Received June 14, 2019, accepted July 14, 2019, date of publication July 30, 2019, date of current version August 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2931917

Lightweight and Privacy-Preserving Medical Services Access for Healthcare Cloud

JINGWEI LIU[1], (Member, IEEE), HUIFANG TANG[1], RONG SUN[2], (Member, IEEE), XIAOJIANG DU[3], (Senior Member, IEEE), AND MOHSEN GUIZANI[4], (Fellow, IEEE)

[1]Shaanxi Key Laboratory of Blockchain and Secure Computing, Xidian University, Xi'an 710071, China
[2]State Key Lab of ISN, Xidian University, Xi'an 710071, China
[3]Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA
[4]Department of Computer Science and Engineering, Qatar University, Doha, Qatar

Corresponding author: Jingwei Liu (jwliu@mail.xidian.edu.cn)

This work was supported in part by the Key Industry Innovation Chain Project of Shaanxi under Grant 2019ZDLGY13-05 and Grant 2019ZDLGY12-01, in part by the 111 Project under Grant B08038, and in part by the Collaborative Innovation Center of Information Sensing and Understanding at Xidian University.

ABSTRACT With the popularity of cloud computing technology, the healthcare cloud system is becoming increasingly perfect, which reduces the time of disease diagnosis and brings great convenience to people's lives. But meanwhile, the healthcare cloud system usually involves users' privacy information, and there is still a challenge on how to ensure that the sensitive information of users is not disclosed. Attribute-based signature (ABS) is a very useful technique for the privacy protection of users and is very suitable for anonymous authentication and privacy access control. However, general ABS schemes usually contain heavy computation overhead in signing and verification phases, which is not conducive for resource-limited devices to access healthcare cloud system. To address the above issues, we propose a lightweight and privacy-preserving medical services access scheme based on multi-authority ABS for healthcare cloud, named LPP-MSA. By using online/offline signing and server-aided verification mechanisms, the proposed scheme can greatly reduce the calculation overhead. In addition, LPP-MSA achieves unforgeability and anonymity and can resist collision attack. The comparisons of computational cost and storage overhead between LPP-MSA and the other existing schemes show that LPP-MSA is more efficient in both signing and verification phases. Therefore, it could be well applied to the scenarios where users access the healthcare cloud system for large scale remote medical services via resource-constrained mobile devices.

INDEX TERMS Healthcare cloud, attribute-based signature (ABS), privacy-preserving, online/offline signing, server-aided verification.

I. INTRODUCTION

Recently, the development of cloud computing has brought new breakthroughs in many fields, such as healthcare, transportation, education, finance, and energy. But traditional healthcare system cannot meet the needs of the highly-developed healthcare services for its certain inefficiencies. With the advantage of cloud computing, the patient-centered medical information system can realize the sharing of medical resources [1]. In such a system, healthcare services of different medical institutions are deployed in healthcare cloud, as shown in Fig. 1. Medical Services Requester (MSR) can access the healthcare cloud server through different terminals (such as smart phone, laptop, and so on) and obtain the service resources from Medical Services Provider (MSP) on demand.

The healthcare cloud system can not only improve the utilization of medical resources, but also bring convenience to patients. However, the healthcare system usually involves a large amount of users' privacy information, such as physical condition and medical records. In addition, when users log in the healthcare cloud to access medical services, their personal information is clear to the system. How to protect the identity privacy of users while authenticating their legitimacy is still a challenge. So, it is necessary to adopt reasonable technologies and effective measures to ensure the sustainable and stable development of healthcare cloud services.

Relevant scholars have done a lot of research [2]–[4], among which digital signature can ensure data integrity and non-repudiation in healthcare cloud system. On the basis

The associate editor coordinating the review of this manuscript and approving it for publication was Petros Nicopolitidis.

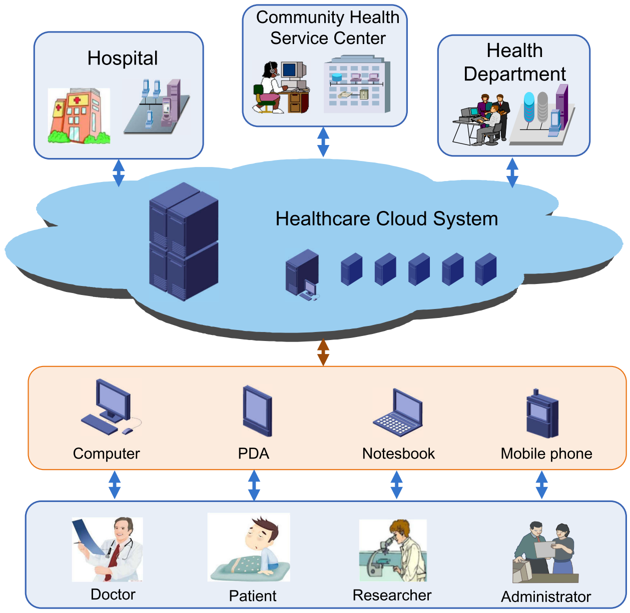


FIGURE 1. The architecture of healthcare cloud system.

of digital signature, Attribute-Based Signature (ABS) can effectively achieve privacy protection and access control for the data stored in cloud. Moreover, it is flexible and convenient for requesters to hide their identities. Therefore, ABS is especially useful for implementing anonymous access to the healthcare cloud system for different kinds of users.

There are two main types of ABS construction: threshold schemes and linear schemes [5]. In threshold schemes, the signature is valid when the number of signer's attributes that satisfy the access structure is no less than the threshold value. This kind of schemes are simple in calculation but not rigorous enough. In linear schemes, the access structure needs to be algebraically transformed, which makes the computation overhead increased. But linear schemes are more rigorous because the signature is valid only when the attributes in signing strictly conform to the access structure. In addition, according to different application scenarios, ABS has many different modification, such as attribute-based ring signature [6], [7], attribute-based group signature [8]–[10] and other schemes [11]–[14].

A. RELATED WORK

ABS started with the concept of attribute in digital signature and was developed from fuzzy identity signature [15]. In 2008, Maji *et al.* [16] proposed the definition of ABS for the first time, in which the identity of the signer was described by a series of attributes, and the attribute authority was responsible for distributing and managing the user's attribute keys. In ABS, the signer can sign the message only when the signer's attributes satisfy the signature policy. If the signature is valid, the verifier can be sure that the signer's attributes satisfy the pre-set signature policy without knowing the specific identity of the signer. After that, Li and Kim [17] and Shahandashti and Safavi-Naini [18] improved the ABS scheme under selective model. However, their schemes only support node predicate [17] and threshold predicate [18] respectively. In 2008, an ABS scheme [19] based on access tree was proposed, in which the access tree could be converted into access matrix and anybody could check the validity of signature by verifying whether the access matrix satisfies the pre-set access structure.

In the above ABS schemes, signer's attributes are managed by only a single authority. But in practical application scenarios, the ABS scheme with one attribute authority has to bear a large burden of management when there are multiple attributes in the system. For example, in the medical field, a user's attribute set is $\{doctor, researcher\}$, the attribute *doctor* is managed by hospital, while the attribute *researcher* is managed by education institution. Obviously, these two attributes come from different attribute authorities and they cannot be generated by an attribute authority. To address this problem, multi-authority ABS schemes have been gradually studied. In [20], a multi-authority scheme was constructed without complete proof. Subsequently, Lin *et al.* [21] raised a multi-authority ABS scheme without central authority, but their scheme involves lots of operations, which reduced the efficiency of the algorithm greatly. In order to make up for the shortcomings of multi-authority ABS in security, signature strategy and efficiency, Cao et al. proposed a multi-authority ABS scheme [22] supporting AND, OR and threshold gates. Recently, many improved multi-authority ABS schemes [23]–[25] emerged.

However, the design of multi-authority also brings too much computing overhead, making the efficiency relatively lower compared with single authority schemes. The online/offline and server-aided mechanisms can effectively reduce the computing overhead of users and greatly improve the efficiency of signing and verification. Even *et al.* [26] first proposed the idea of online/offline signing, but their method increased the signature length so that the practicability of the scheme was greatly reduced. Subsequently, Shamir and Tauman et al. proposed a practical online/offline signing scheme in [27] based on the chameleon hash function, but they could not solve the key exposure problem. In 2007, Chen *et al.* [28] solved the problem of key exposure in general online/offline signing schemes to a certain extent by using the double-trapdoor chameleon hash function. In 2009, Gao *et al.* [29] put forward the concept of divisible online/offline signing. The information obtained in the offline phase needs not to be sent to the verifier after the online phase but be directly sent to the verifier, which thus reduced local storage. In respect of server-aided verification, Wang *et al.* [30] proposed a server-aided verification protocol based on ref. [20], which could also be applied to other ABS schemes. Han *et al.* [31] put forward a privacy-preserving server-aided verification scheme supporting Linear Secret Sharing Scheme (LSSS) matrix, that strengthened the flexibility of the scheme. In 2018, Mo *et al.* [32] put forward a revocable server-aided verification scheme. However, it was based on threshold ABS, which greatly reduced the flexibility. Furthermore, some outsourcing computation schemes [33]–[35] can also reduce the computation cost.

B. OUR CONTRIBUTIONS

In this paper, we come up with a lightweight and privacy-preserving medical services access scheme based on multi-authority ABS for healthcare cloud, named LPP-MSA. It enables MSR to hide their identity without revealing the specific identity information. The main contributions of this paper are concluded as below:

- The online/offline signing mechanism is introduced when MSR requests medical services in healthcare cloud. The pre-computing technique is used in offline signing phase, so that signature can be completed with only a few operations in the online phase, greatly reducing the computation cost of MSR.
- In order to reduce the computation cost of MSP, we use a server to assist verifying the signature. The MSP first converts the signature and then sends the transformed signature to the server. The server completes massive computation, so the MSP's computation cost is very low.
- The proposed LPP-MSA has high computational efficiency in the signing and verification phases, which can be applied to mobile health and telemedicine scenarios. Moreover, LPP-MSA satisfies unforgeability, anonymity and can resist collision attack through our security analysis.

C. ORGANIZATION

The structure of this paper is organized as follows. Section II introduces some preliminaries in LPP-MSA. Next, Section III describes the proposed LPP-MSA scheme in detail. Then, Section IV analyzes the security and performance. Finally, Section V gives a comprehensive conclusion.

II. PRELIMINARIES

We introduce some preliminaries in LPP-MSA firstly, then define the framework and security model of LPP-MSA.

A. BILINEAR MAPS

Let G_1 and G_2 be cyclic groups with prime order p. $e : G_1 \times G_1 \rightarrow G_2$ is a bilinear map if it satisfies the following properties:

- Bilinearity: For any $a, b \in Z_p$ and $X, Y \in G_1$, $e\left(X^a, Y^b\right) = e(X, Y)^{ab}$;
- Non-degeneracy: $\exists\ X, Y \in G_1$, such that $e\left(X, Y\right) \neq 1_{G_2}$. Here, 1_{G_2} is the identity element of the group G_2;
- Computability: There is an effective algorithm for calculating the value of $e\left(X, Y\right)$ for any $X, Y \in G_1$.

Definition 1: Computational Diffie-Hellman (CDH) Problem: Let G_1 be a multiplicative cyclic group with prime order p, g be the generation of G_1, and Z_p^* be a finite field. CDH problem is that given a triple (g, g^a, g^b), it is hard to compute g^{ab} for unknown $a, b \in Z_p^*$.

B. LINEAR SECRET SHARING SCHEME (LSSS)

A LSSS meets the following conditions:

- The secret fragment owned by each participant is a vector on Z_p^n, and the sharing of all participants constitutes a complete secret value s;
- We randomly select $v_2, v_3, \cdots, v_n \in Z_p^n$, construct vector $v = (s, v_2, v_3, \cdots, v_n)^T$, and define a map $\rho : \{1, 2, \cdots, l\} \rightarrow \mathbb{M}v$. Here, $\mathbb{M}$ is an $l \times n$ matrix and ρ maps the j-th row of the matrix to the secret fragment obtained by the participant. $\mathbb{M}v^T$ is a vector related to s to a certain extent, which can be expressed in formula $\lambda_i = \left(\mathbb{M}v^T\right)_i$.

Now, we assume that there is a LSSS with access structure Ψ, and define $I = \{i : \rho(i) \in S\} \subseteq \{1, 2, \cdots, l\}$. For an authorized set $S \in \Psi$, there exists a vector $w = \left\{w_i \in Z_p\right\}_{i\in I}$ such that $\sum_{i\in I} w_i\mathbb{M}_i = (1, 0, \cdots, 0)$ holds. Thus, we can get the equation $\sum_{i\in I} w_i\mathbb{M}_i v^T = \sum_{i\in I} w_i\lambda_i = \sum_{i\in I} (w_i\mathbb{M}_i)\, v^T = s$. For an unauthorized set, there must be a vector w such that $w(1, 0, \cdots, 0)^T = -1$ and $w\mathbb{M}_i^T = 0, i \in I$ hold.

C. DEFINITION OF MULTI-AUTHORITY ABS

Definition 2: Multi-Authority ABS: The multi-authority ABS scheme contains five algorithms.

1) GlobalSetup$(\lambda) \rightarrow GP$: This algorithm is executed by a trust authority. It inputs security parameter λ and generates the public parameters GP;

2) AuthoritySetup$(GP, \theta) \rightarrow \{PK_\theta, SK_\theta\}$: This algorithm is executed by the attribute authority. Each attribute authority generates a public/private key pair $\{PK_\theta, SK_\theta\}$;

3) AttrGen$(GP, \theta, SK_\theta, GID, i) \rightarrow USK_{GID,(\theta,i)}$: This algorithm is executed by the attribute authority. It inputs the public parameters GP, the attribute authority θ, the signer's identity GID, the private key SK_θ of the attribute authority θ, an attribute i and outputs the attribute private key $USK_{GID,(\theta,i)}$ corresponding to the attribute i;

4) Sig$(GP, \left\{PK_\theta, USK_{GID,(\theta,i)}\right\}, m, S) \rightarrow \sigma$: This algorithm is executed by the signer. It inputs the public parameters GP, the public key PK_θ of the attribute authority θ, the private key $USK_{GID,(\theta,i)}$, the signature message m, the signature access structure S and outputs the final signature σ;

5) Ver$(GP, \{PK_\theta\}, m, S, \sigma) \rightarrow$ "1" *or* "0": This algorithm is executed by the verifier. It inputs the public parameters GP, the public key $\{PK_\theta\}$, the signature message m, the signature access structure S, the signature σ and outputs "1" for true or "0" for false.

D. FORMAL DEFINITION OF LPP-MSA

Our LPP-MSA scheme consists of eight algorithms:

1) GlobalSetup$(\lambda) \rightarrow GP$: This algorithm inputs security parameters λ and outputs public parameters GP.

2) AuthoritySetup$(GP, i) \rightarrow \{(PK_i, SK_i)\}$: This algorithm is run by the attribute authority. It inputs the public parameters GP, the attribute i and outputs the public/private key pairs $\{PK_i, SK_i\}$.

3) KeyGen$(GP, GID, \phi) \rightarrow \left\{SK_{i,GID}\right\}$: This algorithm is executed by the attribute authority. It inputs the public parameters GP, the MSR's identity GID, the MSR's attribute set ϕ and outputs the signing key $\left\{SK_{i,GID}\right\}$ corresponding to each attribute i.

4) Offline.Sign$(GP, \left\{SK_{i,GID}\right\}) \rightarrow IS$: This algorithm is executed by the MSR. It inputs the public parameters GP, the signing key $\left\{SK_{i,GID}\right\}$ and outputs the intermediate signature IS.

5) Online.Sign$(IS, M, (\mathbb{A}, \rho), \left\{SK_{i,GID}\right\}) \rightarrow \sigma$: This algorithm is executed by the MSR. It inputs the intermediate signature IS, the message M, the signature policy $(\mathbb{A}, \rho)$, the signature key $\left\{SK_{i,GID}\right\}$ and outputs the signature σ.

6) Transform$(\sigma) \rightarrow (\sigma', \tau)$: This algorithm is executed by the MSP. It inputs the signature σ and outputs the transformed signature σ' and the transformation key τ.

7) CS.Verify$(\sigma', GP) \rightarrow V$: This algorithm is executed by the cloud server. It inputs the transformed signature σ', the public parameters GP and outputs the intermediate verification signature V.

8) MSP.Verify$(GP, M, V, \tau) \rightarrow true/false$: This algorithm is executed by the MSP. It inputs the public parameters GP, the message M, the intermediate verification signature V, the transformation key τ and outputs *true* which represents the signature is valid and *false* represents invalid.

Definition 3: Correctness: We say the LPP-MSA scheme is correct, which means that for any GID with attribute set ϕ satisfying the signature policy $(\mathbb{A}, \rho)$, any message m, if $GP \leftarrow GlobalSetup(\lambda)$, $\{(PK_i, SK_i)\} \leftarrow AuthoritySetup(GP, i)$, $\left\{SK_{i,GID}\right\} \leftarrow KeyGen(GP, GID, \phi)$, $IS \leftarrow Offline.Sign(GP, \left\{SK_{i,GID}\right\})$, $\sigma \leftarrow Online.Sign(IS, M, (A, \rho), \left\{SK_{i,GID}\right\})$, $(\sigma', \tau) \leftarrow Transform(\sigma)$, $V \leftarrow CS.Verify(\sigma', GP)$, then $MSP.\ Verify\ (GP, M, V, \tau) \rightarrow true$.

E. SECURITY MODEL

Definition 4: Unforgeability: The LPP-MSA needs to be satisfied with unforgeability. Here, we describe the security model in detail through the following game.

1) QUERY PHASES

An adversary $\mathcal{A}$ can make the following queries to a challenger $\mathcal{C}$.

- **Signing-key queries**. $\mathcal{A}$ chooses an identity GID firstly, whose attribute set is ϕ. Then, it asks the challenge $\mathcal{C}$ for the attribute signature key corresponding to the identity GID. If the key exists in a list L maintained by cloud server, the challenger $\mathcal{C}$ returns the corresponding signing key $\left\{SK_{i,GID}\right\}$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ runs the algorithm $\left\{SK_{i,GID}\right\} \leftarrow KeyGen(GP, GID, \phi)$ and returns the generated signing key $\left\{SK_{i,GID}\right\}$ to $\mathcal{A}$. Finally, $\mathcal{C}$ will add $\left\{SK_{i,GID}\right\}$ to the list L.
- **Signature queries**. The adversary $\mathcal{A}$ first selects a message M and an access structure $\mathbb{A}$. Then, it makes some queries about the signature to challenge $\mathcal{C}$. $\mathcal{C}$ runs algorithms $IS \leftarrow Offline.Sign(GP, \left\{SK_{i,GID}\right\})$ and $\sigma \leftarrow Online.Sign(IS, M, (A, \rho), \left\{SK_{i,GID}\right\})$ to generate a valid signature σ. Finally, $\mathcal{C}$ returns the signature to $\mathcal{A}$.
- **Transformed signature queries**. For each signature query about (M, σ) by $\mathcal{A}$, $\mathcal{C}$ runs algorithms $(\sigma', \tau) \leftarrow Transform(\sigma)$ and $V \leftarrow CS.Verify(\sigma', GP)$. Then, $\mathcal{C}$ sends the intermediate signature V to adversary $\mathcal{A}$.

2) FORGERY

The adversary $\mathcal{A}$ outputs a message M^*, a signature V^*, and an access structure $\mathbb{A}^*$. $\mathcal{A}$ could win this game if the following conditions hold:

- The adversary did not make any signature queries about $(\mathbb{A}^*, V^*)$ during the *Online.Sign* phase;
- For the challenge access structures $\mathbb{A}^*$, $\phi \subseteq \mathbb{A}^*$;
- $MSP.Verify(GP, M^*, V^*) = 1$.

The winning advantage of $\mathcal{A}$ is defined as: $Adv_{\mathcal{A}}^{UNF}(\lambda) = \Pr[A \quad wins]$. If the probability of $\mathcal{A}$ winning this game is negligible in polynomial time, the LPP-MSA satisfies unforgeability.

III. THE LIGHTWEIGHT AND PRIVACY-PRESERVING MEDICAL SERVICES ACCESS SCHEME FOR HEALTHCARE CLOUD

In this section, we propose the LPP-MSA considering the design objectives and the application scenarios.

A. DESIGN OBJECTIVES

In order to reduce computation overhead of MSR and realize privacy protection, this paper proposes a lightweight and privacy-preserving medical services access scheme based on multi-authority ABS for healthcare cloud. The design objectives of the scheme are as follows:

1) Unforgeability. The proposed scheme should ensure the uniqueness of MSR' identity. And an attacker cannot forge a legitimate signature without knowing the MSR' private keys;

2) Anonymity. The MSR in healthcare cloud system can access medical services anonymously. The scheme should realize the privacy protection of MSR's identity. MSP only knows the legality of MSR, but s/he cannot know the MSR's specific identity;

3) Collusion Resistance. Multiple illegitimate MSRs cannot sign individually, and even if they combine their key information together, they cannot sign legally;

4) Lightweight. In order to ensure that users in the medical cloud system can normally request or provide services while using resource-limited terminals, the design should be lightweight. The signing and verification phases should minimize the computation overhead of users, which can be suit to large scale remote medical services access for users via different kinds of mobile devices.

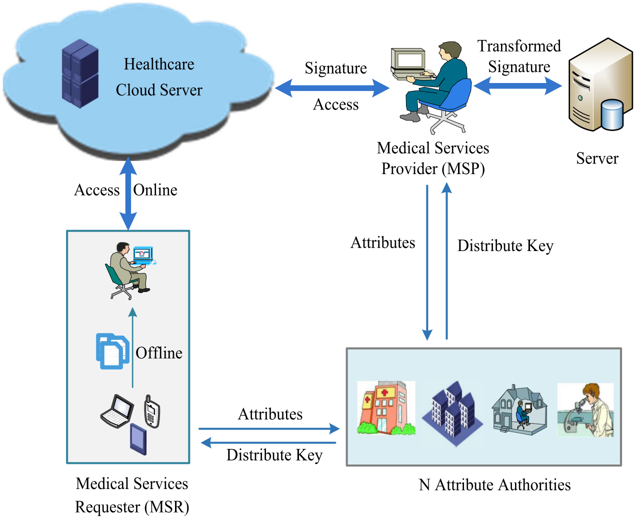


FIGURE 2. System model.

B. SYSTEM MODEL

The proposed LPP-MSA consists of four participants: MSR, MSP, Attribute Authorities, and Healthcare Cloud Server, as shown in Fig. 2. The responsibilities of each participant are as follows:

1) MSR. The MSR uploads his/her medical data to the healthcare cloud server to request medical services;

2) MSP. The MSP can not only check the validity of MSR's data, but also verify the legitimacy of the MSR's identity. Besides, MSP is responsible for providing medical services to MSR's requests;

3) Attribute Authorities. The attribute authorities are responsible for managing MSR' attributes and issuing attribute private keys to MSR according to their attributes;

4) Healthcare Cloud Server. The healthcare cloud server is responsible for storing data of MSR in the cloud. Moreover, it allows legitimate MSR in the system to access or upload data.

In different application scenarios, the specific roles of MSR and MSP are different. For example, in telemedicine scenario, the MSR might be a patient and the MSP might be a doctor. In a physical examination center or health institution, the MSR can be a health consultant, and the MSP can be a health specialist. For better understand, we analyze a specific scenario below.

C. APPLICATION TO HEALTHCARE CLOUD SYSTEM

With the development of healthcare cloud, mobile health and telemedicine are gradually applied to people's daily life, bringing great convenience to people. But at the same time, the information in the healthcare cloud also has extremely high commercial value (such as electronic health records). So, how to ensure the security of mobile health and telemedicine applications becomes a very serious challenge. In the telemedicine system, users can obtain their physical examination data through various ways, such as hospital self-examination or home medical equipments. Users want their medical data to be diagnosed and analyzed by a professional organization, but they do not want doctors to know their specific identity information, such as name and ID number, and they just want to enjoy the services of the telemedicine system anonymously. Therefore, how to make doctors diagnose users' medical data without exposing users' identity information is an urgent problem in telemedicine.

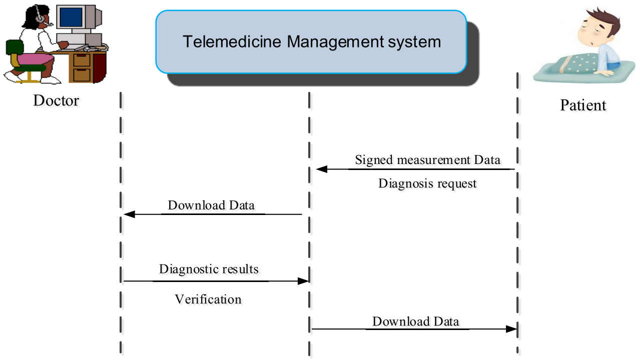


FIGURE 3. The flowchart of telediagnosis.

The proposed LPP-MSA can be applied to the scenario where patient anonymously requests diagnostic services in telemedicine. In this scenario, patient is the MSR and doctor is the MSP. The process of telediagnosis in telemedicine system is shown in Fig. 3. As the transfer station of medical data and diagnosis results between patient and doctor, the telemedicine management system is mainly responsible for processing medical data by its huge storage space and powerful computing power. The doctor is responsible for making diagnosis based on the patient's medical data. First, the patient initiates a medical request in the telemedicine system to get the doctor's telemedicine diagnosis. To further protect patient's privacy, the patient first obtains the private keys corresponding to the attribute set of his/her identity from the attribute authorities, then signs the medical data, and finally sends the signed medical data to the telemedicine system through the secure channel. After receiving the patient's medical request, the doctor downloads the patient's medical data through the telemedicine system. In addition, when the doctor needs to combine the patient's medical history to judge the result during the diagnosis process, the doctor can also ask the patient about the medical history by using encrypted identity. If the doctor's identity is legal, s/he can obtain the medical history from the patient. In the process of diagnosis and queries, the doctor does not know the real identity of the patient, but only knows that the medical data is from a legitimate user in the telemedicine system, and the data is credible.

D. OUR SCHEME

The proposed LPP-MSA contains eight algorithms: Global Setup, Authority Setup, KeyGen, Offline.Sign, Online.Sign, Transform, CS.Verify, and MSP.Verify. The notations used in

TABLE 1. Notations.

Notations	Description	Notations	Description
G_T and G	Multiplicative cyclic groups	p	The prime order of group G and G_T
g	The generator of group G and G_T	e	A bilinear map
U	The MSP's attribute domain	$H(\cdot)$	Hash function $H(\cdot)$: $Z_p^* \rightarrow G$
$F(\cdot)$	Hash function $F(\cdot)$: $U \rightarrow G$	$\mathcal{C}$	A challenger
$\mathcal{A}$	An adversary	GID	MSR's global identity
λ	Security parameter	N	The number of attribute authorities
L	The key list	i	MSP's attribute
MSR	Medical Services Requester	MSP	Medical Services Provider

LPP-MSA is shown in Table 1 and the specific structure is as follows:

1) GlobalSetup: Set a security parameter λ firstly, and give a bilinear group G with order p and generator g. $e : G \times G \rightarrow G_T$ is a bilinear map, where G and G_T are both multiplicative cyclic groups. Next, select two hash functions $H : Z_p^* \rightarrow G$ and $F : U \rightarrow G$ which map the MSR's identity GID and attributes to the elements of G respectively.

Finally, the algorithm takes the public parameters as output:

$$GP = \{p, g, G, H, F\}.$$

2) AuthoritySetup: Suppose that there are N attribute authorities, and each attribute authority manages a series of attributes $\{i_1, i_2 \cdots i_n\}$. According to GP, for each attribute i managed by the attribute authority θ_j, θ_j first selects $\alpha_i, y_i \in Z_p^*$ randomly, then generates the public key:

$$PK_i = \left\{e(g,g)^{\alpha_i}, g^{y_i}\right\},$$

and the private key:

$$SK_i = \{\alpha_i, y_i\}.$$

3) KeyGen: This algorithm inputs GP and the MSR's attribute set ϕ. For each attribute $i \in \phi$ owned by the MSR, if it is managed by the attribute authority θ_j, the algorithm is run to generate signature key:

$$SK_{i,GID} = \left\{g^{\alpha_i}, H(GID)^{y_i}\right\}.$$

4) Offline.Sign: This algorithm inputs (GP, $\{SK_{i,GID}\}$) and runs as follows. For all attributes $i \in \phi$, it randomly selects $r_i, \lambda_i', w_i' \in Z_p^*$ and computes

$$\begin{aligned} Sig_{1,i} &= H(GID)^{r_i}, \\ \sigma_0 &= \frac{e(g,g)^{\lambda_i'}}{e(g^{\alpha_i}, g)}, \\ \sigma_1 &= e(H(GID), g^{w_i'}), \\ \sigma_2 &= e(H(GID)^{y_i}, g^{r_i}). \end{aligned}$$

5) Online.Sign: This algorithm inputs the intermediate signature IS, a message M and a signature policy $(\mathbb{A}, \rho)$, in which $\mathbb{A}$ is a matrix of l rows and n columns, and ρ is a map from the row of the matrix $\mathbb{A}$ to the policy attribute. The MSR does follows to generate the final signature.

- Choose $s, y_2, \cdots, y_n, z_2, \cdots, z_n \in Z_p^*$ randomly and build vectors $v = (s, y_2, \cdots, y_n)^T$, $w = (0, z_2, \cdots, z_n)^T$;
- Compute $\lambda_i = A_i \cdot v$, $w_i = A_i \cdot w$ for each $i \in \{1, 2 \cdots, l\}$. Note that A_i represents the i-th row of the matrix $\mathbb{A}$;
- Choose a hash function h and compute

$$\begin{aligned} Sig_0 &= h(M) \cdot e(g,g)^s, \\ \sigma_3 &= \lambda_i - \lambda_i', \quad \sigma_4 = w_i - w_i', \\ \sigma_5 &= \alpha_{\rho(A_i)} - \alpha_i, \quad \sigma_6 = (y_{\rho(A_i)} - y_i) r_i. \end{aligned}$$

Finally, the algorithm outputs the signature:

$$\sigma = \left\{Sig_0, Sig_{1,i}, \sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6\right\}.$$

6) Transform: In this phase, after receiving the signature σ, the MSP chooses $\tau \in Z_p^*$ randomly as the transformation key and computes

$$\begin{aligned} Sig_0' &= Sig_0^{\tau}, \quad Sig_{1,i}' = Sig_{1,i}^{\tau}, \ \sigma_0' = \sigma_0^{\tau}, \\ \sigma_1' &= \sigma_1^{\tau}, \quad \sigma_2' = \sigma_2^{\tau}, \ \sigma_3' = \tau\sigma_3, \\ \sigma_4' &= \tau\sigma_4, \quad \sigma_5' = \sigma_5, \ \sigma_6' = \tau\sigma_6. \end{aligned}$$

Then, the algorithm outputs the transformed signature:

$$\sigma' = \left\{Sig_0', Sig_{1,i}', \sigma_0', \sigma_1', \sigma_2', \sigma_3', \sigma_4', \sigma_5', \sigma_6'\right\},$$

and the transformation key τ.

7) CS.Verify: After receiving the transformed signature σ', the cloud server computes

$$\begin{aligned} Sig_{2,i} &= \frac{\sigma_0' \cdot e(g,g)^{\sigma_3'} \cdot \sigma_1' \cdot e(H(GID), g^{\sigma_4'})}{e(g,g)^{\sigma_5'} \cdot \sigma_2' \cdot e(H(GID), g^{\sigma_6'})}, \\ V_1 &= e(Sig_{1,i}', g^{y_{\rho(A_i)}}), \\ V_2 &= e(g,g)^{\alpha_{\rho(A_i)}}. \end{aligned}$$

If the MSR's attributes satisfy the given signature policy, the equation $\sum_i c_i A_i = (1, 0, \cdots, 0)$ will be true, where c_i is a set of constants. Then, the cloud server computes

$$V = \prod_i \left(V_1 \cdot V_2 \cdot Sig_{2,i}\right)^{c_i}.$$

Otherwise, the cloud server returns *false* to the MSP.

Finally, the algorithm outputs the intermediate signature V.

8) MSP.Verify: When the MSP receives the intermediate signature V, s/he uses the transformation key τ to compute a verification signature:

$$V' = \left(Sig_0{}' \cdot \frac{1}{h(M)}\right)^{\tau}.$$

Next, the MSP checks whether the equation $V = V'$ is true. If it is, the algorithm outputs *true* which represents the signature is valid. Otherwise, it outputs *false* represents the signature is invalid.

IV. SECURITY AND PERFORMANCE ANALYSIS

Here, the correctness of LPP-MSA is first proved. Next, the security is analyzed detailed. Finally, the performance analysis is compared with several selected schemes.

A. CORRECTNESS ANALYSIS

We can check the correctness of LPP-MSA by the following formulas.

Since $\lambda_i = A_i \cdot v$, $w_i = A_i \cdot w$, therefore

$$\sum_i \lambda_i \cdot c_i = \sum_i A_i \cdot v \cdot c_i = v \cdot (1, 0, \cdots, 0) = s,$$
$$\sum_i w_i \cdot c_i = \sum_i A_i \cdot w \cdot c_i = w \cdot (1, 0, \cdots, 0) = 0.$$

According to the above equations, we derive that:

$$\begin{aligned}
Sig_{2,i} &= \frac{\sigma_0{}' \cdot e(g,g)^{\sigma_3{}'} \cdot \sigma_1{}' \cdot e(H(GID), g^{\sigma_4{}'})}{e(g,g)^{\sigma_5{}'} \cdot \sigma_2{}' \cdot e(H(GID), g^{\sigma_6{}'})} \\
&= \frac{e(g,g)^{\lambda_i{}'\tau} \cdot e(g,g)^{\tau(\lambda_i - \lambda_i{}')} \cdot e(H(GID), g^{w_i{}'})^{\tau}}{e(g^{\alpha_i}, g) \cdot e(g,g)^{\alpha_{\rho(A_i)} - \alpha_i} \cdot e(H(GID)^{y_i}, g^{r_i})^{\tau}} \cdot \\
&\quad \frac{e(H(GID), g^{\tau(w_i - w_i{}')})}{e(H(GID), g^{\tau((y_{\rho(A_i)} - y_i) r_i)})} \\
&= \frac{e(g,g)^{\tau\lambda_i} \cdot e(H(GID), g^{w_i})^{\tau}}{e(g,g)^{\alpha_{\rho(A_i)}} \cdot e(H(GID), g^{\tau y_{\rho(A_i)} r_i})}
\end{aligned}$$

$$\begin{aligned}
&\prod_i \left(V_1 \cdot V_2 \cdot Sig_{2,i}\right)^{c_i} \\
&= \prod_i \left(\begin{array}{c} e(Sig_{1,i}{}', g^{y_{\rho(A_i)}}) \cdot e(g,g)^{\alpha_{\rho(A_i)}} \cdot \\ \frac{e(g,g)^{\tau\lambda_i} \cdot e(H(GID), g^{w_i})^{\tau}}{e(g,g)^{\alpha_{\rho(A_i)}} \cdot e(H(GID), g^{\tau y_{\rho(A_i)} r_i})} \end{array} \right)^{c_i} \\
&= \prod_i \left(\begin{array}{c} e(H(GID)^{\tau r_i}, g^{y_{\rho(A_i)}}) \cdot e(g,g)^{\alpha_{\rho(A_i)}} \cdot \\ \frac{e(g,g)^{\tau\lambda_i} \cdot e(H(GID), g^{w_i})^{\tau}}{e(g,g)^{\alpha_{\rho(A_i)}} \cdot e(H(GID), g^{\tau y_{\rho(A_i)} r_i})} \end{array} \right)^{c_i} \\
&= \prod_i \left(e(g,g)^{\tau\lambda_i} \cdot e(H(GID), g^{w_i})^{\tau}\right)^{c_i} \\
&= \prod_i \left(e(g,g)^{\lambda_i c_i} \cdot e(H(GID), g)^{w_i c_i}\right)^{\tau} \\
&= \left(e(g,g)^{s}\right)^{\tau}
\end{aligned}$$

B. SECURITY ANALYSIS

Based on the security model and design goals, we made the following security analysis on the proposed LPP-MSA.

Theorem 1 (Unforgeability): LPP-MSA is unforgeable under the random oracle model.

Proof: The following game is used to prove the unforgeability of the scheme.

Init: An adversary $\mathcal{A}$ selects a challenge access structure $\mathbb{A}$ firstly. Then, it sends $\mathbb{A}$ to a challenger $\mathcal{C}$.

Setup: The challenger $\mathcal{C}$ sets a bilinear group G of prime order p with a generator g, selects two hash functions $H : Z_p^* \to G$, $F : U \to G$ and outputs public parameters $GP = \{p, g, G, H, F\}$. Next, it selects $\alpha_i, y_i \in Z_p^*$ randomly and generates the verification key $PK_i = \{e(g,g)^{\alpha_i}, g^{y_i}\}$ and the signature key $SK_i = \{\alpha_i, y_i\}$. Finally, $\mathcal{C}$ sends the public parameters GP, the verification key PK_i to adversary $\mathcal{A}$ and keeps the signing key SK_i secretly.

Query phases: The adversary $\mathcal{A}$ can query signing key and signature to challenger $\mathcal{C}$ adaptively.

- **Signing-key queries.** The adversary $\mathcal{A}$ only asks the attribute set that does not meet the challenge structure $\mathbb{A}$. $\mathcal{A}$ first selects an identity GID, whose attribute set is ϕ. Then, it makes the signing key queries to $\mathcal{C}$. If the identity GID exists in the list L of the cloud server, challenger $\mathcal{C}$ returns the corresponding signing key $SK_{i,GID}$. Otherwise, $\mathcal{C}$ first selects $\alpha_i, y_i \in Z_p^*$ randomly. Then, for $i \in \phi$, it calculates $\{g^{\alpha_i}, H(GID)^{y_i}\}$, sets the corresponding signing key as $SK_{i,GID} = \{g^{\alpha_i}, H(GID)^{y_i}\}$ and sends the signing key to $\mathcal{A}$.
- **Signing queries.** The adversary $\mathcal{A}$ adaptively selects a message M and an access structure $\mathbb{A}$. Then, it sends M and $\mathbb{A}$ to challenger $\mathcal{C}$ for signature queries. Challenger $\mathcal{C}$ first selects an arbitrary attribute set $\phi \in \mathbb{A}$ and finds a set of constants c_i to make the equation $\sum_i c_i A_i = (1, 0, \cdots, 0)$ hold. Next, $\mathcal{C}$ runs the algorithms $IS \leftarrow Offline.Sign(GP, \{SK_{i,GID}\})$ and $\sigma \leftarrow Online.Sign(IS, M, (\mathbb{A}, \rho), \{SK_{i,GID}\})$ to generate a valid signature σ. Finally, $\mathcal{C}$ sends the signature σ to $\mathcal{A}$.
- **Transformed signature queries.** The adversary $\mathcal{A}$ chooses $\tau^* \in Z_p^*$ randomly, and computes $Sig_0{}' = Sig_0{}^{\tau^*}$, $Sig_{1,i}{}' = Sig_{1,i}{}^{\tau^*}$, $\sigma_0{}' = \sigma_0{}^{\tau^*}$, $\sigma_1{}' = \sigma_1{}^{\tau^*}$, $\sigma_2{}' = \sigma_2{}^{\tau^*}$, $\sigma_3{}' = \tau^*\sigma_3$, $\sigma_4{}' = \tau^*\sigma_4$, $\sigma_5{}' = \sigma_5$, $\sigma_6{}' = \tau^*\sigma_6$. Then, the adversary $\mathcal{A}$ generates a transformed signature $\sigma' = \{Sig_0{}', Sig_{1,i}{}', \sigma_0{}', \sigma_1{}', \sigma_2{}', \sigma_3{}', \sigma_4{}', \sigma_5{}', \sigma_6{}'\}$. For each query about (M, σ') from $\mathcal{A}$, $\mathcal{C}$ returns the transformed signature to $\mathcal{A}$.

Forgery. The adversary $\mathcal{A}$ outputs the message M^* and the forged signature V^* with the access structure $\mathbb{A}^*$. The analysis is similar to the ref. [32]. The adversary $\mathcal{A}$ can win this game if it can forge a signature in polynomial time with a non-negligible probability. Since this is based on a difficult mathematic problem, the adversary $\mathcal{A}$ cannot forge a valid signature correctly.

Theorem 2 (Anonymity): LPP-MSA can realize the anonymity when the MSR accesses the medical services.

Proof: In LPP-MSA, the MSR uses a series of attributes to represent his/her identity, which allows the MSR to request

TABLE 2. Comparison on computation overhead.

Schemes	Sign	Verify	Multi-Authority
EABS [36]	$(7l+15)\,E$	$(l+1)\,E+(l+2)\,P$	No
PPA-ABS [31]	$(4l+2)\,E+(2l+1)\,M$	$(2l+4)\,E+5M+R$	No
DMA-ABS [37]	$(14l+rl^2)M$	$E+13lM+lP$	Yes
LPP-MSA	$E+P+M$	$E+M$	Yes

medical services anonymously without exposing the specific identity. That is, an MSP cannot link the signature to a specific MSR.

Init: An adversary $\mathcal{A}$ selects a challenge access structure $\mathbb{A}$ firstly. Then, it sends $\mathbb{A}$ to a challenger $\mathcal{C}$.

Setup: The challenger $\mathcal{C}$ first sets a security parameter λ, then runs the algorithms $GP \leftarrow GlobalSetup(\lambda)$ and $\{(PK_i, SK_i)\} \leftarrow AuthoritySetup(GP, i)$. Finally, $\mathcal{C}$ sends the public parameters GP, the verification key PK_i to adversary $\mathcal{A}$ and keeps the signing key SK_i secretly.

Query phases: The phases include signing-key queries, signature queries and transformed signature queries. And the processes are the same as in Theorem 1.

Analysis: According to the scheme construction of LPP-MSA, $\mathcal{A}$ outputs the transformed signature $\sigma' = \{Sig_0', Sig_{1,i}', \sigma_0', \sigma_1', \sigma_2', \sigma_3', \sigma_4', \sigma_5', \sigma_6'\}$. Then MSR uploads the signature σ' to the healthcare cloud. In the verification process of MSP, since the signature's transformation key is random, s/he does not know the entire identity attribute set of the MSR, and thus does not know the specific identity of the MSR. MSP only knows that the medical data comes from a legitimate user and is authoritative. Therefore, MSR can request the medical services anonymously, thus realizing the privacy protection of identity.

Theorem 3 (Collusion Resistance): Our LPP-MSA can resist not only the collusion attack between multiple MSRs, but also the collusion attack between external attacker and the aided server.

Proof: We first show that two or more illegal MSRs cannot conspire to forge a legitimate signature. The proposed LPP-MSA can against collusion attack between MSRs by giving each MSR a global identity GID, in which different MSRs get different $e(H(GID)^{y_i}, g^{r_i})$ from attribute authorities. Suppose there are two MSRs, GID_1 and GID_2, whose attribute sets are ϕ_1 and ϕ_2 respectively, and they try to conspire to forge a signature. Because $e(H(GID_1)^{y_i}, g^{r_i})$ for $i \in \phi_1$ and $e(H(GID_2)^{y_i}, g^{r_i})$ for $i \in \phi_2$ are independent of each other, they cannot forge a valid signature on $\phi_1 \cup \phi_2$.

Next, we explain that an external attacker cannot forge a legitimate signature even if it colludes with the aided server. During the verification of LPP-MSA, the MSP first converts the signature into a transformed signature. Then, s/he sends the transformed signature σ' to the aided cloud server. What the cloud server receives is a transformed signature σ'. Therefore, the cloud server cannot obtain useful information from σ'. Even if the server and external attacker conspire together, they cannot forge a valid signature. So, the LPP-MSA can also resist the collusion attack between server and external attacker.

C. PERFORMANCE ANALYSIS

We now make a comprehensive performance analysis of LPP-MSA with several existing ABS schemes [31], [36], [37] which are named as PPA-ABS, EABS and DMA-ABS respectively.

1) ANALYSIS OF COMPUTATION COST

We analyze the efficiency by comparing the computation cost in signing and verification phases, where l and r represent the number of rows and columns in the access matrix $\mathbb{A}$ respectively, E represents exponentiation operation, P represents bilinear pairing operation, M represents multiplication operation, R represents the amount of calculation required to call the Rand program [38]. Other operations with low computation cost, such as modular addition operation, are ignored.

Table 2 shows the comparison results of computation cost. We can find that EABS and PPA-ABS do not support multi-authority mechanism, while DMA-ABS and LPP-MSA do. In the signing phase, the computation cost of EABS, PPA-ABS, DMA-ABS and LPP-MSA are all linearly associated with l. In LPP-MSA, a large amount of calculation was completed in the offline phase. So, the calculation of MSR is a fixed value $E+P+M$. In the verification phase, the calculation cost of EABS, PPA-ABS and DMA-ABS are linearly related to l. In LPP-MSA, an external server is used to assist MSP to complete the verification of signature, in which MSP only needs to perform lightweight calculation in the verification phase. So, the calculation of verification is a fixed value $E+M$. When there are many users in the system, LPP-MSA still has advantage.

From the above, LPP-MSA has the lowest computation overhead in the signing and verification phases, so it is lightweight and suitable for resource-constrained mobile healthcare terminals.

2) ANALYSIS OF STORAGE COST

We define that $|G|$ represents the length of element in group G, $|S|$ represents the number of user's attributes, and $\left|Z_p^*\right|$ represents the length of element in group Z_p^*.

TABLE 3. Comparison on storage cost.

Schemes	Signing Key.size	Signature.size
EABS [36]	$(l+1)\|G\|$	$(7l+11)\|G\|$
PPA-ABS [31]	$(2+\|S\|)\|G\|$	$(3l+1)\|G\|$
DMA-ABS [37]	$15l\|G\|$	$13l\|G\|$
LPP-MSA	$2\|S\|\|G\|$	$(4l+1)\|G\|+4l\left\|Z_p^*\right\|$

The comparison of storage cost are shown in Table 3. Since the signature is corresponding to the signature policy, the signature size of four schemes are all linearly related with l. In LPP-MSA, the storage cost of signature is $(4l+1)|G|+4l\left|Z_p^*\right|$, which is larger than that of EABS and PPA-ABS. However, due to the high storage capacity of the cloud server, we can ignore the storage cost of signature on cloud server and only consider the users' storage cost. The storage overhead of signing keys in EABS and DMA-ABS increases linearly as l increases. The signing keys of PPA-ABS and LPP-MSA are directly generated by the attribute authorities according to the users' attributes, so the storage overhead is related to $|S|$.

3) PERFORMANCE EVALUATION

In order to further compare each scheme, we first build a test simulation platform. The experimental environment is Linux Ubuntu 16.04 LTS with an Intel (R) Core (TM) i5-4200U CPU 1.6 GHz × 2 processor. Our experiment is based on the pbc-0.5.14 library, in which the type A curve $y^2 = x^3 + x$ is selected. Here, we test each basic operation 1000 times and Table 4 shows the basic operation results.

TABLE 4. Running time of basic operations.

Operations	Time (ms)
P	1.251
M	1.845
E	2.040

For the sake of discussion, we set the user's attribute set as a subset of the policy attribute set and make user's attributes always meet the access policy, so that the attribute set used for decryption is consistent with the user's attribute set, thus ensuring the unity of variables. In the process of the experiment, we selected 10 attribute authorities for the multi-authority ABS schemes, and set 10 attributes for each attribute authority. When the policy attribute changes slowly from 1 to 10, we set the user's attributes to a fixed value 2.

In the signing phase, the computation overhead of EABS, PPA-ABS and DMA-ABS are all related to the number of policy attributes l. EABS needs $(7l + 15)$ exponentiation operations and its total signing time is $14.28l + 30.6$ ms. PPA-ABS includes $(4l + 2)$ exponentiation operations and $(2l + 1)$ multiplication operations. So, its total signing time is $11.85l + 5.925$ ms. DMA-ABS needs to execute $(14l + rl^2)$ multiplication operations and its total signing time is $25.83l + 1.845rl^2$ ms. By contrast, our LPP-MSA only involves one exponentiation operation, one pairing operation and one multiplication operation. Therefore, the total signing time of LPP-MSA is 5.136 ms. Fig. 4 further gives change trend of the computation time with the increase of l in signing phase of different schemes.

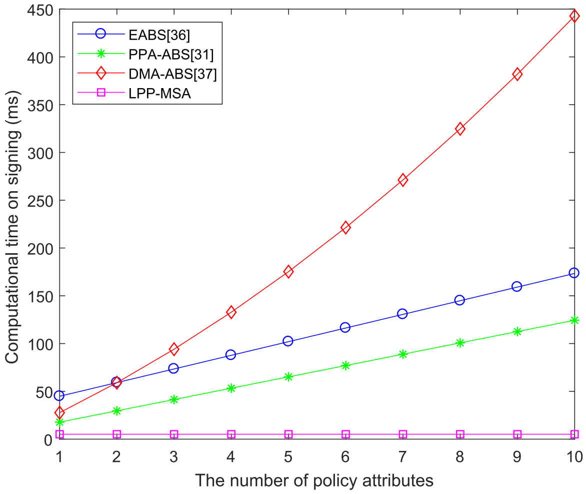


FIGURE 4. Comparison on time consumption of signing between different schemes.

In Fig. 4, when l increases, the computation time of EABS, PPA-ABS and DMA-ABS will also increase accordingly. For $l = 1$, the signing time of EABS, PPA-ABS and DMA-ABS is 44.88, 17.775 and 27.675 ms respectively. For $l = 10$, the signing time of EABS, PPA-ABS and DMA-ABS is 173.4, 124.425 and 442.8 ms respectively. In all schemes, LPP-MSA maintains a lowest computation overhead with the increase of l.

In the verification phase, $(l+1)$ exponentiation operations and $(l + 2)$ pairing operations are needed in EABS. So, its total time of verification is $3.291l + 4.542$ ms. In PPA-ABS, $(2l+4)$ exponentiation operations, five multiplication operations and one R operation are required. Here, R might be negligible. So, the total verification time of PPA-ABS is $4.08l + 17.385$ ms. In DMA-ABS, one exponentiation operation, $13l$ multiplication operations and l pairing operations are needed and the total verification time is $25.236l+2.04$ ms. Finally, in LPP-MSA, only one exponentiation operation and one multiplication operation are involved. Therefore, the total verification time of LPP-MSA is 3.885 ms. Fig. 5 illustrates the change trend of the computation overhead of each scheme in verification phase.

In Fig. 5, the computation overhead of EABS, PPA-ABS and DMA-ABS will rise by increasing l. For $l = 1$, the verification time of EABS, PPA-ABS and DMA-ABS is 7.833, 21.465 and 27.276 ms respectively. For $l = 10$,

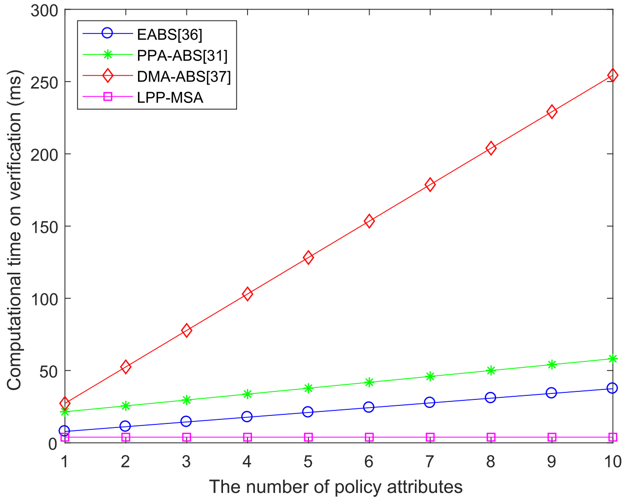


FIGURE 5. Comparison on time consumption of verification between different schemes.

the verification time of EABS, PPA-ABS and DMA-ABS is 37.452, 58.185 and 254.4 ms respectively. Due to the calculation cost of LPP-MSA is a fixed value 3.885 ms, so the computation time does not change with the increase of l.

From the Fig. 4 and Fig. 5, we find that LPP-MSA has the lowest computation time in four schemes. As a result, LPP-MSA is efficient and has more advantages in the phases of signing and verification than the other selected schemes.

V. CONCLUSION

In this paper, we proposed a lightweight and privacy-preserving medical services access scheme based on multi-authority ABS for healthcare cloud, named LPP-MSA. In this scheme, MSR can access remote medical services without fully exposing his/her identity information, thus realizing privacy protection. Using online/offline signing and server-aided verification mechanisms can reduce the calculation cost, which allows MSR to access medical services via resource-limited mobile devices (such as smart phone, laptop, and so on). The security analysis shows that LPP-MSA meets the requirements of unforgeability, anonymity and collusion resistance. Furthermore, the performance analysis of LPP-MSA and several existing schemes shows that LPP-MSA has high computational efficiency. Therefore, LPP-MSA is more suitable for large scale remote medical services access in healthcare cloud system.

REFERENCES

[1] W. Tang, J. Ren, and Y. Zhang, "Enabling trusted and privacy-preserving healthcare services in social media health networks," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 579–590, Mar. 2019.

[2] X. Du and H. H. Chen, "Security in wireless sensor networks," *IEEE Wireless Commun. Mag.*, vol. 15, no. 4, pp. 60–66, Aug. 2008.

[3] M. A. Sahi, H. Abbas, K. Saleem, X. Yang, A. Derhab, M. A. Orgun, W. Iqbal, I. Rashid, and A. Yaseen, "Privacy preservation in e-healthcare environments: State of the art and future directions," *IEEE Access*, vol. 6, pp. 464–478, 2017.

[4] Y. Xiao, X. Du, J. Zhang, F. Hu, and S. Guizani, "Internet protocol television (IPTV): The killer application for the next-generation Internet," *IEEE Commun. Mag.*, vol. 45, no. 11, pp. 126–134, Nov. 2007.

[5] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in *Proc. Cryptogr.' Track RSA Conf.*, 2011, pp. 376–392.

[6] X. Chen, J. Li, X. Huang, J. Li, Y. Xiang, and D. S. Wong, "Secure outsourced attribute-based signatures," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, pp. 3285–3294, Dec. 2014.

[7] T. Yang, B. Yu, H. Wang, and J. Li, "Revocable attribute-based ring signature scheme with constant size signature," in *Proc. IEEE Int. Conf. Comput. Commun. (ICCC)*, Oct. 2015, pp. 100–104.

[8] D. Khader, "Attribute based group signatures," Cryptol. ePrint Arch., Tech. Rep. 2007/159, 2007. [Online]. Available: https://eprint.iacr.org/2007/159

[9] Y. Qian and Y. Zhao, "Strongly unforgeable attribute-based group signature in the standard model," in *Proc. IEEE Int. Conf. Intell. Comput. Intell. Syst.*, Oct. 2010, pp. 843–852.

[10] S. T. Ali and B. Amberker, "Attribute-based group signature without random oracles with attribute anonymity," *Int. J. Inf. Comput. Secur.*, vol. 6, no. 2, pp. 109–132, 2014.

[11] X. Du, M. Guizani, Y. Xiao, and H.-H. Chen, "A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1223–1229, Mar. 2009.

[12] J. Herranz, F. Laguillaumie, B. Libert, and C. Ràfols, "Short attribute-based signatures for threshold predicates," in *Proc. Cryptogr.' Track RSA Conf.*, 2012, pp. 51–67.

[13] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proc. Annu. Int. Cryptol. Conf.*, 2004, pp. 41–55.

[14] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Netw.*, vol. 5, no. 1, pp. 24–34, 2007.

[15] P. Yang, Z. Cao, and X. Dong, "Fuzzy identity based signature," Cryptol. ePrint Arch., Tech. Rep. 2008/002, 2008. [Online]. Available: https://eprint.iacr.org/2008/002

[16] H. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures: Achieving attribute-privacy and collusion-resistance," Cryptol. ePrint Arch., Tech. Rep. 2008/328, 2008. [Online]. Available: https://eprint.iacr.org/2008/328

[17] J. Li and K. Kim, "Attribute-based ring signatures," Cryptol. ePrint Arch., Tech. Rep. 2008/394, 2008. [Online]. Available: https://eprint.iacr.org/2008/394

[18] S. F. Shahandashti and R. Safavi-Naini, "Threshold attribute-based signatures and their application to anonymous credential systems," in *Proc. Int. Conf. Cryptol. Africa*, 2009, pp. 198–216.

[19] G. Shanqing and Z. Yingpei, "Attribute-based signature scheme," in *Proc. Int. Conf. Inf. Secur. Assurance (ISA)*, Apr. 2008, pp. 509–511.

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

[21] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," in *Proc. Int. Conf. Cryptol. India*, 2008, pp. 426–436.

[22] D. Cao, T. Wang, X. Wang, and J. Su, "An expressive attribute-based signature scheme without random oracles," in *Proc. Int. Conf. Comput. Appl. Syst. Modeling*, 2012, pp. 0560–0564.

[23] D. Cao, B. Zhao, X. Wang, and J. Su, "Flexible multi-authority attribute-based signature schemes for expressive policy," *Mobile Inf. Syst.*, vol. 8, no. 3, pp. 255–274, 2012.

[24] Y. Sun, R. Zhang, X. Wang, K. Gao, and L. Liu, "A decentralizing attribute-based signature for healthcare blockchain," in *Proc. 27th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2018, pp. 1–9.

[25] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11676–11686, 2018.

[26] S. Even, O. Goldreich, and S. Micali, "On-line/off-line digital signatures," in *Proc. Conf. Theory Appl. Cryptol.*, 1989, pp. 263–275.

[27] A. Shamir and Y. Tauman, "Improved online/offline signature schemes," in *Proc. Annu. Int. Cryptol. Conf.*, 2001, pp. 355–367.

[28] X. Chen, F. Zhang, and K. Kim, "Chameleon hashing without key exposure," in *Proc. Int. Conf. Inf. Secur.*, 2004, pp. 87–98.

[29] C. Z. Gao, B. Wei, D. Xie, and C. Tang, "Divisible on-line/off-line signatures," in *Proc. Cryptogr.' Track RSA Conf.*, 2009, pp. 148–163.

[30] Z. Wang, R. Xie, and S. Wang, "Attribute-based server-aided verification signature," *Appl. Math. Inf. Sci.*, vol. 8, no. 6, pp. 3183–3190, 2014.

[31] Y. Han, F. Chen, and L. Xi, "Privacy preserved aid-verification attribute based signature scheme," in *Proc. 36th Chin. Control Conf. (CCC)*, Jul. 2017, pp. 5695–5699.

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

[33] R. Mo, J. Ma, X. Liu, and H. Liu, "EOABS: Expressive outsourced attribute-based signature," *Peer-Peer Netw. Appl.*, vol. 11, no. 5, pp. 979–988, 2018.

[34] J. Sun, J. Qin, and J. Ma, "Securely outsourcing decentralized multi-authority attribute based signature," in *Proc. Int. Symp. Cyberspace Saf. Secur.*, 2017, pp. 86–102.

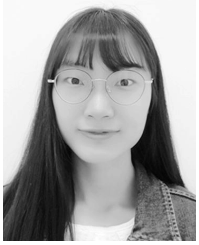
[35] Y. Ren and T. Jiang, "Verifiable outsourced attribute-based signature scheme," *Multimedia Tools Appl.*, vol. 77, no. 14, pp. 18105–18115, 2018.

[36] T. Okamoto and K. Takashima, "Efficient attribute-based signatures for non-monotone predicates in the standard model," in *Proc. Int. Workshop Public Key Cryptogr.*, 2011, pp. 35–52.

[37] T. Okamoto and K. Takashima, "Decentralized attribute-based signatures," in *Proc. Int. Workshop Public Key Cryptogr.*, 2013, pp. 125–142.

[38] Y. Ren, J. Cai, and C. Huang, "Verifiable outsourcing private key generation algorithm in an identity-based encryption scheme," *J. Commun.*, vol. 36, no. 11, pp. 61–66, 2015.

JINGWEI LIU (M'11) received the B.S. degree majoring in applied mathematics, and the M.S. and Ph.D. degrees majoring in communication and information systems from Xidian University, Xi'an, China, in 2001, 2004, and 2007, respectively, where he is currently with the School of Telecommunications Engineering. He has published more than 40 papers in journals and conference proceedings. His research interests include information security, network security, and cryptography. He is a member of the Chinese Association for Cryptologic Research.



HUIFANG TANG is currently pursuing the M.S. degree in electronics and communication engineering with Xidian University. Her research interests include attribute-based encryption/signature and privacy preserving.

RONG SUN (M'10) received the B.E. degree in telecommunications engineering, and the M.E. and Ph.D. degrees in communications and information systems from Xidian University, Xian, China, in 1998, 2001, and 2008, respectively, where she is currently with the School of Telecommunications Engineering. She is a member of IEICE. Her research interests include wireless communications, channel coding design, and information theory.

XIAOJIANG DU (M'04–SM'09) received the B.E. degree from Tsinghua University, China, in 1996, and the M.S. and Ph.D. degrees from The University of Maryland, College Park, in 2002 and 2003, respectively, all in electrical engineering. He is currently a Professor with the Department of Computer and Information Sciences, Temple University. His research interests include security, systems, wireless networks, and computer networks. He has published over 300 journal and conference papers in these areas. He serves on the Editorial Boards of two international journals.

MOHSEN GUIZANI (S'85–M'89–SM'99–F'09) received the B.S. and M.S. degrees in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, in 1984, 1986, 1987, and 1990, respectively. He was the Associate Vice President of Qatar University, the Chair of the Computer Science Department, Western Michigan University, the Chair of the Computer Science Department, University of West Florida, and the Director of graduate studies at the University of Missouri–Columbia. He is currently a Professor with the Department of Computer Science and Engineering, Qatar University. He has authored or coauthored nine books and publications in refereed journals and conferences. His research interests include wireless communications and mobile computing, vehicular communications, smart grid, cloud computing, and security.