IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 5, 2019, accepted December 22, 2019, date of publication December 27, 2019, date of current version January 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962729

A Secure and Lightweight Data Sharing Scheme for Internet of Medical Things

XIUQING LU[1,2] AND XIANGGUO CHENG[2]
[1]College of Business, Qingdao University, Qingdao 266071, China
[2]College of Computer Science and Technology, Qingdao University, Qingdao 266071, China

Corresponding author: Xiuqing Lu (luxiuqing@qdu.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61772294 and Grant 61602214, and in part by the Natural Science Foundation of Shandong Province under Grant ZR2019MF058.

ABSTRACT As cloud computing has many advantages such as large storage capacity, low cost and scalability, more and more patients prefer to store their health data in cloud to share with physicians, researchers or other users. However, storing shared data in remote cloud is out of patient's control and exposes to lots of security problems such as privacy and data integrity. So far, more and more data sharing schemes to preserve data security in health field have been put forward, but in most of them, data encryption and decryption are completely implemented by terminal devices, which increases the communication and computation burden of patient and user. Furthermore, most sharing schemes have no integrity verification mechanism, resulting in incomplete data for users to share. To solve the problems, we propose a secure and lightweight data sharing scheme for Internet of Medical Things. Firstly, the scheme guarantees the privacy and authorized access of shared data. Secondly, the scheme realizes efficient integrity verification before user downloads shared data to avoid incorrect query or computation result. Finally, the scheme achieves lightweight operations of patient and user.

INDEX TERMS Authorized, cloud computing, integrity, privacy.

I. INTRODUCTION

With the rapid development of information technology, Internet of Medical Things (IoMT) has been widely applied in the field of health care [1]–[5]. IOMT can not only bring conveniences to patients such as telemedicine anywhere, but also help medical professionals realize intelligent medical treatments like predicting disease for patients. However, with continuous increase of health data and medical applications, the health information system is faced with challenges of how to efficiently store, retrieve and deal with the big health data. [6], [7]. Cloud computing [8]–[10] is a suitable platform with large storage and computation resources that can support big data applications [11]. Nowadays, more and more patients prefer to upload their personal health data to cloud for disease diagnosis or prediction by medical experts. Outsourcing health data to cloud not only saves the local storage space of the health information system, but also greatly reduces the investment cost in software and hardware maintenance of medical enterprises [12]. However, storing sensitive health data in cloud can also bring some security and privacy issues [13]–[19].

Firstly, the health data not only relates to personal identity information of the patient, but also involves health information such as infectious diseases and so on. The leakage of sensitive data is no doubt harmful to patient's life and work, so it is imperative to ensure the privacy of health data. Secondly, cloud storage servers expose to hardware or software failures, and subject to malicious internal or external attacks. Therefore, it is extremely important to ensure the integrity of shared health data stored in cloud storage servers [20]–[21]. Thirdly, any unauthorized users should not access the shared health data. Once unauthorized users access and tamper with medical records, it will lead to serious results such as misdiagnosis [22]. Consequently, it is important to ensure privacy, integrity and authorization of health data. In addition, the Internet-of-Things terminal are usually resource-constrained devices with small storage space and low processing speed. Therefore, it is essential to propose a secure and lightweight data-sharing scheme for IoMT.

A. MAIN CONTRIBUTIONS

In order to improve the computation efficiency of terminal devices in IoMT and guarantee the security and privacy of shared data, we construct a secure and lightweight data

The associate editor coordinating the review of this manuscript and approving it for publication was Asad Waqar Malik.

sharing scheme for Internet of Medical Things. The main contributions of the paper are as follow.

1) The scheme guarantees the privacy of patient and authorized access of shared data based on identity-based broadcast encryption.
2) The scheme achieves efficient integrity verification before user downloads shared data to avoid incorrect computation.
3) We prove the security of the sharing scheme and evaluate the computation and communication cost of patient and user side. The results indicate that our scheme is more efficient than the previous ones.

B. ORGANIZATION

The organization of the rest paper is as follows. We first introduce the related works in Section II. Then we describe system model, security requirements and design goals in Section III. We present the preliminaries in Section IV and the constructions of data sharing scheme for IoMT in section V. Then we analyze security of the scheme in section VI and performance of the scheme in Section VII. Finally, we conclude this paper in Section VIII.

II. THE RELATED WORKS

So far, many data sharing schemes have been put forward in medical health field. The security of them mainly focus on data integrity, privacy and access control, which are core security problems in cloud data sharing.

Cloud data auditing is a technology for user to verify the availability of remote data. So far, many auditing schemes [23]–[39] have been proposed to verify the integrity of data stored on remote servers. Ateniese *et al.* [23] presented the first public auditing scheme in which provable data possession (PDP) is proposed. To prove the integrity of dynamic data, Ateniese *et al.* [24] presented another scheme based on the symmetric key PDP scheme. The scheme supports dynamic modification and deletion operations, but does not support insertion operation. To achieve dynamic operation, Erway *et al.* [25] raised a dynamic provable data possession (DPDP) scheme by introducing an authenticated skip list. Zhu *et al.* [26] introduced an index-hash table for dynamic verification. Later Yang [27] proposed a data structure named Dynamic-Hash-Table. Wang *et al.* [28] and Liu *et al.* [29] proposed dynamic public auditing schemes based on Merkle Hash Tree (MHT). To protect data privacy, Wang *et al.* [30] put forward an integrity verification scheme by employing a random masking technique. Wang *et al.* [31] designed an auditing scheme with ring signature to achieve secure cloud storage. Yang and Yu [32] also proposed an integrity verification scheme supporting the identity privacy.

To achieve privacy to cloud servers and access control to users, identity-based broadcast encryption (IBBE) is involved in many schemes. IBBE is a specific case of identity-base encryption (IBE), in which the user's public key can be any arbitrary strings such as user's email. In 1984, Shamir [40] proposed the first IBE scheme. Later the bilinear pairing made IBE more efficient because it avoids certificate management. In 2001, Boneh and FrankliN [41] proposed an identity-based encryption scheme from the Weil Pairing. Yoon *et al.* [42] proposed an IDB signature scheme with message recovery. In 2007, Delerablee and Cécile [43] proposed the first IBBE scheme with constant size cipher texts and private keys. Later, Gentry and Waters [44] proposed the first adaptively CPA-secure IBBE scheme, which presents the first adaptively secure system with sublinear cipher-texts and proves security in the standard model. In 2015, Kim and Susilo [45] presented another adaptively secure identity-based broadcast encryption system featuring constant sized cipher-text in the standard model. Since then, many other IBBE schemes [46]–[48] are proposed in diverse fields and applications.

III. SYSTEM MODEL, SECURITY REQUIREMENT AND DESIGN GOALS

In our secure data sharing scheme for IoMT, patient with health sensor devices collects and encrypts his health data before uploading it to cloud servers for sharing. In addition, patient designates the identity set of user for achieving the authorized access. In our scheme, the identity can be any string that can represent user's attributes such as work number of doctor. To ensure cloud data intact before sharing and decrease computation burden of patient, an entity named Security-Mediator (SEM) help patient generate blocks and block tags for later integrity verification. A SEM can be a server within a certain area, such as a community health server. If a user wants to access the health data, he must register his identity to Trusted Authority and gets the warrant to limit his access time. Only when user's identity and valid access time are valid, the user can download and decrypt shared data.

A. SYSTEM MODEL

Fig. 1 shows the system model of secure health data sharing for IOMT, which consists of four entities, namely Trusted Authority (TA), patient, Cloud servers (CS) and users.

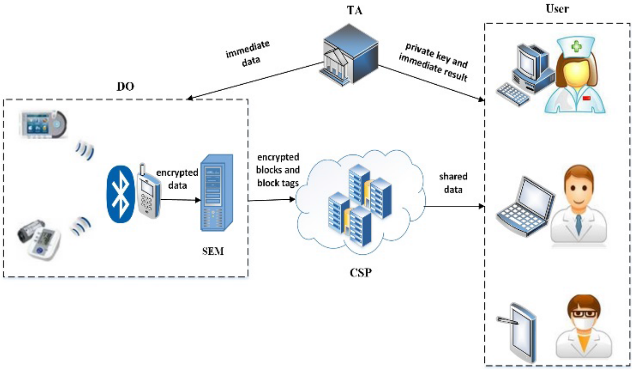

TA
User
immediate data
private key and immediate result
DO
encrypted data
encrypted blocks and block tags
shared data
SEM
CSP

FIGURE 1. System Model.

Trusted Authority (TA): It is trusted by other entities. It is responsible to generate public and private parameters of the system and issues private keys for users according to his identity.

Patient: It refers to entity with sensor devices to gather health data such as temperature and blood pressure, etc. Patient owns his health data and prefer to upload it to CS for data sharing with physicians, nurses or other authorized users. He is responsible to encrypt his health data for privacy and establish authority for user to access his data. To save patient's computation burden, a Security-Mediator (SEM) is introduced to help patient divide encrypted data into blocks and compute block tags for user's later data integrity verification.

Cloud server (CS): It is the entity with large storage and computation resources to maintain and manipulate shared data and can provide data access to legitimate user. CS is managed by CSP (Cloud Server Provider).

User: The entity refers to medical professionals, nurses or medical researchers to utilize shared health data for medical diagnosis and data mining. In the scheme, only the authorized user is able to download shared data from CS and decrypt the data.

B. SECURITY REQUIREMENT

In our sharing scheme, we assume that SEM is semi-trusted. Though it can help patient divide data into blocks and compute block tags, it might be curious about sensitive health data of patient. Therefore, the shared data must keep secret to SEM. Similarly, we suppose CS is also semi-trusted. CS is responsible to store data and block tags in data sharing, but once data is corrupt or lost, it might launch forge attack or replace attack for economic reasons. Furtherly, CS may also be curious about the content of sensitive data, so the data should preserve secret to CS. After patient transferring his data to CS, only the authorized user is able to download and access the plain text. In the scheme, we assume TA is a fully trusted authority and can honestly generate private key for each user. Therefore, the following security requirements of the scheme should be satisfied.

Privacy preserving: The shared data must keep confidential to SEM, CS and any unauthorized users to keep patient's health data secure. The health data involves not only personal identity information, but also medical information such as infectious disease, so any disclosure of health information is undoubtedly harmful to patient's life and work. Consequently, it is imperative to ensure the privacy of patient's health data.

Authorized access: It means only legitimate user designated by patient himself can download and access the health data stored in cloud. Furtherly, the authorized user can only download the data within the definite time limit.

Data Integrity: It ensures that health data not be modified or deleted during transmission and storage process. In the scheme, user can detect any malicious tamper operations of shared data before downloading the data.

C. DESIGN GOALS

Based on the system model and security requirements, our data sharing scheme for IoMT is designed to achieve the following goals.

Security requirements: The scheme should satisfy the security requirements including data privacy, authorized access and data integrity during data sharing process.

Lightweight operations: To improve efficiency of data sharing, the scheme should decrease computation operations of patient and user because the terminals on both sides are mostly mobiles devices. In our scheme, SEM divides encrypted data into blocks and computes block tags instead of patient. Furtherly, before patients encrypts data, TA calculates the intermediate data of encryption to decrease patient's computation overhead. Similarly, when user wants to access shared data, TA help him compute intermediate data of decryption to less user's computation burden.

Effectiveness: The scheme should effectively achieve one-to-many data sharing, allowing patient securely share his data and any authorized user correctly access the data.

IV. PRELIMINARIES

A. NOTATIONS

The notations in this paper are described in Table 1.

TABLE 1. Main notations in the scheme.

Notation	Meaning	Notation	Meaning
$\mathbb{G}_1, \mathbb{G}_2$	multiplicative group	M	health data
e	bilinear map	M'	encrypted data
f_1, f_2	pseudo-random functions	m_i	encrypted blocks
q	prime order of group	T	block tags
g, h	generator of $\mathbb{G}_1$	P	integrity proof
sk	secret key of SEM	K	key of encryption
pk	public key of SEM	Uid	user identity
H_1, H_2, H_3	secure hash function	Pid	patient identity
$Params$	system public parameters	sk_{Uid}	user's private key
Mk	system master key	$warr$	warrant of user

B. BILINEAR MAPS

Suppose $\mathbb{G}_1, \mathbb{G}_2$ are two multiplicative groups with same large prime order q, and g is a generator in $\mathbb{G}_1$. A bilinear map e is a map function $e{:}\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_1$ with the following properties: i) Computability. $\forall u, v \in \mathbb{G}_1$, an efficient algorithm exists to compute $e(u, v)$. ii) Binearity. $\forall a, b \in Z_q, \exists e\left(u^a, v^b\right) = e\,(u, v)^{ab}$. iii) Nondegeneracy. $e\,[g, g] \neq 1$. iv) Security. It is hard to compute Discrete Logarithm (DL) in $\mathbb{G}_1$.

C. DEFINITION

Our secure data sharing scheme for IoMT includes the following polynomial algorithms.

1) *Setup* $(\lambda, y) \rightarrow (Params, Mk)$. It is run by TA. It takes security parameter λ as input and outputs system public parameter *Params* and master key *Mk* of the scheme.

2) $KeyExtract\left(Params, Mk, Uid_j\right) \rightarrow (sk_{UID})$. It is run by TA. Given $Params$, Mk and user identity $Uid_j \in \{0.1\}^*$, it generates the private key sk_{ID} for user.
3) $PatientReg\,(Pid, S) \rightarrow \phi$. It is run by TA. Given patient identity Pid and user identity set S, the algorithm outputs ϕ as the intermediate result for data encryption.
4) $DataEnc\,(M) \rightarrow M^{'}$. It is run by patient and it encrypts sensitive data M to $M^{'}$.
5) $TagGen\left(M^{'}, x\right) \rightarrow T$. It is run by SEM. It takes $M^{'}$ and SEM's private key x as input and outputs block tags T.
6) $ChalGen\,(Pid) \rightarrow chal$. It is run by user. It takes patient identity Pid and outputs challenge information $chal$.
7) $ProfGen\left(M^{'}, T, chal, pk\right) \rightarrow P$. It is run by CS and generates integrity proof P.
8) $ProfVer\,(P, chal, pk) \rightarrow (\text{"}true\text{"}, \text{"}false\text{"})$. It is run by user. It takes P, $chal$ and pk as input and outputs the verification result "$true$" or "$false$".
9) $PreCompute\left(Uid_j, Pid\right) \rightarrow \langle \Delta_{\gamma}(Uid_j, S), \delta \rangle$. It is run by TA and outputs intermediate decryption result $\langle \Delta_{\gamma}(Uid_j, S), \delta \rangle$ for user.
10) $DataDecry\left(M^{'}, sk_{Uid}\right) \rightarrow M$.It is run by user and decrypts $M^{'}$ to M with user's private key sk_{Uid}.

V. CONSTRUCTIONS OF SECURE DATA SHARING SCHEME

In this section, we present the secure sharing scheme for IoMT in detail. We divide the sharing scheme into three phases named initial phase, preprocessing phase and data sharing phase.

A. INITIAL PHASE

In this phase, TA generates public system parameter and master key. Because each user in the scheme must register his identity Uid_j to TA and get his private key before downing shared data, TA is also responsible to generate private key and warrant for each user. Similarly, the patient should register his identity in TA before sharing his data with other users. This phase consists of the following three algorithms and fig. 2 illustrates the flowchart of the phase.

Setup. Given security parameter λ and integer y, TA constructs the bilinear map group system $\Theta = \langle \mathbb{G}_1 \mathbb{G}_2, q, e \rangle$ where $\mathbb{G}_1$,$\mathbb{G}_2$ are multiplicative groups with order q, and e is a bilinear map e:$\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. TA also selects two random generators g, $h \in \mathbb{G}_1$ and picks three secure cryptographic hash functions: H_1: $\{0, 1\}^* \rightarrow Z_q^*$, H_2:$\mathbb{G}_2 \rightarrow \{0, 1\}^l$, H_3: $\{0, 1\}^* \rightarrow \mathbb{G}_1$. Then TA picks a random $\gamma \in \mathbb{Z}_q^*$ and computes $w = g^{\gamma}$, $v = e\,(g, h)$. TA keeps master key $Mk = \langle g, \gamma \rangle$ secretly and publishes public system parameter $Params = \langle \Theta H_1, H_2, H_3, h, h^{\gamma}, \ldots, h^{\gamma^y} \rangle$.

KeyExtract. After receiving identity $Uid_j \in \{0, 1\}^*$ from user, TA extracts the private key $sk_{Uid} = g^{\frac{1}{\gamma + H_1(Uid_j)}}$ for him. Next TA picks random a_1,$a_2 \in \mathbb{Z}_q^*$ and computes $b_1 = h^{a_1}$,$b_2 = h^{a_2}$. Then the warrant of user is $warr = a_1 + a_2 \cdot H_1\left(Uid_j \| \; time\right)$, where $time$ refers to the valid time for user to access shared data. Finally, TA sends sk_{Uid} to user via a secure channel and $\langle Uid_j, warr, b_1, b_2 \rangle$ to CS.

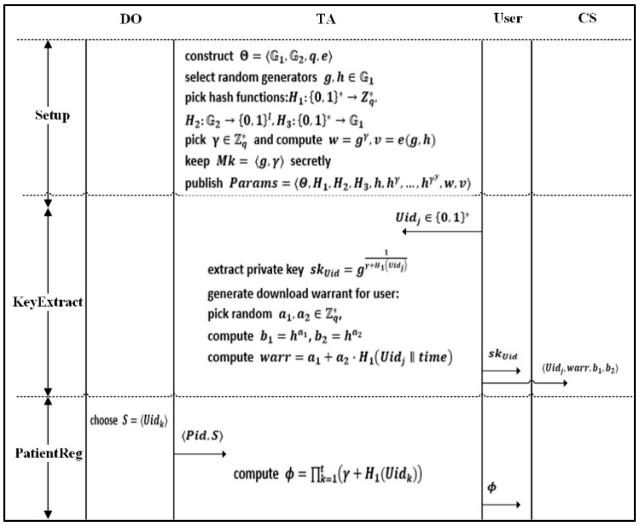

DO
TA
User
CS
Setup
construct $\Theta = \langle \mathbb{G}_1, \mathbb{G}_2, q, e \rangle$
select random generators $g, h \in \mathbb{G}_1$
pick hash functions:H_1:$\{0,1\}^* \rightarrow Z_q^*$,
H_2:$\mathbb{G}_2 \rightarrow \{0,1\}^l$, H_3:$\{0,1\}^* \rightarrow \mathbb{G}_1$
pick $\gamma \in \mathbb{Z}_q^*$ and compute $w = g^{\gamma}, v = e(g,h)$
keep $Mk = \langle g, \gamma \rangle$ secretly
publish $Params = \langle \theta, H_1, H_2, H_3, h, h^{\gamma}, \ldots, h^{\gamma^y}, w, v \rangle$
KeyExtract
$Uid_j \in \{0,1\}^*$
extract private key $sk_{Uid} = g^{\frac{1}{\gamma + H_1(Uid_j)}}$
generate download warrant for user:
pick random $a_1, a_2 \in \mathbb{Z}_q^*$,
compute $b_1 = h^{a_1}, b_2 = h^{a_2}$
compute $warr = a_1 + a_2 \cdot H_1(Uid_j \parallel time)$
sk_{Uid}
$\langle Uid_j, warr, b_1, b_2 \rangle$
PatientReg
choose $S = \langle Uid_k \rangle$
$\langle Pid, S \rangle$
compute $\phi = \prod_{k=1}^{t}(\gamma + H_1(Uid_k))$
ϕ

FIGURE 2. Flowchart of Initial Phase.

PatientReg. Patient Pid first chooses $S = \langle Uid_k \rangle_{k=1}^{t}$, $t \leq y$ to denote user identity set to access his health data. Any user with $Uid_k \subseteq S$ can access shared data M in valid time. After receiving register information $\langle Pid, S \rangle$ from patient, TA computes $\phi = \prod_{k=1}^{t} (\gamma + H_1\,(Uid_k))$. Then TA transfers ϕ to patient secretly and keeps ϕ locally for later computation.

B. PRE-PROCESS PHASE OF SHARING DATA

In our scheme, suppose the max length of shared data is l. To preserve $M \in \{0, 1\}^l$ secret to others, patient first encrypts data M to $M^{'}$ and transfers $M^{'}$ to SEM. Then SEM divides $M^{'}$ into n blocks and gets block tags. This phase includes the following two algorithms and fig. 3 is the flowchart of the phase.

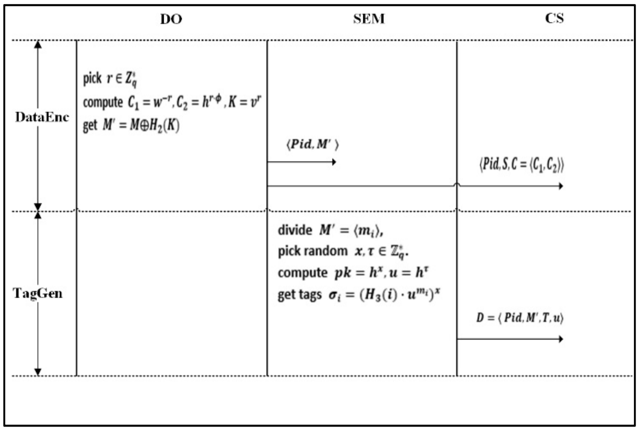

DO
SEM
CS
DataEnc
pick $r \in Z_q^*$
compute $C_1 = w^{-r}, C_2 = h^{r \cdot \phi}, K = v^r$
get $M' = M \oplus H_2(K)$
$\langle Pid, M' \rangle$
$\langle Pid, S, C = \langle C_1, C_2 \rangle \rangle$
TagGen
divide $M' = \langle m_i \rangle$,
pick random $x, \tau \in \mathbb{Z}_q^*$.
compute $pk = h^x, u = h^{\tau}$
get tags $\sigma_i = (H_3(i) \cdot u^{m_i})^x$
$D = \langle Pid, M', T, u \rangle$

FIGURE 3. Flowchart of Preprocess phase.

DataEnc. Patient Pid computes symmetric encryption key K and encrypts M with $H_2\,(K)$ as follows. He picks a random $r \in Z_q^*$ and computes $C_1 = w^{-r}, C_2 = h^{r \cdot \phi}, K = v^r$. Next patient encrypts M as $M^{'}$.

$$M^{'} = M \oplus H_2(K) \quad (1)$$

Finally, patient sends $\langle Pid, S, C = \langle C1, C_2 \rangle \rangle$ to CS, $\langle Pid, M' \rangle$ to SEM and $\langle Pid, S \rangle$ to TA.

TagGen. In order to ensure the integrity of shared data M, SEM computes tag for each block. He first divides $M^{'}$ into n data blocks, namely $M^{'} = \langle m_i \rangle$, with erasure code algorithm. Then he picks random $x, \tau \in \mathbb{Z}_q^*$ and computes $pk = h^x, u = h^\tau$. He denote x his private key and pk his public key. Finally, SEM gets block tags as follows.

$$\sigma_i = (H_3(i) \cdot u^{m_i})^x \quad (2)$$

SEM denotes $T = \langle \sigma_j \rangle$ and transfers $D = \langle Pid, M', T, u \rangle$ to CS.

C. DATA SHARING PHASE

When user wants to access shared data, he first verifies the integrity of data. He generates integrity challenge *chal* and sends *chal* to CS. If the user warrant is valid, CS computes data integrity proof P and send it to user. After user proves the shared data is intact, he downloads and decrypts $M^{'}$. This phase consists of the following five algorithms and fig. 4 describes the flowchart of the phase.

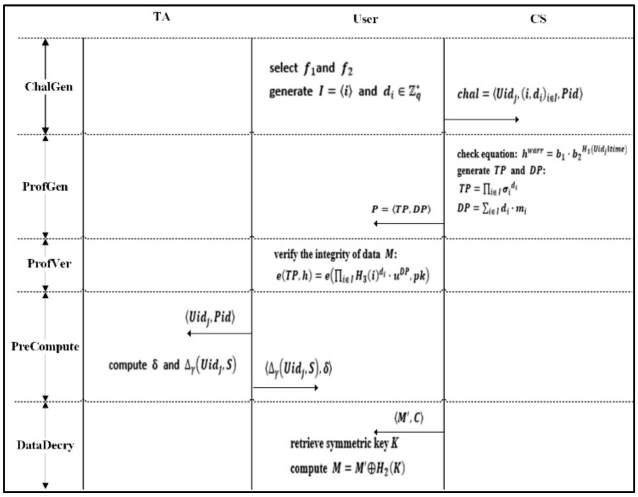

TA
User
CS
ChalGen
select $f_1$and f_2
generate $I = \langle i \rangle$ and $d_i \in \mathbb{Z}_q^*$
$chal = \langle Uid_j, (i, d_i)_{i \in I}, Pid \rangle$
ProfGen
check equation: $h^{warr} = b_1 \cdot b_2^{H_1(Uid_j \| time)}$
generate TP and DP:
$TP = \prod_{i \in I} \sigma_i^{d_i}$
$DP = \sum_{i \in I} d_i \cdot m_i$
$P = \langle TP, DP \rangle$
ProfVer
verify the integrity of data M:
$e(TP, h) = e\left(\prod_{i \in I} H_3(i)^{d_i} \cdot u^{DP}, pk\right)$
PreCompute
$\langle Uid_j, Pid \rangle$
compute δ and $\Delta_\gamma(Uid_j, S)$
$\langle \Delta_\gamma(Uid_j, S), \delta \rangle$
DataDecry
$\langle M', C \rangle$
retrieve symmetric key K
compute $M = M' \oplus H_2(K)$

FIGURE 4. Flowchart of data sharing phase.

ChalGen. Before downloading shared data M, user Uid_j first generates integrity challenge. He selects two pseudo-random functions named $f_1: \{1, 2, \cdots, n\} \rightarrow \{1, 2, \cdots, n\}$ and $f_2: \{1, 2, \cdots, n\} \rightarrow \mathbb{Z}_q^*$. Then he generates a subset $I = \langle i \rangle$ with c elements from $[1,n]$ by f_1 and corresponding random numbers $d_i \in \mathbb{Z}_q^*$ by f_2. Finally, he transfers challenge $chal = \langle Uid_j, (i, d_i)_{i \in I}, Pid \rangle$ to CS.

ProfGen. On receiving *chal* from user, CS first checks user warrant with the following equation:

$$h^{warr} = b_1 \cdot b_2^{H_1(Uid_j \| time)} \quad (3)$$

If eq. (3) holds, CS generates signature proof TP and data proof DP as follows.

$$TP = \prod_{i \in I} \sigma_i^{d_i} \quad (4)$$

$$DP = \sum\nolimits_{i \in I} d_i \cdot m_i \quad (5)$$

Then CS sends $P = \langle TP, DP \rangle$ to user.

ProfVer. On receiving proof P from CS, user verifies the integrity of data M as follows.

$$e(TP, h) = e\left(\prod\nolimits_{i \in I} H_3(i)^{d_i} \cdot u^{DP}, pk\right) \quad (6)$$

If eq. (6) holds, the algorithm outputs "*true*". Otherwise, it outputs “*false*″.

PreCompute. If shared data is intact, user sends $\langle Uid_j Pid \rangle$ to TA to get intermediate result of decryption. TA computes $\delta = \prod_{k=1, k \neq j}^{t} (H_1(Uid_k))$ and $\Delta_\gamma(Uid_j, S) = \gamma^{-1} \cdot \left(\phi \cdot (\gamma + H_1(Uid_j))^{-1} - \delta\right)$ based on $\langle Pid, S \rangle$ and transfers $\langle \Delta_\gamma(Uid_j, S), \delta \rangle$ to user secretly for data decryption.

DataDecry. User Uid_j downloads $M^{'}, C$ from CS and decrypts shared data. He first retrieves symmetric key K as follows.

$$K = \left(e\left(C_1, h^{\Delta_\gamma(Uid_j, S)}\right) \cdot e(sk_{Uid}, C_2)\right)^{\frac{1}{\delta}} \quad (7)$$

Then user computes $M = M^{'} \oplus H_2(K)$ to get plain text of shared data.

VI. SECURITY ANALYSIS

In this section, we analyze the security of the scheme, including correctness, unforgeability and privacy.

Theorem1: Authorized user can correctly verify the integrity of the data stored in CS.

Proof: Theorem 1 can be proved by verifying the correctness of eq. (5). The proof is as follows.

$$\begin{aligned} e(TP, h) &= e\left(\prod_{i \in I} \sigma_i^{d_i}, h\right) \\ &= e\left(\prod_{i \in I} (H_3(i) \cdot u^{m_i})^{x \cdot d_i}, h\right) \\ &= e\left(\prod_{i \in I} (H_3(i) \cdot u^{m_i})^{x \cdot d_i}, h\right) \\ &= e\left(\prod\nolimits_{i \in I} H_3(i)^{d_i} \cdot u^{DP}, h^x\right) \\ &= e\left(\prod\nolimits_{i \in I} H_3(i)^{d_i} \cdot u^{DP}, pk\right) \end{aligned}$$

From the proof of eq. (5), user can verify whether the data is undamaged stored in CS.

Theorem 2: Authorized user can correctly recover K if the identity ID_i is legitimate.

Proof: Theorem 2 can be proved by verifying the correctness of eq. (6). The proof is as follows.

$$
\begin{aligned}
&\left(e\left(C_1,h^{\Delta_\gamma(Uid_j,S)}\right)\cdot e\left(sk_{Uid},C_2\right)\right)^{\frac{1}{\delta}}\\
&=\left(e\left(g^{-r\cdot\gamma},h^{\Delta_\gamma(Uid_i,S)}\right)\right.\\
&\qquad\left.\cdot e\left(g^{\frac{1}{\gamma+H_1\left(Uid_j\right)}},h^{r\cdot\prod_{k=1}^{t}(\gamma+H_1(Uid_k))}\right)\right)^{\frac{1}{\delta}}\\
&=\left(e\left(g,h\right)^{-r\cdot\left(\prod_{k=1,k\neq j}^{t}(\gamma+H_1(Uid_k))-\delta\right)}\right.\\
&\qquad\left.\cdot e\left(g,h\right)^{r\cdot\prod_{k=1,k\neq j}^{t}(\gamma+H_1(Uid_k))}\right)^{\frac{1}{\delta}}\\
&=e\left(g,h\right)^{r\cdot\delta\cdot\frac{1}{\delta}}\\
&=v^r\\
&=K
\end{aligned}
$$

Theorem 3: As long as the DL assumption holds, it is computationally infeasible for unauthorized user, SEM and CS to get health data in the scheme.

Proof: In preprocess phase of shared file, patient encrypts file M to $M^{'}$, therefore the data is private to CS and SEM. In sharing phase, CS sends $P=\{TP,DP\}$ to user, where $DP=\sum_{i\in I}d_i\cdot m_i$. Because m_i are blocks of encrypted data $M^{'}$, unauthorized user cannot get any information on the sensitive data.

Theorem 4: It is computationally impossible for CS to forge an integrity proof to pass the public verification, if the Computational Diffie-Hellman (CDH) problem is hard in bilinear group.

Proof: In sharing phase, After CS receives the challenge *chal* from user, he should send the correct proof $P=\langle TP, DP\rangle$ where $DP=\sum_{i\in I}d_i\cdot m_i$. In the scheme, P is the correct proof and equation $e\left(TP,g\right)=e\left(\prod_{i\in I}H_3\left(i\right)^{d_i}\cdot u^{DP},pk\right)$ holds. Suppose the adversary's proof is $P^{'}=\langle TP', DP'\rangle$, where $DP^{'}=\sum_{i\in I}d_i\cdot m_i^{'}$. Then the equation $e\left(TP^{'},g\right)=e\left(\prod_{i\in I}H_3\left(i\right)^{d_i}\cdot u^{DP^{'}},pk\right)$ also holds. Suppose $\xi=DP=\sum_{i\in I}d_i\cdot m_i$, $\xi^{'}=DP^{'}=\sum_{i\in I}d_i\cdot m_i^{'}$. We can construct a simulator that uses the adversary to solve the CDH problem. Given $g,g^a,\varepsilon\in\mathbb{G}_1$, the simulator is asked to output ε^a.The simulator sets $pk=g^a$ and $u=g^{\mu}\varepsilon^{v}$ where $\mu,v\in\mathbb{Z}_p^*$. From the above two equations and the properties of bilinear maps, we conclude the following: $e\left(TP^{'}/TP,g\right)=e\left(u^{\xi^{'}-\xi},pk\right)=e\left(u^{\Delta\xi},pk\right)=e\left(\left(g^{\mu}\varepsilon^{v}\right)^{\Delta\xi},pk\right)$. From this equation, we can get $e\left(TP^{'}TP^{-1}pk^{-\mu\Delta\xi},g\right)=e\left(\varepsilon,pk\right)^{v\Delta\xi}$, so $\varepsilon^a=\left(TP^{'}TP^{-1}pk^{-\mu\Delta\xi}\right)^{\frac{1}{v\Delta\xi}}$. We can analyze the probability of the game failure through computing the probability that $v\Delta\xi=0\ mod\ q$. Because the probability that $v\Delta\xi=0\ mod\ q$ is only $1/q$, the probability can be negligible.

VII. PERFORMANCE EVALUATION

In this section, we evaluate the computation costs of patient and user in the scheme and compare it with scheme [49].

A. PERFORMANCE ANALYSIS

To analyze computation overhead of the scheme, we define the following notations to denote the corresponding operations: Let *Pair* denote a paring operation, *Hash* denote a hash operation and *Exp* denote an exponentiation operation. Similarly, let *Mul* and *Add* respectively represent a multiplication and addition operations. Xor and *Pref* respectively denote XOR and pseudo-random function operation of the scheme.

1) INITIAL PHASE

In algorithm *Setup*, TA computes $w=g^{\gamma}$, $v=e\left(g,h\right)$, and the computation overhead is $Exp+Pair$. In algorithm *KeyExtract*, TA first computes the private key $sk_{Uid}=g^{\frac{1}{\gamma+H_1\left(Uid_j\right)}}$ for user. Then TA picks random $a_1,a_2\in\mathbb{Z}_q^*$ and computes $b_1=h^{a_1}$, $b_2=h^{a_2}$. The warrant of user represents as $warr=a_1+a_2\cdot H_1\left(Uid_j\|\ time\right)$. Therefore, the computation overhead of the algorithm is $2Hash+3Exp+2pair+2Add+2Mul$. In algorithm *PatientReg*, TA computes $\phi=\prod_{k=1}^{t}\left(\gamma+H_1\left(Uid_k\right)\right)$ for patient and the computation overhead is $t(Add+Hash+Mul)$.

2) PREPROCESS PHASE

In *DataEnc*, Patient computes $C_1=w^{-r}$,$C_2=h^{r\cdot\phi}$,$K=v^r$ and encrypts M as $M^{'}=M\oplus H_2\left(K\right)$. Therefore, the computation overhead of the algorithm is $3Exp+Hash+mul+Xor$. In *TagGen*, SEM generates his public key $pk=h^x$ and computes $u=h^{\tau}$. Then SEM computes n tags as $\sigma_i=\left(H_3\left(i\right)\cdot u^{m_i}\right)^x$. Therefore the computation overhead of the algorithm is $\left(2+2n\right)Exp+nHash+nMul$.

3) SHARING PHASE

In *ChalGen*, user generates a subset $I=\langle i\rangle$ with c elements by f_1 and random numbers $l_i\in\mathbb{Z}_q^*$ by f_2. Therefore, the computation overhead is $2Pref$. In *ProfGen*, CS first checks user authority with equation $h^{warr}=b_1\cdot {b_2}^{H_1\left(Uid_j\|time\right)}$ and generates signature proof $TP=\prod_{i\in I}\sigma_i^{d_i}$ and data proof $DP=\sum_{i\in I}d_i\cdot m_i$. Therefore the computation overhead of the algorithm is $\left(c+2\right)Exp+Hash+\left(2c+1\right)Mul+cAdd$. In *ProfVer*, user verifies the integrity of data F with equation $e\left(TP,g\right)=e\left(\prod_{i\in I}H_1\left(i\right)^{d_i}\cdot u^{DP},pk\right)$, so the computation overhead is $2Pair+cHash+\left(c+1\right)Exp+Mul$. In *PreCompute*, TA computes $\delta=\prod_{k=1,k\neq j}^{t}\left(H_1\left(Uid_k\right)\right)$ and $\Delta_\gamma\left(Uid_j,S\right)=\gamma^{-1}\cdot\left(\phi\cdot\left(\gamma+H_1\left(Uid_j\right)\right)^{-1}-\delta\right)$, so the computation overhead of the algorithm is $tHash+\left(t+1\right)Mul+2Add+2Exp$. In *DataDecry*, user Uid_j first retrieve the symmetric encryption key K with equation $K=\left(e\left(C_1,h^{\Delta_\gamma\left(Uid_j,S\right)}\right)\cdot e\left(sk_{Uid},C_2\right)\right)^{\frac{1}{\delta}}$. Then user computes $M=M^{'}\oplus H_2\left(K\right)$ to get shared data. Therefore, the computation overhead in this

TABLE 2. Computation overhead in different phase of the scheme.

phase	algorithm	computation overhead
initial phase	$Setup$	$Exp + Pair$
	$KeyExtract$	$2Hash + 3Exp + 2Pair + 2Add + 2Mul$
	$PatientReg$	$t(Add + Hash + Mul)$
preprocess phase	$DataEnc$	$3Exp + Hash + Mul + Xor$
	$TagGen$	$(2 + 2n)Exp + nHash + nMul$
sharing phase	$ChalGen$	$2Pref$
	$ProfGen$	$(c + 2)Exp + Hash + (2c + 1)Mul + cAdd$
	$ProfVer$	$2Pair + cHash + (c + 1)Exp + Mul$
	$PreCompute$	$tHash + (t + 1)Mul + 2Add + 2Exp$
	$DataDecry$	$2Pair + Hash + 2Exp + Mul + Xor$

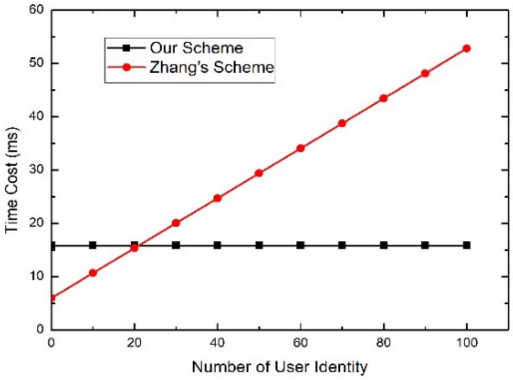


FIGURE 5. Computation Time of DO with Different Number of User Identity.

algorithm is $2Pair + Hash + 2Exp + Mul + Xor$. Table. 2 illustrates the computation overhead of each algorithm. From the table, we can conclude the computation overhead in *DataEnc* algorithm and *DataDecry* algorithm is constant.

B. EXPERIMENTAL RESULTS

We simulate our scheme with the Pairing based Cryptography (PBC) library of version 0.5.14. We compare the computation time of DO and user with scheme [49] by utilizing an MNT d159 curve with 160-bit group order. All the experiment results represent the average of 20 trials.

1) COMPUTAION TIME OF DO IN PREPROCESSING PHASE

The computation time of DO mainly generates in preprocessing phase. We first test the relation between DO's computation time and the number of user identity. From fig. 5, we can see that when the number of user identity varies from 1 to 100, the computation time of DO remains constant. Then we test the relation between DO's computation time and the size of shared data as described in fig. 6. When size of data is 1M, the time cost of DO is 25.1ms. With the size growing, the time increases slowly. When the size reaches 10M, the time cost is 37.72ms. From fig. 5 and fig. 6, we can conclude that DO's computation time in our scheme is lower than that of Zhang's scheme.

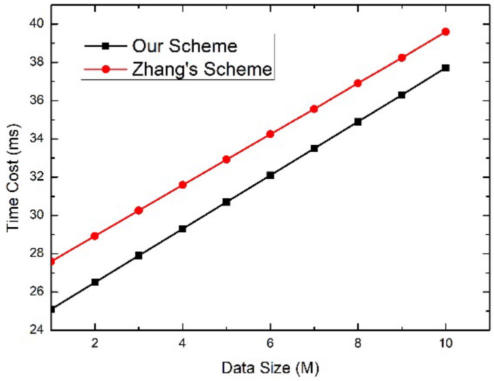


FIGURE 6. Computation Time of DO with Different Size of Shared Data.

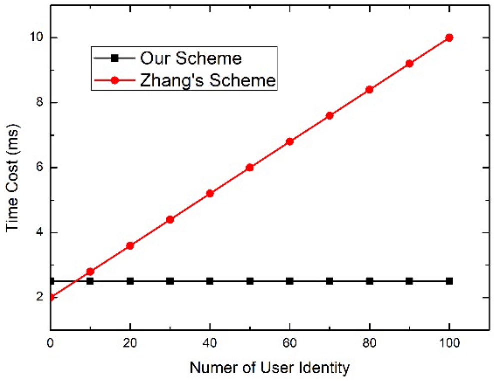


FIGURE 7. Decryption Time of User with Different Number of User Identity.

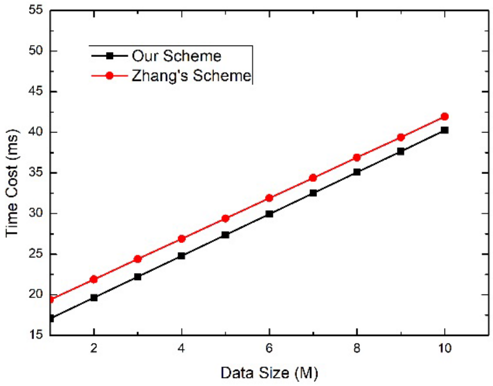


FIGURE 8. Decryption Time of User with Different Size of Shared Data.

2) COMPUTAION TIME OF USER IN SHARING PHASE

In data sharing phase, we first test the relationship between user's computation cost and the identity number as described in fig. 7. Because TA computes the intermediate data of decryption, user's computation time is constant when the number of user identity increases. We also test the relationship between user's computation cost and the size of shared data as described in fig. 8. We can see that with the number of identity growing, the computation cost of user increases slowly. From fig. 7 and fig. 8, we can conclude that the user's computation time in our scheme is less than that in Zhang's scheme.

VIII. CONCLUSION

In this paper, we propose a lightweight and secure health data sharing scheme for IoMT. The scheme ensures the health data private by allowing only the authorized user access the shared data. The scheme can also achieve efficient integrity verification by preventing user downloading damaged data. Finally, the scheme realizes lightweight operations of patient and user by IDDB encryption. From the experiment results and security analysis, we conclude that our scheme is more efficient in computation cost and more secure in health data sharing.

REFERENCES

[1] I. Farahat, A. Tolba, M. Elhoseny, and W. Eladrosy, "A secure real-time Internet of medical smart things (IOMST)," *Comput. Elect. Eng.*, vol. 72, pp. 455–467, Nov. 2018.

[2] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, Jan. 2018.

[3] Y. Zhang, M. Qiu, C.-W. Tsai, M. M. Hassan, and A. Alamri, "Health–CPS: Healthcare cyber–physical system assisted by cloud and big data," *IEEE Syst. J.*, vol. 11, no. 1, pp. 88–95, Mar. 2017.

[4] A. Ghazvini and Z. Shukur, "Security challenges and success factors of electronic healthcare system," *Procedia Technol.*, vol. 11, no. 1, pp. 212–219, 2013.

[5] Z. Guan, Z. Lv, X. Du, L. Wu, and M. Guizani, "Achieving data utility-privacy tradeoff in Internet of medical things: A machine learning approach," *Future Gener. Comput. Syst.*, vol. 98, pp. 60–68, Sep. 2019.

[6] M. Elhoseny, A. Abdelaziz, A. S. Salama, A. Riad, K. Muhammad, and A. K. Sangaiah, "A hybrid model of Internet of Things and cloud computing to manage big data in health services applications," *Future Gener. Comput. Syst.*, vol. 86, pp. 1383–1394, Sep. 2018.

[7] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The Internet of Things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[8] A. Bahga and V. K. Madisetti, "A cloud-based approach for interoperable electronic health records (EHRs)," *IEEE J. Biomed. Health Inform.*, vol. 17, no. 5, pp. 894–906, Sep. 2013.

[9] L. A. Tawalbeh, R. Mehmood, E. Benkhlifa, and H. Song, "Mobile cloud computing model and big data analysis for healthcare applications," *IEEE Access*, vol. 4, pp. 6171–6180, 2016.

[10] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for secure EHRs sharing of mobile cloud based E–health systems," *IEEE Access*, vol. 7, pp. 66792–66806, 2019, doi: 10.1109/access.2019.2917555.

[11] V. Chang, "Towards data analysis for weather cloud computing," *Knowl.-Based Syst.*, vol. 127, pp. 29–45, Jul. 2017.

[12] F. Gao and A. Sunyaev, "Context matters: A review of the determinant factors in the decision to adopt cloud computing in healthcare," *Int. J. Inf. Manage.*, vol. 48, pp. 120–138, Oct. 2019.

[13] G. Manogaran, N. Chilamkurti, and C.-H. Hsu, "Emerging trends, issues, and challenges in Internet of medical things and wireless networks," *Pers. Ubiquitous Comput.*, vol. 22, nos. 5–6, pp. 879–882, Oct. 2018.

[14] Y. Xiao, X. Shen, B. O. Sun, and L. Cai, "Security and privacy in RFID and applications in telemedicine," *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 64–72, Apr. 2006.

[15] D. Halperin, T. S. Heydt-Benjamin, K. Fu, T. Kohno, and W. H. Maisel, "Security and privacy for implantable medical devices," *IEEE Pervasive Comput.*, vol. 7, no. 1, pp. 30–39, Jan. 2008.

[16] Y. Ouyang, Z. Le, G. Chen, J. Ford, and F. Makedon, "Entrapping adversaries for source protection in sensor networks," in *Proc. Int. Symp. Word Wireless, Mobile Multimedia Netw.*, Jun. 2006, pp. 23–24.

[17] J. Zhang, F. Ren, S. Gao, H. Yang, and C. Lin, "Dynamic routing for data integrity and delay differentiated services in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 2, pp. 328–343, Feb. 2015.

[18] G. Yang, L. Xie, M. Mantysalo, X. Zhou, Z. Pang, L. D. Xu, S. Kao-Walter, Q. Chen, and L.-R. Zheng, "A health–IoT platform based on the integration of intelligent packaging, unobtrusive bio–sensor, and intelligent medicine box," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2180–2191, Nov. 2014.

[19] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute–based encryption with verifiable outsourced decryption," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 8, pp. 1343–1354, Aug. 2013.

[20] Q. Zhang, L. T. Yang, and Z. Chen, "Privacy preserving deep computation model on cloud for big data feature learning," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1351–1362, May 2016.

[21] Y. Ming and T. Zhang, "Efficient privacy–preserving access control scheme in electronic health records system," *Sensors*, vol. 18, no. 10, p. 3520, Oct. 2018.

[22] K. S. Kim and I. R. Jeong, "Efficient verifiable data streaming," *Secur. Commun. Netw.*, vol. 8, no. 18, pp. 4013–4018, Dec. 2015.

[23] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proc. ACM Conf. Comput. Commun. Secur.*, vol. 14, 2007, pp. 598–609.

[24] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. 4th Int. Conf. Secur. Privacy Commun. Netw.*, 2008, pp. 1–10.

[25] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proc. ACM Conf. Comput. Commun. Secur.*, vol. 17, 2009, pp. 213–222.

[26] Y. Zhu, G.-J. Ahn, H. Hu, S. S. Yau, H. G. An, and C.-J. Hu, "Dynamic audit services for outsourced storages in clouds," *IEEE Trans. Serv. Comput.*, vol. 6, no. 2, pp. 227–238, Apr. 2013.

[27] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 9, pp. 1717–1726, Sep. 2013.

[28] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 5, pp. 847–859, May 2011.

[29] C. Liu, J. Chen, L. T. Yang, X. Zhang, C. Yang, R. Ranjan, and K. Rao, "Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine–grained updates," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2234–2244, Sep. 2014.

[30] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 362–375, Feb. 2013.

[31] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," in *Proc. IEEE 5th Int. Conf. Cloud Comput. (CLOUD)*, Jun. 2012, pp. 295–302.

[32] G. Yang, J. Yu, W. Shen, Q. Su, Z. Fu, and R. Hao, "Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability," *J. Syst. Softw.*, vol. 113, pp. 130–139, Mar. 2016.

[33] Y. Luo, M. Xu, S. Fu, D. Wang, and J. Deng, "Efficient integrity auditing for shared data in the cloud with secure user revocation," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2015, pp. 434–442.

[34] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, "IoT-based big data storage systems in cloud computing: Perspectives and challenges," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 75–87, Feb. 2017.

[35] C. Hu, W. Li, X. Cheng, J. Yu, S. Wang, and R. Bie, "A secure and verifiable access control scheme for big data storage in clouds," *IEEE Trans. Big Data*, vol. 4, no. 3, pp. 341–355, Sep. 2018.

[36] J. Zhao, C. Xu, F. Li, and W. Zhang, "Identity–based public verification with privacy–preserving for data storage security in cloud computing," *IEICE Trans. Fundam.*, vol. E96.A, no. 12, pp. 2709–2716, 2013.

[37] G. Wu, Y. Mu, and W. Susilo, "Privacy-preserving cloud auditing with multiple uploaders," in *Proc. Int. Conf. Inf. Secur. Pract. Exper.* Zhangjiajie, China: Springer-Verlag, 2016, pp. 224–237.

[38] K. Bhaskaran, P. Ilfrich, D. Liffman, C. Vecchiola, P. Jayachandran, A. Kumar, F. Lim, K. Nandakumar, Z. Qin, V. Ramakrishna, E. G. Teo, and C. H. Suen, "Double–blind consent–driven data sharing on blockchain," in *Proc. IEEE Int. Conf. Cloud Eng. (ICE)*, Apr. 2018, pp. 385–439.

[39] C.-W. Liu, W.-F. Hsien, C.-C. Yang, and M.-S. Hwang, "A survey of public auditing for shared data storage with user revocation in cloud computing," *Int. J. Netw. Secur.*, vol. 18, no. 4, pp. 650–666, Jul. 2016.

[40] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 196. Santa Barbara, CA, USA: Springer-Verlag, 1984, pp. 47–53.

[41] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Adv. Cryptol. (CRYPTO)*. Santa Barbara, CA, USA: Springer-Verlag, 2001, pp. 213–229.

[42] E.-J. Yoon, Y. Choi, and C. Kim, "New ID-based proxy signature scheme with message recovery," in *Grid and Pervasive Computing* (Lecture Notes Computer Science), vol. 7861. Berlin, Germany: Springer-Verlag, 2013, pp. 945–951.

[43] C. Delerablée, "Identity-based broadcast encryption with constant size ciphertexts and private keys," in *Advances in Cryptology—ASIACRYPT*. Berlin, Germany: Springer, 2007, pp. 200–215.

[44] C. Gentry and B. Waters, "Adaptive security in broadcast encryption systems," in *Proc. 28th Annu. Int. Conf. Adv. Cryptol., Theory Appl. Cryptograph. Techn.*, 2009, pp. 171–188.

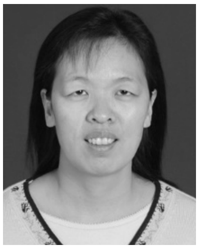
[45] J. Kim, W. Susilo, M. Ho Au, and J. Seberry, "Adaptively secure identity–based broadcast encryption with a constant–sized ciphertext," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 3, pp. 679–693, Mar. 2015.

[46] B.-C. Chen and H.-T. Yeh, "Secure proxy signature schemes from the Weil pairing," *J. Supercomput.*, vol. 65, no. 2, pp. 496–506, Aug. 2013.

[47] X. Liu, J. Ma, J. Xiong, T. Zhang, and Q. Li, "Personal health records integrity verification using attribute based proxy signature in cloud computing," in *Proc. Internet Distrib. Comput. Syst.*, in Lecture Notes Computer Science, vol. 8223. Hangzhou, China: Springer-Verlag, 2013, pp. 238–251.

[48] H. Guo, Z. Zhang, and J. Zhang, "Proxy re-encryption with unforgeable re-encryption keys," in *Cryptology and Network Security* (Lecture Notes Computer Science), vol. 8813. Berlin, Germany: Springer-Verlag, 2014, pp. 20–33.

[49] 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.*, vol. 5, no. 3, pp. 2130–2145, Jun. 2018.



XIUQING LU received the M.S. degree from the College of Computer Science, Shandong University, China. She is currently an Assistant Professor with the Computer Science Technology College, Qingdao University, China. Her current research interests focus on security of cloud computing and privacy of big data.

XIANGGUO CHENG received the Ph.D. degree from the China University of Petroleum. He is currently a Professor with the Computer Science Technology College, Qingdao University, China. His main research interest is network and information security.