An Efficient and Secure Attribute-Based Signcryption Scheme for Smart Grid Applications*

Seyyed Mahdi Sedaghat[1], Mohammad Hassan Ameri[2], Mahshid Delavar[2], Javad Mohajeri[2], Mohammad Reza Aref [1]

***Abstract*— With regards to the development of modern power systems, Smart Grid (SG) as an intelligent generation of electricity networks has been faced with a tremendous attention. Fine-grained data sharing in SG plays a vital role in efficiently managing data flow in the SG. As these data commonly contain sensitive information, design of the secure and efficient privacy preserving schemes for such networks with plenty of resource constrained devices is one of the most controversial issues. In this paper, we propose a secure Ciphertext-Policy Attribute-Based SignCryption (CP-ABSC) scheme which simultaneously provides the authenticity and privacy of the users by enforcing an arbitrary access control policy on encrypted data. Since the number of required pairings in the signcryption and designcryption algorithms are independent to the number of the involved attributes, the computational overhead is reduced in comparison with the existing schemes in the literature. In addition, we formally prove that the unforgeability and indistinguishability of the proposed scheme are reducible to the well-known hardness assumption of the q-Bilinear Diffie-Hellman Exponent (q-BDHE) problem. Moreover, we show that embedding a Physical Unclonable Function (PUF) in each smart meter will significantly reduce the storage overhead of the protocol and secure it against non-volatile memory attackers.**

Keywords: Smart Grid (SG), Ciphertext-Policy Attribute-Based Signcryption (CP-ABSC) scheme, Authentication, Physical Unclonable Function (PUF)

1. Introduction

Recently, Smart Grid (SG) as the next generation of the power grid has fascinated the attention of a great number of researchers. The SG can be regarded as an electrical system that uses two-way and cyber-secure communication technologies along with the computational intelligence in an integrated fashion across electricity generation, transmission, substations, distribution, and consumption. The purpose behind introducing the SG technology is achieving a system which is clean, safe, secure, reliable, resilient, efficient, and sustainable [1, 2]. Alongside these attractive features, SG faces many challenges, specifically in cyber security and privacy [3].

Data sharing activities in SG are useful in several domains, and can be used in different applications [4-7]. Since many grid operators and smart devices participate in managing and controlling the grid, they need to share data and cooperate with each other to efficiently manage the grid behavior [7]. Usually, the shared data contains sensitive information and its privacy should be preserved to provide a secure communication. It should be highlighted that to achieve a secure data sharing scheme in SG, it is required to establish arbitrary access control policies for data encryption and authentication [8]. The Ciphertext-Policy

1. S. M. Sedaghat and M. R. Aref are with the Information Systems and Security Laboratory, Department of Electrical Engineering, Sharif University of Technology, (e-mail: sedaghat_seyyedmahdi@ee.sharif.edu; aref@sharif.edu).

2. M. H. Ameri, M. Delavar and J. Mohajeri are with the Electronics Research Institute, Sharif University of Technology, Tehran 11155-11365, Iran (e-mail: Ameri_mohammadhasan@ee.sharif.edu; m.delavar@mehr.sharif.ir; mohajer@sharif.edu).

* This work was partially supported by Iran NSF under Grant No. 92.32575 and Center of Excellence in Cryptography and Information Security, Sharif University of Technology

Attribute-Based Encryption (CP-ABE) schemes [9-11] are promising solutions for enabling a scalable and secure data sharing in SG.

The concept of CP-ABE was first introduced by Sahai and Waters in 2007 [9]. In their scheme, the users are allowed to implement a fine-grained access control on their data for encrypting and sharing them in a one-to-many communications model. Since then several schemes have been introduced to improve efficiency and security of this scheme for adapting to the SG (e.g., [6, 7, 12, 13]).

Some CP-ABE schemes have been designed based on Linear Secret Sharing Scheme (LSSS) or Boolean formulas to establish an arbitrary access policy. Lewko and Waters [14] introduced a secure construction based on LSSS, which can convert any monotone Boolean formulas to LSSS matrices. Its security has been proved in the composite order bilinear groups. Their scheme is very inefficient, since the length of its ciphertexts and keys, and the number of pairings in decryption are all polynomial in the size of monotone span programs (MSPs). Waters [15] presented a CP-ABE scheme employing LSSS matrix as an access policy based on prime order bilinear pairing. Typically, the existing CP-ABE schemes have heavy computational cost for SG applications. So, it is more efficient to delegate a remote storage center to run the partial decryption of the outsourced encrypted data in the SG [7]. Besides of all the mentioned advantages of the CP-ABE schemes, we should highlight that in these schemes, the authenticity verification of the received messages has not been considered. To address this issue, attribute-based signcryption schemes were introduced [16, 17]. By enforcing an arbitrary access control policy to these schemes, Ciphertext-Policy Attribute-Based Signcryption (CP-ABSC) schemes are constructed. These schemes provide strong security in terms of collusion resistance, message authentication, unforgeability, and data confidentiality [18].

The existing CP-ABE schemes have heavy computational cost for SG applications. So, a storage center with high computational capability can be used for executing partially designcryption of signcrypted data [7]. The present paper aims to propose a secure and efficient attribute-based signcryption scheme which is adapted to the SG in which many of its components have limited computational resources. In this case, the client who plans to outsource its sensitive data, can generate the ciphertexts under specific and arbitrary access control policy which determines authorized entities for decrypting the stored encrypted data.

1.1. Our Contribution

In this paper, we propose an efficient and secure data sharing scheme based on Ciphertext-Policy Attribute-Based Signcryption Scheme (CP-ABSC) as a security mechanism for simultaneously providing user privacy through establishing access control policies, data encryption and message authentication in SG. In the signcryption algorithm, the data is signcrypted according to an arbitrary access control policy such that it can be designcrypted by using a valid secret key related to a set of attributes which satisfies the applied access structure. However, in most of the CP-ABSC schemes, the required number of bilinear pairings to sign and encrypt the data is linearly dependent on the number of attributes. These schemes require heavy computations during the signcrypting and designcrypting because of pairing computations, which grows linearly with the size of the attributes [19, 20]. The scientific contributions of the present paper can be summarized as follows:

- In the proposed scheme, the number of required pairing computations is independent of the number of the intended attributes. This results in lower computational overhead compared to the existing CP-ABSC schemes.
- We formally prove the security of the proposed scheme in the standard model. We show that the unforgeability and CP-ABSC-IND-CCA security of our proposal are tightly related to the hardness assumption of breaking the q-BDHE (q-Bilinear Diffie-Hellman Exponent) problem.

- Performance evaluation of the proposed scheme in terms of both computational complexity and execution time and comparison with the existing works in the literature show the practical and deployable aspects of our proposed scheme.
- Moreover, we consider embedding Physically Unclonable Functions (PUFs) in the smart meters to improve the security of our scheme against the non-volatile memory attackers. In this way the enhanced system will be fully memory leakage resilient. Also, we show that by using PUF-enabled devices the storage overhead of our scheme is significantly decreased.

1.2. Organization

The rest of the paper is organized as follows. In Section II, we present the preliminaries and definition of our scheme. Then, in section III, we describe system architecture and Section IV, presents the construction of our scheme. Section V discusses the security analysis and contains security definitions and security proofs. Section VI shows the performance analysis and implementation of the proposed scheme.

2. Preliminaries and Definitions

In this section, we first present formal definition of monotone access structure [21]. Then, we briefly give some background information on Linear Secret Sharing Schemes (LSSS) and bilinear pairings. We also review the q-BDHE assumption which will be used in the security proof of the proposed scheme. After that we introduce the concept of Physical Unclonable Functions (PUFs). Finally, we describe the applied notations in rest of the paper.

2.1. Access Structures

Definition 1 (Monotone Access Structure [21]). Let $\mathbb{P} = \{P_1, P_2, \dots, P_n\}$ be a set of attributes. An Authorized collection $\mathbb{A} \subseteq 2^{\{P_1, P_2, \dots, P_n\}}$ is called monotone access structure if:

- $\forall B, C: if\ B \in \mathbb{A}\ and\ B \subseteq C \Longrightarrow C \in \mathbb{A}$.

We say that an attribute set B satisfies $\mathbb{A}$ (in other words, $\mathbb{A}$ accepts B) if and only if $B \in \mathbb{A}$.

2.2. Linear Secret Sharing Schemes

Definition 2 (Linear Secret Sharing Schemes (LSSS) [21]). A secret-sharing scheme $\Pi_{\mathbb{A}}$ for the access structure $\mathbb{A}$ over a set of attributes $\mathbb{P}$ is called linear (over Z_P^*) if:

1. The shares of a secret $s \in Z_P^*$ for the set of attributes form a vector over Z_P^*.
2. There exists a matrix $TM_{\ell \times d}$ called the share-generating matrix for $\Pi_{\mathbb{A}}$. The i^{th} row of $TM_{\ell \times d}$, TM_i, is labeled by $\rho(i)$ where ρ is a function from $\{1,2, \dots, \ell\}$ to $\mathbb{P}$. We consider the column vector $\vec{v} = \{s, r_2, \dots, r_d\}$, where $s \in Z_P^*$ is the secret to be shared and $r_2, \dots, r_d \in Z_P$ are randomly chosen. So $TM_{\ell \times d} \times \vec{v}^T$ is the vector of ℓ shares of the secret s according to $\Pi_{\mathbb{A}}$. The share $\lambda_i = (TM_{\ell \times d} \times \vec{v}^T)_i$, corresponds to the attribute $\rho(i)$.

There is a close relation between LSSS and Monotone Span Program (MSP) [20]. Suppose $S \in \mathbb{A}$ be any authorized set, and let $I \subset \{1, \dots, l\}$ be defined as $I = \{i | \rho(i) \in S\}$. Then, there exist constants $\{\omega_i \in Z_p\}_{i \in I}$ such that, if $\{\lambda_i\}_{i \in I}$ are valid shares of the secret s according to $\prod_{\mathbb{A}}$, then $\sum_{i \in I} \omega_i \lambda_i = s$. Let TM_i denotes i^{th} row of $TM_{\ell \times d}$, then $\sum_{i \in I} \omega_i TM_i = (1,0, \dots, 0)_{1 \times d}$. Moreover, it has been proved in [18] that the constants $\{\omega_i\}$ can be found with polynomial time complexity in term of the size of the share-generation matrix $TM_{\ell \times d}$, where ℓ is the number of attributes and d is the level of the access structure. Note that, for unauthenticated sets, such constants $\{\omega_i\}$ cannot be found. For more details about access structure

and LSSS technique, we refer to [15]. For generating an access structure (TM, ρ), we use techniques based on LSSS defined in [22].

2.3. Bilinear Pairings

Definition 3 (Bilinear Maps [23]). Let $\mathbb{G}$ and $\mathbb{G}_1$ be multiplicative cyclic groups of the same prime order p, and let g be a generator of $\mathbb{G}$. The map $e: \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_1$ is said to be bilinear if it has the following properties:

1. For all $b \in Z_p$, $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.
2. $e(g, g) \neq 1$.
3. For all $h, h' \in \mathbb{G}$, there exists an efficient algorithm for computing $e(h, h')$.

2.4. Decisional Bilinear Diffie–Hellman Exponent Assumption

Definition 4 (Decisional Bilinear Diffie–Hellman Exponent (BDHE) assumption [24]). Let $a, s \in Z_p^*$ be chosen at random and g be a generator of $\mathbb{G}$. The decisional q-BDHE assumption which was introduced by Boneh et al. [24] states that no probabilistic polynomial-time adversary $\mathcal{A}$ can distinguish between $e(g, g)^{a^{q+1}s} \in \mathbb{G}_1$ and a random element $R \in \mathbb{G}_1$ with a non-negligible advantage, when given $\vec{y} = \{g, g^s, g^a, g^{a^2}, \ldots g^{a^q}, g^{a^{q+2}}, \ldots, g^{a^{2q}}\}$. The advantage of adversary $\mathcal{A}$ in solving the decisional q-BDHE assumption is:

$$\mathrm{Adv}_{\mathcal{A}} = \left|\Pr[\mathcal{A}(\vec{y}, \mathrm{T} = \mathrm{e(g,g)}^{\mathrm{a^{q+1}s}}) = 0] - \Pr[\mathcal{A}(\vec{y}, \mathrm{T} = \mathrm{R}) = 0]\right| \quad (1)$$

2.5. Physical Unclonable Functions

Definition 5 (Physical unclonable function (PUF) [25]). PUF is a physical entity that is embedded in a device and gives it unique characteristics such that its reproduction by other devices is practically impossible [25]. A PUF takes a bit string as a challenge $C_i \in C$, where C is the set of all possible challenges as input and outputs $R_i^j \in R^j$, where R^j is the set of all possible responses of PUF_j. The CRP term is the abbreviation of Challenge-Response Pair and is used to denote each applied challenge to PUF and its corresponding response. In this paper, we apply the following mathematical relationship to show the behavior of a sample PUF:

$$R_i^j = PUF_j(C_i) \quad (2)$$

In general, the main properties for all PUFs include:

- **Reliability** means that the response of the same PUF to the same challenge is not changed by applying the challenge multiple times.
- **Uniqueness** means that the responses of several PUFs to the same challenge should be different.
- **Unpredictability** means that one should not be able to predict the response of a PUF to a specified challenge by knowing the previous challenge response pairs.
- **Tamper-evidence** means that any attempt to externally obtain its outputs or parameters changes its challenge response behavior.

These properties make PUFs very suitable for key generation and device authentication. The PUF responses can be applied as secret keys or unique IDs. So, by embedding PUFs in the intended devices, the secret keys can be generated

when needed and there will be no need to store them in the non-volatile memories. This decreases the required storage in the device.

2.6. Notations

The applied notations in our protocol are described and summarized in Table.1.

Table 1. Notations

Notation	Definition
MPK, MSK	Master Public Key and Master Secret Key of the Key Generation Center
$H(.)$	A one-way Hash function
U	The set of all possible attributes
$TM_{\ell \times d}$	The matrix of signcryption access structure
A_i	The i^{th} row of the matrix A
$\rho(.)$	The function associated with each row of $TM_{\ell \times d}$
SK_u	The secret key of the user u
$TK_{\psi,u}$	The generated token under attributes set ψ by the user u
m	The plaintext message
CT	The ciphertext
PD_{S_j}	Partially designcrypted ciphertext by the Storage Center under attributes set S_j
δ	The generated signature by the Service Provider

3. System Architecture

We have considered the SG communication system illustrated in Figure 1 as our system model. There are four entities in this system, which are described as follows:

Key Generation Center (KGC) which generates and distributes cryptographic keys for all of the system components including Smart Meters and Service Providers.

Smart Meter (SM) plays an important role in SG systems to control individuals house-hold devices in the hierarchical structure of the SG. When a service provider wants to update special software for one of its productions, it can securely send the new version of the software to a group of smart meters which are utilizing the mentioned product. SMs are typically resource constrained devices with limited computational resources.

Storage Center (SC) is a data repository center in the grid which has sufficient computational capacity to partially designcrypt the received ciphertexts. We assume that the SC is semi-honest. It means that the SC follows the protocols honestly but tries to infer some sensitive information. Therefore, the data should be stored in SC in signcrypted format for the aim of data confidentiality and authenticity.

Service Provider (SP) is considered as an entity that provides some additional services for consumers. For example, a SP can provide a tool for consumers to control and monitor their electricity usage or send the updated version of its produced software. The SP establishes an access policy for signcrypting the data, and then outsources the resulting ciphertexts to the SC. The outsourced ciphertexts will be accessible for the authorized SMs.

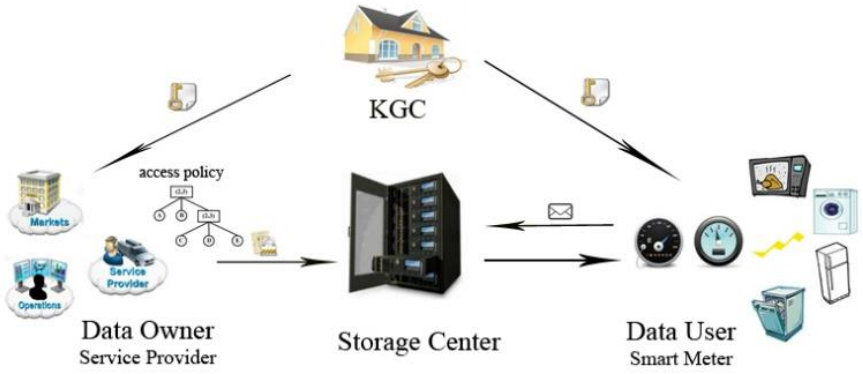


Fig. 1. A communication architecture Data Sharing process in the Smart Grid

4. The proposed Scheme

In this section, we present the proposed scheme and after that show its correctness. In the proposed scheme, when a SP wants to store a ciphertext CT corresponding to the message m in the SC, it applies an access policy which is presented by the tuple (TM, ρ) according to LSSS model, to signcrypts the message, and then uploads the resulting ciphertext to the SC. When an authorized SM wants to access the data outsourced by the SP, it generates a valid token using its credentials under its attributes, and delivers it to the SC. In our system model which is illustrated in Figure 1, we assume that the SC is not fully trusted while it has high computational resources. So, it is applied to help the low-resource SMs to designcrypt the ciphertexts by partially designcryption of the ciphertexts, without inferring any information about the message m. To this end, the SC partially designcrypts all the ciphertexts under the access policies which are satisfied by the SM's attributes and sends them to the SM. The SM receives partially designcrypted ciphertexts and designcrypts them using its secret keys.

4.1. Our scheme

The proposed scheme consists of six algorithms: $Setup$, $KeyGen$, $Signcryption$, $TokenGen$, $PartialDesigncryption$, and $Designcryption$. The $Setup$ and $KeyGen$ algorithms are performed by the KGC while the $Signcryption$, and $PartialDesigncryption$ algorithms are executed by SP and SC, respectively. The $TokenGen$ and $Designcryption$ algorithms are run by SM.

$\boldsymbol{Setup(U, \mathcal{N}) \rightarrow (MPK, MSK)}$**:** This algorithm is run by KGC. The algorithm takes the attribute universe U and the security parameter $\mathcal{N}$, as input and outputs the master secret key, MSK and the public parameters, MPK. To this end, by considering the security parameter $\mathcal{N}$, it chooses a cyclic group $\mathbb{G}$ of prime order p with generator g. Let $e: \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_1$ and $e': \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ are two bilinear maps. For the attribute universe set $U = \{Att_1, Att_2, \dots, Att_n\}$, this algorithm randomly generates the values $h_{Att_1}, h_{Att_2}, \dots, h_{Att_n} \in_r Z_p^*$ to map each attribute to a unique element in Z_p^*. Also, it chooses random integers $\gamma, \alpha, \beta \in_r Z_p^*$, and selects the collision resistance one-way hash function $H: \{0,1\}^* \longrightarrow \mathbb{G}$. Finally, it outputs the master public parameters, MPK, and the master secret key MSK as follows:

$$MPK = \{g, H, e(g,g)^{\alpha}, g^{\gamma}, g^{\beta}, h_{Att_1}, h_{Att_2}, \dots, h_{Att_n}\} \quad (3)$$

$$MSK = \{g^{\alpha}, \alpha, \gamma, \beta\} \quad (4)$$

$\boldsymbol{KeyGen(MSK, S_j) \rightarrow (SK_{S_j,u}, K_{ver_u})}$: The KGC runs this algorithm to generate the secret key of the user u (a smart meter or a service provider), associated to the attribute set S_j. The inputs of this algorithm are the master secret key MSK and the attribute set $S_j \subseteq U$ associated to the user u and its outputs are

the private key of the user u, $SK_{S_j,u} = \left(K_u, K'_u, K_{x_u}, K_{sign_u}\right)$ and the verification key K_{ver_u}. For this purpose, it chooses the integers $t_u, r_{su} \in_r Z_p^*$ uniformly at random. Then, it computes the secret signing key $K_{sign_u} = e(g,g)^{\frac{(\alpha + r_{su})}{\gamma}}$ and the verification key $K_{ver_u} = e(g,g)^{r_{su}}$, which are respectively applied in generating an authenticated message in the *S*igncryption algorithm and verifying the authenticity of the signcrypted messages. The KGC publishes the verification key K_{ver_u} and issues a certificate which ensures that this verification key is associated to the user, u. After that the algorithm sets the users' private key as follows:

$$K_u = g^{\alpha} g^{\beta t_u} \tag{5}$$

$$K'_u = g^{t_u} \tag{6}$$

$$\forall\, x \in S_j \;\rightarrow\; K_{x_u} = h_x{}^{t_u} \tag{7}$$

$$SK_{S_j,u} = \left(K_u, K'_u, K_{x_u}, K_{sign_u}\right) \tag{8}$$

This key is stored in the non-volatile memory of the user u.

$\boldsymbol{Signcryption\left(MPK, m, K_{sign_u}, (TM, \rho)\right) \rightarrow CT:}$ The service provider (SP) runs the signcryption algorithm and outsources the output to the storage center (SC). This algorithm takes the public parameter MPK, the message m, the secret signing key K_{sign_u}, and the LSSS matrix (TM, ρ) corresponding to an arbitrary threshold access tree Υ as inputs and outputs the signcrypted form of m. It should be mentioned that we can transfer any arbitrary threshold access tree to an LSSS matrix according to the technique which is introduced in [22]. As mentioned in definition 2, this algorithm randomly chooses a secret integer $s \in_r Z_p^*$ and a vector $\vec{v} = \{s, r'_2, \dots, r'_d\} \in_r Z_p^*$. Then for $i = 1, \dots, \ell$, it calculates $\lambda_i = \vec{v} \times TM_i$, where TM_i is the i^{th} row of matrix $TM_{\ell \times d}$. Also, it chooses random values $r_1, r_2, \dots, r_\ell \in_r Z_p^*$ and generates the ciphertext

$$CT = \left((TM, \rho), C, C', C'', C_i, D_i, \pi, \Omega\right) \tag{9}$$

Where,

$$C = m.e(g,g)^{\alpha s}, C' = g^s, C'' = g^{\gamma s}$$
$$\left(C_i = g^{\beta \lambda_i} h_{\rho(i)}{}^{-r_i}, D_i = g^{r_i}\right) \; for\; all \; i = 1, \dots, \ell \tag{10}$$
$$\delta = e'(e(C'', g), e(g,g)^{\hbar}), \pi = H(\delta | m)$$

$$\Omega = e(g,g)^{\hbar} (K_{sign_u})^{\pi} \left(\hbar \in_r Z_p^*\right) \tag{11}$$

$\boldsymbol{TokenGen\left(MPK, SK_{S_j}\right) \rightarrow TK_{S_j,u}}$: The SM which posses the set of attributes S_j runs this algorithm to access the shared data in the SC. This algorithm takes the public parameter MPK and the secret key SK_{S_j}, as inputs, and generates a random number $r \in_r Z_p^*$ and then calculate the token $TK_{S_j,u}$ for the set of attributes S_j as follows:

$$TK_{S_j,u} = \left(S_j, K_u{}^r, K'_u{}^r, K_{x_u}^r \; \forall\, x \in S_j\right) \tag{12}$$

The token $TK_{S_j,u}$ is sent to the SC.

$\boldsymbol{PartialDesigncryption\left(MPK, CT, TK_{S_j,u}\right) \rightarrow PD_{S_j}}$: This algorithm takes the public parameter MPK, the ciphertext CT, and the token $TK_{S_j,u}$ as inputs and outputs the partial designcrypted part of the ciphertext CT as PD_{S_j}. The SC, after receiving the token $TK_{S_j,u}$ from the SM, checks whether the set of attributes S_j satisfies the access policy (TM, ρ) or not. Then, it partially designcrypts the ciphertext as follows. The SC first computes W through equation (13).

$$W = \frac{e(C', K_u{}^r)}{\prod_{i \in \mathcal{J}} \left(e\left(g^{\beta \lambda_i} h_{\rho(i)}{}^{-r_i}, K'_u{}^r\right) . e\left(g^{r_i}, K_{x_u}^r\right)\right)^{\omega_i}} = e(g,g)^{rs\alpha} \tag{13}$$

Where $\mathcal{I} = \{i: \rho(i) \in S_j\}$ and supposing that $\{\lambda_i\}_{i\in\mathcal{I}}$ are the valid shares of the secret value s according to $TM_{\ell\times d}$, the values $\{\omega_i \in Z_p^*\}_{i\in\mathcal{I}}$ are chosen such that $\sum_{i\in\mathcal{I}} \lambda_i.\omega_i = s$. Then, it sends back the tuple $PD_{S_j} = (C, C', C'', W, \pi, \Omega)$ to the SM.

$\boldsymbol{Designcryption\left(CT, PD_{S_j}, K_{ver_u}\right) \rightarrow m'}$**:** The authorized SM can run this algorithm to designcrypt CT using the random value r which generated in the $TokenGen$ algorithm, and the received vector PD_{S_j} as follows:

$$m' = \frac{C}{w} = \frac{C}{(W)^{r^{-1}}} = \frac{m.e(g,g)^{\alpha s}}{(e(g,g)^{\alpha rs})^{r^{-1}}} \tag{14}$$

$$\delta' = \frac{e'(e(C'',g),\Omega)}{e'\left(e(g^s,g)^{\pi}, K_{ver_u}\right).e'(e(g,g)^{\pi},w)} \tag{15}$$

If $H(\delta'|m') = \pi$ then the algorithm returns m'; otherwise, it returns a null symbol as its output.

4.2. Correctness of the Proposed Scheme

In this subsection, we illustrate that the proposed scheme works correctly. We claim that the SC can correctly partial designcrypt the ciphertext if and only if the attributes set S_j satisfies the access structure, and the SM can verify whether the received message has been forged or falsified, and whether the received message is indeed sent by the SP or not. For the first step, we verify establishing the equation (13) which is related to partial designcryption by the SC. And then we show the correctness of the equation (15).

$$\begin{aligned}
\mathrm{W} &= \frac{\mathrm{e}(\mathrm{C}', \mathrm{K_u}^{\mathrm{r}})}{\prod_{\mathrm{i}\in\mathcal{I}}\left(\mathrm{e}(\mathrm{g}^{\beta\lambda_\mathrm{i}}\mathrm{h}_{\rho(\mathrm{i})}{}^{-\mathrm{r_i}},\ \mathrm{K'_u}^{\mathrm{r}}).\mathrm{e}\left(\mathrm{g}^{\mathrm{r_i}}, \mathrm{K}^{\mathrm{r}}_{\mathrm{x_u}}\right)\right)^{\omega_\mathrm{i}}} \\
&= \frac{\mathrm{e}\left(\mathrm{g}^{\mathrm{s}}, \mathrm{g}^{(\alpha+\beta\mathrm{t_u})\mathrm{r}}\right)}{\prod_{\mathrm{i}\in\mathcal{I}}\left(\mathrm{e}(\mathrm{g}^{\beta\lambda_\mathrm{i}}\mathrm{h}_{\rho(\mathrm{i})}{}^{-\mathrm{r_i}},\ \mathrm{g}^{\mathrm{rt_u}}).\mathrm{e}(\mathrm{g}^{\mathrm{r_i}}, \mathrm{h}_{\rho(\mathrm{i})}{}^{\mathrm{rt_u}})\right)^{\omega_\mathrm{i}}} \\
&= \frac{\mathrm{e}(\mathrm{g},\mathrm{g})^{(\alpha+\beta\mathrm{t_u})\mathrm{rs}}}{\prod_{\mathrm{i}\in\mathcal{I}}\left(\mathrm{e}\left(\mathrm{g}^{\mathrm{rt_u}}, \mathrm{g}^{\beta\lambda_\mathrm{i}}.\mathrm{h}_{\rho(\mathrm{i})}{}^{-\mathrm{r_i}}.\mathrm{h}_{\rho(\mathrm{i})}{}^{\mathrm{r_i}}\right)\right)^{\omega_\mathrm{i}}} \\
&= \frac{\mathrm{e}(\mathrm{g},\mathrm{g})^{(\alpha+\beta\mathrm{t_u})\mathrm{rs}}}{\mathrm{e}(\mathrm{g},\mathrm{g})^{\beta\mathrm{rt_u}\sum_{\mathrm{i}\in\mathcal{I}}\lambda_\mathrm{i}\omega_\mathrm{i}}} = \frac{\mathrm{e}(\mathrm{g},\mathrm{g})^{\alpha\mathrm{rs}}.\mathrm{e}(\mathrm{g},\mathrm{g})^{\beta\mathrm{t_u}\mathrm{rs}}}{\mathrm{e}(\mathrm{g},\mathrm{g})^{\beta\mathrm{t_u}\mathrm{rs}}} \\
&= \mathrm{e}(\mathrm{g},\mathrm{g})^{\alpha\mathrm{rs}}
\end{aligned} \tag{16}$$

Also,

$$\begin{aligned}
\delta' &= \frac{\mathrm{e}'(\mathrm{e}(\mathrm{C}'',\mathrm{g}),\Omega)}{\mathrm{e}'\left(\mathrm{e}(\mathrm{g}^{\mathrm{s}},\mathrm{g})^{\pi}, \mathrm{K}_{\mathrm{ver_u}}\right).\mathrm{e}'(\mathrm{e}(\mathrm{g},\mathrm{g})^{\pi},\mathrm{w})} \\
&= \frac{\mathrm{e}'\left(\mathrm{e}(\mathrm{g}^{\mathrm{s}\gamma},\mathrm{g}), \mathrm{e}(\mathrm{g},\mathrm{g})^{\hbar}(\mathrm{K}_{\mathrm{sign_u}})^{\pi}\right)}{\mathrm{e}'(\mathrm{e}(\mathrm{g}^{\mathrm{s}},\mathrm{g})^{\pi}, \mathrm{e}(\mathrm{g},\mathrm{g})^{\mathrm{r_{su}}}).\mathrm{e}'(\mathrm{e}(\mathrm{g},\mathrm{g})^{\pi},\mathrm{w})} \\
&= \frac{\mathrm{e}'\left(\mathrm{e}(\mathrm{g}^{\mathrm{s}\gamma},\mathrm{g}), \mathrm{e}(\mathrm{g},\mathrm{g})^{\hbar}\left(\mathrm{e}(\mathrm{g},\mathrm{g})^{\frac{(\alpha+\mathrm{r_{su}})}{\gamma}}\right)^{\pi}\right)}{\mathrm{e}'(\mathrm{e}(\mathrm{g}^{\mathrm{s}},\mathrm{g})^{\pi}, \mathrm{e}(\mathrm{g},\mathrm{g})^{\mathrm{r_{su}}}).\mathrm{e}'(\mathrm{e}(\mathrm{g},\mathrm{g})^{\pi}, \mathrm{e}(\mathrm{g},\mathrm{g})^{\alpha\mathrm{s}})} \\
&= \frac{\mathrm{e}'\left(\mathrm{e}(\mathrm{g},\mathrm{g}),\mathrm{e}(\mathrm{g},\mathrm{g})\right)^{\mathrm{s}\gamma\hbar}.\mathrm{e}'\left(\mathrm{e}(\mathrm{g},\mathrm{g}),\mathrm{e}(\mathrm{g},\mathrm{g})\right)^{(\alpha+\mathrm{r_{su}})\pi\mathrm{s}}}{\mathrm{e}'\left(\mathrm{e}(\mathrm{g},\mathrm{g}),\mathrm{e}(\mathrm{g},\mathrm{g})\right)^{\mathrm{s}\pi\mathrm{r_{su}}}.\mathrm{e}'\left(\mathrm{e}(\mathrm{g},\mathrm{g}),\mathrm{e}(\mathrm{g},\mathrm{g})\right)^{\alpha\mathrm{s}\pi}} \\
&= \frac{\mathrm{e}'\left(\mathrm{e}(\mathrm{g},\mathrm{g}),\mathrm{e}(\mathrm{g},\mathrm{g})\right)^{\mathrm{s}\gamma\hbar}.\mathrm{e}'\left(\mathrm{e}(\mathrm{g},\mathrm{g}),\mathrm{e}(\mathrm{g},\mathrm{g})\right)^{(\alpha+\mathrm{r_{su}})\pi\mathrm{s}}}{\mathrm{e}'\left(\mathrm{e}(\mathrm{g},\mathrm{g}),\mathrm{e}(\mathrm{g},\mathrm{g})\right)^{\mathrm{s}\pi(\mathrm{r_{su}}+\alpha)}} \\
&= \mathrm{e}'\left(\mathrm{e}(\mathrm{g},\mathrm{g}),\mathrm{e}(\mathrm{g},\mathrm{g})\right)^{\mathrm{s}\gamma\hbar} = \mathrm{e}'(\mathrm{e}(\mathrm{g},\mathrm{g})^{\mathrm{s}\gamma}, \mathrm{e}(\mathrm{g},\mathrm{g})^{\hbar}) = \delta
\end{aligned} \tag{17}$$

So $H(\delta'|m') = \pi$, then the SM can conclude that the message is valid and is generated by the SP.

5. Security Analysis

5.1. Security Definitions

In this subsection, we present the required security definitions for proving the indistinguishability and unforgeability of the proposed scheme. It must be noted that the introduced games in Definitions 6 and 7 are motivated by the notion of Selective-ID game in [23, 26-28]. We just slightly modify these notations to adapt them to the system model of our introduced CP-ABSC framework.

Definition 6 (Indistinguishability of a CP-ABSC scheme against adaptive chosen ciphertext attack (IND-CCA)). A Ciphertext-policy Attribute-Based Signcryption (CP-ABSC) scheme is said to be indistinguishable against chosen ciphertext attack (IND-CCA), if no probabilistic polynomial time (PPT) adversary has a non-negligible advantage in winning the following game.

Initialization: In this phase, the challenger $\mathcal{C}$ runs the $Setup(U, \mathcal{N}) \rightarrow (MPK, MSK)$ algorithm, and gives the public parameters MPK to the adversary $\mathcal{A}$, and keeps the master secret key MSK by itself. After that, the adversary $\mathcal{A}$ declares the associated matrix of her target access structure (TM^*, ρ^*), and chooses one of the service providers, SP_x as the signer; Then she sends the matrix and ID_{SP_x} to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ generates K_{sign_x} and K_{ver_x} associated to the SP_x, and then keeps K_{sign_x} and publishes K_{ver_x}.

Query Phase 1: The adversary $\mathcal{A}$ can ask polynomially bounded number of queries from the following oracles:

- $O_{KeyGen}(S_j) \rightarrow \left(SK_{S_j,u}, K_{ver_u}\right)$: The adversary $\mathcal{A}$ has access to this oracle which is provided by the challenger $\mathcal{C}$, to adaptively ask for secret key of the attribute set $S_j = \{Att_1, Att_2, \dots, Att_v\}$. The challenger $\mathcal{C}$ calls $KeyGen(MSK, S_j) \longrightarrow \left(SK_{S_j,u}, K_{ver_u}\right)$ and outputs $SK_{S_j,u}$ and K_{ver_u}. The only condition that has to be satisfied for each query is that none of the queried attributes set satisfies the target access structure, and also it never could query for the signing key of the service provider SP_x.
- $O_{TokenGen}(S_j) \longrightarrow TK_{S_j,u}$: The adversary $\mathcal{A}$ has access to this oracle which is provided by the challenger $\mathcal{C}$, to receive the tokens $TK_{S_j,u}$ corresponding to attribute set S_j which is selected arbitrarily by $\mathcal{A}$. For each attribute set S_j, the challenger first runs $O_{KeyGen}(S_j) \rightarrow \left(SK_{S_j,u}, K_{ver_u}\right)$, and then runs $TokenGen\left(MPK, SK_{S_j,u}\right) \longrightarrow TK_{S_j,u}$ and sends the generated token to the adversary $\mathcal{A}$.
- $O_{Signcryption}(m, (TM, \rho)) \rightarrow CT$: The adversary $\mathcal{A}$ has access to this oracle which is provided by the challenger $\mathcal{C}$, to receive the sincryption of the message m under the access policy (TM, ρ) which are selected arbitrarily by $\mathcal{A}$. For each query, the challenger $\mathcal{C}$ runs the algorithm $Signcryption\left(MPK, m, K_{sign_x}, (TM, \rho)\right) \rightarrow CT$ and forwards CT to the adversary $\mathcal{A}$.
- $O_{PartialDesigncryption}\left(CT, TK_{S_j,u}\right) \rightarrow PD_{S_j}$: The adversary $\mathcal{A}$ has access to this oracle, to receive the partially designcrypted of the ciphertext CT by providing the tokens $TK_{S_j,u}$, which are selected by $\mathcal{A}$. For this aim, the challenger $\mathcal{C}$ runs the algorithm $PartialDesigncryption\left(MPK, CT, TK_{S_j,u}\right) \rightarrow PD_{S_j}$ and returns back PD_{S_j} to the adversary $\mathcal{A}$.

- $O_{Designcryption}\left(CT, PD_{S_j}\right) \rightarrow m'$: The adversary $\mathcal{A}$ has access to this oracle which is provided by the challenger $\mathcal{C}$, to receive the designcryption of the ciphertext CT with PD_{S_j}, which attribute set S_j selected arbitrarily by the adversary $\mathcal{A}$. The challenger $\mathcal{C}$ runs the algorithm $Designcryption\left(CT, PD_{S_j}, K_{ver_u}\right) \rightarrow m'$ and forwards it to the adversary $\mathcal{A}$.

Challenge: The adversary $\mathcal{A}$ chooses two equal length plaintexts m_0 and m_1 and sends them to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ flips a fair coin and produces random bit $b \in \{0,1\}$, and runs the algorithm $Signcryption\left(MPK, m_b, K_{sign_x}, (TM^*, \rho^*)\right) \rightarrow CT^*$. Then the challenger sends CT^* to the adversary as the challenge ciphertext.

Query Phase 2: In this phase, after receiving CT^*, the adversary $\mathcal{A}$ can ask again for polynomially bounded number of queries on the above mentioned oracles adaptively in the same way as Phase 1 except that $\mathcal{A}$ cannot query the tuple $\left(CT^*, S_j\right)$ to the designcryption oracle if $\Upsilon^*(S_j) = 1$.

Guess: The adversary outputs b' as a guess for the value of b. The advantage of the adversary $\mathcal{A}$ in the this game is defined as follows:

$$Adv^{IND-CCA}_{\mathcal{A},CP-ABSC}(\mathcal{N}) = \left|\Pr(b' = b) - \frac{1}{2}\right| \quad (18)$$

As mentioned before a Ciphertext-policy attribute-based signcryption scheme satisfies indistinguishability if $Adv^{IND-CCA}_{\mathcal{A},CP-ABSC}(\mathcal{N})$ is a negligible function for all PPT adversaries.

Definition 7 (Unforgeability against chosen access policy and message attacks). A Ciphertext-policy attribute-based signcryption scheme (CP-ABSC) is said to be unforgeable against chosen access policy and message attacks, if no PPT adversary has a non-negligible advantage in winning the following game.

Initialization: The initialization phase is the same as initialization phase presented in **Definition 6**.

Query Phase: The adversary $\mathcal{A}$ can ask polynomially bounded number of queries to the oracles $O_{KeyGen}(S_j)$, $O_{TokenGen}(S_j)$, $O_{Signcryption}(m, (TM, \rho))$, $O_{PartialDesigncryption}\left(CT, TK_{S_j,u}\right)$ and $O_{Designcryption}\left(CT, PD_{S_j}\right)$ which are previously defined in query phase 1 of **Definition 6**.

Forge: In this phase, the adversary A should output a tuple (CT^*, Υ^*, m^*). Then, the challenger selects S_j^* such that $\Upsilon^*(S_j^*) = 1$, and designcrypts CT^* using the decryption private key $SK_{S_j^*}$ generated by calling algorithm $O_{KeyGen}(S_j^*)$. The adversary $\mathcal{A}$ wins the game if $m^* = Designcryption\left(CT^*, PD_{S_j^*}, K_{ver_x}\right)$. Consequently the tuple (CT^*, Υ^*, m^*) is considered as a forge for the message of m^*. The advantage of the adversary $\mathcal{A}$ in the this game is defined as follows:

$$Adv^{CMF}_{\mathcal{A},CP-ABSC}(\mathcal{N}) = \left|\Pr\left(\begin{array}{c}(CT^*, \Upsilon^*, m^*) \\ O_{KeyGen}(.), O_{Signcryption}(.), O_{TokenGen}(.), \\ O_{PartialDesigncryption}(.), O_{Designcryption}(.); \\ m^* = Designcrypt\left(CT^*, PD_{S_j^*}, K_{ver_x}\right)\end{array}\right)\right| \quad (19)$$

Therefore, a Ciphertext-policy attribute-based signcryption scheme satisfies unforgeability if $Adv^{CMF}_{\mathcal{A},CP-ABSC}(\mathcal{N})$ is a negligible function for all PPT adversaries.

5.2. Security weakness of Hur Scheme

Hur in [7] has claimed that, when an SP sends the ciphertext to the SC, the KGC, which is a semi-honest entity, cannot decrypt it, since the ciphertext component is blinded by a secret key shared between the SC and the SP. In what follows we will show that the KGC can simply decrypt the encrypted stored data

by impersonating itself with an authorized SM. Actually, the KGC can generate the secret key for each arbitrary set of attributes. So if it takes part in the protocol, it first generates a valid token, and sends it to the SC and receives the partially decrypted message. As it generates a valid secret key and knows the associated random value of the issued $TokenGen\left(MPK, SK_{S_{j,u}}\right) \longrightarrow TK_{S_{j,u}}$, it can decrypt the message. Therefore, in contrast with what Hur has claimed, the KGC can easily decrypt each stored ciphertext.

5.3. Security Proves

In what follows, we prove both the security of the proposed scheme. To this end, we show the indistinguishability and unforgeability of the scheme through **Theorem 1** and **Theorem 2**, respectively.

Theorem 1. If the decisional q-BDHE is a computationally hard problem, then the proposed CP-ABSC scheme will be secure against CP-ABSC-IND-CCA.

Proof. Suppose there exists a polynomial-time adversary, $\mathcal{A}$, that can break the proposed scheme in the introduced security game in Definition 6 with the non-negligible advantage ε. Moreover, suppose that the adversary $\mathcal{A}$ chooses a challenge matrix $TM^*_{\ell^* \times d^*}$ such that $d^* \leq q$, and let the service provider SP_x be the signer. We will show that how a probabilistic polynomial-time (PPT) adversary, $\mathcal{B}$, can solve the decisional q-BDHE problem with an advantage of at least $\frac{\varepsilon}{2}$. Actually, the adversary B plays the role of a challenger for the adversary $\mathcal{A}$ and exploits it to solve the mentioned problem.

Let $\mathbb{G}$,$\mathbb{G}_1$ and $\mathbb{G}_2$ be three cyclic groups of prime order p, and g be a generator of $\mathbb{G}$. The challenger $\mathcal{C}$ of the decisional q-BDHE problem first chooses $s, a \in_r Z_p^*$ uniformly at random and then flips a fair binary coin, $\varphi \in_r \{0,1\}$, outside $\mathcal{B}$'s view. If $\varphi = 0$, the challenger $\mathcal{C}$ sets $T = e(g,g)^{a^{q+1}s}$; otherwise, it sets $T = R$, where R is a random element of group $\mathbb{G}_1$. The challenger $\mathcal{C}$ sends T and according to definition 2 vector $\vec{y} = \{g, g^s, g^a, g^{a^2}, \dots g^{a^q}, g^{a^{q+2}}, \dots, g^{a^{2q}}\}$ to the adversary $\mathcal{B}$.

Initialization: In this phase, the adversary $\mathcal{B}$ sets the universe attribute set U, a collision-resistant hash function $H: \{0,1\} \longrightarrow Z_p^*$ and the security parameter $\mathcal{N}$. Also, she receives the challenge access structure (TM^*, ρ^*), and computes the challenge keys $\left(K_{sign_{x'}}, K_{ver_x}\right)$. Then, she sets the public parameters of the system as follows.

The adversary $\mathcal{B}$ implicitly by letting $\alpha = \alpha' + a^{q+1}$ as one part of master key, which is unknown by $\mathcal{B}$, sets the public parameters Y by computing $e(g^a, g^{a^q}) \times e(g,g)^{\alpha'}$, where a, q are chosen in the decisional q-BDHE problem, and α' is an integers which is randomly chosen in Z_p^* by the adversary $\mathcal{B}$. As a result $Y = e(g,g)^{(a^{q+1}+\alpha')} = e(g,g)^{\alpha}$. Also, the adversary $\mathcal{B}$ chooses a random value $z_i \in_r Z_p^*, 1 \leq i \leq |U|$ for each attribute in the universal attribute set. If the i^{th} row of the matrix $TM^*_{\ell^* \times d^*}$ corresponds to the attribute $x \in S_j$, where S_j is a set of attributes which satisfies the access structure Υ^*, then the adversary $\mathcal{B}$ sets the public parameter h_x as:

$$h_x = g^{z_x} g^{aTM^*_{i,1} + a^2 TM^*_{i,2} + \cdots + a^{d^*} TM^*_{i,d^*}} \qquad (20)$$

Otherwise, it sets $h_x = g^{z_x}$.

In this way, the master public key which is given to the adversary $\mathcal{A}$ is $MPK = \{g, H, e(g,g)^{\alpha}, g^{\gamma}, g^a, h_1, \dots, h_{|U|}\}$, and the master secret key which is kept by the adversary $\mathcal{B}$ is $MSK = \{\gamma, \alpha'\}$.

Query Phase 1. During this phase, the adversary $\mathcal{A}$ requests queries and the adversary $\mathcal{B}$ answers them as follows:

$\boldsymbol{O_{KeyGen}(S_j) \rightarrow \left(SK_{S_j}, K_{ver_u}\right)}$**:** The adversary $\mathcal{A}$ has access to this oracle which is provided by the adversary $\mathcal{B}$, to receive the private key corresponding to

the attribute set S_j and verification key from the adversary $\mathcal{B}$. First, the challenger $\mathcal{B}$ chooses a unique and random number $r_{su} \in_r Z_p^*$ and then calculates $K_{sign_u} = e(g,g)^{\frac{(\alpha' + r_{su})}{\gamma}} \times e\left(g^{a^q}, g^a\right)^{\frac{1}{\gamma}}$ and publishes its corresponding verification key $K_{ver_u} = e(g,g)^{r_{su}}$. Assume that the adversary $\mathcal{A}$ requests a private key for a set S_j, where S_j does not satisfy Υ^*. Then, the adversary $\mathcal{B}$ finds a vector $\vec{\omega} = (\omega_1, \omega_2, \dots, \omega_{d^*}) \in Z_p^*$ such that its first component is an arbitrary non-zero element in Z_p^* and $\sum_{i \in \mathcal{J}} \omega_i TM_i = (1,0,\dots,0)_{1\times d^*}$ where $\mathcal{J} = \{i : \rho(i) \in S_j\}$. According to the Definition 2, the vector $\vec{\omega}$ definitely exists. Then, the adversary $\mathcal{B}$ generates the private keys as follows:

The adversary $\mathcal{B}$ chooses the random value $r \in_r Z_p^*$ and sets $K_u' = \mathrm{g}^{\mathrm{r}} \prod_{\mathrm{i}=1}^{\mathrm{d}^*} \left(\mathrm{g}^{\mathrm{a}^{\mathrm{q}+1-\mathrm{i}}}\right)^{\omega_\mathrm{i}} = \mathrm{g}^{\mathrm{t_u}}$, where $\mathrm{t_u}$ is implicitly defined as:

$$\mathrm{t_u} = \mathrm{r} + \omega_1 \mathrm{a}^{\mathrm{q}} + \omega_2 \mathrm{a}^{\mathrm{q}-1} + \cdots + \omega_{\mathrm{d}^*} \mathrm{a}^{\mathrm{q}-\mathrm{d}^*+1} \tag{21}$$

Then, the adversary $\mathcal{B}$ computes $\mathrm{K_u}$ as:

$$\mathrm{K_u} = \mathrm{g}^{\alpha'} \mathrm{g}^{\mathrm{ar}} \prod_{\mathrm{i}=2}^{\mathrm{d}^*} \left(\mathrm{g}^{\mathrm{a}^{\mathrm{q}+2-\mathrm{i}}}\right)^{\omega_\mathrm{i}} = \mathrm{g}^{\alpha' - \mathrm{a}^{\mathrm{q}+1}\omega_1} \mathrm{g}^{\mathrm{a t_u}} \tag{22}$$

Let $\omega_1 = -1$, then $K_u = g^{\alpha} g^{a t_u}$. Now, the adversary $\mathcal{B}$ has to produce secret keys for non-authorized sets of attributes requested by the adversary $\mathcal{A}$. The secret key for each set of attributes is composed of number of components $K_{x_u}, \forall\, x \in S_j$. For each $x \in S_j$ that is not used in the access structure, such that $\rho^*(i) = x$, the adversary $\mathcal{B}$ simply let $K_{x_u} = K_u'^{z_x}$. Otherwise, it computes K_{x_u} as follows:

$$K_{x_u} = K_u'^{z_x} \prod_{j=1}^{d^*} \left(g^{a^j . r} \prod_{\substack{k=1 \\ k \neq j}}^{d^*} \left(g^{a^{q+1+j-k}}\right)^{\omega_k}\right)^{TM_{i,j}^*} \tag{23}$$

The private key corresponding to the attribute set S_j is $SK_{S_j,u} = (K_u, K_u', K_{x_u}\ \forall\, x \in S_j, K_{sign_u})$.

$\boldsymbol{O_{Signcryption}(m, (TM_{\ell\times d}, \rho))} \rightarrow \boldsymbol{CT}$: The adversary $\mathcal{A}$ has access to this oracle which is provided by the adversary $\mathcal{B}$. By using this oracle, the adversary $\mathcal{A}$ can adaptively make signcryption requests. Assume that the adversary $\mathcal{A}$ queries for the signcryption of the message m under access structure matrix $(TM_{\ell\times d}, \rho)$. The adversary $\mathcal{B}$ generates the ciphertext CT as follows:

$$C = m.T.e\left(g^s, g^{\alpha'}\right),\ C' = g^s \text{ and } C'' = (g^s)^{\gamma} \tag{24}$$

Where, T is the challenge term. Intuitively, the adversary $\mathcal{B}$ chooses the random values $y_2', \dots, y_{d^*}' \in_r Z_p^*$ and shares the secret s using the vector $\vec{v}$ as follow:

$$\vec{v} = \left\{s, sa + y_2', sa^2 + y_3', \dots, sa^{d^*-1} + y_d'\right\} \in Z_p^* \tag{25}$$

Also, it chooses the random values $r_1', \dots, r_d' \in_r Z_p^*$ and generates the challenge ciphertext components C_i and D_i for $i = 1, \dots, d$ as follows:

$$D_i = g^{-r_i'} g^s = g^{s - r_i'} \tag{26}$$

$$C_i = h_x^{r_i'} \left(\prod_{j=2,\dots,d} (g^a)^{TM_{i,j}^* y_j'} \right) \times (g^s)^{-z_x} \tag{27}$$

Where $x = \rho^*(i)$ and $h_x = g^{z_x} g^{aTM_{i,1}^* + a^2 TM_{i,2}^* + \cdots + a^{d^*} TM_{i,d^*}^*}$. Finally, the ciphertext is denoted by:

$$CT^* = \left((TM^*, \rho^*), C, C', C'', C_i, D_i, \pi, \Omega\right) \tag{28}$$

$$\delta = e'\left(e(C'', g), e(g,g)^{\hbar}\right), \pi = H(\delta | m)\,, \Omega = e(g,g)^{\hbar}\left(K_{sign_u}\right)^{\pi}, h \in_r Z_p^* \tag{29}$$

Where $\{\pi, \Omega\}$ are the signature components of m that is based on K_{sign_u}, and they are generated by executing the algorithm $Signcryption\left(MPK, m, K_{sign_u}, (TM^*, \rho^*)\right) \rightarrow CT^*$.

$\boldsymbol{O_{Designcryption}\left(CT^*, PD_{S_j}\right) \rightarrow m'}$**:** The adversary $\mathcal{A}$ has access to this oracle which is provided by the adversary $\mathcal{B}$, to receive the designcryption of the ciphertext CT^* by providing the ciphertext CT^* and partially designcrypted PD_{S_j}, which are selected by $\mathcal{A}$. The adversary $\mathcal{B}$ runs the algorithm $Designcryption\left(MPK, PD_{S_j}, K_{ver_x}\right) \rightarrow m'$ to designcrypt CT^* and forwards the output m' to the adversary $\mathcal{A}$.

Challenge. In this phase, we build the challenge ciphertext. The adversary $\mathcal{A}$ selects and sends two equal length plaintexts m_0^* and m_1^*. The adversary $\mathcal{B}$ flips a fair binary coin $b \in \{0,1\}$ and signcrypts m_b^* under challenging matrix of access structure (TM^*, ρ^*) as follows:

$$C = m_b^*.T.e\left(g^s, g^{\alpha'}\right), C' = g^s \text{ and } C'' = (g^s)^{\gamma} \tag{30}$$

Where, T is the challenge term. Intuitively, the adversary $\mathcal{B}$ chooses the random values $y_2', \ldots, y_{d^*}' \in_r Z_p^*$ and shares the secret s using the vector $\vec{v}$ as follow:

$$\vec{v} = \left\{s, sa + y_2', sa^2 + y_3', \ldots, sa^{d^*-1} + y_{d^*}'\right\} \in Z_p^* \tag{31}$$

Also, it chooses the random values $r_1', \ldots, r_{d^*}' \in_r Z_p^*$ and generates the challenge ciphertext components C_i and D_i for $i = 1, \ldots, d^*$ as follows:

$$D_i = g^{-r_i'} g^s = g^{s - r_i'} \tag{32}$$

$$C_i = h_x^{r_i'} \left(\prod_{j=2,\ldots,d} (g^a)^{TM_{i,j}^* y_j'} \right) \times (g^s)^{-z_x} \tag{33}$$

Where $x = \rho^*(i)$ and $h_x = g^{z_x} g^{aTM_{i,1}^* + a^2 TM_{i,2}^* + \cdots + a^{d^*} TM_{i,d^*}^*}$.

If $\varphi = 0$, then $T = e(g,g)^{a^{q+1} s}$, then the ciphertext component C is:

$$C = m_b^*.e(g,g)^{a^{q+1} s}.e\left(g^s, g^{\alpha'}\right) \tag{34}$$

This indicates that the ciphertext is valid for the message m_b^* under the access structure (TM^*, ρ^*). If $\varphi = 1$, then $T = R$ and the ciphertext component C is:

$$C = m_b^*.R.e\left(g^s, g^{\alpha'}\right) \tag{35}$$

Since R is a random element in group $\mathbb{G}_1$, thus from the view of $\mathcal{A}$ ciphertext component C is also a random element in group $\mathbb{G}_1$ and the message contains no information about m_b^*. The challenge ciphertext is denoted by:

$$CT^* = \left((TM^*, \rho^*), C, C', C'', C_i, D_i, \pi, \Omega\right) \tag{36}$$

$$\delta = e'\left(e(C'', g), e(g,g)^{\hbar}\right), \pi = H(\delta | m_b^*), \Omega = e(g,g)^{\hbar}\left(K_{sign_x}\right)^{\pi}, h \in_r Z_p^* \tag{37}$$

Where $\{\pi, \Omega\}$ are the ciphertext components of m_b^* that are based on K_{sign_x}, and they are generated by executing the algorithm $Signcryption\left(MPK, m_b^*, K_{sign_x}, (TM^*, \rho^*)\right) \rightarrow CT^*$.

Query Phase 2. After receiving CT^*, the adversary $\mathcal{A}$ can make polynomially bounded number of queries adaptively in the same way as Phase 1 except that the adversary $\mathcal{A}$ cannot query the tuple $\left(CT^*, S_j\right)$ to the oracle $O_{Designcryption}\left(CT^*, S_j\right) \rightarrow m'$ if $\Upsilon^*\left(S_j\right) = 1$, also cannot query the challenge messages $m_\emptyset$, $\emptyset = \{0,1\}$, to the oracle $O_{Signcryption}\left(m_\emptyset, (TM_{\ell^* \times d^*}^*, \rho^*)\right) \rightarrow CT_\emptyset^*$.

Guess. Let b' and φ' be the values that are guessed respectively about the value of b by the adversary $\mathcal{A}$ and the value of φ by the adversary $\mathcal{B}$. If $b' = b$, the adversary $\mathcal{B}$ outputs $\varphi' = 0$, which indicates that it receives a q-BDHE tuple. Otherwise, the adversary $\mathcal{B}$ outputs $\varphi' = 1$, which indicates that it receives a random tuple. When $\varphi = 1$, the adversary $\mathcal{A}$ obtains no information about b. So

we have $Pr[b' = b|\varphi = 1] = \frac{1}{2}$. On the other side, when $b' \neq b$, the guess value by the adversary B is $\varphi' = 1$, so we have $Pr[\varphi' = \varphi|\varphi = 1] = \frac{1}{2}$. When $\varphi = 0$, the adversary $\mathcal{A}$ has won the game because she has received the truly signcryption of m_b^*. In this situation, the advantage of the adversary $\mathcal{A}$ is defined as ε. Thus, we have $Pr[b' = b|\varphi = 0] = \varepsilon + \frac{1}{2}$. As the adversary $\mathcal{B}$ correctly guesses the value of φ when $\varphi = 0$, we have $Pr[\varphi' = \varphi|\varphi = 0] = \varepsilon + \frac{1}{2}$. Therefore, the overall advantage of the adversary $\mathcal{B}$ in solving the decisional q-BDHE problem is as follows:

$$\begin{aligned}&\big[Pr(\varphi = 0) \times Pr[\varphi' = \varphi|\varphi = 0] + Pr(\varphi = 1) \times \\ &Pr[\varphi' = \varphi|\varphi = 1]\big] - 1/2 = \tfrac{1}{2} Pr[\varphi' = \varphi|\varphi = 0] + \\ &\tfrac{1}{2} Pr[\varphi' = \varphi|\varphi = 1] - \tfrac{1}{2} = \tfrac{1}{2}\varepsilon.\end{aligned} \tag{38}$$

Therefore, the adversary $\mathcal{B}$ can play the decisional q-BDHE game with non-negligible advantage $\frac{\varepsilon}{2}$.

Theorem 2. If the decisional q-BDHE is computationally a hard problem, then the proposed CP-ABSC scheme is unforgeable against chosen access policy and message attacks.

Proof. Suppose that there exists a polynomial-time adversary, $\mathcal{A}$, that can find a forge for the scheme in the selective security game with the non-negligible advantage ε. Moreover, suppose the adversary $\mathcal{A}$ chooses a challenge matrix $(TM^*_{\ell^* \times d^*}, \rho^*)$ such that $d^* \leq q$, which extracted from access tree Υ^*, and let $\left(K_{sign_x}, K_{ver_x}\right)$ are the challenge secret keys. We will show that how a PPT adversary, $\mathcal{B}$, can be constructed based on the algorithm $\mathcal{A}$ to solve the decisional q-BDHE problem with a non-negligible advantage of at least $\frac{\varepsilon'}{2}$. Actually, the adversary $\mathcal{B}$ plays the role of a challenger for the adversary $\mathcal{A}$ and exploits it to solve the mentioned problem.

Let $\mathbb{G}$,$\mathbb{G}_1$ and $\mathbb{G}_2$ be three cyclic groups of prime order p, and g be a generator of $\mathbb{G}$. The challenger $\mathcal{C}$ of the decisional q-BDHE problem first chooses $s, a \in_r Z_p^*$ uniformly at random and then flips a fair binary coin, $\varphi \in_r \{0,1\}$, outside $\mathcal{B}$'s view. If $\varphi = 0$, the challenger $\mathcal{C}$ sets $T = e(g,g)^{a^{q+1}s}$; otherwise, it sets $T = R$, where R is a random element of group $\mathbb{G}_1$. The challenger $\mathcal{C}$ sends T and according to definition 2 vector $\vec{y} = \{g, g^s, g^a, g^{a^2}, \dots g^{a^q}, g^{a^{q+2}}, \dots, g^{a^{2q}}\}$ to the adversary $\mathcal{B}$.

Initialization: In this phase, the adversary $\mathcal{B}$ sets the universe attribute set U, a collision-resistant hash function $H: \{0,1\} \longrightarrow Z_p^*$ and the security parameter $\mathcal{N}$. Also, she receives the challenge access structure (TM^*, ρ^*), and computes the challenge keys $\left(K_{sign_x}, K_{ver_x}\right)$. Then, she sets the public parameters of the system as follows.

The adversary $\mathcal{B}$ implicitly by letting $\alpha = \alpha' + a^{q+1}$ as one part of master key, which is unknown by $\mathcal{B}$, sets the public parameters Y by computing $e(g^a, g^{a^q}) \times e(g,g)^{\alpha'}$, where a, q are chosen in the decisional q-BDHE problem, and α' is an integers which is randomly chosen in Z_p^* by the adversary $\mathcal{B}$. As a result $Y = e(g,g)^{(a^{q+1}+\alpha')} = e(g,g)^{\alpha}$. Also, the adversary $\mathcal{B}$ chooses the random values $z_i \in_r Z_p^*, 1 \leq i \leq |U|$ for each attribute in the universal attribute set. If the i^{th} row of the matrix $TM^*_{\ell^* \times d^*}$ corresponds to the attribute $x \in S_j$, where S_j is a set of attributes which satisfies the access structure Υ^*, then the adversary $\mathcal{B}$ sets the public parameter h_x as:

$$h_x = g^{z_x} g^{aTM^*_{i,1} + a^2 TM^*_{i,2} + \cdots + a^{d^*} TM^*_{i,d^*}} \tag{39}$$

Otherwise, it sets $h_x = g^{z_x}$. Also, the adversary $\mathcal{B}$ computes the challenge data owner's key pair $\left(K_{sign_x}, K_{ver_x}\right)$ as follows:

$$K_{sign_x} = e(g,g)^{\frac{\alpha'}{\gamma}}.(T)^{\frac{1}{\gamma}} \quad (40)$$

$$K_{ver_x} = T.e\left(g^{a^q}, g^a\right)^{-1} \quad (41)$$

In this way, the master public key which is given to the adversary $\mathcal{A}$ is $MPK = \{g, H, e(g,g)^{\alpha}, g^{\gamma}, g^a, h_1, \dots, h_{|U|}, K'_{ver}\}$,

and the master secret key which is kept by the adversary $\mathcal{B}$ is $MSK = \{\gamma, \alpha'\}$.

Query Phase. The adversary $\mathcal{B}$ answers to the $\mathcal{A}$'s queries as follows:

$\boldsymbol{O_{KeyGen}(S_j) \rightarrow \left(SK_{S_j}, K_{ver_u}\right)}$**:** By using this oracle, the adversary $\mathcal{A}$ can adaptively request private keys from the adversary $\mathcal{B}$. The challenger $\mathcal{B}$ chooses a unique and random number $r_{su} \in_r Z_p^*$ and then calculates $K_{sign_u} = e(g,g)^{\frac{(\alpha'+r_{su})}{\gamma}} \times e\left(g^{a^q}, g^a\right)^{\frac{1}{\gamma}}$ and publishes its corresponding verification key $K_{ver_u} = e(g,g)^{r_{su}}$. Assume that the adversary $\mathcal{A}$ requests a private key for a set S_j, where S_j does not satisfy Υ^*. Then, the adversary $\mathcal{B}$ finds a vector $\vec{\omega} = (\omega_1, \omega_2, \dots, \omega_{d^*}) \in Z_p^*$ such that its first component is an arbitrary non-zero element in Z_p^* and $\sum_{i \in \mathcal{I}} \omega_i TM_i = (1,0,\dots,0)_{1\times d^*}$ where $\mathcal{I} = \{i : \rho(i) \in S_j\}$. According to Definition 2, the vector $\vec{\omega}$ definitely exists. Then, the adversary $\mathcal{B}$ generates the private keys as follows:

The adversary $\mathcal{B}$ chooses the random value $r \in_r Z_p^*$ and sets $K'_u = \mathrm{g}^{\mathrm{r}} \prod_{\mathrm{i}=1}^{\mathrm{d}^*} \left(\mathrm{g}^{\mathrm{a}^{\mathrm{q}+1-\mathrm{i}}}\right)^{\omega_{\mathrm{i}}} = \mathrm{g}^{\mathrm{t_u}}$, where $\mathrm{t_u}$ is implicitly defined as:

$$\mathrm{t_u} = \mathrm{r} + \omega_1 \mathrm{a}^{\mathrm{q}} + \omega_2 \mathrm{a}^{\mathrm{q}-1} + \cdots + \omega_{\mathrm{d}^*} \mathrm{a}^{\mathrm{q}-\mathrm{d}^*+1} \quad (42)$$

Then, the adversary $\mathcal{B}$ computes $\mathrm{K_u}$ as:

$$\mathrm{K_u} = \mathrm{g}^{\alpha'} \mathrm{g}^{\mathrm{ar}} \prod_{\mathrm{i}=2}^{\mathrm{d}^*} \left(\mathrm{g}^{\mathrm{a}^{\mathrm{q}+2-\mathrm{i}}}\right)^{\omega_{\mathrm{i}}} = \mathrm{g}^{\alpha' - \mathrm{a}^{\mathrm{q}+1}\omega_1} \mathrm{g}^{\mathrm{at_u}} \quad (43)$$

Let $\omega_1 = -1$, then $K_u = g^{\alpha} g^{at_u}$. Now, the adversary $\mathcal{B}$ has to produce secret keys for non-authorized sets of attributes requested by the adversary $\mathcal{A}$. The secret key for each set of attributes is composed of number of components $K_{x_u}, \forall\, x \in S_j$. For each $x \in S_j$ that is not used in the access structure, such that $\rho^*(i) = x$, the adversary $\mathcal{B}$ simply let $K_{x_u} = {K'_u}^{z_x}$. Otherwise, it computes K_{x_u} as follows:

$$K_{x_u} = {K'_u}^{z_x} \prod_{j=1}^{d^*} (g^{a^j.r} \prod_{\substack{k=1 \\ k\neq j}}^{d^*} \left(g^{a^{q+1+j-k}}\right)^{\omega_k})^{TM^*_{i,j}} \quad (44)$$

The private key corresponding to the attribute set S_j is $SK_{S_j,u} = (K_u, K'_u, K_{x_u}\ \forall\, x \in S_j, K_{sign_u})$.

$\boldsymbol{O_{Signcryption}(m, (TM_{\ell\times d}, \rho)) \rightarrow CT}$: Assume that the adversary $\mathcal{A}$ queries for the signcryption of the message m under the access structure matrix $(TM_{\ell\times d}, \rho)$. The adversary $\mathcal{B}$ generates the ciphertext CT as follows:

$$C = m.T.e\left(g^s, g^{\alpha'}\right),\ C' = g^s \text{ and } C'' = (g^s)^{\gamma} \quad (45)$$

Where, T is the challenge term such that the adversary $\mathcal{B}$ has received from the challenger of q-BDHE problem. Intuitively, the adversary $\mathcal{B}$ chooses the random values $y'_2, \dots, y'_{d^*} \in_r Z_p^*$ and shares the secret s using the vector $\vec{v}$ as follow:

$$\vec{v} = \left\{s, sa + y'_2, sa^2 + y'_3, \dots, sa^{d^*-1} + y'_d\right\} \in Z_p^* \quad (46)$$

Also, it chooses the random values $r'_1, \dots, r'_d \in_r Z_p^*$ and generates the challenge ciphertext components C_i and D_i for $i = 1, \dots, d$ as follows:

$$D_i = g^{-r'_i} g^s = g^{s-r'_i} \quad (47)$$

$$C_i = h_x^{r'_i} \left(\prod_{j=2,\dots,d} (g^a)^{TM^*_{i,j} y'_j} \right) \times (g^s)^{-z_x} \quad (48)$$

Where $x = \rho^*(i)$ and $h_x = g^{z_x} g^{aTM^*_{i,1} + a^2 TM^*_{i,2} + \cdots + a^{d^*} TM^*_{i,d^*}}$. Finally, the ciphertext is denoted by:

$$CT = \left((TM^*, \rho^*), C, C', C'', C_i, D_i, \pi, \Omega\right) \quad (49)$$

$$\delta = e'\left(e(C'', g), e(g, g)^{\hbar}\right), \pi = H(\delta | m), \quad (50)$$

$$\Omega = e(g, g)^{\hbar}\left(K_{sign_u}\right)^{\pi}, h \in_r Z_p^*$$

Where $\{\pi, \Omega\}$ are the signature components of m that is based on K_{sign_u}, and they are generated by executing the algorithm $Signcryption\left(MPK, m, K_{sign_u}, (TM^*, \rho^*)\right) \rightarrow CT$.

$\boldsymbol{O_{Designcryption}(CT^*, S_j) \rightarrow m'}$**:** Assume the adversary $\mathcal{A}$ queries for the designcryption of the ciphertext CT' by providing an attribute set S_j. The adversary $\mathcal{B}$, first executes the oracle $O_{KeyGen}(S_j) \rightarrow \left(SK_{S_j}, K_{ver_x}\right)$ oracle to generate the corresponding private keys $SK_{S_j, x}$. Then, it generates corresponding tokens, calls $O_{PartialDesigncryption}(S_j, CT') \rightarrow PD'_{S_j}$, and runs the algorithm $Designcryption\left(MPK, PD'_{S_j}, K_{ver_x}\right) \rightarrow m'$ to designcrypt CT' and forwards the output m' to the adversary $\mathcal{A}$.

Forgery phase: The adversary $\mathcal{A}$ submits a valid forgery tuple $\{CT^* = \left((TM^*, \rho^*), C, C', C'', C_i, D_i, \pi^*, \Omega^*\right), m^*, K_{ver_x}\}$ for the challenge secret key K_{sign_x}. Then CT^* satisfies two properties:

(i) $Designcryption\left(CT^*, PD_{S_j}, K_{ver_x}\right) \rightarrow m^* \neq \perp$, here $\Upsilon^*(S_j) = 1$.

(ii) The adversary $\mathcal{A}$ has never queried to signcryption oracle with the tuple $\left((TM^*, \rho^*), m^*, K_{sign_x}\right)$.

Now, the adversary $\mathcal{B}$ can solve the q-BDHE problem with a no negligible advantage.

$$\begin{aligned} &\left[Pr(\varphi = 0) \times Pr[m' = m^* | \varphi = 0] + Pr(\varphi = 1) \times \right. \\ &\left. Pr[m' = m^* | \varphi = 1]\right] = \frac{1}{2} Pr[m' = m^* | \varphi = 0] + \\ &\frac{1}{2} Pr[m' = m^* | \varphi = 1] = \frac{1}{2}\varepsilon + \frac{1}{2} negl = \frac{1}{2}(\varepsilon + negl) = \frac{1}{2}\varepsilon'. \end{aligned} \quad (51)$$

Therefore, the adversary $\mathcal{B}$ can break the decisional q-BDHE problem with non-negligible advantage $\frac{\varepsilon'}{2}$. ■

6. Performance Analysis and Improvement

6.1. Performance Analysis

In this section, we evaluate the performance and efficiency of the proposed scheme by computing its computational complexity. For this aim, we compare the ciphertext size, the private and public key size and the length of the generated token with the previous schemes. Table IV presents a list of notations that we use to evaluate the efficiency, and Table II compares the efficiency of the proposed scheme with the Bethencourt et al.'s CP-ABE scheme [9], Hur's CP-ABE scheme

Table 2. Comparison of efficiency and functionality

scheme	ABSC/ABE	Access structure	Secret key size	Ciphertext size	Token size	Message Authentication
[9]	CP-ABE	Threshold policy	$(2v+1)C$	$(2t+1)C + C_1 + C_{\mathcal{T}}$	$-$	No
[7]	CP-ABE	Threshold policy	$(3v+1)C$	$(2t+1)C + C_1 + C_{\mathcal{T}}$	$(3t)C_1$	No
[19]	KP-ABSC	LSSS with AND/OR policy	$(2v+4)C$	$(2t+4)C + C_1 + C_{\mathcal{T}}$	$-$	Yes
Our scheme	CP-ABSC	LSSS with Threshold policy	$(v+3)C$	$(2t+4)C + C_1 + C_{\mathcal{T}}$	$(t+3)C$	Yes

[7], and Rao's CP-ABSC scheme [19]. Also, Table III compares the computational cost of the proposed scheme with the mentioned schemes. It is necessary to mention that, in the Hur's scheme, each ciphertext is partially decrypted by its corresponding token, while in our proposal, the SC can find all of the authorized documents, and partially designcrypts them after receiving just one token. This results in the reduction of the communication and computational overhead of the proposed scheme.

Table 4. Notations which are frequently used in the performance evaluation

C_p	bit length of an element in Z_p^*.
C	bit length of an element in $\mathbb{G}$.
C_1	bit length of an element in $\mathbb{G}_1$.
$C_{\mathcal{T}}$	bit length of an access policy in the ciphertext.
t	the number of attributes associated with the ciphertext.
v	the number of attributes associated with private key of a user.
n	Minimum number of decryption attributes required to recover a message.

Another notable feature of the proposed scheme is that there is no need for pairing computation in the final decryption. In addition, the SM can verify the authenticity of the received results while the mentioned schemes do not support this feature. Also, we evaluate the computational complexity of the algorithms presented in the proposed scheme, and compare the results with other schemes [7, 9] in terms of three parts: (1) for a SP to signcrypt a plaintext with a set of attributes, (2) for a SM to generate a token, and (3) for a SM with own attributes to decrypt a ciphertext. We considered a 3 GHz with a quad-core processor, which all of the cryptographic operations were implemented using the PBC library version 0.4.18 [29]. With implementation requirements in [7], the time execution required for computation of pairing, exponentiation in group $\mathbb{G}$ and $\mathbb{G}_1$ in millisecound time scale are respectively 2.9, 1.0 and 0.2 ms.

Figure 2 illustrates the computational cost of signcryption/encryption, token generation and designcryption/decryption algorithms in all mentioned schemes. As shown in Figure 2, as far as the number of involved attributes in access structure required for recovering a message increases, the running time of

Table 3. Comparison of computational cost

Scheme	Signcryption/Encryption cost			Token cost			Designcryption /Decryption cost			
	Exp.		Pairing	Exp.		Pairing	Exp.		Pairing	
	in $\mathbb{G}$	in $\mathbb{G}_1$		in $\mathbb{G}$	in $\mathbb{G}_1$		in $\mathbb{G}$	in $\mathbb{G}_1$	Decryptio	Verify
[9]	$2t+1$	1	0	$-$	$-$	$-$	0	$\log t$	$2n+1$	$-$
[7]	$2t+3$	1	$t+1$	$2n$	0	n	1	0	1	-
[19]	$6t+6$	1	0	$-$	$-$	$-$	$t+2n+2$	0	$t+5$	2
Our scheme	$3t+2$	4	2	$n+2$	0	0	0	2	0	5

encryption/ signcryption, token generation, and decryption/designcryption algorithms in our scheme is less increased than Hur [7] and BSW [9] schemes.

6.2. PUF-based instantiation of proposed CP-ABSC

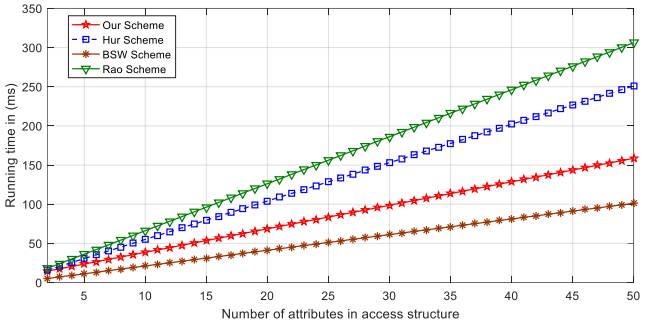
As shown in Table II, the size of the secret key which is planned to be stored in the memory of the intended device is reduced in comparison with [7] and [9]. However, we can also achieve to more decrement in the required storage and enhance the security of the proposed scheme with the favor of PUF. It should be mentioned that in the secure implementation of the proposed scheme, we have a restriction in the capacity of the memory for storing the secret keys [30]. In the proposed scheme and compared schemes, the size of the stored keys is related to the user's attributes which could impose a limitation in implementation. By embedding a PUF in each user's device, the secret keys can be generated during designcryption algorithm and there will be no need to store them in non-volatile memories. In what follows, we describe the required procedure for constructing the secret keys in more details.

To achieve this improvement, the KGC in the $KeyGen$ algorithm applies the challenge C_i to the user's PUF, PUF_u, and gets the corresponding response, $R_i^u = PUF_u(C_i)$. The generated Challenge-Response Pair (CRP) is stored in the KGC. To generate the secret key of the user u, SK_u, the KGC first chooses two random numbers $t_u \in_r Z_p^*$ and $r_{su} \in_r Z_p^*$, corresponding to the user u and computes $K_u = g^{\alpha} g^{\beta t_u R_i^u}$, $K_{sign_u} = e(g,g)^{\frac{(\alpha + r_{su})}{\gamma}}$, $K'_u = g^{t_u}$ and $K_{x_u} = {h_x}^{t_u}$ for each attribute, $x \in S_j$. Finally, the SK_u is generated as follows:

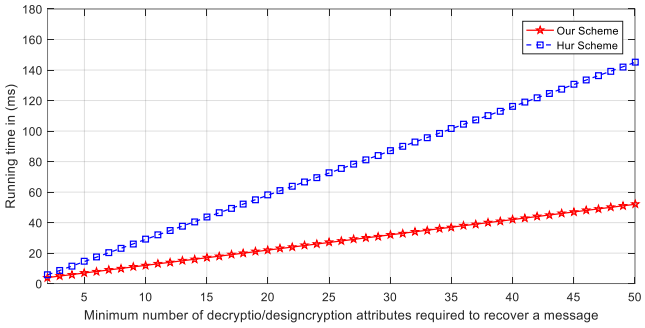
$$SK_{s_j,u} = \left(K_u, (K'_u)^{R_i^u}, \left(K_{x_u}\right)^{R_i^u}, K_{sign_u}\right) \qquad (52)$$

It must be noted that K_u and K_{sign_u} must be kept secret and are stored in the non-volatile memory of user's device. As mentioned in Definition 5, by applying the same challenge to different PUFs, different responses are generated. So, since the keys, $(K'_u)^{R_i^u}, \left(K_{x_u}\right)^{R_i^u}$ can only be generated by the user who has PUF_u, the values K'_u and K_{x_u} can be public and are published by the KGC when the user need to generate its secret key. In this way, instead of storing $(v+3)$ keys in the non-volatile memory of each device where v is the number of attributes, only 2 keys must be stored. Also, since the response of each user's PUF-enabled device is unique and specific for each user, the KGC can be assured that the secret key, $SK_{s_j,u}$ is generated by user u.

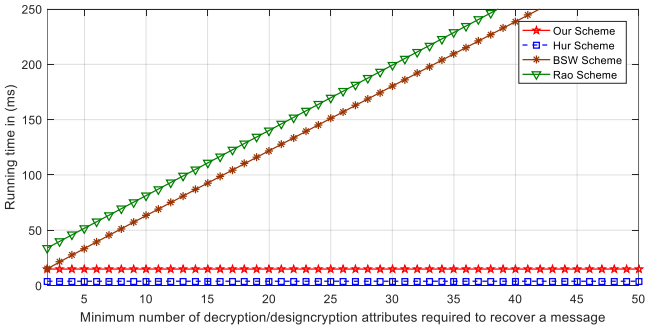
Moreover, by applying PUF-enabled devices, the proposed scheme will be secure against non-volatile memory attackers who aim to extract the secret information from non-volatile memories [25]. In the improved version of our proposed scheme, two components of the secret key are stored and the other ones are generated online using PUF. So, only half of the secret key's components can be achieved by a non-volatile memory attacker. As a result, the improved version is fully memory leakage resilient [25].



a) CC in encryption/signcryption phase



b) CC in TokenGen phase



c) CC in decryption/designcryption phase

Fig. 2. Comparison of Computational Cost (CC)

7. Conclusion

Since the equipment commonly used in the SG are considered to have limited computational resources, performing an efficient, authenticated and secure data sharing process in such networks applies some difficulties. Therefore, the typical existing attribute-based cryptographic primitives cannot be directly applicable. Consequently, the partial designcryption process is delegated to a storage center with powerful computational and storage resources. In this paper, we proposed a Ciphertext-policy attribute-based signcryption for data sharing in the SG. We imply that both the indistinguishability and unforgeability of our proposed scheme are reduced to q-BDHE problem; also we can reduce the required secure memory with the help of PUF-based secure instantiation. The performance evaluation and comparison results show the practical and deployable aspects of our proposed scheme.

REFERENCES

[1] Gharavi, Hamid, and Reza Ghafurian, eds. "Smart grid: The electric energy system of the future." *Vol. 99. IEEE,*: 917-921, 2011.

[2] Fang, Xi, Satyajayant Misra, Guoliang Xue, and Dejun Yang. "Smart Grid—The new and improved power grid: A survey." *IEEE communications surveys & tutorials* 14.4: 944-980, 2012.

[3] Wang, Wenye, and Zhuo Lu. "Cyber Security in the Smart Grid: Survey and Challenges." *Computer Networks* 57.5: 1344-1371, 2013.

[4] Huang, Qinlong, Zhaofeng Ma, Yixian Yang, Jingyi Fu, and Xinxin Niu. "EABDS: attribute-based secure data sharing with efficient revocation in cloud computing." *Chinese Journal of Electronics 24, no. 4 (2015):* 862-868.

[5] Bobba, Rakesh, Himanshu Khurana, Musab AlTurki, and Farhana Ashraf. "PBES: a policy based encryption system with application to data sharing in the power grid." *Proceedings of the 4th International Symposium on information, computer, and communications security*. ACM, 2009.

[6] Sedaghat, Seyyed Mahdi, Mohammad Hassan Ameri, Javad Mohajeri, and Mohammad Reza Aref. "An efficient and secure data sharing in Smart Grid: Ciphertext-policy attribute-based signcryption." *In Electrical Engineering (ICEE), 2017 Iranian Conference on, pp. 2003-2008*. IEEE, 2017.

[7] Hur, Junbeom. "Attribute-based secure data sharing with hidden policies in Smart Grid." *IEEE Transactions on Parallel and Distributed Systems* 24.11, 2171-2180, 2013.

[8] Hu, Chunqiang. "Privacy-Preserving and Secure Cryptographic Schemes for Wireless Applications." *Ph.D. diss., THE GEORGE WASHINGTON UNIVERSITY*, 2016.

[9] Bethencourt, John, Amit Sahai, and Brent Waters. "Ciphertext-policy attribute-based encryption." *symposium on security and privacy (SP'07)*. IEEE, 2007.

[10] Liang, Kaitai, and Willy Susilo. "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage." *IEEE Transactions on Information Forensics and Security 10, no. 9 (2015)*: 1981-1992.

[11] Lewko, Allison, and Brent Waters. "Decentralizing attribute-based encryption." *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer Berlin Heidelberg, 2011.

[12] So, Hayden K-H., Sammy HM Kwok, Edmund Y. Lam, and King-Shan Lui. "Zero-configuration identity-based signcryption scheme for the smart grid." In Smart Grid Communications (SmartGridComm), *2010 First IEEE International Conference on*, pp. 321-326. IEEE, 2010.

[13] Delavar, Mahshid, Sattar Mirzakuchaki, Mohammad Hassan Ameri, and Javad Mohajeri. "PUF-based solutions for secure communications in Advanced Metering Infrastructure (AMI)." *International Journal of Communication Systems*,2016.

[14] Lewko, Allison, and Brent Waters. "Decentralizing attribute-based encryption." *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer Berlin Heidelberg, 2011.

[15] Waters, Brent. "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization." *International Workshop on Public Key Cryptography*. Springer Berlin Heidelberg, 2011.

[16] Emura, Keita, Atsuko Miyaji, and Mohammad Shahriar Rahman. "Dynamic attribute-based signcryption without random oracles." *International Journal of Applied Cryptography* 2.3: 199-211, 2012.

[17] Gagné, Martin, Shivaramakrishnan Narayan, and Reihaneh Safavi-Naini. "Threshold attribute-based signcryption." *International Conference on Security and Cryptography for Networks*. Springer Berlin Heidelberg, 2010.

[18] Hu, Chunqiang, Xiuzhen Cheng, Zhi Tian, Jiguo Yu, Kemal Akkaya, and Limin Sun. "An Attribute-Based Signcryption Scheme to Secure Attribute-Defined Multicast Communications." *International Conference on Security and Privacy in Communication Systems*. Springer Int. Publishing, 2015.

[19] Rao, Y. Sreenivasa. "A secure and efficient Ciphertext-Policy Attribute-Based Signcryption for Personal Health Records sharing in cloud computing." *Future Generation Computer* Systems 67 : 133-151, 2017.

[20] Liu, Jianghua, Xinyi Huang, and Joseph K. Liu. "Secure sharing of personal health records in cloud computing: ciphertext-policy attribute-based signcryption." *Future Generation Computer Systems 52 (2015):* 67-76.

[21] A. Beimel et al. , "Secure schemes for secret sharing and key distribution.", *Technion-Israel Institute of Technology*, Faculty of Computer Science, 1996.

[22] Liu, Zhen, Zhenfu Cao, and Duncan S. Wong. "Efficient generation of linear secret sharing scheme matrices from threshold access trees." *Vol. 2010. IACR Cryptology ePrint Archive*, 2010. Available at: https://eprint.iacr.org/2010/374

[23] Boneh, Dan, and Matt Franklin. "Identity-based encryption from the Weil pairing." *Annual International Cryptology Conference*. Springer Berlin Heidelberg, 2001.

[24] Boneh, Dan, Xavier Boyen, and Eu-Jin Goh. "Hierarchical identity-based encryption with constant size ciphertext." *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer Berlin Heidelberg, 2005.

[25] B. Gassend, D. Clarke, M. van Dijk, S. Devadas, "Silicon Physical random functions", *in Proceedings of CCS, 2002*, pp. 148–160.

[26] Sahai, Amit, and Brent Waters. "Fuzzy identity-based encryption." *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer Berlin Heidelberg, 2005.

[27] Malone-Lee, John. "Identity-Based Signcryption." *IACR Cryptology ePrint Archive 2002* (2002): 98.

[28] Selvi, S. Sharmila Deva, et al. "ID-based signcryption scheme in standard model." *International Conference on Provable Security*. Springer Berlin Heidelberg, 2012

[29] The Pairing-Based Cryptography Library, http://crypto.stanford.edu/pbc

[30] Fouda, Mostafa M., Zubair Md Fadlullah, Nei Kato, Rongxing Lu, and Xuemin Sherman Shen. "A lightweight message authentication scheme for smart grid communications." *IEEE Transactions on Smart Grid 2, no. 4 :* 675-685, 2011.