IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 8, 2020, accepted February 19, 2020, date of publication February 27, 2020, date of current version March 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2976746

Efficient and Secure Ciphertext-Policy Attribute-Based Encryption Without Pairing for Cloud-Assisted Smart Grid

YONG WANG[1], BIWEN CHEN[2], LEI LI[1], QIANG MA[1], HUICONG LI[1], AND DEBIAO HE[3]

[1]State Grid Shandong Electric Power Company, Jinan 250001, China
[2]School of Computer Science, Wuhan University, Wuhan 430072, China
[3]School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China

Corresponding author: Biwen Chen (macrochen@whu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772377, Grant 61972294, and Grant 61932016, in part by the Natural Science Foundation of Hubei Province of China under Grant 2017CFA007, and in part by the Science and Technology Planning Project of Shenzhen under Grant JCYJ20170818112550194.

ABSTRACT Cloud-assisted smart grid has been broadly deployed to improve the economics, efficiency, sustainability, and reliability of electricity services. The new revolution of technology will also bring new challenges to data security, particularly data confidentiality. As a promising encryption method for supporting fine-grained access control, attribute-based encryption (ABE) is widely given attention by the academia and industry. However, most existing schemes suffer from the efficiency problem limiting its deployment for the cloud-assisted smart grid, especially for resource-limited users. To address this problem, this paper proposes two efficient pairing-free ciphertext-policy attribute-based schemes that eliminate the computation-intensive bilinear pairing operation. The basic scheme only uses some simple scalar multiplications based on elliptic curves, and the enhanced scheme provides an important usability feature, namely the users and attributes revocation. The security of the proposed schemes is reduced to the elliptic curve decisional Diffie-Hellman problem. Finally, the performance analysis and extensive experiments demonstrate that our proposed schemes are suitable for the cloud-assisted smart grid.

INDEX TERMS Cloud-assisted smart grid, access control, ciphertext-policy attribute-based encryption, pairing-free.

I. INTRODUCTION

Smart grid as the next-generation power supply network has been broadly adopted to address the problems of the traditional power grid. It brings significant improvement in the economics, efficiency, sustainability, and reliability of electricity services. The classical framework of a smart grid is shown as FIGURE 1. The new architecture of the smart grid motivates the use of various technologies for addressing different challenges. With the help of cloud computing, the smart grid can not only effectively solve the issue of large data management but also deliver electricity more efficiently and reliably. The amount of data generated by various applications is collected and uploaded to the cloud center to be stored and processed efficiently, then be flexibly accessed by different data users.

Although a cloud-assisted smart grid can permit lower kilowatt costs, more efficient transmission of electricity and reduced energy use, its problems of data confidentiality cannot be ignored, because privacy protection is a fundamental requirement of the whole framework. A malicious cloud server may infer private information of users' daily lives, such as identity and rest schedule,from uploaded unencrypted data, by leveraging data mining technologies [1]. To address the above concerns, some works have been designed based on different cryptographic primitives, such as [2]–[5]. However, there are still several problems in the cloud-assisted smart grid environments. First, the efficiency of schemes should be improved to ensure acceptable response time. Second, the encrypted data in turn hinders data analysis and access control. Hence, how to effectively ensure the confidentiality

The associate editor coordinating the review of this manuscript and approving it for publication was Aniello Castiglione.

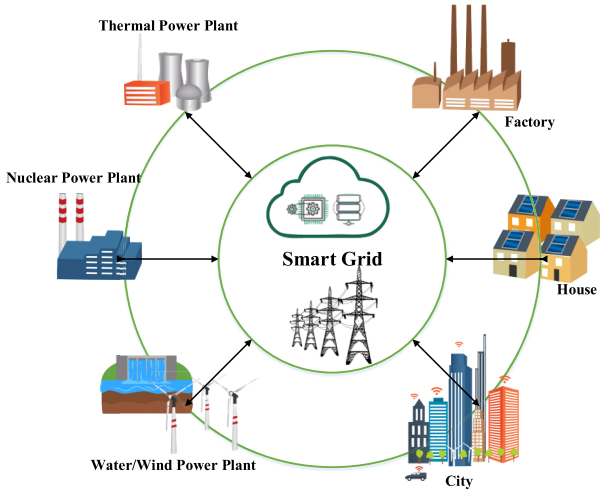


FIGURE 1. System model of our design.

of data and maintain fine-grained access control has been a significant challenge in the cloud-assisted smart grid.

Sahai and Waters [6] addressed the above problem by introducing a new cryptographic primitive, namely attribute-based encryption (ABE). The data owner in the ABE system can specify a concrete control policy for accessing data, while every data user can only decrypt ciphertexts that his attributes' secret key satisfies the corresponding policy. ABE has become one of the most promising techniques to provide data confidentiality and flexible access control in loosely coupled environments. Unfortunately, most existing schemes can not be widely deployed and applied due to its efficiency problem. One reason why most previous ABE schemes are inefficiency is the number of computation-intensive operations (e.g., bilinear pairing) grows with the complexity of the access policy. Thus, one way to improve efficiency is to eliminate the computation-intensive operations. Compared with scalar multiplication, the bilinear pairing operation in pairing-based ABE schemes has been regarded as the most intensive operations. Generally, the computation cost of scalar multiplication is about one-third of that of the bilinear pairing. Hence, it is a challenging work for constructing the pairing-free ABE schemes without sacrificing its security.

A. MOTIVATIONS AND CONTRIBUTIONS

Currently, a few works [7]–[9] have been focused on designing lightweight attribute-based encryption schemes. However, they still have some drawbacks as follows: 1) These solutions have limited application scenarios. Most existing lightweight schemes [7], [10] are key-policy attribute-based encryption (KP-ABE), which is only suitable for users with fixed access rights. 2) A few lightweight schemes have weak security. The known pairing-free ciphertext-policy attribute-based encryption (CP-ABE) scheme [8] can not resist the illegal key sharing attack. 3) The efficiency of schemes can be improved. Although online/offline schemes can move some computation tasks to the offline phase by adding public parameters, computation-intensive operations still affect the efficiency of them.

To address the above problems, we propose two efficient and secure ciphertext-policy attribute-based encryption schemes without pairing for cloud-assisted smart grid. Specifically, the basic scheme is a pairing-free ciphertext-policy attribute-based encryption which is complementary to KP-ABE in the application scenarios, and the enhanced scheme supports the users and attributes revocation without increasing computational overhead. The key contributions are made as follows:

- We propose two lightweight ciphertext-policy attribute-based encryption without pairing for the cloud-assisted smart grid. In the proposed schemes, the complexity bilinear pairing operation is removed to adapt the entities with resource-limited.
- The enhanced scheme supports users and attributes revocation through a simple transformation based on the basic scheme. Also, based on the Decisional Diffie-Hellman problem, the security of our designs is proved to be semantic security under the chosen-plaintext attack.
- We do extensive experimental evaluations and implement a prototype of the basic scheme. The experimental results show the practicability of the proposed schemes.

B. ORGANIZATION

The rest of this paper is arranged as follows. Section II introduces the related work in the attribute-based encryption area. The preliminaries are described in Section III, and the system overview is presented in Section IV. Section V constructs two lightweight ciphertext-policy attribute-based encryption schemes without pairing. Section VI and VII provide the security proof and performance evaluation, respectively. A conclusion is drawn in Section VIII.

II. RELATED WORK

Since Sahai and Waters [6] introduced the definition of fuzzy identity-based encryption (IBE), ABE as an extension of IBE has been one of the hot research directions in the information security area due to its fine-grained access control [7], [8], [10]–[13]. Currently, existing ABE schemes consist roughly of key-policy attribute-based encryption (KP-ABE) and ciphertext-policy attribute-based encryption (CP-ABE). In the former, the data user's secret key is associated with the access policy, while in the latter the ciphertext is associated with the access policy.

Extensive ABE schemes with new features [13]–[16] have been designed to meet different requirements of applications. To reduce the burden of users, Li *et al.* [17] constructed an outsourced ABE scheme that can shift the computational burden of users to outsourcing cloud servers. Li *et al.* [13] constructed a hierarchical attribute-based encryption (HABE), where the key manager does not have to do the number of efforts to key management even though there are plenty of attributes. To support the multi-domains

environment, multi-authority ABE schemes [14], [16] were designed to support the multiple authorized entities to distribute decryption keys. To protect user's privacy, scholars proposed anonymous ABE schemes [18], [19] and policy-hidden ABE schemes [20], [21] respectively. The former prevents user identities from being leaked, while the latter hides access control policies. To achieve the traceability of the secret key, traceable ABE schemes were proposed in the works [22], [23]. To adapt the dynamic of users and attributes, some flexible ABE schemes [11], [24], [25] were designed to revoke users and attributes.

Although diverse ABE schemes can meet the functional requirements of practical applications, the efficiency problem is still a huge challenge, especially for users with limited resources. Currently, a number of mechanisms have been used to improve the practicability of ABE schemes, such as online/offline technology [9], [26], [27] and outsourcing technology [17], [28], [29]. However, their schemes have a common insufficient that the use of bilinear pairing affects its efficiency. To eliminate the negative impact, a few lightweight pairing-free ABE schemes [7], [8], [30] were proposed in recent years. However, these schemes can not cover a variety of scenarios because of the drawbacks described in the previous section.

III. PRELIMINARIES

A. NOTATIONS

Table 1 gives the summary of notations in our proposed scheme.

B. ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic Curve (EC) has been widely used in the cryptographic protocol since its short key and high security. The EC is defined by the following cubic equation:

$$y^2 = x^3 + ax + b, \quad and \quad 4a^3 + 27b^2 \neq 0(mod \quad p)$$

where p is a large prime and a, b are two elements of finite field F_p. All the points in EC form a group $\mathbb{G}$ that exists a generator P with the order q. Fundamental operations in the group include the point scalar multiplication ($Q = kP$, where $k \in \mathbb{Z}_q^*$) and the point double ($Q = 2P$).

The elliptic curve cryptography generally can be constructed based on some hard mathematical problems under EC, such as EC discrete logarithms problem (DL), EC computational Diffie-Hellman problem (CDH), EC Decisional Diffie-Hellman problem (DDH) and other variants.

- **DL**: Given two points $P, Q \in \mathbb{G}$, it is hard to find k such that $Q = kP$.
- **CDH**: Given three points $(P, aP, bP) \in \mathbb{G}$, it is hard to find a point Q such that $Q = abP$.
- **DDH**: Given four points $(P, aP, bP, Z) \in \mathbb{G}$, it is hard to decide whether or not Z is equal to abP.

Let $BG(\lambda)$ denote an efficient elliptic curve parameter generator that inputs a parameter λ and outputs $(q, a, b, \mathbb{G}, P)$.

TABLE 1. Summary of notations.

Notation	Description
λ	security parameter
$(q, a, b, \mathbb{G}, P)$	elliptic curve parameter
(α, β, γ)	master secret key
(P_1, P_2, P_3)	master public key
A_{att_i}, a_{att_i}	public/secret key pair of attribute att_i
(pp, sp)	public/secret parameter
$\mathcal{U}$	the whole system attribute space
U_j	j-th data user
UA_j	the attribute set of user U_j
att_i^j	i-th attribute of user U_j
(sk_j^1, sk_j^2)	decryption key components of user U_j
$(r^1_{att_i^j}, r^2_{att_i^j})$	two keys of attribute att_i^j of user U_j
τ	access control tree with root node $root$
$root$	root node of tree τ
τ_x	access control tree with root node x
f_x	a polynomial of node x
$(a_0^x, ..., a_{k_x-1}^x)$	coefficients of polynomial f_x with degree $k_x - 1$
$f_x(0)$	constant term of polynomial f_x of node x
y	a leaf node associated with an attribute
CT_τ	ciphertext under access control tree τ
$\|S\|$	size of set S
T_v	v-th time period

C. ACCESS CONTROL STRUCTURE

The access tree is an expression for the access policy and the definitions are as follows:

Access Tree: In an access tree τ, let n_x be the number of children of node x and k_x be the threshold value of node x, where $k_x \in (0, n_x]$. For each leaf node y, it is associated with an attribute and the threshold value $k_y = 1$, while each non-leaf node x is described by its children. Let $parent(x)$ denote the parent node of node x, $children(x)$ denote all child nodes of node x and $att(y)$ denote the attribute associated with the leaf note y. We assume that every node x is ordered in the access tree and let $index(x)$ denote the unique index of node x. Let τ_x be a tree of τ rooted at the node x.

We say that the attribute set A satisfies the tree τ_x if and only if $\tau_x(A) = 1$. The process to compute $\tau_x(A)$ is defined as follows.

- For a non-leaf node x, $\tau_x(A) = 1$ if and only if the return values $\tau_{children(x)}(A)$ of at least k_x child nodes output 1.
- For a leaf node y, the $\tau_y(A) = 1$ if and only if the attribute corresponding to the node y is one of the members of A, that is, $att(y) \in A$.

(n, t) *Shamir Secret Sharing (SSS):* It is used to construct the flexible access tree. In a (n, t) SSS scheme, a secret s will be divided into n shares such that any subset of m shares $(t \leq m \leq n)$ can together reconstruct the secret.

- If a dealer D wants to share a secret $s > 0$ among any m of n users. Firstly, D chooses a prime $q > max(s, n)$ and sets $a_0 = s$. Then, D constructs a polynomial

$f(X) = \sum_{i=0}^{t-1} a_i X^i$ of degree $t-1$, where a_i is chosen at random from $\mathbb{Z}_q^*$. Finally, for each $x_i \in \{x_i | i = 1, 2, \ldots, n \wedge x_i \in \mathbb{Z}_q\}$, D computes $y_i = f(x_i)\ mod\ q$ for the user U_i, where x_i denotes the public index of the user U_i.

- m users can reconstruct s by providing their distinct shares (x_i, y_i). Without loss of generality, we assume that users $\{x_1, x_2, \ldots, x_m\}$ are trying to reconstruct the secret s. They compute the coefficient $a_0 = s$ of $f(X)$ by leveraging Lagrange interpolation, $f(0) = \sum_{i=1}^{m} \triangle_{i,S}\ y_i$, where $S = \{1, 2, \ldots, m\}$ and $\triangle_{i,S} = \Pi_{j \in S, j \neq i} \frac{-x_j}{x_i - x_j}$.

IV. SYSTEM OVERVIEW

A. SYSTEM MODEL

The system model of our design is shown in FIGURE 2 and the main entities include:

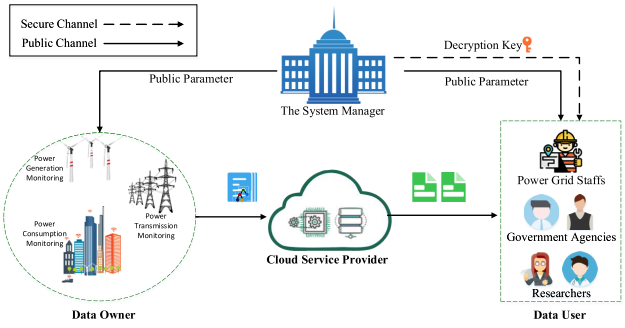


FIGURE 2. System model of our design.

System Manager: There exists a completely trusted entity (such as the government), which is responsible to initialize the whole system and authenticate the data users.

Cloud Service Provider: It possesses almost infinite computation resources and storage capability, for instance, Amazon AWS. It is responsible for storing the data and responding to user queries. Meanwhile, it also is considered as an honest-but-curious entity, who is honest to execute the predefined protocol, but curious to the information of storage data.

Data Sender: It is responsible for generating and collecting data, and the large amounts of data need to be stored and shared. For instance, it may be power consumption monitoring. The large vital data is collected and uploads them to the data storage center. To achieve privacy protection, sensitive information has to be encrypted before uploading.

Data User: It tries to access data stored in the data storage center. There are various types of data users in our system, such as third-party organizations, government agencies or power grid staffs. Each user possesses a set of attributes to describe the features of roles and decrypts the authorized ciphertexts using his own decryption keys.

B. FORMAL DEFINITION

Our designed basic CP-ABE scheme consists of the following algorithms:

- **Setup**$(\lambda) \rightarrow (pp, sp)$: It takes as input a security parameter λ, and then outputs the public parameter pp and the secret parameter sp. The public parameter is published, while the secret parameter is secretly kept by the system manager (or the authority center).
- **KeyGen**$(pp, sp, UA_j) \rightarrow (sk_j)$: It takes as input the public parameter pp, the secret parameter sp and a set of attributes of user U_j, and outputs a decryption key sk_j associated with UA_j.
- **Encryption**$(pp, M, \tau) \rightarrow (CT_\tau)$: It takes as input the public parameter pp, a message M and an access tree τ, and outputs the ciphertext CT_τ that can be only decrypted by users whose attributes satisfy the access tree.
- **Decryption**$(pp, CT_\tau, sk_j) \rightarrow (M$ or $\perp)$: It takes as input the public parameter pp, a ciphertext CT_τ associated with the tree τ, a decryption key sk_j, and outputs a message M or an error $\perp$.

C. SECURITY MODEL

We assume that the system manager is a trust entity and correctly execute all predefined algorithms and some users may be corrupted by adversaries. Semantic security under chosen-plaintext attack of CP-ABE schemes can be modeled by a game which is played between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. In this game, $\mathcal{C}$ runs the proposed scheme and responses queries from $\mathcal{A}$, and the adversary $\mathcal{A}$ tries to distinguish which message the challenge ciphertext corresponds to based on the information obtained from queries. The game is defined as follows:

- **Initialization**: $\mathcal{A}$ sends an access tree τ^* to $\mathcal{C}$ to be challenge.
- **Setup**: $\mathcal{C}$ executes the **Setup** algorithm to generate the system parameter (pp, sp) and returns the public parameter pp to $\mathcal{A}$.
- **Phase 1**: $\mathcal{A}$ can adaptively make secret key queries for any attribute set with the restriction that the attribute sets of queries do not satisfy the access tree τ^*. The challenger $\mathcal{C}$ executes the **KeyGen** algorithm to generate the decryption keys and sends them to $\mathcal{A}$.
- **Challenge**: $\mathcal{A}$ sends two equal length challenge messages M_0, M_1 to $\mathcal{C}$. After receiving them, $\mathcal{C}$ flips a random coin $b \in \{0, 1\}$ and returns a challenge ciphertext CT_{τ^*} generated by encrypting the message M_b.
- **Phase 2**: $\mathcal{A}$ can continue to submit the decryption queries like **Phase** 1.
- **Guess**: Finally, $\mathcal{A}$ outputs a guess b' of b.

$\mathcal{A}$ will win the above game if (s)he guesses the correct result $(b' = b)$. Thus, the advantage of $\mathcal{A}$ winning the above game is defined as:

$$Adv_{\mathcal{A}}(\lambda) = Pr[b' = b] - \frac{1}{2}$$

where $Pr[b' = b]$ is the probability of $b' = b$.

Definition 1: Our proposed CP-ABE scheme is semantic security under chosen-plaintext attack if no polynomial-time

adversary can win the above game with a non-negligible advantage.

V. CONSTRUCTION

We construct a basic pairing-free CP-ABE scheme and an enhanced scheme with attribute revocation.

A. BASIC SCHEME

Setup(λ): The algorithm is used to initialize the whole system. Given a security parameter λ, the system manager performs the following steps:

1 generates the elliptic curve parameters $(q, a, b, \mathbb{G}, P)$ by running the generator $BG(\lambda)$.
2 chooses a secure hash function $H : \mathbb{G} \to \{0, 1\}^{\lambda}$.
3 chooses three random numbers $(\alpha, \beta, \gamma) \in \mathbb{Z}_q^*$ as master secret key $MS = \{\alpha, \beta, \gamma\}$, and then computes the master public key $MP = \{P_1 = \alpha P, P_2 = \beta P, P_3 = \gamma P\}$.
4 for each attribute $att_i \in \mathcal{U}$, chooses a number $a_{att_i} \in \mathbb{Z}_q^*$ and computes $A_{att_i} = a_{att_i} P$, where $\mathcal{U}$ denotes the attribute space in our scheme.
5 publishes the public parameters $pp = \{q, a, b, \mathbb{G}, P, P_1, P_2, P_3, A_{att_1}, \ldots, A_{att_{|\mathcal{U}|}}\}$ and secretly keeps the privacy parameter $sp = \{\alpha, \beta, \gamma, a_{att_1}, \ldots, a_{att_{|\mathcal{U}|}}\}$.

KeyGen(pp, sp, UA_j): The algorithm is used to generate the legal decryption keys for authenticated users. Given an attribute set $UA_j = \{att_1^j, \ldots, att_m^j\} \subset \mathcal{U}$ of the registered user U_j, where m denotes the number of attributes of user U_j, the system manager performs th following steps:

1 chooses a random number $u_j \in \mathbb{Z}_q^*$ and computes $sk_j^1 = \alpha - u_j$.
2 for each attribute $att_i^j \in UA_j$, chooses two random numbers $r^1_{att_i^j}, r^2_{att_i^j}$ such that $a_{att_i^j} + u_j = \gamma r^1_{att_i^j} + \beta r^2_{att_i^j}$, where $i = 1, \ldots, m$.
3 sends the user decryption keys $sk_j = \{sk_j^1, sk_j^2\}$ to user j via a secure channel, where $sk_j^2 = \{(r^1_{att_i^j}, r^2_{att_i^j}) | i = 1, \ldots, m\}$.

Encryption(pp, M, τ): The algorithm is used to encrypt the message under an access control structure, which is executed by the data owner. Given the public parameter pp, a message M and an access tree τ, the data owner performs the following steps:

1 computes $C_1 = rP$, where $r \in \mathbb{Z}_q^*$.
2 computes $C_2 = H(rP_1) \oplus M = H(r\alpha P) \oplus M$.
3 in a top-down manner, for each node $x \in \tau$, chooses a polynomial f_x of degree $k_x - 1$ with the following way:
  3.1 for the root node $root$, set $a_x^{root} = r$. Then, it chooses $a_1^{root}, \ldots, a_{k_{root}-1}^{root}$ randomly to define $f_{root}(X) = \sum_{i=0}^{k_{root}-1} a_i^{root} X^i$ completely.
  3.2 for other node x, sets $a_0^x = f_{parent(x)}(index(x))$ and randomly chooses $a_1^x, \ldots, a_{k_x-1}^x$ to completely define $f_x(X) = \sum_{i=0}^{k_x-1} a_i^{k_x-1} X^i$.
4 for every leaf node $y \in Y$, computes $C_y^1 = f_y(0)P_2, C_y^2 = f_y(0)P_3, C_y^3 = f_y(0)A_{att(y)}$, where $Y \subset \mathcal{U}$ denotes the leaf node set in tree τ..

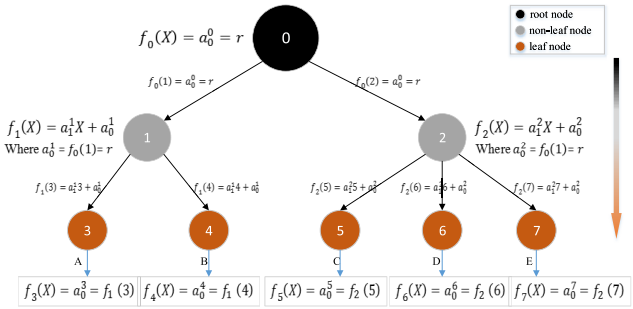


FIGURE 3. An example of assigning secret shares to the access tree.

5 outputs the ciphertext $CT_\tau = (\tau, C_1, C_2, \forall y \in Y : C_y^1, C_y^2, C_y^3)$.

FIGURE 3 is a simple example of constructing the access tree τ, which represents an access policy: (A $\wedge$ B) $\vee$ (at least two attributes among C, D, E). The access tree consists of eight nodes with indexes of $(0, 1, \ldots, 7)$ and the threshold values for all nodes are $k_0 = 1, k_1 = 2, k_2 = 2, k_{3,4,5,6,7} = 1$, respectively. Starting with the root node 0, the data owner constructs polynomials for each node. In this process, a_1^1, a_1^2 are chosen randomly from the finite field $\mathbb{Z}_q^*$.

Decryption(pp, CT_τ, sk_j): The data user runs the algorithm to obtain the shared message. Given the public parameter pp, a ciphertext $CT_\tau = (\tau, C_1, C_2, \{C_y^1, C_y^2, C_y^3 | \forall y \in Y\})$ and a secret key $sk_j = \{sk_j^1, sk_j^2\}$ associated with the attribute set $UA_j = \{att_1^j, \ldots, att_m^j\}$, the user U_j performs as follows:

1 if UA_j does not satisfy τ, return $\perp$.
2 otherwise, we use a recursive algorithm $Decrypt\_Node(CT_\tau, sk_j, \tau_x)$, which takes as input a ciphertext CT_τ, a secret key $sk_j = \{sk_j^1, sk_j^2\}$ corresponding with the attribute set UA_j and an access tree τ_x with root node x. The recursive algorithm is defined as follow:
  2.1 for each leaf node y, if $att(y) = att_i^j \in UA_j$, then:

$$\begin{aligned} &Decrypt\_Node(CT_\tau, sk_j, \tau_y) \\ &= r^1_{att_i^j} C^2_{att_i^j} + r^2_{att_i^j} C^1_{att_i^j} - C^3_{att_i^j} \\ &= r^1_{att_i^j} f_y(0)\gamma P + r^2_{att_i^j} f_y(0)\beta P - C^3_{att_i^j} \\ &= (r^1_{att_i^j}\gamma + r^2_{att_i^j}\beta) f_y(0)P - C^3_{att_i^j} \\ &= (a_{att_i^j} + u_j) f_y(0)P - f_y(0) a_{att_i^j} P \\ &= u_j f_y(0)P \end{aligned}$$

  If $att_i^j \notin UA_j$, $Decrypt\_Node(CT_\tau, sk_j, \tau_y) = \perp$.
  2.2 for each non-leaf node x, the recursive algorithm $Decrypt\_Node(CT_\tau, sk_j, \tau_x)$ proceeds as follows: each child node $z \in children(x)$ of x recursively calls the algorithm $Decrypt\_Node(CT_\tau, sk_j, \tau_z)$ and obtains an output as F_z. Then, we assume that $L_x \in children(x)$ denotes a child note index set of node x of size k_x such that $F_x \neq \perp$. If there is no such set, and then the algorithm $Decrypt\_Node(CT_\tau, sk_j, \tau_x) = \perp$.

Otherwise, computes the value F_x as follow:

$$\begin{aligned} F_x &= \sum_{z\in L_x} \triangle_{index(z),L_x} F_z \\ &= \sum_{z\in L_x} \triangle_{index(z),L_x} u_j f_z(0)P \\ &= u_j \sum_{z\in L_x} \triangle_{index(z),L_x} f_z(0)P \\ &= u_j \sum_{z\in L_x} \triangle_{index(z),L_x} f_{parent(z)}(index(z))P \\ &= u_j \sum_{z\in L_x} \triangle_{index(z),L_x} f_x(index(z))P \\ &= u_j f_x(0)P \end{aligned}$$

where $\triangle_{index(z),L_x} = \Pi_{i\in L_x, i\neq z} \frac{-index(i)}{index(z)-index(i)}$.

3 if the attribute set UA_j of user U_j satisfies the access tree τ, the algorithm $Decrypt\_Node(CT_\tau, sk_j, \tau)$ takes as input the ciphertext CT_τ, a secret key sk_j and an access tree τ, and outputs $Decrypt\_Node(CT_\tau, sk_j, \tau) = u_j rP$. Then, computes R and M, respectively:

$$\begin{aligned} R &= sk_j^1 C_1 + Decrypt\_Node(CT_\tau, sk_j, \tau) \\ &= (\alpha - u_j)rP + u_j rP \\ &= r\alpha P \\ &= rP_1 \\ M &= C_2 \oplus H(R) \\ &= H(rP_1) \oplus M \oplus H(rP_1) \end{aligned}$$

B. ENHANCE SCHEME

In reality, the attributes of data users may change over time since the change in location, work environment, and thus the property of the users and attributes revocation should be provided to facilitate the wide application of the ABE scheme. To achieve this goal, we propose an enhanced scheme based on the above basic construction.

The enhanced scheme achieves the users and attributes revocation by leveraging timed rekeying mechanism. The detailed description is as follows:

- **Setup**: The system manager generates the parameters (pp, sp) as in the basic scheme, and then chooses an element $t_v \in \mathbb{Z}_q^*$ for each time period T_i in the system lifetime. Finally, it publishes the public parameter $pp' = pp \cup \{T_v = t_v P | v = 1, \ldots\}$ and secretly keeps the secret parameter $sp' = sp \cup \{t_v | v = 1, \ldots\}$.
- **KeyGen**: The system manager generates the decryption keys for user j with the attribute set $UA_j = \{att_1^j, \ldots, att_m^j\}$ at the initial time period T_1. It first computes $sk_{j,t_1}^1 = \alpha - u_j - t_1$, and then for each attribute att_i^j in UA_j, computes a pair $(r^1_{att_i^j,t_1}, r^2_{att_i^j,t_1})$ such that $a_{att_i^j} + u_j + t_1 = \gamma r^1_{att_i^j,t_1} + \beta r^2_{att_i^j,t_1}$. Finally, it sends the secret keys $sk_{j,t_1} = \{sk_{j,t_1}^1, sk_{j,t_1}^2\}$ to user j, where $sk_{j,t_1}^2 = \{(r^1_{att_i^j,t_1}, r^2_{att_i^j,t_1}) | i = 1, \ldots, m\}$.
- **Key Updating**: When the time period evolves from T_v to T_{v+1}, the system manager computes the update keys $\{(\Theta^1, \Theta^2) | \Theta^1 = t_v + t_{v+1}, \Theta^2 = \{\Delta_1, \Delta_2\}\}$, where $\{\Delta_1, \Delta_2\}$ satisfy $t_{v+1} - t_v = \gamma\Delta_1 + \beta\Delta_2$. Once receiving the update keys (Θ^1, Θ^2), the user U_j computes $sk_{j,t_{v+1}} = sk_{j,t_v} + \Theta^1$, $sk_{j,t_v}^2 = \{(r^1_{att_i^j,t_v} + \Delta_1, r^2_{att_i^j,t_v} + \Delta_2) | v = 1, \ldots, m\}$ as the decryption key at time period T_{v+1}.
- **Encryption/Decryption**: The encryption/decryption algorithms are the same as the basic scheme.

Correctness: If we set u_j in the basic scheme is equal to the value $u_j + t_v$ in the enhanced scheme, and then the enhanced scheme is consistent with the basic scheme. In the time period T_v, the process of computing R is as follow:

$$\begin{aligned} R &= sk_j^1 C_1 + Decrypt\_Node(CT_\tau, sk_j, \tau) \\ &= (\alpha - u_j - t_v)rP + (u_j + t_v)rP \\ &= r\alpha P \\ &= rP_1 \end{aligned}$$

VI. SECURITY PROOF

We prove that our basic CP-ABE scheme is semantic security under chosen-plaintext attack by leveraging the method of reduction. Since the security of our scheme relies on the DDH problem, that is, if a probabilistic-polynomial time (PPT) adversary $\mathcal{A}$ can break the basic scheme with a non-negligible advantage, then the adversary's ability can be used to solve the DDH problem.

Theorem 1: Suppose the DDH problem is hard, then no PPT adversary can break the basic scheme associated with a challenge access tree τ^*.

Proof: Suppose the adversary $\mathcal{A}$ can win the game of section III with a non-negligible advantage ε, and then a simulator $\mathcal{S}$ can be build to solve the DDH problem with advantage $\frac{\varepsilon}{2}$. The procedure for constructing $\mathcal{S}$ is described as follows:

Before the game starts, the challenger $\mathcal{C}$ first selects a group $\mathbb{G}$ with generator P. Then, $\mathcal{C}$ flips a random coin $\mu \in \{0, 1\}$ and randomly chooses two numbers $a, b \in \mathbb{Z}_q^*$. Finally, $\mathcal{C}$ sets $Z_\mu = abP \in \mathbb{G}$ if $\mu = 0$ and randomly chooses an element from the group $\mathbb{G}$ as Z_μ otherwise. The challenger $\mathcal{C}$ sends the DDH problem instance (P, aP, bP, Z_μ) to $\mathcal{S}$. The simulator $\mathcal{S}$ will act as the challenger of $\mathcal{A}$ in the game of section III as follow:

Initialization: $\mathcal{A}$ sends a challenge access tree τ^* to the simulator $\mathcal{S}$. Suppose the attributes of the access tree is U_{τ^*} and the leaf node set of the access tree is Y.

Setup: $\mathcal{S}$ chooses at random $\beta, \gamma \in \mathbb{Z}_q^*$ and sets $P_1 = aP, P_2 = \beta P, P_3 = \gamma P$, which implicitly sets $\alpha = a$. For every attribute $att_i \in \mathcal{U}$, $\mathcal{S}$ randomly chooses a number $t_{att_i} \in \mathbb{Z}_q^*$ and sets $A_{att_i} = t_{att_i} P$ if $att_i \in U_{\tau^*}$ or sets $A_{att_i} = t_{att_i}$P-aP $= t_{att_i} P - P_1$ if $att_i \notin U_{\tau^*}$, thus, implicitly it sets $a_{att_i} = t_{att_i}$ or $a_{att_i} = t_{att_i} - a$. The simulator $\mathcal{S}$ sends the public parameters to $\mathcal{A}$. Since t_{att_i} is chosen at random, the public

parameters generated by $\mathcal{S}$ are identically distributed in the same as those output by the **Setup** algorithm.

Phase 1: $\mathcal{A}$ adaptively makes decryption key queries for any attribute set $UA_j = \{att_i^j | i = 1, \ldots, m\}$ with the restriction that the set does not satisfy the challenge access tree τ^*, that is, $\tau^*(UA_j) \neq 1$. To response each query, the simulator $\mathcal{S}$ processes as follows:

1 chooses a random number $h_j \in \mathbb{Z}_q^*$ and sets $sk_j^1 = h_j$, which implicitly set $u_j = a - h_j$ since

$$\begin{aligned} sk_j^1 &= h_j \\ &= a - (a - h_j) \\ &= \alpha - u_j \end{aligned}$$

2 for each attribute $att_i^j \in UA_j$, the simulator $\mathcal{S}$ must construct decryption key components sk_j^2 form as $(r^1_{att_i^j}, r^2_{att_i^j})$ that satisfies $a_{att_i^j} + u_j = \gamma r^1_{att_i^j} + \beta r^2_{att_i^j}$. Since $\mathcal{S}$ implicitly sets $u_j = a - h_j$ and $a_{att_i^j} = t_{att_i} - a$ for each $att_i^j \notin U_{\tau^*}$, the valid decryption key component can be computed:

$$\begin{aligned} \gamma r^1_{att_i^j} + \beta r^2_{att_i^j} &= a_{att_i^j} + u_j \\ &= t_{att_i^j} - a + a - h_j \\ &= t_{att_i^j} - h_j \end{aligned}$$

where $\gamma, \beta, t_{att_i^j}, h_j$ are known.

3 The simulator $\mathcal{S}$ sends to $\mathcal{A}$: $sk_j = (sk_j^1, sk_j^2)$, where $sk_j^2 = \{(r^1_{att_i^j}, r^2_{att_i^j}) | i = 1, \ldots, |UA_j|\}$.

According to the above analysis, the decryption keys are valid in the view of $\mathcal{A}$.

Challenge: $\mathcal{A}$ sends two equal length challenge messages M_0, M_1 to $\mathcal{S}$. $\mathcal{S}$ flips a random coin $b \in \{0, 1\}$ and returns the challenge ciphertext CT_{τ^*} of M_b. $\mathcal{S}$ computes the ciphertext CT_{τ^*} as follows:

1 sets $C_1 = bP$ and computes $C_2 = H(Z_\mu) \oplus M_b = H(abP) \oplus M_b = H(bP_1) \oplus M_b$.

2 to correctly generate the ciphertexts $\{(C_y^1, C_y^2, C_y^3) | y \in Y\}$, the simulator performs the following:

2.1 sets $f_{root}(0) = a_0^{root} = b$, and select at random $\{a_1^{root}, \ldots, a_{k_{root}-1}^{root} \in \mathbb{Z}_q^*\}$. The polynomial of the root node is $f_{root}(X) = \sum_{i=1}^{k_{root}-1} a_i^{root} X^i + b$. Note that although the simulator cannot know the value of b, it does not affect the following process.

2.2 for each non-leaf node x, selects at random $\{a_1^x, \ldots, a_{k_x-1}^x\}$ and constructs the polynomial $f_x(X) = \sum_{i=1}^{k_x-1} a_i^x X^i + f_{parent(x)}(index(x))$, where $f_{parent(x)}(index(x))$ forms as $W_x + b$ and W_x can be computed efficiently.

2.3 for each non-leaf node y, the polynomial $f_y(X) = f_{parent(x)}(index(y)) = W_y + b$, W_y can be computed efficiently.

2.4 for each non-leaf node y, computes $C_y^1 = f_y(0)P_2$, $C_y^2 = f_y(0)P_3$, $C_y^3 = f_y(0)A_{att(y)}$. Although b is unknown, $C_y^1 = f_y(0)P_2 = (W_y + b)\beta P = W_y \beta P + \beta bP$. Thus, $\{C_y^1, C_y^2, C_y^3 | y \in Y\}$ can be effectively carried out based on bP.

3 sends the ciphertexts $CT_{\tau^*} = (\tau, C_1, C_2, \{(C_y^1, C_y^2, C_y^3) | y \in Y\})$ to $\mathcal{A}$.

Phase 2. The phase is the same as the **Phase** 1, that is, $\mathcal{A}$ can continue to issue the decryption key requests with the same restriction.

Guess. $\mathcal{A}$ sends a guess b' to simulator $\mathcal{S}$.

If $b' = b$, $\mathcal{S}$ outputs $\mu = 0$ to show that $Z_\mu = abP$. If $b' \neq b$, $\mathcal{S}$ outputs $\mu = 1$ to show that Z_μ is a random element from group $\mathbb{G}$.

According to the above game, if $\mu = 1$, Z_μ is a random element and the ciphertext C_2 is indistinguishable with a random element in the view of adversary $\mathcal{A}$. Thus, we have

$$Pr[b' = b | \mu = 1] = Pr[b' \neq b | \mu = 1] = \frac{1}{2}$$

If $\mu = 0$, $\mathcal{S}$ outputs a valid ciphertext, and thus $\mathcal{A}$ can output $b' = b$ with the probability $\frac{1}{2} + \epsilon$. Therefore, we have:

$$Pr[b' = b | \mu = 0] = 1/2 + \epsilon$$

Finally, the advantage of $\mathcal{S}$ is $Pr[b' = b] - \frac{1}{2}$, where

$$\begin{aligned} Pr[b' = b] &= \frac{1}{2} Pr[b' = b | \mu = 0] + \frac{1}{2} Pr[b' = b | \mu = 1] \\ &= \frac{1}{2}(1/2 + \epsilon) - \frac{1}{2}\frac{1}{2} \\ &= \frac{\epsilon}{2} \end{aligned}$$

Therefore, it turns out that if $\mathcal{A}$ can break our basic scheme with a non-negligible advantage ϵ, and there exists is a simulator $\mathcal{S}$ can solve the DDH problem with advantage $\frac{\epsilon}{2}$. However, it is a conflict with the fact of DDH problem.

VII. COMPARISONS AND EVALUATION

We present the results of theoretical analysis and evaluate the performance in the real experimental environment. The results demonstrate that the practicability of our proposed CP-ABE schemes.

A. THEORETICAL ANALYSIS

To show the features of our proposed schemes, we compare them with three existing ABE schemes ([7]–[9]) in theoretical aspects. The reasons for choosing these schemes are as follows: 1) all schemes are suitable for resource-constrained users; 2) these schemes have high efficiency in terms of encryption or decryption. The detailed results are summarized in Table 2, where E, P, SM represent an exponentiation operation ($E : \mathbb{G}_t^{\mathbb{Z}_q^*} \rightarrow \mathbb{G}_t$), a bilinear pairing operation ($P : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_t$) and a scalar multiplication operation ($SM : \mathbb{Z}_q^* \times \mathbb{G} \rightarrow \mathbb{G}_1$), respectively. Note we ignore some less time-consuming operations such as general hash operation, arithmetic operations in $\mathbb{Z}_q^*$, etc. Let k be the number of attributes of the access control structure. Let $|\mathbb{G}|$ and $|\mathbb{G}_t|$ represent the size of an element in $\mathbb{G}$ and $\mathbb{G}_t$

TABLE 2. Comparisons of the features of ABE.

Scheme	Computation Cost		Communication Cost		Access Structure	Revocation	Pairing-Free	Type	Security
	Encryption	Decryption	PK	CT					
[7]	$(k+1)SM$	$2kSM$	$(2+2l)\|\mathbb{G}\|$	$(2+2k)\|\mathbb{G}\|$	Tree	×	√	KP-ABE	CPA
[8]	$(4k+1)SM$	$2kSM$	$(1+l)\|\mathbb{G}\|$	$(1+2k)\|\mathbb{G}\|$	LSSS	√	√	CP-ABE	CPA*
[9]	$3E+(k+2)SM$	$(3kP+2kE+3kSM)$	$(6+2l)\|\mathbb{G}\|$	$(3+3k)\|\mathbb{G}\|+\|\mathbb{G}_t\|$	LSSS	×	×	KP-ABE	CCA2
Basic	$(3k+2)SM$	$(3k+1)SM$	$(4+l)\|\mathbb{G}\|$	$(2+3k)\|\mathbb{G}\|$	Tree	×	√	CP-ABE	CPA
Enhance	$(3k+2)SM$	$(3k+1)SM$	$(5+l)\|\mathbb{G}\|$	$(2+3k)\|\mathbb{G}\|$	Tree	√	√	CP-ABE	CPA

(a bilinear pairing group). Let *CPA* represent that the scheme is semantic security under the chosen-plaintext attack and *CCA* 2 represents that the scheme is semantic security under adaptive chosen-ciphertext attack. The symbol × means that the schemes fail to support the corresponding property and √ means the opposite of ×.

In Table 2, each scheme is compared in terms of the computation cost, the communication cost, the access structure, the revocation property, the pairing-free property, the scheme type, and the security. Compared with the scheme [9] based on the bilinear pairing, the pairing-free schemes ([7], [8], basic scheme, enhanced scheme) have lower computational cost by replacing complicated pairing with simple scalar multiplication. Although [9] costs the lesser in encryption by constructing an intermediate ciphertext pool, the trade-off is that it has to publish more public parameters and costs the most communication overhead. We note that both [7] and [8] are more efficient than our proposed schemes in terms of the computation cost and the communication cost. However, [7] is not a perfect scheme for the cloud-assisted smart grid since it is a KP-ABE scheme in which the policy is bounded in the private keys and users have to stores a large number of keys for different access policies. Besides, [8] is an efficient CP-ABE scheme, however, it has to face the problem of illegal key sharing among users, that is, an unauthorized user can obtain valid decryption keys only by simple transforming any legal decryption keys. To sum up, our proposed schemes are more efficient with most pairing-based ABE schemes and more secure than the known pairing-free CP-ABE scheme [8].

B. EXPERIMENTAL EVALUATION

To precisely evaluate the performance, we first test the time costs to execute different operations under different experimental environments, and then implement our proposed basic scheme.

1) OPERATIONAL TEST

We implement different cryptographic operations using a standard open-source C/C++ Cryptographic Library (Miracl, version 7.0.0),[1] and leverage a resource-limited Raspberry to simulate the system entities (e.g., data owner and data user). The detailed information is shown in Table 3, where $2^{159}+2^{17}+1$ and $2^{255}+2^{41}+1$ correspond to security levels of AES-80 and AES-128, respectively. Table 4 shows the various cryptographic operations under the security levels, and the results are the average values for 1000 rounds. We find that the bilinear operation takes an average time of 216.8ms, 3.567s while the scalar multiplication operation only needs 92.68ms, 1.12s, that is, the bilinear operation takes nearly three times as long as scalar multiplication. Therefore, reducing the number of computation-intensive bilinear operation is one of the most effective ways to ensure the practicality of schemes.

[1] https://github.com/miracl/MIRACL

TABLE 3. Experimental environment.

Sensors	RASPBERRY PI 3 MODEL B
OS	Linux 5.3.0-1042-raspi2 (8G RAM)
Compiler	gcc 9.2.1
Cryptographic Library	Miracl 7.0.0
Elliptic Curver	$y^2=x^3-3x$
Group Order	$2^{159}+2^{17}+1(2^{255}+2^{41}+1)$

TABLE 4. Time costs of three cryptographic operations under different security levels.

Security Level	Exponentiation (E)	Pairing (P)	Scalar Multiplication (SM)
AES-80 (ms)	24.5	216.8	92.68
AES-128 (s)	0.30	3.567	1.12

Public parameters size and ciphertexts size are important indicators to evaluate the communication performance of an ABE scheme. In reality, we have $|\mathbb{Z}_q^*|=160$ bits, $|\mathbb{G}|=|\mathbb{G}_t|=1024$ bits (AES-80) and $|\mathbb{Z}_q^*|=255$ bits,$|\mathbb{G}|=|\mathbb{G}_t|=3072$ bits (AES-128), respectively. As shown in Table 2, the communication cost from the data owner to the date user is $(2+3k)|\mathbb{G}|$ in our proposed schemes, while that of schemes [7], [8] are $(2+2k)|\mathbb{G}|$ and $(1+2k)|\mathbb{G}|$. Obviously, the communication costs of our proposed schemes are slightly higher than that of other schemes, however, we consider that it is a trade-off method to ensure the security of our schemes since the security is the most important thing.

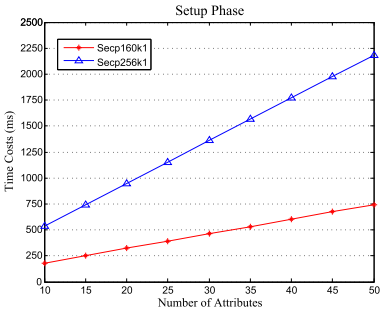


FIGURE 4. Time costs of setup phase under different elliptic curves.

2) SCHEME IMPLEMENT

To evaluate the performance, we implement the prototype of our basic scheme based on the different popular elliptic curves that do not support the bilinear pairing operation. The curves chosen are Secp160k1 and Secp256k1,[2] the corresponding security levels of that are AES-80 and AES-128, respectively. Meanwhile, we also test the time costs of the scalar multiplication. In Secp160k1, the time cost of scalar multiplication is 14.03 ms, while its time cost s 41.23 ms in Secp256k1.

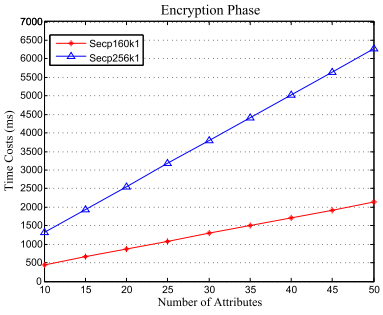


FIGURE 5. Time costs of encryption phase under different elliptic curves.

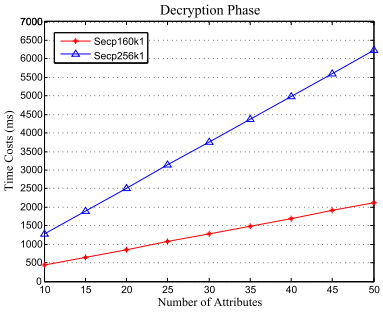


FIGURE 6. Time costs of decryption phase under different elliptic curves.

We vary different numbers of attributes from 10 to 50 to measure the computation costs of different phases. The experimental results are shown in FIGURE 4,5 and 6. We observe that the time cost of every phase (Setup, Encryption and Decryption) is linearly related with the number of attributes. Besides, the encryption and decryption phases have similar time costs in the worst case (all attributes of the access tree are used in the decryption process). Our experimental basic library is open.[3]

VIII. CONCLUSION

Assured that schemes achieve flexible access control efficiently is an important but challenging issue in resource-limited users for the cloud-assisted smart grid. In this paper, we propose two secure and effective pairing-free ciphertext-policy attribute-based encryption schemes, in which the computation-intensive bilinear pairing operation is replaced by the scalar multiplication on elliptic curves. The security proof shows that the concrete constructions are semantic security under the chosen-plaintext attack. Also, the comparison and experimental evaluation show that the proposed ABE schemes are efficient and suitable for deployment in the cloud-assisted smart grid.

An interesting future work would be to study how to efficiently privacy protection. Although our schemes ensure efficient data sharing among resource-limited users, there exists an obvious drawback, namely publicly shared access policies, which will possible users' privacy.

ACKNOWLEDGMENT

The authors would like to thank all anonymous reviewers for their suggestions and comments for improvement of this article.

REFERENCES

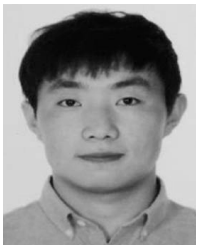
[1] A. Fromm, F. Kelbert, and A. Pretschner, "Data protection in a cloud-enabled smart grid," in *Proc. Int. Workshop Smart Grid Secur.* Berlin, Germany: Springer, 2012, pp. 96–107.
[2] J. W. Bos, W. Castryck, I. Iliashenko, and F. Vercauteren, "Privacy-friendly forecasting for the smart grid using homomorphic encryption and the group method of data handling," in *Proc. Int. Conf. Cryptol. Afr.* Berlin, Germany: Springer, 2017, pp. 184–201.
[3] Z. Guan, J. Li, L. Wu, Y. Zhang, J. Wu, and X. Du, "Achieving efficient and secure data acquisition for cloud-supported Internet of Things in smart grid," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1934–1944, Dec. 2017.
[4] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "PPFA: Privacy preserving fog-enabled aggregation in smart grid," *IEEE Trans Ind. Informat.*, vol. 14, no. 8, pp. 3733–3744, Aug. 2018.
[5] E. Ahene, Z. Qin, A. K. Adusei, and F. Li, "Efficient signcryption with proxy re-encryption and its application in smart grid," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9722–9737, Dec. 2019.
[6] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2005, pp. 457–473.
[7] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the Internet of Things," *Future Gener. Comput. Syst.*, vol. 49, pp. 104–112, Aug. 2015.
[8] S. Ding, C. Li, and H. Li, "A novel efficient pairing-free CP-ABE based on elliptic curve cryptography for IoT," *IEEE Access*, vol. 6, pp. 27336–27345, 2018.
[9] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Comput. Secur.*, vol. 72, pp. 1–12, Jan. 2018.

[2]https://www.secg.org/SEC2-Ver-1.0.pdf

[3]https://github.com/macrochen123/raspberry_miracl/tree/master

[10] J. Li, Q. Yu, Y. Zhang, and J. Shen, "Key-policy attribute-based encryption against continual auxiliary input leakage," *Inf. Sci.*, vol. 470, pp. 175–188, Jan. 2019.
[11] L. Xue, Y. Yu, Y. Li, M. H. Au, X. Du, and B. Yang, "Efficient attribute-based encryption with attribute revocation for assured data deletion," *Inf. Sci.*, vol. 479, pp. 640–650, Apr. 2019.
[12] J. Li, S. Wang, Y. Li, H. Wang, H. Wang, H. Wang, J. Chen, and Z. You, "An efficient attribute-based encryption scheme with policy update and file update in cloud computing," *IEEE Trans Ind. Informat.*, vol. 15, no. 12, pp. 6500–6509, Dec. 2019.
[13] J. Li, Q. Yu, and Y. Zhang, "Hierarchical attribute based encryption with continuous leakage-resilience," *Inf. Sci.*, vol. 484, pp. 113–134, May 2019.
[14] H. Qian, J. Li, Y. Zhang, and J. Han, "Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation," *Int. J. Inf. Secur.*, vol. 14, no. 6, pp. 487–497, 2015.
[15] I. Kim, S. O. Hwang, J. H. Park, and C. Park, "An efficient predicate encryption with constant pairing computations and minimum costs," *IEEE Trans. Comput.*, vol. 65, no. 10, pp. 2947–2958, Oct. 2016.
[16] V. K. Arthur Sandor, Y. Lin, X. Li, F. Lin, and S. Zhang, "Efficient decentralized multi-authority attribute based encryption for mobile cloud data storage," *J. Netw. Comput. Appl.*, vol. 129, pp. 25–36, Mar. 2019.
[17] J. Li, X. Lin, Y. Zhang, and J. Han, "KSF-OABE: Outsourced attribute-based encryption with keyword search function for cloud storage," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 715–725, Sep./Oct. 2017.
[18] T. Jung, X.-Y. Li, Z. Wan, and M. Wan, "Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 1, pp. 190–199, Jan. 2015.
[19] H. Ma, R. Zhang, and W. Yuan, "Comments on 'control cloud data access privilege and anonymity with fully anonymous attribute-based encryption,'" *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 4, pp. 866–867, Apr. 2016.
[20] T. V. X. Phuong, G. Yang, and W. Susilo, "Hidden ciphertext policy attribute-based encryption under standard assumptions," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 1, pp. 35–45, Jan. 2016.
[21] H. Xiong, Y. Zhao, L. Peng, H. Zhang, and K.-H. Yeh, "Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing," *Future Gener. Comput. Syst.*, vol. 97, pp. 453–461, Aug. 2019.
[22] J. Ning, Z. Cao, X. Dong, and L. Wei, "White-box traceable CP-ABE for cloud storage service: How to catch people leaking their access credentials effectively," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 883–897, Sep./Oct. 2018.
[23] Z. Liu, S. Duan, P. Zhou, and B. Wang, "Traceable-then-revocable ciphertext-policy attribute-based encryption scheme," *Future Gener. Comput. Syst.*, vol. 93, pp. 903–913, Apr. 2019.
[24] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and fine-grained attribute-based data storage in cloud computing," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 785–796, Sep./Oct. 2017.
[25] J. Cui, H. Zhou, H. Zhong, and Y. Xu, "AKSER: Attribute-based keyword search with efficient revocation in cloud computing," *Inf. Sci.*, vol. 423, pp. 343–352, Jan. 2018.
[26] H. Ma, R. Zhang, G. Yang, Z. Song, S. Sun, and Y. Xiao, "Concessive online/offline attribute based encryption with cryptographic reverse firewalls–secure and efficient fine-grained access control on corrupted machines," in *Proc. Eur. Symp. Res. Comput. Secur.* Berlin, Germany: Springer, 2018, pp. 507–526.
[27] Y. Miao, Q. Tong, K.-K.-R. Choo, X. Liu, R. H. Deng, and H. Li, "Secure online/offline data sharing framework for cloud-assisted industrial Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8681–8691, Oct. 2019.
[28] J. Li, Y. Wang, Y. Zhang, and J. Han, "Full verifiability for outsourced decryption in attribute based encryption," *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2017.2710190.
[29] Z. Liu, Z. L. Jiang, X. Wang, and S. M. Yiu, "Practical attribute-based encryption: Outsourcing decryption, attribute revocation and policy updating," *J. Netw. Comput. Appl.*, vol. 108, pp. 112–123, Apr. 2018.
[30] M. Ali, J. Mohajeri, and M.-R. Sadeghi, "A fully distributed revocable ciphertext-policy hierarchical attribute-based encryption without pairing," IACR Cryptol. ePrint Arch., Lyon, France, Tech. Rep. 2018/1102, 2018, p. 1102.

YONG WANG received the M.S. degree in electronic engineering from Shandong University, Jinan, China, in 2000. He is currently a Senior Engineer with State Grid Shandong Electric Power Company, Jinan. His main research interests include electronic engineering and information security.



BIWEN CHEN received the M.S. degree in computer sciences from the Hubei University of Technology, Wuhan, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Computer Science, Wuhan University, Wuhan. His main research interests include cryptography and information security, in particular, cryptographic protocols.

LEI LI received the M.S. degree in electronic engineering from Shandong University, Jinan, China, in 1998. He is currently a Senior Engineer with State Grid Shandong Electric Power Company, Jinan. His main research interests include electronic engineering and information security.

QIANG MA received the M.S. degree in electronic engineering from Shandong University, Jinan, China, in 2008. He is currently a Senior Engineer with State Grid Shandong Electric Power Company, Jinan. His main research interests include electronic engineering and information security.

HUICONG LI received the M.S. degree in electronic engineering from Shandong University, Jinan, China, in 2004. She is currently a Senior Engineer with State Grid Shandong Electric Power Company, Jinan. Her main research interests include electronic engineering and information security.

DEBIAO HE received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, in 2009. He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University. His main research interests include cryptography and information security, in particular, cryptographic protocols.