This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3073740, IEEE Transactions on Services Computing

Secure Outsourced Attribute-based Sharing Framework for Lightweight Devices in Smart Health Systems

Leyou Zhang, Wenting You, and Yi Mu, *Senior Member, IEEE*

***Abstract*—The rapid evolution of the Internet of Things has led to the development of smart health. As a form of medical care that uses advanced Internet technology to realize better diagnosis and treatment of patients, smart health transitions medical services move toward real intelligence and greatly helps users. And in smart health, the secure sharing of personal health records (PHRs) is one of the main concerns of patients and medical personnel. Many attribute-based sharing models have been proposed to secure the sharing of PHRs, but there are still two problems to resolve. One is the potential disclosure of the patient data. The attribute-based model achieves flexible access control, but the access policies contain sensitive information of patients. The disclosure of the policy will lead to the leakage of data of the users. The other is the high computational and storage overhead, particularly in smart health systems with limited computing power. In this paper, we present a Smart Health-Lightweight Fine-Grained Sharing (SH-LFGS) framework based on attribute-based encryption (ABE). It achieves a fully hidden access policy by adopting Viéte's formula. SH-LFGS introduces an online/offline mechanism in the PHR encryption phase and the outsourced verifiable decryption mechanism. Because the decrypting test requires only one bilinear pair operation, the SH-LFGS can achieve the task of lightweight computation. Analysis of the performance and security of the proposed model confirm its efficiency and security.**

***Index Terms*—Attribute-based encryption, Personal health records (PHR), Lightweight devices, Fully hidden policy, Decryption test**

I. Introduction

WITH the rapid evolution of cloud computing and Internet of Things technology, multi-user information-sharing mechanisms have attracted much attention. More and more patients are taking part in smart health by uploading their personal health records (PHRs) to the cloud for sharing, which greatly improves the quality of medical care and reduces medical costs. However, cloud service providers are not completely reliable, and patients are not sure whether the data stored in the cloud is safe. Moreover, attacks on cloud servers threaten the security of the data. Therefore, the sharing of PHRs in the cloud can lead to leaks of sensitive data.

To avoid PHR data breaches and give patients more control over shared PHR data, many attribute-based encryption (ABE)[2-8] schemes have been proposed for PHR systems. In these scenarios, the patient specifies an access strategy and encrypts the PHR. Only attributes that satisfy the policy can access the encrypted PHR. However, many existing schemes consider only the confidentiality of the data, and not the privacy of the recipient. ABE schemes [1-7] that support access policy hiding seem to solve this problem. However, most of these schemes only implement partial concealment of access policy; that is, each attribute is divided into attribute name and attribute value. Because attribute value usually contains more sensitive information, it is hidden. For example, for the access policy in Fig. 1, "Doctor Number", "Hospital", and "Department" are the attribute names, while "302-1156", "Central Hospital", and "Cardiologist" are the attribute values, which obviously contain more sensitive information. If the attribute value is not hidden, a malicious user can infer from the information contained in the access policy that a patient has had a heart attack and is being treated in a city hospital. As a result, schemes that only partially hide access policies can be considered insecure. Should a malicious user master the user attribute value by illegal means, and then try to decrypt it by an offline dictionary attack, the exposure of the attribute name will greatly increase the probability of the attack succeeding. In addition, because the cost of encryption, decryption, and storage of existing schemes increases linearly with the complexity of access policies, the use of lightweight devices is discouraged. Therefore, it is necessary to develop a scheme for lightweight devices that can completely hide access policy.

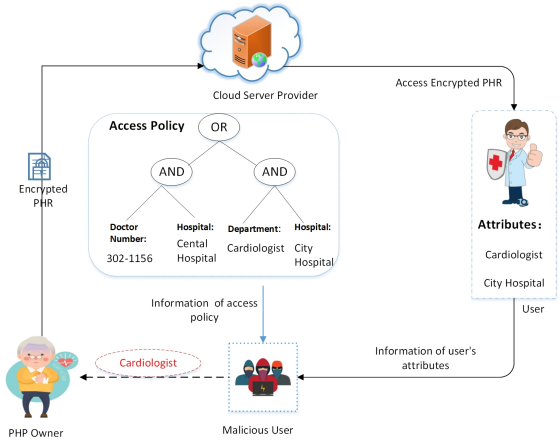

Fig. 1. Example of patient data leakage.

L. Zhang and W. You are with the School of Mathematics and Statistics, Xidian University, Xi'an 710071, China e-mail: lyzhang@mail.xidian.edu.cn; ywtstudy@126.com.

Y. Mu is with the Institute of Data Science, City University of Macau, Macau 999078, China e-mail: ymu.ieee@gmail.com.

A. Related Work

1) Attribute-Based Encryption: Fuzzy IBE [1] was first put forward by Sahai and Waters in 2005. In [2], Goyal et al. proposed a KP-ABE scheme based on fuzzy IBE, which introduced the access tree structure to the access strategy. Ostrovsky [3] and Lai [4] et al. conducted further research on KP-ABE. Bethencourt et al. [5] proposed CP-ABE, which placed the policy in the ciphertext. Cheung et al. [6] proposed the first CP-ABE scheme based on standard assumptions to prove security under the standard model. Waters [7] proposed a new method to demonstrate the security of encryption system: Dual System Encryption. Subsequently, Waters [8] used a linear secret sharing scheme (LSSS) to realize any monotonous access structure, while the scheme supported universal access structure and demonstrated its security based on decisional Parallel Bilinear Diffie-Hellman Exponent (PBDHE). The above ABE schemes [9-12] consider only the confidentiality of the data while ignoring the importance of user privacy. Nishide et al. [13] proposed an ABE scheme that achieved partial hidden access policy by using wildcards. However, this scheme's drawback was that its computational overhead was too high, and it was not applicable to an actual electronic medical system. To solve the problem, a scheme that implemented hidden policy [14] was proposed, in which the calculation costs of secret key, ciphertext, and decryption were constant. This scheme resulted in greatly improved efficiency, but it was based on composite order groups. Zhang et al. [15] proposed an anonymous ABE system based on hierarchical ABE, which implemented short public key and constant secret key, supported fast decryption. Zhang et al. [16,17] and Chaudhari et al. [32] also managed to hide access policy. To reduce the decryption computation overhead in ABE, the method given in [25] was proposed as a solution to achieve outsourced decryption. However, in this scenario, the user cannot verify that the cloud server has been properly decrypted. Lai et al. [26] proposed a verifiable outsourcing mechanism to validate the decryption results. However, all the above schemes support only the AND gate. To achieve a more flexible access policy, Lai et al. [18] applied LSSS to a CP-ABE scheme hidden by access policy, where each attribute contains attribute name and its corresponding multiple attribute values. In a similar fashion, Cui et al. [19,20] and Wang et al. [33] also implemented the semi-hiding of access policy. All the above schemes have high computational overhead. To reduce the calculation cost, Zhang et al. [21] proposed a CP-ABE scheme that supports the hidden access strategy of a decryption test, in which the public parameters are small and constant, and the decryption test in this scheme requires only constant pair calculation. The schemes presented in [22-24] and other schemes that support partially-hidden access policy use online/offline encryption, which reduces the online computing cost of the data owner by carrying out a high percentage of the calculations offline. However, because the LSSS access structure could not completely hide the access structure, Phuong et al. [27] achieved the complete hiding of the access structure by inner product encryption (IPE) [28] [29] and wildcard. This scheme converted the access policy into a vector through Viète's formula [30]. Jin et al. [31] improved the scheme presented in [27] by basing the scheme on composite order groups and demonstrated the complete security of the scheme. Unfortunately, the computational cost of the scheme mentioned above for inner product encryption is very high.

2) Viète's Formulas: The main advantage of Viète's formula is that the user-defined policy and the attribute set can be transformed into vectors so that the access policy can be completely hidden. In 2010, Sedghi et al.[30] used Viète's formula to propose a wildcard searchable encryption scheme with support vector hiding. The scheme was demonstrated to be selectively safe. In 2014, Phuong et al.[38] proposed ciphertext strategy schemes based on a compound order group and a prime number group, with the schemes of constant ciphertext size. The two schemes were demonstrated to be secure in the selective security model. In 2015, Phuong et al. [39] proposed two attribute-based broadcast encryption schemes supporting AND-gate. The first scheme was based on key policy, which supports short ciphertext and constant decryption key. The second scheme was based on ciphertext policy, which supports constant ciphertext and short decryption key. Subsequently, Phuong et al. [27] proposed two CP-ABE schemes, which realized the function of constant ciphertext size and complete hiding of access policy. In 2017, Wang et al. [40] proposed a scheme with equality test. A semi-trusted entity was entrusted to perform equivalence tests on ciphertext encrypted by different access policies, while the entrusted entity could not know any message about the plaintext. The scheme was also a constant ciphertext size. In 2018, Hu et al. [41] put forth a new broadcast encryption scheme based on anonymous attribute. In this scheme, each user had a group identity and attribute set, and the data was encrypted by the policy specified by the user and a set of group identities, so that only the users who belonged to the specified group and the attribute set met the access policy and could be successfully decrypted. At the same time, the scheme achieved greatly improved efficiency by taking into account online/offline encryption and outsourcing decryption. In 2020, Hsu et al.[42] proposed an attribute-based sharing method of edge computing to protect private data. The scheme guaranteed anonymous IoT user/device authentication and supported switching between areas covered by two adjacent edge devices.

B. Our Motivation and Contribution

Based on the analyses of the above, there are the followings to be solved in smart health system. *1)* How to control the uploading data by the data owner. The data owner can set the access policies before uploading the data. Only the users who match the policies can access the sharing data. *2)* Privacy-preserving. The anonymous features are needed for the data. It will protect the privacy of the users. *3)* Test-verification. The proposal must provide a test algorithm to keep the confidentiality and integrity of the sensitive data, which can avoid the data tempering. In addition, if the construction achieves anonymity, the scheme must provide a fast test algorithm to help the receivers make sure whether the receives is valid or not. *4)*

Lightweight computation. For the limited computation power of the smart health system, the construction must achieve lightweight computation.

Motivated by the above problems, we propose a secure sharing scheme based on ABE for the lightweight devices in a Smart Health - Lightweight Fine-Grained Sharing (SH-LFGS) system. Our contributions can be said to consist of four aspects.

1) Fully Access Policy Hidden: Most previous schemes can only achieve semi-hiding of access policy. In our scheme, the access policy can be completely hidden by applying Viète's formula.

2) Online/Offline Encryption: The data owner will do a good deal of computation during the offline phase without knowing the exact message and access policy. This allows the data owner to quickly assemble the ciphertext in the online phase.

3) Efficient Decryption Test: To improve decryption efficiency, this paper designs a decryption test that requires only one bilinear pair of operations before full decryption.

4) Verifiable Outsourced Decryption: The decryption of ciphertext is outsourced to a third-party cloud service provider to reduce the amount of decryption for users. At the same time, the data verification is added in the outsourcing decryption, to ensure that only the correct calculation results can pass the verification.

C. Paper Organization

The remainder of this paper is organized as follows. In Section II we explain the preliminaries that contain bilinear pairings, complexity assumption, access policy, and Viète's formulas. The definition of the SH-LFGS scheme and its security model are given in Section III. Section IV describes the design of the SH-LFGS model. The proof of the scheme is given in Section V. The performance analysis is provided in Section VI. In Section VII, we present our conclusions.

II. PRELIMINARIES

A. Bilinear Pairings

Let G and G_T be two cyclic multiplicative groups with the prime order q, and $e : G \times G \to G_T$ is a bilinear map with the following properties:

1) Bilinear: $\forall g, v \in G$ and $\forall \alpha, \beta \in Z_q$ has $e(g^\alpha, v^\beta) = e(g, v)^{\alpha\beta}$.

2) Non-degenerate: There exists $g, v \in G$ such that $e(g, v) \neq 1$.

3) Computable: $e(g, v)$ can be calculated by the efficient algorithms, where $g, v \in G$.

B. Assumption of Complexity

1) Decisional Bilinear Diffie-Hellman (DBDH) Assumption: Given a tuple $(g, A = g^\alpha, B = g^\beta, C = g^\gamma, T) \in G \times G_T$ to determine whether $T = e(g, g)^{\alpha\beta\gamma}$ or $T = e(g, g)^\delta$ where $\alpha, \beta, \gamma, \delta$ are chosen randomly from Z_q. If ϵ is negligible for any PPT algorithm $\mathscr{C}$, then the assumption holds in G.

$$Adv_{\mathscr{A}} = | Pr[\mathscr{C}(A, B, C, R | R = e(g, g)^{\alpha\beta\gamma}) = 0] - Pr[\mathscr{C}(A, B, C, R | R = e(g, g)^\delta) = 0] | < \epsilon$$

2) Decisional Linear (DLIN) Assumption: Given a tuple $(g, A = g^\alpha, B = g^\beta, C = g^{\alpha\gamma}, D = g^\eta, T) \in G \times G_T$ to determine whether $T = e(g, g)^{\beta(\gamma+\eta)}$ or $T = e(g, g)^t$ where α, β, γ, η, t are chosen randomly from Z_q. If ϵ is negligible for any PPT algorithm $\mathscr{C}$, then the assumption holds in G.

$$Adv_{\mathscr{A}} = |Pr[\mathscr{C}(A, B, C, D, T | T = e(g, g)^{\beta(\gamma+\eta)}) = 0] - Pr[\mathscr{C}(A, B, C, D, T | T = e(g, g)^t) = 0]| < \epsilon$$

C. Access Policy

Let $U = (Att_1, Att_2, ..., Att_n)$ represent the attribute universe in the system, where each Att_i has a unique value A_i. When a user takes part in the system, the user will be bound to his/her attributes $L = (L_1, L_2, ..., L_n)$, where each L_i contains two values: "+" and "-". The access policy can be defined as $W = (W_1, W_2, ..., W_n)$, where each W_i contains three values: "+", "-", "$*$". The wildcards "$*$" denotes *don't care*. We say that $L \models W$ denotes that user's attribute set satisfies the access policy W. $L \not\models W$ denotes that the user's attribute set does not satisfy the policy.

D. Viète's Formulas

Given two vectors $\vec{\sigma} = (\sigma_1, \sigma_2, ..., \sigma_P)$ and $\vec{\tau} = (\tau_1, \tau_2, ..., \tau_P)$, where $\vec{\sigma}$ involves alphabets and wildcards, $\vec{\tau}$ contains nothing but alphabets. Besides that definite the $Pos = (r_1, r_2, ..., r_n) \subset (1, 2, ..., P)$ where $r_j (j = 1, 2, ..., n)$ denotes the position of the wildcards in vector $\vec{\sigma}$.

Since $\prod_{r_j \in Pos}(i - r_j) = \sum_{k=0}^{n} \delta_k i^k$, we point out that δ_k denotes the coefficients dependent on Pos. So we have

$$\sum_{i=1, i \notin Pos}^{P} \sigma_i \prod_{r_j \in Pos}(i - r_j) = \sum_{k=0}^{n} \delta_k \sum_{i=1}^{P} \tau_i i^k \quad (1)$$

This is true only if $\sigma_i = \tau_i \vee \sigma_i = *$, for $i = 1, 2, ..., P$.

Let us select a random group element F_i and use σ_i, τ_i as the exponents to conceal the calculation completely. Then (1) will become

$$\prod_{i=1, i \notin Pos}^{P} F_i^{\sigma_i \prod_{r_j \in Pos}(i - r_j)} = \prod_{k=0}^{n} (\prod_{i=1}^{P} F_i^{\tau_i i^k})^{\delta_k}$$

According to Viète's formulas,we construct the coefficients δ_k as follows:

$$\delta_{n-k} = (-1)^k \sum_{1 \leq j_1 \leq j_2 \leq ... \leq j_k \leq n} r_{j_1} r_{j_2} ... r_{j_k}$$

where $0 \leq k \leq n$.

III. SYSTEM MODEL AND THREAT MODEL

A. System Model

In the system, the actors consist of four entities: Trusted Authority (SH-TA), PHR Owner (SH-DO), Cloud Service Provider (SH-CSP) and Data User (SH-DU).

1) SH-TA: SH-TA is an independent and trusted party. It is mainly responsible for the management of attributes and generating public key and the master secret key for the system as well as the attribute private key for users in the system.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3073740, IEEE Transactions on Services Computing

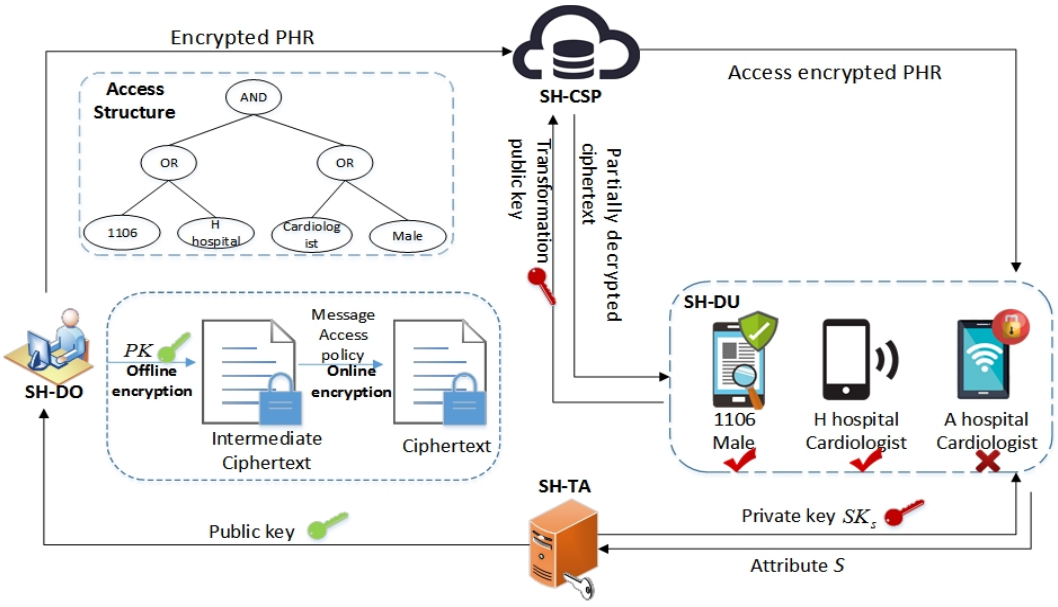

Fig. 2. System model

2) SH-DO: The PHR owner will encrypt his or her own PHR and upload it to the cloud server. The PHR owner will also conduct offline encryption during idle time, pre-process some complex calculations and generate intermediate ciphertext IT. During online encryption, the ciphertext CT will be uploaded to the cloud server. The PHR owner is a fully trusted entity.

3) SH-CSP: SH-CSP is mainly used to store ciphertext and complete the partial calculation of the decryption. During the process of decryption, the user submits a decryption request to the cloud server, which decrypts the ciphertext using the transformation key to send it back to the user. Because no encrypted PHR information is disclosed, the decryption burden on the end user's device is reduced greatly. At the same time, we consider the cloud server to be semi-trusted.

4) SH-DU: The user interacts with the SH-CSP to help decrypt the ciphertext. Only the user's attributes that satisfy the access policy can complete the decryption, obtain the PHR, and verify the decryption result.

B. Overview of SH-LFGS System

1) SH-Setup (λ,U)→ PK, MSK: Let λ and U represent security parameter and attribute universe respectively. The algorithm inputs λ and U, then outputs the public key PK and the master key MSK.

2) SH-Keygen $(PK, MSK, S) \rightarrow SK_s$*:* Input the public key PK, the master key MSK and the user's attribute list $S \subset U$, and output the secret key SK_s.

3) SH-Encrypt $(PK, W, M) \rightarrow CT$*:* SH-DO carries out the algorithm in two steps: offline encryption and online encryption.

- *SH-OffEnc* $(PK) \rightarrow IT$*:* At this stage, the algorithm inputs the public key PK and outputs the intermediary ciphertext IT.
- *SH-OnEnc* $(W, M, IT) \rightarrow CT$*:* The algorithm inputs the access policy W, the message M, and the intermediary ciphertext IT, and outputs the ciphertext CT.

4) SH-Decrypt $(PK, CT, SK_s) \rightarrow M$ *or* $\perp$*:* The decryption takes the public key PK, the ciphertext CT, and the secret key SK_s as input. The user first performs a decryption test (SH-Decrypt Test), and it outputs the message M if $S \models W$. Otherwise, it outputs $\perp$.

- *SH-Decrypt Test:* It returns $\perp$ to abort the decryption algorithms with the strong probability if $S \not\models W$, where W is a completely hidden access policy. Otherwise, it ends by launching the decryption step.

5) SH-$GenTK_{out}$ $(PK, SK_s) \rightarrow TK^s_{out}, SK^s_{out}$*:* This algorithm takes PK and SK_s as input. It outputs the transformation public key TK^s_{out} and the transformation secret key SK^s_{out}.

6) SH-$Transform_{out}$ $(PK, CT, TK^s_{out}) \rightarrow CT_{out}$*:* The algorithm takes the public key PK, the ciphertext CT, and the transformation key TK^s_{out} as input, and outputs a partially decrypted ciphertext CT_{out}.

7) SH-$Decrypt_{out}$ $(PK, CT_{out}, SK^s_{out}) \rightarrow M$ *or* $\perp$*:* Input the public key PK, the ciphertext CT_{out}, and the transformation secret key SK^s_{out}, and output the message M'. Then the user can verify whether $M' = M$. If the equation is true, it outputs the message M. Otherwise, it terminates the algorithm.

C. Security Model

We define the selective security model for our scheme through the following game between the adversary $\mathscr{A}$ and challenge $\mathscr{C}$.

1) Initialization: The adversary $\mathscr{A}$ commits to the challenger $\mathscr{C}$ two challenge access policies W_1, W_2.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3073740, IEEE Transactions on Services Computing

TABLE I
SYSTEM SYMBOLS AND THEIR MEANINGS

Symbol	Description
λ	Security parameter
P	Attribute number
R_1	Maximum number of wildcard
R_2	Maximum number of positive attribute
R_3	Maximum number of negative attribute
SK_s	Secret key
$\overrightarrow{X\Psi}, \overrightarrow{X\Phi}$	Transformed vector of the attribute set
$\vec{\vartheta'}$	Transformed vector of the access policy
IT	Intermediate ciphertext
CT	Uploaded ciphertext
TK^s_{out}	Transformation public key
SK^s_{out}	Transformation private key
CT_{out}	Transformation ciphertext

2) Setup: $\mathscr{C}$ obtains the public key PK and master secret key MSK by running this algorithm and gives PK to the adversary $\mathscr{A}$.

3) Phase 1: $\mathscr{A}$ queries the secret key corresponding to the attribute list S. If $(S \models W_1 \wedge S \models W_2)$ or $(S \not\models W_1 \wedge S \not\models W_2)$, then $\mathscr{C}$ gives $\mathscr{A}$ the secret key SK_s.

4) Challenge: $\mathscr{A}$ submits messages M_1, M_2 to $\mathscr{C}$. If $\mathscr{A}$ obtained the SK_s associated with the attribute list S that $S \models W_1$ and $S \models W_2$, where W_1 and W_2 are defined in phase 1, then it is required that $M_1 = M_2$. $\mathscr{C}$ chooses $\theta = \{1, 2\}$ randomly, and sends CT_θ=SH-Encrypt(PK, W_θ, M_θ) to $\mathscr{A}$.

5) Phase 2: Phase 1 is repeated.

6) Guess: $\mathscr{A}$ outputs a guess θ' of θ and win the game if $\theta = \theta'$. The advantage of adversary is $|Pr[\theta = \theta'] - \frac{1}{2}|$.

IV. DESIGN OF SH-LFGS

First, the SH-TA inputs the security parameter λ and runs the group generator $\mathcal{G}(1^\lambda)$ to obtain (q, g, G, G_T, e). Suppose there are P attributes involved in the universe of the scheme. Suppose also there are two possible values: positive and negative are considered in SH-Keygen phase. In addition, wildcard is considered in access policy, which is mainly in the SH-Encrypt phase. Assume that R_1, R_2, R_3 are the three upper boundaries of wildcards, positive attributes, and negative attributes, respectively.

1) *SH-Setup*: SH-TA uniformly chooses δ_1, δ_2, $\{\upsilon_{1,i}\}$, $\{\upsilon_{2,i}\}$, $\{\mu_{1,i}\}$, $\{\mu_{2,i}\}$ in Z_q, ξ in G and sets $n = R_1 + 3$. Then it chooses a random $\Omega \subset Z_q$ to meet the condition: $\Omega = \delta_1\upsilon_{2,i} - \delta_2\upsilon_{1,i}$, and calculates:

$$V_{1,i} = g^{\upsilon_{1,i}} \qquad \Gamma_{1,i} = g^{\mu_{1,i}} \qquad Z_1 = g^{\delta_1}$$
$$V_{2,i} = g^{\upsilon_{2,i}} \qquad \Gamma_{2,i} = g^{\mu_{2,i}} \qquad Z_2 = g^{\delta_2}$$

SH-TA also calculates h=g^Ω, Λ=e(g, ξ). Therefore, the system public key is PK=(H, g, G, G_T, q, e, h, Λ, $\{V_{1,i}, V_{2,i}, \Gamma_{1,i}, \Gamma_{2,i}\}_{i=1}^n$, $\{Z_i\}_{i=1}^2$), where H is an anti-collusion hash function. The master secret key is MSK=(ξ, $\{\upsilon_{1,i}, \upsilon_{2,i}, \mu_{1,i}, \mu_{2,i}\}_{i=1}^n$, $\{\delta_i\}_{i=1}^2$).

2) *SH-Keygen*: First the user submits an attribute set with n_2 positive attributes and n_3 negative attributes to SH-TA. Assume that the positive and negative attributes occur at positions Ψ=$\{\sigma_1, \sigma_2, ..., \sigma_{n_2}\}$ and Φ=$\{\phi_1, \phi_2, ..., \phi_{n_3}\}$, respectively. Through the Viète's formulas and position set, the attribute set is transformed into two vectors $\overrightarrow{X\Psi} = (x_{\sigma_0}, x_{\sigma_1}, ..., x_{\sigma_{R_1}}, 1, 0)$ and $\overrightarrow{X\Phi} = (x_{\phi_0}, x_{\phi_1}, ..., x_{\phi_{R_1}}, 0, 1)$. For $\{x_{\sigma_k}\}_{k=0}^{R_1}$ and $\{x_{\phi_k}\}_{k=0}^{R_1}$, SH-TA sets: $x_{\sigma_k} = -\sum_{\sigma_i \in \Psi} \sigma_i^k$, and $x_{\phi_k} = +\sum_{\phi_i \in \Phi} \phi_i^k$.

Then SH-TA uniformly chooses $\{\{\eta_i\}_{i=1}^n, d, f, t, r\} \in Z_q$, and outputs the secret key SK_s=$(\kappa_i, \{SK_{1,i}, SK_{2,i}\}_{i=1}^n, SK_3, SK_4)$, where

$$\kappa_i = f(tx_{\sigma_i} + rx_{\phi_i})$$
$$SK_{1,i} = g^{-2\delta_2\eta_i} g^{d(tx_{\sigma_i} + rx_{\phi_i})\upsilon_{2,i}}$$
$$SK_{2,i} = g^{2\delta_1\eta_i} g^{-d(tx_{\sigma_i} + rx_{\phi_i})\upsilon_{1,i}}$$
$$SK_3 = \xi \cdot \prod_{i=1}^n SK_{1,i}^{-\mu_{1,i}} SK_{2,i}^{-\mu_{2,i}}$$
$$SK_4 = \prod_{i=1}^n g^{-2\eta_i}$$

3) *SH-Encrypt*:SH-DO carries out the algorithm in two steps: offline encryption(SH-OffEnc) and online encryption(SH-OnEnc):

- *SH-OffEnc*: SH-DO uniformly chooses $\{\alpha_1, \alpha_2, \zeta, \{\beta_i\}_{i=1}^n\} \in Z_q$ and generates the intermediate ciphertext IT=$(\{C_{1,i}, C_{2,i}, \beta_i'\}_{i=1}^n, \zeta, C_3, C_4, \tilde{C})$, where

$$C_{1,i} = V_{1,i}^{\alpha_1} \Gamma_{1,i}^{\alpha_2} Z_1^{\zeta\beta_i} \qquad C_{2,i} = V_{2,i}^{\alpha_1} \Gamma_{2,i}^{\alpha_2} Z_2^{\zeta\beta_i}$$
$$C_3 = g^{\alpha_2} \qquad \tilde{C} = \Lambda^{\alpha_2} \qquad C_4 = h^{\alpha_1}$$

- *SH-OnEnc*: First, SH-Do specifies the access policy W that contains $\bar{n}_1 \leq R_1$ wildcards, $\bar{n}_2 \leq R_2$ positive, and $\bar{n}_3 \leq R_3$ negative attributes. Assume that their positions are labeled as: Δ'=$\{\varpi_1', ..., \varpi_{\bar{n}_1}'\}$, Ψ'=$\{\sigma_1', ..., \sigma_{\bar{n}_2}'\}$, Φ'=$\{\phi_1', ..., \phi_{\bar{n}_3}'\}$. The access policy W can be transformed into vector $\vec{\vartheta'} = (\gamma_0', \gamma_1', ..., \gamma_{n1}', 0_{n1+1}', ..., 0_{R_1}', \prod_{\Psi'}, \prod_{\Phi'})$ by Viète's formulas and position set, where the coefficient $(\gamma_0', \gamma_1', ..., \gamma_{\bar{n}_1}')$ is calculated as follows:

$$\begin{cases} \gamma_{\bar{n}_1}' = & 1 \\ \gamma_{\bar{n}_1-1}' = & -(\varpi_1' + \varpi_2' + ... + \varpi_{\bar{n}_1}') \\ \gamma_{\bar{n}_1-2}' = & (\varpi_1'\varpi_2' + \varpi_1'\varpi_3' + ... + \varpi_{\bar{n}_1-1}'\varpi_{\bar{n}_1}') \\ ..... & \\ \gamma_0' = & -(\varpi_1'\varpi_2'...\varpi_{\bar{n}_1}') \end{cases}$$

Then, the SH-DO needs to calculate:

$$\prod_{\Psi'} = +\sum_{i \in \Psi'} \prod_{\varpi_j \in \Delta'} (i - \varpi_j')$$
$$\prod_{\Phi'} = -\sum_{i \in \Phi'} \prod_{\varpi_j \in \Delta'} (i - \varpi_j')$$

Finally, SH-DO uniformly chooses b,$\gamma \in Z_q$ and calculates ciphertext as follows:

$$C = (M||\gamma) \bigoplus \tilde{C} \qquad H_i = h^{b\vartheta_i}$$
$$E = g^{H(M)} h^{H(\gamma)} \qquad c_i = \zeta(\vartheta_i' - \beta_i)$$

SH-DO sends the ciphertext CT to the cloud server after encryption, where CT=$(C, \{C_{1,i}, C_{2,i}, c_i, H_i\}_{i=1}^n, C_3, C_4, E)$.

4) *SH-Decrypt*: SH-DU decrypts the ciphertext CT using the secret key SK_s. The process is as follows.

- *SH-Decrypt Test*: Before outsourcing decryption, SH-DU first verifies whether $e(\prod_{i=1}^n (H_i)^{\kappa_i}, C_4)$=1 is true. If the equation is true, SH-DU will perform the decryption outsourcing operation, otherwise, the algorithm will be terminated.

5) *SH-GenTK$_{out}$*: SH-DU generates his or her transformation public key TK^s_{out} and the transformation secret key SK^s_{out} with his or her secret key SK_s. He or she uniformly chooses $z \in Z_q$ and computes the transformation public key as $TK^s_{out} = (SK'_{1,i}, SK'_{2,i}, SK'_3, SK'_4)$ where

$$SK'_{1,i} = SK^{1/z}_{1,i} \qquad SK'_{2,i} = SK^{1/z}_{2,i}$$
$$SK'_3 = SK^{1/z}_3 \qquad SK'_4 = SK^{1/z}_4$$

and generates the transformation secret key $SK^s_{out} = z$.

6) *SH-Transform$_{out}$*: Given the transformation key TK^s_{out} and ciphertext CT, SH-CSP calculates the transformation ciphertext CT_{out} as follows:

$$\begin{aligned}&\textstyle\prod_{i=1}^{n}(e(C_{1,i}Z_1^{c_i}, SK'_{1,i}) \cdot e(C_{2,i}Z_2^{c_i}, SK'_{2,i}))\\ &\cdot e(C_3, SK'_3) \cdot e(C_4, SK'_4)\\ &= e(g,\xi)^{\alpha_2/z}\end{aligned}$$

7) *SH-Decrypt$_{out}$*: SH-DU decrypts the message M' with the transformed ciphertext CT_{out} sent by SH-CSP and the existing transformation secret key SK^s_{out}. $M'||\gamma'$=C/CT^z_{out}. If $E \neq g^{H(M')}h^{H(\gamma')}$, it outputs $\perp$. Otherwise, it outputs the message M.

Correctness Analysis:

$$\begin{aligned}\textstyle\bigcap &= C_{1,i} \cdot Z_1^{c_i}\\ &= V_{1,i}^{\alpha_1}\Gamma_{1,i}^{\alpha_2}Z_1^{\zeta\beta_i} \cdot Z_1^{\zeta(\vartheta'_i-\beta_i)}\\ &= g^{\upsilon_{1,i}\alpha_1}g^{\mu_{1,i}\alpha_2}g^{\delta_1\zeta\beta_i}g^{\zeta\delta_1(\vartheta'_i-\beta_i)}\\ &= g^{\upsilon_{1,i}\alpha_1}g^{\mu_{1,i}\alpha_2}g^{\delta_1\zeta\vartheta i'}\end{aligned}$$

$$\begin{aligned}\textstyle\bigcup &= C_{2,i} \cdot Z_1^{c_i}\\ &= V_{2,i}^{\alpha_1}\Gamma_{2,i}^{\alpha_2} \cdot Z_2^{\zeta\beta_i} \cdot Z_2^{\zeta(\vartheta'_i-\beta_i)}\\ &= g^{\upsilon_{2,i}\alpha_1}g^{\mu_{2,i}\alpha_2}g^{\delta_2\zeta\beta_i}g^{\zeta\delta_2(\vartheta'_i-\beta_i)}\\ &= g^{\upsilon_{2,i}\alpha_1}g^{\mu_{2,i}\alpha_2}g^{\delta_2\zeta\vartheta i'}\end{aligned}$$

$$\begin{aligned}&e(\textstyle\bigcap, SK_{1,i})\\ &= e(g^{\upsilon_{1,i}\alpha_1}g^{\mu_{1,i}\alpha_2}g^{\delta_1\zeta\vartheta i'}, g^{-2\delta_2\eta_i}g^{d(tx_{\sigma_i}+rx_{\phi_i})\upsilon_{2,i}})\\ &= e(g,g)^{-2\upsilon_{1,i}\alpha_1\delta_2\eta_i} \cdot e(g,g)^{\upsilon_{1,i}\alpha_1 d(tx_{\sigma_i}+rx_{\phi_i})\upsilon_{2,i}}\\ &\cdot e(g, SK_{1,i})^{\mu_{1,i}\alpha_2} \cdot e(g,g)^{-2\delta_1\zeta\vartheta'_i\delta_2\eta_i}\\ &\cdot e(g,g)^{\delta_1\zeta\vartheta'_i d(tx_{\sigma_i}+rx_{\phi_i})\upsilon_{2,i}}\end{aligned}$$

$$\begin{aligned}&e(\textstyle\bigcup, SK_{2,i})\\ &= e(g^{\upsilon_{2,i}\alpha_1}g^{\mu_{2,i}\alpha_2}g^{\delta_2\zeta\vartheta i'}, g^{2\delta_1\eta_i}g^{-d(tx_{\sigma_i}+rx_{\phi_i})\upsilon_{1,i}})\\ &= e(g,g)^{2\upsilon_{2,i}\alpha_1\delta_1\eta_i} \cdot e(g,g)^{-\upsilon_{2,i}\alpha_1 d(tx_{\sigma_i}+rx_{\phi_i})\upsilon_{1,i}}\\ &\cdot e(g, SK_{2,i})^{\mu_{2,i}\alpha_2} \cdot e(g,g)^{2\delta_2\zeta\vartheta'_i\delta_1\eta_i}\\ &\cdot e(g,g)^{-\delta_2\zeta\vartheta'_i d(tx_{\sigma_i}+rx_{\phi_i})\upsilon_{1,i}}\end{aligned}$$

$$\begin{aligned}&\textstyle\prod_{i=1}^{n} e(\bigcap, SK_{1,i}) \cdot e(\bigcup, SK_{2,i})\\ &= e(g,g)^{2\alpha_1\Omega\sum\eta_i}e(g,g)^{\zeta d\Omega\sum(\vartheta'_i(tx_{\sigma_i}+rx_{\phi_i}))}\\ &\cdot\textstyle\prod_{i=1}^{n} e(g, SK_{1,i})^{\mu_{1,i}\alpha_2}e(g, SK_{2,i})^{\mu_{2,i}\alpha_2}\end{aligned}$$

Since

$$\begin{aligned}&e(C_3, SK_3) \cdot e(C_4, SK_4)\\ &= e(g^{\alpha_2}, \xi \cdot \textstyle\prod_{i=1}^{n} SK_{1,i}^{-\mu_{1,i}}SK_{2,i}^{-\mu_{2,i}})\\ &\cdot e((g^{\Omega})^{\alpha_1}, \textstyle\prod_{i=1}^{n} g^{-2\eta_i})\\ &= e(g,\xi)^{\alpha_2}\textstyle\prod_{i=1}^{n} e(g, SK_{1,i})^{-\mu_{1,i}\alpha_2} \cdot e(g, SK_{2,i})^{-\mu_{2,i}\alpha_2}\\ &\cdot e(g,g)^{-2\alpha_1\Omega\sum\eta_i}\end{aligned}$$

We have

$$\begin{aligned}&C \Big/ e(C_3, SK_3)e(C_4, SK_4)\textstyle\prod_{j=1}^{2}\prod_{i=1}^{n} e(C_{j,i}Z_j^{c_i}, SK_{j,i})\\ &= M \Big/ e(g,g)^{\zeta d\Omega\sum(\vartheta'_i(tx_{\sigma_i}+rx_{\phi_i}))}\end{aligned}$$

Therefore, the message M can be obtained if $(\overrightarrow{\vartheta'}, t\overrightarrow{X\Psi} + r\overrightarrow{X\Phi})$=0. This means that the user's attribute list satisfies the access policy.

V. PROOF OF SECURITY

The proof of the security of the SH-LFGS scheme can be considered in two cases, namely: $M_1 = M_2$ and $M_1 \neq M_2$. In the case $M_1 = M_2$, where only policy hiding is considered, and its security can be proven by considering the following games from $Game_A$ to $Game_D$, whereas in another case, its safety proof needs to be considered from $Game_A$ to $Game_E$. Next, we describe each game in detail, including the challenge ciphertext $CT = (C, C_3, C_4, \{C_{1,i}, C_{2,i}, c_i\}_{i=1}^n)$. The c_i in the challenge ciphertext will be given in the special proof.

$Game_A$: The challenge ciphertext CT_A had come into being under $\vec{\vartheta}$ and M_1, where $C = M_1 \cdot \Lambda^{-\alpha_2}$, $C_3 = g^{\alpha_2}$, $C_4 = h^{\alpha_1}$, $C_{1,i} = V_{1,i}^{\alpha_1}\Gamma_{1,i}^{\alpha_2}Z_1^{\zeta\beta'_i}$, $C_{2,i} = V_{2,i}^{\alpha_1}\Gamma_{2,i}^{\alpha_2}Z_2^{\zeta\beta'_i}$.

$Game_B$: The challenge ciphertext CT_B had come into being under $\vec{\vartheta}$ and a random message $R \in G_T$, where $C = R$, $C_3 = g^{\alpha_2}$, $C_4 = h^{\alpha_1}$, $C_{1,i} = V_{1,i}^{\alpha_1}\Gamma_{1,i}^{\alpha_2}Z_1^{\zeta\beta'_i}$, $C_{2,i} = V_{2,i}^{\alpha_1}\Gamma_{2,i}^{\alpha_2}Z_2^{\zeta\beta'_i}$.

$Game_C$: The challenge ciphertext CT_C had come into being under $\vec{0}$ and a random message $R \in G_T$, where $C = R$, $C_3 = g^{\alpha_2}$, $C_4 = h^{\alpha_1}$, $C_{1,i} = V_{1,i}^{\alpha_1}\Gamma_{1,i}^{\alpha_2}Z_1^{\zeta\beta'_i}$, $C_{2,i} = V_{2,i}^{\alpha_1}\Gamma_{2,i}^{\alpha_2}Z_2^{\zeta\beta'_i}$.

$Game_D$: The challenge ciphertext CT_D had come into being under $\vec{\varrho}$ and a random message $R \in G_T$, where $C = R$, $C_3 = g^{\alpha_2}$, $C_4 = h^{\alpha_1}$, $C_{1,i} = V_{1,i}^{\alpha_1}\Gamma_{1,i}^{\alpha_2}Z_1^{\zeta\beta'_i}$, $C_{2,i} = V_{2,i}^{\alpha_1}\Gamma_{2,i}^{\alpha_2}Z_2^{\zeta\beta'_i}$.

$Game_E$: The challenge ciphertext CT_E had come into being under $\vec{\varrho}$ and a message M_2, where $C = M_2 \cdot \Lambda^{-\alpha_2}$, $C_3 = g^{\alpha_2}$, $C_4 = h^{\alpha_1}$, $C_{1,i} = V_{1,i}^{\alpha_1}\Gamma_{1,i}^{\alpha_2}Z_1^{\zeta\beta'_i}$, $C_{2,i} = V_{2,i}^{\alpha_1}\Gamma_{2,i}^{\alpha_2}Z_2^{\zeta\beta'_i}$.

Theorem 1: If $|Adv_{\mathscr{A}}^{Game_A} - Adv_{\mathscr{A}}^{Game_B}| \leq \epsilon$, where ϵ is negligible, then the DBDH assumption holds.

Proof: We construct a simulation algorithm $\mathscr{B}$ and give $\mathscr{B}$ a tuple $(g, A = g^{\alpha}, B = g^{\beta}, C = g^{\gamma}, T) \in G$. The specific simulation process is shown below.

Initialization: $\mathscr{A}$ submits two challenge attribute vectors $\vec{v_1}$ and $\vec{v_2}$ to $\mathscr{B}$, where $\vec{v_1}$ and $\vec{v_2}$ are converted from the two challenge access policies W_1 and W_2 selected by the adversary $\mathscr{A}$.

Setup: $\mathscr{B}$ selects elements: δ_1, δ_2, $\{\upsilon_{1,i}\}_{i=1}^n$, $\{\mu_{1,i}\}_{i=1}^n$, $\{\mu_{2,i}\}_{i=1}^n$, $\lambda = Z_q$. Then select the random $\Omega \in Z_q$ to obtain $\{\upsilon_{2,i}\}_{i=1}^n$ under the condition: $\Omega = \delta_1\upsilon_{2,i} - \delta_2\upsilon_{1,i}$ Then $\mathscr{B}$ sets:

$$V_{1,i} = g^{\upsilon_{1,i}}, \qquad V_{2,i} = g^{\upsilon_{2,i}}$$
$$\Gamma_{1,i} = (g^{\beta})^{\vartheta_i\delta_1}g^{\mu_{1,i}}, \quad \Gamma_{2,i} = (g^{\beta})^{\vartheta_i\delta_2}g^{\mu_{2,i}}$$

where i= 1 to n, and sets:

$$Z_1 = g^{\delta_1}, \quad Z_2 = g^{\delta_2}, \quad h = g^{\Omega}$$
$$\Lambda = e(g^{\alpha}, g^{\beta})^{-\Omega}e(g,g)^{\lambda}$$

Due to the existence of random exponents, each public key component is appropriately assigned as:

$$\overline{\mu_{1,i}} = \vartheta_i \delta_1 \beta + \mu_{1,i}, \quad \overline{\mu_{2,i}} = \vartheta_i \delta_2 \beta + \mu_{2,i}, \quad \xi = g^{\alpha\beta\Omega} g^{\lambda}$$

Finally, $\mathscr{B}$ sends PK to $\mathscr{A}$, where PK=(g, G, G_T, q, e, h, Λ, $\{V_{1,i}, V_{2,i}, \Gamma_{1,i}, \Gamma_{2,i}\}_{i=1}^{n}$, Z_1, Z_2).

Phase 1: $\mathscr{A}$ queries the secret key corresponding to the attribute list S. Consider two vectors $\overrightarrow{x\psi} = (x_{\psi_1}, x_{\psi_2}, ..., x_{\psi_n})$ and $\overrightarrow{x\phi} = (x_{\phi_1}, x_{\phi_2}, , ..., x_{\phi_n})$, where $\overrightarrow{x\psi}$ and $\overrightarrow{x\phi}$ are derived from the transformation of the attribute set S into vectors by Viète's formula and position set. $\mathscr{A}$ can request the secret key inquiry as long as $(\vec{\vartheta}, \overrightarrow{x\psi}) = (\vec{\vartheta}, \overrightarrow{x\phi}) = c_u \neq 0$. $\mathscr{B}$ selects random elements $\{\eta_i\}_{i=1}^{n}$, d' t, r from Z_q. Then $\mathscr{B}$ computes:

$$\begin{aligned} SK_{1,i} &= g^{-2\delta_2\eta_i} g^{(\frac{\alpha}{2c_u}+d')(tx_{\psi_i}+rx_{\phi_i})\upsilon_{2,i}} \\ &= g^{-2\delta_2\eta_i} g^{d'(tx_{\psi_i}+rx_{\phi_i})\upsilon_{2,i}} g^{\frac{\alpha}{2c_u}(tx_{\psi_i}+rx_{\phi_i})\upsilon_{2,i}} \\ &= g^{\frac{\alpha}{2c_u}(tx_{\psi_i}+rx_{\phi_i})\upsilon_{2,i}} SK'_{1,i} \end{aligned}$$

$$\begin{aligned} SK_{2,i} &= g^{2\delta_1\eta_i} g^{-(\frac{\alpha}{2c_u}+d')(tx_{\psi_i}+rx_{\phi_i})\upsilon_{1,i}} \\ &= g^{2\delta_1\eta_i} g^{-d'(tx_{\psi_i}+rx_{\phi_i})\upsilon_{1,i}} g^{-\frac{\alpha}{2c_u}(tx_{\psi_i}+rx_{\phi_i})\upsilon_{1,i}} \\ &= g^{-\frac{\alpha}{2c_u}(tx_{\psi_i}+rx_{\phi_i})\upsilon_{1,i}} SK'_{2,i} \end{aligned}$$

which implicitly sets: $d = \frac{\alpha}{2c_u} + d'$.

Then SK_3 is computed as:

$$\begin{aligned} & SK_{1,i}^{-\overline{\mu_{1,i}}} SK_{2,i}^{-\overline{\mu_{2,i}}} \\ &= (g^{\frac{\alpha}{2c_u}(tx_{\psi_i}+rx_{\phi_i})\upsilon_{2,i}} SK'_{1,i})^{-\overline{\mu_{1,i}}} \\ &\quad \cdot (g^{-\frac{\alpha}{2c_u}(tx_{\psi_i}+rx_{\phi_i})\upsilon_{1,i}} SK'_{2,i})^{-\overline{\mu_{2,i}}} \\ &= (g^{\frac{\alpha}{2c_u}(tx_{\psi_i}+rx_{\phi_i})\upsilon_{2,i}})^{-(\vartheta_i\delta_1\beta+\mu_{1,i})} \\ &\quad \cdot (g^{-\frac{\alpha}{2c_u}(tx_{\psi_i}+rx_{\phi_i})\upsilon_{1,i}})^{-(\vartheta_i\delta_2\beta+\mu_{2,i})} \\ &\quad \cdot (SK'_{1,i})^{-\overline{\mu_{1,i}}} \cdot (SK'_{2,i})^{-\overline{\mu_{2,i}}} \\ &= g^{-\frac{\alpha\beta\Omega}{2c_u}\vartheta_i(tx_{\psi_i}+rx_{\phi_i})} \\ &\quad \cdot g^{\frac{\alpha}{2c_u}(tx_{\psi_i}+rx_{\phi_i})(\upsilon_{1,i}\mu_{2,i}-\upsilon_{2,i}\mu_{1,i})} \\ &\quad \cdot (SK'_{1,i})^{-\overline{\mu_{1,i}}} \cdot (SK'_{2,i})^{-\overline{\mu_{2,i}}} \end{aligned}$$

By using $\xi = g^{-\alpha\beta\Omega} g^{\lambda}$, we can calculate SK_3 as:

$$\begin{aligned} SK_3 &= \xi \cdot \textstyle\prod_{i=1}^{n} SK_{1,i}^{-\overline{\mu_{1,i}}} SK_{2,i}^{-\overline{\mu_{2,i}}} \\ &= g^{-\alpha\beta\Omega} g^{\lambda} \textstyle\prod_{i=1}^{n} g^{-\frac{\alpha\beta\Omega}{2cu}\vartheta_i(tx_{\psi_i}+rx_{\phi_i})} \\ &\quad \cdot g^{\frac{\alpha}{2c_u}(tx_{\psi_i}+rx_{\phi_i})(\upsilon_{1,i}\mu_{2,i}-\upsilon_{2,i}\mu_{1,i})} \\ &\quad \cdot (SK'_{1,i})^{-\overline{\mu_{1,i}}} \cdot (SK'_{2,i})^{-\overline{\mu_{2,i}}} \end{aligned}$$

Then $\mathscr{B}$ submits the secret key $SK_s = (\{SK_{1,i}, SK_{2,i}\}_{i=1}^{n}, SK_3, SK_4)$ corrected to the attribute set S to $\mathscr{A}$.

Challenge ciphertext: $\mathscr{A}$ submits two challenge messages M_1, M_2 to $\mathscr{B}$. Then $\mathscr{B}$ selects random $\theta \in \{1, 2\}$ and calculates the ciphertext of the plaintext message M_θ under the access policy W_θ and $\mathscr{B}$ selects random $\tilde{\beta_1}$, $\tilde{\beta_2}$, $\tilde{\beta_n}$, $\tilde{\alpha_1}$, $\tilde{\alpha_2}$, $\tilde{\zeta}$ from Z_q, which implicitly sets:

$$\alpha_1 = \tilde{\alpha_1}, \quad \alpha_2 = \gamma, \quad \zeta = -\beta\gamma + \tilde{\zeta}$$

Then $\mathscr{B}$ sets $C_3 = g^{\gamma} = g^{\alpha_2}$,$C_4 = \mathrm{g}^{\Omega\alpha_1}$ and calculates:

$$\begin{aligned} C_{1,i} &= (g^{\upsilon_{1,i}})^{\alpha_1} ((g^{\beta})^{\vartheta_i\delta_1} g^{\mu_{1,i}})^{\gamma} g^{\tilde{\beta_i}\delta_1(-\beta\gamma+\tilde{\zeta})} \\ &= V_{1,i}^{\alpha_1} \Gamma_{1,i}^{\alpha_2} Z_1^{\tilde{\beta_i}\zeta} \\ C_{2,i} &= (g^{\upsilon_{2,i}})^{\alpha_1} ((g^{\beta})^{\vartheta_i\delta_2} g^{\mu_{2,i}})^{\gamma} g^{\tilde{\beta_i}\delta_2(-\beta\gamma+\tilde{\zeta})} \\ &= V_{2,i}^{\alpha_1} \Gamma_{2,i}^{\alpha_2} Z_2^{\tilde{\beta_i}\zeta} \\ c_i &= \zeta(\vartheta_i - \tilde{\beta_i}) \\ C &= M_\theta T^{\Omega} e(g, g^{\gamma})^{\lambda} \end{aligned}$$

Finally, $\mathscr{B}$ sends the ciphertext CT to $\mathscr{A}$.

Phase 2: *Phase 1* is repeated.

Guess: The adversary $\mathscr{A}$ outputs a guess bit $\theta' \in \{1, 2\}$ and wins the game if $\theta' = \theta$. If $T = e(g, g)^{\alpha\beta\gamma}$, this means the simulation algorithm $\mathscr{B}$ is the same as $Game_A$. Or if T is a random number in G_T, then $\mathscr{B}$ is the same as $Game_B$.

Therefore, $\mathscr{B}$ can solve the DBDH problem if $\mathscr{A}$ can distinguish between these two games.

Theorem 2: If $|Adv_{\mathscr{A}}^{Game_B} - Adv_{\mathscr{A}}^{Game_C}| \leq \epsilon$, where ϵ is negligible, then the DLIN assumption holds.

Proof: We construct a simulation algorithm $\mathscr{B}$ and give $\mathscr{B}$ a tuple $(g, A = g^{\alpha}, B = g^{\beta}, C = g^{\alpha\gamma}, D = g^{\eta}, T) \in G$. $\mathscr{B}$ can solve the DLIN problem with advantage ϵ. The specific simulation process is shown below.

Initialization: $\mathscr{A}$ submits two challenge attribute vectors $\vec{\vartheta}$, $\vec{0}$ to $\mathscr{B}$, where $\vec{\vartheta}$ and $\vec{0}$ are converted from the two challenge access policies W_1 and W_2 selected by the adversary $\mathscr{A}$.

Setup $\mathscr{B}$ selects elements: δ_1, δ_2, $\{\upsilon_{1,i}\}_{i=1}^{n}$, $\{\mu_{1,i}\}_{i=1}^{n}$, $\{\mu_{2,i}\}_{i=1}^{n}$, $\lambda \in Z_q$, then selects the random $\Omega \in Z_q$ to obtain $\{\upsilon_{2,i}\}_{i=1}^{n}$ under the condition: $\Omega = \delta_1\mu_{2,i} - \delta_2\mu_{1,i}$

Then $\mathscr{B}$ sets:

$$\begin{aligned} &V_{1,i} = (g^{\alpha})^{\upsilon_{1,i}} (g^{\beta})^{\delta_1\vartheta_i}, \qquad V_{2,i} = (g^{\alpha})^{\upsilon_{2,i}} (g^{\beta})^{\delta_2\vartheta_i} \\ &\Gamma_{1,i} = g^{\mu_{1,i}} (g^{\beta})^{\delta_1\vartheta_i}, \qquad \Gamma_{2,i} = g^{\mu_{2,i}} (g^{\beta})^{\delta_2\vartheta_i} \\ &Z_1 = g^{\delta_1}, \quad Z_2 = g^{\delta_2}, \quad h = (g^{\alpha})^{\Omega}, \quad \xi = g^{\lambda} \end{aligned}$$

Due to the existence of random exponents, each public key component is appropriately assigned as:

$$\begin{aligned} &\overline{\upsilon_{1,i}} = \alpha\upsilon_{1,i} + \delta_1\beta\vartheta_i, \qquad \overline{\upsilon_{2,i}} = \alpha\upsilon_{2,i} + \delta_2\beta\vartheta_i \\ &\overline{\mu_{1,i}} = \vartheta_i\delta_1\beta + \mu_{1,i}, \qquad \overline{\mu_{2,i}} = \vartheta_i\delta_2\beta + \mu_{2,i} \end{aligned}$$

Finally, $\mathscr{B}$ sends PK to $\mathscr{A}$, where PK=(g, G, G_T, q, e, h, Λ, $\{V_{1,i}, V_{2,i}, \Gamma_{1,i}, \Gamma_{2,i}\}_{i=1}^{n}$, Z_1, Z_2).

Phase 1: $\mathscr{A}$ queries the private key corresponding to the attribute list S. Given two vectors $\overrightarrow{x\psi} = (x_{\psi_1}, x_{\psi_2}, ..., x_{\psi_n})$ and $\overrightarrow{x\phi} = (x_{\phi_1}, x_{\phi_2}, , ..., x_{\phi_n})$, where $\overrightarrow{x\psi}$ and $\overrightarrow{x\phi}$ are derived from the transformation of the attribute set S into vectors by Viète's formula and position set. $\mathscr{B}$ selects random exponents $\{\eta_i\}_{i=1}^{n}$, d, t, r from Z_q and calculates:

$$\begin{aligned} SK_{1,i} &= g^{-2\delta_2(\vartheta_i\beta(tx_{\psi_i}+rx_{\phi_i})+\alpha\eta'_i)} g^{d(tx_{\psi_i}+rx_{\phi_i})\upsilon_{2,i}} \\ &= g^{-2\delta_2\vartheta_i\beta(tx_{\psi_i}+rx_{\phi_i})} g^{-2\delta_2\eta'_i\alpha} g^{d(tx_{\psi_i}+rx_{\phi_i})\upsilon_{2,i}} \\ &= g^{-2\delta_2\vartheta_i\beta(tx_{\psi_i}+rx_{\phi_i})} \cdot SK'_{1,i} \end{aligned}$$

$$\begin{aligned} SK_{2,i} &= g^{2\delta_1(\vartheta_i\beta(tx_{\psi_i}+rx_{\phi_i})+\alpha\eta'_i)} g^{-d(tx_{\psi_i}+rx_{\phi_i})\upsilon_{1,i}} \\ &= g^{2\delta_1\vartheta_i\beta(tx_{\psi_i}+rx_{\phi_i})} g^{2\delta_1\eta'_i\alpha} g^{-d(tx_{\psi_i}+rx_{\phi_i})\upsilon_{1,i}} \\ &= g^{2\delta_1\vartheta_i\beta(tx_{\psi_i}+rx_{\phi_i})} \cdot SK'_{2,i} \end{aligned}$$

which implicitly sets: $\eta_i = \vartheta_i\beta(tx_{\psi_i}+rx_{\phi_i})+\alpha\eta'_i$. Then SK_3 and SK_4 can be calculated as:

$$SK_3 = \xi \textstyle\prod_{i=1}^{n} SK_{1,i}^{-\overline{\mu_{1,i}}} SK_{2,i}^{-\overline{\mu_{2,i}}}$$

$$SK_4 = \textstyle\prod_{i=1}^{n} g^{-2\eta_i} = \prod_{i=1}^{n} g^{-2\vartheta_i\beta(tx_{\psi_i}+rx_{\phi_i})+\alpha\eta'_i}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3073740, IEEE Transactions on Services Computing

For SK_3, we can calculate:

$$\begin{aligned}
&SK_{1,i}^{-\overline{\mu_{1,i}}} SK_{2,i}^{-\overline{\mu_{2,i}}} \\
&= (g^{-2\delta_2\vartheta_i\beta(tx_{\psi_i}+rx_{\phi_i})})^{-(\vartheta_i\delta_1\beta+\mu_{1,i})} \\
&\cdot (g^{-2\delta_2\eta_i'\alpha})^{-(\vartheta_i\delta_1\beta+\mu_{1,i})} \cdot (g^{d(tx_{\psi_i}+rx_{\phi_i})\upsilon_{2,i}})^{-(\vartheta_i\delta_1\beta+\mu_{1,i})} \\
&\cdot (g^{2\delta_1\vartheta_i\beta(tx_{\psi_i}+rx_{\phi_i})})^{-(\vartheta_i\delta_2\beta+\mu_{2,i})} \cdot (g^{2\delta_1\eta_i'\alpha})^{-(\vartheta_i\delta_2\beta+\mu_{2,i})} \\
&\cdot (g^{-d(tx_{\psi_i}+rx_{\phi_i})\upsilon_{1,i}})^{-(\vartheta_i\delta_2\beta+\mu_{2,i})} \\
&= (g^{d(tx_{\psi_i}+rx_{\phi_i})\upsilon_{2,i}})^{-(\vartheta_i\delta_1\beta+\mu_{1,i})} \cdot g^{-2\vartheta_i\beta(tx_{\psi_i}+rx_{\phi_i})\Omega} \\
&\cdot g^{-2\alpha\eta'\Omega} \cdot (g^{-d(tx_{\psi_i}+rx_{\phi_i})\upsilon_{1,i}})^{-(\vartheta_i\delta_2\beta+\mu_{2,i})} \\
&= (g^{d(tx_{\psi_i}+rx_{\phi_i})\upsilon_{2,i}})^{-(\vartheta_i\delta_1\beta+\mu_{1,i})} \cdot g^{-(2\vartheta_i\beta(tx_{\psi_i}+rx_{\phi_i})+\alpha\eta')\Omega} \\
&\cdot (g^{-d(tx_{\psi_i}+rx_{\phi_i})\upsilon_{1,i}})^{-(\vartheta_i\delta_2\beta+\mu_{2,i})}
\end{aligned}$$

Since $\xi = g^{\lambda}$, SK_3 can be calculated as:

$$\begin{aligned}
SK_3 &= \xi \prod_{i=1}^{n} SK_{1,i}^{-\overline{\mu_{1,i}}} SK_{2,i}^{-\overline{\mu_{2,i}}} \\
&= g^{\lambda} \prod_{i=1}^{n} (g^{d(tx_{\psi_i}+rx_{\phi_i})\upsilon_{2,i}})^{-(\vartheta_i\delta_1\beta+\mu_{1,i})} \\
&\cdot (g^{-d(tx_{\psi_i}+rx_{\phi_i})\upsilon_{1,i}})^{-(\vartheta_i\delta_2\beta+\mu_{2,i})} \\
&\cdot g^{-(2\vartheta_i\beta(tx_{\psi_i}+rx_{\phi_i})+\alpha\eta')\Omega}
\end{aligned}$$

Then $\mathscr{B}$ submits the secret key $SK_s = (\{SK_{1,i}, SK_{2,i}\}_{i=1}^{n}, SK_3, SK_4)$ to $\mathscr{A}$.

Challenge Ciphertext: $\mathscr{A}$ submits two challenge messages M_1 and M_2 to $\mathscr{B}$. Then $\mathscr{B}$ selects random $\theta \in \{1,2\}$ and calculates the ciphertext of the plaintext message M_θ under the access policy W_θ. And $\mathscr{B}$ selects random $\{\tilde{\beta}_i\}_{i=1}^{n}$, $\{\tilde{\alpha}_i\}_{i=1}^{2}$, $\tilde{\zeta}$ in Z_q, which implicitly sets: $\alpha_1 = \gamma$, $\alpha_2 = \eta$.
Then $\mathscr{B}$ sets $C_3 = g^{\eta} = g^{\alpha_2}$, $C_4 = (g^{\alpha\gamma})^{\Omega} = h^{\alpha_1}$. And $\mathscr{B}$ computes:

$$\begin{aligned}
C_{1,i} &= (g^{\alpha\upsilon_{1,i}})^{\gamma} (g^{\eta})^{\mu_{1,i}} T^{\delta_1\vartheta_i} \\
C_{2,i} &= (g^{\alpha\upsilon_{2,i}})^{\gamma} (g^{\eta})^{\mu_{2,i}} T^{\delta_2\vartheta_i}
\end{aligned}$$

If $T = g^{\beta(\gamma+\eta)} g^{r_1}$, where r_1 is randomly selected in Z_q, then $\mathscr{B}$ simulates $Game_B$ with $\zeta = r_1$, $\vartheta_i' = \vartheta_i - \tilde{\beta}_i$:

$$\begin{aligned}
C_{1,i} &= (g^{\alpha\upsilon_{1,i}})^{\gamma} (g^{\eta})^{\mu_{1,i}} (g^{\beta(\gamma+\eta)} g^{r_1})^{\delta_1\vartheta_i} (g^{-\tilde{\beta}_i})^{\delta_1 r_1} \\
&= V_{1,i}^{\alpha_1} \Gamma_{1,i}^{\alpha_2} Z_1^{\vartheta_i'\zeta} \\
C_{2,i} &= (g^{\alpha\upsilon_{2,i}})^{\gamma} (g^{\eta})^{\mu_{2,i}} (g^{\beta(\gamma+\eta)} g^{r_1})^{\delta_2\vartheta_i} (g^{-\tilde{\beta}_1})^{\delta_1 r_1} \\
&= V_{2,i}^{\alpha_1} \Gamma_{2,i}^{\alpha_2} Z_2^{\vartheta_i'\zeta} \\
c_i &= \zeta\tilde{\beta}_i
\end{aligned}$$

If $T = g^{\beta(\gamma+\eta)}$, then $\mathscr{B}$ simulates $Game_C$, and ζ' is randomly chosen:

$$\begin{aligned}
C_{1,i} &= (g^{\alpha\upsilon_{1,i}})^{\gamma} (g^{\eta})^{\mu_{1,i}} (g^{\beta(\gamma+\eta)})^{\delta_1\vartheta_i} g^{-\delta_1\zeta'\tilde{\beta}_i} \\
&= V_{1,i}^{\alpha_1} \Gamma_{1,i}^{\alpha_2} Z_1^{-\tilde{\beta}_i\zeta'} \\
C_{2,i} &= (g^{\alpha\upsilon_{2,i}})^{\gamma} (g^{\eta})^{\mu_{2,i}} (g^{\beta(\gamma+\eta)})^{\delta_2\vartheta_i} g^{-\delta_2\zeta'\tilde{\beta}_i} \\
&= V_{2,i}^{\alpha_1} \Gamma_{2,i}^{\alpha_2} Z_2^{-\tilde{\beta}_i\zeta'} \\
c_i &= \zeta'\tilde{\beta}_i
\end{aligned}$$

Finally, $\mathscr{B}$ sends the ciphertext CT to $\mathscr{A}$.

Phase 2: *Phase 1* is repeated.

Guess: The adversary $\mathscr{A}$ outputs a guess bit $\theta' \in \{1,2\}$ and wins the game if $\theta' = \theta$. Therefore, if $\mathscr{A}$ can distinguish between these two games, $\mathscr{B}$ can solve the DLIN problem.

Theorem 3: If $|Adv_{\mathscr{A}}^{Game_C} - Adv_{\mathscr{A}}^{Game_D}| \leq \epsilon$, where ϵ is negligible, then the DLIN assumption holds.

Theorem 4: If $|Adv_{\mathscr{A}}^{Game_D} - Adv_{\mathscr{A}}^{Game_E}| \leq \epsilon$, where ϵ is negligible, then the DBDH assumption holds.

Theorem 5: Suppose the outsourcing scheme [37] is selectively RCCA-secure. Then the SH-LFGS outsourcing scheme is selectively RCCA-secure.

Proof: The private key in scheme [37] is $K_{\phi,3} = g^{r_\phi}$ and $K_{\phi,4} = (w^{\alpha_\phi} f)^{r_\phi} b^{-r}$, where r_ϕ, α_ϕ, r are selected randomly. And in our scheme, the part private key is $SK_{1,i} = g^{-2\delta_2\eta_i} g^{d(t\sigma_i+r\phi_i)\upsilon_{2,i}}$, $SK_4 = \prod_{i=1}^{n} g^{-2\eta_i}$, where δ_2, η_i, d, t, σ_i, r, ϕ_i, $\upsilon_{2,i}$ are selected randomly. Therefore, we can assume that $r_\phi' = -2\eta_i$, which means that $g^{r_\phi'} = g^{-2\eta_i}$ and $K_{\phi,3}$ are structurally similar. SK_4 is multiplied by $g^{-2\eta_i}$, so that SK_4 is similar to $K_{\phi,3}$ in structure. Now, let us think about $SK_{1,i}$ and $K_{\phi,4}$. Because $r_\phi' = -2\eta_i$, that means $\delta_2 r_\phi' = -2\delta_2\eta_i$ is true. It also means that $g^{-2\delta_2\eta_i}$ is similar to f^{r_ϕ} in structure. Because α_ϕ , r_ϕ in $K_{\phi,4}$ and σ_i, $\upsilon_{2,i}$, ϕ_i in $SK_{1,i}$ are all randomly selected. Therefore $\sigma_i \cdot \upsilon_{2,i}$, $\phi_i \cdot \upsilon_{2,i}$, and $\alpha_\phi \cdot r_\phi$ are similar in structure, and $g^{d(t\sigma_i+r\phi_i)\upsilon_{2,i}}$ is similar to $w^{\alpha_\phi r_\phi}$ in structure. Because b^{-r} is just a random number, the structural similarity between $SK_{1,i}$ and $K_{\phi,4}$ is not affected. In addition, because $SK_3 = \xi \cdot \prod_{i=1}^{n} SK_{1,i}^{-\mu_{1,i}} SK_{2,i}^{-\mu_{2,i}}$, where $\mu_{1,i}$, $\mu_{2,i}$ are random numbers, and ξ is the master private key, the security of SK_3 depends on $SK_{1,i}$ and $SK_{2,i}$. Therefore, if the outsourcing scheme in literature [37] is secure, then our outsourcing scheme is also secure.

VI. PERFORMANCE ANALYSIS

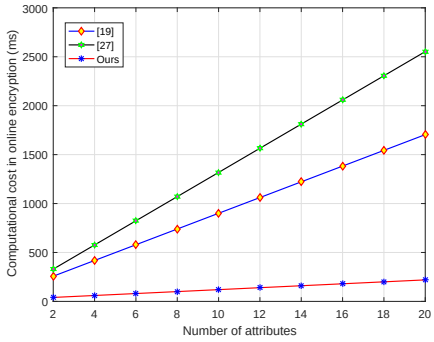

Fig. 3. Time cost in online encryption

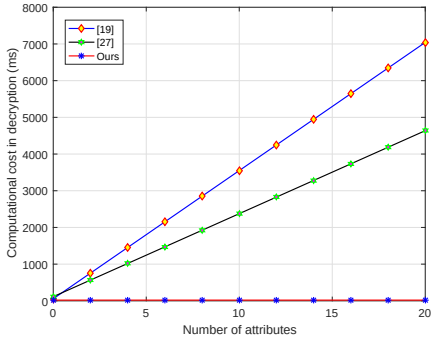

Fig. 4. Time cost in decryption

We compared our scheme with [34], [27], [19], and [31] in terms of security features and performance. Table II provides a comparison of key features such as access policies,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3073740, IEEE Transactions on Services Computing

TABLE II
SECURITY PROPERTIES

Schemes	Access Policy	Group Order	Fully Hidden Policy	Decrypt Test	Online/Ofline Enc	Outsourced Dec
[34]	LSSS	q	×	×	×	✓
[27]	AND	q	✓	×	×	×
[19]	LSSS	q	×	×	×	×
[31]	AND	$q_1q_2q_3$	✓	×	×	×
Ours	AND	q	✓	✓	✓	✓

TABLE III
DETAILED COMPARISONS

Schemes	PK Size	SK Size	CT Size	Decryption overhead
[34]	$(11+2m)\|G\|$	$(5n+2)\|G\|$	$(3+6l)\|G\|$	3p+3e
[27]	$(8n+8)\|G\|+\|G_T\|$	$(4n+2)\|G\|$	$(4n+2)\|G\|+\|G_T\|$	$4np+2p$
[19]	$9\|G\|+\|G_T\|$	$(5n+2)\|G\|$	$(6l+2)\|G\|+\|G_T\|$	$6\|I\|p+p$
[31]	$(8n+2)\|G\|+\|G_T\|$	$(4n+2)\|G\|$	$(4n+2)\|G\|+\|G_T\|$	$4np+2p$
Ours	$(6n+4)\|G\|+\|G_T\|$	$(3n+2)\|G\|$	$(3n+3)\|G\|+\|G_T\|$	$2e$

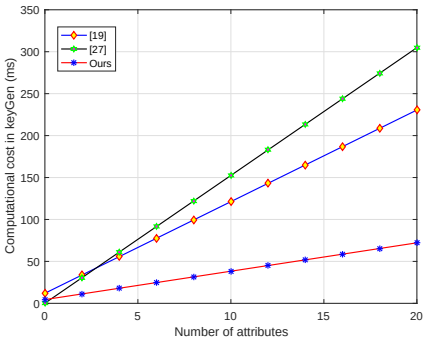

Fig. 5. Time cost in keyGen

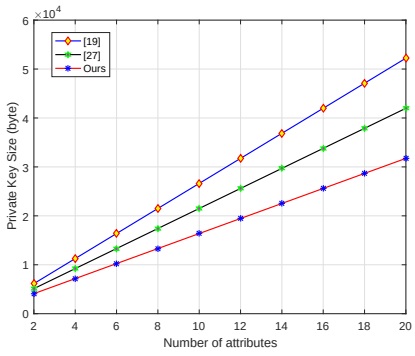

Fig. 6. Storage cost of the secret key

group order, fully hidden access policy, decryption test, and online/offline encryption. It can be seen from Table II that schemes [34] and [19] support LSSS access strategy and are more flexible, but these two schemes cannot achieve a fully hidden access policy, while [27] [31] and our scheme support only AND gates but can achieve a fully hidden access policy. Furthermore, all schemes are based on prime-order groups except [31]. Our scheme and [34] support outsourced decryption, but only our scheme supports decryption test and online/offline encryption.

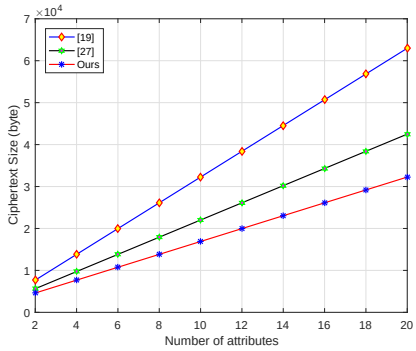

Fig. 7. Storage cost of the ciphertext

TABLE IV
COMPUTATION TIME (MS)

	PC	Smartphone
Bilinear Pairing	8.538	56.554
Exponentiation on group G	10.885	33.217
Exponentiation on group G_T	1.077	9.965
Multiplication on G	0.154	0.326
Multiplication on Z_q	0.001	0.011

Table III provides a detailed comparison of public key size, secret key size, decryption overhead, and ciphertext size in all the schemes. n is the amount of attributes, l represents the max count of rows of the LSSS access matrix, and m represents the maximum quantity of users. We use p and e to denote bilinear and exponentiation operation in G_T, respectively. $|G|$ and $|G_T|$ represent the bit length of the elements belonging to G and G_T. The size of an element in G, G_T is set to 512 bits. I and $|I|$ represent the minimum user set and its size respectively. Although the cost of encryption calculation is high, the online encryption cost of our scheme is only $(2n+1)|G|$. In addition, Table III shows that the size of public key, ciphertext, secret key, and decryption overhead in our scheme is smaller than in the previous scheme.

We used the PBC library based on Java pairing to conduct

TABLE V
COMPUTATION TIME (MS)($|S|$=100)

Phase	[19]	[27]	Ours
KeyGen	1104.501	1523.454	343.424
Online Encrypt	8123.94	12432.063	2013.582
Decrypt	34985.456	22734.708	19.93

simulation experiments on first a PC and then a smartphone to evaluate the performance. The configuration of the PC for the simulation experiment consisted of an Intel Core i7-10710U CPU @1.10 GHz (with an overclock speed of 1.61 GHz), with 16 GB RAM, running the Windows 10 64-bit operating system. The smartphone was an Android 8.0.0 system, with 8-core CPU processor and 6 GB memory. In addition, because an A-type elliptic curve is capable of generating bilinear pairing with minimum time consumption, this curve was selected for our simulation experiment. Due to the PC's computational performance, we used it to simulate cloud service provider SH-CSP and trusted center SH-TA, while the less powerful smartphone was used to simulate PHR owner SH-DO and data user SH-DU. Therefore, the computing for cloud service providers and trusted centers was tested on the PC, and the computing for PHR owners and data users was performed on the smartphone. As shown in Table IV, the experimental data show that compared with the PC, the smartphone took seven times longer for bilinear pairing operation, three times longer on the exponentiation operation on group G, and nine times longer on the exponentiation operation on group G_T. The experimental results in Table V clearly show that our algorithm always had the least computation time of all the algorithms. The comparisons of the computational cost of encryption, decryption, and keyGen are shown in Figs. 3-5. In Fig.3, we consider only the online encryption time cost, while in Fig.4, our decryption time cost is the time cost after outsourcing. Fig. 6 shows the size of the secret key, and Fig. 7 shows the size of the ciphertext. Figs. 3-7 show that the performance of our scheme is significantly better than the schemes [27], [19] in terms of encryption time, decryption time, secret key generation time, secret key size, and ciphertext size.

VII. CONCLUSION

A smart health system can improve the service efficiency of medical institutions and improve the doctor-patient relationship. We propose a scheme that can fully hide the access policy, one which is suitable for lightweight devices whose storage and computational performance capability are relatively limited. The proposed scheme supports online/offline encryption and decryption test with only one bilinear pairing operation. It also supports outsourcing verification decryption, which greatly improves decryption efficiency and which can secure the decryption users' data on the cloud server. Lastly, we proved the chosen plaintext security based on DBDH and DLIN assumptions. The evaluation of the proposed scheme's security and performance shows that it is more applicable to lightweight devices in a smart health system than other schemes.

ACKNOWLEDGMENT

We thank LetPub (www.letpub.com) for its linguistic assistance during the preparation of this manuscript. This work was part supported by the National Cryptography Development Fund under grant(MMJJ20180209), Xi'an Science and Technology Plan Project(NO. 2020KJRC0109), Key Foundation of National Natural Science Foundation of China under grant(NO.U19B2021).

REFERENCES

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. 24th Annu. Int. Conf. Theory Appl. Cryptographic Tech.*, 2005, pp. 457-473.
[2] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Commun. Security*, 2006, pp. 89-98.
[3] R. Ostrovsky, A. Sahai and B. Waters. "Attribute-based encryption with non-monotonic access structures," in *Proc. 14th ACM Conf. Commun. Security*, 2007, pp. 195-203.
[4] J. Lai, R. H. Deng, Y. Li and J. Weng, "Fully secure key-policy attribute-based encryption with constant-size ciphertexts and fast decryption," 2014.
[5] J. Bethencourt, A. Sahai and B. Waters, " Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Security Privacy*, May 2007, pp. 321-334.
[6] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. 14th ACM Conf. Commun. Security*, 2007, pp. 456-465.
[7] B. Waters, "Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions," in *Proc. 29th Annu. Int. Cryptology Conf. Advances Cryptology*, Aug. 2009, pp. 619-636.
[8] B. Waters, "Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization," in *Proc. 14th Int. Conf. Practice Theory Public Key Cryptography Conf. Public Key Cryptography*, Mar. 2011, pp. 53-70.
[9] A. Lewko, T. Okamoto, A. Sahai, K. Takashima and B.Waters, "Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption," *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010*. Proceedings Springer-Verlag, 2010.
[10] K. Emura, A. Miyaji, A. Nomura, K. Omote and M. Soshi, "A ciphertext policy attribute-based encryption scheme with constant ciphertext length," in *Proc. 5th Int. Conf. Inf. Security Practice Experience*, Apr. 2009, pp. 13-23.
[11] S. Wang, J. Zhou, J.K. Liu, J. Yu, J. Chen and W. Xie, An efficient file hierarchy attribute-based encryption scheme in cloud computing, *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 6, pp. 1256-1277, Jun. 2016.
[12] J. Li, J. W. Li, X.F. Chen, C. Jia and W. Lou, "Identity-based Encryption with Outsourced Revocation in Cloud Computing," *IEEE TRANSACTIONS ON COMPUTERS*, vol. 64, no. 2, 2015.
[13] T. Nishide, K. Yoneyama and K. Ohta, "Attribute-Based Encryption with Partially Hidden Encryptor-Specified Access Structures," *International Conference on Applied Cryptography and Network Security*. Springer, Berlin, Heidelberg, 2008.
[14] Y. S. Rao and R. Dutta, "Recipient Anonymous Ciphertext-Policy Attribute Based Encryption," *International Conference on Information Systems Security* Springer Berlin Herdelberg, 2013.
[15] L. Zhang, Q. Wu, Y. Mu and J.Zhang, " Privacy-Preserving and Secure Sharing of PHR in the Cloud," *J. Med. Syst.*,vol. 40, pp. 1-13, 2016.
[16] Y. Zhang, X. Chen, J. Li, D. Wong and H. Li, "Anonymous attribute-based encryption supporting efficient decryption test," in *Proc. 8th ACM Symp. Inf. Comput. Commun. Secur.*, May 2013, pp. 511-516.
[17] Y. Zhang, J. Li and H. Yan, "Constant Size Ciphertext Distributed CP-ABE Scheme With Privacy Protection and Fully Hiding Access Structure," in *IEEE Access*, vol. 7, pp. 47982-47990, 2019.
[18] J. Lai, X. Zhou, R. H. Deng, Y. Li and K. Chen, "Expressive CP-ABE with partially hidden access structures," in *Proc. 7th ACM Symp. Inf. Comput. Commun. Secur.*, May 2012, pp. 18-19.
[19] H. Cui, R. H. Deng, G. Wu and J. Lai, "An Efficient and Expressive Ciphertext-Policy Attribute-Based Encryption Scheme with Partially Hidden Access Structures," in *Proc. 10th Int. Conf. Prov. Secur.*, Nov. 2016, pp. 19-38.

[20] H. Cui, R. H. Deng, J. Lai, X. Yi and S. Nepal,“ An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures, revisited,” *Computer Networks*, 2018, 133: 157-165.

[21] Y. Zhang, Z. Dong and R. H. Deng, “Security and Privacy in Smart Health: Efficient Policy-Hiding Attribute-Based Access Control,” *IEEE Int. Things J.*, vol. 5, no. 3, pp. 2130-2145, Jun. 2018.

[22] Q. Han, Y. Zhang and H. Li, “Efficient and robust attribute-based encryption supporting access policy hiding in Internet of Things,” *Future Generation Comput. Syst.*, vol. 83, pp. 269-277, 2018.

[23] Y. Liu, Y. Zhang, J. Ling and Zh. Liu, “Secure and fine-grained access control on e-healthcare records in mobile cloud computing,” *Future Generation Comput. Syst.*, vol. 78, pp. 1020-1026,2018.

[24] Y. Zhang and Z. Dong, “Efficient and Expressive Anonymous Attribute-Based Encryption for Mobile Cloud Computing,” *International Conference on Broadband & Wireless Computing*. Springer, Cham, 2016.

[25] M. Green, S. Hohenberger and B. Waters, “Outsourcing the decryption of ABE ciphertexts,” in *Proc. 20th USENIX Conf. Security*, 2011.

[26] J. Lai, R. H. Deng, C. Guan and J. Weng, “ Attribute-Based Encryption With Verifiable Outsourced Decryption,” *IEEE Trans. Inf. Forensics Secur.*, 2013, 8(8):1343-1354.

[27] T. V. X. Phuong, G. Yang and W. Susilo, “Hidden Ciphertext Policy Attribute-Based Encryption Under Standard Assumptions,” *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 1, pp. 35-45, Jan. 2015.

[28] J. Katz, A. Sahai and B. Waters, “Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products”, *Journal of Cryptology*, 2013, 26(2):191-224.

[29] E. Shi and B. Waters, “Delegating Capabilities in Predicate Encryption Systems,” Icalp, 2008.

[30] S. Sedghi, P. V. Liesdonk, S. Nikova, P. Hartel and W.Jonker, “Searching Keywords with Wildcards on Encrypted Data,” *International Conference on Security and Cryptography for Networks*, Springer, Berlin, Heidelberg, 2010.

[31] C. Jin, X. Feng and Q. Shen, “Fully Secure Hidden Ciphertext Policy Attribute-Based Encryption with Short Ciphertext Size,” *In Proceedings of the 6th International Conference on Communication and Network Security (ICCNS ’16). ACM, New York, NY, USA*, 91-98. 2016.

[32] P. Chaudhari, M. L. Das and A. Mathuria, “On Anonymous Attribute Based Encryption,” *Information Systems Security*, Springer International Publishing, 2015.

[33] Z. Wang and M. He, “CP-ABE with Hidden Policy from Waters Efficient Construction,” *International Journal of Distributed Sensor Networks*, 2015, 2016(5).

[34] X. Hu, Y. Zhao, P. Li, H. Zhang and K. H. Yeh, “Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing,” *Future Generation Comput. Syst.*, vol. 97, pp. 453-461, 2019.

[35] Y. Rao, “A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing,” *Future Generation Comput. Syst.*, vol. 67, pp. 133-151, Feb. 2017.

[36] J. Li, W. Yao, Y. Zhang, H. Qian and J. Han, “Flexible and fine-grained attribute-based data storage in cloud computing,” *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 785-796, Jan. 2016.

[37] X. Fu, X. Nie, T. Wu and F. Li, “Large universe attribute based access control with efficient decryption in cloud storage system,” *The Journal of Systems and Software*. 2018.

[38] TVX. Phuong, G. Yang and W. Susilo,“Efficient Hidden Vector Encryption with Constant-Size Ciphertext,” 2014.

[39] TVX. Phuong, G. Yang, W. Susilo and X.Chen,“Attribute Based Broadcast Encryption with Short Ciphertext and Decryption Key,” *European Symposium on Research in Computer Security Springer International Publishing*. 2015.

[40] Q. Wang, L. Peng, H. Xiong, J. Sun and Z. Qin, “ Ciphertext-policy attribute-based encryption with delegated equality test in cloud computing,” *IEEE Access*, 2017.

[41] H. Xiong, H. Zhang and J. Sun,“Attribute-Based Privacy-Preserving Data Sharing for Dynamic Groups in Cloud Computing,” *IEEE systems journal* 13.3(2019):2739-2750.

[42] R.-H. Hsu, Y.-H. Hu, G.-W. Lin and B.-C. Ko, “Privacy-preserving Data Sharing with Attribute-based Private Matching Based on Edge Computation in the Internet-of-Things,” *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Auckland, New Zealand*,2020, pp. 1578-1587.

[43] Y. Yang, X. Liu, R.H. Deng and Y. Li, ”Lightweight Sharable and Traceable Secure Mobile Health System,” *in IEEE Transactions on Dependable and Secure Computing,* vol. 17, no. 1, pp. 78-91, 1 Jan.-Feb. 2020, doi: 10.1109/TDSC.2017.2729556.

Leyou Zhang received the M.S. and Ph.D. degrees from Xidian University, in 2002 and 2009, respectively.

He is currently a Professor with Xidian University. His current research interests include cryptography, network security, cloud security, and computer security.

Wenting You received the B. S. degree in mathematics in 2017 from Taiyuan Normal University, China. She is current studying for the M.S. degree in applied mathematics with Xidian University, China.

Her research interests include applied cryptography and network security.

Yi Mu received the Ph.D. degree from the Australian National University in 1994. He was a Professor of computer science with the University of Wollongong, Wollongong.

He is currently a Professor with the Institute of Data Science, City University of Macau. His current research interests include block chain, cybersecurity, access control and cryptography. Prof Mu has served as editor-in-chief of international journals and the National Journal of Applied Cryptography