Efficient Multi-Authority Attribute-Based Signcryption with Constant-Size Ciphertext

Yang Zhao, Ankang Ruan, Guohang Dan, Jicheng Huang and Yi Ding

2021 IEEE Conference on Dependable and Secure Computing (DSC) | 978-1-7281-7534-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/DSC49826.2021.9346249

***Abstract*—Recently, efficient fine-grained access mechanism has been studied as a main concern in cloud storage area for several years. Attribute-based signcryption (ABSC) which is logical combination of attribute-based encryption(ABE) and attribute-based signature(ABS), can provide confidentiality, authenticity for sensitive data and anonymous authentication. At the same time it is more efficient than previous "encrypt-then-sign" and "sign-then-encrypt" patterns. However, most of the existing ABSC schemes fail to serve for real scenario of multiple authorities and have heavy communication overhead and computing overhead. Hence, we construct a novel ABSC scheme realizing multi-authority access control and constant-size ciphertext that does not depend on the number of attributes or authorities. Furthermore, our scheme provides public verifiability of the ciphertext and privacy protection for the signcryptor. Specially, it is proven to be secure in the standard model, including ciphertext indistinguishability under adaptive chosen ciphertext attacks and existential unforgeability under adaptive chosen message attack.**

***Index Terms*—Multi-authority, Attribute-based Signcryption, Constant-Size, Public Verifiability, Signcryptor Privacy Protection.**

I. Introduction

In this age of advanced information transmission, the rapid development of cloud computing is making a huge difference to enterprises and individuals. More and more users tend to use cloud servers with high computing performance and high capacity to store and encrypt large amounts of private data [1], [2]. By storing data in cloud servers, users need not preserve local storage, which helps to save hardware and software deployment costs. Despite the benefits of cloud storage, the security of data access control remains a major concern because in most scenarios users store data containing sensitive information on cloud servers [3], [4] which are usually deemed to be untrusted. Furthermore, from the perspective of data owners sharing data with other users, it is more efficient to achieve fine-grained access on stored data [5], [6], [7], [8]. Taking Personal Health Record (PHR) system for example, the PHR provider can employ cloud servers to store sensitive data for relieving the pressure of local storage and client device computing. Without the guarantee of the security protection, unauthorized users can access and even modify PHR data stored in cloud servers. Moreover, if an attacker provides PHR service provider with fake PHR data, it may contaminate the real PHR data collected from patients. Therefore, ensuring that PHR data is inaccessible to unauthorized users designated by PHR managers, and ensuring that data acquired from patients is credible, are two key prerequisites for the PHR system to employ cloud servers safely.

Currently, Attribute-based Encryption (ABE) [9] has received extensive attentions, on account of fine-grained access on sensitive data and realizing data confidentiality. ABE includes Key-policy ABE (KP-ABE) [10], [11] and Ciphertext-policy ABE (CP-ABE) [12], [13]. In KP-ABE schemes, attributes and access policies are associated with ciphertexts and private keys, respectively. Contrastly, in CP-ABE schemes, attributes and access policies are associated with private keys and ciphertexts, respectively. Meanwhile, in a real PHR scenario, not only is data confidentiality and fine-grained access control required, but anonymous authentication is also needed for checking that whether the patient with the corresponding attribute corresponds to the data. An effective and flexible strategy achieving fine-grained access on sensitive data, anonymous authentication and confidentiality synchronously is Attribute-based Signcryption (ABSC) [14], [15]. The ABSC scheme is superior to "encrypt-then-sign" and "sign-then-encrypt" scheme in both communication and computing overhead because it makes use of ABE and Attribute-based signatures (ABS) [16]. Similarly, there are two models for ABSC, Key-policy ABSC (KP-ABSC) [14] and Ciphertext-policy ABSC (CP-ABSC) [15]. In KP-ABSC schemes, the data owners signcrypt the message with the encryption and signature attribute sets, then outsource ciphertexts to cloud servers. Subsequently, the data users unsigncrypt ciphertexts to check whether ciphertexts are sended by the data owners. And if the encryption attribute sets satisfy the decryption access structures of the data users, they recover the message. Contrastly, in CP-ABSC schemes, the signature and encryption access structures are associated with ciphertexts.

In spite of these advantages of the ABSC scheme, multiple authorities and overhead of communication and computing, remain to be considered in cloud storage. In some existing ABSC schemes, such as [17], [18], [19], [20], there is one fully credible authority which is in charge of secret keys distribution and attributes management. After applying attribute-base signcrption to the PHR healthcare system, the single credible authority model exposes two problems. First, the PHR is a social public welfare system with huge user scale, which brings heavy burden of user key generation and management to the authorization center and affects the efficiency of the system. Second, when the only authorization center is attacked, all users' private keys are likely to be leaked,

Y. Zhao, A. Ruan, G. Dan, J. Huang and Y. Ding (Corresponding author) are with the Network and Data Security Key Laboratory of Sichuan Province, University of Electronic Science and Technology of China, Chengdu, Sichuan, 610054 China; Y. Ding is also with Institute of Electronic and Information Engineering of UESTC in Guangdong, Guangdong, 523808, China.
E-mail: yi.ding@uestc.edu.cn (Yi Ding)

thus causing great security risks. Moreover, users' attributes can be managed by multiple trusted authorities in many real scenarios. Therefore, the attributes management and corresponding keys generation from one fully credible authority to multiple mutually independent authorities is a necessary condition. The other issue is about communication overhead and computing overhead. In the cloud computing scenario, due to the limited computing power of the user's device, we need to consider various expensive-cost operations in ABSC schemes, such as numerous bilinear pairings operations. At the same time, for bandwidth-limited applications, we also need to consider the size of ciphertext during transmission.

A. Related Work

- **Attribute-based Signcryption**
  In 2010, the earliest ABSC scheme was presented by Gagné et al. [15], including supporting threshold access policy and complete security analysis about data confidentiality and ciphertext unforgeability. Subsequently, many ABSC schemes [14], [17], [18], [19], [20], [21] are proposed. In the scheme [15], there exists a limiting condition that the access structure of signing must be determined during the setup phase. For the restriction, Emura et al. [22] presented dynamic ABSC scheme in 2011. This scheme can flexibly update the access structure without redistributing the user's signing key. Moreover, it provides public verifiability. It saves the computation overhead for the user to verify the validity of signature. In 2012, Chen et al. [17] made efforts to combine ABE and ABS schemes, then constructed the CP-ABSC scheme. Nevertheless, this scheme has some performance defects due to its inability to expose the public verification. In 2013, Han et al. [23] presented threshold ABSC scheme with constant-size ciphertexts. In 2014, based on KP-ABS model, Rao [14] presented a KP-ABSC scheme with constant-size ciphertexts. In 2015, Liu et al. [18] constructed ABSC scheme with expressive access structure for PHR system. However, it also failed to support public verifiability and was provably secure in random oracle model. Subsequently, in 2017, Rao [19] pointed out the problems in the security proof of scheme [18] and developed the CP-ABSC scheme supporting public verifability and shielding privacy information of the signcryptor. It provided public verifiability, message confidentiality, authenticity and privacy protection of signcryptor. Nevertheless, its computation overhead of the decryption phase is too high because of the huge number of exponential operations and bilinear pairings. In 2018, Deng et al. [24] presented CP-ABSC scheme supporting users in outsourcing complex computing in the decryption phase to cloud servers to reduce the computing overhead of terminal equipments. In 2019, Deng et al. [25] developed an innovative ABSC scheme with revocable cloud-assisted signcryption based on key-splitting technology and broadcast encryption. Nevertheless, there is a common weakness in all of the above ABSC schemes that fail to be applied in a practical multi-authority scenario.
- **Multi-authority Attribute-based Signcryption**
  To release limitations of single authority in real scenario, Chase et al. [26] put forward to Multi-authority ABE scheme (MA-ABE) involving multiple authorities. Subsequently, they constructed a ABSC scheme without central authority by using the pseudo random function and key-exchange method in [27]. Recently, many MA-ABE schemes have been published [28], [29], [30], [31]. Han et al. [28], [29] presented multi-authority CP-ABE (MACP-ABE) scheme without the central authority. However, this scheme cannot provide authentication. In [30] scheme, they constructed decentralized MA-ABE scheme by associating users' secret keys released by each authority with the users' global identification. The MA-ABE scheme constructed by Yang et al. [31] improved the performance of the scheme by outsourcing most of the computing process in the decryption phase to cloud servers. However, as the scheme exposes users' attributes information to semi-trusted cloud servers, there are potential security risks due to the users' attribute set related to the users' personal privacy. Subsequently, Meng et al. [32] presented a MA-ABSC scheme by combining identity signature and MA-ABE in a decentralized setting. However, the scheme failed to support expressive access structures due to the limitation on threshold predicates. In [33] scheme, Xu et al. developed MA-ABSC scheme which provides user's attribute privacy protection function and supports public verification and outsourcing decryption for ease the computation overhead for users. However, the size of ciphertext in the two schemes of MA-ABSC is dependent of the number of authorities and attributes, hence their communication overhead is especially heavy.

B. Our Contributions

We aim to develop a ABSC scheme for addressing the major issues with these existing ABSC schemes in cloud storage system, including the high communication overhead, the high computing overhead and incompatibilities in Multi-authority scenarios.

We develop Multi-authority Attribute-based Signcryption with Constant-size Ciphertext (MACSC-ABSC) which is combining access control for multiple authorities and the existing KP-ABSC as pointed out in [14]. For the multi-authority scenario, this scheme realizes that the increase in the number of authorities and attributes in the cloud storage system does not affect the ciphertext size. In brief, the ciphertext size remains constant. Meanwhile, the public verifiability for all trusted cloud servers and the signcrptor privacy are also implemented in the MACSC-ABSC scheme [34]. Specially, the computation overhead required for generating the ciphertext and recovering the plaintext message does not depend on the number of authorities and attributes. Furthermore, this scheme utilize the linear secret-sharing realizable access structure which is more expressive compared with the threshold access structure.

In terms of security proofs, our scheme has been rigorously proved to be secure in the standard model, including ciphertext

indistinguishability under adaptive chosen ciphertext attacks and existential unforgeability under adaptive chosen message attack.

We compare these schemes [14], [24], [25], [33] both theoretically and experimentally with ours. According to the analysis, it can be seen that our scheme has significant advantage in communication overhead and computation overhead. Additionally, our scheme is the first ABSC scheme to implement the ciphertext with constant-size under multi-authority.

C. Paper Organization

In part II, we summarize necessary knowledge of cryptography. After that, we introduce our models of system and security and concrete construction of our MACSC-ABSC scheme in part III. Particularly, we prove the public verifiability. The security analysis and detailed performance comparison are elaborated in part IV and V, respectively. In the end, we summarize this paper in part VI.

II. Preliminaries

A. Bilinear Map

Define p as a prime number and $\mathbb{Z}_p$ as a integers set whose modulus is p. G_1 and G_2 are defined as multiplication cyclic groups of order p. Let $e : G_1 \times G_1 \rightarrow G_2$ be bilinear map as follows:

1) Bilinear: $\forall a, b \in \mathbb{Z}_p, e(g^a, g^b) = e(g, g)^{ab}$;
2) Non-Degenerate: $e(g, g) \neq 1$;
3) Computable: $\forall g_1, g_2 \in G_1$, $e(g_1, g_2)$ is computable.

B. Linear Secret-sharing Scheme

Define $\mathbb{T}$ as monotone access structure (MAS). Let $\mathbb{M}$ be $l \times n$ matrix with linear secret-sharing scheme (LSSS) for $\mathbb{T}$. ρ is row labeling function associating every row i of $\mathbb{M}$ with each attribute $\rho(i)$ in $\mathbb{T}$. Define two algorithms as follows:

- **Distribute**$(\alpha, \rho, \mathbb{M})$: It picks up a secret value α and $b_2, b_3, ..., b_k$ from $\mathbb{Z}_p$ and then sets $\vec{v} = (\alpha, b_2, b_3, ..., b_k) \in \mathbb{Z}_p^k$. It inputs α, ρ and $\mathbb{M}$ and outputs $\{\lambda_{\rho(i)} : \lambda_{\rho(i)} = \vec{M}_i \vec{v}\}_{i \in [l]}$. $\lambda_{\rho(i)}$ corresponds to $\rho(i)$, and $\vec{M}_i \in Z_p^k$ is i^{th} row of $\mathbb{M}$.
- **Reconstruct**$(\mathcal{R}, \rho, \mathbb{M})$: It picks up a set $(\mathcal{R})$ of authorized attributes from $\mathbb{T}$. It inputs $\mathcal{R}$, ρ and $\mathbb{M}$ and outputs $\{\omega_i\}_{i \in I} \subset \mathbb{Z}_p$. Specially, $\sum_{i \in I} \omega_i \vec{M}_i = (1, 0, ..., 0)$, where each ω_i is constant and $I = \{i \in [l] : \rho(i) \in \mathcal{R}\}$. Thus, $\sum_{i \in I} \omega_i \lambda_{\rho(i)} = \alpha$.

C. Complexity Assumption

- Decisional Bilinear Diffie–Hellman exponent assumption $(n-dBDHE)$: Define $\mathcal{A}$ as the algorithm which breaks the $n - dBDHE$ assumption in (G_1, G_2). It inputs $(\vec{y}_{a,\beta}, Z)$, where $\vec{y}_{a,\beta} = (g, g^\beta, g^a, ..., g^{a^n}, g^{a^{n+2}}, ..., g^{a^{2n}})$, $a, \beta \in \mathbb{Z}_p$, g is any one generator of G_1. Then, algorithm $\mathcal{A}$ judges whether $Z = e(g^{a^{n+1}}, g^\beta)$ or an element from G_2. Define the probability of breaking $n - dBDHE$ assumption as follows.

$$Adv = |P[\mathcal{A}(\vec{y}_{a,\beta}, Z) \rightarrow 1 | Z = e(g^{a^{n+1}}, g^\beta)] \\ P[\mathcal{A}(\vec{y}_{a,\beta}, Z) \rightarrow 1 | Z \in G_2]|$$

When $Adv \leq \epsilon$, we define the $n - dBDHE$ assumption in (G_1, G_2) is ϵ-hard [14].

- Computational Diffie–Hellman exponent assumption $(n-cDHE)$: Define $\mathcal{A}$ as the algorithm which breaks the $n - cDHE$ assumption in G_1. It inputs $\vec{y}_a = (g, g^a, ..., g^{a^n}, g^{a^{n+2}}, ..., g^{a^{2n}})$, wherein $a \in \mathbb{Z}_p$, g is any generator from G_1. Then, the algorithm $\mathcal{A}$ outputs $g^{a^{n+1}}$. Define the probability of breaking $n-cDHE$ assumption as follows.

$$Adv = P[\mathcal{A}(\vec{y}_a) \rightarrow g^{a^{n+1}}]$$

When $Adv \leq \epsilon$, we define the $n-cDHE$ assumption in G_1 is ϵ-hard [14].

III. Construction

In this part, we develop Multi-authority ABSC scheme with Constant-size ciphertext described in detail as follows.

A. System Model

Considering the multi-authority cloud storage environment in practical applications, our MACSC-ABSC scheme is divided into System Initialization, Key Generation, Signcryption and Unsigncryption. Specifically, these four phases contain seven algorithms: GlobalSetup, AuthoritySetup, SignKeyGen, DecKeyGen, Signcrypt, Verify and Decrypt. Furthermore, our scheme consists of five different objects as shown in Fig. 1: Cloud Server (CS), Certificate Authority (CA), Attribute Authorities (AA), Data Owner (DO) and Data Users (DU).

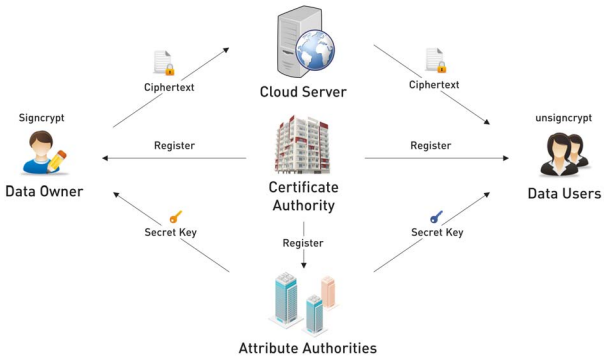

Fig. 1: cloud storage system framwork

- **Cloud Server** (CS): The CS takes charge of storing sensitive data from DO and providing corresponding access control to data users. Particularly, due to supporting public verification, the CS is able to verify integrity and validity of ciphertexts.
- **Certificate Authority** (CA): The CA takes charge of publishing the public parameters PP by executing **GlobalSetup** algorithm. Specially, the CA is in charge of

registrations of DU and AA, and verifies the legitimacy of DU and AA.

- **Attribute Authorities** (AA): The AA take charge of generating the secret keys for DO and DU. Before generation of the secret keys, the AA must initialize itself by outputting their public and secret keys from **AuthoritySetup** algorithm and verify the identity of legitimate users who are registered by CA.
- **Data Owner** (DO): Before the DO signcrypts a message M, he/she can choose an encryption attribute set $\mathcal{R}_e$ deciding legitimate DU and a signature attribute set $\mathcal{R}_s$ for signing a message. Then, the DO can obtain the secret keys from the AA and signcrypt a message M by performing **Signcrypt** algorithm. Furthermore, the data owner can upload the ciphertext to CS.
- **Data User** (DU): DU first requests AA to get the decryption keys, then performs **Unsigncrypt** algorithm in order to get the message corresponding to the ciphertext. Note that our scheme provides public verification so that the DU can employ trusted third party for checking validity of ciphertexts.

MACSC-ABSC consists of seven algorithms wherein $\mathcal{U}_s$ is disjoint universe of signature and encryption attribute set, and $\mathcal{U}_e$ is disjoint universe of signature and decryption attribute set.

1) **GlobalSetup**(1^κ): Executed by CA which inputs the security parameter 1^κ , then outputs public parameter collection PP. In addition, this algorithm contains registering DU and AA.
2) **AuthoritySetup**(PP): This algorithm is executed by AA. Taking as input PP and outputs public keys and secret keys (PK, MK) of each authority.
3) **SignKeyGen**($PP, PK_j, MK_j, (\mathbb{S}_j, \rho_j)$): This algorithm is executed by AA. Taking as input $PP, (PK_j, SK_j)$ of each authority AA_j and a signature access structure $(\mathbb{S}_j, \rho_j)$ over $\mathcal{U}_s$, the algorithm outputs the secret key $SK_{(\mathbb{S}_j,\rho_j)}$ for the DO.
4) **DecKeyGen**($PP, PK_j, MK_j, (\mathbb{D}_j, \phi_j)$): This algorithm is executed by AA. Taking as input $PP, (PK_j, SK_j)$ of each authority AA_j and a decryption access structure $(\mathbb{D}_j, \phi_j)$ over $\mathcal{U}_e$, the algorithm outputs the decryption key $SK_{(\mathbb{D}_j,\phi_j)}$ for the DU.
5) **Signcrypt**($PP, M, PK_j, \{SK_{(\mathbb{S}_j,\rho_j)}\}_{j\in[N]}, \mathcal{R}_s, \mathcal{R}_e$): Executed by the DO and inputted $PP, M, \{SK_{(\mathbb{S}_j,\rho_j)}\}_{j\in[N]}$ of the DO, encryption attribute set $\mathcal{R}_e$ and signature attribute set $\mathcal{R}_s$. It outputs a signcryption ciphertext $CT_{(\mathcal{R}_s,\mathcal{R}_e)}$.
6) **Verify**($PP, PK_j, CT_{(\mathcal{R}_s,\mathcal{R}_e)}$): This algorithm is executed by CS or any trusted third party and inputted PP, PK_j and $CT_{(\mathcal{R}_s,\mathcal{R}_e)}$. It can verify whether $CT_{(\mathcal{R}_s,\mathcal{R}_e)}$ is valid.
7) **Decrypt**($PP, PK_j, CT_{(\mathcal{R}_s,\mathcal{R}_e)}$,$\{SK_{(\mathbb{D}_j,\phi_j)}\}_{j\in[N]}$): Executed by DU and inputted $PP, PK_j, CT_{(\mathcal{R}_s,\mathcal{R}_e)}$ and $\{SK_{(\mathbb{D}_j,\phi_j)}\}_{j\in[N]}$. It outputs plaintext M or $\perp$.

B. Security Requirements

MACSC-ABSC needs to satisfy message confidentiality, ciphertext unforgeability and signcryptor privacy.

- **Message confidentiality**: If the encryption attributes of the DO fail to satisfy decryption access structure of DU, unauthorized users fail to access sensitive data stored in CS.
- **Ciphertext unforgeability**: Unauthorized users fail to generate the ciphertext which satisfies signature access structure.
- **Signcryptor privacy**: If the ciphertext is not associated with signature keys, then DU cannot gain the information relating to access structure and attribute set of DO form the ciphertext.

C. Security Model

Define the two roles in the security model, the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$. The detailed constructs of two types of security games are presented in **Definition 1** and **Definition 2**, respectively. Specially, due to the space limitation of this conference, we present detailed security game procedures in the full version of the manuscript.

- **Definition 1** Message confidentiality: Indistinguishability of ciphertexts under selective attribute set and adaptive chosen ciphertext attack $(IND - sAtt - CCA)$.
  Our MACSC-ABSC scheme is $(\mathcal{T}, q_{sG}, q_{dG}, q_{SC}, q_{US}, \epsilon)$ $-IND - sAtt - CCA$ secure when $\mathcal{A}$ can run for $\mathcal{T}$ (maximum time) and query secret keys, signcryption and unsigncryption up to $(q_{sG} + q_{dG})$ times, q_{SC} times and q_{US} times, respectively.
- **Definition 2** Ciphertext Unforgeability: existential unforgeability under selective attributes set and adaptive chosen message attack $(EUF - sAtt - CMA)$.
  Our MACSC-ABSC scheme is $(\mathcal{T}, q_{sG}, q_{dG}, q_{SC}, q_{US}, \epsilon)$ $-EUF - sAtt - CMA$ secure when $\mathcal{A}$ can run for $\mathcal{T}$ (maximum time) and query secret keys, signcryption and unsigncryption up to $(q_{sG} + q_{dG})$ times, q_{SC} times and q_{US} times, respectively.

D. Concrete Construction

1) **System Initialization**
   In first phase, this system needs to issue public parameters PP and register each authority and users. Subsequently, each of AA initializes itself with public parameters. Note that $\mathcal{M} = \{0,1\}^{l_m}$ denotes the message space, and $\mathcal{U}_s = \{A_s\}$ and $\mathcal{U}_e = \{A_e\}$ are the universes of signature attributes and encryption attributes, respectively.
   - **GlobalSetup**(1^κ): Takes security parameter $\kappa \in \mathbb{N}$ encoded in unary to it then generates multiplicative cyclic groups G_1 and G_2 of order p, g any generator of G_1 and symmetric bilinear map: e: $G_1 \times G_1 \to G_2$. The CA samples four cryptographic collision resistant Hash Functions: $H_1 : G_2 \times G_1 \times \{0,1\}^{l_\tau} \to \{0,1\}^{l_m}$, $H_2 : \{0,1\}^* \to \{0,1\}^l$, $H_3 : G \to \mathbb{Z}_p$, $H_4 : \{0,1\}^* \to \mathbb{Z}_p$ where l and l_m are large enough for collision resistant. The CA picks $K_0, K_1, \delta_1, \delta_2, \mu_0, \mu_1, ..., \mu_l \in G_1$ randomly. Hence, the public parameters PP are generated.

$$PP = (p, G_1, G_2, g, e, K_0, K_1, \delta_1, \delta_2, \mu_0, \mu_1, ..., \mu_l, H_1, H_2, H_3, H_4)$$

Afterwards, the CA selects an independent number $j \in [1, N]$ for each authority AA_j as unique identification and picks a set of unique numbers for users' identification GID.

- **AuthoritySetup**(PP): The algorithm is executed by AA_j for generating public/secret keys pairs. Firstly, the AA_j chooses uniform random $\alpha_j \in \mathbb{Z}_p$ and sets $Y_j = e(g,g)^{\alpha_j}$. Subsequently, the $A_{j,i} \in \widetilde{AA_j}$ picks $T_{j,i} \in G_1$, where $\widetilde{AA_j}$ is the universe of each authority AA_j attributes and $A_{j,i}$ denotes every attribute of AA_j. Therefore, the key pair of each authority AA_j is generated as follows.

$$PK_j = (Y_j, \{T_{j,i}\}_{i \in \widetilde{AA_j}})$$
$$MK_j = (\alpha_j)$$

2) **Key Generation**

After the system initialization, when a user with identification GID requests secret keys from AA_j, AA_j performs it to generate secret keys for this user. The output consists of signature secret keys for DO and decryption secret keys for DU.

- **SignKeyGen**($PP, PK_j, MK_j, (\mathbb{S}_j, \rho_j)$): $(\mathbb{S}_j, \rho_j)$ is a signature LSSS access structure, where $\mathbb{S}_j$ is $l_{s,j} \times n_{s,j}$ matrix on $\mathbb{Z}_p$ and ρ_j is the function corresponding each row of $\mathbb{S}_j$ to signature attribute $A_{s,j,\rho_j(i)}$. The AA_j executes **Distribute**($\mathbb{S}_j, \rho_j, \alpha_j$) to obtain $\{\lambda_{j,\rho_j(i)} = \vec{S}_{j,i} \cdot \vec{v}_{s,j} : i \in [l_{s,j}]\}$, wherein $\vec{S}_{j,i}$ is i^{th} row of $\mathbb{S}_j$, $\vec{v}_{s,j} \in \mathbb{Z}_p^{n_{s,j}}$ such as $\vec{v}_{s,j} \cdot (1, 0, ..., 0) = \alpha_j$.
Subsequently, the AA_j selects $r_{j,i} \in \mathbb{Z}_p$ and computes as follows for each row $i \in [l_{s,j}]$. Note that $T_{s,j,x} \in \mathbb{Z}_p$ is selected for each attribute $A_s \in \mathcal{U}_s$.

$$Q_{s,j,i} = g^{\lambda_{j,\rho_j(i)}} (K_0 T_{j,\rho_j(i)})^{r_{j,i}}$$
$$Q'_{s,j,i} = g^{r_{j,i}}$$
$$Q''_{s,j,i,x} = T_{s,j,x}^{r_{j,i}}, \forall A_{s,j,i} \in \mathcal{U}_s \setminus \{A_{s,\rho_j(i)}\}$$
$$Q''_{s,j,i} = \{Q''_{s,j,i,x}\}$$

Hence, signature key $SK_{(\mathbb{S}_j,\rho_j)}$ is distributed.

$$SK_{(\mathbb{S}_j,\rho_j)} = ((\mathbb{S}_j, \rho_j), Q_{s,j,i}, Q'_{s,j,i}, Q''_{s,j,i} : i \in [l_{j,s}])$$

- **DecKeyGen**($PP, PK_j, MK_j, (\mathbb{D}_j, \phi_j)$): $(\mathbb{D}_j, \phi_j)$ is a decryption LSSS access structure, where $\mathbb{D}_j$ is $l_{e,j} \times n_{e,j}$ matrix on $\mathbb{Z}_p$ and ϕ_j is the function corresponding each row of $\mathbb{D}_j$ to decryption attribute $A_{e,j,\phi_j(i)}$. The AA_j executes **Distribute**($\mathbb{D}_j, \phi_j, \alpha_j$) to obtain a set $\{\lambda_{j,\phi_j(i)} = \vec{D}_{j,i} \cdot \vec{v}_{e,j} : i \in [l_{e,j}]\}$, wherein $\vec{D}_{j,i}$ is i^{th} row of $\mathbb{D}_j$, $\vec{v}_{e,j} \in \mathbb{Z}_p^{n_{e,j}}$ such as $\vec{v}_{e,j} \cdot (1, 0, ..., 0) = \alpha_j$. Subsequently, the AA_j selects $\eta_{j,i} \in \mathbb{Z}_p$ and computes as follows for each row $i \in [l_{e,j}]$. Note that $T_{e,j,y} \in \mathbb{Z}_p$ is selected for each attribute $A_e \in \mathcal{U}_e$.

$$Q_{e,j,i} = g^{\lambda_{j,\phi_j(i)}} (K_1 T_{j,\phi_j(i)})^{\eta_{j,i}}$$
$$Q'_{e,j,i} = g^{\eta_{j,i}}$$
$$Q''_{e,j,i,y} = T_{e,j,y}^{\eta_{j,i}}, \forall A_{e,j,i} \in \mathcal{U}_e \setminus \{A_{e,\phi_j(i)}\}$$
$$Q''_{e,j,i} = \{Q''_{e,j,i,y}\}$$

Hence, decryption key $SK_{(\mathbb{D}_j,\phi_j)}$ is distributed:

$$SK_{(\mathbb{D}_j,\phi_j)} = ((\mathbb{D}_j, \phi_j), Q_{e,j,i}, Q'_{e,j,i}, Q''_{e,j,i} : i \in [l_{e,j}])$$

3) **Signcryption**

During this phase, the data owner DO performs the following algorithm for signcrypting message $M = \{0,1\}^{l_m}$ and uploads ciphertexts to the cloud server. Before signcrypting, the DO formulates authorized signature attribute set $\mathcal{R}_s$ of signature LSSS access structure $(\mathbb{S}_j, \rho_j)$ and selects encryption attribute set $\mathcal{R}_e$, where $\mathcal{R}_e$ indicates what type of data users can decrypt the ciphertext. Afterwards, the DO signcrypts the message $\mathcal{M}$ as follows.

- **Signcrypt**($PP, M, PK_j, \{SK_{(\mathbb{S}_j,\rho_j)}\}_{j\in[N]}, \mathcal{R}_s, \mathcal{R}_e$): The data owner DO executes **Reconstruct**($\mathbb{S}_j, \rho_j, \mathcal{R}_s$) to obtain a constants set $\{\omega_{j,i} \in \mathbb{Z}_p : i \in \mathcal{I}_{s,j}\}$, where $\mathcal{I}_{s,j} = \{i \in [l_{s,j}] : A_{s,j,\rho_j(i)} \in \mathcal{R}_s\}$, $\sum_{i\in\mathcal{I}_{s,j}} \omega_{j,i} \vec{S}_{j,i} = \vec{1}$.
Subsequently, the DO performs as follows:
  - picks $\beta, \varsigma, \xi \in \mathbb{Z}_p$ and calculates

$$\begin{cases} C_1 = g^{\beta} \\ C_2 = (K_1 \prod\limits_{A_{e,y}\in\mathcal{R}_e, j\in[N]} T_{e,j,y})^{\beta} \\ \sigma_1 = g^{\beta\varsigma} \\ \sigma_2 = g^{\xi} \prod\limits_{i\in\mathcal{I}_{s,j}, j\in[N]} (Q'_{s,j,i})^{\omega_{j,i}} \end{cases}$$

  - encrypts the message M such as

$$\Lambda = \prod_{j\in[N]} Y_j^{\beta} = \prod_{j\in[N]} e(g,g)^{\beta\alpha_j}$$
$$\sigma_3 = \pi$$
$$C_3 = H_1(\Lambda, \sigma_1, \sigma_3) \oplus M$$

    where $\pi \in \{0,1\}^{l_\pi}$
  - computes

$$\mu = H_3(C_1), C_4 = (\delta_1^{\mu}\delta_2)^{\beta}$$
$$H_2(\sigma_2||\sigma_3||\mathcal{R}_s||\mathcal{R}_e) = (m_1, m_2, ..., m_l)$$
$$H_4(\sigma_1||C_2||C_3||C_4||\mathcal{R}_s||\mathcal{R}_e) = \theta$$

    where $(m_1, m_2, ..., m_l) \in \{0,1\}^l$
  - finally calculates

$$\sigma_4 = \prod_{i\in\mathcal{I}_{s,j}, j\in[N]} (Q_{s,j,i} \prod_{A_{s,x}\in\mathcal{R}_s, x\neq\rho_j(i)} Q''_{s,j,i,x})^{\omega_{j,i}}$$
$$\cdot (K_0 \prod_{A_{s,x}\in\mathcal{R}_s, j\in[N]} T_{s,j,x})^{\xi} \cdot (\mu_0 \prod_{k\in[l]} \mu_k^{m_k})^{\beta} \cdot C_4^{\theta\varsigma}$$

Ultimately, the DO uploads the signcryption ciphertext $CT_{(\mathcal{R}_s,\mathcal{R}_e)}$ of M as follows. ctt is current time.

$$CT_{(\mathcal{R}_s,\mathcal{R}_e)} = (\mathcal{R}_s, \mathcal{R}_e, C_1, C_2, C_3, C_4, \sigma_1, \sigma_2, \sigma_3, \sigma_4, ctt)$$

4) **Unsigncryption**

In this phase, if DU possessing authorized attribute set requests ciphertexts data from CS, CS sends back ciphertexts $CT_{(\mathcal{R}_s,\mathcal{R}_e)}$ to this DU. Before verifing the signature and recovering the message, the DU needs to check $\widetilde{ctt}$. Define predefined time threshold for unsigncryption as Δ_{ctt}. If $|\widetilde{ctt} - ctt| < \Delta_{ctt}$, the unsigncryption algorithm runs by the following steps:

- **Verify**($PP, PK_j, CT_{(\mathcal{R}_s,\mathcal{R}_e)}$): The data user DU checks whether $\mathcal{R}_e$ satisfies $(\mathbb{D}_j, \phi_j)$ or not. Afterwards the DU computes as follows.

$$\begin{aligned}\mu &= H_3(C_1)\\ (m_1, m_2, ..., m_l) &= H_2(\sigma_2||\sigma_3||\mathcal{R}_s||\mathcal{R}_e)\\ \theta &= H_4(\sigma_1||C_2||C_3||C_4||\mathcal{R}_s||\mathcal{R}_e)\end{aligned}$$

DU needs to check the validity of $CT_{(\mathcal{R}_s,\mathcal{R}_e)}$ by verifying the following equation (1).

$$\begin{aligned}e(\sigma_4, g) \stackrel{?}{=} (\prod_{j\in[N]} Y_j) \cdot e(K_0 \prod_{A_{s,x}\in\mathcal{R}_s, j\in[N]} T_{s,j,x}, \sigma_2)\\ \cdot e(\mu_0 \prod_{k\in[l]} \mu_k^{m_k}, C_1) \cdot e((\delta_1^{\mu}\delta_2)^{\theta}, \sigma_1)\end{aligned} \tag{1}$$

If it is not valid, the algorithm returns $\perp$. Otherwise, the DU continues the following decryption procedure for recovering the message M.

- **Decrypt**($PP, PK_j, CT_{(\mathcal{R}_s,\mathcal{R}_e)}$,$\{SK_{(\mathbb{D}_j,\phi_j)}\}_{j\in[N]}$): In order to obtain a constants set $\{\varpi_{j,i} \in \mathbb{Z}_p : i \in \mathcal{I}_{e,j}\}$, the DU executes **Reconstruct**($\mathbb{D}_j, \phi_j, \mathcal{R}_e$), where $\mathcal{I}_{e,j} = \{i \in [l_e] : A_{e,j,\phi_j(i)} \in \mathcal{R}_e\}$ such that $\sum_{i\in\mathcal{I}_{e,j}} \varpi_{j,i}\vec{D}_{j,i} = \vec{1}$. Subsequently, the DU calculates E_1, E_2 as follows in order to recover $\Lambda = \prod_{j\in[N]} Y_j^{\beta}$ with the following equation.

$$\begin{aligned}E_1 &= \prod_{i\in\mathcal{I}_{e,j}, j\in[N]} (Q_{e,j,i} \cdot \prod_{A_{e,y}\in\mathcal{R}_e, y\neq\phi_j(i)} Q''_{e,j,i,y})^{\varpi_{j,i}}\\ E_2 &= \prod_{i\in\mathcal{I}_{e,j}, j\in[N]} (Q'_{e,j,i})^{\varpi_{j,i}}\\ \Lambda &= \frac{e(C_1, E_1)}{e(C_2, E_2)}\end{aligned}$$

Finally, it outputs the complete message M as follows.

$$M = C_3 \oplus H_1(\Lambda, \sigma_1, \sigma_3)$$

E. Public Verifiability

Note that critical step in the **Verify** algorithm, i.e., equation (1) is constructed according to public parameters PP, public keys PK_j of each authority and ciphertexts CT components except decryption secret keys SK of DU. In consequence, any intermediate party, e.g., DU, CS, can check whether ciphertexts satisfies the integrity and validity of the signcryptor. Hence, our scheme is public verifiability.

IV. Security Analysis

In this part, we give overviews of Message Confidentiality and Ciphertext Unforgeability, as well as a detailed proof of signcryptor privacy protection. Given the space limitation of this conference, detailed formal security proofs of two security requirements are presented in the full version of the manuscript.

A. Message Confidentiality

According to **Definition 1**, our scheme satisfies indistinguishability of ciphertexts under selective attributes set and adaptive chosen ciphertexts attack($IND - sAtt - CCA$).

Theorem 1 Suppose that encryption attributes universe $\mathcal{U}_e$ consists of n attributes and the four Hash Functions H_1, H_2, H_3, H_4 are collision resistant. Our scheme is $(\mathcal{T}, q_{sG}, q_{dG}, q_{SC}, q_{US}, \epsilon) - IND - sAtt - CCA$ secure assuming n-dBDHE assumption in (G_1, G_2) is $(\mathcal{T}', \epsilon') - hard$, where

$$\begin{aligned}\mathcal{T} =&\mathcal{T}' - \mathcal{O}(|\mathcal{U}_s|^2 \cdot (q_{sG} + q_{SC}) + n^2 \cdot (q_{dG} + q_{US})) \cdot \mathcal{T}_e\\ &- \mathcal{O}(q_{US}) \cdot T_p\\ \epsilon =&\epsilon' + (\frac{q_{US}}{p})\end{aligned}$$

Especially, $\mathcal{T}_e$ and $\mathcal{T}_p$ are the time of one exponentiation and one pairing operation, respectively.

B. Ciphertext Unforgeability

According to **Definition 2**, our scheme satisfies the ciphertext unforgeability under selective attributes set and adaptive chosen message attack($EUF - sAtt - CMA$).

Theorem 2 Suppose that signature attributes universe $\mathcal{U}_s$ consists of n attributes and the four Hash Functions H_1, H_2, H_3, H_4 are collision resistant. Our scheme is $(\mathcal{T}, q_{sG}, q_{dG}, q_{SC}, q_{US}, \epsilon) - EUF - sAtt - CMA$ secure assuming the n-cDHE assumption in G_1 is $(\mathcal{T}', \epsilon') - hard$,

$$\begin{aligned}\mathcal{T} =&\mathcal{T}' - \mathcal{O}(n^2 \cdot (q_{sG} + q_{SC}) + |\mathcal{U}_e|^2 \cdot (q_{dG} + q_{US})) \cdot \mathcal{T}_e\\ &- \mathcal{O}(q_{SC} + q_{US}) \cdot T_p\\ \epsilon =&\epsilon' \kappa(l+1)\end{aligned}$$

where l is the size of output of Hash Function H_2, and κ is security parameter. Especially, $\mathcal{T}_e$ and $\mathcal{T}_p$ are the time of one exponentiation and one pairing operation, respectively.

C. Signcryptor Privacy

By analyzing the compositions of the ciphertext CT as follows,

$$CT_{(\mathcal{R}_s,\mathcal{R}_e)} = (\mathcal{R}_s, \mathcal{R}_e, C_1, C_2, C_3, C_4, \sigma_1, \sigma_2, \sigma_3, \sigma_4, ctt)$$

where $\mathcal{R}_s, \mathcal{R}_e$ are a signature attribute set and an encryption attribute set, respectively, and ctt is current time.

$$\begin{cases}C_1 =g^{\beta}\\ C_2 =(K_1 \prod_{A_{e,y}\in\mathcal{R}_e, j\in[N]} T_{e,j,y})^{\beta}\\ C_3 =H_1(\Lambda, \sigma_1, \sigma_3) \oplus M\\ C_4 =(\delta_1^{\mu}\delta_2)^{\beta}\\ \sigma_1 =g^{\beta\varsigma}\\ \sigma_2 =g^{\xi'}\\ \sigma_3 =\pi\\ \sigma_4 =g^{\alpha}(K_0 \prod_{A_{s,x}\in\mathcal{R}_s, j\in[N]} T_{s,j,x})^{\xi'}\\ \quad \cdot (\mu_0 \prod_{k\in[l]} \mu_k^{m_k})^{\beta} \cdot (C_4)^{\theta\varsigma}\end{cases}$$

Furthermore, β, ξ', ς are random exponents and $\mu = H_3(C_1)$, $(m_1, m_2, ..., m_l) = H_2(\sigma_2||\sigma_3||\mathcal{R}_s||\mathcal{R}_e)$, $\pi \in \{0,1\}^{l_\pi}$, $\theta = H_4(\sigma_1||C_2||C_3||C_4||\mathcal{R}_s||\mathcal{R}_e)$. Specially, $\alpha = \sum_j \alpha_j$, $\Lambda = \prod_{j\in[N]} Y_j^{\beta} = \prod_{j\in[N]} e(g,g)^{\beta\alpha_j}$ are from the public/secret key pair of each authority. Obviously, the compositions of CT do not depend on signature keys of DO. Therefore, our scheme provides signcrytor privacy.

V. Scheme Analysis

In this part, we analyze security and functionality as well as performance of MACSC-ABSC scheme with these ABSC schemes [14], [24], [25], [33] as follows.

A. Functionality and Security

In the part, we compare and analyze functionality and security as shown in **TABLE I**. Note that ✓ indicates the capability to achieve the corresponding index, whereas × represents the opposite. According to this comparison, [24], the security of the schemes [24], [25] are not as strong as [14], [33] and ours. And they do not support public verifiability and signcryptor privacy. Compared with [14], we implement multi-authority setting with the other functionalities and security consistent. Specially, compared with [33], our scheme has many extension functions that can be implemented.

TABLE I: Comparison of security and functionality

Scheme	AS	MC	CU	Model	MA	PV	SP	OS
[14]	LSSS	CCA	EUF	Standard	×	✓	✓	×
[24]	MSP	CPA	EUF	Rom	×	×	×	✓
[25]	MSP	CPA	EUF	Rom	×	×	×	✓
[33]	MBF	CCA2	EUF	Standard	✓	✓	✓	✓
ours	LSSS	CCA	EUF	Standard	✓	✓	✓	×

AS = Access Structure, MC = message confidentiality, CU = ciphertext unforgeability, MA = Multi-Authority, PV = public verifiability, SP = Signcryptor privacy, OS = Outsource, CCA = indistinguishability of ciphertexts under selective chosen ciphertext attack, CPA = indistinguishability of ciphertexts under selective chosen plaintext attack, CCA2 = indistinguishability of ciphertexts under adaptive chosen ciphertext attack, ROM = random oracle model.

B. Performance

We analyze and compare the ABSC schemes [14], [24], [25], [33] performance about communication overhead from the factors of the size of signing key, decryption key and ciphertexts, and computaion overhead about the phases of signcryption and unsigncryption in **TABLE II** and **TABLE III**, respectively. Results in tables indicate that our sheme has high efficiency in communication transmission due to the constant ciphertext size.

As for comparison of computation efficiency, we implement simulation experiments of these schemes by the Charm framework [35], a framework for rapidly prototyping cryptosystems. We perform experiments on a computer with Intel i7-8550U processor and 8GB RAM, which is configured with python 3.7.2, Charm-0.50 framework and PBC-0.5.14 library for the backend. Fig. 2 and Fig. 3 show the trend diagram of the computation overhead in the phase of signcrypt and unsigncrypt as the number of attributes increases, respectively. It can be seen from the analysis of the above figures that our scheme is more efficient than [33] under multi-authority scenario.

TABLE II: Comparison of communication overhead

Scheme	SKS	DKS	Ciphertext
[14]	$(2l_s+2)\|G_1\|$	$(2l_d+2)\|G_1\|$	$6\|G_1\|$
[24]	$(2+l_s)\|G_1\|$	$(2+l_d)\|G_1\|$	$2(U_e+U_s+2)\|G_1\|$
[25]	$(2+l_s)\|G_1\|$	$(2+l_d)\|G_1\|$	$(2U_e+U_s)\|G_1\|$
[33]	$(9+l_s)\|G_1\|$	$(9+l_d)\|G_1\|$	$\|G_2\|+(4+3U_e+U_s)\|G_1\|$
ours	$(2l_s+2)\|G_1\|$	$(2l_d+2)\|G_1\|$	$6\|G_1\|$

SKS = size of the signature key, DKS = size of decryption key, $|G_1|$ and $|G_2|$ represent the length of G_1 and G_2, respectively. $l_s(l_d), U_s(U_e)$ denote the count of attributes associated with signature or decryption key, the count of signature or encryption attributes, respectively.

TABLE III: Comparison of computation overhead

Scheme	Signcryption	Unsigncryption
[14]	$(3U_e+3U_s+4)E_{G_1}$	$2U_eE_{G_1}+6P$
[24]	$(4+U_e+2U_s)E_{G_1}$	$(2+U_s)E_{G_1}+P$
[25]	$(U_e+U_s+5)E_{G_1}$	$(U_e^2+U_e+U_s)E_{G_1}+9P$
[33]	$E_{G_2}+(5U_e+U_s+4)E_{G_1}$	$E_{G_2}+(3U_e+U_s+3)E_{G_1}+P$
ours	$(3U_e+3U_s+4)E_{G_1}$	$2U_eE_{G_1}+6P$

P denotes one pairing computation, E_{G_1} (E_{G_2}) represents one modular exponentiation computation in G_1 (G_2).

VI. Conclusion

In this paper, we are committed to extending the attribute-based signcryption scheme [14] into a multi-authority scenario with constant-size ciphertext, public verifiability and signcryptor privacy. Moreover, the security of our scheme is proven in the standard model. In the future, we will implement more extensibility functions based on this scheme, such as outsourcing part of decryption and encryption steps to the cloud server for less computation overhead.

VII. Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. 62076054, No. 62072074, No. 62027827, No.61902054), Natural Science Foundation of Guangdong Province (Grant No. 2018A030313354), the Frontier Science and Technology Innovation Projects of National Key R&D Program (No.2019QY1405), the Sichuan Science-Technology Support Plan Program (No. 2020YFSY0010, No.2019YJ0636).

References

[1] P. Mell, T. Grance *et al.*, "The nist definition of cloud computing," 2011.
[2] H. Xiong, Y. Hou, X. Huang, and Y. Zhao, "Secure message classification services through identity-based signcryption with equality test towards the internet of vehicles," *Vehicular Communications*, p. 100264, 2020.
[3] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *International Conference on Financial Cryptography and Data Security*. Springer, 2010, pp. 136–149.
[4] H. Xiong, K.-K. R. Choo, and A. V. Vasilakos, "Revocable identity-based access control for big data with verifiable outsourced computing," *IEEE Transactions on Big Data*, 2017.
[5] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.

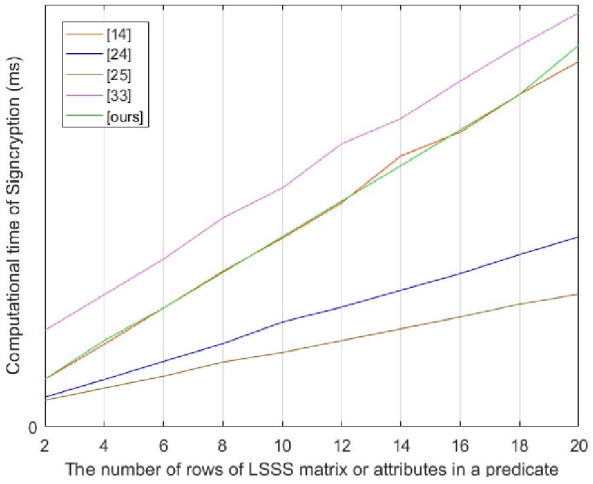

Fig. 2: Computation of Signcryption

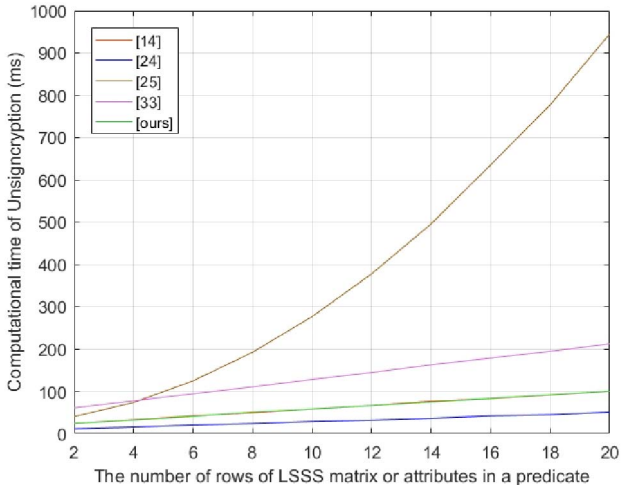

Fig. 3: Computation of Unsigncryption

[6] 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. 49 232–49 239, 2019.

[7] 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," *Journal of the Chinese Institute of Engineers*, vol. 42, no. 1, pp. 20–28, 2019.

[8] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, "A secure authentication protocol for internet of vehicles," *Ieee Access*, vol. 7, pp. 12 047–12 057, 2019.

[9] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2005, pp. 457–473.

[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*. Acm, 2006, pp. 89–98.

[11] A. Lewko and B. Waters, "Unbounded hibe and attribute-based encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2011, pp. 547–567.

[12] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE symposium on security and privacy (SP'07)*. IEEE, 2007, pp. 321–334.

[13] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2010, pp. 62–91.

[14] Y. S. Rao and R. Dutta, "Expressive attribute based signcryption with constant-size ciphertext," in *International Conference on Cryptology in Africa*. Springer, 2014, pp. 398–419.

[15] M. Gagné, S. Narayan, and R. Safavi-Naini, "Threshold attribute-based signcryption," in *International Conference on Security and Cryptography for Networks*. Springer, 2010, pp. 154–171.

[16] H. Xiong, Y. Bao, X. Nie, and Y. I. Asoor, "Server-aided attribute-based signature supporting expressive access structures for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1013–1023, 2019.

[17] C. Chen, J. Chen, H. W. Lim, Z. Zhang, and D. Feng, "Combined public-key schemes: the case of abe and abs," in *International Conference on Provable Security*. Springer, 2012, pp. 53–69.

[18] J. Liu, X. Huang, and J. K. Liu, "Secure sharing of personal health records in cloud computing: ciphertext-policy attribute-based signcryption," *Future Generation Computer Systems*, vol. 52, pp. 67–76, 2015.

[19] Y. S. Rao, "A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing," *Future Generation Computer Systems*, vol. 67, pp. 133–151, 2017.

[20] G. Yu and Z. Cao, "Attribute-based signcryption with hybrid access policy," *Peer-to-Peer Networking and Applications*, vol. 10, no. 1, pp. 253–261, 2017.

[21] H. Xiong, Y. Zhao, Y. Hou, X. Huang, C. Jin, L. Wang, and S. Kumari, "Heterogeneous signcryption with equality test for iiot environment," *IEEE Internet of Things Journal*, 2020.

[22] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in *Cryptographers? track at the RSA conference*. Springer, 2011, pp. 376–392.

[23] Y. Han, W. Lu, and X. Yang, "Attribute-based signcryption scheme with non-monotonic access structure," in *2013 5th International Conference on Intelligent Networking and Collaborative Systems*. IEEE, 2013, pp. 796–802.

[24] F. Deng, Y. Wang, L. Peng, H. Xiong, J. Geng, and Z. Qin, "Ciphertext-policy attribute-based signcryption with verifiable outsourced designcryption for sharing personal health records," *IEEE Access*, vol. 6, pp. 39 473–39 486, 2018.

[25] F. Deng, Y. Wang, L. Peng, M. Lai, and J. Geng, "Revocable cloud-assisted attribute-based signcryption in personal health system," *IEEE Access*, vol. 7, pp. 120 950–120 960, 2019.

[26] M. Chase, "Multi-authority attribute based encryption," in *Theory of cryptography conference*. Springer, 2007, pp. 515–534.

[27] M. Chase and S. S. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 121–130.

[28] J. Han, W. Susilo, Y. Mu, J. Zhou, and M. H. Au, "Ppdcp-abe: Privacy-preserving decentralized ciphertext-policy attribute-based encryption," in *European Symposium on Research in Computer Security*. Springer, 2014, pp. 73–90.

[29] J. Han, W. Susilo, Y. Mu, J. Zhou, and M. H. A. Au, "Improving privacy and security in decentralized ciphertext-policy attribute-based encryption," *IEEE transactions on information forensics and security*, vol. 10, no. 3, pp. 665–678, 2014.

[30] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2011, pp. 568–588.

[31] K. Yang, X. Jia, K. Ren, B. Zhang, and R. Xie, "Dac-macs: Effective data access control for multiauthority cloud storage systems," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1790–1801, 2013.

[32] X. Meng and X. Meng, "A novel attribute-based signcryption scheme in cloud computing environments," in *2016 IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2016, pp. 1976–1979.

[33] Q. Xu, C. Tan, Z. Fan, W. Zhu, Y. Xiao, and F. Cheng, "Secure multi-authority data access control scheme in cloud storage system based on attribute-based signcryption," *IEEE Access*, vol. 6, pp. 34 051–34 074, 2018.

[34] 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 Journal*, vol. 19, no. 18, pp. 8317–8331, 2019.

[35] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.