

12-2018

Secure smart health with privacy-aware aggregate authentication and access control in Internet of Things

Yinghui ZHANG

Robert H. DENG


Singapore Management University, robertdeng@smu.edu.sg

Gang HAN

Dong ZHENG

DOI: <https://doi.org/10.1016/j.jnca.2018.09.005>

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

 Part of the [Information Security Commons](#), and the [Medicine and Health Sciences Commons](#)

Citation

ZHANG, Yinghui; DENG, Robert H.; HAN, Gang; and ZHENG, Dong. Secure smart health with privacy-aware aggregate authentication and access control in Internet of Things. (2018). *Journal of Network and Computer Applications*. 123, 89-100. Research Collection School Of Information Systems.

Available at: https://ink.library.smu.edu.sg/sis_research/4214

This Journal Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email libIR@smu.edu.sg.

Secure smart health with privacy-aware aggregate authentication and access control in Internet of Things

Yinghui Zhang^{a,b,c,d}, Robert H. Deng^c, Gang Han^{e,*}, Dong Zheng^{a,d}

^a National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an, 710121, China

^b State Key Laboratory of Cryptology, P.O. Box 5159, Beijing, 100878, China

^c School of Information Systems, Singapore Management University, Singapore

^d Westone Cryptologic Research Center, Beijing, 100070, China

^e School of Electronics and Information, Northwestern Polytechnical University, Xi'an, 710129, China

ARTICLE INFO

Keywords:

Smart health
Security
Privacy
Aggregate authentication
Access control

ABSTRACT

With the rapid technological advancements in the Internet of Things (IoT), wireless communication and cloud computing, smart health is expected to enable comprehensive and qualified healthcare services. It is important to ensure security and efficiency in smart health. However, existing smart health systems still have challenging issues, such as aggregate authentication, fine-grained access control and privacy protection. In this paper, we address these issues by introducing SSH, a Secure Smart Health system with privacy-aware aggregate authentication and access control in IoT. In SSH, privacy-aware aggregate authentication is enabled by an anonymous certificateless aggregate signature scheme, in which users' identity information is protected based on symmetric encryption mechanisms. In addition, privacy-aware access control is based on anonymous attribute-based encryption technologies. Our formal security proofs indicate that SSH achieves batch authentication and non-repudiation under the Computational Diffie-Hellman assumption. Extensive experimental results and performance comparisons show that SSH is practical in terms of computation cost and communication overheads.

1. Introduction

The improvement of people's living standards makes qualified healthcare services attractive which have recently drawn worldwide attentions. In particular, the advancements in the Internet of Things (IoT) and wireless communication technologies make the collection of health data more and more convenient. As a context-aware complement of health services in mobile scenarios and smart cities, IoT enabled smart health significantly promotes the scale and flexibility of data collection. To make the most of collected health data, which is called smart health records (SHRs), it is necessary to design a secure and efficient health data sharing system. The integration of IoT and cloud computing technologies has become a promising solution to the above problem. As shown in Fig. 1, different types of smart devices can collect SHRs and outsource them to the cloud server for storage and sharing. As we know, health data is usually sensitive and related to people's lives. Therefore, security and privacy protection measures should be adopted to eliminate erroneous SHRs from malicious users and prevent

privacy leakage of SHR owners. Most importantly, SHRs should be collected in a timely fashion to respond to such time-sensitive scenarios as the medical emergency. Generally, it is still necessary to simultaneously address the issues of *aggregate authentication*, *fine-grained access control* and *privacy protection* to realize a secure and efficient smart health system.

In smart health, a large number of IoT devices collect SHRs and transmit them to a cloud service provider (CSP) for storage and sharing among different users. Upon receiving SHRs from different IoT devices, CSP should check the validity of the SHRs, which can be realized by the digital signature technology. However, if CSP performs the verification one by one, the computation time will increase with the number of SHRs, which is not suitable for the time-sensitive medical emergency. The technique of aggregate authentication allows CSP to check many SHRs at one time and hence should be enabled to improve the verification efficiency (Boneh et al., 2003). However, most of existing aggregate authentication cannot protect users' identity privacy. Besides SHR storage, fine-grained access control of SHRs is important for practi-

* Corresponding author. School of Electronics and Information, Northwestern Polytechnical University, Xi'an, 710129, China.

E-mail addresses: yinghuizhang@smu.edu.sg (Y. Zhang), robertdeng@smu.edu.sg (R.H. Deng), hangang021@gmail.com (G. Han), zhengdong@xupt.edu.cn (D. Zheng).



Fig. 1. Smart health integrating IoT and cloud computing.

cal applications because different users usually have diverse attributes. Attribute-based encryption (ABE) is envisioned as a highly promising technique which can be used to realize fine-grained access control mechanisms (Sahai and Waters, 2005). In a desirable smart health system, it is indispensable to combine the aggregate authentication technology and ABE in a privacy-aware manner. To be specific, users' public keys such as identity and attributes cannot be known to adversaries. To the authors' knowledge, most of previous health-related schemes cannot authenticate the collected SHRs at one time while enabling fine-grained access control and ensuring users' privacy protection.

1.1. Our contribution

In this paper, we simultaneously address the data security and privacy issues in IoT enabled smart health by introducing SSH, a Secure Smart Health system with privacy-aware aggregate authentication and fine-grained access control. In SSH, we focus on the aforementioned issues including aggregate authentication, fine-grained access control and privacy protection. Our contributions can be summarized as follows.

- Firstly, we propose an anonymous certificateless aggregate signature scheme, which serves as a fundamental building block of SSH. The signature scheme is used by CSP to aggregate SHRs from different IoT terminals and then authenticate the collected data at one time. To enable fine-grained access control, we combine the proposed aggregate signature scheme and anonymous attribute-based encryption techniques, and hence unauthorized users cannot access corresponding SHRs.
- Secondly, we formally prove the security of SSH in the random oracle model under the Computational Diffie-Hellman (CDH) assumption. In particular, the proposed signature scheme is existentially unforgeable against adaptively chosen-message attacks, which can prevent invalid SHRs from being uploaded to CSP. The authentication is key-escrow free because the signing secret key is jointly generated by the user and the registration center.
- Finally, we analyze the security features, computation cost and communication overhead of SSH. Our extensive experiments based on a laptop and a smart mobile phone indicate that SSH is more efficient than other related schemes in terms of the signing time, the verification time, and the aggregate verification time.

1.2. Related work

In the era of IoT and cloud computing, smart health is indispensable for the realization of proactive and comprehensive healthcare, which enables the early-stage diagnosis. For a secure smart health system, many security technologies should be adopted, such as authentication (Shen et al., 2018a; Xu et al., 2018), access control (Li et al., 2018a; Castiglione et al., 2016), and network security solutions (Zhang et al., 2014; Fan et al., 2017), etc.

IoT and cloud computing. To achieve a secure IoT environment, Wang et al. (2018) proposed a security mechanism in IoT based on an instant encrypted transmission. Shen et al. (2018b) proposed a secure data uploading scheme for smart home systems. Jhaveri et al. (2018) made a sensitivity analysis of an attack pattern discovery in industrial IoT. Besides IoT security, cloud computing security is very important for smart health storage and sharing. Wu et al. (2018) proposed a biometric key generation method for flexible authentication in cloud computing. Yang et al. (2018) proposed a remote data encryption mechanism for mobile cloud computing. Zhang et al. (2018a) proposed an efficient and privacy-aware data sharing scheme for cloud storage. Li et al. (2015) proposed a secure data deduplication scheme for hybrid clouds. Wang et al. (2011) proposed a method of ensuring data integrity in cloud computing. Based on blockchain technologies, Zhang et al. (2018b) proposed a trustworthy searchable encryption scheme in cloud computing. The scheme realizes two-side verifiability and can resist malicious users and malicious cloud servers. Many other IoT and cloud computing security schemes (Jiang et al., 2015; Chen et al., 2016; Zhang et al., 2016a, 2017; Shu et al., 2018a, 2018b; Cai et al., 2017) have been proposed in recent years. Particularly, the promising blockchain technologies have been used to realize decentralized outsourcing services such as provable data possession, searchable encryption and outsourcing computation in cloud computing (Zhang et al., 2018c, 2018d).

Authentication. As a basic requirement in IoT enabled smart health, authentication can be realized by digital signature techniques. However, traditional signature schemes cannot realize aggregate authentication and hence suffer efficiency drawbacks. Boneh et al. (2003) proposed a novel technique called aggregate signature, which can be used to reduce the verification cost. Selvi et al. (2012) proposed an identity-based partial aggregate signature scheme without bilinear pairing operations. Shen et al. (2017) proposed identity-based aggregate signature scheme for wireless sensor networks. To further address the key-escrow problem in aggregate signature, Castro and Dahab (2007) proposed the notion of certificateless aggregate signature. Xiong et al. (2013) proposed a certificateless aggregate signature scheme with constant bilinear pairing operations. However, He et al. (2014) pointed that the scheme (Xiong et al., 2013) cannot resist forgery attacks and presented an improved scheme. Very recently, Li et al. (2018b) have showed that the improved scheme (He et al., 2014) is not secure if the key generator center is malicious-but-passive. Tu et al. (2014) also pointed that the scheme (Xiong et al., 2013) cannot resist forgery attacks and presented a new improved scheme. Malhi and Batra (2015) proposed a certificateless aggregate signature scheme suitable for vehicular ad-hoc networks. Chu et al. (2014) proposed a key-aggregate cryptosystem for scalable data sharing in cloud storage.

Access control. In cloud computing, ABE is a promising tool for realizing fine-grained access control. ABE is categorized into key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE) (Goyal et al., 2006). In smart health, CP-ABE is very useful because it enables SHR owners to determine the authorized users. In CP-ABE, a user can apply an attribute secret key based on his attribute list, which is used to decrypt SHR ciphertexts. The ciphertext is generated under ciphertext policy and the decryption is successful only if the user's attribute list satisfies the policy. Zhang et al. (2016b) proposed an efficient attribute-based data sharing scheme for mobile cloud computing, in which decryption only needs constant bilinear pairing operations. Wang et al. (2017) proposed a directly revocable attribute-based encryption scheme and showed its applications in cloud storage. In particular, access control suitable for resource-constrained users (Zhang et al., 2018e) and access control with leakage resilience (Zhang et al., 2018f) have been studied.

Privacy protection. In smart health, privacy is a very important issue. Very recently, Liu et al. (2018) proposed an anonymous certificateless aggregate signature for mobile healthcare crowd sensing. However, the scheme suffers signature forgery attacks from malicious participants because the relationship between the public key and the partial private key is not reflected in the signature (Zhang et al., 2018g).

Zhang et al. (2018h) proposed a policy-hiding access control scheme to address security and privacy issues in smart health. In recent years, privacy protection technologies have received more and more attentions such as accountability in cloud computing (Xhafa et al., 2015) and 5G security (Zhang et al., 2018i). Nevertheless, previous schemes cannot simultaneously address the aforementioned security and privacy issues in IoT enabled smart health.

1.3. Organization

The rest of the paper is organized as follows. Some preliminaries are reviewed in Section 2. We then give the system architecture, adversary model and definitions in Section 3. The proposed smart health system is detailed in Section 4 followed by its security analysis in Section 5. Our experimental results are presented in Section 6. Finally, concluding remarks are made in Section 7.

2. Preliminaries

2.1. Notations

In Table 1, we describe notations used throughout this paper.

2.2. Bilinear pairing

Let \mathbb{G} and \mathbb{G}_T be a cyclic additive and a multiplicative group of the same prime order p , respectively. We call \hat{e} a bilinear pairing if for $P, Q \in \mathbb{G}$, $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a map satisfying the following three properties:

1. Bilinear: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for $a, b \in \mathbb{Z}_p^*$.
2. Non-degenerate: There exists $P, Q \in \mathbb{G}$ such that $\hat{e}(P, Q) \neq 1$.
3. Computable: $\hat{e}(P, Q)$ can be efficiently computed.

2.3. Number-theoretic problem and assumption

Computation Diffie-Hellman (CDH) Problem: Given a tuple $(P, aP, bP) \in \mathbb{G}^3$, compute $abP \in \mathbb{G}$. An algorithm \mathcal{A} is said to solve the CDH problem in \mathbb{G} with an advantage ϵ if

$$\Pr [\mathcal{A}(P, aP, bP) = abP] \geq \epsilon,$$

where the probability is over the random choice of $P \in \mathbb{G}$, $a, b \in \mathbb{Z}_p^*$, and the random coins used by \mathcal{A} .

CDH Assumption: The CDH assumption is said to hold in \mathbb{G} if no polynomial-time algorithm has a non-negligible advantage in solving the CDH problem in \mathbb{G} .

3. Model, design goal and definition

In this section, we first describe the system architecture of SSH and then give the adversary model and design goals. Finally, we present related definitions.

3.1. System model

As shown in Fig. 2, the system model of SSH involves a registration center (RC), data owners (DO), data users (DU) and a cloud service provider (CSP). These entities are detailed as follows:

- RC is responsible for the registration of DO and DU. In SSH, both DO and DU will not fully trust RC, that is, RC may maliciously use partial secret parameters which are generated by RC for DO and DU. In the process, the identity privacy is realized because the identity ciphertext is used by RC.
- DO collects SHR based on various smart terminals, such as smart devices and wireless sensors. DO generates secret parameters himself and also obtains the other secret parameter partially generated by RC. In order to realize privacy-preserving fine-grained access control over SHRs and SHR authentication, DO encrypts SHRs and generates a certificateless signature of the SHR ciphertext. Both the ciphertext and the signature are sent to CSP for storage and sharing.
- CSP enables privacy-preserving SHR storage and management. In order to reduce the computation cost, CSP will anonymously aggregate the received SHR ciphertext and signature. CSP can check the validity of received signatures at one time. If and only if the verification result is true, CSP stores the corresponding SHR ciphertexts and signatures, which will be accessed by DU. In the process of aggregation and verification, DO's privacy is preserved because his actual identity is not revealed to CSP.
- DU intends to get SHRs from CSP for particular use in practice. DU can be a doctor, a researcher, or a hospital, etc. It is significant to ensure that only authorized DU can access corresponding SHRs. Similar to DO, DU can get his secret parameters including a user secret key and a partial secret key from RC. After downloading outsourced SHR, DU first checks the validity based on the signature and then decrypts the ciphertext. If and only if his attributes satisfy the underlying ciphertext policy, the original SHR can be recovered.

3.2. Adversary model and design goals

In SSH, no trusted entities are required. Specifically, DO may maliciously outsource invalid SHRs to CSP, which can lead to severe results if the SHR is adopted by DU in practical applications. DO also tries to forge signatures of randomly chosen SHRs on behalf of other data owners. CSP may also forge signatures on randomly chosen SHRs and tries to reveal the actual identity of DO corresponding to a SHR. DU wants to access SHRs even if he is not an authorized user. Particularly, different UDs may collude with each other to access some SHRs which cannot be accessed by each of them individually. Most importantly, RC is not a fully trusted entity, and it may be compromised by adversaries.

In general, we aim to realize a secure and efficient smart health system supporting privacy protection, authentication and fine-grained access control. Concretely, SSH should achieve the following security and performance goals.

Table 1
Notations used in SSH.

Notation	Meaning	Notation	Meaning
RC	The registration center	p	A prime
DO	The data owner	\mathbb{G}	A cyclic additive group of order p
DU	The data user	\mathbb{G}_T	A cyclic multiplicative group of order p
CSP	The cloud service provider	H_i	Secure hash functions
upk_{ID}	The user public key of ID	TS	A timestamp
usk_{ID}	The user secret key of ID	TE	The expiration date of registration
psk_{ID}	The partial user secret key of ID	(m_i, σ_i)	A message and signature pair of ID_i
L	An attribute list	sk_L	The attribute secret key of L
W	A ciphertext policy	ct_W	A ciphertext with W

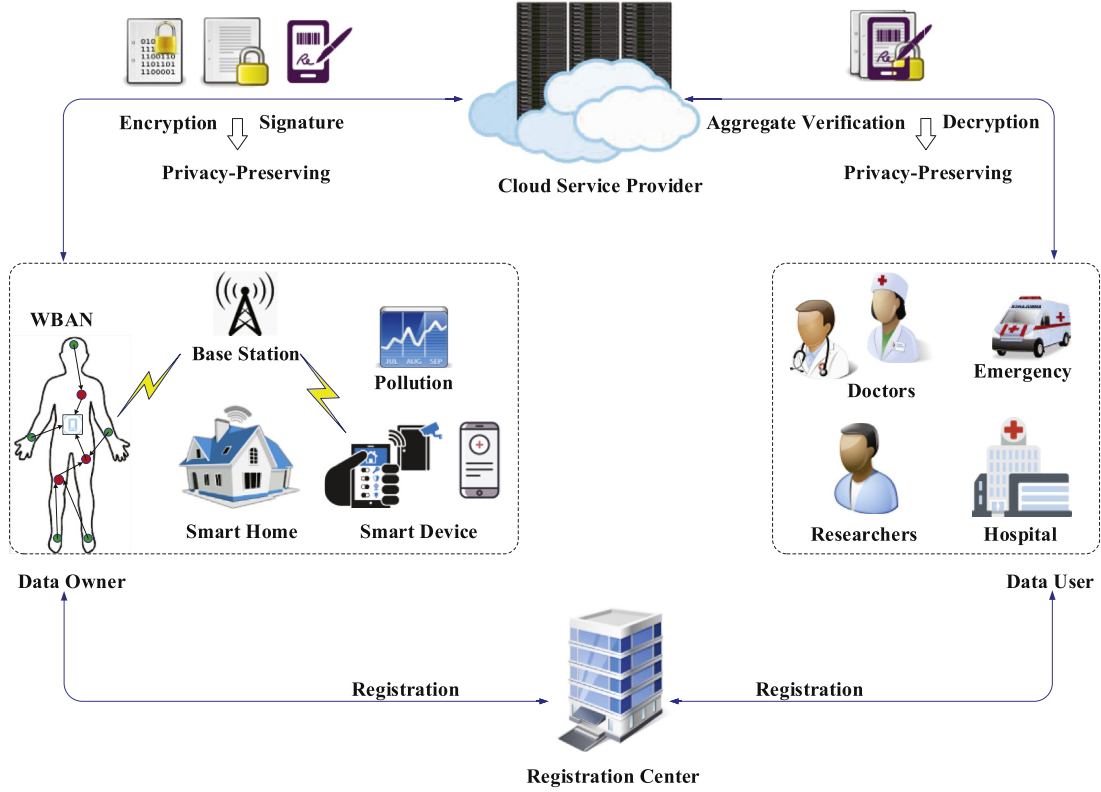


Fig. 2. System architecture of SSH.

3.2.1. Security goals

- *SHR confidentiality.* Unauthorized DU cannot obtain the plaintext of outsourced SHRs. In particular, unauthorized access from CSP should also be prevented.
- *Collusion-resistance.* Even if many DUs and CSP collude, it is infeasible for them to access the plaintext of outsourced SHRs if each one cannot individually access.
- *Batch authentication.* The outsourced SHRs from different DOs can be efficiently aggregated and verified by CSP and DU. If some forged SHRs exist, the batch authentication fails and hence invalid SHRs are prevented to be shared by DUs.
- *Non-repudiation.* If DO submits a SHR to CSP for storage and sharing, he cannot deny the fact. That is, DO is responsible for the validity of his SHR.
- *Privacy protection.* CSP and DU can only aggregate the outsourced SHRs from DOs and check the aggregated result. The actual identities of DOs cannot be known by CSP and DU. In addition, DU cannot obtain ciphertext policies embedded in SHR ciphertexts, which protects DO's attribute privacy.

3.2.2. Performance goals

- *Communication overhead.* For a desirable smart health system, the ciphertext length and signature size of an outsourced SHR should be as small as possible.
- *Computation cost.* The computation cost should be as small as possible because many smart devices are resource-constrained. In particular, time-sensitive applications in smart health such as emergency should be taken into consideration.

3.3. Definition of anonymous certificateless aggregate signature

As an ingredient of SSH, an anonymous certificateless aggregate signature scheme consists of seven algorithms MasterKeyGen , UserKeyGen , PartialKeyGen , AnonSign , AnonVerify , AnonAggregate , and AnonAggVerify , which are defined as follows¹:

- $\text{MasterKeyGen}(1^\lambda) \rightarrow (\text{params}, \text{msk})$: The master key generation algorithm is performed by RC. On input a security parameter λ , it generates a system public parameter params and a master secret key msk .
- $\text{UserKeyGen}(\text{params}) \rightarrow (\text{upk}_{ID}, \text{usk}_{ID})$: The user key generation algorithm is run by DO itself. Suppose DO has an identity ID . On input params , it generates a user public and secret key pair $(\text{upk}_{ID}, \text{usk}_{ID})$.
- $\text{PartialKeyGen}(\text{params}, \text{msk}, C, \text{upk}_{ID}) \rightarrow \text{psk}_{ID}$: The partial user key generation algorithm is performed by RC. On input params , msk , a symmetric encryption ciphertext C of ID , and upk_{ID} , it returns a partial user secret key psk_{ID} .

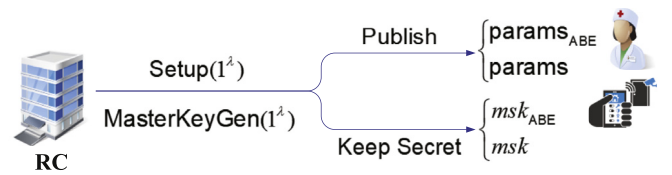


Fig. 3. The system initialization phase.

¹ Anonymity means the actual identity of DO is not known by DUs.

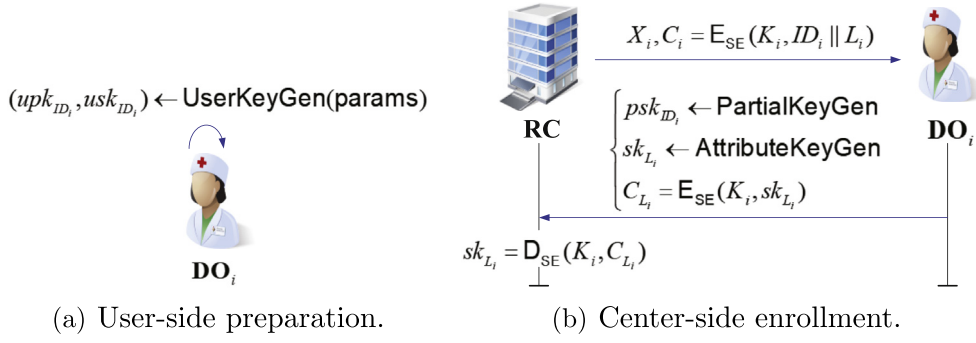


Fig. 4. The user registration phase.

- $\text{AnonSign}(\text{params}, usk_{ID_i}, upk_{ID_i}, psk_{ID_i}, m_i) \rightarrow \sigma_i$: The anonymous signature generation algorithm is performed by DO_i . Suppose DO_i has an identity ID_i . On input params , usk_{ID_i} , upk_{ID_i} , psk_{ID_i} and a message m_i , it generates a signature σ_i .
- $\text{AnonVerify}(\text{params}, m_i, \sigma_i) \rightarrow \text{true or false}^2$: The anonymous signature verification algorithm is performed by CSP or DU. On input params , m_i and σ_i from DO_i (resp. CSP) for a given i , CSP (resp. DU) outputs true if σ_i is a valid signature of m_i , otherwise outputs false.
- $\text{AnonAggregate}(\text{params}, \{m_i, \sigma_i\}_{1 \leq i \leq n}) \rightarrow \sigma$: The anonymous aggregate signature generation algorithm is performed by CSP or DU. On input params and a message signature pair (m_i, σ_i) from DO_i (resp. CSP) for $1 \leq i \leq n$, CSP (resp. DU) generates an aggregate signature σ on messages m_1, m_2, \dots, m_n .
- $\text{AnonAggVerify}(\text{params}, \{m_i\}_{1 \leq i \leq n}, \sigma) \rightarrow \text{true or false}$: The anonymous aggregate signature verification algorithm is performed by CSP and DU. On input params , $\{m_i\}_{1 \leq i \leq n}$ and σ , it outputs true if σ is a valid aggregate signature on $\{m_i\}_{1 \leq i \leq n}$, otherwise it outputs false.

3.4. Definition of anonymous CP-ABE

Another building block of SSH is anonymous CP-ABE. An anonymous CP-ABE scheme is composed of four algorithms Setup, AttributeKeyGen, AnonEncrypt, and AnonDecrypt, which are defined as follows³:

- $\text{Setup}(1^\lambda) \rightarrow (\text{params}, msk)$: The system setup algorithm is run by RC. On input a security parameter λ , it generates a system public parameter params and a master secret key msk .
- $\text{AttributeKeyGen}(\text{params}, msk, L) \rightarrow sk_L$: The key generation algorithm is run by RC. On input params , msk and an attribute list L , it returns an attribute secret key sk_L associated with L .
- $\text{AnonEncrypt}(\text{params}, m, W) \rightarrow ct_W$: The anonymous encryption algorithm is run by DO. On input params , a message m and a ciphertext policy W , it outputs a ciphertext ct_W of m under W , where W is hidden in ct_W .
- $\text{AnonDecrypt}(\text{params}, ct_W, sk_L) \rightarrow m \text{ or } \perp$: The anonymous decryption algorithm is performed by DU. On input params , a ciphertext ct_W of message m under W , and an attribute secret key sk_L , it returns \perp and terminates if L does not match W . Otherwise, it outputs m .

4. SSH: secure smart health system integrating IoT and cloud

SSH consists of four phases: system initialization, user registration, health data outsourcing and health data access, which are described below.

4.1. System initialization

RC first specifies a security parameter λ , a symmetric encryption cryptosystem (E_{SE}, D_{SE}) , and an anonymous CP-ABE scheme $(\text{Setup}, \text{AttributeKeyGen}, \text{AnonEncrypt}, \text{AnonDecrypt})$. Then, it runs $\text{Setup}(1^\lambda)$ to get params_{ABE} and msk_{ABE} , and performs the following algorithm $\text{MasterKeyGen}(1^\lambda)$ to obtain params and msk . Finally, RC publishes params_{ABE} and params , and keeps msk_{ABE} and msk secret. The initialization phase is illustrated in Fig. 3.

$\text{MasterKeyGen}(1^\lambda) \rightarrow (\text{params}, msk)$: RC first chooses a bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, where \mathbb{G} and \mathbb{G}_T are a cyclic additive and a multiplicative group of the same prime order p , respectively. Then, it chooses three hash functions $H_1 : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}$, $H_2 : \{0, 1\}^* \rightarrow \mathbb{G}$ and $H_3 : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$. RC also picks $s \in_R \mathbb{Z}_p^*$ and computes $P_{pub} = sP$, $Q = H_1(P \| P_{pub})$, where P is a generator of \mathbb{G} . Finally, RC sets $\text{params} = \langle P, P_{pub}, Q, p, \mathbb{G}, \mathbb{G}_T, H_1, H_2, H_3 \rangle$ and $msk = s$.

4.2. User registration

Suppose a data owner DO_i has an identity ID_i and an attribute list L_i , and it intends to join SSH. As shown in Fig. 4, DO_i sequentially performs the following procedures:

- (1) *User-Side Preparation*. DO_i makes a user-side preparation based on the algorithm UserKeyGen as below.

$\text{UserKeyGen}(\text{params}) \rightarrow (upk_{ID_i}, usk_{ID_i})$: DO_i chooses $x_i \in_R \mathbb{Z}_p^*$ and computes $X_i = x_i P$. Then, $upk_{ID_i} = X_i$ and $usk_{ID_i} = x_i$ are returned.

- (2) *Center-Side Enrollment*. DO_i sends X_i and $C_i = E_{SE}(K_i, ID_i \| L_i)$ to RC for center-side enrollment, where $K_i = x_i P_{pub}$.⁴ Upon receiving X_i and C_i , RC performs the algorithm PartialKeyGen as below to get psk_{ID_i} and returns it to DO_i . Furthermore, RC performs the algorithm AttributeKeyGen($\text{params}_{ABE}, msk_{ABE}, L_i$) to get sk_{L_i} and sends $C_{L_i} = E_{SE}(K_i, sk_{L_i})$ to DO_i . After receiving C_{L_i} , DO_i can get $sk_{L_i} = D_{SE}(K_i, C_{L_i})$.

$\text{PartialKeyGen}(\text{params}, msk, C_i, upk_{ID_i}) \rightarrow psk_{ID_i}$: RC computes $K_i = sX_i$ and recovers $ID_i \| L_i = D_{SE}(K_i, C_i)$. After ensuring the validity of ID_i , RC locally stores (ID_i, X_i, TE_i) and sends $S_i = sY_i$ to DO_i , where $Y_i = H_2(X_i \| TE_i)$ and TE_i is the expiration date of registration. In addition, RC sets $psk_{ID_i} = S_i$.

4.3. Health data outsourcing

- (1) *SHR Uploading*. As shown in Fig. 5, after collecting a SHR m_i , DO_i specifies a ciphertext policy W_i and performs $\text{AnonEncrypt}(\text{params}_{ABE}, m_i, W_i)$ to get ct_{W_i} . Then, DO_i chooses

² The user public key upk_{ID_i} is implicitly used as a component of σ_i .

³ Anonymity means the access policy is hidden in ciphertexts.

⁴ In symmetric encryption, a hash function can be used if necessary for the suitable value of symmetric key.

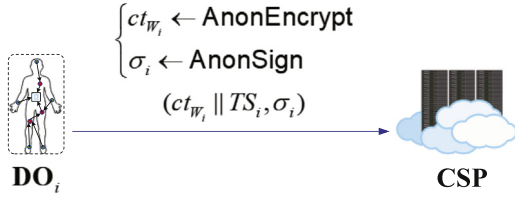


Fig. 5. The SHR uploading phase.

a timestamp TS_i and runs the algorithm AnonSign as below to obtain σ_i , in which $ct_{W_i} || TS_i$ is used as the message. Finally, DO_i sends $(ct_{W_i} || TS_i, \sigma_i)$ to CSP.

$\text{AnonSign}(\text{params}, usk_{ID_i}, upk_{ID_i}, psk_{ID_i}, ct_{W_i} || TS_i) \rightarrow \sigma_i$: DO_i chooses $r_i \in_R \mathbb{Z}_p^*$ and computes $U_i = r_i X_i$, $h_i = H_3(ct_{W_i} || TS_i || X_i || U_i)$, and $V_i = S_i + r_i h_i x_i P_{\text{pub}} + x_i Q$. Then, DO_i sets $\sigma_i = \langle U_i, V_i, X_i \rangle$ as the signature on $ct_{W_i} || TS_i$.

(2) *SHR Storage*. Upon receiving $(ct_{W_i} || TS_i, \sigma_i)$ from DO_i for $1 \leq i \leq n$, CSP first checks the freshness of TS_i , and if it is fresh, CSP ensures the signature component X_i has not expired. Then, CSP performs the algorithm AnonAggregate as below to obtain the aggregate signature σ . Finally, CSP runs the algorithm AnonAggVerify as below to check the validity of σ . If and only if AnonAggVerify returns true, CSP stores $\{ct_{W_i}, TS_i, TE_i, \sigma_i\}_{1 \leq i \leq n}$.

$\text{AnonAggregate}(\text{params}, \{ct_{W_i}, TS_i, TE_i, \sigma_i\}_{1 \leq i \leq n}) \rightarrow \sigma$: For $1 \leq i \leq n$, CSP computes $h_i = H_3(ct_{W_i} || TS_i || X_i || U_i)$, $Y_i = H_2(X_i || TE_i)$ and generates $\sigma = \langle U, V, X, Y \rangle$, where

$$U = \sum_{1 \leq i \leq n} h_i U_i, V = \sum_{1 \leq i \leq n} V_i, X = \sum_{1 \leq i \leq n} X_i, Y = \sum_{1 \leq i \leq n} Y_i.$$

$\text{AnonAggVerify}(\text{params}, \{ct_{W_i}, TS_i\}_{1 \leq i \leq n}, \sigma) \rightarrow \text{true or false}$: CSP returns true to indicate $\{ct_{W_i}, TS_i\}_{1 \leq i \leq n}$ are valid if and only if $\hat{e}(V, P) = \hat{e}(Y + U, P_{\text{pub}}) \hat{e}(X, Q)$.

4.4. Health data access

As shown in Fig. 6, DU downloads outsourced SHR data $\{ct_{W_i}, TS_i, TE_i, \sigma_i\}_{1 \leq i \leq n}$ from CSP. In order to access SHRs, DU performs Algorithm 1.

$\text{AnonVerify}(\text{params}, ct_{W_i}, TS_i, TE_i, \sigma_i) \rightarrow \text{true or false}$: DU first ensures TS_i is fresh and the signature component X_i has not expired. Then, DU computes $h_i = H_3(ct_{W_i} || TS_i || X_i || U_i)$, $Y_i = H_2(X_i || TE_i)$, and checks if $\hat{e}(V_i, P) = \hat{e}(Y_i + h_i U_i, P_{\text{pub}}) \hat{e}(X_i, Q)$. If it holds, true is returned.

5. Security analysis

In SSH, an anonymous certificateless aggregate signature scheme, an anonymous CP-ABE scheme and a symmetric encryption scheme are used, in which the encryption schemes have been proven secure. As shown in Section 3.2, the security goals of SSH are *SHR confidentiality*, *collusion-resistance*, *batch authentication*, *non-repudiation*, and *privacy*

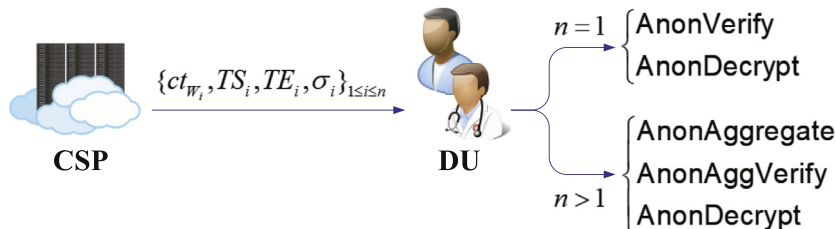


Fig. 6. The health data access phase.

protection. The security of the adopted CP-ABE scheme realizes *SHR confidentiality* and *collusion-resistance*. Furthermore, *identity anonymity* is enabled based on the symmetric encryption technology. Therefore, in the following, we only need to prove the security of the underlying signature scheme, which ensures *batch authentication* and *non-repudiation* of SSH.

5.1. Formalized security model

We consider such a security model that no entity is fully trusted by the others. In Au et al. (2007), a formalized security model of certificateless signature is described, in which the adversary can launch attacks before generating a system public parameter. Concretely, two types of adversaries should be taken into account.

- **Type I Adversary \mathcal{A}_I** . It is capable of replacing the public key of other entities with a value of his choice, but is not allowed to access the master secret key.
- **Type II Adversary \mathcal{A}_{II}** . It is allowed to access the master secret key, but cannot make public key replacement.

Because identity is encrypted based on a symmetric encryption mechanism in SSH, we further consider aggregation in our security model. In fact, our formalized security model is similar to the one in Au et al. (2007) and the difference lies in that the algorithms AnonAggregate and AnonAggVerify are performed to aggregate and check signatures.

5.2. Security results

Theorem 1. For type I adversaries, SSH is existentially unforgeable against adaptively chosen-message attacks in the random oracle model under the CDH assumption.

Proof. Given a random CDH instance $(P, aP, bP) \in \mathbb{G}^3$, where $a, b \in_R \mathbb{Z}_p^*$, the goal of C is to compute $abP \in \mathbb{G}$. Denote aP and bP by A and B , respectively. Suppose there is a type I adversary \mathcal{A}_I that breaks SSH in time t_1 with advantage ϵ_1 . In the following, we show that \mathcal{A}_I is used as a subroutine by the challenger C to solve the CDH problem with advantage ϵ in time t after making q_i queries to H_i for $i = 1, 2, 3$, q_{us} user secret key queries, q_{ps} partial secret key queries, q_{pk} public key queries, q_{sig} signing queries, and q_{as} aggregate signing queries, where

$$\epsilon \geq \frac{\epsilon_1}{(q_{ps} + 1)e} \text{ and } t \leq t_1$$

$$+ \mathcal{O}(q_1 + q_2 + q_3 + q_{us} + q_{ps} + q_{pk} + q_{sig} + nq_{as})T_m.$$

Here, e is the base of the natural logarithm, n represents the number of aggregated signatures, and T_m is the computation time of a scalar multiplication operation in \mathbb{G} .

Init. C sets $P_{\text{pub}} = A$, $Q = H_1(P || P_{\text{pub}})$ and sends $\text{params} = \langle P, P_{\text{pub}}, Q, p, \mathbb{G}, \mathbb{G}_T, H_1, H_2, H_3 \rangle$ to \mathcal{A}_I , in which H_1, H_2, H_3 are answered by C in the following queries. A list $\mathcal{L} = (ID_i, x_i, S_i, X_i)$ is maintained by C and the symbol \perp means the corresponding value is invalid.

Queries on Oracle H_1 . C maintains a list $\mathcal{L}_1 = (P, P_{\text{pub}}, c_Q, Q)$ which is initially empty. Upon receiving a query (P, P_{pub}) on H_1 from \mathcal{A}_I , C

Algorithm 1 Health data access.**Input:** Outsourced SHR ciphertext and signature

$$\{ct_{W_i}, TS_i, TE_i, \sigma_i\}_{1 \leq i \leq n}$$

Output: SHRs $\{m_i\}_{1 \leq i \leq n}$

```

1 if  $n = 1$  then
2   perform AnonVerify(params,  $ct_{W_1}, TS_1, TE_1, \sigma_1$ ).
3   if AnonVerify returns true then
4     run AnonDecrypt(params,  $ct_{W_1}, sk_{L_1}$ ).
5     if  $L_1$  matches  $W_1$  then
6       return SHR  $m_1$ .
7 else
8   perform AnonAggregate(params,  $\{ct_{W_i}, TS_i, TE_i, \sigma_i\}_{1 \leq i \leq n}$ ) to get  $\sigma$ .
9   run AnonAggVerify(params,  $\{ct_{W_i}, TS_i\}_{1 \leq i \leq n}, \sigma$ ).
10  if AnonAggVerify returns true then
11    for  $1 \leq i \leq n$  do
12      perform AnonDecrypt(params,  $ct_{W_i}, sk_{L_i}$ ).
13      if  $L_i$  matches  $W_i$  then
14        return SHR  $m_i$ .

```

returns the same value if the query has ever been made. Otherwise, \mathcal{C} chooses $c_Q \in_R \mathbb{Z}_p^*$, returns $Q = c_Q P$ and adds $(P, P_{\text{pub}}, c_Q, Q)$ to \mathcal{L}_1 .

Queries on Oracle H_2 . \mathcal{C} maintains a list $\mathcal{L}_2 = (X_i, TE_i, c_i, y_i, Y_i)$ which is initially empty. Upon receiving a query (X_i, TE_i) , \mathcal{C} answers as follows:

- If the query exists in \mathcal{L}_2 , the corresponding Y_i is returned.
- Otherwise, \mathcal{C} flips a coin $c_i \in \{0, 1\}$ such that $c_i = 0$ with probability μ and $c_i = 1$ with probability $1 - \mu$. Then, \mathcal{C} chooses $y_i \in_R \mathbb{Z}_p^*$. If $c_i = 0$, \mathcal{C} sets $Y_i = y_i B$. Otherwise, $c_i = 1$, it sets $Y_i = y_i P$. In both cases, \mathcal{C} inserts $(X_i, TE_i, c_i, y_i, Y_i)$ to \mathcal{L}_2 .

Queries on Oracle H_3 . \mathcal{C} maintains a list $\mathcal{L}_3 = (ct_{W_i}, TS_i, X_i, U_i, h_i)$ which is initially empty. Upon receiving a query $(ct_{W_i}, TS_i, X_i, U_i)$, \mathcal{C} returns the same value if the query has ever been made. Otherwise, \mathcal{C} chooses $h_i \in_R \mathbb{Z}_p^*$, returns h_i and adds $(ct_{W_i}, TS_i, X_i, U_i, h_i)$ to \mathcal{L}_3 .

User Secret Key Queries. Suppose \mathcal{A}_I makes a query on identity ID_i . \mathcal{C} answers as follows:

- If (ID_i, x_i, S_i, X_i) is included in \mathcal{L} , \mathcal{C} does the following:
 - If $x_i = \perp$, \mathcal{C} chooses $x_i \in_R \mathbb{Z}_p^*$, returns x_i , sets $X_i = x_i P$ and adds (ID_i, x_i, S_i, X_i) to \mathcal{L} .
 - Otherwise, $x_i \neq \perp$, \mathcal{C} returns x_i .
- If (ID_i, x_i, S_i, X_i) is not included in \mathcal{L} , \mathcal{C} chooses $x_i \in_R \mathbb{Z}_p^*$, returns x_i , sets $X_i = x_i P$ and adds (ID_i, x_i, S_i, X_i) to \mathcal{L} .

Partial Secret Key Queries. Suppose \mathcal{A}_I makes a query on (X_i, TE_i) . \mathcal{C} retrieves $(X_i, TE_i, c_i, y_i, Y_i)$ from \mathcal{L}_2 and answers as follows:

- If $c_i = 0$, \mathcal{C} returns failure.
- If $c_i = 1$ and (ID_i, x_i, S_i, X_i) is included in \mathcal{L} , \mathcal{C} does the following:
 - If $S_i \neq \perp$, \mathcal{C} returns S_i .
 - If $S_i = \perp$, we know $Y_i = y_i P$. \mathcal{C} sets $S_i = y_i P_{\text{pub}}$, returns S_i and adds (ID_i, x_i, S_i, X_i) to \mathcal{L} .
- If $c_i = 1$ and (ID_i, x_i, S_i, X_i) is not included in \mathcal{L} , \mathcal{C} sets $S_i = y_i P_{\text{pub}}$, returns S_i and adds (ID_i, x_i, S_i, X_i) to \mathcal{L} .

Public Key Queries. Suppose \mathcal{A}_I makes a public key query on identity ID_i . \mathcal{C} answers as follows:

- If (ID_i, x_i, S_i, X_i) is included in \mathcal{L} , \mathcal{C} retrieves $(X_i, TE_i, c_i, y_i, Y_i)$ from \mathcal{L}_2 and does the following:
 - If $c_i = 0$, \mathcal{C} updates (ID_i, x_i, S_i, X_i) in \mathcal{L} by setting $X_i = B$, $x_i = \perp$, and returns X_i .
 - If $c_i = 1$, \mathcal{C} does the following:

* If $X_i = \perp$, \mathcal{C} chooses $x_i \in_R \mathbb{Z}_p^*$, sets $X_i = x_i P$, returns X_i , and adds (ID_i, x_i, S_i, X_i) to \mathcal{L} .

* Otherwise, $X_i \neq \perp$, \mathcal{C} returns X_i .

- If (ID_i, x_i, S_i, X_i) is not included in \mathcal{L} , \mathcal{C} sets $S_i = \perp$, chooses $x_i \in_R \mathbb{Z}_p^*$, computes $X_i = x_i P$, returns X_i , and adds (ID_i, x_i, S_i, X_i) to \mathcal{L} .

Replace Public Key Queries. Suppose \mathcal{A}_I chooses a new public key X'_i for identity ID_i . \mathcal{C} answers as follows:

- If (ID_i, x_i, S_i, X_i) is included in \mathcal{L} , \mathcal{C} updates \mathcal{L} by setting $X_i = X'_i$ and $x_i = \perp$.
- If (ID_i, x_i, S_i, X_i) is not included in \mathcal{L} , \mathcal{C} sets $X_i = X'_i$, $x_i = \perp$, $S_i = \perp$ and adds (ID_i, x_i, S_i, X_i) to \mathcal{L} .

Signing Queries. Suppose \mathcal{A}_I makes a signing query on (ID_i, ct_{W_i}, TS_i) . Based on \mathcal{L} , \mathcal{L}_1 , \mathcal{L}_2 and \mathcal{L}_3 , \mathcal{C} can generate signatures for \mathcal{A}_I . If \mathcal{L} does not contain an item (ID_i, x_i, S_i, X_i) with $X_i \neq \perp$, \mathcal{C} performs a public key query on ID_i to get (x_i, X_i) . Then, \mathcal{C} retrieves the corresponding item $(X_i, TE_i, c_i, y_i, Y_i)$ from \mathcal{L}_2 , and does the following:

- If $c_i = 1$ and $x_i \neq \perp$, the signature can be directly generated. Specifically, \mathcal{C} performs:
 - Choose $r_i \in_R \mathbb{Z}_p^*$ and compute $U_i = r_i X_i$.
 - Generate $h_i = H_3(ct_{W_i} \| TS_i \| X_i \| U_i)$ based on the oracle H_3 .
 - Set $V_i = S_i + r_i h_i x_i P_{\text{pub}} + x_i Q$, where Q is obtained from \mathcal{L}_1 .
 - Return $\sigma_i = (U_i, V_i, X_i)$.

Obviously, σ_i is a valid signature on $ct_{W_i} \| TS_i$ because:

$$\begin{aligned} \hat{e}(V_i, P) &= \hat{e}(S_i + r_i h_i x_i P_{\text{pub}} + x_i Q, P) \\ &= \hat{e}(Y_i + h_i U_i, P_{\text{pub}}) \hat{e}(X_i, Q). \end{aligned}$$

- Otherwise, \mathcal{C} performs:

- Get $(X_i, TE_i, c_i, y_i, Y_i)$ from \mathcal{L}_2 . We know $Y_i = y_i B$.
- Choose $r_i, h_i \in_R \mathbb{Z}_p^*$ and compute $U_i = h_i^{-1}(r_i P - y_i X_i)$. Define $H_3(ct_{W_i} \| TS_i \| X_i \| U_i)$ as h_i . Note that if $H_3(ct_{W_i} \| TS_i \| X_i \| U_i)$ has already been defined as other value, \mathcal{C} returns failure.
- Set $V_i = r_i P_{\text{pub}} + c_Q X_i$, where c_Q is obtained from \mathcal{L}_1 .
- Return $\sigma_i = (U_i, V_i, X_i)$.

σ_i is a valid signature on $ct_{W_i} \| TS_i$ because:

$$\begin{aligned} \hat{e}(V_i, P) &= \hat{e}(r_i P_{\text{pub}} + c_Q X_i, P) \\ &= \hat{e}(r_i P, P_{\text{pub}}) \hat{e}(c_Q X_i, P) \\ &= \hat{e}(y_i X_i + h_i U_i, P_{\text{pub}}) \hat{e}(X_i, c_Q P) \end{aligned}$$

$$\begin{aligned}
&= \widehat{e}(y_i B + h_i U_i, P_{\text{pub}}) \widehat{e}(X_i, Q) \\
&= \widehat{e}(Y_i + h_i U_i, P_{\text{pub}}) \widehat{e}(X_i, Q).
\end{aligned}$$

Aggregate Signing Queries. It can be directly realized based on **Signing Queries** and the specification of SSH.

Forgery. Suppose \mathcal{A}_I outputs a forgery $(ct_{W_i}, TS_i, U_i, V_i, X_i)$. \mathcal{C} retrieves $(X_i, TE_i, c_i, y_i, Y_i)$ from \mathcal{L}_2 . If $c_i = 1$, \mathcal{C} returns failure. Otherwise, \mathcal{C} replays \mathcal{A}_I with a different choice of H_3 and the same choice of H_1 and H_2 . According the forking lemma, \mathcal{C} can get another forgery $(ct_{W_i}, TS_i, U_i, V'_i, X_i)$ within polynomial time, where $V'_i \neq V_i$ because $h'_i \neq h_i$ for the two choices of H_3 on $(t_{W_i}, TS_i, X_i, U_i)$.

Solving the CDH Problem. Suppose the forgeries are valid. Then,

$$\begin{aligned}
\widehat{e}(V_i, P) &= \widehat{e}(Y_i + h_i U_i, P_{\text{pub}}) \widehat{e}(X_i, Q), \widehat{e}(V'_i, P) \\
&= \widehat{e}(Y_i + h'_i U_i, P_{\text{pub}}) \widehat{e}(X_i, Q),
\end{aligned}$$

where Q is obtained from \mathcal{L}_1 . To be specific,

$$\begin{aligned}
\widehat{e}(V_i, P) &= \widehat{e}(S_i + r_i h_i x_i P_{\text{pub}} + x_i Q, P), \widehat{e}(V'_i, P) \\
&= \widehat{e}(S_i + r_i h'_i x_i P_{\text{pub}} + x_i Q, P).
\end{aligned}$$

Hence,

$$V_i = S_i + r_i h_i x_i P_{\text{pub}} + x_i Q, \quad (1)$$

$$V'_i = S_i + r_i h'_i x_i P_{\text{pub}} + x_i Q. \quad (2)$$

Multiplying both sides of Equation (1) by h_i^{-1} and both sides of Equation (2) by h'_i^{-1} , we have

$$h_i^{-1} V_i = h_i^{-1} S_i + r_i x_i P_{\text{pub}} + h_i^{-1} x_i Q, \quad (3)$$

$$h'_i^{-1} V'_i = h'_i^{-1} S_i + r_i x_i P_{\text{pub}} + h'_i^{-1} x_i Q. \quad (4)$$

Based on Equations (3) and (4), we have

$$h_i^{-1} V_i - h'_i^{-1} V'_i = (h_i^{-1} - h'_i^{-1})(S_i + x_i Q),$$

where $S_i = aY_i$ and $Y_i = y_i B$. Therefore, \mathcal{C} can solve the CDH instance by computing

$$abP = y_i^{-1} \left((h_i^{-1} - h'_i^{-1})^{-1} (h_i^{-1} V_i - h'_i^{-1} V'_i) - x_i Q \right).$$

Probability and Time Complexity. In the above process, \mathcal{C} succeeds if the following three events occur.

- E_1 : \mathcal{C} does not return failure in any partial secret key queries from \mathcal{A}_I .
- E_2 : The forgeries from \mathcal{A}_I are verified to be valid.
- E_3 : E_2 occurs and \mathcal{C} does not return failure.

Table 2

Comparisons of aggregate signature schemes.

Schemes	Signing	Verification	Aggregate Verification	KEFA [†]	BA	NR	PP
Selvi et al. (2012)	Exp + 2H	4Exp + 3H	$(7n + 1)\text{Exp} + (4n - 1)M + 2nH$	×	–	✓	×
Shen et al. (2017)	2SM + H	3Pair [‡] + 2H + M	$(n + 2)\text{Pair} + 4\text{Exp}_T + (n - 1)M + nM_T + (n + 2)H$	×	–	✓	×
Malhi and Batra (2015)	4SM + H	3Pair + 3SM + 2H	3Pair + 3nSM + 2H	✓	✓	✓	×
Xiong et al. (2013)	3SM + H	3Pair + 2SM + 2H	3Pair + 2nSM	✓	– ^a	– ^a	×
Tu et al. (2014)	3SM + 4H	4Pair + 2SM + 5H	4Pair + 2nSM	✓	✓	✓	– ^b
Li et al. (2018b)	5SM + 3H	3Pair + 2SM + 4H	3Pair + 2nSM + 4nH + M _T	✓	✓	✓	×
Ours	3SM + H	3Pair + SM + 2H	3Pair	✓	✓	✓	✓

[†] KEFA: Key-escrow Free Authentication; BA: Batch Authentication; NR: Non-repudiation; PP: Privacy Protection.

[‡] Pair: A bilinear pairing operation; Exp (resp. Exp_T): An exponentiation operation in group \mathbb{G} (resp. \mathbb{G}_T); SM: A scalar multiplication operation in group \mathbb{G} ; M (resp. M_T): A multiplication operation in group \mathbb{G} (resp. \mathbb{G}_T); H: A hash operation.

^a As shown in He et al. (2014), it cannot resist the Type II adversary.

^b It is based on the use of pseudonyms and hence limits the practical applications of the system.

It easily follows that $\Pr[E_1] \geq (1 - \mu)^{q_{ps}}$, $\Pr[E_2|E_1] \geq \epsilon_1$, and $\Pr[E_3|E_1 \wedge E_2] \geq \mu$. Accordingly,

$$\begin{aligned}
\Pr[E_1 \wedge E_2 \wedge E_3] &= \Pr[E_1] \cdot \Pr[E_2|E_1] \cdot \Pr[E_3|E_1 \wedge E_2] \\
&\geq (1 - \mu)^{q_{ps}} \epsilon_1 \mu.
\end{aligned}$$

For the optimal value of $\mu = \frac{1}{q_{ps} + 1}$, we know

$$\epsilon \geq \frac{\epsilon_1}{(q_{ps} + 1)e}.$$

In addition, $t \leq t_1 + \mathcal{O}(q_1 + q_2 + q_3 + q_{us} + q_{ps} + q_{pk} + q_{sig} + q_{as})T_m$. ■

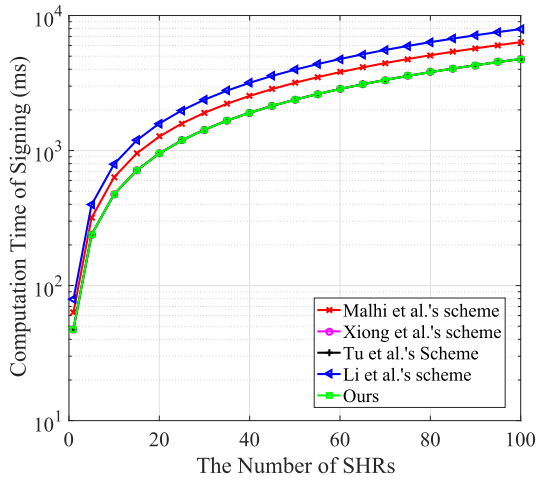
Theorem 2. For type II adversaries, SSH is existentially unforgeable against adaptively chosen-message attacks in the random oracle model under the CDH assumption.

Proof Sketch. As for type II adversaries, the *Replace Public Key Queries* cannot be made. In this case, the CDH instance can be integrated into the *Public Key Queries*. In addition, because the adversary is able to access the master secret key, the *Partial Secret Key Queries* are not needed. The other queries can be answered by the challenger similar to the case of type I adversaries. ■

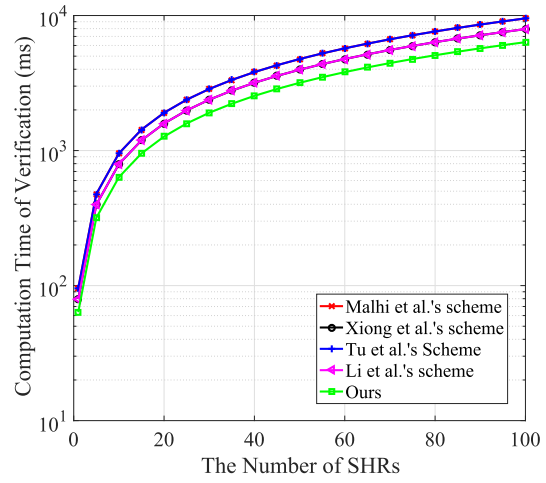
6. Performance evaluation

6.1. Performance analysis and feature comparison

In this section, we compare the performance and features of SSH and typical aggregate signature schemes including Selvi et al.'s scheme (Selvi et al., 2012), Shen et al.'s scheme (Shen et al., 2017), Malhi et al.'s scheme (Malhi and Batra, 2015), Xiong et al.'s scheme (Xiong et al., 2013), Tu et al.'s scheme (Tu et al., 2014), and Li et al.'s scheme (Li et al., 2018b). As shown in Table 2, we analyze the computation cost in terms of the basic cryptographic operations involved in signing, verification, aggregate verification. The features are taken into consideration such as key-escrow free, batch authentication, non-repudiation, and privacy protection. In the signing phase, the computation of our scheme is 3SM + H, which is less than 4SM + H of Malhi and Batra (2015), 3SM + 4H of Tu et al. (2014), and 5SM + 3H of Li et al. (2018b). Although the scheme (Selvi et al., 2012; Shen et al., 2017) only needs Exp + 2H and 2SM + H, respectively. These schemes cannot realize key-escrow free authentication and suffer severe efficiency drawbacks in aggregate verification. At a matter of fact, in verification, our scheme needs 3Pair + SM + 2H, which is less than 3Pair + 3SM + 2H of Malhi and Batra (2015), 3Pair + 2SM + 2H of Xiong et al. (2013), 4Pair + 2SM + 5H of Tu et al. (2014), and 3Pair + 2SM + 4H of Li et al. (2018b). In particular, in aggregate verification, the proposed scheme only performs three pairing operations, which is constant and far less than that of the other schemes. As for security features, only the schemes (Malhi and Batra, 2015; Xiong et al., 2013; Tu et al., 2014; Li et al., 2018b) and our SSH realize key-escrow free authentication.

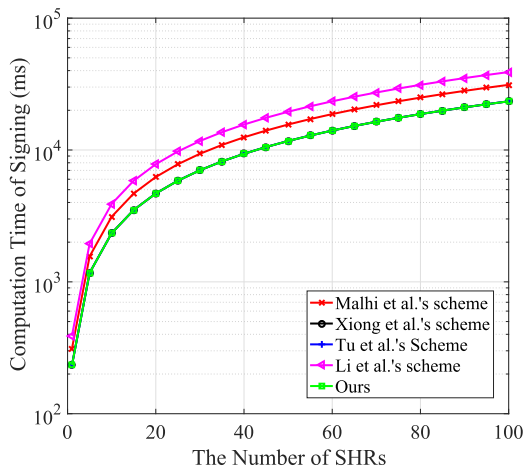


(a) Signing

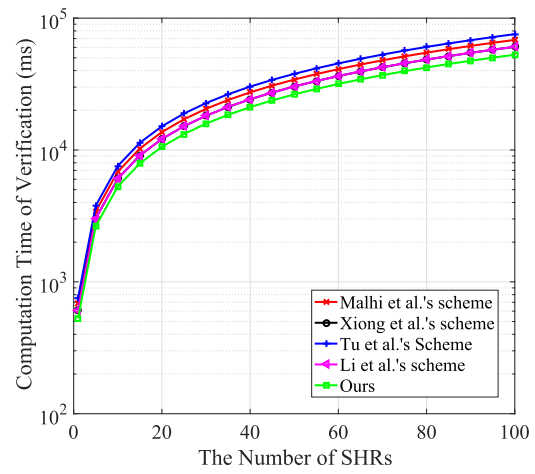


(b) Verification

Fig. 7. The computation cost based on a laptop.

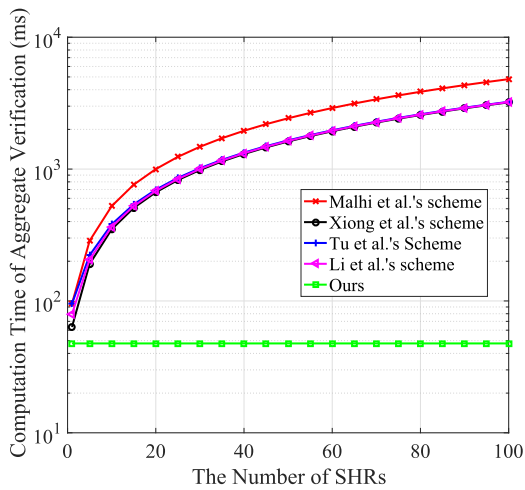


(a) Signing

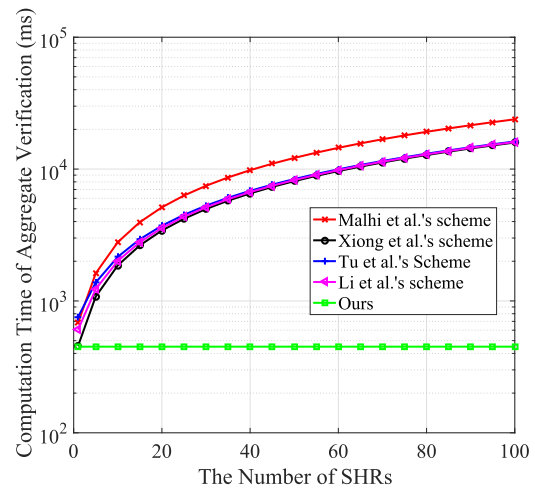


(b) Verification

Fig. 8. The computation cost based on a mobile phone.



(a) The Case of Laptops



(b) The Case of Mobile Phones

Fig. 9. The aggregate verification time.

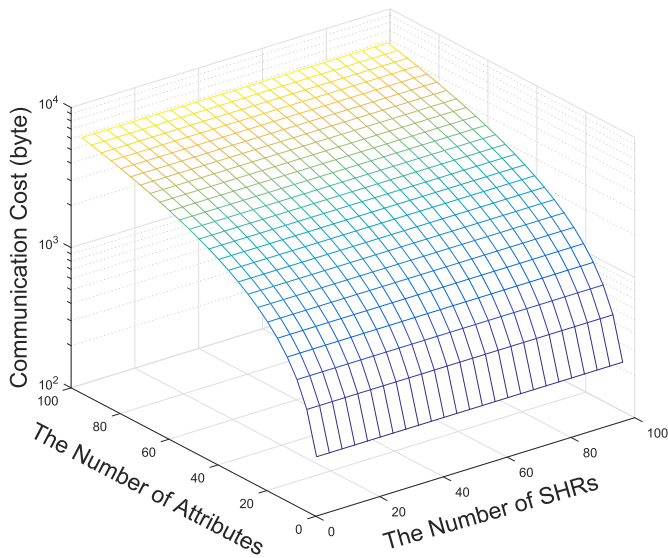
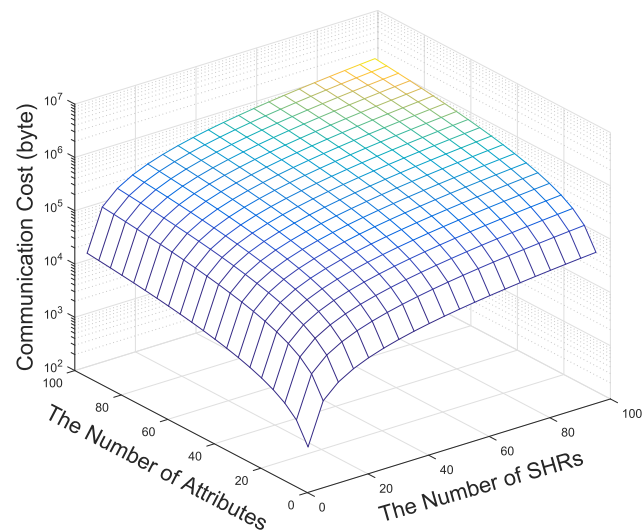


Fig. 10. The communication cost from RC to DO.

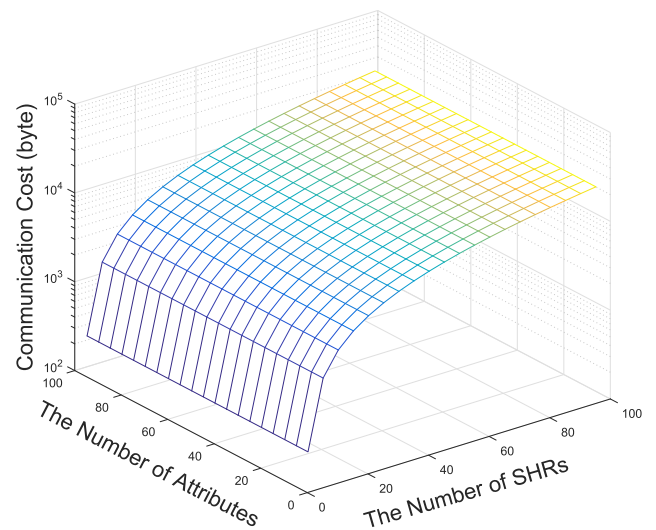
In He et al. (2014), the authors showed that the scheme (Xiong et al., 2013) cannot resist the Type II adversary, and hence the batch authentication and non-repudiation are not realized in essence. The schemes (Selvi et al., 2012; Shen et al., 2017; Malhi and Batra, 2015; Xiong et al., 2013; Li et al., 2018b) fail to realize privacy protection. The scheme (Tu et al., 2014) realizes privacy protection based on the trivial technique of pseudonyms and hence has limited practical applications. In general, only our SSH enables users' privacy protection and SHR confidentiality. In the following, we further compares the computation cost and communication overhead of SSH and the schemes (Malhi and Batra, 2015; Tu et al., 2014; Xiong et al., 2013; Li et al., 2018b).

6.2. Computation cost

In this section, we show the variation of signing time and verification time with the increase of SHRs. We conduct experiments based on a laptop and a mobile phone. The laptop-based experiment uses a virtual machine with 64 bit Ubuntu 16.04 LTS, which has two processors



(a) With SHR Confidentiality



(b) No SHR Confidentiality

Fig. 11. The communication cost from DO to CSP.

of Intel Core i7-7820HK CPU @ 2.9GHz and memory of 5.8GB. The mobile phone-based experiment is run on XiaoMi 5s with Qualcomm Snapdragon 821 of 4 processors and 2.15GHz. The operation system is MIUI v9.5 (Android 7.0). Note that, for clearness, the vertical axis is log scale in the following figures. As shown in Fig. 7, we compare the computation time of signing and verification of schemes (Malhi and Batra, 2015; Tu et al., 2014; Xiong et al., 2013; Li et al., 2018b) and SSH based on a laptop. In the Fig. 7a, when the number of SHRs is 1, the computation time is about 60 ms and the time increases to about 10 s when there are 100 SHRs. The signing time of SSH is lowest. In the Fig. 7b, when the number of SHRs is 1, the computation time is about 70 ms and the time increases with the number of SHRs. The verification time of SSH is lowest. In practical applications, DO and DU can be smart mobile devices, which are resource-constrained to some extent. Therefore, we further compare the computation time of signing and verification of schemes (Malhi and Batra, 2015; Tu et al., 2014; Xiong et al., 2013; Li et al., 2018b) and SSH based on a smartphone in Fig. 8. In the Fig. 8a, when the number of SHRs is 1, the computation time is about 300 ms and the time is more than 20 s when there are 100 SHRs. The verification time of SSH is lowest. In the Fig. 8b, when the number of SHRs is 1, the computation time is about 500 ms and the time increases to about 50 s when the number of SHRs is 100. The verification time of SSH is still lowest.

In the signing and verification, aggregation is not taken into account. In practice, if there are many SHRs, to improve the authentication efficiency and make the system suitable for time-sensitive use case such as emergency, it is necessary to perform aggregation verification. In Fig. 9, we compare the aggregate verification time of schemes (Malhi and Batra, 2015; Tu et al., 2014; Xiong et al., 2013; Li et al., 2018b) and SSH. The results based on a laptop and a mobile phone are shown in Fig. 9a and b, respectively. Obviously, in both cases, our SSH is most efficient and the computation time is small and almost constant. The aggregate verification time of the other schemes increases with the number of SHRs.

6.3. Communication overhead

Because only our SSH realizes users' privacy protection and SHR fine-grained access control, and SSH is most efficient in terms of the computation cost. In the following, we further show the communication overhead of SSH. In the registration phase, RC needs to return

secret keys to DO. In this case, we consider the communication overhead from RC to DO, which is introduced by the transmission of an attribute secret key and a partial secret key. In Fig. 10, we can see that the communication cost from RC to DO is only related to the number of attributes. In Fig. 11, we further consider the communication overhead from DO to CSP. According to the design of SSH, we know that DO uploads many SHRs to CSP for storage and sharing. We consider the cases of SHR confidentiality in Fig. 11a and no SHR confidentiality in Fig. 11b. In Fig. 11a, it follows that the communication overhead is determined by both the number of attributes and the number of SHRs. In Fig. 11b, we know that the communication overhead is only affected by the number of SHRs.

Generally speaking, the proposed scheme is more suitable for smart health based on IoT and cloud computing.

7. Conclusion

In this paper, we successfully addressed data security and user privacy issues in smart health by introducing SSH, a secure smart health system with privacy-aware aggregate authentication and access control for IoT. The main building blocks of SSH include an anonymous certificateless aggregate signature and an anonymous CP-ABE scheme. In SSH, users' identity information and sensitive attributes are hidden and hence privacy is preserved. The cloud service provider can check the uploaded SHRs from different users at one time, which is suitable for time-sensitive scenarios in smart health. In the random oracle model, we proved that SSH is existentially unforgeable against adaptively chosen-message attacks, which can prevent invalid SHRs from being uploaded to the cloud. Comprehensive theoretical analysis and extensive simulation results indicated that SSH is efficient in terms of computation cost and communication cost.

Acknowledgment

This work is supported by the National Key R&D Program of China (2017YFB0802000), the AXA Research Fund, the National Natural Science Foundation of China (Nos. 61772418, 61472472, 61402366), the Natural Science Basic Research Plan in Shaanxi Province of China (Nos. 2018JZ6001, 2015JQ6236). Yinghui Zhang is supported by New Star Team of Xi'an University of Posts & Telecommunications (2016-02).

References

Au, M.H., Mu, Y., Chen, J., Wong, D.S., Liu, J.K., Yang, G., 2007. Malicious kgc attacks in certificateless cryptography. In: Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security. ACM, pp. 302–311.

Boneh, D., Gentry, C., Lynn, B., Shacham, H., 2003. Aggregate and verifiably encrypted signatures from bilinear maps. In: International Conference on the Theory and Applications of Cryptographic Techniques. Springer, pp. 416–432.

Cai, Z., Yan, H., Li, P., Huang, Z.-a., Gao, C., 2017. Towards secure and flexible ehr sharing in mobile health cloud under static assumptions. *Cluster Comput.* 20 (3), 2415–2422.

Castiglione, A., De Santis, A., Masucci, B., Palmieri, F., Castiglione, A., Li, J., Huang, X., 2016. Hierarchical and shared access control. *IEEE Trans. Inf. Forensics Secur.* 11 (4), 850–865.

Castro, R., Dahab, R., 2007. Efficient certificateless signatures suitable for aggregation. *IACR Cryptol. ePrint Archive* 454, 1–24.

Chen, X., Li, J., Weng, J., Ma, J., Lou, W., 2016. Verifiable computation over large database with incremental updates. *IEEE Trans. Comput.* 65 (10), 3184–3195.

Chu, C.K., Chow, S.S., Tzeng, W.G., Zhou, J., Deng, R.H., 2014. Key-aggregate cryptosystem for scalable data sharing in cloud storage. *IEEE Trans. Parallel Distr. Syst.* 25 (2), 468–477.

Fan, L., Lei, X., Yang, N., Duong, T.Q., Karagiannis, G.K., 2017. Secrecy cooperative networks with outdated relay selection over correlated fading channels. *IEEE Trans. Veh. Technol.* 66 (8), 7599–7603.

Goyal, V., Pandey, O., Sahai, A., Waters, B., 2006. Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS'06. ACM, New York, pp. 89–98.

He, D., Tian, M., Chen, J., 2014. Insecurity of an efficient certificateless aggregate signature with constant pairing computations. *Inf. Sci.* 268, 458–462.

Jhaveri, R.H., Patel, N.M., Zhong, Y., Sangaiah, A.K., 2018. Sensitivity analysis of an attack-pattern discovery based trusted routing scheme for mobile ad-hoc networks

in industrial iot. *IEEE Access* 6, 20085–20103.

Jiang, T., Chen, X., Li, J., Wong, D.S., Ma, J., Liu, J.K., 2015. Towards secure and reliable cloud storage against data re-outsourcing. *Future Generat. Comput. Syst.* 52, 86–94.

Li, J., Li, Y.K., Chen, X., Lee, P.P., Lou, W., 2015. A hybrid cloud approach for secure authorized deduplication. *IEEE Trans. Parallel Distr. Syst.* 26 (5), 1206–1216.

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

Li, J., Yuan, H., Zhang, Y., 2018. Cryptanalysis and improvement for certificateless aggregate signature. *Fundam. Inf.* 157 (1–2), 111–123.

Liu, J., Cao, H., Li, Q., Cai, F., Du, X., Guizani, M., 2018. A large-scale concurrent data anonymous batch verification scheme for mobile healthcare crowd sensing. *IEEE Internet Things J.*, <https://doi.org/10.1109/JIOT.2018.2828463>.

Malhi, A.K., Batra, S., 2015. An efficient certificateless aggregate signature scheme for vehicular ad-hoc networks. *Discrete Math. Theor. Comput. Sci.* 17 (1), 317.

Sahai, A., Waters, B., 2005. Fuzzy identity-based encryption. In: Cramer, R. (Ed.), *Advances in Cryptology-EUROCRYPT'05*, Vol. 3494 of Lecture Notes in Computer Science. Springer, Berlin-Heidelberg, p. 557.

Selvi, S.S.D., Vivek, S.S., Shriram, J., Rangan, C.P., 2012. Identity based partial aggregate signature scheme without pairing. In: Sarnoff Symposium (SARNOFF), 2012 35th IEEE. IEEE, pp. 1–6.

Shen, L., Ma, J., Liu, X., Wei, F., Miao, M., 2017. A secure and efficient id-based aggregate signature scheme for wireless sensor networks. *IEEE Internet Things J.* 4 (2), 546–554.

Shen, J., Gui, Z., Ji, S., Shen, J., Tan, H., Tang, Y., 2018. Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks. *J. Netw. Comput. Appl.* 106, 117–123.

Shen, J., Wang, C., Li, T., Chen, X., Huang, X., Zhan, Z.-H., 2018. Secure data uploading scheme for a smart home system. *Inf. Sci.* 453, 186–197.

Shu, J., Jia, X., Yang, K., Wang, W., 2018. Privacy-preserving task recommendation services for crowdsourcing. *IEEE Trans. Serv. Comput.*, <https://doi.org/10.1109/TSC.2018.2791601>.

Shu, J., Liu, X., Jia, X., Yang, K., Deng, R.H., 2018. Anonymous privacy-preserving task matching in crowdsourcing. *IEEE Internet Things J.* 5 (4), 3068–3078.

Tu, H., He, D., Huang, B., 2014. Reattack of a certificateless aggregate signature scheme with constant pairing computations. *Sci. World J.* 2014, 1–10.

Wang, Q., Wang, C., Ren, K., Lou, W., Li, J., 2011. Enabling public auditability and data dynamics for storage security in cloud computing. *IEEE Trans. Parallel Distr. Syst.* 22 (5), 847–859.

Wang, H., Zheng, Z., Wu, L., Li, P., 2017. New directly revocable attribute-based encryption scheme and its application in cloud storage environment. *Cluster Comput.* 20 (3), 2385–2392.

Wang, C., Shen, J., Liu, Q., Ren, Y., Li, T., 2018. A novel security scheme based on instant encrypted transmission for internet of things. *Secur. Commun. Network.* 2018, 1–7.

Wu, Z., Tian, L., Li, P., Wu, T., Jiang, M., Wu, C., 2018. Generating stable biometric keys for flexible cloud computing authentication using finger vein. *Inf. Sci.* 433–434, 431–447.

Xhafa, F., Feng, J., Zhang, Y., Chen, X., Li, J., 2015. Privacy-aware attribute-based phr sharing with user accountability in cloud computing. *J. Supercomput.* 71 (5), 1607–1619.

Xiong, H., Guan, Z., Chen, Z., Li, F., 2013. An efficient certificateless aggregate signature with constant pairing computations. *Inf. Sci.* 219, 225–235.

Xu, J., Wei, L., Zhang, Y., Wang, A., Zhou, F., Gao, C.-z., 2018. Dynamic fully homomorphic encryption-based merkle tree for lightweight streaming authenticated data structures. *J. Netw. Comput. Appl.* 107, 113–124.

Yang, L., Han, Z., Huang, Z., Ma, J., 2018. A remotely keyed file encryption scheme under mobile cloud computing. *J. Netw. Comput. Appl.* 106, 90–99.

Zhang, Y., Chen, X., Li, J., Li, H., 2014. Generic construction for secure and efficient handoff authentication schemes in eap-based wireless networks. *Comput. Network.* 75, 192–211.

Zhang, Y., Zheng, D., Li, Q., Li, J., Li, H., 2016. Online/offline unbounded multi-authority attribute-based encryption for data sharing in mobile cloud computing. *Secur. Commun. Network.* 9 (16), 3688–3702.

Zhang, Y., Zheng, D., Chen, X., Li, J., Li, H., 2016. Efficient attribute-based data sharing in mobile clouds. *Pervasive Mob. Comput.* 28, 135–149.

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

Zhang, Y., Wu, A., Zheng, D., 2018. Efficient and privacy-aware attribute-based data sharing in mobile cloud computing. *J. Ambient Intell. Humanized Comput.* 9 (4), 1039–1048.

Zhang, Y., Deng, R.H., Shu, J., Yang, K., Zheng, D., 2018. TKSE: trustworthy keyword search over encrypted data with two-side verifiability via blockchain. *IEEE Access* 6 (1), 31077–31087.

Zhang, Y., Deng, R.H., Liu, X., Zheng, D., 2018. Blockchain based efficient and robust fair payment for outsourcing services in cloud computing. *Inf. Sci.* 462, 262–277.

Zhang, Y., Deng, R.H., Liu, X., Zheng, D., 2018. Outsourcing service fair payment based on blockchain and its application in cloud computing. *IEEE Trans. Serv. Comput.*, <https://doi.org/10.1109/TSC.2018.2864191>.

Zhang, Y., Zheng, D., Guo, R., Zhao, Q., 2018. Fine-grained access control systems suitable for resource-constrained users in cloud computing. *Comput. Inf.* 37 (2), 327–348.

Zhang, Y., Yang, M., Zheng, D., Lang, P., Wu, A., Chen, C., 2018. Efficient and secure big data storage system with leakage resilience in cloud computing. *Soft Comput.*, <https://doi.org/10.1007/s00500-018-3435-z>.

Zhang, Y., Shu, J., Liu, X., Li, J., Zheng, D., 2018. Security analysis of a large-scale concurrent data anonymous batch verification scheme for mobile healthcare crowd sensing. *IEEE Internet Things J.*, <https://doi.org/10.1109/JIOT.2018.2862381>.

Zhang, Y., Zheng, D., Deng, R.H., 2018. Security and privacy in smart health: efficient policy-hiding attribute-based access control. *IEEE Internet Things J.* 5 (3), 2130–2145.

Zhang, Y., Li, J., Zheng, D., Li, P., Tian, Y., 2018i. Privacy-preserving communication and power injection over vehicle networks and 5G smart grid slice. *J. Netw. Comput. Appl.* 122, 50–60.



Yinghui Zhang is currently an associate professor at National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, China. He obtained his Ph.D degree from Xidian University, China, in 2013. His current research includes cloud computing security, access control, security and privacy in IoT, etc. He has published more than 60 papers on the topics of cloud security, access control, and IoT security. He served for the program committee of several conferences and the editorial members of several international journals in information security. He is a member of IEEE and ACM.



Robert H. Deng is AXA Chair Professor of Cybersecurity and Professor of Information Systems in the School of Information Systems, Singapore Management University since 2004. His research interests include data security and privacy, multimedia security, network and system security. He served/is serving on the editorial boards of many international journals in security, including the *IEEE Transactions on Information Forensics and Security*, *IEEE Transactions on Dependable and Secure Computing*, and *IEEE Security and Privacy Magazine*. He is a fellow of the IEEE.



Gang Han is currently a Ph.D. student in Northwestern Polytechnical University, China. He received the B.S degree from Northwestern Polytechnical University, China, in 2013, and the M.S. degree from Northwestern Polytechnical University, China, in 2016. Now, he is pursuing his Ph.D. degree in Northwestern Polytechnical University. His current research includes security and privacy in the Internet of Things, cloud computing security, and wireless network security.



Dong Zheng received his Ph.D. degree in communication engineering from Xidian University, China, in 1999. He is currently a Professor at National Engineering Laboratory for Wireless Security (NELWS), Xi'an University of Posts and Telecommunications. His current research includes cloud computing security, wireless network security, code-based cryptography, access control, security and privacy in IoT, etc. He has published over 100 research articles including *CT-RSA*, *IEEE Transactions on Services Computing*, etc.