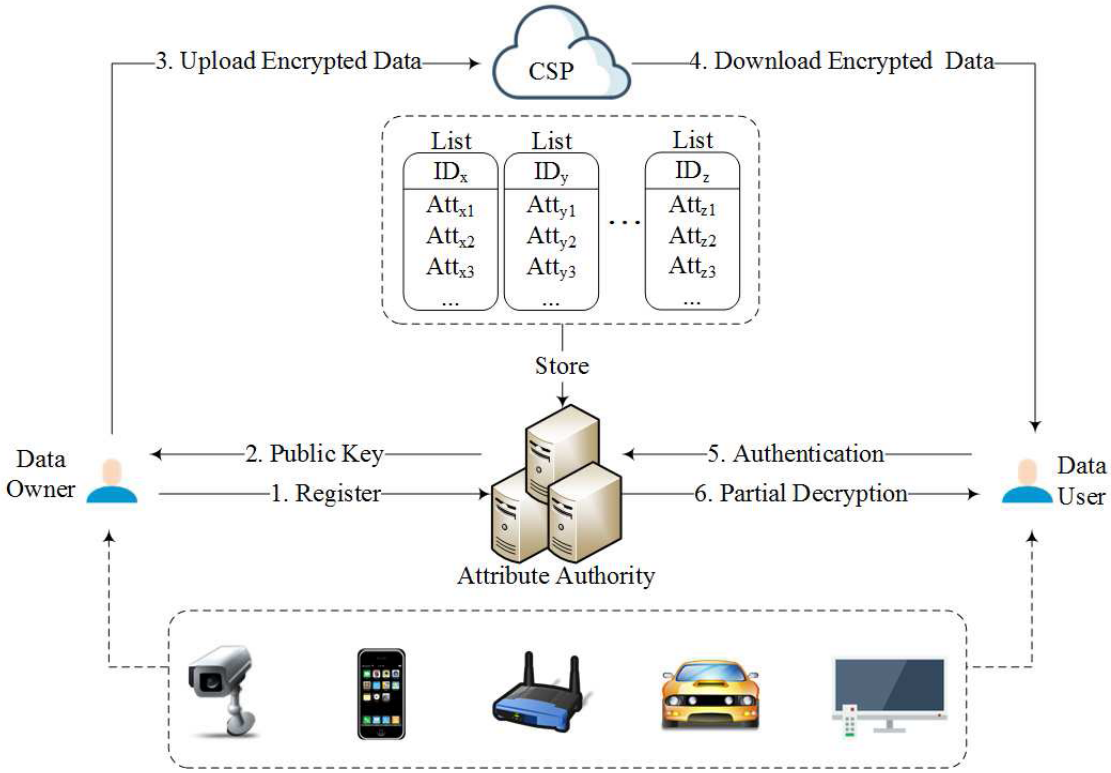


FIGURE 3. System Model.

Initialization. The adversary first chooses a challenge access structure (A, ρ) and then sends it to the challenger.

Setup. The setup algorithm generates the necessary public parameters for the system as well as the public and secret key pair for each attribute. The challenger sends the public keys to the adversary.

Phase 1. The adversary can adaptively query for the attribute secret keys with a restriction that any set of the keys can not decrypt the challenge ciphertext. The challenger responds by recording the attributes on the attribute list corresponding the adversary's GID.

Challenge Phase. The adversary selects two equal length messages $M_0, M_1 \in P$ and submits them to the challenger. Then the challenger flips a coin $\beta \in \{0, 1\}$ and sends the encryption of M_β under access matrix (A, ρ) to the adversary.

Phase 2. The adversary may submit additional key queries (i, GID) with the same restriction in Phase 1.

Guess. The adversary may output a guess β' for β. The adversary advantage in this game is defined as $Pr[\beta' = \beta] - \frac{1}{2}$.

Definition 3: Our PF-CP-ABE is selective CPA secure if any polynomial time adversary has at most a negligible advantage to win this security game.

3) DIFFIE-HELLMAN ASSUMPTION

The definition of the decisional Diffie-Hellman (DDH) assumption on elliptic curve is described below. A challenger selects a cyclic group P of prime order r. Let G be a generator of P and a, b be randomly chose from Z_r. If the challenger gives the adversary a tuple (G, aG, bG), it must be difficult for the adversary to distinguish a valid element $abG \in P$ from a random element $R \in P$. The advantage of an algorithm $\mathcal{B}$ in breaking the DDH assumption in P is ε if

$$|Pr[\mathcal{B}(G, aG, bG, Z = abG) = 0] - Pr[\mathcal{B}(G, aG, bG, Z = R) = 0]| \geq \varepsilon$$

Definition 4: The DDH assumption holds if all polynomial time algorithms have at most a negligible advantage in solving the DDH problem.

IV. PROPOSED SCHEME

In this section, we give the detailed construction of our PF-CP-ABE scheme for efficient and secure data sharing. To thoroughly improve the performance of the whole algorithm, we replace complicated bilinear pairing with simple scalar multiplication on elliptic curves, thereby simplifying the calculation. To be specific, the data owner first encrypts the message M with sG, where G is a generator of a cyclic subgroup of an elliptic curve with order r, and s is a random chosen value in $\mathbb{Z}_r$. Then the encryption algorithm splits the value s into shares λ_x according to the LSSS matrix, and a value 0 is split into shares ω_x in the same way. To recover the message M, the data user needs to combine her attribute keys with the ciphertext elements to get the blinding factor sG.

In order to prevent collusion attacks, each attribute belonging to a certain user will be bound with a global identity. In this way, different user's attributes cannot be successfully combined in decryption. The decryption algorithm will introduce some new terms of the form $H(GID)\omega_x nG$, where nG is the public key of the attribute authority. If the data user has a satisfying set of keys with a same identity, these redundant terms will be cancel from the final result, as ω_x are shares of 0. If two users with different identities intend to collude with each other, there will be different terms of the form $H(GID)\omega_x nG$ which can not be eliminated. This will result in a failure of the recovery of sG, as well as the message M. Our PF-CP-ABE is composed of the following five algorithms:

Setup Let $GF(q)$ be a finite field of order q, E be an elliptic curve defined over $GF(q)$ and G be an element of a large prime order r in E. The point G generates a cyclic subgroup of E, in which the elliptic curve discrete logarithm problem (EC-DLP) is intractable. In addition, a hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_r^*$ is chose to map GID to elements of $\mathbb{Z}_r$.

Authority Setup The attribute authority chooses a random number $n \in \mathbb{Z}_r$ as its master secret key and publishes nG as its public key. For each attribute i in the system, the attribute authority randomly selects a $k_i \in \mathbb{Z}_r$ and publishes $PK_i = k_i G$ as its public key. For each data user in the system, the authority maintains an attribute list corresponding to its GID.

Key Generate To generate a key of an attribute i for a user with GID, the attribute authority can computes

$$SK_{i,GID} = k_i + H(GID)n,$$

and record this attribute i on its corresponding attribute list.

Encrypt The encryption algorithm consists of following stages:

- 1. The plaintext message is firstly mapped to a point M on the elliptic curve E. It chooses a random $s \in \mathbb{Z}_r$ and computes

$$C_0 = M + sG.$$

- 2. The encryption algorithm takes in the access policy made by the data owner and then outputs an $n \times l$ access matrix A with ρ mapping its rows to attributes.
- 3. It chooses a random vector $v \in \mathbb{Z}_r^l$ with s as its first entry and let λ_x denote $A_x \cdot v$, where A_x is row x of A. A random vector $u \in \mathbb{Z}_r^l$ with 0 as its first entry is also chose and let ω_x denote $A_x \cdot u$.
- 4. The ciphertext is computed as:

$$C_{1,x} = \lambda_x G + \omega_x PK_{\rho(x)}, C_{2,x} = \omega_x G, \forall x.$$

Decrypt To decrypt the ciphertext, the data user should first find out a satisfying set of rows A_x of A such that $(1, 0, 0, \cdots, 0)$ is in the span of these rows, and then submit its GID with $(C_{2,x}, \rho(x))$ of each such x. The authority verifies its identity and whether it does possess these attributes according to its attribute list. If the request is valid, for each $(C_{2,x}, \rho(x))$, the authority computes:

$$\begin{aligned}\sum C_{2,x} SK_{\rho(x),GID} &= \sum(\omega_x G(k_{\rho(x)} + H(GID)n) \\ &= \sum(\omega_x k_{\rho(x)} G + \omega_x H(GID)nG).\end{aligned}$$

Then the authority sends the result to the data user in a secure channel. With the result, the data user can compute

$$\begin{aligned}&\sum C_{1,x} - \sum C_{2,x} SK_{\rho(x),GID} \\ &= \sum(\lambda_x G + \omega_x PK_{\rho(x)}) - \sum(\omega_x k_{\rho(x)} G + \omega_x H(GID)nG) \\ &= \sum(\lambda_x G - \omega_x H(GID)nG)\end{aligned}$$

for all such x.

The data user then selects constants $c_x \in \mathbb{Z}_r$ such that $\sum_x c_x A_x = (1, 0, \cdots, 0)$ and computes:

$$\sum_x c_x(\lambda_x G - \omega_x H(GID)nG) = sG,$$

as $v \cdot (1, 0, \cdots, 0) = s$ and $u \cdot (1, 0, \cdots, 0) = 0$. Finally, the data user can just compute:

$$C_0 - sG = M$$

and then map M back to the message.

Attribute Revocation Our scheme facilitates the attribute revocation as users' secrete key are generated and implicitly maintained by the attribute authority, which make it possible to directly revoke an user or an attribute without updating other users' keys during the attribute revocation phase. To revoke a user, the attribute authority needs to delete its attribute list corresponding to its GID. To revoke an attribute, the attribute authority needs to delete the public key of this attribute. To revoke an attribute owned by a user, the attribute authority needs to delete this attribute from its attribute list.

V. SECURITY ANALYSIS

Now we prove our PF-CP-ABE access control scheme for IoT systems to be secure under the DDH assumption.

Theorem 1: If there exists a PPT adversary $\mathcal{A}$ that can break the proposed scheme with a non-negligible advantage $\varepsilon > 0$, then there is a PPT algorithm $\mathcal{B}$ that can distinguish a DDH tuple from a random tuple with advantage $\frac{\varepsilon}{2}$.

Let G be the generator of group P of a large prime order r. Firstly, the DDH challenger $\mathcal{C}$ first randomly chooses $a, b \in \mathbb{Z}_r$, $\beta \in \{0, 1\}$ and $R \in P$. We let Z to be abG if $\beta = 0$, otherwise $Z = R$. The challenger $\mathcal{C}$ sends a tuple (G, aG, bG, Z) to $\mathcal{B}$. Then $\mathcal{B}$ plays the role of challenger instead of $\mathcal{C}$ in the following game.

- **Initialization**. $\mathcal{A}$ first chooses a challenge access structure (A, ρ) and then sends it to $\mathcal{B}$.
- **Setup**. In order to create the public key for each attribute i in the system to adversary $\mathcal{A}$, $\mathcal{B}$ randomly chooses $k_i \in \mathbb{Z}_r$ and let $PK_i = k_i aG$. For the attribute authority, $\mathcal{B}$ chooses a random $n \in \mathbb{Z}_r$ and publish nG as its public key. As k_i is randomly chose, the public parameters are randomly distributed as well.

- **Phase 1**. $\mathcal{A}$ adaptively submits pairs (i, GID) to $\mathcal{B}$ to request the corresponding secret key with the following constraints. For each identity GID, we let V_{GID} denote the subset of rows of A labeled by attributes i for which the attacker has queried (i, GID). For each GID, we require that the subspace spanned by V_{GID} must not include $(1, \cdots, 0)$. In other words, the attacker cannot ask for a set of keys that allow decryption. $\mathcal{B}$ responds by recording this attribute i on the attribute list corresponding to GID. Then $\mathcal{B}$ chooses a random $t \in \mathbb{Z}_r$ and computes $k_i a + t$ as its secret key.
- **Challenge**. $\mathcal{A}$ selects two equal length messages $M_0, M_1 \in P$ and submits them to $\mathcal{B}$. $\mathcal{B}$ flips a coin β and chooses a random $s \in \mathbb{Z}_r$. It creats $C = M_\beta + sG$. Then $\mathcal{B}$ randomly chooses a vector $v \in \mathbb{Z}_r^l$ with s as its first entry and let λ_x denote $A_x \cdot v$, where A_x is row x of A. A random vector $u \in \mathbb{Z}_r^l$ with 0 as its first entry is also chose and let ω_x denote $A_x \cdot u$. Finally $\mathcal{B}$ generates the challenge ciphertext $C_{2,x} = \omega_x bG$ and $C_{1,x} = \lambda_x G + k_{\rho(x)}\omega_x Z$, then returns the challenge ciphertext $CT = \{(A, \rho), C, C_{1,x}, C_{2,x}\}$ to adversary $\mathcal{A}$.
- **Phase 2**. Same as Phase 1. The adversary $\mathcal{A}$ may submit additional secret key queries (i, GID) without violating the constraints.
- **Guess**. $\mathcal{A}$ outputs a guess β' of β. Then $\mathcal{B}$ outputs 0 to indicate that $Z = abG$ in the above game if $\beta' = \beta$; otherwise, $\mathcal{B}$ outputs 1 to guess that $Z = R$.

If $Z = abG$, it is a real ciphertext. In this case, $\mathcal{A}$'s advantage is ε as defined in the assumption. Thus

$$Pr[\mathcal{B}(G, aG, bG, Z = abG) = 0] = \frac{1}{2} + \varepsilon.$$

If $Z = R$, it is completely random from adversary $\mathcal{A}$'s view. Hence,

$$Pr[\mathcal{B}(G, aG, bG, Z = R) = 0] = \frac{1}{2}.$$

At last, $\mathcal{B}$'s advantage to break this security game is

$$\begin{aligned}\mathcal{B} &= \frac{1}{2}(Pr[\mathcal{B}(G, aG, bG, Z = abG) = 0] \\ &\quad + Pr[\mathcal{B}(G, aG, bG, Z = R) = 0]) - \frac{1}{2} \\ &= \frac{1}{2}(\frac{1}{2} + \varepsilon + \frac{1}{2}) - \frac{1}{2} \\ &= \frac{\varepsilon}{2}.\end{aligned}$$

A. DATA SECURITY

In our PF-CP-ABE, only valid users who possess a certain attribute can be granted its corresponding secret key k_i from the attribute authority. As the protocol is based on ECC, in which ECDLP is intractable, an invalid user without the attribute cannot get any information about its secret key k_i from corresponding public key $k_i G$ in polynomial time.

The message is implicit in ciphertext C_0. Suppose M can be mapped to mG, where $m \in \mathbb{Z}_r$, as s is randomly chose by the data owner, $C_0 = (m + s)G$ is just a random point on the elliptic curve in attacker's view. Due to ECDLP, the attacker can not get any valuable information about M without s. By means of secret sharing scheme, s is a secret split by λ_x and can be recovered only when there are sufficient shares, that is to say only if the data user has a satisfying set of attributes can it decrypt the ciphertext. For any invalid user, who does not have the attributes claimed by the access policy, there does not exist attributes corresponding to rows A_x, such that $\sum_x c_x \lambda_x = (1, 0, \cdots, 0)$. Thus, s, the first entry of vector v, cannot be calculated. Hence, our PF-CP-ABE indeed ensures data security.

B. FORWARD SECURITY

Our PF-CP-ABE guarantees forward Security of the outsourced data against revoked user. Forward security ensures that any revoked user can not obtain the access to future data. As keys are issued to users after registering in other schemes, the traditional solution is to generate new keys for all other users and re-encrypt all of the affected data. It obviously increases the overhead of both computation and communication. In our PF-CP-ABE, keys are stored in the attribute authority. For users, all they can get are just public keys of attributes and ciphertexts. As we mentioned above, users can get nothing about the secret key of a certain attribute from its corresponding public key. To revoke a user, the authority can just delete the attribute list corresponding to its GID. When the revoked user attempt to decrypt, the attribute authority will reject its request as its GID is not in the system and the authority can not determine whether the user has the attribute according to its attribute list. To revoke an attribute of a certain user, the attribute authority can just modify its attribute list. The decrypt request will also be reject as the attribute claimed is not in the list. Hence, forward security is guaranteed in our scheme.

C. COLLISION RESISTANT

To ensure the access control correct, the proposed scheme must be capable of resisting collusion attack. In other words, if multiple users collude with each other, they can not decrypt the ciphertexts except if one of them can decrypt it independently at least. We use GID to tie together the various attributes belonging to a specific user so that they cannot be successfully combined with others' attributes in decryption. For example, suppose Alice intends to collude with Bob to decrypt a ciphertext under an access policy $A \wedge B \wedge (C \vee D)$. Alice only has attribute A and B, Bob only owns C. It's obvious that neither of them can decrypt the ciphertext on their own. If they collude with each other, Alice will get results from the authority just like

$$\lambda_x G - H(GID_{Alice})\omega_x nG$$

and Bob will get

$$\lambda_x G - H(GID_{Bob})\omega_x nG,$$

for some x. Normally, a valid user can choose constants $c_x \in \mathbb{Z}_r$ such that $\sum_x c_x A_x = (1, 0, \cdots, 0)$ and then

TABLE 1. The summary of the new notations used in the analysis.

Notation	Role
N_a	Total number of attributes in the system
N_r	Number of rows in access matrix A
C_G	Exponentiation or multiplication in the group
C_e	Bilinear pairing
C_{Z_N}	Field exponentiation in Z_N,$\mid N \mid = 1024$
D_a	Minimum number of attributes satisfying the access policy
U_a	Set of user attributes

figure out sG. As Alice and Bob have two different GID,

$$H(GID_{Alice}) \neq H(GID_{Bob})$$

makes

$$\sum_x H(GID)\omega_x nG \neq 0,$$

they can not successfully collude to recover sG. In this way, our PF-CP-ABE is collusion resistant.

VI. PERFORMANCE ANALYSIS

In this section, we first analyze the storage and the communication overhead of our scheme, then compare the computation efficiency with other schemes through experiment.

In order to facilitate understanding, new notations which we used in the comparison are listed in Table 1.

A. STORAGE OVERHEAD

As the majority of the devices in IoT systems are resource-constrained, storage overhead is an important factor that needs to be considered. Hence, we analyze the storage overhead on the attribute authority, each user and the cloud server respectively in detail.

- **Attribute Authority**. The attribute authority is responsible for generating, issuing and revoking the attribute keys for the users in the system. In addition to all of the information about the attributes, the attribute authority needs to store an attribute list for each user in the system. Thus, the storage overhead on the attribute authority is linear to the number of attributes and users in the system.
- **Each User**. Generally speaking, the storage of the public parameters and the secret keys issued by the attribute authority is the most of the storage overhead on each user. However, in our scheme, there is no need for users to store their secret keys in local as these keys are generated and maintained in the form of attribute list by the authority. Hence, each user needs only store the public parameters in local for any further encryption.
- **Server**. Just like other schemes, the ciphertexts contribute the vast majority of the storage overhead on the server. In our scheme, each ciphertext consists of three parts and is linear to the number of the attributes used in the encryption.

B. COMMUNICATION OVERHEAD

Compared with other schemes, the data user in our scheme needs the attribute authority's help to complete the decryption as the secret key is implicitly maintained in the attribute authority. It seems that our scheme increases the communication cost. However, we greatly decrease the overhead of attribute revocation as the attribute authority needs to update any other's secret keys in traditional schemes which is indeed a huge overhead. In our scheme, the attribute authority can just modify the attribute list of the one to be revoked to complete the attribute revocation without affecting others in the system. Therefore, our scheme obviously decrease the overall communication overhead.

C. COMPUTATION OVERHEAD

Table 2 shows the computational overhead comparison among our PF-CP-ABE and other schemes. To analyze computation efficiency systematically, the rough estimations of various cryptographic operations are provided in Table 3.

It seems that [29] costs the least in encryption and decryption as its computation overhead is independent of the number of attributes. However, the tradeoff is that it only supports AND gate access structure which is insufficient to handle the fine-grained access control. And it uses RSA as its cryptography primitive whose required key length, at least 2048 bits, is too long for resource-constrained devices in IoT systems. References [15] and [28] are both efficient CP-ABE schemes with constant-size secret keys. The problem they both face is that the overhead is proportional to the difference between the number of the attributes used in the access policy and the total number of the attributes defined in the system. To decrease the computation overhead of encryption and decryption, the access policy needs to be quite complicated. And these two schemes also only support AND gate access structure. Reference [7] use expressive LSSS access structure and the computation overhead is proportional to the number of attributes used in encryption, which is similar to us to some extent.

Therefore, we compare the efficiency of our PF-CP-ABE with scheme [7] in experimental aspect. For a fair comparison, we let the order of the group in scheme [7] be a prime, as the subgroups G_{p2} and G_{p3} in their protocol are only used to apply the dual system encryption technique for security proof. Then we implement our scheme and [7] with Intel Pentium G620 CPU at 2.60GHz and 2GB RAM. The system runs Ubuntu Linux 16.04LTS. Based on the pairing-based cryptography library (version 0.5.14), the implementation uses a 160-bit elliptic curve group based on the supersingular curve $y^2 = x^3 + x$ over a 512-bit finite field to achieve a 80-bit

TABLE 2. Comparison.

Scheme	Our scheme	[7]	[28]	[29]	[15]
Encryption	$(4N_r+1)C_g$	$C_e+(7N_r+2)C_g$	$(N_a-\mid A\mid+2)C_g$	$3C_{Z_N}$	$(2(N_a-\mid A\mid)+2)C_g$
Decryption	$(D_a+1)C_g$	$2D_aC_e+4D_aC_g$	$(N_a-\mid A\mid+3)C_g$	$3C_{Z_N}$	$(2(\mid U_a\mid-\mid A\mid)+1)C_g+3C_e$

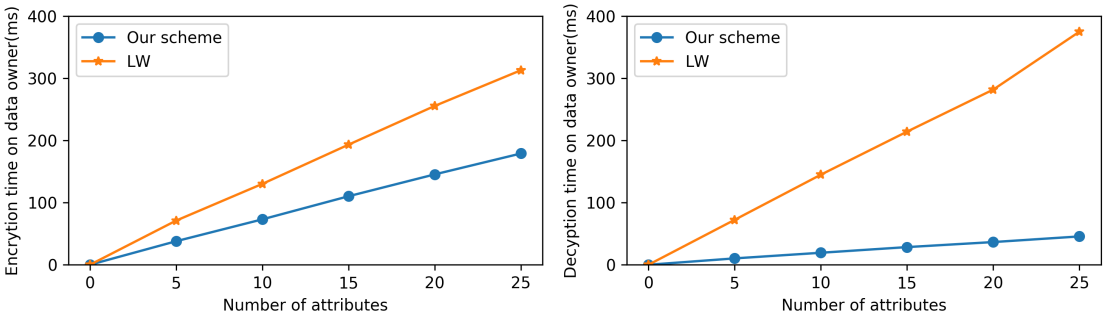


FIGURE 4. Comparisons of encryption and decryption time.

TABLE 3. Execution time of various operations used in the experiment.

C_e	C_G	C_{Z_N}
3.62ms	1.71ms	1.03ms

security level. All experimental results represent the average values for 30 rounds.

We compare the computing time incurred in encryption and decryption, as this two parts have direct impact on the user experience. Fig.4 shows the comparison of encryption and decryption time with different number of attributes. From Fig.4 we can see that our PF-CP-ABE saves time nearly by half in the execution time of encryption, as our scheme requires less computation of group elements, no matter it is scalar multiplication or exponentiation. And it's obvious that our PF-CP-ABE significantly lower the computation overhead of decryption. In scheme [7], users have to pair their attribute secret keys with the ciphertexts in the phase of decryption. Based on ECC algorithm, we not only replace bilinear pairing with scalar multiplication on elliptic curves, but also reduce the calculation times, thereby saving the time overhead. Besides, the attribute authority helps data user doing part of the decryption, leaving only a small amount of computation overhead to data user.

In a word, our PF-CP-ABE effectively improves the efficiency, and is more suitable for practical application in IoT systems.

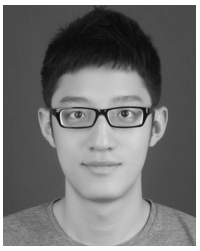
VII. CONCLUSION

In this paper, we proposed a novel efficient CP-ABE access control scheme for data sharing in IoT systems, called PF-CP-ABE. We replaced complicated bilinear pairing with simple scalar multiplication on elliptic curves, which results in significantly reducing the overall overhead for users. We also designed a new way of key distribution, so that the system can directly revoke a user or an attribute without updating other users' keys. Our scheme adopted expressive LSSS access structure to meet various access control demands in practical application. The analysis proved our scheme's security and the experiments demonstrated its efficiency.

REFERENCES

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2005, pp. 457–473.
[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, May 2007, pp. 321–334.
[3] A. Joux, "A one round protocol for tripartite Diffie-Hellman," *J. Cryptol.*, vol. 17, no. 4, pp. 263–276, 2004.
[4] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *J. Cryptol.*, vol. 17, no. 4, pp. 297–319, 2004.
[5] B. Dan and M. K. Franklin, "Identity-based encryption from the weil pairing," in *Proc. Int. Cryptol. Conf. Adv. Cryptol.*, 2001, pp. 213–229.
[6] C. Ling and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2007, pp. 456–465.
[7] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. Adv. Cryptol.-EUROCRYPT*, Tallinn, Estonia, 2011, pp. 568–588.
[8] M. Horváth, "Attribute-based encryption optimized for cloud computing," *Infocommun. J.*, vol. 7, no. 2, pp. 1–9, 2015.
[9] J. Hur, "Improving security and efficiency in attribute-based data sharing," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2271–2282, Oct. 2013.
[10] K. Liang, W. Susilo, and J. K. Liu, "Privacy-preserving ciphertext multi-sharing control for big data storage," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 8, pp. 1578–1589, Aug. 2015.
[11] K. Liang and W. Susilo, "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1981–1992, Sep. 2015.
[12] K. Liang, J. K. Liu, D. S. Wong, and W. Susilo, "An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing," in *Proc. Eur. Symp. Res. Comput. Secur.*, Springer, 2014, pp. 257–272.
[13] Y. Yang, H. Zhu, H. Lu, J. Weng, Y. Zhang, and K.-K. R. Choo, "Cloud based data sharing with fine-grained proxy re-encryption," *Pervasive Mobile Comput.*, vol. 28, pp. 122–134, Jun. 2016.

[14] S. Wang, K. Liang, J. K. Liu, J. Chen, J. Yu, and W. Xie, "Attribute-based data sharing scheme revisited in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 8, pp. 1661–1673, Aug. 2016.
[15] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan, "CP-ABE with constant-size keys for lightweight devices," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 5, pp. 763–771, May 2014.
[16] J. L. Beuchat, S. Mitsunari, E. Okamoto, and T. Teruya, "High-speed software implementation of the optimal ate pairing over Barreto-Naehrig curves," in *Proc. Int. Conf. Pairing-Based Cryptogr.*, 2010, pp. 21–39.
[17] P. S. Barreto, S. D. Galbraith, C. ÓhÉigeartaigh, and M. Scott, "Efficient pairing computation on supersingular Abelian varieties," *Des., Codes Cryptogr.*, vol. 42, no. 3, pp. 239–271, 2007.
[18] S. Canard, J. Devigne, and O. Sanders, "Delegating a pairing can be both secure and efficient," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, Springer, 2014, pp. 549–565.
[19] A. Guillevic and D. Vergnaud, "Algorithms for outsourcing pairing computation," in *Proc. Int. Conf. Smart Card Res. Adv. Appl.*, 2014, pp. 193–211.
[20] S. Canard, N. Desmoulins, J. Devigne, and J. Traoré, "On the implementation of a pairing-based cryptographic protocol in a constrained device," in *Proc. Int. Conf. Pairing-Based Cryptogr.*, 2012, pp. 210–217.
[21] D. Freeman, M. Scott, and E. Teske, "A taxonomy of pairing-friendly elliptic curves," *J. Cryptol.*, vol. 23, no. 2, pp. 224–280, 2010.
[22] M. Scott, "On the efficient implementation of pairing-based protocols," in *Proc. IMA Int. Conf. Cryptogr. Coding*, 2011, pp. 296–308.
[23] M. Rivain, "Fast and regular algorithms for scalar multiplication over elliptic curves," *IACR Cryptol. Eprint Arch.*, no. 2011, 2011.
[24] B. Chevallier-Mames, J. S. Coron, N. Mccullagh, D. Naccache, and M. Scott, "Secure delegation of elliptic-curve pairing," in *Proc. Int. Conf. Smart Card Res. Adv. Appl.*, 2010, pp. 24–35.
[25] X. Chen *et al.*, "Efficient algorithms for secure outsourcing of bilinear pairings," *Theor. Comput. Sci.*, vol. 562, pp. 112–121, Jan. 2015.
[26] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. Usenix Conf. Secur.*, 2011, p. 34.
[27] J. Li, X. Chen, J. Li, C. Jia, J. Ma, and W. Lou, "Fine-grained access control system based on outsourced attribute-based encryption," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2013, pp. 592–609.
[28] V. Odelu and A. K. Das, *Design of a New CP-ABE with Constant-Size Secret Keys for Lightweight Devices Using Elliptic Curve Cryptography*. Hoboken, NJ, USA: Wiley, 2016.
[29] V. Odelu, A. K. Das, M. K. Khan, K.-K. R. Choo, and M. Jo, "Expressive CP-ABE scheme for mobile devices in IoT satisfying constant-size keys and ciphertexts," *IEEE Access*, vol. 5, pp. 3273–3283, Feb. 2017.
[30] A. Beimel, "Secure schemes for secret sharing and key distribution," Fac. Comput. Sci., Technion-Israel Inst. Technol., Haifa, Israel, 1996.



SHENG DING received the B.Eng. degree in information security from Xidian University, China, in 2012, where he is currently pursuing the Ph.D. degree with the School of Cyber Engineering. His current research interests include cryptography, data security, and access control.

CHEN LI received the Ph.D. degree in cryptography from Xidian University, China, in 2015. He is currently a Post-Doctoral Fellow with the School of Telecommunications Engineering, Xidian University. His current research interest is cryptography.

HUI LI received the B.S. degree from Fudan University in 1990 and the M.S. and Ph.D. degrees from Xidian University in 1993 and 1998, respectively. In 2009, he was with the Department of Electrical and Computer Engineering, University of Waterloo, as a Visiting Scholar. He is currently a Professor with the School of Cyber Engineering, Xidian University. His research interests include the areas of cryptography, security of cloud computing, wireless network security, and information theory. He served as the TPC Co-Chair for ISPEC 2009 and IAS 2009, the General Co-Chair for E-Forensic 2010, ProvSec 2011, and ISC 2011, and the Honorary Chair of NSS 2014 and ASIACCS 2016.