IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 23, 2018, accepted May 21, 2018, date of publication May 25, 2018, date of current version June 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2840504

Efficient and Privacy-Preserving Medical Data Sharing in Internet of Things With Limited Computing Power

DONG ZHENG[1,2], AXIN WU[1], YINGHUI ZHANG[1,2], (Member, IEEE), AND QINGLAN ZHAO[1]

[1]National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
[2]Westone Cryptologic Research Center, Beijing 100070, China

Corresponding author: Qinglan Zhao (zhaoqinglan@foxmail.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0802000, in part by the National Natural Science Foundation of China under Grants 61772418, 61472472, and 61402366, and in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grants 2016JM6033 and 2015JQ6236.

ABSTRACT With the application of Internet of Things (IoT) technologies in smart city, intelligent medical terminals play a more and more significant role in our daily life. These terminals can monitor our physical conditions and get lots of medical data in time. For the sake of data security and practicality, the collected big data can be encrypted and then stored on a cloud server such that only authorized users, such as the data owner and the doctors, can access. However, smart terminals are usually limited in computing power and users' privacy issues remain. To tackle this challenging problem, an efficient medical data sharing scheme is presented in this paper. To solve the privacy issues in users' data sharing, we utilize attribute-based encryption to enable data sharing. In addition, we remove the attribute matching function and use the attribute bloom filter to hide all the attributes in the access control structure. In order to improve the efficiency of encryption, we introduce the online/offline encryption technology in the encryption phase. Before the message is known, a large amount of work that is needed at the encryption stage will be done. Then, once the message is known, the ciphertext can be generated quickly. Besides, the initialization stage of the system does not need to specify all attributes. When the overall attributes of the system users increase, the system does not need to be reinitialized, which will also improve the system efficiency. Security analysis and performance analysis show that the data sharing scheme is secure and can improve data processing ability in IoT based data sharing.

INDEX TERMS Attribute-based encryption, data share, attribute bloom filter, privacy-preserving.

I. INTRODUCTION

With the application of information and communication technology in the smart city paradigm, the Internet of Things (IoT) is affecting our daily routine and lifestyle. The IoT system is made up of a large number of heterogeneous intelligent objects (our smartphones, portable health devices) that can collect a lot of sensitive data, then stored on a cloud server to be shared by people. Due to the distributed characteristics of the IoT [1], to share the collected data with other people and things requires a flexible access control strategy. How to control data access more flexibly has become an urgent problem. To deal with this problem, the technology called Attribute-Based Encryption (ABE) is proposed.

In the attribute-based encrypted system, a specific ciphertext can be decrypted by a particular private key if and only if the user's attribute set matches the access control policy. In order to express more flexible access policies, attribute encryption is divided into CP-ABE [2]–[4] and KP-ABE [5], [6], depending on whether ciphertext is contacted with attributes of users or access control policy. Furthermore, The CP-ABE is more suitable for data access control. In the CP-ABE system, the ciphertext is connected to an access control structure in the encryption stage while the users' private key is connected with the attribute sets in the process of building a secret key. For the owner of the data, he can control the access control strategy flexibly. For visitors, the ciphertext can be decrypted if the attribute set of users matches the predefined access structure.

The IoT scenarios have changed the way people communicate with the environment. And IoT also provides an exchange sensitive or personal data platform between doctors and patients in a smart health system. If the doctor can

get the patient's medical information in time, the patient will get better medical service. The various medical parameters of people can be collected through medical sensors. Then, these medical parameters are sent through IoT to the cloud server. In addition, people can also pass their personal medical information through IoT to the cloud server through mobile terminals. When the data is exposed to an open network, the privacy and security of data will be threatened. On the one hand, privacy protection is a must to consider when these data is shared, which contains a lot of personally sensitive information, so how to protect user privacy has become a challenging problem. On the other hand, since these sensors and mobile terminals are resource constrained devices [7], how to quickly produce ciphertext stored on the cloud server is another challenging question.

A. OUR WORK

In order to protect the privacy of users and improve the efficiency of encryption, we propose a secure medical data sharing system, where sensors and mobile terminals can encrypt sensitive data of users, then send it cloud servers. And users who can satisfy access control structure can access data in this system. The contribution of this article is mainly three points below.

- Firstly, in our security system, when ciphertext is uploaded to the cloud server, the access control structure (M, ρ) will also be uploaded. If the attribute matching function is removed, attributes will be hidden into the access structure. The access control structure will also leakage user privacy. By using the attribute bloom filter (ABF), we can hide in the entire attributes in the anonymous access control structure. To this end, the data stored on the cloud server will be protected.
- Secondly, to generate the ciphertext more quickly, we use online/offline encryption technology. Before the encrypted information is known, a large amount of work that is needed at the encryption stage will be done. When the encrypted information is known, the ciphertext can be generated quickly. To this end, the efficiency of encryption will also be solved.
- Finally, in our scheme, the initialization stage of the system does not need to specify all attributes. When the overall attributes of the system users increase, the system does not need to be reinitialized, which will is also a way to improve the efficiency.

B. RELATED WORK

With the application of the IoT terminals, more and more people attach importance to the security and privacy of personal information, not only from the biological characteristics [8], [9], but also the security of the stored data. The IOT scenario provides an exchange sensitive or personal data platform [10], and a lot of privacy protection works on the IoT and the cloud storage are done. Cai *et al*. [11] implement electronic health record in mobile health cloud to solve the problem of privacy and efficiency. Privacy in many applications can also be guaranteed by machine learning. Li *et al*. [12] propose a novel privacy-preserving Naive Bayes learning scheme with multiple data sources. Bernabe *et al.* [13] proposed an IoT security framework to protect the privacy of data in the exchange of data, but, it doesn't take into account the resource constraints. Odelu *et al*. [14] proposed a novel CP-ABE schema in order to make the ciphertext and the private key with constant-size, but, it not apply to the a high expressivity in access policies. Xiong and Xu [15] puts forward another security model by combining CP-ABE with symmetric key cryptographic to share data for security, which has an expensive computational cost. Liu *et al*. [16] propose DivORAM based on additively homomorphic encryption scheme to save the client computing overhead and the network bandwidth cost. Lin *et al*. [17] propose an ID-based linearly homomorphic signature schemes in E-business and cloud computing. Some of the solutions in cloud storage [18]–[21] have solved some of the privacy issues and potential safety hazard [22], but it does not apply to a resource-constrained intelligent medical system environment. Zhang *et al*. [23] considers resource constraints and data processing efficiency and focuses on the decryption phase. But, when the intelligent terminals of the Internet of things collect data, we attach more importance to the efficiency of the encryption phase. In a word, the attribute encryption can be well used in data sharing scenarios.

Sahai and Waters [24] put forward the first attribute-base encryption (ABE). Here are two ways of ABE called CP-ABE [2], [3] and KP-ABE [5], [6] that Goyal *et al*. [5] proposed to access encrypted data more flexibly. The first large universe KP-ABE scheme that was presented by Lewko and Waters [25] is constructed on composite order bilinear group in the standard model. Next, the first large universe KP-ABE scheme that Lewko [26] presented is constructed on prime order bilinear group in the standard model. In addition, another large universe scheme was proposed by Rouselakis and Waters [27] in the standard model in prime order bilinear group. Li *et al*. [28] propose a privacy-aware multi-authority ciphertext-policy ABE scheme with accountability. Their ABE scheme may hide the attribute information in the ciphertext to trace the dishonest user identity who shares the decryption key.

In practical applications, efficiency is an obstacle to application. To address this problem, an efficient outsourcing calculation algorithm was proposed by Chen *et al*. [29], but it really increases the communication cost. Some efficiency methods [30], [31] have been proposed to reduce the cost of communication. However, they do not apply the sharing of data. Another effective solution is the offline/online technique. This technique is first used in the signature scheme. The concept of offline/online technique was first presented by Even *et al*. [32]. Rouselakis and Waters [33] first applies the offline/online technology to the attribute-based encryption scheme.

When we upload encrypted data and policies to the open network such as the cloud server, the security of the data

and the privacy of the user will become more challenging. More accurately, the access policy leakage some sensitive information. From the point of view of policy hiding, some researches [34]–[37] has been done. Unfortunately, in these ABE schemes the attributes are not hidden or anonymized. Further, these studies [34], [35] realized privacy protection by hiding the values of each attribute. The research [36] realized privacy protection by hiding vector encryption. The study [37] achieved privacy protection by inner product encryption. Although hiding value of attributes protected the user's privacy to a certain extent, the names of attributes may also leak some sensitive information. In addition, most of these schemes were used in a specific access policy. Yang *et al.* [38] proposed a novel scheme with privacy-preserving policy by using the ABF. To access data more flexibly, Lai *et al.* [39] presented a scheme which hides attribute values in LSSS structure.

C. THE ORGANIZATION OF THIS ARTICLE

The structure of this article is as follows: firstly, we introduce some preliminary knowledge in part 2. Next, we put forward the architecture of our scheme in part 3. After that, we propose the scheme in part 4. Then, the security proof is given in part 5, and the performance of the presented scheme is given in part 6. Finally, the conclusion is arrived in part 7.

II. PRELIMINARY

The basic knowledge and concepts related to this paper will be introduced in this part.

A. BILINEAR PAIRING

The paper [40] gives the background of bilinear maps.

Two multiplicative groups with the same prime order p are expressed by G and G_T. The mapping from G to G_T is expressed in e. We can get the following properties of the bilinear map e:

- Bilinearity: $e(g_1^u, g_2^v) = e(g_1, g_2)^{uv}$ for $\forall g_1, g_2 \in G$ and $u, v \in Z_p$
- Non-degeneracy: $e(g, g) \neq 1$ unless g=1.

B. LINEAR SECRET-SHARING SCHEMES

Waters [40] applies a linear secret sharing scheme (LSSS) to attribute-based encryption. If the following conditions are set up, A LSSS Π is called linear over the prime order Z_p over a set of parties P.

1) each attribute can be represented as a vector over Z_p;
2) the matrix $M_{k\times m}$ is called the share-generating matrix for Π. For each $i \in [1, \ldots, k]$, the i_{th} row of the matrix is expressed by a party $\rho(i)$, where ρ is a mapping function from $\{1, \ldots, k\}$ to P. If we want to share the secret $s \in Z_p$, we first choose $r_2, \ldots, r_m \in Z_p$ randomly. Then we can get the vector $v = (s, v_2, \ldots, v_m)$. The vector of k shares of the secret s is $A\overrightarrow{v}$ according to Π.

The LSSS with the linear reconstruction property is described as follows: If Π is an linear secret sharing scheme for the access structure A. S expresses the any authorized set, and let $I \in \{1, 2, \ldots, k\}$ be expressed as $I = \{i : \rho(i) \in S\}$. Next, we can find the constants $\{\omega_i \in F_p\}_{i\in I}$ in polynomial time according to Π, which is able to satisfy the equation $\sum_{i\in I} \omega_i \lambda_i = s$.

C. BLOOM FILTER

Bloom [41] in 1970 put forward the concept of bloom filter (BF). The BF is a kind of effective in probabilistic data structure on the storage space. we can judge whether an element is in a set by using BF. Clearly, l independent hash functions make up a BF, which is an array of n bits, explained as follows: $h_j : \{0, 1\}^* \to [1, n]$ for $1 \leq j \leq l$.

At first, the n bits of the array are set to 0. If we want to add an element e to the collection, the BF building algorithm computes l hash functions as indices $\{h_i(e)\}_{i\in[1,l]}$ and changes the values at $\{h_i(e)\}$ in the n bit array to 1.

To judge whether a given element y is a member of the set S, all the hash values $\{h_j(y)\}_{j\in[1,l]}$ will be computed by the BF query algorithm to get l array positions. If all of the bits at these positions are equal to 0, the element y not belong to the set. However, if any of the bits is equal to 1, we can say the given element y probably is a member of the set S. There is a possibility for some element y not $\in S$, any of the bits at the corresponding positions of $h_i(y)$ is 1, which is known as the miscarriage of Justice.

D. DECISIONAL ASSUMPTION

The following definition is about the decisional q-bilinear Diffie-Hellman exponent (Decisional q-BDHE) problem.

According to the system security parameter λ, we choose a group G of prime order p. And we also select $a, s \in Z_p^*$ randomly and let g and g_i represent the generator of G and g^{a_i} respectively. When the adversary is given $\hat{y} = (g, g_1, \ldots, g_q, , g_{q+2}, \ldots, g_{2q}, g^s)$, he can tell $\hat{e}(g, g)^{a^{q+1}s} \in G_T$ from a random element R in G_T. If

$$|Pr[B(\bar{y}, T = (g, g)^{a^{q+1}s}) = 0] - Pr[B(\bar{y}, T = R) = 0]| \geq \epsilon$$

holds, we say that the algorithm B has advantage ϵ in dealing with decisional q-BDHE problem in G.

Definition 1: We think that the Decisional q-BDHE assumption holds if no polynomial time algorithm solves the q-BDHE problem with the advantage that can not be ignored.

III. MODEL DEFINITION

We first introduce the model of the medical data sharing system. Then, we present the definition of model and give the security model in this part.

A. SYSTEM MODEL

The medical data sharing model is as shown in Figure 1, It is made up of the following four entities:

- Cloud Server: The cloud server is provided by third parties, which was used to store the ciphertext, access structure and bloom filter. However, the cloud server is semi-trusted. At the same time, we assume that there is

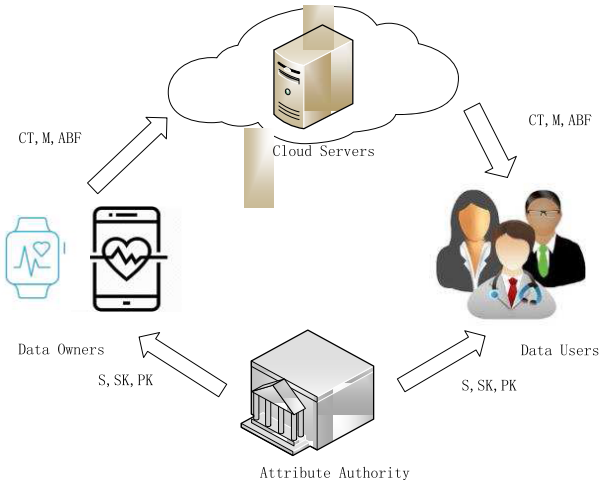

FIGURE 1. System model.

no conspiracy between the cloud server and data owners and data users.

- Attribute Authority: All the attributes of the system are administrated by the attribute authority. And it issues the attribute public key of this system and the private key for the users. The attribute center is completely trustworthy.
- Data Owner: The owner of the data is the person who wants to share personal health information. But, information is collected through sensors or smartphones. Before sharing share, Plaintext needs to be encrypted. Then, the ciphertext and the access control structure as well as bloom filter are uploaded to the cloud server. The access control strategy is controlled by the owner of the data.
- Data User: The users of the data are those people who want to access the ciphertext on the cloud server. They may be doctors, relatives, or friends of the data owners. When the attributes of users satisfy the access control strategy, the ciphertext can be deciphered correctly.

B. SCHEME MODEL

The medical data sharing scheme we proposed is made up of the following algorithms:

- $Initialization(1^{\lambda}) \rightarrow (PK, MSK)$. This algorithm is executed by the attribute authority, which takes the security parameter λ as an input, then takes outputs the public key PK and the master secret key MSK as outputs.
- $KeyGen(PK, MSK, S) \rightarrow SK$. This algorithm is executed by the attribute authority. It takes in the public parameters PK, the master secret key MSK, and the attribution of users set S. It outputs the secret key SK of users.
- $Encryption(PK, m, (M, \rho)) \rightarrow (CT, ABF)$. The encryption algorithm contains three subroutines: $Offline.Encrypton$, $Online.Encryption$, $ABFbulid$.
  - $Offline.Encryption(PK) \rightarrow IT$. The data owner inputs the public parameters PK and outputs an intermediate ciphertext IT.
  - $Online.Encryption(PK, IT, m, (M, \rho)) \rightarrow CT$. The sensors or smartphones take as input the public parameters PK, an intermediate ciphertext IT and the information m to be encrypted as well as an access structure (M, ρ). Then,it outputs a ciphertext CT.
  - $ABFBuild(M, \rho) \rightarrow ABF$. The data owner takes in an access structure (M, ρ) and outputs the attribute bloom filter.
- $Decryption(M, ABF, PK, SK, CT) \rightarrow m$. The decryption algorithm is executed by the data user, which contains the two subalgorithms: $ABFQuery$ and Dec.
  - $ABFQuery(S, ABF, PK) \rightarrow \rho'$. The data user inputs the attribution set S, the ABF and the PK. The ABFQuery algorithm outputs the secret a reconstructed attribute mapping $\rho' = (rownum, att)_S$. The mapping shows the cascading of the corresponding row number of the matrix M and all the attributes $att \in S$.
  - $Dec(SK, CT, (M, \rho')) \rightarrow m \ or \ \perp$. The data user inputs the SK, the ciphertext CT and the reconstructed attribute mapping ρ', then returns the message m if the attributes can satisfy the access policy, Otherwise, this algorithm outputs $\perp$.

C. DEFINITION OF SECURITY MODEL

Our scheme is against the selectively chosen plaintext attack between an adversary A and a simulator B with the indistinguishability, the game is as follows

- Init: The adversary A sends the challenge access structure (M^*, ρ^*), which he will try to attack, to the challenger.
- Setup: The simulator performs the setup algorithm and outputs the public parameters PK, then send it to the adversary A.
- Phase 1: At this stage, A queries the private key associated with the attribute S_{attr} for the simulator B. B generates the secret key related the attribute S_{attr} and return it to the adversary A, but the attribute S_{attr} was inquired cannot meet the access control structure (M^*, ρ^*).
- Challenge: A gives two equal length messages m_0 and m_1 to the B. B randomly selects $b \in \{0, 1\}$ and encrypts m_b under the challenge access structure M^*. At last, the simulator returns the challenge ciphertext CT^* to the adversary A.
- Phase 2: Phase 2 is the same as Phase 1.
- Guess: A outputs a guess $b' \in \{0, 1\}$ for b and wins this security game if $b = b'$. Then, the advantage of A is defined as $Adv(A) = |Pr[b = b'] - \frac{1}{2}|$.

IV. THE PROPOSED SCHEME

The concrete data sharing scheme will be given in this part. The steps are as follows:

A. SYSTEM INITIALIZATION

The attribute authority takes a security parameter λ, and selects a group G of primer p, a generater g and a bilinear map $e : G \times G \rightarrow G_T$. L_{at} expresses the maximum bit length of attributes in the system. L_{rn} expresses the maximum bit length of the row numbers of access matrix. L_{ABF} expresses the size of bit array of the ABF. k expresses the number of hash functions associated with the ABF. Then, it randomly choose $\alpha, a, \in Z_p^*$ and generates k hash functions $H_1(), H_2(), \ldots, H_k()$ that maps an element to a position in the range of $[1, L_{ABF}]$.

The public key and the master secret key are as follows

$$PK = < G, G_T, g, g^a, e(g,g)^\alpha, L_{at}, L_{rn}, L_{ABF}, H_1(), H_2(), \ldots, H_k(), l, H > .$$

$$MSK = (\alpha).$$

B. KEY GENERATION

Every data owner and data user should register and authenticate to the attribute authority. If they are not lawful, it aborts. Otherwise they will assign attributes and the secret key related to the attributes.

The attribute authority chooses a random number $t, x_1, x_2, \ldots, x_{|S|}$ and calculates $h_i = g^{x_i}$ and $K = g^\alpha g^{at}, K_0 = g^t, K_i = h_i^t$ for $i \in S$. Then, it inputs the PK, the MSK and an attribute set S, and outputs the private key related to attribute set S:

$$SK = (S, K, K_0, \{K_i\}_{i \in S}).$$

C. DATA ENCRYPTION

The Data owner does a lot of pre-encryption work and store it in sensors and smart phones, for example, the intermediate ciphertext and attribute storage bloom filter. When the information to be encrypted is known, sensors and smart phones can quickly generate ciphertext and upload them to cloud servers. The date encryption algorithm consists of the following three step Subalgorithms:

Offline.Encryption(PK): The data owner only inputs the public parameters PK, we assumes that the P is the maximum bound of matrix M in any LSSS access structure. The data owner first selects a random $s \in Z_p$ and computers

$$key = e(g,g)^{\alpha s}, \quad C_2 = g^s.$$

Then, for j = 1 to P, it picks randomly λ_i', r_i and computers

$$C_{1,i} = g^{a\lambda_i'} h_i^{-r_i}, \quad C_{2,i} = g^{r_i}.$$

Intermediate ciphertext is

$$IT := (s, \lambda_i', key, C_2, \{C_{1,i}, C_{2,i}\}_{i \in [1,p]}).$$

Online.Encryption($PK, IT, m, (M, \rho)$): The data owner selects randomly $y_2, y_3, \ldots, y_n \in Z_p$, sets the vector $y = (s, y_2, y_3, \ldots, y_n)^T$ and computers a vector of shares of s as $(\lambda_1, \ldots, \lambda_l)^T = My$. The public parameters PK, an intermediate IT, and an LSSS access structure M, where M is an $l \times n$ matrix, and $l \leq p$ will be stored on the sensors and the smartphones.

When sensors and smartphones collect medical data and parameters, they can quickly generate ciphertext. The concrete steps are as follows:

For j = 1 to l, it computers

$$C_1 = key \cdot m,$$
$$C_{3,i} = \lambda_i - \lambda_i'.$$

Finally, the ciphertext is

$$CT := ((M, \rho), C_1, C_2, \{C_{1,i}, C_{2,i}, C_{3,i}\}_{j \in [1,p]}).$$

When the access policy is uploaded in the form of the plaintext, it may leak some sensitive information about the data user. If we can remove the ρ, the whole attributes will be hidden in the anonymous access control structure. When we decrypt the data, we reconstruct mapping function ρ'.

$ABFBuild((M, \rho)) \rightarrow ABF$: The data owner inputs the access policy (M, ρ). In order to lower false positive property in BF. We use a garbled BF [42] as the building block of our attribute localization algorithm (ABF), and a specific string [38] as the element of the garbled BF in order to precisely locate attributes to the corresponding row number in the access matrix. Let L_{rn}-bit be the bit number of the row number. Let L_{at}-bit be the bit number of the attribute. Then let two fixed length strings be concatenated, where $L_{rn} + L_{at} = \lambda$. If the length of the row number i and the attribute at_e is less than the length of the maximum bit length, zeros will be filled on left of the bit strings, and the set of elements $S_e = \{i||atte\}_{i \in [1,l]}$ will be get, which is such a relationship $att_e = \rho(i)$. When we input the set of elements S_e, we can construct the ABF by calling the garbled BF Building algorithm.

When an element e in the set S_e will be added to the ABF, this algorithm first randomly generates k-1 λ-bit strings $r_{1,e}, r_{2,e}, \ldots, r_{k-1,e}$, then sets

$$r_{k,e} = r_{1,e} \oplus r_{2,e} \ldots \oplus e.$$

In this way, this algorithm shares the element e with(k, k) secret sharing scheme.

Next, this algorithm calculates hash function values of the attribute att_e that is related with the element e by calling k independent and identically distributed hash functions $H_1(), \ldots, H_k()$ and obtains

$$H_1(att_e), H_2(att_e), \ldots, H_k(att_e).$$

where the corresponding position index of ABF is expressed each $H_i(att_e)(i \in [1, k])$. Then, this algorithm then stores r_i that is the i_{th} element share to the position of the ABF as

$$r_{1,e} \rightarrow H_1(att_e) \text{ position in } ABF.$$
$$\ldots$$
$$r_{k,e} \rightarrow H_k(att_e) \text{ position in } ABF.$$

After that, the ABF will be store in the sensors or smartphones. Finally, the sensors or smartphones will upload the data (CT,M,ABF) to cloud servers.

D. DATA DECRYPTION

In our scheme, because of hiding the attributes mapping function ρ, the data users should first run the ABF query subroutine to check whether attributes they owned are in the access matrix. The algorithm is as follows:

$ABFQuery(S, ABF, PK) \rightarrow \rho'$: The data users input the attribute set S of users, the ABF and the system PK. For each attribute att owned by the data user in attribute set S, the algorithm computes the k hash functions $H_1(), \ldots, H_k()$ and obtains

$$H_1(att), H_2(att), \ldots, H_k(att).$$

Then, the algorithm gets the corresponding strings from the positions as follows:

$$H_i(att) \text{ position in ABF} \rightarrow r_{i,e} \text{ for } i \in \{1, \ldots, k\}$$

Next, the element e will be reconstructed as

$$\begin{aligned} e &= r_{1,e} \oplus r_{2,e} \oplus \ldots \oplus r_{k-1,e} \oplus r_{k,e} \\ &= r_{1,e} \oplus r_{2,e} \oplus \ldots \oplus r_{k-1,e} \oplus r_{1,e} \\ &\quad \oplus r_{2,e} \oplus \ldots \oplus r_{k-1,e} \oplus e. \end{aligned}$$

When the algorithm wants to reconstruct the attribute mapping, it should first judge whether the attributes owned by him are in the access matrix. In this process, in order to get the string att_e, the algorithm first fetches the last L_{att} bits from the string e. Next, it removes all the zero bits on the left of the string. If the attribute at is the same as att_e, the algorithm thinks that this attribute at is in the access matrix. Otherwise, his inquiry is terminated. Then, the algorithm should judge whether the attributes owned by him are in the access policy. In this process, in order to get the corresponding row number, it gets the first L_{rn} bits from the string e. Next, the algorithm removes all the zero bits at the left. If the attribute at is not the same as att_e, which means that the attribute at does not belong to the access policy.

At last, the algorithm outputs the reconstructed attribute mapping as

$$\rho' = \{(rownum, att)\}_{att \in S}.$$

which shows the corresponding relationship between row number and the access matrix M. When obtaining the access policy(M, ρ), the process of decryption is the same as that of the traditional attribute-based encryption systems.

$Dec(SK, CT, (M, \rho')) \rightarrow m$ or $\perp$: The data user inputs the SK, the ciphertext CT and the access matrix (M, ρ'). If S not satisfies M, the algorithm outputs $\perp$, Otherwise, this algorithm calculates

$$key = \frac{e(C_2, K)}{\prod_{i \in I}(e(K_0, C_{1,i} \cdot g^{a \cdot C_{3,i}})e(K_i, C_{2,i}))^{\omega^i}} = e(g, g)^{\alpha s}.$$

Next, the returns the message

$$m = \frac{C_1}{e(g, g)^{\alpha s}}.$$

V. SECURITY ANALYSIS

The security proof of the scheme will be given in this section. For the sake of convenience, we simplify our scheme to *OO-CPA* scheme, and denote $\prod_{OO} = (Setup, KeyGen, Encryption, Decryption)$. In order to prove the security of our scheme, our scheme can be reduced to the Waters system. The Waters system that is denoted $\prod_W = (SetupW, KeyGenW, EncryptionW, DecryptionW)$ is reduced to a "q-BDHE" assumption in prime order groups.

Theorem 1: The above *OO-CPA* scheme is selectively *CPA*-secure assuming that the scheme of the Rouselakis and Waters [27] is a selectively *CPA*-secure *KP-ABE* system.

Proof: The *OO-CPA* scheme is constructed from the access scheme with privacy preserving scheme in [38], which is constructed from the ciphertext-policy attribute-based encryption scheme in [40]. So, it is shown that if any PPT adversary A with non-negligible advantage break through the $\prod_{OO}$-Exp experiment, he can also break the decision q-BDHE problem with non-negligible advantage.

The simulator B plays a challenger role who interacts with A in $\prod_{OO}$-Exp scheme with security parameter λ.

- Initialization. A gives a challenge access control structure M^* to the B, B will send it to the Water challenge.
- Setup. Water challenge receives a M^*, and returns $PK =< g, g^a, e(g, g)^\alpha, h_1, \ldots, h_U >$ to B. Next, B chooses the parameters of the bloom filter. The maximum bit length of attributes in the system is expressed by L_{at}. The maximum bit length of the row numbers of access matrix is expressed by L_{rn}. The size of bit array of the ABF is expressed by L_{ABF}. The number of hash functions associated with the ABF is expressed by k. Then passes $PK =< g, g^a, e(g, g)^\alpha, L_{at}, L_{rn}, L_{ABF}, H_1(), H_2(), \ldots, H_k() >$ them to A.
- Phase 1. The secret key algorithm is the same as in the two scenario. So A can get the secret key by B who as a middleman.
- Challenge. B selects two distinct and random messages m_0, m_1 with equal length in the Water message space, and sends the messages to Water. Then, the Water challenger returns ciphertext $CT = (C, C', \{C_i, D_i\}_{i \in [1,\ldots,l]})$ to B. Here, C is the encrypted message times $e(g, g)^{\alpha s}$,$C' = g^s$ and for each attribute, we have $C_i = g^{a\lambda_i} h_i^{-r_i}$, $D_i = g^{r_i}$.
  It chooses random values $z_1, \ldots, z_{|S|} \in Z_p$ and computes the ciphertext

$$\begin{aligned} C_{1,i} &= C_i * g^{-az_i}, C_{2,i} = D_i, C_{3,i} = z_i, \\ CT^* &= (C', \{C_{1,i}, C_{2,i}, C_{3,i}\}_{i \in [1,\ldots,l]}). \end{aligned}$$

  After that, B guesses which message was encrypted $b \in \{0, 1\}$ and computes $key_g = C/m_b$. Finally, B sends to A the tuple (key_g, CT^*).
- Phase 2. B proceeds as in Phase 1.
- Guess. A outputs a bit t. If $t_a = 0$,which means that adversary A guesses that key_g is the key encapsulated

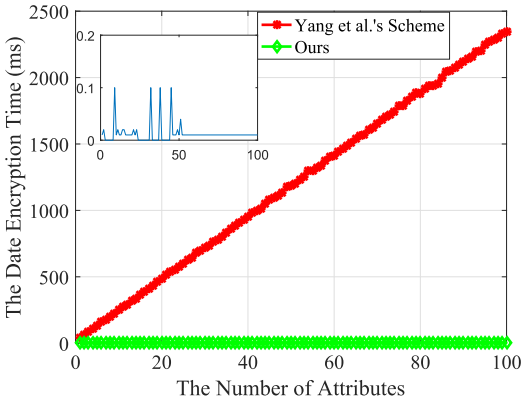

FIGURE 2. The performance comparison of the encryption phase.

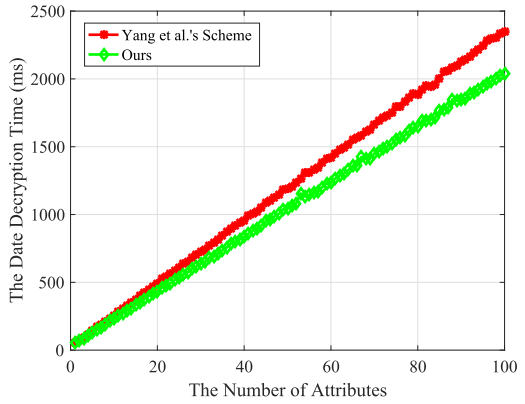

FIGURE 3. The performance comparison of the decryption phase.

by CT^*, then B outputs t_b. If $t_b = 1$, which means that A guesses that key_g is a random key, then B outputs $1 - t_b$. If the adversary A has advantage ϵ in our scheme, simulator B breaks the Kan *KP-ABE* system with the advantage ϵ.

VI. PERFORMANCE ANALYSIS

The analysis of the performance of our scheme will be given in this section. In our scheme, the encryption algorithm is composed of *offline.encryption*, *online.encryption* and *ABFBuild* algorithm. The decryption algorithm is composed of *ABFQuery* and *decryption* algorithms. In order to more accurately measure the scheme performance involved, we are testing in a unified environment, where the processor is the Intel(R) Core(TM) i5-3320M, and the RAM is the 8G.

Now, we give the computation cost of the encryption process and decryption process between Yang's scheme [38] and our scheme. Since the construction and inquiry of ABF is the same, the comparison of ABF is omitted. Figure. 2 displays the performance comparison of the encryption phase. Our scheme generates ciphertext faster that the Yang's scheme in the process of encryption. Because pre-encryption technology is applied, a lot of work is done ahead of time, and the results are stored on sensors and smart phones. When we know the information to be encrypted, we can quickly generate ciphertext. This can be applied well in the case of limited computing power. Figure. 3 displays the performance comparison of the decryption phase. Our scheme has a slight advantage in the performance of decryption. The performance analysis shows that our scheme does not increase the amount of computation in the encryption and decryption stages. However, in the encryption phase, the efficiency of encryption does not increase linearly with the increase of attributes, which can be well used in computing resource constrained Internet of things.

VII. CONCLUSION

In this article, the effective medical data sharing scheme in the cloud storage is presented. In our security system, we remove the attribute matching function, where attributes will be hidden into the anonymous access structure. More precisely, the ABF is used in our scheme to hide the entire attributes. In the decryption phase, the legitimate user will be able to restructure the attribute mapping function and decrypt the ciphertext. To generate ciphertext more quickly, the online/offline encryption technology is used in the encryption phase. In the offline stage, we do not know the information to be encrypted. But, we do a lot of calculation work that is needed at the encryption stage, then store them on sensors and smartphones. When we knew the plaintext of encryption, we could quickly make ciphertext. In addition, when the overall attributes of the system users increase, the system does not need to be reinitialized, which will also improve the efficiency of the system. That our scheme is secure is showed in the security analysis. And, the performance analysis shows that our scheme can improve data processing ability in the encryption stage. Therefore, the users' data privacy stored on a cloud server can be protected by using our scheme. Our scheme also improves data processing ability in the encryption stage, which can be well applied to terminal devices of IoT with limited computing power. In the next work, we will verify whether the ciphertext stored on the cloud server is tampered with. If it is tampered, we can find it in time.

REFERENCES

[1] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed Internet of Things," *Comput. Netw.*, vol. 57, no. 10, pp. 2266–2279, 2013.

[2] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2007, pp. 456–465.

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

[4] X. Chen, J. Li, J. Weng, J. Ma, and W. Lou, "Verifiable computation over large database with incremental updates," *IEEE Trans. Comput.*, vol. 65, no. 10, pp. 3184–3195, Oct. 2016.

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

[6] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. Acm Conf. Comput. Commun. Secur. (CCS)*, 2007, pp. 195–203.

[7] X. Wang, J. Zhang, E. M. Schooler, and M. Ion, "Performance evaluation of attribute-based encryption: Toward data privacy in the IoT," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2014, pp. 725–730.

[8] Y. Li, G. Wang, L. Nie, Q. Wang, and W. Tan, "Distance metric optimization driven convolutional neural network for age invariant face recognition," *Pattern Recognit.*, vol. 75, pp. 51–68-2, Mar. 2018, doi: 10.1016/j.patcog.2017.10.015.

[9] R. F. Nogueira, R. de A. Lotufo, and R. C. Machado, "Fingerprint liveness detection using convolutional neural networks," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1206–1213, Jun. 2016.

[10] A. Lounis, A. Hadjidj, A. Bouabdallah, and Y. Challal, "Healing on the cloud: Secure cloud architecture for medical wireless sensor networks," *Future Generat. Comput. Syst.*, vol. 55, pp. 266–277, Feb. 2016.

[11] Z. Cai, H. Yan, P. Li, Z.-A. Huang, and C. Gao, "Towards secure and flexible EHR sharing in mobile health cloud under static assumptions," *Cluster Comput.*, vol. 20, no. 3, pp. 1–8, 2017.

[12] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, "Differentially private Naive Bayes learning over multiple data sources," *Inf. Sci.*, vol. 444, pp. 89–104, May 2018.

[13] J. B. Bernabe, J. L. Hernández, M. V. Moreno, and A. F. S. Gomez, *Privacy-Preserving Security Framework for a Social-Aware Internet of Things*. London, U.K.: Springer, 2014, pp. 408–415.

[14] V. Odelu, A. K. Das, Y. S. Rao, S. Kumari, M. K. Khan, and K. K. R. Choo, "Pairing-based CP-ABE with constant-size ciphertexts and secret keys for cloud environment," *Comput. Standard Interfaces*, vol. 54, pp. 3–9, Nov. 2016.

[15] A. Xiong and C. Xu, "Cloud storage access control scheme of ciphertext algorithm based on digital envelope," *Intell. Automat. Soft Comput.*, vol. 22, no. 3, pp. 289–294, 2015.

[16] Z. Liu, Y. Huang, J. Li, X. Cheng, and C. Shen, "DivORAM: Towards a practical oblivious RAM with variable block size," *Inf. Sci.*, vol. 447, pp. 1–11, Jun. 2018.

[17] Q. Lin, H. Yan, Z. Huang, W. Chen, J. Shen, and Y. Tang, "An ID-based linearly homomorphic signature scheme and its application in blockchain," *IEEE Access*, vol. 6, pp. 20632–20640, Feb. 2018.

[18] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2018.2825289.

[19] X. Chen, J. Li, X. Huang, J. Ma, and W. Lou, "New publicly verifiable databases with efficient updates," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 5, pp. 546–556, Sep. 2015.

[20] Y. Zhang, J. Li, X. Chen, and H. Li, "Anonymous attribute-based proxy re-encryption for access control in cloud computing," *Secur. Commun. Netw.*, vol. 9, no. 14, pp. 2397–2411, 2016.

[21] P. Li *et al.*, "Multi-key privacy-preserving deep learning in cloud computing," *Future Generat. Comput. Syst.*, vol. 74, pp. 76–85, Sep. 2017.

[22] Z. Huang, S. Liu, X. Mao, K. Chen, and J. Li, "Insight of the protection for data security under selective opening attacks," *Inf. Sci.*, vols. 412–413, pp. 223–241, Oct. 2017.

[23] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Inf. Sci.*, vol. 379, pp. 42–61, Feb. 2017.

[24] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2005, pp. 457–473.

[25] A. Lewko and B. Waters, "Unbounded HIBE and attribute-based encryption," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn., Adv. Cryptol.*, 2011, pp. 547–567.

[26] A. Lewko, "Tools for simulating features of composite order bilinear groups in the prime order setting," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2012, pp. 318–335.

[27] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 463–474.

[28] J. Li, X. Chen, S. S. M. Chow, Q. Huang, D. S. Wong, and Z. Liu, "Multi-authority fine-grained access control with accountability and its application in cloud," *J. Netw. Comput. Appl.*, vol. 112, pp. 89–96, Jun. 2018.

[29] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New algorithms for secure outsourcing of modular exponentiations," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2386–2396, Sep. 2012.

[30] B. Li, Y. Huang, Z. Liu, J. Li, Z. Tian, and S.-M. Yiu, "HybridORAM: Practical oblivious cloud storage with constant bandwidth," *Inf. Sci.*, to be published, doi: 10.1016/j.ins.2018.02.019.

[31] J. Xu, L. Wei, Y. Zhang, A. Wang, F. Zhou, and C.-Z. Gao, "Dynamic fully homomorphic encryption-based merkle tree for lightweight streaming authenticated data structures," *J. Netw. Comput. Appl.*, vol. 107, pp. 113–124, Apr. 2018.

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

[33] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *Proc. Int. Workshop Public Key Cryptogr.*, vol. 8383, 2014, pp. 293–310.

[34] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, 2008, pp. 111–129.

[35] J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo, "Anonymous and traceable group data sharing in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 912–925, Apr. 2018.

[36] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proc. Theory Cryptogr. Conf.*, 2007, pp. 535–554.

[37] J. Lai, R. H. Deng, and Y. Li, "Fully secure cipertext-policy hiding CP-ABE," in *Proc. Int. Conf. Inf. Secur. Pract. Exper.*, 2011, pp. 24–39.

[38] K. Yang, Q. Han, H. Li, K. Zheng, Z. Su, and X. Shen, "An efficient and fine-grained big data access control scheme with privacy-preserving policy," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 563–571, Apr. 2017.

[39] J. Lai, R. H. Deng, and Y. Li, "Expressive CP-ABE with partially hidden access structures," in *Proc. ACM Symp. Inf., Comput. Commun. Secur.*, 2012, pp. 18–19.

[40] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography* (Lecture Notes in Computer Science). 2011, pp. 53–70.

[41] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, Jul. 1970.

[42] C. Dong, L. Chen, and Z. Wen, "When private set intersection meets big data: An efficient and scalable protocol," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 789–800.

DONG ZHENG received the Ph.D. degree in communication engineering from Xidian University, China, in 1999. He is currently a Professor with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications. He has published over 100 research articles, including CT-RSA, and the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS. His research interests include cloud security and wireless network security.

AXIN WU received the B.S. degree from the Zhengzhou University of Light Industry in 2016. He is currently pursuing the M.Eng. degree with the Xi'an University of Post and Telecommunications, Xi'an, China. His research interests include cloud security.

YINGHUI ZHANG (M'18) received the B.S. degree from Nanchang Hangkong University in 2007 and the M.S. degree in mathematics and the Ph.D. degree in cryptography from Xidian University, China, in 2010 and 2013, respectively. He is currently an Associate Professor with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications. He has published over 50 research articles, including ASIACCS, ACISP, IEEE CSE, IEEE DSC, Computer Networks, and Computers & Security. His research interests include cloud security and wireless network security.

QINGLAN ZHAO received the B.S. degree from Shaanxi Normal University in 1999 and the M.S. degree from Northwestern Polytechnical University in 2006. She is currently pursuing the Ph.D. degree with Shanghai Jiao Tong University, China. Since 2014, she has been an Associate Professor with the Xi'an University of Post and Telecommunications, Xi'an, China. Her research interests focus on cryptographic functions and symmetric cryptography.