IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 24, 2019, accepted April 27, 2019, date of publication May 3, 2019, date of current version May 21, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2914862

A Data Sharing Protocol to Minimize Security and Privacy Risks of Cloud Storage in Big Data Era

SI HAN[1,2], **KE HAN**[3], **AND SHOUYI ZHANG**[4]
[1]Department of Science and Technology, China University of Political Science and Law, 102249 China
[2]School of Information Management for Law, China University of Political Science and Law, 102249 China
[3]School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100083, China
[4]School of Mechanical Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China

Corresponding author: Si Han (hansi@cupl.edu.cn)

This work was supported in part by the School Youth Fund Project of the China University of Political Science and Law under Grant 10818435

ABSTRACT A cloud-based big data sharing system utilizes a storage facility from a cloud service provider to share data with legitimate users. In contrast to traditional solutions, cloud provider stores the shared data in the large data centers outside the trust domain of the data owner, which may trigger the problem of data confidentiality. This paper proposes a secret sharing group key management protocol (SSGK) to protect the communication process and shared data from unauthorized access. Different from the prior works, a group key is used to encrypt the shared data and a secret sharing scheme is used to distribute the group key in SSGK. The extensive security and performance analyses indicate that our protocol highly minimizes the security and privacy risks of sharing data in cloud storage and saves about 12% of storage space.

INDEX TERMS Big data, security and privacy, cloud storage, data sharing.

I. INTRODUCTION

The emerging technologies about big data such as Cloud Computing [1], Business Intelligence [2], Data Mining [3], Industrial Information Integration Engineering(IIIE) [4] and Internet-of-Things [5] have opened a new era for future Enterprise Systems(ES) [6]. Cloud computing is a new computing model, in which all resource on Internet form a cloud resource pool and can be allocated to different applications and services dynamically. Compared with traditional distribute system, a considerable amount of investment saved and it brings exceptional elasticity, scalability and efficiency for task execution. By utilizing Cloud Computing services, the numerous enterprise investments in building and maintaining a supercomputing or grid computing environment for smart applications can be effectively reduced.

Despite these advantages, security requirements dramatically rise when storing personal identifiable on cloud environment [7], [8]. This raise regulatory compliance issues since migrate the sensitive data from federate domain to distribute domain. To take the benefit enabled by big data technologies, security and privacy issues [9], [10] must be addressed firstly.

Building security mechanism for cloud storage is not an easy task. Because shared data on the cloud is outside the control domain of legitimate participants, making the shared data usable upon the demand of the legitimate users should be solved. Additionally, increasing number of parties, devices and applications involved in the cloud leads to the explosive growth of numbers of access points, which makes it more difficult to take proper access control. Lastly, shared data on the cloud are vulnerable to lost or incorrectly modified by the cloud provider or network attackers. Protecting shared data from unauthorized deletion, modification and fabrication is a difficult task.

Conventionally, there are two separate methods to promote the security of sharing system. One is access control [11], in which only authorized user recorded in the access control table has the access privilege of the shared data. The other method is group key management [12]–[16] in which a group key is used to protect the shared data. Although access control makes the data only be accessed by legitimate participants, it cannot protect the attack from cloud providers. In the existing group key sharing systems, the group key is generally managed by an independent third party. Such methods assume that the third party is always honest. However, the assumption is not always real especially in the environment of cloud storage.

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

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

To address the security problem of sharing data on the cloud storage, a secret sharing group key management protocol is proposed in the paper and the following means are taken by our protocol to help detect or prevent frauds. Firstly, in order to make the shared data usable upon demand by the legitimate users, symmetric encryption algorithms [17] are used to encrypt the shared data. Once one data owner wants to share data with others, the decryption key is distributed to the legitimate sharers by the data owner. Secondly, the key used to decrypt the shared data controls the access permission for shared data. Asymmetric encryption algorithms [18] are used to encrypt the interactive message and makes only legitimate participants have the ability to decrypt the key. Thirdly, in case of shared data being known by unauthorized users, this protocol uses secret sharing scheme to assign key to the legitimate participants. By adding security mechanism to conventional service oriented clouds, we obtain a security aware cloud and guarantee the privacy of data sharing on cloud storage. Building security mechanism on cloud storage may accelerate the deployment of a cloud in mission critical business scenario.

The rest of the paper is organized as follow: Section 2 discusses the related work in brief; Section 3 presents the application scenario of our protocol and the security requirements; Section 4 presents the design of SSGK; Section 5 discusses the security properties, storage overload and computational overload of our protocol; Section 6 concludes the paper.

II. RELATED WORKS

Many solutions have been proposed to solve the privacy risks of cloud-based storage.

Rao [19] proposed a secure sharing schemes of personal health records in cloud computing based on ciphertext-policy attributed-based(CP-ABE) signcryption [20]. It focus on restricting unauthorized users on access to the confidential data. Liu *et al.* [21] proposed an access control policy based on CP-ABE for personal records in cloud computing as well. In [19] and [21],only one fully trusted central authority in the system is responsible for key management and key generation.

Huang *et al.* [22] introduced a novel public key encryption with authorized equality warrants on all of its ciphertext or a specified ciphertext. To strengthen the securing requirement, Wu *et al.* [23] proposed an efficient and secure identity-based encryption scheme with equality test in cloud computing. Xu *et al.* [24] proposed a CP-ABE using bilinear pairing to provide users with searching capability on ciphertext and fine-grained access control. He *et al.* [25] proposed a scheme named ACPC aimed at providing secure, efficient and fine-grained data access control in P2P storage cloud. Recently, Xue *et al.* [26] proposed a new framework, named RAAC, to eliminate the single-point performance bottleneck of the exiting CP-ABE based access control schemes for public cloud storage. While these schemes use identity privacy by using attribute-based techniques which fail to protect user attribute privacy.

The most recent work addressing the privacy issues in a cloud-based storage is carried out by Pervez *et al.* [27], who proposed a privacy aware data sharing scheme SAPDS. It combines the attribute based encryption along with proxy re-encryption and secret key updating capability without relying on any trusted third party. But the storage and communication overhead of SAPDS is decided by attribute encryption scheme.

In SSGK, an efficient solution is proposed to solve the secure problems of data sharing on the cloud storage without relying on any trust third party. Beyond using symmetric encryption algorithm [11] to encrypt the shared data, asymmetric algorithm [12] and secret sharing scheme [28], [29] is used to prevent the key used to decrypt the shared data from getting by unauthorized users. Secret sharing schemes were introduced by both Blakley [30] and Shamir [31] independently in 1979 as solution for safe guarding cryptography keys. In a secret sharing scheme, a secret is divided into n shares by a dealer and shared among n shareholders. Any t shares can reconstruct this secret. Chor *et al.* [32] extended the notion of the original secret sharing and presented a notion of verifiable secret sharing (VSS). The property of verifiability means that shareholders are able to verify whether their shares are consistent.

III. SECURED CLOUD STORAGE IN BIG DATA ERA

In this section, we define the application scenario of our protocol and security requirements.

A. CLOUD STORAGE FOR BIG DATA

The architecture of cloud based big data is illustrated in Figure.1. It consists of three parts: source data, cloud center and services. Between source data and cloud center layer, unstructured or semi-structured source data is structured. They includes processing methods such as data collection [33], data mining [34] and data aggregation [35]. The processed source data is stored on cloud in relational or NoSQL databases [36]. Lastly, service layer answers information requests submitted by consumers by integrating information stored in cloud.

Beyond allowing customers to put all data into cloud, cloud storage provides all kinds of data services for customers. Because scale horizontally runs on cheap commodity hard in a distributed configuration and there is no need for customers to purchase and maintain their own IT facilities, cloud based big data stores brings in inherent availability, scalability and cost effectiveness.

B. AN EXAMPLE OF HEALTHCARE INFORMATION SYSTEM

Cloud storage provides not just low cost, but high scalability and availability. It may be a natural solution to some of problems in storing and analyzing the increasing patients' medical records [37]. For healthcare providers, only based on the aggregation of all patients' medical records, could proper diagnosis be made. Reference [38] proposed a cloud based platform for healthcare. Cloud storage provides a common

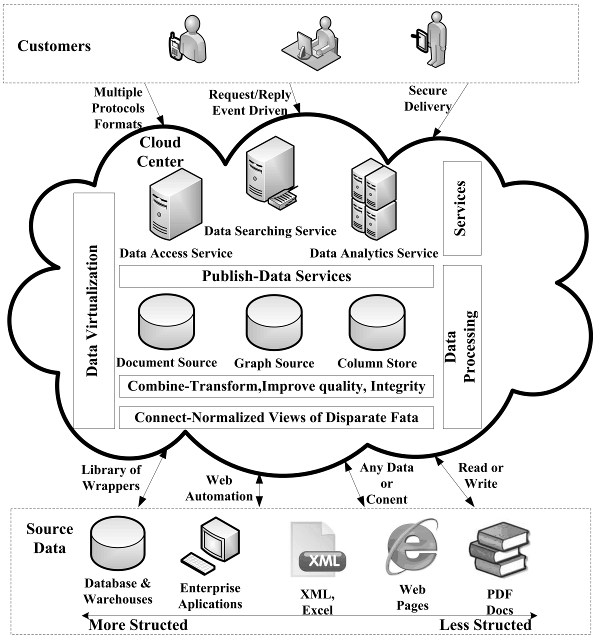

FIGURE 1. Cloud storage architecture for big data.

place for storing medical records which overcome the delay of transferring medical records between different healthcare providers and make diagnostic process more efficient.

The e-healthcare cloud provides many advantages in collaboration and data sharing among healthcare providers. Nevertheless, in consider of the highly privacy of medical data it comes with significant risks of medical records.

Firstly, medical records are shared on the public channel where many attackers on the channel to eavesdrop the medical records.

Additionally, due to the increasing number of parties, devices and applications involved in cloud, unauthorized parties or cloud providers may have the ability to access shared medical records.

Last but not the least, some authorized parties may work together to get some unauthorized medical records illegally.

E-healthcare services require a security mechanism to protect the privacy of medical records.

IV. THE PROPOSED SSGK PROTOCOL

In this section we describe more about the proposed protocol model and algorithm of SSGK.

A. PROTOCOL MODEL

1) DATA SHARING MODEL

Consider a cloud storage data sharing system with multiple entities and the data sharing model is shown as Figure.2. The protocol model consists of three types of entities: cloud provider, data owner and group members.

The cloud provider: provides a public platform for data owners to store and share their encrypted data. The cloud provider doesn't conduct data access control for owners. The encrypted data can be download freely by any users.

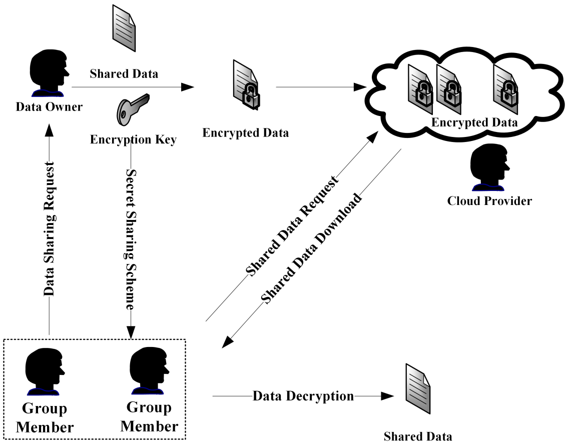

FIGURE 2. Data protocol model of the proposed SSGK.

Data owner: defines the access policy and encrypts its data with a symmetric encryption algorithm using a group key. The group members who satisfied the access policy constitute a sharing group. Then secret sharing scheme is used by the owner to distribute the encryption key to the sharing group.

Group members: every group member including the data owner is assigned with an unique and a pair of keys. The group members can freely get any interested encrypted data from the public cloud. However the user can decrypt the data if and only if it get the data decryption key from the data owner.

2) SECURITY MODEL

In SSGK, we have the following assumptions:

The data owner is totally trusted and will never be corrupted by any adversaries. Cloud provider is semi-trusted, it correctly executes the task assigned to them for profits, but they would try to find out as much secret information as possible based on the data owners uploaded data.

We now describe the security model of SSGK by listing possible attacks.

The group key is distributed by running the secret sharing scheme. Parts of the group members can gather their sub-secret shares to reconstruct the group key.

Moreover, the communication channel of our protocol is defined as: Every pair of participants have a point-to-point channel to send messages. Additionally, all the participants access to a broadcast channel: when a participant puts a message m on this channel, all the other participants receive m. The group key is distributed on the public channel and the key may be tempered by adversaries.

B. DEFINITIONS AND NOTATIONS

Definition 1 ((t,n)VSS): A verified secret sharing scheme contains four steps:

Sharing Generation Algorithm: An algorithm that, on input a security parameter K and a random polynomial $f(x)$ of degree $t-1$, output n sub-shares and a verified value v;

Distribution: The dealer distributes each sub-share and v to every scheme participant secretly;

TABLE 1. **Notations.**

Notation	Description for the Notation
O	data owner
D	shared data
K	group key used to encrypt shared data
P_i	Group member P with indentified ID_i
$cipher(x)$	Cipher-text of x
$E_k(P)$	encrypt P with key k using encryption algorithm E
$E^{-1}{}_k(C)$	decrypt C with key k using decryption algorithm E^{-1}
SK_i	Secret key of P_i
PK_i	Public key of P_i

Verify: A verification algorithm that, on input a sub-share and v, output whether the sub-share is tempered during distribution;

Secret Reconstructed: For any t sub-shares, the security parameter K can be reconstructed.

Definition 2 (Equity and Availability): Verified secret sharing scheme guaranteeing equity and availability with two conditions: Any participant set in the share group, where the size of the set is less than the total quantity, the participants in the set cannot get any information about K; Only with cooperation of all the legitimate participants, K could be reconstructed.

Definition 3 (Confidentiality): Verified secret sharing scheme guarantees confidentiality if any users outside the sharing group cannot get any information of K even with the knowledge of enough interactive messages.

Definition 4 (Integrity): Once the interactive messages are tempered during VSS, any information about K could be gotten by participants. We said that verified secret sharing scheme guarantees integrity.

The notations in Table 1 are used throughout the remainder of this paper.

C. PROTOCOL DETAILS

The scene describes as a protocol participant O wishes to share data D with the legitimate participants $P_i, i = 1, 2, \ldots, n$. Firstly, O generates a secret key K and uses K to encrypt D, then O stores the encrypted data $cipher(D)$ to the cloud. Secondly, O shares K with the legitimate participants and all participants work together to certify and reconstruct K. Finally, every participant gets K and downloads $cipher(D)$ from the cloud. The detailed process of SSGK is shown as Figure.3.

1) KEY DISTRIBUTION AND DATA SHARING

The data owner O creates the secret key and encrypts the data using symmetric encryption algorithm AES. Then secret sharing scheme is used by O to distribute the secret key. As the public channel is available for communications between every pair of participants, an asymmetric encryption algorithm RSA is used to protect the key sub-shares from known by unauthorized users. The distribution protocol is summarized as followed steps and shown as Figure.4.

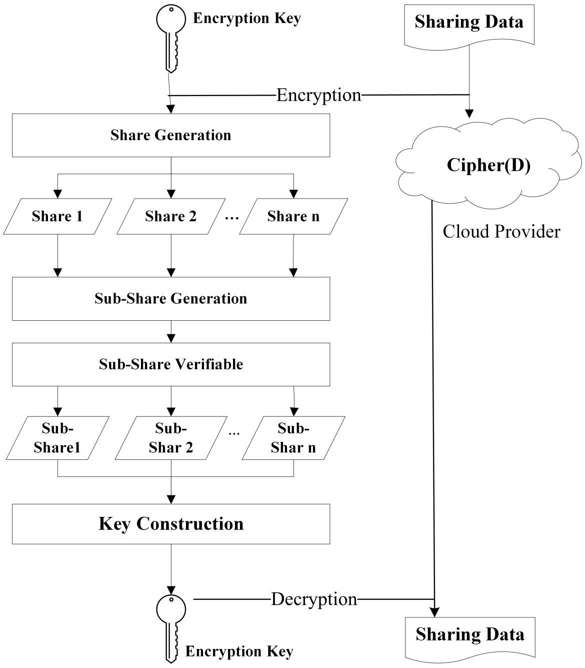

FIGURE 3. **Data sharing model of the proposed SSGK protocol.**

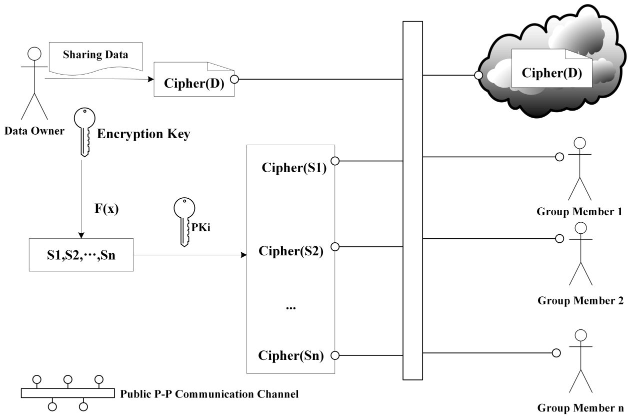

FIGURE 4. **Data sharing model of the proposed SSGK protocol.**

Step 1: Every participant produces a pair of PK_i, SK_i and it sends PK_i to the cloud provider.

Step 2: The data owner O produces group key K randomly and it encrypts the shared data D using equation 1; Then it uploads $Cipher(D)$ to the cloud.

$$Cipher(D) = AES_K(D) \tag{1}$$

Step 3: The data owner generates a random polynomial $F(x)$ of degree $n-1$: $F(x) = a_0 + a_1x + a_2x^2 + \ldots a_{n-1}x^{n-1}$ where $a_0 = k$ and $a_1, a_2, \ldots, a_n$ are produced randomly by data owner with the same size of K. $F(x)$ is used to share the group key K using secret sharing scheme.

Step 4: According to the secret sharing scheme, the data owner computes n sub-shares $s_1, s_2, \ldots, s_n$, and the verified element V, v using equation 2, 3 and 4. After calculation, the data owner gets the public key PK_i of P_i, $i = 1, 2, \ldots, n$ from cloud provider and encrypts every sub-share s_i using equation 5 with public key PK_i of U_i. Then, the encrypted sub-share and v are sent to ${}_i$, $i = 1, 2, \ldots, n$ throng point to point public channel.

$$s_1 = F(1), s_2 = F(2), \ldots, s_{n-1} = F(n-1) \qquad (2)$$

$$V = s_1 \times s_2 \times \ldots s_n. \qquad (3)$$

$$v = Vmoda_0 \qquad (4)$$

$$Cipher(s_i) = RSA_{PK_i}(s_i) \qquad (5)$$

After key distribution protocol, every participant may get an encrypted sub-share s_i.

2) KEY RECONSTRUCTION AND VERIFICATION

All the participants may get $Cipher(D)$ and v from the point-to-point public channel. The next goal of the participants is to reconstruct K with collaboration and to verify whether there are any corrupted participants. The steps of the reconstruction protocol are described as follows:

Step 1: Every participant $P_i, i = 1, 2, \ldots, n$ except for O interchanges their sub-shares. P_i receive $cipher(s_i)$ and v from O through public channel and decrypts $Cipher(s_i)$ using equation 6. P_i gets the public key PK_i of $P_j, j = (i+1)modn, (i+2)modn, \ldots, (i+n-1)modn$ from cloud provider; If the value of j is 0, j would be changed to n. After encrypted s_i, it sends the encrypted s_i to $P_j, j = (i+1)modn, (i+2)modn, \ldots, (i+n-1)modn$. If the value of j is 0, j would be changed to n.

$$Cipherj(s_i) = {RSA^{-1}}_{PK_j}(s_i) \qquad (6)$$

After these steps, every participant P_i may receive $n-1$ sub-shares from other participants, the sub-shares received by every participant are shown as Table 2.

Step 2: Every participant $P_i, i = 1, 2, \ldots, n$ except for O calculates K_i, a copy of K, and V using Lagrange Interpolating Formula. Then, it computes $V \bmod K_i$, if $VmodK_i \neq v$, it will broadcast that there are compromised participants; Otherwise, it will send its ID encrypted with s_j to the cloud provider for data access permission. As P_i calculates the group key K, it can download the $Cipher(D)$ from cloud and decrypts it by K using equation 7.

$$D = {AES^{-1}}_K(Cipher(D)) \qquad (7)$$

Through above steps, K used to encrypt the shared data is distributed to the participants secretly through public channel and shared data is decrypted by authorized participants. Our sharing protocol protects the shared data from being known by the cloud provider and unauthorized users.

V. SECURITY ANALYSIS AND PERFORMANCE

A. SECURITY ANALYSIS

Risks exist in both the sharing data distribution phase and the regular broadcast phase. In this section we address some security properties of SSGK by showing some theorems.

TABLE 2. Interactive secret sub-shares.

Notation	Description for the Notation
P_1	$Cipher_1(s_2), Cipher_1(s_3), ..., Cipher_1(s_n)$
P_2	$Cipher_2(s_3), Cipher_2(s_4), ..., Cipher_2(s_n)Cipher_2(s_1)$
...	...
P_i	$Cipher_i(s_{i+1}), Cipher_i(s_{i+2}), ..., Cipher_i(s_{i-1})$
...	...
P_n	$Cipher_n(s_1), Cipher_n(s_2), ..., Cipher_n(s_{n-1})$

Theorem 1 (Lagrange Polynomial): Given a set of $k+1$ data points $(x_0, y_0), \ldots, (x_j, y_j), \ldots, (x_t, y_t)$, where no two x_j are the same, interpolation polynomial in the Lagrange form is a linear combination.

$$F(x) = \sum_{j=0}^{t} y_i l_i(x) \qquad (8)$$

Theorem 2: SSGK could guarantee equity and availability.

Proof of Throrem 2: In order to prove that our protocol guarantees equity and availability, according to definition 3, we need to prove that the online available m participants cannot get any information about K when $m < n$. Where n stands for the capacity of this sharing group.

There are m participants online, they cloud calculate m sub-shares $s_1, s_2, \ldots, s_m$ using equation 2. The group key $K = F(0) = a_0$ can be calculated as follows using determinant operation:

step1:

$$\begin{pmatrix} 1^0 & 1^1 & 1^2 & \ldots & 1^n \\ 2^0 & 2^1 & 2^2 & \ldots & 2^n \\ & & \ldots & & \\ m^0 & m^1 & m^2 & \ldots & m^n \end{pmatrix}, \quad |A| = \begin{cases} = 0, & \text{if } m < n \\ \neq 0, & \text{if } m = n \end{cases} \qquad (9)$$

step2:

$$A \begin{pmatrix} a_0 \\ a_1 \\ \ldots \\ a_n \end{pmatrix} = \begin{pmatrix} s_0 \\ s_1 \\ \ldots \\ s_m \end{pmatrix} \qquad (10)$$

The value of K which not be calculated as A^{-1} does not exist when $m < n$, so that the available m participants cannot get any information of K. When $|A| \neq 0$ and $m = n$, a_i, $i = 1, 2, ..., n$ cloud be calculated by step 3. And every participant gets K.

step3:

$$\begin{pmatrix} a_0 \\ a_1 \\ \ldots \\ a_n \end{pmatrix} = A^{(-1)} \begin{pmatrix} s_0 \\ s_1 \\ \ldots \\ s_m \end{pmatrix} \qquad (11)$$

Theorem 3: SSGK guarantees confidentiality.

Proof of Theorem 3: SSGK guarantees confidentiality by protecting encrypted shared data from decrypting by unauthorized users. There are two kinds of unauthorized users: cloud provider and attacker on the public channel.

TABLE 3. Security comparison of ACPC, RAAC, SAPDS and SSGK.

Schemes	ACPC	RAAC	SAPDS	SSGK
Data Equity Protection	√	×	×	√
Data Confidentiality Protection	√	√	√	√
Data Integrity Protection	×	×	×	√
Conclusion Resistance	×	√	×	√
Trusted Third Party	Honest But Curious	Honest	Honest But Curious	Honest But Curious
Channel Condition	Secure Channle	Secure Channel	Public Channle	Public Channel

As all the communication and intermediate information are transmitted on the public channel, an easy way to attack our protocol by unauthorized users is to capture the intermediate information. The information of $Cipher(s_i)$, v could be tempered by unauthorized users.

Suppose that there is an unauthorized user on the public channel and received all the intermediate information. A normal way to reconstruct K is brute force attack. There are two different methods to reconstruct K. One is identifying n sub-shares from $n \times n$ sub-shares and getting private key of one participant. Another is identifying $m \times n$ sub-shares from $n \times n$ sub-shares and getting private keys of m participants.

In the first method, with the knowledge of the public key's size L, the probability of unauthorized users getting the private key of one participant is $\frac{1}{2^L}$ and the probability of identifying n sub-shares from $n \times n$ sub-shares is $\frac{1}{C^n_{n\times n}}$. Unauthorized users guess K with the probability P_1 which can be ignored.

$$P_1 = \frac{1}{C^n_{n\times n}} \times \frac{1}{2^L} \tag{12}$$

In the other method, the probability of unauthorized users getting the public keys of m participants is $\frac{1}{2^{m\times L}}$ and the probability of identifying $m \times n$ sub-shares from $n \times n$ sub-shares is $\frac{1}{C^{n\times n}_{m\times n}}$. The unauthorized users guess K with the probability P_2 which can be ignored as well.

$$P_2 = \frac{1}{C^{m\times n}_{n\times n}} \times \frac{1}{2^{L\times m}} \tag{13}$$

Due to its mathematical property, RSA is vulnerable to cipher-text attacks. The unauthorized user may obtain the group key using mathematical or chosen-cipher-text attack. But in our protocol, under the secure access control of cloud provider, only with the correct pair of PK and ID, the encrypted shared data can be obtained.

The shared data stored on the cloud is encrypted using AES algorithm. As the security performance of AES is excellent and unknown attack methods can attack non-linear components, we conclude that shared data could not be decrypted by cloud provider.

Theorem 4: SSGK could guarantee integrity.

Proof of Theorem 4: In order to prove that our protocol guarantees integrity, we must make sure that the shared data cloud not be decrypted if there is a corrupted participant.

Now, we assume that P_i decides to crack our protocol. P_i sends the wrong value of $Cipher_{i-1}(s_i)$ to P_{i-1} while it gets the correct $Cipher_i(s_{i+1})$ from P_{i+1}.P_{i-1} gets the wrong K and calculates the verifiable value which is not equal to v. P_{i-1} broadcasts the protocol is abandoned and no one could get the shared data from cloud.

B. PERFORMANCE

In this section, we provide the performance assessment of the proposed protocol. The following experiments focus on the storage and computation overhead of SAPDS and the proposed protocol SSGK. These experiments are running on a server with Intel core 2, 2.93GHz Dual Core processor and 2GB RAM.

1) COMPARISON ON SECURITY

This section puts forwards detailed comparison on various security and functionality features of the proposed scheme with some recently developed CP-ABE based schemes. For comparison, we consider related schemes ACPC [25], RAAC [26], and SAPDS [27]. Table3 tabulates the comparison results on various security attributes. It is noted that our scheme supports many useful properties, such as data equity, confidentiality and integrity protection, collusion resistance and privacy protection.

2) STORAGE OVERHEAD

The storage overhead of ACPC, RAAC, SAPDS and SSGK is tested in order to compare their scalability. The number of private and public keys of these schemes are counted. We assume that the number of the group participants is n and the key size is L bits.

Private keys, represent the storage consumption on one group participants in protocol.

In ACPC, secret key and user attributes are used to compute the encryption key.

In RAAC, multiple CAs are used for key generation, four kinds of different keys are kept by users: the symmetric algorithm key to decrypt shared data, user's secret attribute based key, user attributes and CA verified keys(Six CAs are simulated in our experiment).

In SAPDS, three kinds of keys are kept to achieve fine-grained access control over the shared data: key used to decrypt shared data, users' secret attribute-based key of the access tree and user attributes.

In SSGK, only secret key and sub-share are used to compute the encryption key.

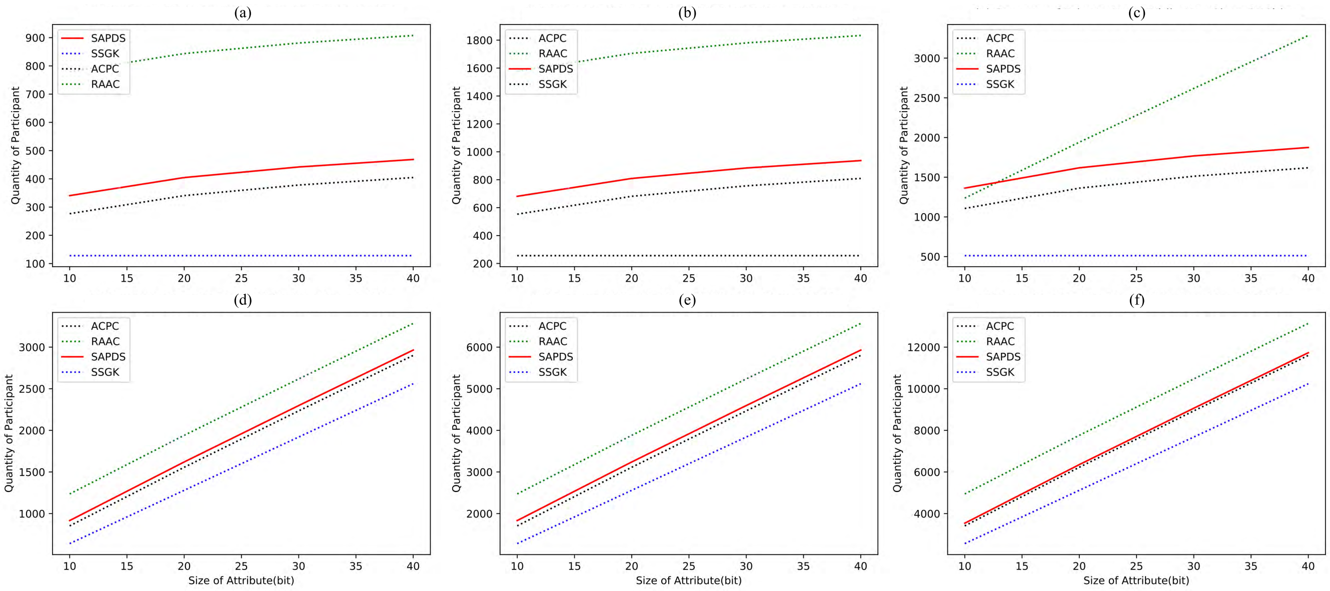

FIGURE 5. Storage overhead of ACPC,RAAC, SAPDS and SSGK.

We assume the number of attributes is $log_2 n$ and the size of attributes key is L bit.

$$NUM_s(ACPC) = (1 + log_2 n) \times Lbits \quad (14)$$

$$NUM_s(RAAC) = (1 + 1 + log_2 n + 6)Lbits \quad (15)$$

$$NUM_s(SAPDS) = (1 + 1 + log_2 n) \times Lbits \quad (16)$$

$$NUM_s(SSGK) = 2 \times Lbits \quad (17)$$

Figure.5(a)(b)(c) shows the change of storage overhead of these schemes with the capacity of participant varies from 10 to 50 and the size of attribute size varies from 64bits to 256bits. Statistics indicate that when the size of attribute grows, the difference value rises fast, while the number of participant has letter influence on storage cost in SSGK.

Public keys, represent the storage overhead on cloud provider.

In ACPC, the cloud provide stores n public keys for users and attributes of an access tree.

In RAAC, the cloud provides stores attributes of an access tree, public keys of the group participants and six CAs.

In SAPDS, cloud provider stores the encrypted key used to decrypt the shared data, public keys of the group participants and attributes of an access tree.

In SSGK, only the public keys of group participants are stored.

$$NUM_p(ACPC) = (n + log_2 n) \times Lbits \quad (18)$$

$$NUM_p(RAAC) = (n + 6 + log_2 n) \times Lbits \quad (19)$$

$$NUM_p(SAPDS) = (n + 1 + log_2 n) \times Lbits \quad (20)$$

$$NUM_p(SSGK) = (n \times L)bits \quad (21)$$

With the growing quantity of participants, the total public key size of these schemes are shown as Fig5(d)(e)(f). As the statistics shown, about 12% of storage overhead is saved by SSGK.

3) COMMUNICATION OVERHEAD

The first step in SSKG is generating a key K, a random polynomial $F(x)$ of degree n and computing n sub-shares. The first task of ACPC, RAAC and SAPDS is defining an access policy which the secret K is concealed. The following step in these four schemes is retrieving the secret key K. In SSGK, participants execute secret sharing scheme to distribute K and CP-ABE is executed by the participants of ACPC, RAAC and SAPDs. Key generation and distribution are needed by the schemes to get K. The communication overhead on these schemes contains two aspects: key generation times and key distribution times. And the time is decided by the capacity of the sharing group.

Figure.6(a) illustrates the time taken by ACPC, RAAC, SAPDS and SSGK to generate key. The generation time is shown to be depended on the number of group members. It takes at most 14s to generate group key by RAAC, however it just take at most 143ms by SSGK. About 98 percent of generation time is saved.

Figure.6(b) illustrates that the distribution times with six different participants over 128bits of and 512bits of participants' public key. Shown in the figure, associated with the quantity of participants, ACPC,RCCA and SAPDS tends to consume more times compared with SSGK.

Figure.6(c) shows the total computation cost with six different participants. About 95% of total times are saved by SSGK than RAAC.

SSGK does not rely on any third party to govern the key management. Schemes based on CP-ABE delegate key management to cloud server may raised same security

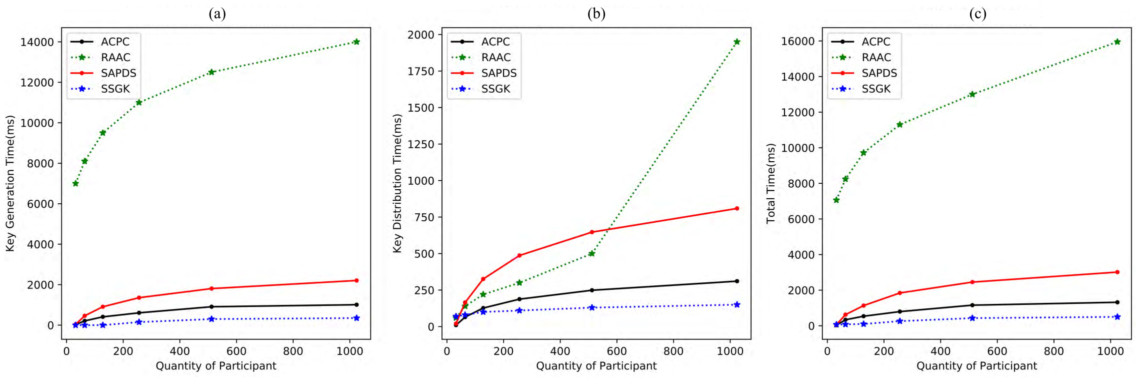

FIGURE 6. Communication overhead of ACPC, RAAC, SAPDS and SSGK.

problems as mentioned before. And the simulation results show that our protocol incurs less storage computation overhead.

The evaluation result highlights the fact that SSGK enables the owner to maintain fine-grained control over the outsourced data with minimal expenses.

VI. CONCLUSION

In this paper, we propose a novel group key management protocol for the data sharing in the cloud storage. In SSGK, we uses RSA and verified secret sharing to make the data owner achieve fine-grained control over the outsourced data without relying on any third party. In addition, we give detailed analysis of possible attacks and corresponding defenses, which demonstrates that GKMP is secure under weaker assumptions. Moreover we demonstrate that our protocol exhibits less storage and computing complexity.

Security mechanism in our scheme guarantees the privacy of grids data in cloud storage. Encryption secures the transmission on the public channel; verified security scheme make the grids data only accessed by authorized parties. The better performance in terms of storage and computation make our scheme more practical.

The problem of forward and backward security in group key management may require some additions to our protocol. An efficient dynamic mechanism of group members remains as future work.

REFERENCES

[1] P. Zhao, W. Yu, S. Yang, X. Yang, and J. Lin, "On minimizing energy cost in Internet-scale systems with dynamic data," *IEEE Access*, vol. 5, pp. 20068–20082, 2017.

[2] D. Wu, G. Zhang, and J. Lu, "A fuzzy preference tree-based recommender system for personalized business-to-business E-services," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 1, pp. 29–43, Feb. 2015.

[3] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 97–107, Jan. 2014.

[4] X. Shi, L. X. Li, L. Yang, Z. Li, and J. Y. Choi, "Information flow in reverse logistics: An industrial information integration study," *Inf. Technol. Manage.*, vol. 13, no. 4, pp. 217–232, Dec. 2012.

[5] N. Bizanis and F. A. Kuipers, "SDN and virtualization solutions for the Internet of Things: A survey," *IEEE Access*, vol. 4, pp. 5591–5606, May 2016.

[6] S. Li, L. Xu, X. Wang, and J. Wang, "Integration of hybrid wireless networks in cloud services oriented enterprise information systems," *Enterprise Inf. Syst.*, vol. 6, no. 2, pp. 165–187, Nov. 2012.

[7] K.-Y. Teng, S. A. Thekdi, and J. H. Lambert, "Risk and safety program performance evaluation and business process modeling," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 6, pp. 1504–1513, Nov. 2012.

[8] Z. Fu, X. Sun, S. Ji, and G. Xie, "Towards efficient content-aware search over encrypted outsourced data in cloud," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.

[9] J. Han, W. Susio, Y. Mu, and J. Hou, "Improving privacy and security in decentralized ciphertext-policy attribute-based encryption," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 3, pp. 665–678, Mar. 2015.

[10] D. Zou, Y. Xiang, and G. Min, "Privacy preserving in cloud computing environment," *Secur. Commun. Netw.*, vol. 9, no. 15, pp. 2752–2753 Oct. 2016.

[11] Y. Tang, P. P. C. Lee, John C. S. Lui, and R. Perlman, "Secure overlay cloud storage with access control and assured deletion," *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 6, pp. 903–916, Nov./Dec. 2012.

[12] J. Shao, R. Lu, and X. Lin, "Fine-grained data sharing in cloud computing for mobile devices," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr./May 2015, pp. 2677–2685.

[13] 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.

[14] S. Tanada, H. Suzuki, K. Naito, and A. Watanable, "Proposal for secure group communication using encryption technology," in *Proc. 9th Int. Conf. Mobile Comput. Ubiquitous Netw.*, Oct. 2016, pp. 1–6.

[15] J. Zhou *et al.*, "Securing outsourced data in the multi-authority cloud with fine-grained access control and efficient attribute revocation," *Comput. J.*, vol. 60, no. 8, pp. 1210–1222, Aug. 2017.

[16] R. Ahuja, S. K. Mohanty, and K. Sakurai, "A scalable attribute-set-based access control with both sharing and full-fledged delegation of access privileges in cloud computing," *Comput. Elect. Eng.*, vol. 57, pp. 241–256, Jan. 2017.

[17] J. Thakur and N. Kumar, "AES and blowfish: Symmetric key cryptography algorithms simulation based performance analysis," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 1, no. 2, pp. 6–12, Dec. 2011.

[18] E. Fujisaki, T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," *J. Cryptol.*, vol. 26, no. 1, pp. 80–101, Jan. 2013.

[19] Y. S. Rao, "A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing," *Future Gener. Comput. Syst.*, vol. 67, pp. 133–151 Feb. 2017.

[20] S. Jin-Shu, C. Dan, W. Xiao-Feng, and S. Yi-Pin, "Attributed-based encryption schemes," *J. Softw.*, vol. 22, no. 6, pp. 1299–1315, 2011.

[21] H. liu, Y. huang, and J. K. Liu, "Secure sharing of Personal Health Records in cloud computing: Ciphertext-Policy Attribute-Based Signcryption," *Future Gener. Comput. Syst.*, vol. 52, pp. 67–76, Nov. 2015.

[22] K. Huang *et al.*, "PKE-AET: Public key encryption with authorized equality test," *Comput. J.*, vol. 58, no. 10, pp. 2686–2697, Oct. 2015.

[23] L. Wu, Y. Zhang, K.-K. R. Choo, and D. He, "Efficient and secure identity-based encryption scheme with equality test in cloud computing," *Future Gener. Comput. Syst.*, vol. 73, pp. 22–31, Aug. 2017.

[24] Q. Xu, C. Tan, Z. Fan, W. Zhu, Y. Xiao, and F. Cheng, "Secure multi-authority data access control scheme in cloud storage system based on attribute-based signcryption," *IEEE Access*, vol. 6, pp. 34051–34074, 2018.

[25] H. He, R. Li, X. Dong, and Z. Zhang, "Secure, efficient and fine-grained data access control mechanism for P2P storage cloud," *IEEE Trans. Cloud Comput.*, vol. 2, no. 4, pp. 471–484, Oct./Dec. 2014.

[26] K. Xue *et al.*, "RAAC: Robust and auditable access control with multiple attribute authorities for public cloud storage," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 4, pp. 953–967, Apr. 2017.

[27] Z. Pervez, A. M. Khattak, S. Lee, and Y.-K. Lee, "SAPDS: Self-healing attribute-based privacy aware data sharing in cloud," *J. Supercomput.* vol. 62, no. 1, pp. 431–460, Oct. 2012.

[28] Y. Tian, J. Ma, C. Peng, and Q. Jiang, "2 Fair (t, n) threshold secret sharing scheme," *IET Inf. Secur.*, vol. 7, no. 2, pp. 106–112, Jun. 2013.

[29] L. Harn and C. Lin, "Strong (n, t, n) verifiable secret sharing scheme," *Inf. Secur.*, vol. 180, no. 6, pp. 3059–3064, Aug. 2010.

[30] G. R. Blakley, "Safeguarding cryptographic keys," in *Proc. AFIPS 79th Nat. Comput. Conf.*, Jun. 1979, pp. 313–317.

[31] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[32] B. Chor, S. Goldwasser, S. Micali, B. Awerbuch, and S. sharing, "Verifiable secret sharing and achieving simultaneity in the presence of faults," in *Proc. 26th IEEE Symp. Found. Comput. Sci.*, Oct. 1985, pp. 383–395.

[33] N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Data collection and wireless communication in Internet of Things (IoT) using economic analysis and pricing models: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2546–2590, 4th Quart., 2016.

[34] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.

[35] M. Rezvani, A. Ignjatovic, E. Bertino, and S. Jha, "Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 1, pp. 98–110, Jan./Feb. 2015.

[36] J. R. Lourenço, V. Abramova, B. Cabral, J. Bernardino, P. Carreiro, and M. Vieira, "No SQL in practice: A write-heavy enterprise application," in *Proc. IEEE Int. Congr. Big Data*, Jun./Jul. 2015, pp. 584–591.

[37] V. Casola, A. Castiglione, K. K. Choo, and C. Esposito, "Healthcare-related data in the cloud: Challenges and opportunities," *IEEE Cloud Comput.*, vol. 3, no. 6, pp. 10–14, Apr. 2016.

[38] T. Muhammed, R. Mehmood, A. Albeshri, and I. Katib, "UbeHealth: A personalized ubiquitous cloud and edge-enabled networked healthcare system for smart cities," *IEEE Access* vol. 6, pp. 32258–32285, 2018.

SI HAN was born in Suzhou, Anhui, China, in 1988. She received the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), China, in 2015. She is currently a lecturer with the China University of Political Science and Law. Her research interests mainly include information security, cloud computing, and the Internet of Thing.

KE HAN was born in Heze, Shangdong, China, in 1980. He received the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), China, in 2008, where he is currently an Associate Professor. His research interests mainly include electronics and telecommunications.

SHOUYI ZHANG was born in Linfen, Shanxi, China, in 1988. He received the M.S. degree from Beijing Jiaotong University (BJTU), China, in 2014, where he is currently pursuing the Ph.D. degree in mechanical engineering. His research interest includes manufacturing system optimization.