Hindawi
Security and Communication Networks
Volume 2019, Article ID 3249726, 9 pages
https://doi.org/10.1155/2019/3249726

Research Article

Efficient Attribute-Based Encryption with Privacy-Preserving Key Generation and Its Application in Industrial Cloud

Yujiao Song,[1] Hao Wang,[1,2] Xiaochao Wei,[1] and Lei Wu[1,3]

[1]*School of Information Science and Engineering, Shandong Normal University, China*
[2]*School of Computing and Information Technology, University of Wollongong, Australia*
[3]*Shandong Provincial Key Laboratory of Software Engineering, China*

Correspondence should be addressed to Hao Wang; wanghao@sdnu.edu.cn

Received 9 March 2019; Accepted 30 April 2019; Published 23 May 2019

Guest Editor: Mingwu Zhang

Due to the rapid development of new technologies such as cloud computing, Internet of Things (IoT), and mobile Internet, the data volumes are exploding. Particularly, in the industrial field, a large amount of data is generated every day. How to manage and use industrial Big Data primely is a thorny challenge for every industrial enterprise manager. As an emerging form of service, cloud computing technology provides a good solution. It receives more and more attention and support due to its flexible configuration, on-demand purchase, and easy maintenance. Using cloud technology, enterprises get rid of the heavy data management work and concentrate on their main business. Although cloud technology has many advantages, there are still many problems in terms of security and privacy. To protect the confidentiality of the data, the mainstream solution is encrypting data before uploading. In order to achieve flexible access control to encrypted data, attribute-based encryption (ABE) is an outstanding candidate. At present, more and more applications are using ABE to ensure data security. However, the privacy protection issues during the key generation phase are not considered in the current ABE systems. That is to say, the key generation center (KGC) knows both of attributes and corresponding keys of each user. This problem is especially serious in the industrial big data scenario, because it will cause great damage to the business secrets of industrial enterprises. In this paper, we design a new ABE scheme that protects user's privacy during key issuing. In our new scheme, we separate the functionality of attribute auditing and key generating to ensure that the KGC cannot know user's attributes and that the attribute auditing center (AAC) cannot obtain the user's secret key. This is ideal for many privacy-sensitive scenarios, such as industrial big data scenario.

1. Introduction

Due to the rapid development of new technologies such as cloud computing, Internet of Things (IoT), and mobile Internet, the data volumes are exploding, and we have truly entered the era of "Big Data." Big Data technology has been focused and applied to almost every industry, retail, healthcare, financial services, government, and so on. Particularly, in the field of industrial production, a large amount of data is generated every day, and it includes business data from information systems, machine data from industrial IoT systems, and some other data from related websites, etc. For a manufacturing enterprise, Big Data can not only be used to improve the efficiency of the business, but more importantly change the manufacturing process and business model. Industrial Big Data is the core of intelligent manufacturing and industrial IoT and provides the most favorable support for the development of Industry 4.0. How to manage and use industrial Big Data efficiently is a great challenge for every enterprise manager.

Cloud computing technology can provide better solutions to the above challenge. Using cloud technology, enterprises get rid of the heavy data management work and concentrate on their main business. Nowadays, large cloud service providers, such as Amazon, Microsoft, IBM, etc., have launched industrial cloud platforms, and more and more industrial enterprises migrate their data to these platforms. However, hosting data to third-party platforms will create new problems, because the security and privacy of the data have to depend on the credibility of the third-party.

For businesses, the biggest concern is the confidentiality of industrial data. The main solution to this problem is to use encrypting methods to protect data before uploading it. However, traditional symmetric and asymmetric encryption schemes are not appropriate for providing fine-grained access control. Therefore, the above problems have brought new challenges to data encryption, and numerous studies have focused on these issues [1–3].

Among various solutions, attribute-based encryption (ABE) [4, 5] has become an excellent candidate because of its ability to provide data confidentiality and fine-grained access control for cloud storage. Currently, more and more industrial enterprises are using ABE. In an industrial alliance, enterprises can share encrypted data based on the attributes. Only those enterprises whose attributes meet the access policy can decrypt the encrypted data. Although much research has been done on ABE [6–8], there are still some problems that have not been solved well. The current ABE systems do not consider privacy protection during the key generation phase. That is to say, the key generation center (KGC) knows the attributes and corresponding keys of each user in this system. This causes great damage to the user's privacy and data confidentiality. Particularly, in the application scenarios of industrial big data, the attributes of enterprise users may be related to the business secrets of enterprises.

1.1. Our Contribution. In order to solve the privacy protection problem in key generation phase, we propose a new ABE system, in which we separate the functionality of attribute auditing and key extracting to ensure that the KGC does not know the specific attributes of the user and that the attribute auditing center (AAC) does not obtain the user's key. In this system, when user applies its private key, it authenticates its attributes to AAC first and gets a blind token, which only certificates its attributes blindly and reveals nothing about specific attributes. The user presents the blind token to the KGC to obtain the corresponding blind key, from which user can extract the final private key. During this process, no information about the user's attributes is leaked to the KGC, and no information about the private key is leaked to the AAC. We implicitly use the oblivious transfer (OT) protocol to solve this problem. This protects the user's privacy during key generation phase.

Our ABE is suitable for privacy sensitive scenarios. Particularly, in the encryption system of industrial cloud, the attributes often involve business secrets of industrial enterprises. KGC, as a technology department, should not know these types of secret information. Therefore, we expressly introduce an application of our new scheme in the industrial cloud.

1.2. Related Work

1.2.1. Attribute-Based Encryption. Attribute-based encryption is a one-to-many public key encryption. Only the user, whose attributes satisfy the access policy set by the encryptor, can decrypt the ciphertext. This concept originates from identity-based encryption [9]. In 2005, Sahai and Waters [4] proposed the concept of fuzzy identity encryption, which became a precedent for attribute-based encryption. In 2006, Goyal et al. [5] first proposed the formal definition of attribute-based encryption (ABE), which classifies as key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). They also constructed the first KP-ABE scheme. In the next year, Bethencourt et al. [10] gave the CP-ABE construction for the first time. In a CP-ABE scheme, the encryptor sets an access policy in the ciphertext to determine which kind of users can decrypt the data. This is very consistent with the security requirements of cloud storage. In recent years, more and more researches focus on CP-ABE [11–13]. However, none of the aforementioned works deals with privacy protection problem in the key generation phase.

1.2.2. Oblivious Transfer. The concept of oblivious transfer (OT) is originally proposed by Rabin [14] in 1981, and then it became an important basic primitive in the field of cryptography. In an OT protocol, the sender delivers part of messages to the receiver and is still unaware of which parts (if any) are delivered. In other words, a secure OT protocol must satisfy two security features: (1) the sender cannot obtain the selection information of the receiver; (2) the receiver cannot obtain any information about other messages except for its choice.

In 1985, Even et al. [15] presented a specific 1-out-of-2 OT protocol (OT_2^1), in which the sender S has 2 values, and receiver R only gets one of them. Then, Brassard et al. [16] extended OT_2^1 to OT_n^1. In 1998, Stern [17] gave a generalized construction of OT protocol based on public key encryption. In 2001, Naor and Pinkas [18] gave a 2-round OT_n^1 protocol based on Diffie-Hellman assumption without random oracle. In the same year, Aiello et al. [19] constructed a 2-round OT_n^1 protocol based on homomorphic public-key encryption. In 2002, Tzeng [20] gave an OT_n^1 protocol with better round complexity and better communication complexity. In 2003, Ishai et al. [21] proposed OT extension, from which a large number of OTs can be performed using only cheap symmetric-key operations. In the past decades, OT protocol has been fully studied and widely used [22–25].

1.3. Organization. In Section 2, we introduce the preliminaries of this paper. In Section 3, we introduce the concept of attribute-based encryption with privacy preserving key generation (PPKG-ABE) and its security definition. In Section 4, we propose a specific PPKG-ABE scheme and analyze its security in Section 5. In Section 6, we introduce the application of PPKG-ABE in industrial cloud environment for protecting the security of industrial Big Data.

2. Preliminaries

2.1. CP-ABE. In CP-ABE system, there are three types of entities, i.e., key generation center (KGC), encryptor, and decryptor. The KGC issues secret key according to users' attributes. The encryptor encrypts the messages according to a designated access policy. The decryptor can decrypt the ciphertext successfully only if its attributes satisfy the corresponding access policy.

There are four algorithms in a CP-ABE scheme:

(1) Setup: it takes security parameters as input and outputs public parameters PP and master secret key MSK.

(2) KeyGen: it takes public parameters PP, master secret key MSK, and a set of attributes S as input and outputs secret key SK_S corresponding to S.

(3) Encryption: it takes public parameters PP, access policy $\mathbb{W}$, and message M as input and outputs the ciphertext $CT_{\mathbb{W}}$.

(4) Decryption: it takes public parameters PP, ciphertext $CT_{\mathbb{W}}$, and secret key SK_S as input and outputs the message M, if and only if the attributes S satisfy the access policy $\mathbb{W}$; i.e., $S \vDash \mathbb{W}$.

2.2. Oblivious Transfer. The oblivious transfer (OT) protocol is a two-party computation protocol in which one party is the sender ($\mathcal{S}$) and the other is the recipient ($\mathcal{R}$). The protocol ensures the following: $\mathcal{S}$ sends a group of messages to $\mathcal{R}$. $\mathcal{R}$ can get a subset of these messages, but $\mathcal{S}$ does not know which messages that $\mathcal{R}$ received.

In this paper, we draw on a classic (OT_2^1) protocol [26]:

Party $\mathcal{S}$ has two elements δ_0, δ_1 of group $\mathbb{G}$ and party $\mathcal{R}$ has a bit $b \in \{0, 1\}$. The descriptions of group $\mathbb{G}$ are known to both parties, where $|\mathbb{G}| = q$ and g is a generator.

(1) $\mathcal{R}$ randomly chooses $\alpha, \beta, \gamma \in [1, q]$ and sets τ as follows:

(a) If $\sigma = 0$, then $\tau = (g^\alpha, g^\beta, g^{\alpha\beta}, g^\gamma)$.

(b) If $\sigma = 1$, then $\tau = (g^\alpha, g^\beta, g^\gamma, g^{\alpha\beta})$.

$\mathcal{R}$ sends τ to $\mathcal{S}$.

(2) $\mathcal{S}$ receives $\tau = (x, y, z_0, z_1)$. Then, $\mathcal{S}$ checks $z_0 \neq z_1$. If not, it outputs $\perp$, and aborts.

In addition, $\mathcal{S}$ chooses $u_0, u_1, v_0, v_1 \in [1, q]$ randomly and computes the following 4 values:

$$\begin{aligned} \omega_0 &= x^{u_0} \cdot g^{v_0}, \\ k_0 &= (z_0)^{u_0} \cdot y^{v_0} \\ \omega_1 &= x^{u_1} \cdot g^{v_1}, \\ k_1 &= (z_1)^{u_1} \cdot y^{v_1} \end{aligned} \tag{1}$$

Then, $\mathcal{S}$ calculates $c_0 = x_0 \cdot k_0, c_1 = x_1 \cdot k_1$ and sends (ω_0, c_0) and (ω_1, c_1) to $\mathcal{R}$.

Finally, $\mathcal{R}$ calculates $k_\sigma = (\omega_0)^\beta$ and obtains $\delta_\sigma = c_\sigma \cdot (k_\sigma)^{-1}$.

2.3. Bilinear Maps. Let $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ be three q order cyclic groups. The bilinear pairing operation e is a bilinear map, $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$, and satisfies the following properties:

(1) $\forall g \in \mathbb{G}_1, \forall h \in \mathbb{G}_2, \forall x, y \in Z_q^*$, there is $e(g^x, h^y) = e(g, h)^{xy}$(2) $\exists g_0 \in \mathbb{G}_1, \exists h_0 \in \mathbb{G}_2, e(g_0, h_0) \neq 1$(3) $\forall g \in \mathbb{G}_1$, $\forall h \in \mathbb{G}_2, e(g, h)$ can be computed in polynomial time

In this paper, we use asymmetric bilinear groups; that is, $\mathbb{G}_1 \neq \mathbb{G}_2$.

2.4. Security Assumption

Definition 1. Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ form bilinear groups, let g be a generator of $\mathbb{G}_1$, and let h be a generator of $\mathbb{G}_2$. For some unknown $\alpha \in \mathbb{Z}_p^*$, define $g_i = g^{\alpha^i}$, and set $\overrightarrow{y}_{g,\alpha,n} = (g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n})$. We say an algorithm $\mathcal{B}$ solves n-BDHE problem with advantage ϵ, if on input $g, h, \overrightarrow{y}_{g,\alpha,n}$,

$$\left| Pr\left[\mathcal{B}\left(e\left(g_{n+1}, h\right)\right) = 1\right] - Pr\left[\mathcal{B}\left(Z\right) = 1\right]\right| \geq \epsilon, \tag{2}$$

where Z is a random element of $\mathbb{G}_T^*$.

The decision n-BDHE assumption holds if ϵ is negligible for any polynomial algorithm.

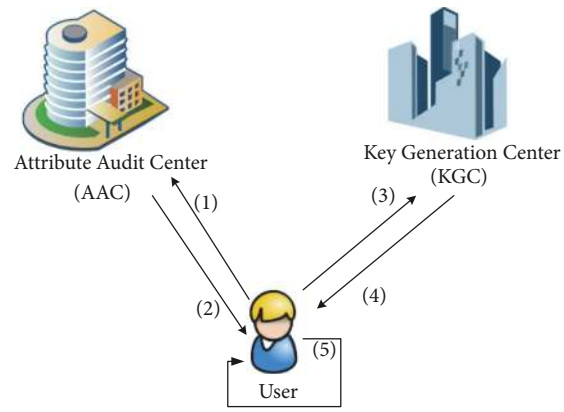

Figure 1: System model.

3. Attribute-Based Encryption with Privacy Preserving Key Generation

In the key generation phase of traditional ABE, KGC always knows the attribute information of each user. This has greatly damaged the privacy of users. In order to solve this problem, we separate the two functions of attribute auditing and key extracting. We introduce an attribute audit center (AAC) in ABE system to authenticate the attributes of users and to make blind token for them. KGC, as a simple technical support institution, is only responsible for generating keys, but it does not know the corresponding attributes of these keys.

3.1. System Model. In the key generation phase (as shown in Figure 1), there are three types of entities: attribute audit center (AAC), key generation center (KGC), and data user. In this system, user submits its attributes and relevant evidence to AAC. The AAC audits the user's attributes and returns a blind token with the signature of AAC to user. In practical applications, AAC is often carried out by the institutions that provide certification for user's attributes, such as government offices, because they know the attributes of users themselves and do not cause extra leaks. In other words, the blind token is the evidence for users owning some attributes. This token does not reveal any information of user's attributes and only ensures the authenticity. When user needs to obtain its attributes key, it will submit the blind token to KGC, which is a technical institution. The KGC first checks the legitimacy of the token; if the token is invalid, it aborts; otherwise, it runs the key generation algorithm on the token and returns a blind

key. After user obtains the blind key, it extracts the secret key locally.

The specific process is as follows:

(1) The user shows its attributes and relevant evidence to the attribute audit center (AAC).

(2) The AAC audits the user's attributes and returns a blind token to the user with its signature.

(3) When a user needs to obtain its attributes key, it will submit its blind token to the key generation center (KGC). The KGC cannot get any information about the user's attributes. It only can confirm that the user truly has related attributes.

(4) The key generation center (KGC) first checks the legitimacy of the token, and if the signature is illegal, it aborts; otherwise, it runs the key generation algorithm and outputs a blind key.

(5) The user receives the blind key from KGC and extracts the private key.

3.2. Syntax. In detail, an attribute-based encryption with privacy preserving key generation scheme (PPKG-ABE) includes seven fundamental algorithms: *Setup*, *UserTemKeyGen*, *BlindTokenGen*, *BlindKenGen*, *KeyExtra*, *Encrypt*, and *Decrypt*. The specific algorithms are described as follows:

Setup(κ)$\longrightarrow$ PP, MK: the setup algorithm is run by KGC, it inputs security parameter κ, and it outputs public parameters PP and master secret key MSK.

UserTemKeyGen(PP, κ)$\longrightarrow$ TPK_{User}, TSK_{User}: the user's temporary-key generation algorithm is run by user. It takes PP and security parameters κ as input and outputs user's temporary public key TPK_{User} and user's temporary secret key TSK_{User}.

BlindTokenGen(PP, S, TPK_{User})$\longrightarrow$ T_S: the blind token generation algorithm is run by AAC. It takes PP, user's attributes set S, and user's temporary public key TPK_{User} as input and outputs a blind token T for attributes set S.

BlindKenGen(PP, MSK, T_S)$\longrightarrow$ BSK: the blind key generation algorithm is run by KGC. It takes PP, master secret key MSK, and user's blind token T_S as input and outputs blind secret key BSK for attributes set S.

KeyExtra(BSK_S, TSK_{User})$\longrightarrow$ SK_S: the key extract algorithm is run by user locally. It takes blind secret key BSK_S and user's temporary secret key TSK_{User} as input and outputs the final secret key SK for attributes set S.

Encrypt(PP, M, $\mathbb{W}$)$\longrightarrow$ CT: the encryption algorithm is run by encryptor. It takes PP, message M, and access structure $\mathbb{W}$ as input and outputs ciphertext CT.

Decrypt($CT_{\mathbb{W}}$, SK_S)$\longrightarrow$ M: the decryption algorithm is run by decryptor. It takes ciphertext $CT_{\mathbb{W}}$ and secret key SK_S as input and outputs message M, if $S \vDash \mathbb{W}$.

We note, in PPKG-ABE scheme, that AAC is responsible for auditing user's attributes and issuing blind token T_S to user. The blind token includes a description of the authenticity of user's attributes, along with the signature of AAC, and reveals on information about specific attributes.

3.3. Security Model. We define the security in two aspects: confidentiality and privacy. Specifically, in this security model, we do not allow AAC and KGC to collude.

3.3.1. Confidentiality. We introduce the selective security model of choosing plaintext attacks for the PPKG-ABE scheme. The specific process is working between adversary $\mathscr{A}$ and challenger $\mathscr{C}$:

Init. $\mathscr{A}$ specifies an access structure $\mathbb{W}^*$ for challenge.

Setup. $\mathscr{C}$ calls the Setup algorithm and returns PP to $\mathscr{A}$.

Phase 2. $\mathscr{A}$ queries secret key on any attributes set $S \nvDash \mathbb{W}^*$. $\mathscr{C}$ returns the secret key SK for S.

Challenge. $\mathscr{A}$ submits two messages M_0^* and M_1^*, where $|M_0^*| = |M_1^*|$. $\mathscr{C}$ chooses $b \in 0, 1$ randomly and encrypts M_b^* under $\mathbb{W}^*$. Then, it returns CT^* to $\mathscr{A}$.

Phase 3. Repeats as Phase 2.

Guess. $\mathscr{A}$ guesses b' for b. The advantage Adv for $\mathscr{A}$ is defined as $Pr[b' = b]$ - 1/2.

Definition 4. The PPKG-ABE scheme is selectively IND-CPA secure if Adv is negligible for any polynomial time adversaries.

3.3.2. Privacy. We introduce a new security game for defining privacy. In this game, we define the following two oracles.

Blind Token Oracle $\mathcal{O}_{BT}$(S): it takes attributes set S as input and outputs corresponding blind token T_S.

Blind Key Oracle $\mathcal{O}_{BK}(T_S)$: it takes blind token T_S as input and outputs corresponding blind key BSK_S.

The specific process is working between adversary $\mathscr{A}$ and challenger $\mathscr{C}$:

Setup. $\mathscr{C}$ calls the Setup algorithm and returns PP to $\mathscr{A}$.

Phase 5. $\mathscr{A}$ queries blind token oracle $\mathcal{O}_{BT}$ and blind key oracle freely.

Challenge. $\mathscr{A}$ submits two attributes sets S_0^* and S_1^*, where $|S_0^*| = |S_1^*|$. $\mathscr{C}$ chooses $b \in \{0, 1\}$ randomly and queries blind token oracle $\mathcal{O}_{BT}$ on input S_b^*. It returns blind token $T_{S_b^*}$ to $\mathscr{A}$.

Phase 6. Repeats as Phase 5.

Guess. $\mathscr{A}$ guesses b' for b. The advantage Adv for $\mathscr{A}$ is defined as $Pr[b' = b]$ - 1/2.

Definition 7. The PPKG-ABE scheme is privacy-protected in key generation phase, if Adv is negligible for any polynomial time adversaries.

4. A Specific PPKG-ABE Scheme

4.1. Construction. In this construction, the PPKG-ABE scheme is constructed on the basis of [27], which only supports AND gates. Suppose that the attribute universe is U={$att_1, att_2, \ldots, att_n$}, where each att_i has 2 values: "+" and "−". The "+" denotes that user owns this attribute, while the

"−" denotes that user does not own this attribute. The specific scheme is as follows:

Setup(κ, U): the setup algorithm is run by KGC. It takes security parameters κ and attribute universe U as input, where $|U| = n$. The algorithm first chooses q order bilinear groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$, where g is a generator of $\mathbb{G}_1$ and h is a generator of $\mathbb{G}_2$. Let H be a cryptographic hash function; $H : \mathbb{G}_1 \longrightarrow \mathbb{G}_2$. For $i \in [1,n]$, it chooses $r_i, r_{n+i} \in \mathbb{Z}_p^*$, $s_i, s_{n+i} \in \mathbb{G}_2$ randomly and sets $u_i = g^{-r_i}$ and $h_i = e(g, s_i)$. It outputs

$$PP := \left\{g, h, \left(u_k, h_k\right)_{k\in[1,2n]}, H\right\},$$
$$MSK := \left\{\left(r_k, s_k\right)_{k\in[1,2n]}\right\}. \tag{3}$$

In general speaking, $\{\langle u_i, h_i\rangle | i \in [1,n]\}$ correspond to the positive attributes and $\{\langle u_{n+i}, h_{n+i}\rangle | i \in [1,n]\}$ correspond to the negative attributes.

UserTemKeyGen(PP, κ): the user's temporary-key generation algorithm is run by user. It takes public parameters PP and security parameters κ as input and chooses $\beta_i \longleftarrow \mathbb{Z}_q$ randomly for $i \in [1,n]$, as its temporary secret key TSK_{User}. Then, it calculates the temporary public key $TPK_{User} = \{h^{\beta_i}\}_{i\in[1,n]}$.

BlindTokenGen(PP, S, TPK_{User}): the blind token generation algorithm is run by AAC. It takes public parameters PP, user's attributes set S, and user's temporary public key TPK_{User} as input. S expresses an attributes set, which includes n signs, e.g., $S = (+, -, +\ldots, +)$, where "+" indicates that the user owns this attribute and "−" indicates that the user does not own this attribute. It selects $\alpha_i, \gamma_i \longleftarrow \mathbb{Z}_p$ randomly and calculates h^{α_i}, $(h^{\beta_i})^{\alpha_i}$, and h^{γ_i}, for $i \in [1,n]$.

If the attribute $att_i =$"+"$\in S$, then it sets $x_i = h^{\alpha_i}$, $y_i = h^{\beta_i}$, $z_{i,0} = h^{\alpha_i\beta_i}$, $z_{i,1} = h^{\gamma_i}$. Otherwise, the attribute $att_i =$ "−" $\in S$, and it sets $x_i = h^{\alpha_i}$, $y_i = h^{\beta_i}$, $z_{i,0} = h^{\gamma_i}$, $z_{i,1} = h^{\alpha_i\beta_i}$.

$$t_S = \left\{t_i = \left(x_i \parallel y_i \parallel z_{i,0} \parallel z_{i,1}\right)\right\}_{i\in[1,n]}. \tag{4}$$

Then, it runs standard signature algorithm on t_S to get a signature Σ and returns $T_S = (t_S, \Sigma)$ to user.

BlindKenGen(PP, MSK, T_S): the user submits the token T_S corresponding to attributes set S to the KGC for applying secret key. KGC first checks that all $x_i, y_i, z_{i,0}, z_{i,1} \in \mathbb{G}_1$, $z_{i,0} \neq z_{i,1}$ and that Σ is legal. If not, it aborts outputting $\perp$; otherwise it randomly chooses $u_{i,0}, u_{i,1}, v_{i,0}, v_{i,1} \longleftarrow \mathbb{Z}_q$ for $i \in [1,n]$ and calculates the following values:

$$w_{i,0} = x_i^{u_{i,0}} \cdot h^{v_{i,0}},$$
$$k_{i,0} = \left(z_{i,0}\right)^{u_{i,0}} \cdot y_i^{v_{i,0}}$$
$$w_{i,1} = x_i^{u_{i,1}} \cdot h^{v_{i,1}}, \tag{5}$$
$$k_{i,1} = \left(z_{i,1}\right)^{u_{i,1}} \cdot y_i^{v_{i,1}}$$

Then, it randomly chooses $v \in \mathbb{G}_2$ for each user and calculates $\sigma_{i,0} = s_i v^{r_i}$, $\sigma_{i,1} = s_{n+i} v^{r_{n+i}}$, for $i \in [1,n]$. It calculates $c_{i,0} = \sigma_{i,0} \cdot k_{i,0}$, $c_{i,1} = \sigma_{i,1} \cdot k_{i,1}$, for $i \in [1,n]$.

The blind secret key

$$BSK = \left\langle v, \left\{\left(w_{i,0}, c_{i,0}\right), \left(w_{i,1}, c_{i,1}\right)\right\}_{i\in[1,n]}\right\rangle. \tag{6}$$

It returns BSK to user.

KeyExtra(TSK_{User}, BSK): the key extract algorithm is run by user.

For $i \in [1,n]$, if $att_i =$"+", it sets $w_i = w_{i,0}$, $c_i = c_{i,0}$; else if $att_i =$ " − ", it sets $w_i = w_{i,1}$, $c_i = c_{i,1}$ and calculates

$$k_i = \left(w_i\right)^{\beta_i},$$
$$\sigma_i = \frac{c_i}{k_i}. \tag{7}$$

It outputs

$$SK := \left\langle v, \left\{\sigma_i\right\}_{i\in[1,n]}\right\rangle. \tag{8}$$

We note, in the above key issuing procedure, that KGC cannot obtain the specific attributes of user, and AAC cannot obtain the secret key.

Encrypt(PK, M, $\mathbb{W}$): it takes public key PK, AND gate structure $\mathbb{W}$, and message M as input, where $\mathbb{W} = \bigwedge_{att_i\in A}\underline{att_i}$, for A is the related attributes set, and $\underline{att_i} \in \{$" + ","− "$\}$. It chooses $s \in \mathbb{Z}_p^*$ randomly and sets $\langle u_A, h_A\rangle = \langle\prod_{att_i\in A} u_{\underline{i}}, \prod_{att_i\in A} h_{\underline{i}}\rangle$, where for each $att_i \in A$,

if $att_i =$"+", $\langle u_{\underline{i}}, h_{\underline{i}}\rangle = \langle u_i, h_i\rangle$

if $att_i =$ " − ", $\langle u_{\underline{i}}, h_{\underline{i}}\rangle = \langle u_{n+i}, h_{n+i}\rangle$

Then, it computes $ct_0 = M \cdot h_A^s, ct_1 = g^s, ct_2 = u_A^s$.

The ciphertext is defined as $CT = (\mathbb{W}, ct_0, ct_1, ct_2)$.

Decrypt(PP, SK_S, CT): if $S \vDash \mathbb{W}$, the decryption algorithm computes $sk = \langle v, \sigma = \prod_{att_i\in A}\sigma_i\rangle$, for related attributes set A. Then, it computes the message

$$M = \frac{ct_0}{e\left(ct_1, \sigma\right) \cdot e\left(ct_2, v\right)}. \tag{9}$$

4.2. Correctness. The correctness is guaranteed by

$$\begin{aligned} e\left(ct_1, \sigma\right) \cdot e\left(ct_2, v\right) &= e\left(g^s, \prod_{att_i\in A}\sigma_i\right) e\left(\prod_{att_i\in A} u_{\underline{i}}^s, v\right) \\ &= \prod_{att_i\in A}\left(e\left(g, \sigma_i\right) \cdot e\left(u_{\underline{i}}, v\right)\right)^s \\ &= \prod_{att_i\in A}\left(e\left(g, s_{\underline{i}} v^{r_{\underline{i}}}\right) \cdot e\left(g^{-r_{\underline{i}}}, v\right)\right)^s \\ &= \prod_{att_i\in A}\left(e\left(g, s_{\underline{i}}\right) \cdot e\left(g, v^{r_{\underline{i}}}\right) \cdot e\left(g^{-r_{\underline{i}}}, v\right)\right)^s \\ &= \prod_{att_i\in A}\left(e\left(g, s_{\underline{i}}\right)\right)^s = \prod_{att_i\in A} h_{\underline{i}}^s = h_A^s. \end{aligned} \tag{10}$$

5. Proof of Security

5.1. Confidentiality

Theorem 8. *If the decisional n−BDHE assumption holds for bilinear groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$, our PPGK-ABE scheme is selectively IND-CPA secure.*

Proof. If the adversary $\mathcal{A}$ can win above security game with nonnegligible advantage, we can construct an algorithm $\mathcal{B}$ to break the decision n−BDHE assumption. $\mathcal{B}$ plays the security game with $\mathcal{A}$ as follows:

Init. $\mathcal{B}$ receives challenge gate $\mathbb{W}^* = \bigwedge_{att_{i_j} \in A^*} \underline{att_{i_j}}$ from $\mathcal{A}$. We suppose $|A^*| = m < n$, and $I^* = \{i_j | att_{i_j} \in A^*\}$.

Setup. $\mathcal{B}$ chooses $j^* \in [1, m]$, $r'_k \in \mathbb{Z}_p^*$, $x_k \in \mathbb{G}_2$, $r_{i_j}, a_{i_j} \in \mathbb{Z}_p^*$ randomly, for $k \in [1, 2n]$, $j \in [1, m]$.

For $i_j \in I^* - \{j^*\}$, $\mathcal{B}$ computes public parameters as follows:

(1) If $\underline{att_{i_j}} = +$,

$$\begin{aligned} \left(u_{i_j}, h_{i_j}\right) &= \left(g^{r_{i_j}} g_{n+1-i_j}^{-1}, e(g,h)^{a_{i_j}}\right), \\ \left(u_{i_j+n}, h_{i_j+n}\right) &= \left(g^{-r'_{i_j+n}}, e\left(g, x_{i_j+n}\right)\right). \end{aligned} \tag{11}$$

(2) If $\underline{att_{i_j}} = -$,

$$\begin{aligned} \left(u_{i_j}, h_{i_j}\right) &= \left(g^{-r'_{i_j}}, e\left(g, x_{i_j}\right)\right), \\ \left(u_{i_j+n}, h_{i_j+n}\right) &= \left(g^{r_{i_j}} g_{n+1-i_j}^{-1}, e(g,h)^{a_{i_j}}\right). \end{aligned} \tag{12}$$

For $att_{i_j} = att_{i_{j^*}}$, $\mathcal{B}$ computes as follows:

(1) If $\underline{att_{i_{j^*}}} = +$,

$$\begin{aligned} &\left(u_{i_{j^*}}, h_{i_{j^*}}\right) \\ &= \left(g^{r_{i_{j^*}}} \cdot \prod_{k \in I^* - \{i_{j^*}\}} g_{n+1-k}, e(g,h)^{a_{i_{j^*}}} e(g,h)^{\alpha^{n+1}}\right) \\ &\left(u_{i_{j^*+n}}, h_{i_{j^*+n}}\right) = \left(g^{-r'_{i_{j^*}+n}}, e\left(g, x_{i_{j^*}+n}\right)\right). \end{aligned} \tag{13}$$

(2) If $\underline{att_{j^*}} = -$,

$$\begin{aligned} &\left(u_{i_{j^*}}, h_{i_{j^*}}\right) = \left(g^{-r'_{i_{j^*}}}, e\left(g, x_{i_{j^*}}\right)\right), \\ &\left(u_{i_{j^*+n}}, h_{i_{j^*+n}}\right) \\ &= \left(g^{r_{i_{j^*}}} \cdot \prod_{k \in I^* - \{i_{j^*}\}} g_{n+1-k}, e(g,h)^{a_{i_{j^*}}} e(g,h)^{\alpha^{n+1}}\right), \end{aligned} \tag{14}$$

For $i_j \notin I^*$, $\mathcal{B}$ computes

$$\begin{aligned} \left(u_{i_j}, h_{i_j}\right) &= \left(g^{-r'_{i_j}}, e\left(g, x_{i_j}\right)\right), \\ \left(u_{i_j+n}, h_{i_j+n}\right) &= \left(g^{-r'_{i_j+n}}, e\left(g, x_{i_j+n}\right)\right). \end{aligned} \tag{15}$$

□

Phase 9. $\mathcal{A}$ queries secret key on any attributes set $S \not\models \mathbb{W}^* = \bigwedge_{att_{i_j} \in A^*} \underline{att_{i_j}}$; therefore, $\exists att_{i_j} \in \mathrm{A}^*$, s.t. either $att_{i_j} \in S$ for $\underline{att_{i_j}} = -$, or $att_{i_j} \notin S$ for $\underline{att_{i_j}} = +$. Suppose that $\underline{att_{i_j}} \in S \not\models \mathbb{W}$. $\mathcal{B}$ chooses $z \in \mathbb{Z}_p^*$ randomly and computes $\nu = \overline{g_{i_j}} g^z$.

For att_{i_j}, $\mathcal{B}$ computes $\sigma_{att_{i_j}} = x_{i_j+n} (g_{i_j} g^z)^{r'_{i_j+n}}$.

For $att_{\mathrm{i}} \neq att_{i_j}$, σ_{att_i} is computed as follows:

(1) If $i = i_k \in I^* - \{j^*\}$ $(k \neq j^*)$, calculate

$$\sigma_{att_i} = g^{a_{i_k}} \left(g_{i_j}\right)^{r_{i_k}} g_{n+1-i_k+i_j} \left(u_{i_k}\right)^{-z}. \tag{16}$$

(2) If $i = i_{j^*}$, calculate

$$\sigma_{att_i} = g^{a_{i_{j^*}}} \left(g_{i_j}\right)^{r_{i_{j^*}}} \left(\prod_{k \in I^* - \{j^*\}}^{k \neq i_j} \right) \left(u_{i_j^*}\right)^{-z}. \tag{17}$$

(3) If $i \notin I^*$, calculate

(a) $\sigma_{att_i} = x_i (g_{i_j} g^z)^{r'_i}$, if $\underline{att_i} = +$;

(b) $\sigma_{att_i} = x_{i+n} (g_{i_j} g^z)^{r'_{i+n}}$, if $\underline{att_i} = -$.

$\mathcal{B}$ answers secret key query for S:

$$SK = \left\langle \nu, \left\{\sigma_{att_i} \mid i \in [1, n]\right\}\right\rangle. \tag{18}$$

Challenge. For $a_{I^*} = \sum_{j=1}^{m} a_{i_j}$, $r_{I^*} = \sum_{j=1}^{m} r_{i_j}$, $\langle u_{I^*}, h_{I^*} \rangle$ is calculated as follows:

$$\begin{aligned} u_{I^*} &= \underline{u_{i_{j^*}}} \prod_{k \in I^* - \{j^*\}} \underline{u_k} \\ &= \left(g^{r_i j^*} \prod_{k \in I^* - \{j^*\}} g_{n+1-k}\right) \prod_{k \in I^* - \{j^*\}} g^{r_k} g_{n+1-k}^{-1} = g^{r_{I^*}}, \\ h_{I^*} &= \underline{h_{i_{j^*}}} \prod_{k \in I^* - \{j^*\}} \underline{h_k} \\ &= e(g,h)^{a_{i_{j^*}}} \cdot e(g,h)^{\alpha^{n+1}} \prod_{k \in I^* - \{j^*\}} e(g,h)^{a_k} \\ &= e(g,h)^{a_{I^*} + \alpha^{n+1}}. \end{aligned} \tag{19}$$

$\mathcal{A}$ submits M_0 and M_1 with $|M_0| = |M_1|$. $\mathcal{B}$ randomly chooses $b \in \{0, 1\}$, $s' \in \mathbb{Z}_p^*$ and computes

$$\begin{aligned} CT^* &= \left(\mathbb{W}^*, ct_0^* = M_b T e(g,h)^{s' a_{I^*}}, ct_1^* = g^{s'}, ct_2^*\right. \\ &\quad \left. = g^{s' r_{I^*}}\right). \end{aligned} \tag{20}$$

If $T = e(g_{n+1}, h)$, CT^* is a valid ciphertext; else if T is random, CT^* is independent of b.

Guess. If $\mathcal{A}$ outputs $b' = b$, $\mathcal{B}$ guesses that $T = e(g_{n+1}, h)$. Otherwise, $\mathcal{B}$ guesses that T is random.

Therefore, $\mathcal{B}$ can break the decisional n−BDHE assumption with nonnegligible advantage.

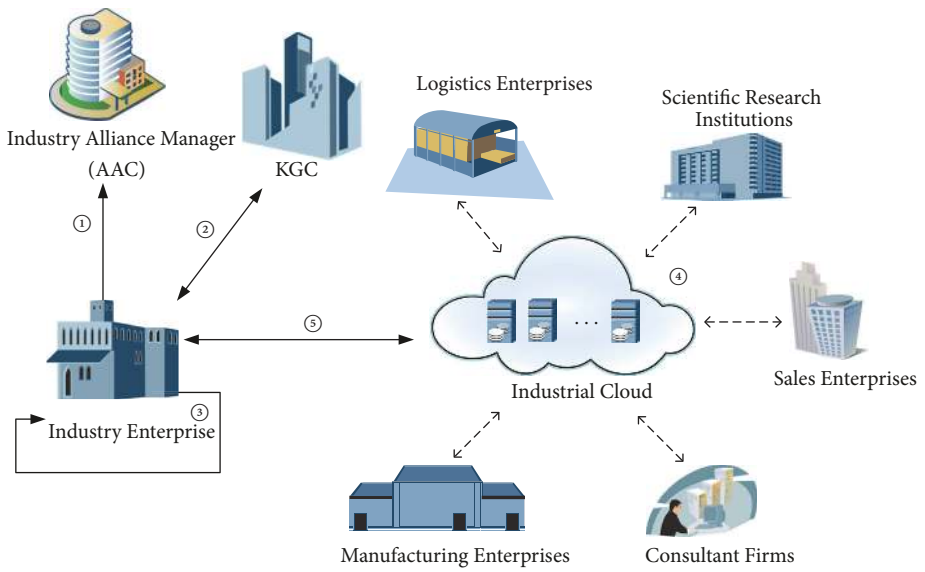

Figure 2: Application in industrial cloud.

5.2. Privacy

Theorem 10. *If the DDH assumption holds in $\mathbb{G}_2$, our PPGK-ABE scheme is privacy preserving in key generation phase.*

Proof. If DDH assumption holds in $\mathbb{G}_2$, no probabilistic polynomial-time adversary can distinguish following tuple: $(h^{\alpha}, h^{\beta}, h^{\alpha\beta}, h^{\gamma})$ and $(h^{\alpha}, h^{\beta}, h^{\gamma}, h^{\alpha\beta})$, where h is a generator of group $\mathbb{G}_2$, and α, β, γ are selected from Z_p^* randomly. Therefore, no probabilistic polynomial-time adversary can win the security game for privacy. □

6. Application in Industrial Cloud

Nowadays, new technological revolution represented by Big Data, cloud computing, and Internet of Things is changing the traditional industrial manufacturing system [28, 29]. Industrial cloud provides more convenient and secure cooperation model for the industrial enterprises [30–32]. The ABE scheme has been gradually used in the industrial cloud environment. In these applications, the qualifications, patents, and procurement plans owned by an enterprise often represent its attributes. Using traditional ABE system, the enterprise has to disclose these attributes' information that may relate to business secret to the KGC for applying the corresponding private key. Our PPGK-ABE scheme can solve this problem correctly. In this section, we introduce how to deploy our scheme in the industrial cloud environment. Figure 2 shows the specific structure of the application using our PPGK-ABE scheme. It consists of the following entities:

(i) Industry Enterprise: in this system, the role of industry enterprise is data user. They want to get useful information according to their business, but they do not want to reveal their attributes information that may relate to their business secret to KGC.

(ii) Industry Alliance Manager: in this system, the role of the industry alliance manager is AAC, which issues blind token for the attributes of industry enterprises after reviewing the relevant evidence.

(iii) KGC: its responsibility is to issue the corresponding key to the attributes of industry enterprises. In this system, KGC cannot get these attributes.

(iv) Industrial Information Provider: in this system, industrial information providers are the members of the industrial alliance and include manufacturing enterprises, sales enterprises, logistics enterprises, scientific research institutions, consultant firms, and so on. They will use ABE scheme to share their encrypted data.

(v) Industrial Cloud: industrial cloud serves as data storage center and data sharing center in this system. In order to protect security and privacy of industrial Big Data, the industrial information providers upload their data in encrypted form.

The specific workflow is as follows:

(1) After checking the relevant evidence, the industry enterprise and the industry alliance manager run *UserTemKeyGen* and *BlindTokenGen* algorithms, respectively. The industry enterprise gets the blind token corresponding to its attributes.

(2) When the industry enterprises need to ask for their attributes keys, they will submit their blind tokens to KGC. The KGC runs *BlindKenGen* algorithm and returns blind secret keys to the industry enterprises. In this process, the KGC cannot get any information about the enterprises' attributes.

(3) After receiving the blind secret keys, the industry enterprises run *KeyExtra* algorithm to obtain their

own secret key. Even if the industry alliance manager knows the attributes of industry enterprises, it does not know the secret keys corresponding to these attributes.

(4) The industrial information providers run *Encrypt* algorithm to encrypt the industrial data based on some access policies. Then, they share encrypted data on the cloud. Only the enterprises that meet the policies can access corresponding data.

(5) The industry enterprises acquire encrypted data from the cloud and run *Decrypt* algorithm to get plaintext.

In the above application, industrial information providers can share industrial data according to enterprises' attributes. Only the enterprises that meet the access policy are able to access data. Unlike traditional ABE solutions, in this application, the attributes information of enterprises will not be known by KGC. The business secret of enterprises is protected.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61602287, No. 61802235, No. 61672330, and No. 61702168), the Primary Research & Development Plan of Shandong Province (No. 2018GGX101037), and the Major Scientific and Technological Innovation Project of Shandong Province (No. 2018CXGC0702).

References

[1] F. Wang, J. Mickens, N. Zeldovich, and V. Vaikuntanathan, "Sieve: Cryptographically enforced access control for user data in untrusted clouds," in *Proceedings of the13th USENIX Symposium on Networked Systems Design and Implementation, (NSDI, '16)*, pp. 611–626, Santa Clara, Calif, USA, 2016.

[2] M. Zhang, Y. Zhang, Y. Jiang, and J. Shen, "Obfuscating eves algorithm and its application in fair electronic transactions in public clouds," *IEEE Systems Journal*, pp. 1–9, 2019.

[3] X. Li, Y. Zhu, J. Wang, Z. Liu, Y. Liu, and M. Zhang, "On the soundness and security of privacy-preserving SVM for outsourcing data classification," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 906–912, 2018.

[4] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology - EUROCRYPT '05*, vol. 3494 of *Lecture Notes in Computer Science*, pp. 457–473, Springer, Aarhus, Denmark, 2005.

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

[6] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology - EUROCRYPT '10*, vol. 6110 of *Lecture Notes in Computer Science*, pp. 62–91, Springer, Monaco, French Riviera, 2010.

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

[8] H. Wang, D. He, J. Shen, Z. Zheng, C. Zhao, and M. Zhao, "Verifiable outsourced ciphertext-policy attribute-based encryption in cloud computing," *Soft Computing*, vol. 21, no. 24, pp. 7325–7335, 2017.

[9] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of the CRYPTO '84 Advances in Cryptology*, vol. 196 of *Lecture Notes in Computer Science*, pp. 47–53, Springer, Santa Barbara, Calif, USA, 1984.

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

[11] H. Wang, D. He, J. Shen, Z. Zheng, X. Yang, and M. H. Au, "Fuzzy matching and direct revocation: a new CP-ABE scheme from multilinear maps," *Soft Computing*, vol. 22, no. 7, pp. 2267–2274, 2018.

[12] H. Wang and Y. Song, "Secure cloud-based EHR system using attribute-based cryptosystem and blockchain," *Journal of Medical Systems*, vol. 42, no. 8, pp. 152:1–152:9, 2018.

[13] H. Wang, Z. Zheng, L. Wu, and P. Li, "New directly revocable attribute-based encryption scheme and its application in cloud storage environment," *Cluster Computing*, vol. 20, no. 3, pp. 2385–2392, 2017.

[14] D. J. Lehmann and M. O. Rabin, "On the advantages of free choice: A symmetric and fully distributed solution to the dining philosophers problem," in *Proceedings of the Conference Record of the Eighth Annual ACM Symposium on Principles of Programming Languages*, pp. 133–138, Williamsburg, Va, USA, January 1981.

[15] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," *Communications of the ACM*, vol. 28, no. 6, pp. 637–647, 1985.

[16] G. Brassard, C. Crépeau, and J.-M. Robert, "All-or-nothing disclosure of secrets," in *Proceedings of the CRYPTO '86 - Advances in Cryptology*, vol. 263 of *Lecture Notes in Comput. Sci.*, pp. 234–238, Springer, Santa Barbara, Calif, USA, 1986.

[17] J. P. Stern, "A new efficient all-or-nothing disclosure of secrets protocol," in *Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security, Advances in Cryptology - ASIACRYPT '98*, vol. 1514 of *Lecture Notes in Computer Science*, pp. 357–371, Springer, Beijing, China, 1998.

[18] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols," in *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms*, pp. 448–457, SIAM, Washington, DC, USA, 2001.

[19] B. Aiello, Y. Ishai, and O. Reingold, "Priced oblivious transfer: How to sell digital goods," in *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology - EUROCRYPT '01*, vol. 2045 of *Lecture Notes in Comput. Sci.*, pp. 119–135, Springer, Innsbruck, Austria, 2001.

[20] W.-G. Tzeng, "Efficient 1-out-of-n oblivious transfer schemes with universally usable parameters," *IEEE Transactions on Computers*, vol. 53, no. 2, pp. 232–240, 2004.

[21] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank, "Extending oblivious transfers efficiently," in *Proceedings of the 23rd Annual International Cryptology Conference-Advances in Cryptology - CRYPTO '03*, vol. 2729, pp. 145–161, Springer, Santa Barbara, Calif, USA, 2003.

[22] H. Qin, H. Wang, X. Wei, L. Xue, and L. Wu, "Privacy-preserving wildcards pattern matching protocol for IoT applications," *IEEE Access*, vol. 7, pp. 36094–36102, 2019.

[23] Q. Wang, L. Gao, H. Wang, and X. Wei, "Face detection for privacy protected images," *IEEE Access*, vol. 7, pp. 3918–3927, 2019.

[24] H. Xia, J. Yu, C.-L. Tian, Z.-K. Pan, and E. Sha, "Light-weight trust-enhanced on-demand multi-path routing in mobile ad hoc networks," *Journal of Network and Computer Applications*, vol. 62, pp. 112–127, 2016.

[25] H. Xia, J. Yu, Z.-K. Pan, X.-G. Cheng, and E. H.-M. Sha, "Applying trust enhancements to reactive routing protocols in mobile ad hoc networks," *Wireless Networks*, vol. 22, no. 7, pp. 2239–2257, 2016.

[26] C. Hazay and Y. Lindell, *Efficient Secure Two-Party Protocols*, Information Security and Cryptography, Springer, Berlin, Germany, 2010.

[27] C. Chen, Z. Zhang, and D. Feng, "Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost," in *Proceedings of the 5th International Conferenceo on Provable Security, ProvSec '11*, vol. 6980 of *Lecture Notes in Computer Science*, pp. 84–101, Springer, Xi'an, China, 2011.

[28] H. Xia, S. Zhang, B. Li, L. Li, and X. Cheng, "Towards a novel trust-based multicast routing for VANETs," *Security and Communication Networks*, vol. 2018, Article ID 7608198, 12 pages, 2018.

[29] H. Xia, C. Hu, F. Xiao, X. Cheng, and Z. Pan, "An efficient social-like semantic-aware service discovery mechanism for large-scale Internet of Things," *Computer Networks*, vol. 152, pp. 210–220, 2019.

[30] R. Chaudhary, G. S. Aujla, S. Garg, N. Kumar, and J. J. P. C. Rodrigues, "SDN-Enabled multi-attribute-based secure communication for smart grid in IIoT environment," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2629–2640, 2018.

[31] M. Zhang, Y. Yao, Y. Jiang, B. Li, and C. Tang, "Accountable mobile E-commerce scheme in intelligent cloud system transactions," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 6, pp. 1889–1899, 2018.

[32] Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo, and L. Zhou, "On emerging family of elliptic curves to secure internet of things: ECC comes of age," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 3, pp. 237–248, 2017.