IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 23, 2019, accepted August 2, 2019, date of publication August 7, 2019, date of current version September 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2933636

Revocable Cloud-Assisted Attribute-Based Signcryption in Personal Health System

FUHU DENG, YALI WANG, LI PENG, MIAO LAI, AND JI GENG
School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

Corresponding author: Fuhu Deng (fuhu.deng@uestc.edu.cn)

This work was supported by the Sichuan Science and Technology Project under Grant 2018GZ0236, Grant 2017FZ0004, and Grant 2018KZ0007.

ABSTRACT PHR System is a favorable platform for personal health information exchange. In order to ensure that the personal information is not falsified and leaked by malicious users, we use the attribute-based signcryption technology to provide secure and reliable data protection. At the same time, in order to prevent users from accessing the data in the system by collusion of attributes, we proposes a revocable cloud-assisted attribute-based signcryption scheme which using the broadcast encryption technology and key segmentation technology realize user revocation function. Moreover, the proposed scheme is proven to be confidentiality and unforgeability under chosen plaintext attack in the random oracle model. And the experimental evaluation indicates that the proposed scheme is practical and feasible.

INDEX TERMS Cloud computing, attribute-based signcryption, verifiable outsourcing technology, user revocation function, server-assisted signature.

I. INTRODUCTION

With the development of medical information technology, Personal Health Record (PHR) system is gradually developing and improving. PHR system is a health record storage service system, which allows patients to create, control and share their HR data with a wide range of target users, including doctors, nurses, health insurance providers and family members. In order to improve the quality of PHR services at a lower cost, PHR service providers want to store PHR users' personal medical data on cloud servers. However, it will bring a series of security and privacy issues about patients' personal health information. For example, malicious users may obtain unauthorized data and modify it before authorized users (such as doctors) access it. This may lead to misdiagnosis of patients and wrong treatment of patients. Therefore, when sharing PHR data with other users, we should also ensure that PHR data is not forged or leaked. Therefore, how to securely share or store PHR data in a PHR system is an important problem.

To solve the above problems, it is essential to have robust cryptographic mechanism which is able to provide fine-grained data access control with confidentiality, authenticity and anonymity, simultaneously. At present, Attribute-Based Encryption (ABE) [1]–[3] is an encryption system with fine-grained access control recognized by the cryptography community. In addition, public key encryption technology has also developed greatly in the Industrial Internet of Things (IIoT) [4]–[8]. At the same time, Attribute-Based Signature (ABS) [9]–[12] has become an effective way to sign messages without revealing the identity of the signer. In recent years, many scholars have also studied the issue of the secure authentication protocol, such as Wang *et al.* [13] and Chen *et al.* [14] presented a new ultra-lightweight authentication protocol which are used in IIoT environment. Xiong *et al* presented an efficient and provably secure certificateless parallel key-insulated signature for IIoT environments [15]–[18]. In addition, Xiong *et al.* [19] introduces a scalable and forward secure privacy-preserving scheme in the cloud-assisted internet of things. Therefore, we use ABE and ABS combined Attribute-Based Signcryption (ABSC) [20]–[22] to provide safe and reliable guarantee for PHR data. In addition, due to the inevitability of users' right modification and key leakage, etc, attribute-based signcryption system should consider the issue of efficient user revocation [23].

In order to realize the function of user revocation while ensuring efficiency, we proposes a revocable cloud-assisted attribute-based signcryption scheme based on [24].

The associate editor coordinating the review of this article and approving it for publication was Chien-Ming Chen.

This scheme, using the attribute authority to update the key periodically, realize the function of user revocation for the first time in the attribute-based signcryption scheme. The scheme also uses key segmentation technology and the trusted third-party server to assist the signature, so as to reduce the computational overhead incurred by the user when performing the signcryption operation. Moreover, we demonstrate the security proof of confidentiality and unforgeability to ensure the security of the scheme, and show the feasibility of the scheme through further analysis of communication and computational overhead. And the specific system model and System schematic is illustrated in FIGURE. 1 and FIGURE. 2.

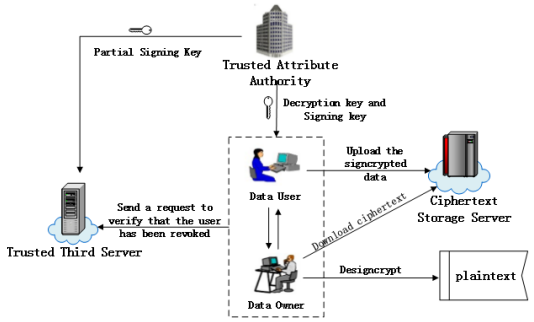


FIGURE 1. Personal health record system model.

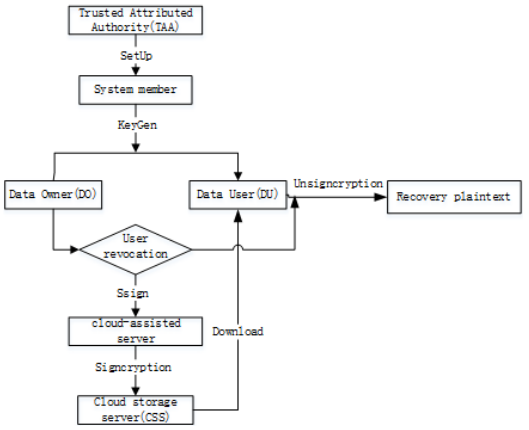


FIGURE 2. System schematic.

A. CONTRIBUTIONS

In this paper, In order to the attribute-based signcryption system can satisfy the needs of various application scenarios, we analysis the related research on the user revocation mechanism in the existing attribute-based cryptosystem scheme and apply it to the attribute-based signcryption system. We first present a novel revocable cloud-assisted signature attribute-based signcryption (RCS-ABSC) scheme in the cloud-based PHR system which is realized by using the broadcast encryption technology [25] and key splitting technology [26]. That is, the revoked users in the system can neither sign messages nor decrypt messages. The main contributions as follows:

1) The RCS-ABSC scheme is the first time to implement revocation function in attribute-based signcryption scheme. In particular, the communication and computing costs of our scheme are close to those of Rao's scheme [24], but compared with Rao's scheme [24], it adds user revocation function, which shows that our scheme is feasible.
2) In order to reduce the computation overhead of signcryption in user side, we use the trusted third-party server to assist signature.
3) We also demonstrate the security proof of confidentiality and unforgeability to ensure the security of the scheme, and shows the feasibility of the scheme through further analysis of communication and computational overhead.

B. ORGANIZATION

The rest of the paper is organized as follows : In section II, we introduced the preliminaries, and described the system model and security model of our RCS-ABSC scheme. In section III, we descirbed the specific algorithm of our scheme. Subsequently, the security proof and performance analysis of RCS-ABSC scheme are given in section IV and section V, respectively. In section VI, we mode a conclusion of this paper.

C. RELATED WORK

1) BROADCAST ENCRYPTION

The broadcast encryption schemes [27]–[31] allow the sender to specify the receiver group when encrypting. Readers may want to know if we can only use a public-key broadcast encryption system instead of ABE in the case when the sender knows the revocation list by simply specifying all non-revoked users as the receiver group. The answer is that we cannot, since we focus on the attribute-based setting, which means that the sender is supposed not to even know whose access policy will match the attribute set associated to ciphertext. In addition, Xiong et al [15], [32] proposed a partially policy-hidden attribute-based broadcast encryption scheme which can protect the privacy information of the data user well. For KP-ABE, a direct revocation method is, however, not possible yet for the normal present form of KP-ABE algorithm since a normal KP-ABE scheme allows just specifying attribute set associated to the ciphertext, not access policy. Golläe *et al.* [33] proposed a directly revocable KP-ABE which is heuristic and works only when the number of attributes associated to each ciphertext is exactly half of the universe size. On the other hand, for CP-ABE, such direct revocation can be done by using ABE that supports negative clauses, proposed by Ostrovsky *et al.* [34]. To do so, one just adds conjunctively the AND of negation of revoked user identities (where each is considered as an attribute here).

2) SERVER-AIDED TECHNOLOGY

"Server-aided computation" was first proposed by Matsumoto *et al.* [35] to speed up the computation, where a powerful server is applied to help the low-power devices execute

heavy cryptographic operations. As a desirable solution to reduce computational overheads, server-aided computation has been widely used in various schemes to help with heavy calculations in the algorithms including key computation, signature generation, signature verification, encryption, decryption, and so on [36]. For example, server-aided signature [37] was proposed to reduce the exponentiation and pairing calculations. In addition, in order to improve the efficiency of system, Xiong *et al.* [38] proposed an outsourced attribute-based encryption scheme, and the scheme has better verifiability and scalability. In addition to reduce the computational overheads, the server-aided technique has also been utilized for efficient user revocation (e.g., [26]), where a semi-trusted server immediately terminates partial decryption operations for revoked users. Compared to the traditional revocation methodology, such an approach does not require the private key of users to be updated regularly, and greatly simplifies the revocation operation, but it does not allow the server to collude with the users. Recently, Xiong et al [39] presents a server-aided attribute-based signature for industrial internet of things.

3) ATTRIBUTED-BASED SIGNCRYPTION

In 2010, Gagne *et al.* [20] proposed the first ABSC scheme which can support threshold access policy. However this scheme [20] has a restriction that the signing access structure of the signcryptor needs to be fixed in the setup phase. To solve this problem, in 2011, Emura *et al.* [9] gave the definition of dynamic ABSC, where the access structure of signcryptor can be updated flexible without re-issuing secret keys of users. In 2012, Chen *et al.* [40] investigated the combination of ABS and ABE, and give a general construction of combined ABSC schemes from combined ABE and ABS schemes. In 2013, Wang *et al.* [41] show that the threshold ABSC scheme in [20] is not secure and give a concrete forgery attack. Han *et al.* [42] propose a threshold ABSC scheme with constant-size ciphertext by employing Inner-product encryption. In 2015, Liu *et al.* [43] proposed an attribute-based signcryption scheme for PHR system, which has a expressive access structure, but does not provide public authentication. In 2017, Rao [24] proved that the security of scheme [43] was incorrect, and proposed a CP-ABSC scheme for PHR system. This scheme not only provides the function of public verification, but also guarantees the confidentiality, authenticity of the data and the privacy protection of the signer. However, due to the large number of bilinear pairings and exponential operations involved in the process of decryption, the efficiency of user decryption is greatly reduced.

II. PRELIMINARIES

A. BILINEAR MAP

Let $\mathcal{G}$ be a group of a prime order p with a generator g. A function $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is a bilinear map. If an algorithm that inputs a security parameter λ and outputs a tuple $(p, \mathbb{G}_1, \mathbb{G}_2, e)$ that has the following properties:

1) Bilinear, For all $x, y \in \mathbb{Z}_p$, we have $e(g^a, g^b) = e(g, g)^{ab}$;
2) Non-degenerate, For any generator g $\mathbb{G}$, $e(g, g)$ is a generator of $\mathbb{G}_2$ (i.e.,$e(g, g) \neq 1$).
3) Computability, for any $(g_1, g_2) \in \mathbb{G}_1^2$, there is an efficient algorithm to compute $e(g_1, g_2)$.

B. COMPLEXITY ASSUMPTION

Definition 1 (CDH Problem): The Computational Diffie-Hellman(CDH) problem is that, for every probabilistic polynomial time algorithm $\mathcal{A}$, there exists a negligible function $negl(\cdot)$ such that $Pr|\mathcal{A}(1^l, g, g^a, g^b) = g^{ab}| \leq negl(l)$ for all l, where $a, b \in_R \mathbb{Z}_p$, and g is the generator of a group $\mathbb{G}_1$ of order p, which is a prime of length approximately l. We say that $(t, \epsilon)-$CDH assumption holds in $\mathbb{G}_1$ if there is no adversary $\mathcal{A}$ that runs within time t and solves CDH problem with probability at least ϵ.

C. PREDICATES

Definition 2: We use A to be the universe of attributes. A predicate (over A) is a monotone boolean function whose inputs are related to the attributes of A. An attribute set $U \in A$ is said to satisfy a predicate Ω (or Ω accepts L) if $\Omega(U) = 1$. Here an input is set to be 1 (i.e., true) if its corresponding attribute is a member of U. Otherwise, the input is set to be 0 (i.e., false) if its corresponding attribute is not a member of U. If U doesn't satisfy Ω , we denote it by $\Omega(U) = 0$.

Since the predicate Ω is monotone, $\Omega(L) = 1$ indicates $\Omega(V) = 1$ for every attribute set $V \supset L$.

Assuming Ω is a predicate. L_Ω denotes the set of attributes utilized in Ω. Then the corresponding MSP for Ω is a labeled matrix $\Phi = (M_{l\times n}, \varphi)$, where $\varphi : [l] \to L_\Omega$ is a labeling of the rows of $\mathbf{M}$ by attributes from L_Ω.

We define $y_1 = \{i \in [l] : [\varphi(i) = u] \wedge [u \in U]\}$ and $y_0 = \{i \in [l] : [\varphi(i) = u] \wedge [u \notin U]\}$. Then $y_1 = \{i \in [l], \varphi(i) \in U\}$ and $y_0 = \{i \in [l], \varphi(i) \notin U\}$. On the other hand, $y_1 \cup y_0 = [l]$, where $U \subset A$ represents an attribute set.

A predicate Ω (with its $\Phi = (M_{l\times n}, \varphi)$) accepts an input attribute set U by the following criterion as stated in *Eq.* (4).

$$\begin{aligned} \Omega(U) &= 1 \Longleftrightarrow \Phi(U) = 1 \\ &\Longleftrightarrow [\exists(a_1, a_2, \ldots, a_\ell) \in \mathbb{Z}_p^l \\ \text{such that} \sum_{i\in[l]} a_i \cdot \vec{M}^{(i)} &= \vec{1}_n \text{ and } a_i = 0 \forall i \in y_0]. \end{aligned}$$

Hence,

$$\begin{aligned} \Omega(U) = 1 \Longleftrightarrow\ & [\exists(a_1, a_2, \ldots, a_l) \in \mathbb{Z}_p^l \\ \text{such that} \sum_{i\in[l]} a_i \cdot \vec{M}^{(i)} &= \vec{1}_n \\ \text{and } a_i = 0\ \forall i \text{ where } \varphi(i) \notin U]. & \end{aligned} \tag{1}$$

The following result is very useful to present the security proof of the CP-OABSC scheme which will be proposed in section 3.6.

Lemma 1: Let Ω, U be a predicate and attribute set, respectively. If $\Omega(U) = 0$, then there exists a vector

$\vec{u} = (u_1, u_2, \ldots, u_n) \in \mathbb{Z}_p^n$ with $u_1 = -1$ such that $\vec{u} \cdot \vec{M}^{(i)} = 0$ for all i where $\varphi(i) \in U$.

D. SYSTEM MODEL

1) **Setup** (PK, MK): The *Setup* algorithm is run by the Trusted Attribute Authority (TAA), which takes security parameter λ, attribute universe $\mathcal{U}$ as inputs. Then, it outputs the public parameters PK and a master secret key MK.
2) **sExtract**(id_s, PK, MK, A_s): This algorithm takes as inputs a user index set $id_s \in I$, the master key MK, the public parameters PK and an attributes set A_s. Then, it outputs the signing key SK_{id_s,A_s} for PHR owner and outputs partial signing key TK_{id_s,A_s} to the server.
3) **dExtract**(id_d, PK, MK, A_d): This algorithm takes as inputs a user index set $id_d \in I$, the master key MK, the public parameters PK and an attributes set A_d. Then, it outputs the decryption key SK_{id_d,A_d} for PHR owner.
4) **SSign**$(PK, TK_{id_s,A_s}, \chi_s, M)$:Taking the public parameter PK, the partial signing key TK_{id_s,A_s} of a user id_s, a signing predicate χ_s and a message M as the input, this algorithm outputs a partial signature S' on the message M. This algorithm is run by the server.
5) **USign**(PK, SK_{id_s,A_s}, S'):Taking the public parameters PK, the attribute-based signing key SK_{id_s,A_s} and a partial signature S' on a message M and a signing predicate χ_s as the input, this algorithm outputs a signature S. This algorithm is run by the singer.
6) **Signcryption** $(S, PK, M, SK_{id_s,A_s}, \chi_s, \chi_e)$: This is a randomized algorithm that takes a user index set $S \subseteq I$, the public parameters PK, a PHR file M, signer's attribute set A_s, signing key SK_{id_s,A_s}, signing predicate χ_s and encryption predicate χ_e as inputs. Only in the case of A_s satisfies χ_s where $\chi_s(A_s) = 1$, the signer can signcrypt the PHR file M. Finally, it will signcrypt a plaintext M and generate a ciphertext CT_{χ_e} such that only a PHR user who possesses a set of attributes A_d that satisfies χ_e will be able to unsigncrypt the corresponding ciphertext.
7) **Unsigncryption** $((id_s, id_d, A), M, PK, CT_{\chi_e}, \chi_s, SK_{id_d,A_d})$: This algorithm takes as inputs the public parameters PK, ciphertext CT_{χ_e}, signing predicate χ_s and a decryption key SK_{id_d,A_d} of a decryption attribute set A_d. It will correctly recover the message M only if $\chi_e(A_d) = 1$ and the ciphertext CT_{χ_e} contains a valid signature corresponding to the signing predicate χ_s. Otherwise, it outputs $\perp$, indicating that either the ciphertext is not valid or the ciphertext cannot be decrypted.

E. SECURITY MODEL

Confidentiality: Similar to [24], we use a security game to describe the confidentiality of message where $\mathcal{C}$ and $\mathcal{A}$ be a challenger and an adversary respectively.

1) *Setup*: $\mathcal{C}$ runs *Setup* algorithm to get the public parameters PK and a master key MK. Then it sends PK to $\mathcal{A}$ and keeps MK to itself.
2) *Query Phase* 1: $\mathcal{C}$ creates an empty table R and an empty set L. Then, $\mathcal{A}$ can adaptively issues the following queries:
   a) *dExtractoracle*: For each attribute set θ_d, the challenger $\mathcal{C}$ runs $dExtract(id_d, PK, MK, A_d) \rightarrow SK_{id_d,A_d}$ (where $\chi_e^*(A_d) = 0$) and sets $R = R \cup \{A_d\} \cup \{SK_{id_d,A_d}\}$. Then, it sends the decryption key SK_{id_d,A_d} to $\mathcal{A}$.
3) *Challenge Phase*: $\mathcal{A}$ submits two equal length messages m_0, m_1 and a decryption predicate χ_e^*. Note that none of attribute sets in R satisfy χ_e^*. Then, $\mathcal{C}$ chooses a random bit $b \in \{0, 1\}$. Finally, $\mathcal{C}$ signcrypts m_b under $\chi_e^*(A_d) = 1$ with the signcryption algorithm and sends it to $\mathcal{A}$.
4) *Query Phase* 2: After receiving $CT_{\chi_e}^*$, $\mathcal{A}$ can continue adaptively to issue queries in the same way as *Query Phase*1 except the *Designcrypt oracle*, for any attribute set A_d and A_s such that $\chi_e^*(A_d) = 1$ $(\chi_s^*(A_s) = 1)$.
5) *Guess*: The adversary $\mathcal{A}$ outputs a guess bit $b \in \{0, 1\}$ and wins the game if and only if $b' = b$. The advantage of $\mathcal{A}$ in this game is defined as $Adv(\mathcal{A}) = |Pr[b' = b] - \frac{1}{2}|$.

Definition 3: A RCS-ABSC scheme is CPA-secure if no polynomial time adversaries who possess a non-negligible advantage in the above security game.

Unforgeability: The formal definition of unforgeability is based on the following game involving a challenger $\mathcal{C}$ and an adversary $\mathcal{F}$.

1) *Setup*: $\mathcal{C}$ selects a security parameter $\lambda \in \mathbb{N}$ and runs *Setup* algorithm. It obtains the master key MK and sends the public parameters PK to $\mathcal{F}$.
2) *Queries*: Besides a table R and an empty list L, $\mathcal{F}$ adaptively issues the following queries:
   a) *dExtract oracle*: is identical to those in the CPA-secure game.
   b) *Signing key oracle*: Algorithm $\mathcal{F}$ issues a signging key query on an identity id_s and an attribute set A_s. If a tuple $(id_s, A_s, TK_{id_s,A_s}, SK_{id_s,A_s})$ exists in the list L, Algorithm $\mathcal{C}$ returns the corresponding signing key SK_{id_s,A_s}. Otherwise, Algorithm $\mathcal{C}$ runs the **sExtract** algorithm to generate $(TK_{id_s,A_s}, SK_{id_s,A_s})$, and returns the signing key SK_{id_s,A_s} to Algorithm $\mathcal{F}$. Also, Algorithm $\mathcal{B}$ adds $L = \{id, A_s, TK_{id_s,A_s}, SK_{id_s,A_s}, SK_{id_d,A_d}\}$ to the list R.
   c) *sExtract oracle*: For each θ_s, $\mathcal{C}$ runs $sExtract(id_s, PK, MK, A_s) \rightarrow SK_{id_s,A_s}$ (where $\chi_s^*(A_s) = 0$) and sets $R = R \cup \{SK_{id_s,A_s}\} \cup \{A_s\} \cup L$. Then, it sends the signing key SK_{id_s,A_s} to $\mathcal{F}$.
   d) *Signcryption oracle*: By taking $\bar{t} \in tt$, a sender's identity $\{id_d, id_s\} \in I$ and message M as input,

this oracle execute the *Signcryption* algorithm with the input $S, PK, M, SK_{id_s,A_s}, \chi_s, \chi_e$ to output the ciphertext, where SK_{id_s,A_s} can be extracted by running the *sExtract* algorithm.

3) *Forgey*: Finally, $\mathcal{F}$ outputs a ciphertext $CT^*_{\chi_e}$ associated with $(ID^*, \bar{t}^*)$. $\mathcal{F}$ wins this game if *Unsigncryption* $((ID^*, A^*), M^*, PK, CT_{\chi_e}{}^*, \chi_s{}^*, SK_{id_s,A_d}{}^*) = (\triangle^s)^*$ with the signing predicate $\chi_s{}^*$.

The advantage $Adv^{EUF}_{OABSC,F}(\lambda)$ of $\mathcal{F}$ is defined as the probability that it wins the game above.

Definition 4: A RCS-ABSC scheme is considered to be secure against existential unforgeability, if no PPT $\mathcal{F}$ can win the security game with a non-negligible advantage.

III. CONSTRUCTION

In this section, we present a revocable cloud-assisted attribute-based signcryption(RCS-ABSC) scheme. The function of user revocation is a combination of broadcast encryption technology [29] and key splitting technology [26].

In this construction, both the signing end encryption predicates are represented by MSPs. In MSP $\chi = (M, \rho)$, it is assumed that the row labeling function ρ is injective, that is, all the attributes associated with the rows of the matrix M are different. In addition, we adopt a one-time symmetric-key encryption scheme with key space $\phi = \{0,1\}^l$ and message space $M = \{0,1\}^*$ that can be defined as $\prod_{SE} = (\text{SE-Enc}, \text{SE-Dec})$. Here, we take the tuple $\exists = (p, \mathbb{G}_1, \mathbb{G}_2, e)$ as a bilinear group. The remaining algorithms are described as follows.

Setup (1^λ): The algorithm first picks a random generator $g \in \mathbb{G}$ and a random $\alpha \in \mathbb{Z}^*_p$. It computes $g_i = g^{(\alpha^i)} \in \mathbb{G}$ for $i = 1, 2, \ldots, n, n+2, \ldots, 2n$ and sets $\triangle = e(g,g)^\alpha$. Next, it randomly picks $\beta \in \mathbb{Z}^*_p$ and sets $v = g^\beta \in \mathbb{G}$. It then chooses three cryptographic collision resistent hash functions $H_1 : \{0,1\}^* \to \{0,1\}^l$, $H_2 : G \to \mathbb{Z}^*_p$ and $H_3 : \{0,1\}^* \to \mathbb{Z}^*_p$.

1) For each attribute $x \in A$, it randomly samples $h_x \leftarrow \mathbb{G}$.
2) Besides, it randomly selects $\theta, Y_1, Y_2, y_0, y_1, \ldots, y_l \leftarrow \mathbb{G}$.
3) The system public parameters PK and master key MK are given by

$$\begin{aligned} PK = (&\exists, \triangle, \theta, Y_1, Y_2, y_0, y_1, \ldots, \{y_i\}_{i\in[l]}, \{h_x\}_{x\in U}, \\ &H_1, H_2, H_3, \Pi_{SE}, \mathsf{KDF}, M, U, g, g_1, \ldots, g_n, \\ &g_{n+2}, \ldots, g_{2n}, v) \\ MK = (&\alpha, \beta). \end{aligned}$$

sExtract (id_s, PK, MK, A_s): On input the public parameter PK, the master key MK and a data owner index $id_s \in \mathcal{U}_s$ and an attribute set $A_s \in A$. It randomly samples $t_s, t_u, b \in \mathbb{Z}^*_p$ where $T = t_s + t_u$, and computes

$$\begin{aligned} K_s &= g^{\beta\alpha^{id_s} - b} \cdot \theta^{t_s}, \quad K'_s = g^{t_s}, \\ K_{s,x} &= h_x^{t_s}, \quad \forall x \in A_s \\ K_u &= g^{\alpha\cdot b} \cdot \theta^{t_u}, \quad K'_u = g^{t_u}, \\ K_{u,x} &= h_x^{t_u}, \quad \forall x \in A_s \end{aligned}$$

It sends $TK_{id_s,A_s} = (A_s, K_s, K'_s, K_{s,x_{x\in A_s}})$ as the partial signing key for the data owner id_s to the server, and the signing key $SK_{id_s,A_s} = (A_s, K_u, K'_u, K_{u,x_{x\in A_s}})$ to the data owner id_s.

dExtract (id_d, PK, MK, A_d): On input the public parameter PK, the master key MK and a data user index $id_d \in \mathcal{U}_d$ and an attribute set $A_d \in A$. It randomly samples $r \in \mathbb{Z}^*_p$ and computes

$$\begin{aligned} K_d &= g^{\alpha^{id_d}\beta} \cdot \theta^r, \quad K'_d = g^r, \\ K_{d,x} &= h_x^r, \quad x \in A_d \end{aligned}$$

It sends $SK_{id_d,A_d} = (A_d, K_d, K'_d, K_{d,x_{x\in A_d}})$ as the decryption key to the data user id_d.

SSign$(PK, M, TK_{id_s,A_s}, \chi_s)$: On input the public parameter PK, the partial signing key TK_{id_s,A_s} and an signging predicate χ_s with the property that $\chi_s(A_s) = 1$. Here, $\chi_s = (\vec{M}_s, \rho_s)$, where $\vec{M}_s$ is an $l_s \times n_s$ matrix with row labeling function $\rho_s : [l_s] \to U$. Let $\vec{M}^{(i)}_s$ be the ith row of the matrix $\vec{M}_s$.

1) Since $\chi_s(A_s) = 1$, this algorithm computes a vector $\vec{a} = (a_1, a_2, \ldots, a_{l_s}) \in \mathbb{Z}^{l_s}_p$ such that $\vec{a} \cdot \vec{M}^{(i)}_s = \vec{1}_{n_s}$, that is, $\sum_{i\in[l_s]} a_i \vec{M}^{(i)}_s = \vec{1}_{n_s}$, and $a_i = 0$ for all i where $\rho_s(i) \notin A_s$.
2) It randomly chooses a vector $(b_1, b_2, \ldots, b_{l_s}) \leftarrow \mathbb{Z}^{l_s}_p$ such that $\sum_{i\in[l_s]} b_i \cdot \vec{M}^{(i)}_s = \vec{0}_{n_s}$.
3) It samples $t' \leftarrow Z^*_p$ and re-randomizes the signing key TK_{id_s,A_s} as follows:

$$\begin{aligned} \check{K}_s &= K_s \cdot \theta^{t'}, \quad \check{K}_s{}' = K'_s \cdot g^{t'}, \\ \check{K}_{s,x} &= K_{s,x} \cdot h_x^{t'}, \quad \forall x \in A_s. \end{aligned}$$

4) It randomly chooses a vector $(b_1, b_2, \ldots, b_{l_s}) \leftarrow \mathbb{Z}^{l_s}_p$ such that $\sum_{i\in[l_s]} b_i \cdot \vec{M}^{(i)}_s = \vec{0}_{n_s}$ and computes

$$\begin{aligned} S'_2 &= \{S_2^{(i)} = g^{b_i}(\check{K}'_s)^{a_i}\}_{i\in[l_e]}, \\ S'_3 &= \check{K}_s \cdot (\prod_{i\in[l_s]} (\check{K}_{s,\rho_s(i)})^{a_i} \cdot h^{b_i}_{\rho_s(i)}). \end{aligned}$$

It outputs the partial signature $S' = (\chi_s, M, S'_2, S'_3)$.

USign$(PK, M, SK_{id_s,A_s}, \chi_s, S'_2, S'_3)$: On input the public parameter PK, the signing key SK_{id_s,A_s} and a partial signature σ' under a claim predicate χ_s on a message m, this algorithm runs as follows.

1) It samples $t' \leftarrow Z^*_p$ and re-randomizes the signing key TK_{id_s,A_s} as follows:

$$\begin{aligned} \check{K}_u &= K_u \cdot \theta^{t'}, \quad \check{K}_u{}' = K'_u \cdot g^{t'}, \\ \check{K}_{u,x} &= K_{u,x} \cdot h_x^{t'}, \quad \forall x \in A_s. \end{aligned}$$

2) It randomly chooses a vector $(b_1, b_2, \ldots, b_{l_s}) \leftarrow \mathbb{Z}^{l_s}_p$ such that $\sum_{i\in[l_s]} b_i \cdot \vec{M}^{(i)}_s = \vec{0}_{n_s}$ and computes

$$\begin{aligned} \tilde{S}_2 &= S'_2 \cdot g^{b_i}(\check{K}'_u)^{a_i}, \\ \tilde{S}_3 &= S'_3 \cdot \check{K}_u(\prod_{i\in[l_s]} (\check{K}_{u,\rho_s(i)})^{a_i} \cdot h^{b_i}_{\rho_s(i)}). \end{aligned}$$

Finally, it outputs the signature $\tilde{S} = (\chi_e, M, \tilde{S}_2, \tilde{S}_3)$.

Signcryption $(S, PK, M, SK_{id_s,A_s}, \tilde{S}, \chi_e)$: Inputs a user index set $S \subseteq \{\mathcal{U}_s \cup \mathcal{U}_d\}$ and an signging predicate χ_s with the property that $\chi_e(A_e) = 1$. Here, $\chi_e = (\vec{M}_e, \rho_e)$, where $\vec{M}_e$ is an $l_e \times n_e$ matrix with row labeling function $\rho_e : [l_e] \to U$. Let $\vec{M}_e^{(i)}$ be the ith row of the matrix $\vec{M}_e$. When a data owner is intended to signcrypt a message m, he/she first needs to send a request to the server to query partial signing key TK_{id_s,A_s}. The serve would update transformation signing keys periodically. If the server finds that there is no id_s of the person in the revocation list $\mathcal{U}_s$, it will send the partial signing key to the data owner. Then, the data owner will use

In addition, the data owner encryption is done with the symmetric-key encryption algorithm (such as AES), SE-Encryption using the key derivation function KDF.

1) It picks $\delta \leftarrow \mathbb{Z}_p^*$ and sets $\vec{\lambda} = (s, \varphi_2, \ldots, \varphi_{n_e})$, here $\varphi_2, \ldots, \varphi_{n_e} \leftarrow \mathbb{Z}_p^*$.
2) It now computes the following terms

$$
\begin{aligned}
E_0 &= (v \cdot \prod_{j \in S} g_{n+1-j})^s, \\
E_1 &= g^s, S_1 = g^{s \cdot \delta}, \\
E_2 &= \text{SE-Encrypt}(\text{KDF} \\
&\quad (\triangle^s || S_1 || tt), M), \\
E_3 &= \{E_3^{(i)} = \theta^{\vec{\lambda} \cdot \vec{M}_e^{(i)}} \cdot h_{\rho_{e(i)}}^s\}_{i \in [l_e]}, \\
\mu &= H_2(E_1), E_4 = (Y_1 Y_2^{\mu})^s, \\
H_1(S_2, tt, \chi_s, \chi_e) &= (K_1, \ldots, K_l) \in \{0,1\}^l, \\
H_3(S_1, E_2, E_3, E_4, \chi_s, \chi_e) &= \eta, \\
S_3 &= (y_0 \prod_{i \in [l]} y_i^{k_i})^s E_4^{\eta \cdot \delta}.
\end{aligned}
$$

The ciphertext is $CT_{\chi_e} = (\chi_e, E_0, E_1, E_2, E_3, E_4, S_1, S_2, S_3, tt)$.

Now, the PHR owner uploads the data file $\mathcal{D} := (CT_{\chi_e}, \chi_s)$ to the cloud server.

Revoke$_{sign}$($id_s, \mathcal{U}_s$): On input an identity id_s, this algorithm adds id_s to the revocation list $\mathcal{U}_s$. For any user id_s in the list $\mathcal{U}_s$, the server immediately terminates sending transformed signing key for this user id_s.

Unsigncryption((id_d, id_s), A), PK, CT_{χ_e}, χ_s, SK_{id_d,A_d}): When a PHR user intends to access a PHR file, he/she sends the request to the cloud server. The server sends back the corresponding data file of the form $\mathcal{D} := (CT_{\chi_e}, \chi_s)$ to the user using SSH protocol. If the PHR user has the privilege to access the file, he/she can decrypt and view the message m as described subsequently.

The PHR user executes this algorithm with the input. The algorithm first checks the current time $\tilde{tt}$. Assume that ϖ is a predefined time limit for message decryption. If $\tilde{tt} - tt > \varpi$ or $\chi_e(A_d) \neq 1$, it returns $\perp$. Otherwise (that is, $\tilde{tt} - tt \leq \varpi$ and $\chi_e(A_d) = 1$, it verifies the ciphertext and decrypts the message. Note that $\chi_s = (M_s, \rho_s)$ and $\chi_e = (M_e, \rho_e)$, where M_s (resp.M_e) is an $l_s \times n_s$ (resp.$l_e \times n_e$) matrix. Let $\vec{M}_s^{(i)}$ (resp.$\vec{M}_e^{(i)}$) be the ith row of the matrix M_s (resp.M_e).

1) It randomly samples $\varrho_2, \ldots, \varrho_{n_s} \leftarrow \mathbb{Z}_p^*$ and computes $\vec{\omega} = (1, \varrho_2, \ldots, \varrho_{n_s}) \cdot \vec{M}_s^{(i)}, \forall i \in [l_s]$.
2) It next computes $\mu = H_2(E_1)$, $H_1(S_2, tt, \chi_s, \chi_e) = (K_1, \ldots, K_l) \in \{0,1\}^l$, $H_3(S_1, E_2, E_3, E_4, \chi_s, \chi_e) = \eta$.
3) It then proceeds in the following way in order to recover the PHR.
   a) Check the validity of the ciphertext CT_{χ_e} using the equation

$$
\begin{aligned}
e(g,g)^{\alpha^{id_s}\beta} \stackrel{?}{=} & \frac{e(S_3, g)}{(\prod_{i \in [l_s]} e(\theta^{\vec{\omega}_i} \cdot h_{\rho_s(i)}, S_2^{(i)}))} \\
& \times \frac{e((Y_1 Y_2^{\mu})^{\eta}, S_1)^{-1}}{e(y_0 \prod_{i \in [l]} y_i^{k_i}, E_1)}
\end{aligned}
$$

   if it is invalid, return $\perp$, otherwise, proceed as follows.
   b) Because $\chi_e(A_d) = 1$, compute a vector $\vec{a}' = (a'_1, a'_2, \ldots, a'_{l_s}) \in \mathbb{Z}_p^{l_e}$ such that $\vec{a}' \cdot M_e = \vec{1}_{n_e}$, that is, $\sum_{i \in [l_e]} a'_i \cdot \vec{M}_e^{(i)} = \vec{1}_{n_e}$. and $\vec{a}'_i = 0$ for all i where $\rho_e(i) \notin A_d$.
   c) Recover $\triangle^s$ from the following computation

$$
\frac{e(g_{id_d}, E_0)}{e(\prod_{j \in S, j \neq id_d} g_{n+1-j+id_d}, E_1)} \cdot \frac{e(K'_d, \prod_{i \in [l_e]} (E_3^{(i)})^{a'_i})}{e(K_d \cdot \prod_{i \in [l_e]} K_{d,\rho_e(i)}^{a'_i}, E_1)} = e(g,g)^{\alpha^{id_d} \cdot s}.
$$

   Finally, obtain the correct PHR $M = \text{SE-Decrypt}(\text{KDF}(\triangle^s || S_1 || tt), E_2)$.

Revoke$_{dec}$: On input an identity id_d, this algorithm adds id_d to the revocation list $\mathcal{U}_d$. For any user id_d in the list $\mathcal{U}_d$, the KGC immediately terminates sending transformed signing key for this user id_d.

Remark 1: The PHR user may not have the decryption key components $K_{d,\rho_e(i)}^{a'_i}$ for every attribute $\rho_e(i)$ in the computation of $\prod_{i \in [l_e]} K_{d,\rho_e(i)}^{a'_i}$. But, $a'_i = 0$ for each attribute $\rho_e(i) \notin A_d$ (A_d is the attribute set associated with the decryption key) and hence the PHR user is able to compute Eq.(5) without knowing the values of $K_{d,\rho_e(i)}$ for $\rho_e(i) \notin A_d$.

Remark 2 (Public Verifiability): The validity of a signcrypted ciphertext can be verified using the identity stated in Eq.(4) based only on the system public parameters and the ciphertext terms. Hence any user who has access to the ciphertext can verify the integrity and validity of the sender and the ciphertext. This makes our scheme publicly verifiable signcryption scheme.

Using this test, either cloud or PHR user can detect whether the ciphertext has modified or not during transmission and hence authenticity of ciphertext is achieved. This is one of the essential security goals of the attribute-based PHR sharing system.

A. CORRECTNESS

If $|\tilde{tt} - tt| \leq \varpi$ and $\chi_e(A_d) = 1$, the PHR user can verify the ciphertext and recover the message correctly as explained

subsequently. We have that $\sum_{i\in[l_s]} a_i \cdot \vec{M}_s^{(i)} = \vec{1}_{n_s}$ and $\sum_{i\in[l_s]} b_i \cdot \vec{M}_s^{(i)} = \vec{0}_{n_s}$. Also, $\sum_{i\in[l_e]} a'_i \cdot \vec{M}_e^{(i)} = \vec{1}_{n_e}$.

$$\begin{aligned} S_3 &= \check{K}_s(\prod_{i\in[l_s]}(\check{K}_{s,\rho_s(i)})^{a_i}\cdot h_{\rho_s(i)}^{b_i})(y_0\prod_{i\in[l]} y_i^{k_i})^s E_4^{\eta\cdot\delta} \\ &= g^{\alpha^{id_s}\beta}\cdot\theta^T(\prod_{i\in[l_s]} h_{\rho_s(i)}^{T\cdot a_i+b_i})(y_0\prod_{i\in[l]} y_i^{k_i})^s\cdot E_4^{\eta\cdot\delta} \end{aligned}$$

where $T = t + t'$

$$\begin{aligned} S_2^{(i)} &= g^{b_i}(\check{K}'_s)^{a_i} = g^{b_i+T\cdot a_i} \\ &\quad \sum_{i\in[l_s]}(T\cdot a_i+b_i)\cdot\vec{\omega}_i \\ &= \sum_{i\in[l_s]}(T\cdot a_i+b_i)\cdot((1,\varrho_2,\ldots,\varrho_{n_s})\cdot\vec{M}_s^{(i)}) \\ &= (T, T\varrho_2,\ldots,T\varrho_{n_s})\cdot\sum_{i\in[l_s]} a_i\cdot\vec{M}_s^{(i)} + (1,\varrho_2,\ldots,\varrho_{n_s}) \\ &\quad \times\sum_{i\in[l_s]} b_i\cdot\vec{M}_s^{(i)} \\ &= (T, T\varrho_2,\ldots,T\varrho_{n_s})\cdot(1,0,\ldots,0) \\ &\quad +(1,\varrho_2,\ldots,\varrho_{n_s})\cdot(0,0,\ldots,0) \\ &= T \end{aligned}$$

$$\frac{e(S_3,g)}{(\prod_{i\in[l_s]} e(\theta^{\vec{\omega}_i}\cdot h_{\rho_s(i)}, S_2^{(i)}))\cdot e(y_0\prod_{i\in[l]} y_i^{k_i}, E_1)\, e((Y_1Y_2^{\mu})^{\eta}, S_1)}$$

$$\begin{aligned} &= \frac{e(g,g)^{\alpha^{id_s}\beta}\cdot e(\theta,g)^T\cdot e(\prod_{i\in[l_s]} h_{\rho_s(i)}^{T\cdot a_i+b_i}, g)}{e(\theta,g)^{\sum_{i\in[l_s]}\vec{\omega}\cdot(b_i+T\cdot a_i)}\cdot e(h_{\rho_s(i)},g)^{\sum_{i\in[l_s]} T\cdot a_i+b_i}} \\ &\quad\times\frac{e((y_0\prod_{i\in[l]} y_i^{k_i})^s, g)\cdot e(E_4^{\eta\cdot\delta}, g)}{e(y_0\prod_{i\in[l]} y_i^{k_i}, g^s)\cdot e((Y_1Y_2^{\mu})^{\eta}, g^{s\cdot\delta})} \\ &= e(g,g)^{\alpha^{id_s}\beta} \end{aligned}$$

This exhibits the correctness of Eq. (4). The following argument establishes the correctness of Eq. (5). $\chi_e(A_d) = 1$ implies

$$\begin{aligned} &\sum_{i\in[l_e]} a'_i\cdot(\vec{\lambda}\cdot\vec{M}_e^{(i)}) \\ &= \vec{\lambda}\cdot\sum_{i\in[l_e]} a'_i\cdot\vec{M}_e^{(i)} \\ &= \vec{\lambda}\cdot\vec{1}_{n_e} = (s,\varphi_2,\ldots,\varphi_{n_e})\cdot(1,0,\ldots,0) \\ &= s\frac{e(g_{id_s},E_0)}{e(\prod_{j\in S, j\neq id_s} g_{n+1-j+id_s}, E_1)}\cdot\frac{e(K'_d,\prod_{i\in[l_e]}(E_3^{(i)})^{a'_i})}{e(K_d\cdot\prod_{i\in[l_e]} K_{d,\rho_e(i)}^{a'_i}, E_1)} \\ &= \frac{e(g^{\alpha^{id_d}}, v^s)e(g,\prod_{j\in s} g_{n+1-j+id_d})}{e(\prod_{j\in s, j\neq id_d} g_{n+1-j+id_d}, g^s)}\cdot\frac{e(g^r,\theta^{\sum_{i\in[l_e]} a'_i\cdot(\vec{M}_e^{(i)}\cdot\vec{\lambda})})}{e(g^{\alpha^{id_d}\beta}\cdot\theta^r, g^s)} \\ &\quad\times\frac{e(g^r, h_{\rho_e(i)}^{\sum_{i\in[l_e]} s\cdot a'_i})}{e(h_{\rho_e(i)}^{\sum_{i\in[l_e]} r\cdot a'_i}, g^s)} \\ &= \frac{e(g^{\alpha^{id_d}}, g^{\beta\cdot s})\cdot e(g, g^{\alpha^{id_d}\cdot s})}{e(\theta^r, g^s)}\cdot\frac{e(g^r,\theta^s)}{e(g^{\alpha^{id_d}\beta}, g^s)} \\ &= e(g,g)^{\alpha^{id_d}\cdot s} \end{aligned}$$

Finally, the computation SE-Decrypt(KDF($\triangle^s||S_1||tt$), E_2) yields the correct message M.

IV. SECURITY PROOF

Theorem 1 (Confidentiality): Suppose the security of Rao's scheme in [24] is guaranteed, then the proposed scheme is secure.

Proof 1: Assume an adversary $\mathcal{A}$ with non-negligible advantage can attack the above RCS-ABSC scheme. Similarly, the scheme in [24] can also be attacked by an algorithm $\mathcal{S}$ with non-negligible advantage.

Let $\mathcal{C}$ be the challenger associated with $\mathcal{S}$ in the selectively CPA-secure game of Rao's scheme in [24]. $\mathcal{S}$ runs $\mathcal{A}$ to execute the following steps.

1) **Setup**: $\mathcal{C}$ executes *Setup* algorithm in [24] to get the public parameters

$$\begin{aligned} PK' = (&\Sigma, \Delta, \vartheta, \gamma_1, \gamma_2, y_0, \{y_i\}_{i\in[\ell]}, \\ &\{h_x\}_{x\in U}, H_2, H_3, H_4, \Pi_{SE}, \mathsf{KDF}, M, U) \end{aligned}$$

and sends it to $\mathcal{S}$. Then, $\mathcal{S}$ runs *Setup* algorithm in this paper to get the public parameters

$$\begin{aligned} PK = (&\exists, \Delta, \theta, Y_1, Y_2, y_0, y_1, \ldots, \{y_i\}_{i\in[l]}, \{h_x\}_{x\in U}, \\ &H_1, H_2, H_3, \Pi_{SE}, \mathsf{KDF}, M, U, g, g_1, \ldots, g_n, \\ &g_{n+2}, \ldots, g_{2n}, v) \end{aligned}$$

Finally, it gives PK to $\mathcal{A}$.

2) **Query Phase 1**: Firstly, $\mathcal{S}$ initializes an empty table R and an empty list L. Then, $\mathcal{A}$ adaptively issues the following queries:
   a) *dExtract oracle*: If $\mathcal{A}$ makes a decryption key query for a set of attributes A_d. $\mathcal{S}$ sends A_d to $\mathcal{C}$ and obtains the decryption key SK_{id_d,A_d}. Then SK_{id_d,A_d} can be another type of $SK^*_{id_d,A_d}$. So, in the view of the data owner, it is similar between cloud computing center and users. Finally, $\mathcal{S}$ will set $R = R\cup\{A_d\}\cup\{SK_{id_d,A_d}\}$ and send SK_{id_d,A_d} to $\mathcal{A}$.
3) *Challenge Phase*: $\mathcal{A}$ sends $\mathcal{S}$ the challenge access policy χ_e^*, then $\mathcal{S}$ picks two (equal length) messages m_0, m_1 and sends them to $\mathcal{C}$ to obtain a challenge ciphertext $SCT'_{\chi_e} = (\chi_e, E_0, E_1, E_2, E_3, E_4, S_1, S_2, S_3, tt)$ by running *Signcryption* algorithm of [24]. Finally, $\mathcal{S}$ sends SCT'_{χ_e} to $\mathcal{A}$ as its challenge ciphertext.
4) *Query Phase*2: $\mathcal{A}$ requests a second series of queries, $\mathcal{S}$ answers these queries in the same way as it simulated in *Query Phase* 1, and returns the answer as *Query Phase* 1.
5) *Guess*: $\mathcal{A}$ outputs its guess b. $\mathcal{S}$ also outputs b.

TABLE 1. Comparison of communication overheads.

Scheme	Signing Key Size	Decryption key Size	Ciphertext Size
[24]	$(\|r_s\|+2)L_{\mathbb{G}_1}$	$(\|r_d\|+2)L_{\mathbb{G}_1}$	$(l_s+l_e+4)L_{\mathbb{G}_1}$
[44]+ [45]	$(\|r_s\|+2)L_{\mathbb{G}_1}$	$(\|r_d\|+8)L_{\mathbb{G}_1}$	$(2l_s+2l_e+4)L_{\mathbb{G}_1}$
[26]	$(2\|r_s\|+4)L_{\mathbb{G}_1}$	$-$	$(2l_s+2)L_{\mathbb{G}_1}$
[25]	$-$	$(\|r_d\|+2)L_{\mathbb{G}_1}$	$l_e\cdot p+(2+l_e)L_{\mathbb{G}_1}$
our scheme	$(\|r_s\|+2)L_{\mathbb{G}_1}$	$(\|r_d\|+2)L_{\mathbb{G}_1}$	$(l_s+2l_e)L_{\mathbb{G}_1}$

‡ $L_{\mathbb{G}_1}$ and $L_{\mathbb{G}_2}$ denote the length of an element in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. $l_s(l_e)$, $|r_s|(|r_d|)$ indicate the number of attributes in a signing (encryption) predicate, the number of signing (decryption) key attributes, respectively. Here we assume the signcryption system supports up to 20 attributes, that is, $N = 20$.

According to the above discussion, if $\mathcal{A}$ can attack our RCS-ABSC scheme in the selectively CPA-secure model with non-negligible advantage. Similarly, $\mathcal{A}$ can attack the scheme in [24].

Theorem 2 (Unforgeability): The RCS-ABSC scheme in this paper is unforgeable under the assumption of CDH.

Proof 2: Suppose an adversary $\mathcal{F}$ can break the scheme of this paper with non-negligible advantage, then an algorithm $\mathcal{B}$ can be built to solve the CDH problem. Given $\{g, g^x, g^y\}$ as a random CDH instance, the purpose of $\mathcal{B}$ is to output g^{xy} such that x, y are selected from $\mathbb{Z}_p^*$ at random.

1) *Setup*: The simulator $\mathcal{B}$ sets $(g_1 = E_1 = g^x, g_2 = y_0 = g^y)$ and delivers (g, g_1, g_2) to $\mathcal{F}$ as the public key.
2) *Queries*: Besides a table T and an empty list L, $\mathcal{B}$ adaptively issues the following queries:
   a) *dExtract oracle* is identical to those in the above CPA-secure game.
   b) *Partial signing key oracle*:For a partial-signing-key generation query on an identity id_s with an attribute set A_s from Algorithm $\mathcal{F}$, Algorithm $\mathcal{B}$ checks whether there exits a tuple $(id_s, A_s, TK_{id_s,A_s}, SK_{id_s,A_s})$ in the list L. If so, it returns the partial-signing-key TK_{id_s,A_s}. Otherwise, it randomly chooses $b \in \mathbb{Z}_p^*$, and generates the partial-signing-key $TK_{id_s,A_s} = (A_s, K_s, K_s', K_{s,x_{x\in A_s}})$ as required.
   c) *sExtract oracle*: If $\mathcal{F}$ makes a signing key query for a set of attributes A_s. $\mathcal{B}$ sends A_s to $\mathcal{C}$ and obtains the signing key SK_{id_s,A_s}. Then, $\mathcal{B}$ sets $R = R \cup \{SK_{id_s,A_s}\} \cup \{A_s\} \cup L$, $L = \{id_s, A_s, TK_{id_s,A_s}, SK_{id_s,A_s}, *\}$ and gives SK_{id_s,A_s} to $\mathcal{F}$.
   d) *Signcryption Query*: When $\mathcal{F}$ queries the plaintext M associated with identity ID within time period $\bar{t}$, $\mathcal{B}$ can answer this query by executing the *Signcryption* algorithm in case $ID \neq ID'$. If it satisfied $ID = ID'$, $\mathcal{B}$ simply aborts.
3) *Forgey Phase*: During this phase, $\mathcal{F}$ outputs a forged ciphertext $CT_{\chi_e}^* = (\chi_e^*, E_0^*, E_1^*, E_2^*, E_3^*, E_4^*, S_1^*, S_2^*, S_3^*, tt^*)$ on M^* under the identity ID^*. If $ID^* \neq ID'$, $\mathcal{B}$ aborts. Otherwise, the submitted $CT_{\chi_e}^*$ is valid Since

$$\frac{e(S_3^*, g)\cdot e((Y_1Y_2^{\mu})^{\eta}, S_1^*)^{-1}}{(\prod\limits_{i\in[l_s]} e(\theta^{\vec{\omega}_i}\cdot h_{\rho_{s(i)}}, S_2^{(i)*}))\cdot e(y_0\prod\limits_{i\in[l]} y_i^{k_i}, E_1^*)} = (\triangle^{\beta})^*$$

It is obvious that

$$\frac{e(S_3^*, g)\cdot e((Y_1Y_2^{\mu})^{\eta}, S_1^*)^{-1}}{(\prod\limits_{i\in[l_s]} e(\theta^{\vec{\omega}_i}\cdot h_{\rho_{s(i)}}, S_2^{(i)*}))\cdot e(g^y\prod\limits_{i\in[l]} y_i^{k_i}, g^x)} = (\triangle^{\beta})^*$$

According to our setting,

$$g^{xy} = \frac{S_3^*}{\prod\limits_{i\in[l_s]}(\theta^{\vec{\omega}_i}\cdot h_{\rho_{s(i)}})^{b_i+T\cdot a_i}\cdot\prod\limits_{i\in[l]} y_i^{k_i}\cdot(Y_1Y_2^{\mu})^{\eta\cdot s\cdot\delta}\cdot(\triangle^{\beta})^*}$$

can be calculated as the solution of the given CDH instance.

V. PERFORMANCE ANALYSIS

A. THEORETICAL COMPARISON

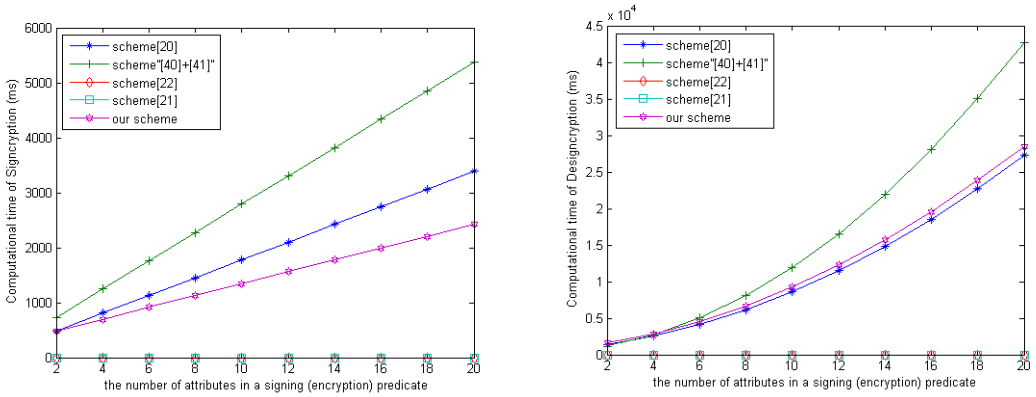
In this section we theoretically evaluate the performance of the proposed RCS-ABSC scheme in terms of communication cost and computation complexity as well as functionality and access policy. Since the RCS-ABSC scheme realizes the user revocation function based on the scheme [24], the scheme focuses on comparison with the scheme [24]. In addition, since the RCS-ABSC scheme is the first time to implement revocation function in attribute-based signcryption scheme, the selected scheme "[44]+[45]" is compared with our scheme. At the same time, since the user revocation function in the RCS-ABSC scheme is implemented by using the broadcast encryption technology [25] and the key splitting technology [26]. Therefore, the scheme [25], [26] is compared with our scheme. The comparison results are shown in TABLE 1 and TABLE 2.

As shown in TABLE 1, the schemes are compared and analyzed in terms of function and computational overhead. Firstly, from the perspective of the expressiveness of the access policy, both the RCS-ABSC scheme and scheme [24] use MSP as the access structure, the scheme [26] uses the access structure of the threshold strategy, and the scheme "[44]+[45]" [25] use LSSS as the access structure. Compared with the access structure of the threshold policy,

TABLE 2. Comparison of computation overheads.

Scheme	$Accesspolicy$	$Revocable function$	$Signc.$	$Designc.$
[24]	MSP	$\times$	$(3+2l_s+l_e)E_{\mathbb{G}}$	$(l_e^2+2l_e+l_s+3)E_{\mathbb{G}}+(5+l_s)p$
[44]+ [45]	LSSS	$\surd$	$l_e\cdot p+(4+2l_s+l_e)E_{\mathbb{G}}$	$(2l_e+2+l_s^2)p+4E_{\mathbb{G}}$
[26]	Threshold	$\surd$	–	–
[25]	LSSS	$\surd$	–	–
our scheme	MSP	$\surd$	$(5+l_s+l_e)E_{\mathbb{G}}$	$(l_e^2+3l_e+l_s)E_{\mathbb{G}}+(8+l_s)p$

‡ P represents a pairing computation, E_{G_1} and E_{G_2} represents a modular exponentiation computation in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Signc. and Designc. represents the signcryption and designcryption computational cost.



(a) Computational time of Signcryption

(b) Computational time of Designcryption

FIGURE 3. Performance evaluation comparison.

MSP and LSSS are more flexible and rich in the expression of access policies, so they have higher practicability. Secondly, from the perspective of revocation function, the scheme [24] does not provide the revocation function, the scheme [26]provides an indirect revocation function for the user's signature operation, and the scheme [25] provides a direct revocation function for the user's encryption operation, and the combination scheme " [44]+ [45]"implements the user's direct revocation function, and the RCS-ABSC scheme implements the revocation function for the user through the hybrid revocation method. Since the key update is very likely to become a bottleneck of system performance in the case of a large number of users, the direct revocation can avoid the communication cost and computational burden caused by the key update algorithm. Therefore, the realization of revocation function in the RCS-ABSC scheme is a compromise way scheme. Lastly, comparing the computational overhead, since signcryption and designcryption operations are not involved in the scheme [25], [26], no comparison is made here. From the data in the table, it can be analyzed that the calculation cost of the signcryption and designcryption of the RCS-ABSC scheme is slightly lower than the scheme [24], but much lower than the combination scheme " [44]+ [45]". Therefore, the RCS-ABSC scheme is not only functionally superior to most of the existing attribute-based signcryption schemes, but also close to the existing work in terms of efficiency.

As shown in TABLE 2, the RCS-ABSC scheme and the scheme [24]" [44]+ [45]" [25], [26] are analyzed in terms of communication overhead. From the data in the table, we can see that the signing key and decryption key of the our scheme are almost the same as that of the scheme [24], [26], and much smaller than that of the scheme " [44]+ [45]". In addition, since the scheme periodically updates the signature key by means of a trusted third party server. Therefore, it will inevitably lead to an increase in its communication overhead and computational overhead. In summary, compared with most existing attribute-based signcryption schemes, the RCS-ABSC scheme not only has a expressive access structure, but also provides user revocation function. At the same time, its computational and communication overhead are closely to Rao's [24] scheme and much lower than "signing then encryption" scheme.

B. EXPERIMENTAL SIMULATION

In order to evaluate the performance of our scheme, we implement our scheme in software based on a laptop equipped with an Intel $i3-380$ processor running at $2.53GHz$ and $8GB$ memory. Furthermore, our scheme was implemented in java with JPBC library 2.0.0, and set the size of $\mathbb{G}_1$ and $\mathbb{Z}_p^*$ to $64B$ ($512bits$), as well as, we set the size of $\mathbb{G}_2$ to $128B$ ($1024bits$). Through the above settings, we can get the results that a pairing operation costs $647ms$; an exponentiation operation in $\mathbb{G}_1$ and $\mathbb{G}_2$ costs $66ms$ and $13ms$ respectively.

As the result of the theoretical analysis above, it is obvious from Fig.3(a) that the computation cost of the signcryption of the RCS-ABSC scheme is lower than that of the scheme [24] "[44]+[45]". Because the RCS-ABSC scheme generates partial signature by a trusted third cloud server, it reduces the computation of users at the signcryption side. As can be seen from Fig.3(b), the computation cost of the our scheme and scheme [24] at the signcryption side is very close. Overall, we can see that on the basis of providing user revocation function, the efficiency of this scheme is not much lower than that of the comparative scheme, but far lower than that of the combined scheme" [44]+ [45]". Therefore, this scheme is innovative and feasible.

VI. CONCLUSION

In order to solve the problem of user revocation in the attribute-based signcryption scheme, we study the attributed-based signcryption scheme proposed by Rao's scheme [24], and present an efficient and secure RCS-ABSC scheme that supports a user revocation mechanism. At the same time, our scheme has almost the same efficiency to Rao's [24] scheme. Note that it is the first revocable attributed-based signcryption scheme in the literature. Furthermore, we prove the security of our system in the random oracle model. Finally, the experimental evaluation result demonstrates that the proposed scheme is secure and feasible.

REFERENCES

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Proc. 24th Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT), vol. 3494. Aarhus, Denmark: Springer, 2005, pp. 457–473.

[2] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.

[3] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2007, pp. 321–334.

[4] T.-Y. Wu, C.-M. Chen, K.-H. Wang, and J. M.-T. Wu, "Security analysis and enhancement of a certificateless searchable public key encryption scheme for iiot environments," *IEEE Access*, vol. 7, pp. 49232–49239, 2019.

[5] J.-S. Pan, C.-Y. Lee, A. Sghaier, M. Zeghid, and J. Xie, "Novel systolization of subquadratic space complexity multipliers based on toeplitz matrix-vector product approach," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 7, pp. 1614–1622, Mar. 2019.

[6] T.-Y. Wu, C.-M. Chen, K.-H. Wang, C. Meng, and E. K. Wang, "A provably secure certificateless public key encryption with keyword search," *J. Chin. Inst. Eng.*, vol. 42, no. 1, pp. 20–28, 2019.

[7] L. Ni, F. Tian, Q. Ni, Y. Yan, and J. Zhang, "An anonymous entropy-based location privacy protection scheme in mobile social networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 93, 2019. doi: 10.1186/s13638-019-1406-4.

[8] J.-S. Pan, L. Kong, T.-W. Sung, P.-W. Tsai, and V. Snášel, "α-Fraction first strategy for hierarchical model in wireless sensor networks," *J. Internet Technol.*, vol. 19, no. 6, pp. 1717–1726, 2018.

[9] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in *Topics in Cryptology*, vol. 6558. Berlin, Germany: Springer-Verlag, 2011, pp. 376–392.

[10] J. Herranz, F. Laguillaumie, B. Libert, and C. Ráfols, "Short attribute-based signatures for threshold predicates," in *Proc. Cryptogr. Track RSA Conf.*, in Lecture Notes in Computer Science, San Francisco, CA, USA, vol. 7178. Berlin, Germany: Springer, Feb./Mar. 2012, pp. 51–67.

[11] J. Li, M. H. Au, W. Susilo, D. Xie, and K. Ren, "Attribute-based signature and its applications," in *Proc. 5th ACM Symp. Inf., Comput. Commun. Secur.*, 2010, pp. 60–69.

[12] M. Gagné, S. Narayan, and R. Safavi-Naini, "Short pairing-efficient threshold-attribute-based signature," in *Proc. Int. Conf. Pairing-Based Cryptogr.* Cologne, Germany: Springer, 2012, pp. 295–313.

[13] K.-H. Wang, C.-M. Chen, W. Fang, and T.-Y. Wu, "On the security of a new ultra-lightweight authentication protocol in IoT environment for RFID tags," *J. Supercomput.*, vol. 74, no. 1, pp. 65–70, 2018.

[14] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, "A secure authentication protocol for Internet of vehicles," *IEEE Access*, vol. 7, pp. 12047–12057, 2019.

[15] H. Xiong, Q. Mei, and Y. Zhao, "Efficient and provably secure certificateless parallel key-insulated signature without pairing for IIoT environments," *IEEE Syst. J.*, to be published. doi: 10.1109/JSYST.2018.2890126.

[16] H. Xiong, H. Zhang, and J. Sun, "Attribute-based privacy-preserving data sharing for dynamic groups in cloud computing," *IEEE Syst. J.*, to be published.

[17] K.-H. Yeh, "A secure transaction scheme with certificateless cryptographic primitives for IoT-based mobile payments," *IEEE Syst. J.*, vol. 12, no. 2, pp. 2027–2038, Jun. 2018.

[18] C.-M. Chen, K.-H. Wang, K.-H. Yeh, B. Xiang, and T.-Y. Wu, "Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 8, pp. 3133–3142, 2019.

[19] H. Xiong, Q. Mei, Y. Zhao, L. Peng, and H. Zhang, "Scalable and forward secure network attestation with privacy-preserving in cloud-assisted Internet of Things," *IEEE Sensors J.*, to be published. doi: 10.1109/JSEN.2019.2919508.

[20] M. Gagné, S. Narayan, and R. Safavi-Naini, "Threshold attribute-based signcryption," in *Proc. 7th Int. Conf. Secur. Cryptogr. Netw.*, vol. 6280. Amalfi, Italy: Springer, 2010, pp. 154–171.

[21] K. Emura, A. Miyaji, and M. S. Rahman, "Dynamic attribute-based signcryption without random oracles," *Int. J. Adv. Comput. Technol.*, vol. 2, no. 3, pp. 199–211, 2012.

[22] Y. S. Rao and R. Dutta, "Efficient attribute-based signature and signcryption realizing expressive access structures," *Int. J. Inf. Secur.*, vol. 15, no. 1, pp. 81–109, Feb. 2016.

[23] Q. Mei, Y. Zhao, and H. Xiong, "A new provably secure certificateless signature with revocation in the standard model," *Informatica*, to be published.

[24] Y. 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, Feb. 2017.

[25] N. Attrapadung and H. Imai, "Conjunctive broadcast and attribute-based encryption," in *Proc. 3rd Int. Conf. Pairing-Based Cryptogr.* Palo Alto, CA, USA: Springer, 2009, pp. 248–265.

[26] H. Cui, R. H. Deng, J. K. Liu, X. Yi, and Y. Li, "Server-aided attribute-based signature with revocation for resource-constrained industrial-Internet-of-Things devices," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3724–3732, Aug. 2018.

[27] A. Fiat and M. Naor, "Broadcast encryption," in *Proc. 13th Annu. Int. Cryptol. Conf. Adv. Cryptol. (CRYPTO)*. Santa Barbara, CA, USA: Springer, 1993, pp. 480–491.

[28] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Proc. 21st Annu. Int. Cryptol. Conf. Adv. Cryptol. (CRYPTO)*. Santa Barbara, CA, USA: Springer, 2001, pp. 41–62.

[29] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proc. 25th Annu. Int. Cryptol. Conf. Adv. Cryptol. (CRYPTO)*. Santa Barbara, CA, USA: Springer, 2005, pp. 258–275.

[30] A. B. Lewko, A. Sahai, and B. Waters, "Revocation systems with very small private keys," in *Proc. 31st IEEE Symp. Secur. Privacy (SP)*, May 2010, pp. 273–285.

[31] C. Gentry and B. Waters, "Adaptive security in broadcast encryption systems (with short ciphertexts)," in *Proc. 28th Annu. Int. Conf. Theory Appl. Cryptogr. Techn.* Berlin, Germany: Springer, Apr. 2009, pp. 171–188.

[32] H. Xiong, Y. Zhao, L. Peng, H. Zhang, and K.-H. Yeh, "Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing," *Future Gener. Comput. Syst.*, vol. 97, pp. 453–461, Aug. 2019.

[33] J. Staddon, P. Golle, M. Gagné, and P. Rasmussen, "A content-driven access control system," in *Proc. 7th Symp. Identity Trust Internet*, 2008, pp. 26–35.

[34] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, Oct. 2007, pp. 195–203.

[35] T. Matsumoto, K. Kato, and H. Imai, "Speeding up secret computations with insecure auxiliary devices," in *Proc. 8th Annu. Int. Cryptol. Conf. Adv. Cryptol. (CRYPTO)*, 1988, pp. 497–506.
[36] C. Hui, M. Yi, and F. Guo, "Server–aided identity–based anonymous broadcast encryption," *Int. J. Secur. Netw.*, vol. 8, no. 1, pp. 29–39, 2013.
[37] F. Guo, M. Yi, W. Susilo, and V. Varadharajan, "Server-aided signature verification for lightweight devices," *Comput. J.*, vol. 57, no. 4, pp. 481–493, 2018.
[38] H. Xiong and J. Sun, "Comments on 'verifiable and exculpable outsourced attribute-based encryption for access control in cloud computing,'" *IEEE Trans. Dependable Secure Comput.*, vol. 14, no. 4, pp. 461–462, Jul. 2017.
[39] H. Xiong, Y. Bao, X. Nie, and Y. I. Assor, "Server-aided attribute-based signature supporting expressive access structures for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, to be published. doi: 10.1109/TII.2019.2921516.
[40] C. Chen, J. Chen, H. W. Lim, Z. Zhang, and D. Feng, "Combined public key schemes: The case of ABE and ABS," in *Proc. Int. Conf. Provable Secur.* Berlin, Germany: Springer, 2012, pp. 53–69.
[41] C.-J. Wang, J.-S. Huang, W.-L. Lin, and H.-T. Lin, "Security analysis of Gagne et al.'s threshold attribute-based signcryption scheme," in *Proc. 5th Int. Conf. Intell. Netw. Collaborative Syst. (INCoS)*, Sep. 2013, pp. 103–108.
[42] Y. Han, W. Lu, and X. Yang, "Attribute-based signcryption scheme with non-monotonic access structure," in *Proc. 5th Int. Conf. Intell. Netw. Collaborative Syst. (INCoS)*, Sep. 2013, pp. 796–802.
[43] J. Liu, X. Huang, and J. K. Liu, "Secure sharing of personal health records in cloud computing: Ciphertext-policy attribute-based signcryption," *Future Gener. Comput. Syst.*, vol. 52, pp. 67–76, Nov. 2015.
[44] J.-L. Qian and X.-L. Dong, "Fully secure revocable attribute-based encryption," *J. Shanghai Jiaotong Univ. (Sci.)*, vol. 16, no. 4, pp. 490–496, 2011.
[45] Y. Lian, L. Xu, and X. Huang, "Attribute-based signatures with efficient revocation," in *Proc. 5th Int. Conf. Intell. Netw. Collaborative Syst. (INCoS)*, Sep. 2013, pp. 573–577.

FUHU DENG received the B.Eng. and M.Eng. degrees in electronic engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2006 and 2009, respectively, and the Ph.D. degree in wireless communications from the Dublin Institute of Technology, Ireland, in 2014. His research interests include network security, wireless communications, and resource management. He is a member of the ACM.

YALI WANG received the B.S. degree from Gansu Agricultural University. She is currently pursuing the M.S. degree with the School of Information and Software Engineering, University of Electronic Science and Technology of China (UESTC). Her research interests include attribute-based signcryption and malicious code detection.

LI PENG received the B.S. degree from Guangxi University. He is currently pursuing the M.S. degree with the School of Information and Software Engineering, University of Electronic Science and Technology of China. His research interests include attribute-based encryption and malicious code detection.

MIAO LAI received the B.S. degree from Tianjin Normal University. She is currently pursuing the M.S. degree with the School of Information and Software Engineering, University of Electronic Science and Technology of China. Her research interest includes workflow scheduling.

JI GENG received the M.S. degree from Southwest Jiaotong University and the Ph.D. degree from the University of Electronic Science and Technology of China (UESTC), where he is currently a Professor with the School of Information and Software Engineering, University of Electronic Science and Technology of China. He has been involved in distributed computing and information security research.