Secure Decentralized Access Control Policy for Data Sharing in Smart Grid

Yadi Ye, Leyou Zhang
school of Mathematics and Statistics,Xidian University
Xian, China,710126
yeyadi@yeah.net,lyzhang@mail.xidian.edu.cn

Yi Mu
University of Wollongong
Wollongong, Australia
ymu@uow.edu.au.

***Abstract*—Smart grid has improved the security, efficiency of the power system and balanced the supply and demand by intelligent management, which enhanced stability and reliability of power grid. The key point to achieve them is real-time data and consume data sharing by using fine-grained policies. But it will bring the leakage of the privacy of the users and losing of control over data for data owners. The reported solutions can not give the best trade-off among the privacy protection, control over the data shared and confidentiality. In addition, they can not solve the problems of large computation overhead and dynamic management such as users' revocation. This paper aims at these problems and proposes a decentralized attribute-based data sharing scheme. The proposed scheme ensures the secure sharing of data while removing the central authority and hiding user's identity information. It uses attribute-based signcryption (ABSC) to achieve data confidentiality and authentication. Under this model, attribute-based encryption gives the access policies for users and keeps the data confidentiality, and the attribute-based signature is used for authentication of the primary ciphertext-integrity. It is more efficient than "encrypt and then sign" or "sign and then encrypt". In addition, the proposed scheme enables user's revocation and public verifiability. Under the random oracle model, the security and the unforgeability against adaptive chosen message attack are demonstrated.**

***Index Terms*—Attribute based signcryption,Decentralized attribute based encryption,Smart grid**

I. INTRODUCTION

With the rapid development of science and technology, smart grid emerges as the times require. It meets electricity demand and uses information networks to integrate power. The difference between the smart grid and the traditional power grid is that it breaks the form of one-way information exchange and realizes the two-way information exchange between users and power companies. The realization of this two-way information flow enables the power supply company to generate power in real time according to user requirements and power requests, and enables users to collect and analyze the power consumption data in the residential building in real time according to the smart meter. Smart grid is divided into power flow and information flow [11, 12]. After the power is generated, it is distributed by the substation for use by household appliances. The general structure of a smart grid consists of six parts: bulk generation, transmission, distribution, users, control center (CC), and market. CC is the core of the smart grid, which collects the user's electricity consumption by smart meter. This collected data can help the market efficiently distribute electricity. What we need to note is that the data on user electricity consumption has economic value in the market because it can predict future consumption conditions. The information of power generation, power transmission, power distribution and power consumption is sent to the cloud server, which faces some risks, such as the disclosure of privacy and the authentication of legal users.

The privacy and security of users in information networks is something we should focus on. The schemes of [13], [16] and other schemes have proposed attribute-based access control in the smart grid. Hur et.al [9] strengthens security by hiding access policies. All of the above schemes are managed by a single center, which runs the risk of being overburdened and overpowered. Liu et.al [10] added attribute revocation in smart grid to achieve multiauthority based attribute based encryption so that the scheme is more flexible. In the same way, Decentralized attribute based encryption is added in smart grid in order to avoid key escrow problem in a single auhtority, which was introduced by Ruj et al [12]. In the meanwhile, the scheme of identity based signcryption was introduced by Alharbi et al [14] to achieve privacy-preserving in smart grid. In addition, the schemes of [21]-[24] use different methods to achieve privacy protection. However, these schemes do not achieve unforgeability while achieving large attribute domains and efficient revocation.

A. Our Contributions

In response to the above problems, our paper proposes efficient decentralized attribute-based signcryption scheme for secure data sharing that supports revocation and large attribute sets. In this scheme, The user generates private key for himself. The authority generates private key for the cloud server, and the data owner generates the signing public and private keys for the attributes sets, and generates ciphertexts. Only when the users' attributes sets satisfy the signcryption policy, the cloud server could verify successfully and partially decrypt the ciphertext for users. When users are to be revoked, the

This work was partly supported by the National Nature Science Foundation of China under Grant 61872087, the National Cryptography Development Fund under grant No.MMJJ20180209, International S&T Cooperation Program of Shaanxi Province No.2019KW-056, and Key Foundation of National Natural Science Foundation of China under grant NO.U19B2021.

cloud server only needs to delete the cloud server's private keys corresponding to users. The main features of the scheme are as follows.

1. **High efficiency:** In the signcryption phase, massive calculation arrangements to third parties, which alleviates the computational burden of RTU. In the unsigncryption phase, the user's decryption overhead is only an exponential operation and a bilinear pairing operation, regardless of the number of attributes and the complexity of the access strategy.

2. **Large universe:** The attribute in the scheme can be any string, and the length of the public key is independent of the number of attributes of the system.

3. **Public verification:** In our scheme, when the user downloads the ciphertext on the cloud server, the user's identity needs to be verified. During the verification process, any third party can perform it, which will reduce the users computational burden.

4. **Revocable:** When the user is revoked, even if the user's attributes meet the access policy, the user cannot decrypt any ciphertext.

II. Related Work

A. Multiauthority attribute based encryption

Attribute based encryption(ABE) which is also known as fuzzy identity-based encryption[1], not only protects data security, but also enables fine-grained access control. ABE is mainly divided into two parts, namely ciphertext-policy ABE(CP-ABE)[3] and key-policy ABE(KP-ABE)[2]. In CP-ABE, the data owner defines an access policy for the attribute set, and then encrypts the data according to the corresponding attribute public key. The access structure is embedded in the ciphertext, and users' private keys are generated by their attributes. If the user's attributes set satisfies the access policy defined by the data owner, the user can decrypt the ciphertext. However, KP-ABE, its access policy is contrary to CP-ABE. The above solutions posses a single authority to manage all the private keys generated by single auhtority, which may cause key escrow problem.

To solve the key escrow problem mentioned above, Chase[4] introduced multiauthority attribute based encryption. It has multiple authorities to manage disjoint sets of attributes and to generate private keys for different attributes respectively. It includes a trusted central authority. In 2011, Waters et.al [19] proposed multi-authority attribute-based encryption which consists of multiple attributes authorities to take the workload together. In 2017, Zhang et al [20] present a scheme which supports large universe and efficient revocation in multiauthority attribute based encryption. However, these schemes do not achieve ciphertext unforgeability. In the meanwhile, any string can be new attribute to be added.

B. Attribute based signcryption

The concept of attribute based signcryption(ABSC) was firstly introduced by Zheng et al [21], which combines encryption and digital signatures. The advantages of ABSC include much less communication overhead than encryption and signing steps, and ABSC can achieve confidentiality and authenticity. Gagn et al. [23] proposed an ABSC scheme in 2010 which contains a threshold access policy. Among them, users choose their access structure in the setup phase. However, in 2013, wang et al proves that Zheng et al's scheme is not secure. In 2011, Wang et al [24] proposed a ciphertext strategy and a predicate ABSC scheme based on bilinear pairs. Its efficiency is much higher than the combination of ciphertext policy attribute based signature (CP-ABS) and CP-ABE. In 2012, Emura et al [24] introduced a dynamic CP-ABSC scheme, which achieves the signing access policy updating when users' signing keys are not re-sending. Immediately, the scheme of fuzzy attribute-based signcryption[31] was proposed to achieve data encryption, access control and signature. ABSC was added in personal health records to protect users' privacy in [6], [27], [28] and [29]. In [29], this scheme combines with cuckoo filter. In [27], Huang et al present ABSC with non-monotonic access structure, and Rao et al [28] proposed ABSC realizing expressive access structures. In 2014, ABSC was introduced and achieved strong unforgeability. the scheme of Xu et al [30] used ABSC to achieve secure data's access control.

III. Preliminaries

A. Bilinear Groups

Define G_0 and G_T as two cyclic groups whose prime order is p. g is a generator of G_0. Let e be a bilinear map such that e: $G_0 \times G_0 \rightarrow G_T$. The map's properties has three aspects:

1. Bilinearity: For all $g_1, g_2 \in G$ and $u, v \in Z_p$, $e(g_1^u, g_2^v) = e(g_1, g_2)^{uv}$;

2. Non-degeneracy: $e(g, g) \neq 1$;

3. Computability: For all $g_1, g_2 \in G$, there is an effective algorithm to compute $e(g_1, g_2)$.

B. Linear Secret-sharing Scheme (LSSS)

A linear secret scheme $\prod$ is linear over Z_p if

1. The shares of a secret $s \in Z_p$ corresponding to attributes can comprise a vector over Z_p.

2. For every access structure of the property set, there is a matrix and functions.This function maps the line number of the matrix to the property. The specific definition refers to [30].

C. Decisional q-parallel bilinear diffe-hellman exponent(q-PBDHE) assumption

The q-DBPBDHE problem in group G is described below: Randomly select $a, s, b_1, b_2, \cdots, b_q \in Z_p^*$,

$$D = (p, g, G, e, g^s, \{g^{a^i}\}_{i\in[2q], i\neq q+1}, \{g^{b_j a^i}\}_{(i,j)\in[2q,q], i\neq q+1},$$

$$\{g^{\frac{s}{b_i}}\}_{i\in[q]}, \{g^{\frac{sa^i b_j}{b_{j'}}}\}_{(i,j,j')\in[q+1,q,q], j\neq j'}).$$

Distinguish between $e(g, g)^{sa^{q+1}}$ and R which is random element in G. when

$$\mid Pr[\mathcal{A}(D, e(g, g)^{sa^{q+1}}) = 0] \mid - \mid Pr[\mathcal{A}(D, R) = 0] \mid \geq \mathcal{E},$$

the advantage of the algorithm $\mathcal{A}$ to solve the $q-DBPBDHE$ problem is $\mathcal{E}$. If the advantage of any polynomial time algorithm to solve the problem of $q-DBPBDHE$ is negligible, then the hypothesis of $q-DBPBDHE$ holds in G.

D. Zero-knowledge Proof

The zero-knowledge proof protocol is essentially an agreement involving two or more parties. The prover proves to the verifier and convinces him that he knows or possesses a certain message, but does not disclose any information about the proven to the verifier during the certification process information.

IV. Overview

Following is an overview of the proposed scheme.

A. System Model

In this section, we give the core building of our scheme, as shown in Fig.2. There are cloud server, attribute authorities, RTU, user. Their roles are as follows.

(1) **Attribute authority(AA):** Every attribute authority independently manages its own attribute universe and sets up its own public key and secret key. It is responsible for generating cloud server private key and the signing secret key. It is worth noting that at least one attribute authority is credible.

(2) **RTU:** It is an trusted entity that owns data and would like to share data by outsourcing them to cloud server. RTU defines two intended access policies over attributes. The message is signcrypted using encryption access policy and signing access policy. Finally, RTU sends the confidential information to cloud server.

(3) **User(U):** Each user owns a unique global identify in the system, and possesses a set of attributes. Then U uses the above information to generate public key and decryption secret key. U downloads the ciphertext from the cloud and unsigncrypt it.

(4) **Semitrusted cloud server(C):** A cloud server is in charge of storing and verifying those encrypted data from RTUs. The cloud server verifies user's authorization, and then decrypts part of the ciphertext using the cloud server private key obtained from the attribute authorities. We assume that it is curious but honest, namely, it will honestly perform the valid assignments, but will attempt to learn information about the outsourced data as much as possible.

B. Detail of the framework

Our scheme consists of the following algorithms.

GlobalSetup(λ): Take security parameter λ as input, this algorithm outputs global public parameters of the system.

AuthoritySetup(GP)$\rightarrow \{PK_\theta, SK_\theta\}$**:** Every attribute authority generates public and private keys for itself by using global public parameters.

UserKeyGen(GP,id)$\rightarrow\{UPK_{id}, USK_{id}\}$**:** U generates the public key and private key for himself.

CSKeyGen$(GP, \{SK_\theta\}, cert(id), UPK_{id}, S) \rightarrow CSK_{id,S}$: Input $GP, \{SK_\theta\}, UPK_{id}$, user's identity certificate $cert(id)$ and the user's attribute set S, output the cloud server private key $CSK_{id,S}$ for the user. The algorithm is executed by attribute authorities and $CSK_{id,S}$ is sent to the cloud server in a secret channel. Finally, an array $\{cert(id), CSK_{id,S}\}$ is added to KT, the list of cloud secret keys.

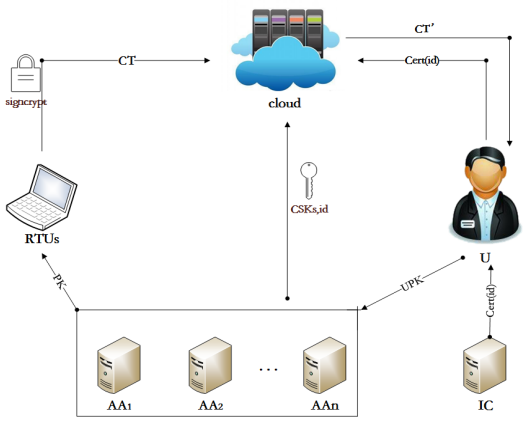

Fig. 1. System Model

SignKeyGen(PK)$\rightarrow SK_s$**:** This algorithm which is executed by RTU takes attribute authorities public keys PK to generate RTU's signing secret keys SK_s.

Signcrypt: There are two algorithms in the following steps.

(1).OS.Encrypt(GP,$\{PK_\theta\}, SK_s, W_e, W_s) \rightarrow CT_1$**:** Input GP, attribute authorities public keys $\{PK_\theta\}$, RTU's signing secret key SK_s, message M and encryption access policy W_e to generate ciphertext CT_1. This algorithm is executed by the third party and CT_1 is sent to RTU.

(2).RTU.Encrypt($GP, CT_1, W_s, m) \rightarrow CT$**:** This algorithm is executed by RTU, input GP, CT_1, the message m, and the signing policy W_s, output the ciphertext CT and upload it to the cloud server.

Unsigncrypt: There are two algorithms.

(1).Cloud.Decrypt(CT, $cert(id), CSK_{id,S}, UPK_{id}) \rightarrow CT_{id}/\perp$**:** Given ciphertext CT, user's identity certification $cert(id)$, the cloud server secret key $CSK_{id,S}$, user's public key UPK_{id}. If the user is valid, the cloud server verifies successfully and sends transformed ciphertext CT_{id} to user. Otherwise, the user gets nothing.

(2).User.Decrypt($CT_{id}, USK) \rightarrow m$**:** The user U obtains m by CT_{id} and USK.

C. Security Model

Initializaion: The adversary $\mathcal{A}$ sends challenge signature access structure W_s^* to challenger $\mathcal{C}$.

Setup: $\mathcal{C}$ sends the public parameters PK to $\mathcal{A}$ and keeps private keys secret.

Query1: $\mathcal{A}$ issues query in polynomial time.

(a) Signing queries: Choose the signing attribute set $A_s \notin W_s^*$, $\mathcal{C}$ performs the corresponding algorithm and returns the signing private keys SK_{A_s} to $\mathcal{A}$.

(b) Decryption queries: Choose decryption attribute set A_d. $\mathcal{C}$ performs the corresponding algorithm and returns private keys SK_{A_d} to $\mathcal{A}$.

(c) Signcrypt queries: For the message m, the signing attribute set A_s, the decryption attribute set A_d, the encryption access policy W_e and the signing access policy W_s, $\mathcal{C}$ performs the Signcrypt algorithm to generate the ciphertext CT returns to $\mathcal{A}$.

Unsigncrypt queries: Given a query on the ciphertext CT, the decryption attribute set A_d, the signing attribute set A_s, $\mathcal{C}$ sends message m that is generated by $\mathcal{C}$ to $\mathcal{A}$.

Challenge: After querying, $\mathcal{A}$ outputs two equal messages m_0, m_1 and the decryption access policy W_e^*. $\mathcal{C}$ chooses $\theta \in \{0,1\}$ at random and generates the ciphertext CT^*. CT^* is acted as the challenge ciphertext.

Phase 2: $\mathcal{A}$ adaptively makes queries as in Phase 1 except $A_s \notin W_s^*$.

Guess: $\mathcal{A}$ outputs a guess $\theta^{'} \in \{0,1\}$. If $\theta^{'} = \theta$, $\mathcal{A}$ wins the game, and its advantage is $Adv = \mid Pr[\theta^{'} = \theta] - \frac{1}{2} \mid$.

Definition: This scheme is unforged against adaptive chosen message attack, if there is a non-negligible advantage for adversary who loses the following game with polynomial time.

V. Construction

In our scheme, the function $\delta(\cdot) = T(\rho(\cdot))$ maps the row of a matrix to an attribute authority because $T : U \to U_\theta$ maps an attribute $i \in U$ to the attribute authority $\theta \in U_\theta$ that manages attribute i.

GlobalSetup(λ): Firstly, choose the security parameter λ, a bilinear group G with prime order p. g is generator of G. $e : G \times G \to G_T$ is bilinear pair on G. Secondly, select five collision resistant cryptographic hash functions: $H : Z_p^* \to G, H_1 : 0,1^* \to 0,1^l, H_2 : G \to Z_p^*, H_3 : 0,1 \to Z_p^*, F : U \to G$. Moreover, pick $g_1, g_2, y_0, y_1, \cdots, y_l \in G$. Output system parameters

$$GP = \{p, g, g_1, y_0, \{y_i\}_{i\in[1,l]}, G, H, H_1, H_2, H_3, F, U, U_\theta, T\}.$$

AuthoritySetup(GP): Every AA_θ ($\theta \in U_\theta$) chooses $\alpha_\theta, y_\theta \in Z_p^*$, and computes their public keys $PK_\theta = \{e(g,g)^{\alpha_\theta}, g^{y_\theta}\}$, secret keys $SK_\theta = \{\alpha_\theta, y_\theta\}$.

UserKeyGen(GP, id): The user U registeres in the identity management center(ICC), and then gets identity certificate $cert(id)$ and the id representing the identity. U picks $x_{id} \in Z_p^*$ at random, computes user's public keys

$$UPK_{id} = \{g^{x_{id}}, H(id)^{x_{id}}\},$$

and saves user's secret key $USK = (\frac{1}{x_{id}})$ secretly.

CSKeyGen($GP, \{SK_\theta\}, cert(id), UPK_{id}, S$): Zero-knowledge proof is executed between every AA_θ and the user, and the detail of the protocol is shown in Table 1. After that, AA_θ uses secret keys $\{SK_\theta\}$, system global parameters GP, user's public keys UPK_{id}, user's identity certificate $cert(id)$ and attribute set S to generate the cloud's secret key $CSK_{id,S}$. For all $i \in U$, if $T(i) = \theta$, attribute authority AA_θ randomly selects $t_i \in Z_p$ and calculates

$$K_{i,id} = g^{x_{id}\alpha_\theta} \cdot H(id)^{x_{id}y_\theta} F(i)^{t_i}, K_{i,id}^{'} = g^{t_i}.$$

AA_θ outputs the cloud's secret key about identity id

$$CSK_{id,S} = \{K_{i,id}, K_{i,id}^{'}\}_{i\in U}$$

and sends it to the cloud. Finally, the cloud C adds array $\{cert(id), CSK_{id,S}\}$ to the cloud server secret key list KT.

SignKeyGen(PK): AA_θ randomly selects $\gamma_\theta, t_\theta \in Z_p$ as its secret key, and the corresponding public key is $R_\theta = g^{\gamma_\theta}, T_\theta = g^{t_\theta}$. AA_θ also picks $v_\theta \in Z_p$ and computes the signing private keys

$$SK_s = \{K_{s,\theta} = g^{\alpha_\theta}(T_\theta)^{\gamma_\theta v_\theta}, K_{s,\theta}^{'} = g_1^{t_\theta v_\theta}\}_{\theta\in U_\theta},$$

Signcrypt(GP,$\{PK_\theta\}, SK_s, M, W_e, W_s$): There are two algorithms: *OE.Encrypt* and *RTU.Encrypt*. RTU chooses three random numbers $s, s_1, z \in Z_p$, and computes $s_2 = (s - s_1)$ mod p. Then, RTU selects the encryption policy $W_e = (M_e, \rho_e)$, the signing policy $W_s = (M_s, \rho_s)$, and sends s_2, z and W_e to the third-party server.

OE.Encrypt: The algorithm selects $s_2, y_2, \cdots, y_n, z_2, \cdots, z_n \in Z_p$, the vector $\overrightarrow{v} = (s_2, v_2, \cdots, v_n)^T, \overrightarrow{w} = (0, z_2, \cdots, z_n)^T$, and calculates $\lambda_x = (M_e\overrightarrow{v})_x, w_x = (M_e\overrightarrow{w})_x$ for $x = 1, 2, \cdots, n$, where M_x is the x'th row of matrix M_e. $r_1, r_2, \cdots, r_n$, $a_1^{'}, a_2^{'}, \cdots, a_n^{'} \in Z_p$ are randomly selected. CT_1 is computed as below.

$$\{C_{1,x} = e(g,g)^{\lambda_x} e(g,g)^{\alpha_{\delta(x)} r_x}, C_{2,x} = g^{-r_x},$$

$$C_{3,x} = g^{y_{\delta(x)} r_x} g^{w_x}, C_{4,x} = F(\rho(x))^{r_x}\}_{x\in[1,n]},$$

$$S_{\theta,j}^{'} = (K_{s,\theta}^{'})^{a_j^{'}}, S_2^{'} = (\prod_{\theta\in U_\theta} K_{s,\theta})^z.$$

$CT_1 = \{\{C_{1,x}, C_{2,x}, C_{3,x}, C_{4,x}\}_{x\in[1,n]}, \{S_{\theta,j}^{'}\}_{\theta\in U_\theta, j\in[1,n]}, S_2^{'}\}$ is sent to RTU.

RTU.Encrypt: For a given message m, RTU computes the ciphertext CT_2. The steps are as follows.

(a): RTU randomly chooses $\{\phi_x\}_{x\in Z_p}$, generates a vector $\overrightarrow{\phi} = (\phi_1, \phi_2, \cdots, \phi_n)^T$ and calculates $\lambda_x^{'} = (M_s\overrightarrow{\phi})_x$.

(b): RTU randomly selects $a_1, a_2, \cdots, a_n \in Z_p$ and computes

$$C_0 = me(g,g)^s, C^{'} = g_1^s, \mu = H_1(C^{'}),$$

$$C^{''} = (g_1^\mu g)^s, C^{'''} = g^{s_1}, \{S_j^{'} = a_j - a_j^{'}\}_{j\in[1,n]},$$

$$H_2(W_e, W_s, C_0, C^{'}, C^{''}, C^{'''}) = \beta,$$

$$H_3(\prod_{x=1}^{n}(C_{1,x}, C_{2,x}, C_{3,x}, C_{4,x}), W_e, W_s) = (j_1, \cdots, j_l),$$

$$S_2 = (S_2^{'})^{\frac{1}{z}}(y_0\prod_{i=1}^{l} y_i^{j_i})^s (C^{''})^\beta.$$

The ciphertext is $CT = \{C_0, C^{'}, C^{'''}, \{C_{1,x}, C_{2,x}, C_{3,x}, C_{4,x}\}_{x\in[1,n]}, \{S_{\theta,j}^{'}, S_j^{'}\}_{\theta\in U_\theta, j\in[1,n]}, S_2\}$.

Unsigncrypt(CT, $cert(id), CSK_{id,S}, UPK_{id}$): There are two algorithms: *Cloud.Decrypt* and *User.Decrypt*.

User	AA
1.Randomly pick $z_1, z_2 \leftarrow Z_p$ and compute $\overline{y_1} = g^{z_1}, \overline{y_2} = g_1^{z_2}$ and send $\overline{y_1}, \overline{y_2}$ to AA.	
$\overline{y_1}, \overline{y_2} \xrightarrow{}$	
	2. Randomly selects $c_1, c_2 \in Z_p$
$\xleftarrow{c_1, c_2}$	
3. compute $p_1 = z_1 - x_{id}c_1$, $p_2 = z_2 - x_{id}c_2$	
$\xrightarrow{p_1, p_2}$	4. Verify if $\overline{y_1} = g^{p_1} g^{x_{id}c_1}, \overline{y_2} = g^{p_2} g^{x_{id}c_2}$

TABLE I
ZERO-KNOWLEDGE PROOF

Cloud.Decrypt: Before the user downloads the ciphertext from the cloud server, he needs to be authenticated. This publicly verifiable process is performed by any third party, because the verification process does not require any message. U submits the identity certificate $cert(id)$ to the cloud, the cloud verifies

$$e(y_0 \prod_{i=1}^{l} y_i^{j_i}, C^{'}) \prod_{j=1}^{n} \prod_{\theta \in U_\theta} e(R_\theta^{\lambda_j^{'}}, S_{\theta,j}^{'}(K_{s,\theta}^{'})^{S_j^{'}}) \cdot e((gg_1^{\mu})^{\beta}, C^{'})$$

$$= \frac{e(S_2, g_1)}{\prod_{\theta \in U_\theta} e(g^{\alpha_\theta}, g_1)},$$

where $\sum_{j=1}^{n} \lambda_j^{'} a_j = 1$. If verifying successfully, the cloud picks $c_x \in Z_p$ which satisfies $\sum_{x=1}^{n} c_x \lambda_x = s_2, \sum_{x=1}^{n} c_x w_x = 0$, and computes $C_{1,id} = \prod_{x=1}^{n} C_{1,x}^{c_x}$, $C_{2,id} = \prod_{x=1}^{n} \left\{ e(K_{\rho(x),id}, C_{2,x}) e(H(id)^{x_{id}}, C_{3,x}) e(K_{\rho(x),id}^{'}, C_{4,x}) \right\}^{c_x}$ and returns partial ciphertext $CT_{id} = (C_0, C_{1,id}, C_{2,id})$ to user. Otherwise, the cloud outputs $\perp$.

User.Decrypt: U utilizes $USK = (\frac{1}{x_{id}})$ to decrypt message m:

$$C_{1,id} C_{2,id}^{\frac{1}{x_{id}}} = e(g,g)^{s_2}, m = \frac{C_0}{e(g,g)^{s_2} e(g, C^{'''})}.$$

.

Revoke($cert(id)$,KT): Input user's identity certificate $cert(id)$ and the cloud private keys list KT, find and delete array $\{cert(id), CSK_{id,S}\}$ in the KT. So we update the secret key list $KT = KT \setminus \{cert(id), CSK_{id,S}\}$.

VI. SECURITY ANALYSIS

A. Unforgeability

Theorem 1: If the adversary $\mathscr{A}$ could break the $EUF - CMA$ security of our scheme with a non-negligible advantage $\mathcal{E}$, then we could use the algorithm $\mathcal{B}$ to solve $q - CDHE$ problem with the probability $\mathcal{E}^{'} = \mathcal{E}\lambda(l+1)$, where λ is a security parameter and l is the hash function H_1's outputs length.

Proof: The specific procedure of proof is in the appendix.

B. Confidentiality

Theorem 2: Assume that the scheme of Rouselakies-Waters is statically secure, then our encryption scheme is also statically secure.

Proof: A detailed proof is in the Appendix.

VII. PERFORMANCE ANALYSIS

A. Security and Functionality

In this section, we compare the six aspects of authority, large attribute domain, revocation, access structure, unforgeability, and authentication. Through comparison in Table 1, this scheme realizes efficient revocation, flexible access structure, unforgeability, authentication and reaches large attribute domains in a decentralized environment. As shown in Table 2,

TABLE II
COMPARISONS OF FUNCTIONALITIES WITH OTHER SCHEMES

Schemes	[5]	[26]	[32]	[38]	[39]	Ours
Authority	Multiple	Multiple	Single	Single	Multiple	Multiple
Attribute universe	Small	Large	Small	Small	Small	Large
Revocation	No	Yes	No	No	Yes	Yes
Access structure	Circuit	LSSS	LSSS	LSSS	LSSS	LSSS
Unforgeability	No	No	Yes	Yes	Yes	Yes
Authentication	Yes	No	Yes	No	Yes	Yes
Encryption Outsourcing	No	No	No	No	Yes	Yes
Decryption Outsourcing	Yes	Yes	No	No	Yes	Yes

we provide a comparison of some important features, such as ciphertext storage, decryption key storage, and the signing key storage. this scheme is compared with [32], [37], [38] and [39] in the length of decryption keys, signing keys, and ciphertext. $\mid U_\theta \mid$ represents the number of authorities. $\mid G \mid$ and $\mid G_T$ represent the size of groups G and G_T. l_e, l_s are the number of encryption attributes and signing attributes, respectively. By comparison, it can be obtained that the length of the decryption private key of this scheme is only $\mid G \mid$.Note that $\mid G \mid, \mid G_T \mid$ is set to 512bits.

From the comparison of Figures 1-3, although the ciphertext storage of this scheme is relatively large, the decryption key storage is only constant and has the least storage.

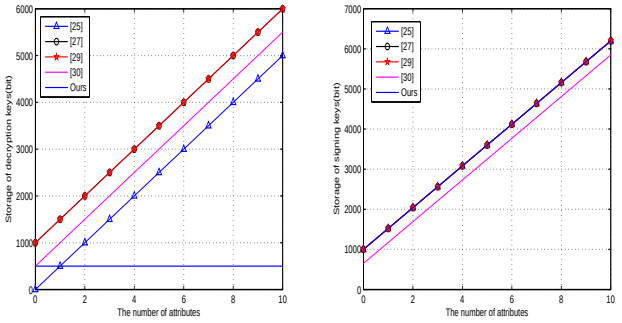

Fig. 2. Comparison on storage of decryption and signing keys between different schemes

B. Performance

Figure 3 shows the comparison between this scheme and the previous scheme in signcryption and decryption. Here, E represents the time taken for the exponential operation, and P represents the time taken for the pairing operation. It can be seen from Fig. 4 that compared with other schemes, the time required for decrypting the signcryption in this scheme is only an exponential operation and a bilinear pairing operation.

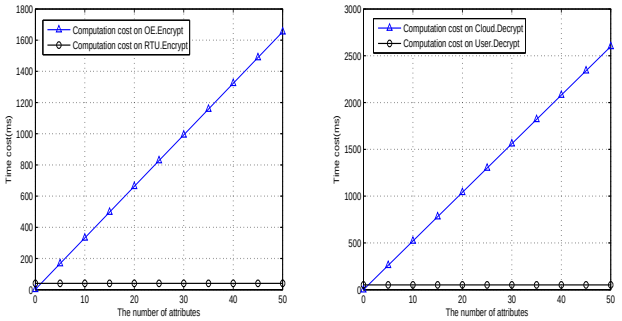

Fig. 3. The time cost of signcryption and unsigncryption

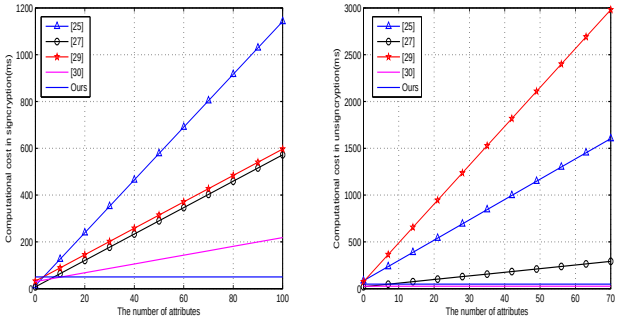

Fig. 4. Comparison on computation cost of signcryption and unsigncryption between different schemes

VIII. Conclusion

In this paper, we propose decentralized attribute-based signcryption in smart grids, which achieves secure and efficient data sharing. The solution meets large attribute domains and can support efficient user revocation. This solution generates a cloud secret key so that the cloud can decrypt part of ciphertext to reduce the amount of calculation on the user side. Moreover, outsourced signcryption and outsourced unsigncryption are applied to alleviate computational burden. Therefore, this system can be used in lightweight mobile device.

References

[1] A.Sahai,B.Waters,Fuzzy Identity-Based Encryption. in *Proc.EUROCRYPT*, vol. LNCS 3494, pp. 457-473, May 2005.

[2] V.Goyal,O.Pandey,A.Sahai and B.Waters,Attribute-based encryption for fine-grained access control of encrypted data,*Proc.13th ACM conference on Computer and Communication Security*,pp:89-98,2006.

[3] J.Bethencourt,A.Sahai and B.Waters,Ciphertext-policy attribute-based encryption,*Proc.IEEE Symposium on Security and Privacy*,pp:321-334,2007.

[4] Chase,M.Multi-authority attribute based encryption.In:Theory of Cryptography Conference.Springer,pp:515-534,2007.

[5] J Xu, Q Wen, W, Li, J Shen and D ,He, Succinct multi-authority attribute-based access control for circuits with authenticated outsourcing, *Soft Comput*,pp:5265-5279,2017.

[6] Y.S.Rao, A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing, *Future Gener.Comput.Syst.*,vol.67, pp:133-151, 2017.

[7] S.M.Sedaghat; M.H.Ameri; J.Mohajeri and M.R.Aref, An efficient and secure Data Sharing in Smart Grid:ciphertext-policy Attribute-Based Signcryption. *25th Iranian Conference on Electrical Engineering*. pp:2003-2008,2017.

[8] FH,Deng;YL,Wang;L,Peng;H,Xiong;J,Geng. Ciphertext-Policy Attribute-Based Signcryption With Verifiable Outsourced Designcryption for Sharing Personal Health Records. in *IEEE Access*, vol.6, pp.39473-39486, 2018.

[9] J. Hur, Attribute-based secure data sharing with hidden policies in smart grid, in *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2171-2180, Nov 2013.

[10] D. Liu, H. Li, Y. Yang, and H. Yang,Achieving multi-authority access control with efficient attribute revocation in smart grid, in *IEEE International Conference on Communications (ICC)*, pp. 634-639, Jun 2014.

[11] F. Li, B. Luo, and P. Liu, Secure information aggregation for smart grids using homomorphic encryption, in *Proc.IEEE SmartGridComm*, pp. 327-332,2010.

[12] S. Ruj, A. Nayak,A Decentralized Security Framework for Data Aggregation and Access Control in Smart Grids, in *IEEE TRANSACTIONS ON SMART GRID*, VOL.4, NO.1, pp. 196-205, March 2013.

[13] Y. Liu, W. Guo, C.-I. Fan, L. Chang and C. Cheng,A Practical Privacy-Preserving Data Aggregation (3PDA) Scheme for Smart Grid, in *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, VOL. 15, NO.3, pp. 1767-1774, March 2019.

[14] K. Alharbi, X. Lin,Efficient and Privacy-preserving Smart Grid Downlink Communication Using Identity Based Signcryption, in *2016 IEEE Global Communications Conference*, pp. 56-62, December 2016.

[15] C. Ruland, J. Sassmannshausen,Firewall for Attribute-Based Access Control in Smart Grids, in 2018 the 6th IEEE International Conference on Smart Energy Grid Engineering, pp.336-341, 2018.

[16] R. Chaudhary, G. S. Aujla, S. Garg and Joel J.P.C. Rodrigues,SDNEnabled Multi-Attribute-Based Secure Communication for Smart Grid in IIoT Environment, in *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, VOL. 14, NO. 6, pp:2629-2640, June 2018.

[17] A. Mutsvangwa and B. Nleya,Secured Access Control Architecture Consideration For Smart Grids, in *2016 IEEE PES Power Africa Conference*, pp:228-233, 2016.

[18] Z. Cao, B. Lang and J. Wang,An Efficient and Fine-grained Access Control Scheme for Multidimensional Data Aggregation in Smart Grid, in *2016 IEEE TrustCom-BigDataSE-ISPA*, pp:362-369, 2016.

TABLE III
COMPARISONS OF STORAGE WITH OTHER SCHEMES

Schemes	[32]	[37]	[38]	[39]	Ours
Decryption keys	$l_e \mid G \mid$	$(l_e+2) \mid G \mid$	$(l_e+2) \mid G \mid$	$(l_e+1) \mid G \mid$	$\mid G \mid$
Singing keys	$(l_s+2) \mid G \mid$	$(l_s+2) \mid G \mid$	$(l_s+2) \mid G \mid$	$(l_s+1) \mid G \mid$	$2 \mid U_\theta \mid\mid G \mid$

TABLE IV
COMPARISONS OF COMPUTATION COST WITH OTHER SCHEMES

Schemes	[32]	[37]	[38]	[39]	Ours
Signcryption	$(4l_s+2l_e+7)E$	$(l_e+2l_s+4)E$	$(2l_e+l_s+l+7)E$	$(l_s+l+6)E$	$P+(l+8)E$
Unsigncryption	$(l_s+2l_e+2)E$ $+(l_s+5)P$	$P+(l_s+2)E$	$(2l_e+4)P+(5l_e$ $+5)E$	E	$E+P$

[19] Y Rousealkis and B.Waters, Efficient statically-secure large universe multi-authority attribute-based encryption, in *Finanical Cryptography and Data Security*. Berlin: Springer,pp:568-588, 2015.

[20] K.Zhang, JF.Ma, H.Li, JH.Zhang and T.Zhang, Multi-auhtority attribute based encryption scheme supporting efficient revocation. *Journal on Communications*, VOL.38, NO.3, PP:83-91, March 2017.

[21] Zheng, Y. Digital signcryption or how to achieve cost (signature & encryption) $\leq$ cost (signature)+ cost(encryption). In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA,17-21, pp. 165-179, August 1997.

[22] Gagn, M.; Narayan, S.; Safavi-Naini, R. Threshold attribute-based signcryption. In Proceedings of the International Conference on Security and Cryptography for Networks, Amalfi, Italy, 13-15, pp. 154-171, September 2010.

[23] Wang, C.; Huang, J. Attribute-based signcryption with ciphertext-policy and claim-predicate mechanism. In Proceedings of the International Conference on Computational Intelligence and Security (CIS), Sanya,Hainan, China, 3-4, pp: 905-909, December 2011.

[24] Emura, K.; Miyaji, A.; Rahman, M. S. Dynamic attribute-based signcryption without random oracles. Int. J.Appl. Cryptogr. 2, 199-211, 2012.

[25] Rao, Y.S. A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing. Future Gener. Comp. Syst. 67, 133-151,2017.

[26] Pandit, T.; Pandey, S.K.; Barua, R. Attribute-based signcryption: Signer privacy, strong unforgeability and ind-cca2 security in adaptive-predicates attack. In Proceedings of the International Conference on Provable Security, Hong Kong, China, 9C10 , pp. 274-290,October 2014.

[27] Rao, Y.S.; Dutta, R. Efficient attribute-based signature and signcryption realizing expressive access structures.Int. J. Inf. Secur. 15, 81-109, 2016.

[28] F.H.Deng, Y.L.Wang, L.Peng, H.Xionga and J.Geng, Ciphertext-Policy Attribute-Based Signcryption With Verifiable Outsourced Designcryption for Sharing Personal Health Records, *IEEE*, pp:39473-39486,2018.

[29] Yang,M.;Zhang,T.;Efficient privacy-preserving access control scheme in electronic health records system. *Sensors*.18,3520-3525, 2018,.

[30] Q.Xu, C,Tan, Z.Fan, W.Zhu,Y.Xiao and F.Cheng, Secure Data Access Control for Fog Computing Based on Multi-Authority Attribute-Based Signcryption with Computation Outsourcing and Attribute Revocation, *Sensors*, May,2018.

IX. APPENDIX

X. SECURITY ANALYSIS

A. Unforgeability

Theorem 1: If the adversary $\mathscr{A}$ could break the $EUF-CMA$ security of our scheme with a non-negligible advantage $\mathcal{E}$, then we could use the algorithm $\mathcal{B}$ to solve $q-CDHE$ problem with the probability $\mathcal{E}^{'}=\mathcal{E}\lambda(l+1)$, where λ is a security parameter and l is the hash function H_1's outputs length.

Proof: There is an $q-CDHE$ problem 's instance $y=(g,g_1,g_2,\cdots,g_q,g_{q+2},\cdots,g_{2q})$, where $a\in Z_p$, g is a generator of G and $g_i=g^{a^i}$. The algorithm $\mathcal{B}$'s goal is to compute g_{q+1}. Firstly, $\mathcal{B}$ chooses four hash functions$H_1:\{0,1\}^*\rightarrow\{0,1\}^l, H_2:G\rightarrow Z_p^*, H_3:\{0,1\}\rightarrow Z_p^*$ and $H_4:\{0,1\}\rightarrow Z_p^*$. Secondly, $\mathcal{B}$ acts as the challenger in $EUF-CMA$ security game and interacts with $\mathcal{A}$ in the following steps.

Initialization: $\mathcal{A}$ selects the challenge signing access policy $W_s^*=(A_s^*,\rho_s^*)$ to $\mathcal{B}$. A_s^* , a matrix of $l^*\times n^*$, labels the function ρ_s^* in the W_s^*.

Setup: $\mathcal{B}$ selects $\beta^{'}\in Z_p^*, d,d^{'}\in Z_p^*$ at random and defines $\beta=\beta^{'}+a^{q+1}, e(g,g)^{\beta}=e(g^a,g^{a^q})e(g,g)^{\beta^{'}}, g_1=g^d, g_2=g^{d^{'}}$. $\mathcal{B}$ also pocks $(z_0,z_1,\cdots,z_l)\in Z_p^{l+1}$ at random, $\lambda^{'}=\lambda$ and $\lambda^{'}(l+1)<p$, where λ is a security parameter. In addition, $\mathcal{B}$ sets $y_0=g_p^{p-\lambda^{'}\pi+b_0}, y_i=g_p^{b_i}g^{z_i}$ for all $i\in[1,l]$, where $\pi(0<\pi<l)$ and $(b_0,b_1,\cdots,b_l)\in Z_{\lambda^{'}}^{l+1}$. $\mathcal{B}$ also defines two functions $F_1(\overrightarrow{j})=p-\lambda^{'}\pi+b_0+\sum_{i=1}^{l}j_ib_i$ and $F_2(\overrightarrow{j})=z_0+\sum_{i=1}^{l}j_iz_i$ so that $y_0\prod_{i=1}^{l}y_i^{j_i}=g_q^{F_1(\overrightarrow{j})}g^{F_2(\overrightarrow{j})}$. For each attribute $x\in S$, we note that X is the set of indices i, such that $\rho_s^*(i)=x$. If $X\neq\emptyset$, $\mathcal{B}$ picks $f_x\in Z_p$ and computes $h_x=g^{f_x}g^{aM_{i,1}^*}g^{a^2M_{i,2}^*}\cdots g^{a^{n^*}M_{i,n}^*}$, else $X=\emptyset$, $h_x=g^{f_x}$.

Finally, $\mathcal{B}$ sends the global parameters

$$GP=\{p,g,g_1,g_2,y_0,\{y_i\}_{i\in[1,l]},G,H,H_1,H_2,H_3,F,U,U_\theta,T\}$$

and the signing public key is $e(g,g)^{\beta}$.

Signing keys queries: For the signing attribute set M_s, $\mathcal{A}$ makes a query. If $M_s\notin W_s^*$, $\mathcal{B}$ randomly picks $\hat{r}\in Z_p$ and computes $\overrightarrow{\eta}=(\eta_1,\eta_2,\cdots,\eta_{n^*})\in Z_p^{n^*}$, where $\eta_1=-1$ so that $\overrightarrow{\eta}M_i^*=0$ and $\rho_s^*(i)\in M_s$ for all i. Moreover, $\mathcal{B}$ defines $r_s=\hat{r}+\eta_1a^q+\eta_2a^{q-1}+\cdots+\eta_{n^*}a^{q-n^*+1}$, and then calculates

$$K_s^{'}=g^{\hat{r}}\prod_{i=1}^{n^*}(g^{a^{q+1-i}})^{\eta_o}, K_s=g^{\beta^{'}}g^{a\hat{r}}\prod_{i=2}^{n^*}(g^{a^{q+2-i}})^{\eta_i}$$

$$K_{S,x}=(K_s^{'})^{f_x}\prod_{j=1}^{n^*}(g^{a^j\hat{r}}\prod_{o=1,\cdots,n^*o\neq j}(g^{a^{q+1+j-o}})^{\eta_o})^{M_{i,j}^*},$$

where $x\in A_s$. If $\rho_s^*(i)\neq x$ for all i, $\mathcal{B}$ implicitly selects $K_{s,x}=(K_s^{'})^{f_x}$. Finally, $\mathcal{B}$ sends the signing secret keys SK_{M_s} to $\mathcal{A}$.

decryption keys queries: Firstly, select some valid users and ask the corresponding public and private keys of users:

$UPK_{id} = \{g^{x_{id}}, H(id)^{x_{id}}\}, USK_{id} = \{\frac{1}{x_{id}}\}$. Secondly, select the uncorrupted AAs and ask for the corresponding public keys: $\{PK_\theta\}$. Moreover, ask for the private key of the cloud server with $\{S, id\}$ and we could obtain $CSK_{S,id}$= $\{K_{i,id} = g^{x_{id}\alpha_\theta} H(id)^{y_\theta} F(i)^{t_i}, K_{i,id}^{'} = g^{t_i}\}$.

Signcrypt queries: When $\mathcal{A}$ makes a query to $\mathcal{B}$, and $\mathcal{B}$ makes the corresponding responds according to the two following conditions:

(a) If $M_s \notin W_S$, $\mathcal{B}$ runs the signing keys queries to obtain the private keys SK_{M_s}, generates CT by running the Signcrypt algorithm and returns to $\mathcal{A}$.

(b) If $M_s \in W_s$, $\mathcal{B}$ performs in the following steps: $\mathcal{B}$ chooses $\phi \in Z_p^l$ at random and computes a vector $\overrightarrow{\phi} = (-\phi_1, -\phi_2, \cdots, -\phi_l)$ such that $\overrightarrow{\phi} M_s = -\overrightarrow{1_n}$. If $\rho_s(i) \notin M_s$, $\phi_i = 0$ for all $i \in [1, l]$. $\mathcal{B}$ computes $C_0 = me(g,g)^s, S_1 = g^{r_s}$ and $H_1(S_1, W_e, W_s) = (j_1, j_2, \cdots, j_l)$. If $F(\overrightarrow{j}) = 0$, $\mathcal{B}$ stops. Otherwise, $\mathcal{B}$ randomly selects $s^{'} \in Z_p^*$, defines $s = s^{'} - \frac{a}{F_1(j)}$ and calculates

$$C^{'} = g^{s^{'}} g_1^{-1/F_1(\overrightarrow{j})}, C^{''} = g^{(d\mu + d^{'})s^{'}} g_1^{-(\mu d + d^{'})/F_1(\overrightarrow{j})}$$

where $\mu = H_2(C^{'})$.

$$S_2 = g^{\beta^{'}} g^{ar_s} (\prod_{i=1}^{l} (h_{\rho_s(i)}^{r_s})^{\phi_i}) (g_q^{F_1(\overrightarrow{j})} g^{F_2(\overrightarrow{j})})^{s^{'}} (g_1^{-F_2(\overrightarrow{j})/F_1(\overrightarrow{j})})$$

$(C^{''})^{\beta}$, where $\beta = H_3(W_e, W_s, C, C^{'}, C^{''}, \{C_{1,x}, C_{2,x}, C_{3,x}, C_{4,x}\}_{x\in[1,l]}, S_1, S_2)$. Finally, $\mathcal{B}$ sends the ciphertext $CT = \{C, C_{\prime}, C_{\prime\prime}, \{C_i\}_{i\in[1,l]}, S_1, S_2\}$ to $\mathcal{A}$.

Unsigncrypt queries: When $\mathcal{A}$ makes a query on CT, $\mathcal{B}$ calculates the decryption private key $CSK_{S,id}$ and USK_{id} and performs the Unsigncrypt algorithm to generate the message m and returns to $\mathcal{A}$.

Forgery: $\mathcal{A}$ publishes the effective forgery ciphertext

$$CT^* = \{C^*, C^{'*}, C^{''*}, \{C_{1,x}^*, C_{2,x}^*, C_{3,x}^*, C_{4,x}^*\}_{i\in[1,l]}, S_1^*, S_2^*\}.$$

which is on (m^*, W_e^*, W_s^*). CT^* meets two conditions: (1), the result of unsigncrypt algorithm is $m^* \neq \perp$ when $A_d^* \in W_e^*$; (2), $\mathcal{A}$ can not issue the signcrypt queries on (m^*, W_e^*, W_s^*). In the following step, $\mathcal{B}$ uses the method to solve the $q - CDHE$ problem.

$\mathcal{B}$ calculates $\overrightarrow{j}^* = (j_1^*, j_2^*, \cdots, j_l^*)$. If $b_0 + \sum_{i=1}^{l} j_i b_i \neq \lambda\pi$, then $\mathcal{B}$ aborts. Otherwise, $F_1(\overrightarrow{j}^*) = 0$ mod p, and $\mathcal{B}$ computes

$$C_0^* = m^* e(g,g)^s, C^{'} = g^s, C^{''} = g^{(d\mu + d^{'})s}$$

$$\{C_{1,x}^* = e(g,g)^{\lambda_x^*} e(g,g)^{\alpha_{\delta(x)^*} r_x}, C_{2,x}^* = g^{-r_x},$$

$$C_{3,x}^* = g^{y_{\delta(x)^*} r_x} g^{w_x}, C_{4,x}^* = F(\rho(x)^*)^{r_x}\}_{x\in[1,l]},$$

$$S_1 = K_s^{'} = g^{r_s}, H_1((S_1^*, W_e^*, W_s^*) = (j_1^*, \cdots, j_l^*),$$

$$H_3(W_e^*, W_s^*, C^*, C^{'*}, C^{''*}, \{C_{1,x}^*, C_{2,x}^*, C_{3,x}^*, C_{4,x}^*\}_{x\in[1,l]}, S_1,$$

$$S_2) = \beta^*, S_2^* = K_s (y_0 \prod_{i\in[1,l^*]} y_i^{j_i^*})^s \prod_{x\in[1,l^*]} (K_{S,\rho(x)^*})^{\phi_x^* A_{S,x}^*} (C^{''})^{\beta^*}$$

where $\mu^* = H_2(C^{'*}), \beta^* = H_3(W_e^*, W_s^*, C^*, C^{'*}, C^{''*}, \{C_{1,x}^*, C_{2,x}^*, C_{3,x}^*, C_{4,x}^*\}_{x\in[1,l^*]}, S_1^*, S_2^*)$ and the vector $\overrightarrow{\phi}^* = (-\phi_1, -\phi_2, \cdots, -\phi_{l^*})$ meets $\sum_{i=1}^{l^*} \phi^* M_{S,i}^* = -\overrightarrow{1_{n^*}}$. Finally, $\mathcal{B}$ can compute $S_2^* / \{g^{\beta} (\prod_{i=1}^{l^*} (S_1^*)^{f_{\rho_s^*}(i)}) (C^{'*})^{F_2(\overrightarrow{j^*}) + (d\mu + d^{'})\beta^*}\} = g_{q+1}$.

B. Confidentiality

Theorem 2: Assume that the scheme of Rouselakies-Waters is statically secure, then our encryption scheme is also statically secure.

Proof: Assuming there is the adversary $\mathcal{A}$ who is capable of breaking the our scheme with a non-negligible probability $\mathcal{E}$, then we can construct a simulator $\mathcal{B}$ who breaks the scheme of RW with an advantage $\mathcal{E}$. Let $\mathcal{C}$ is the challenger who is in the RW's scheme to interact with $\mathcal{B}$.

GlobalSetup: The challenger $\mathcal{C}$ sends global parameters

$$GP = \{p, g, g_1, g_2, y_0, \{y_i\}_{i\in[1,l]}, G, H, H_1, H_2, H_3, F, U, U_\theta, T\}$$

to the simulator $\mathcal{B}$. $\mathcal{B}$ publishes GP to be global parameter of our scheme and sends GP to $\mathcal{A}$.

Signing keys queries: $\mathcal{B}$ picks $\alpha, r_s \in Z_p$ computes the singing private keys

$$SK_s = \{K_s = g^{\alpha} g^{y r_s}, K_s^{'} = g^{r_s}, \{K_{S,i} = h_i^{r_s}\}_{i\in A_s}\},$$

where $y = \sum_{\theta\in l} y_\theta = \sum_{\theta\in l} (\widetilde{y_\theta} + \sum_{x\in X} \sum_{j=2}^{n^{'}} b_x a^{q+2-j} A_{x,j}^{'})$.

Decryption keys queries: The adversary $\mathcal{A}$ chooses some corrupt $\{AAs\} \subseteq U_\theta$, and generates their public keys $\{PK_\theta\}_{\theta\in AAs}$ and sends them to $\mathcal{B}$. Next, $\mathcal{A}$ issues queries to $\mathcal{B}$ according to our scheme;

(1). $\mathcal{B}$ chooses some authorized $\{AAs^{'}\} \in U_\theta$, makes queries about their public keys and sends public keys to $\mathcal{A}$.

(2). Choose some authorized users $\{id_i\}_{i=1}^m$, and issue queries about their public keys and secret keys. $\mathcal{B}$ sends these users' public keys to $\mathcal{A}$.

(3). Choose $\{S_i, id_i\}_{i=1}^m$ and make queries about the cloud secret keys, where S_i is attributes set possessed by user id_i. $T(S_j) \cap C_\theta = \emptyset$. There are two cases:

(a) For $1 \leq i \leq m, j \in S_i$, compute $K_{j,id_i} = (g^{\alpha_\theta} H(id_i)^{y_\theta} F(j)^t)^{x_{id_i}}, K_{j,id_i}^{'} = (F(j)^t)^{x_{id_i}}$.

(b) For $m \leq i \leq n, j \in S_i$, select $g_j \in G$ and $t_j \in Z_p^*$ at random, calculate $K_{j,id_i} = g_j F(j)^{t_j} g_j, K_{j,id_i} = F(j)^{t_j}$, where $g_j = (g^{\alpha_\theta} H(id_i)^{y_\theta})^{x_{id_i}}$. Consequently, $K_{j,id_i} = g^{\alpha_\theta x_{id_i}} H(id_i)^{y_\theta x_{id_i}} F(j)^{t_j}, K_{j,id_i}^{'} = F(j)^{t_j}$.

$\mathcal{B}$ sends the above the cloud secret keys $\{CSK_{S_i,id_i}\}_{i=1}^m$ to $\mathcal{A}$.

Signcrypt queries: $\mathcal{A}$ chooses two messages m_0, m_1 with equal length, and a challenging access policy (A^*, ρ^*). It's noting that every user who issues query can not satisfy the access policy (A^*, ρ^*). $\mathcal{B}$ signcrypts the message similar to the above way.

Guess: $\mathcal{A}$ outputs the guess $b^{'} \in \{0, 1\}$. In a similar way, $\mathcal{B}$ outputs $b^{'}$.

Finally, if $\mathcal{A}$ could break our encryption scheme with a non-negligible advantage $\mathcal{E}$, then $\mathcal{B}$ could break RW scheme with advantage $\mathcal{E}$. Assume that the assumption of $q - DPBDHE$

holds, our scheme is statically secure in the random oracle model. From article[10], we know that if the assumption of $q-DPBDHE$ holds, the scheme of RW is statically secure in the random oracle model. Finally, we can prove this theorem.