Hindawi
Security and Communication Networks
Volume 2019, Article ID 7541269, 16 pages
https://doi.org/10.1155/2019/7541269

Research Article

Secure Information Sharing System for Online Patient Networks

Hyun-A Park

Department of Medical Health Sciences, Kyungdong University, Republic of Korea

Correspondence should be addressed to Hyun-A Park; kokokzi@naver.com

Received 9 August 2018; Revised 31 January 2019; Accepted 14 February 2019; Published 12 March 2019

Guest Editor: Chunhua Su

Copyright © 2019 Hyun-A Park. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, privacy emerged as a hot issue again, as the General Data Protection Regulation (GDPR) of EU has become enforceable since May 25, 2018. This paper deals with the problem of health information sharing on a website securely and with preserving privacy. In the context of patient networks (such as 'PatientsLikeMe' or 'CureTogether'), we propose the model Secure Information Sharing System (SISS) with the main method of group key cryptosystem. SISS addresses important problems of group key systems. (1) The new developed equations for encryption and decryption can eliminate the rekeying and redistribution process for every membership-change of the group, keeping the security requirements. (2) The new 3D Stereoscopic Image Mobile Security Technology with AR (augmented reality) solves the problem of conspiracy by group members. (3) SISS uses the reversed one-way hash chain to guarantee forward secrecy and backward accessibility (security requirements for information sharing in a group). We conduct a security analysis of SISS according to group information sharing secrecy and an experiment on its performance. Consequently, although current IT paradigm is changing to be more and more 'complicated', 'overlapped', and 'virtualized', SISS makes it possible to securely share sensitive information from collaborative work.

1. Introduction

Patients around the world want to connect to people who are suffering from the same symptoms and try to find the best treatments. These days, there are some online patient websites for health information sharing such as "PatientsLikeMe"[1] or "CureTogether"[2], where patients talk about their symptoms and successful (or failed) experiences. Researchers can also discover new and better solutions based on the patient-contributed data.

1.1. Problem Identification. The problem is that these kinds of health data may be sensitive and private information. Therefore, patients want the sensitive health data to be protected and managed with safety, and the data to be revealed to only limited persons. That is, the individuals' right to privacy and control over the circulation of their information [3]. In particular, privacy emerged as a hot issue again, as the General Data Protection Regulation (GDPR) of EU has become enforceable since May 25, 2018.

One of the substantial ways to share patients' information and to protect privacy is a group key management system. However, a group key system has some peculiar characteristics; the group key should be updated whenever members leave or join the group. This is called rekeying and redistribution. These processes need a complicated and high level of security. Another feature that differentiates an information sharing system from other general group key management systems is that leaving members cannot access the group's information anymore, but joining members can access all the previous group's information to get more information for better treatment. Moreover, current members may conspire with leaving members or other people by revealing their group key [4].

In this paper, the proposed model SISS (Secure Information Sharing System) addresses the above problems in the context of online patient networks such as PatientsLikeMe or CureTogether. The main methods are group key management system, including the newly developed encryption/decryption algorithm and reversed hash key chain, and 3D Stereoscopic Image Mobile Security Technology

(augmented reality technique). Namely, that is the secure version for the websites which can make people with the same symptoms share their health information safely and securely.

1.2. Main Goals. The SISS's principle to manage users' information with safety is that general information should be presented in plaintexts but sensitive information should be managed in ciphertexts. The solution for sharing ciphertexts with users is group key management system. The following is the main goals of SISS's group key system.

There Should Be No Processes of Rekeying and Redistribution. Whenever membership-changes (users leaving or joining groups) happen in the general group key system, the system should generate a new group key and distribute the new key to all members very quickly. This process needs more complicated and secure level of techniques, so 'rekeying and redistribution' falls under the hard problem in group key systems. To solve this problem, we develop new equations of the encryption and decryption with the group key. Therefore, SISS has no process of rekeying and redistribution for membership-changes. To the best of my knowledge, this is the first trial to eliminate the process of rekeying and redistribution keeping the security of the group key.

SISS Guarantees Forward Secrecy and Backward Accessibility. The representative security requirements for general group key systems are forward secrecy and backward secrecy. However, the security requirements of SISS to share information with group members are forward secrecy and backward accessibility, not backward secrecy. The reversed hash chain can guarantee the security properties of SISS.

SISS is Collusion-Resistant with New Technology. In every group member system, one of the important security problems is the potential for conspiracy between users and illegal members. As a solution, SISS proposes a new concept of 3D Stereoscopic Image Mobile Security Technology using VR/AR technique.

1.3. Methods and Contributions. The main methods and contributions are listed in detail as follows:

(1) The group key system for encryption and decryption: Only valid users can share the secret information with their group keys and pseudonyms. Others (invalid users) cannot know the contents and ownership for the information: 'what about' and 'whose information'.

(2) SISS addresses the security problem with reversed hash chain and 3D Stereoscopic Image Mobile Security Technology. And, the newly developed encryption and decryption algorithms are used for efficiency. It is because the information sharing has been and will be highly increased in the networked collaborative computing environments.

(2.1) *Equations for Encryption and Decryption*: SISS has no need for rekeying and redistribution for membership-changes. The principle of group key generation for each member is that a fixed master group key is assigned to each group, and random numbers for each user and every session are newly generated by applying random numbers to hash function respectfully, reversedly, and repeatedly s times. Thereafter, each random number and the master group key are combined according to the developed equation algorithm. Then, a total of five subgroup keys are generated as each member's group session key for every session. Hence, every member has completely different group keys for each session and each member because of generating different random number each time. However, one of the most important things is that the results after calculation of all other group keys for the developed equation (for encryption and decryption) are the same as the result of master group key for encryption and decryption, which makes it possible for SISS not to have rekeying and redistribution processes whenever there are membership-changes.

(2.2) *3D Stereoscopic Image Mobile Security Technology*: It is a new concept of a security solution using VR/AR techniques against conspiracy. The combination of human's facial expressions and gestures which are identified at the registration time of a legitimate group member (LGM) should be rendered as a 3D image in the login process to authenticate a legitimate user. Consequently, the rendering of users' own facial expressions and gestures can prohibit conspiracy between users and invalid persons.

(2.3) *Reversed Hash Chain*: SISS should guarantee forward secrecy and backward accessibility, along with group key secrecy [5], which are security requirements in group information sharing system. In SISS, every member's group key is generated based on reversed one-way hash chain. Due to one-way properties of reversed hash function[6], a leaving member cannot know the next group key (forward secrecy) but a joining member can know all the previous keys and information (backward accessibility). Therefore, SISS is suitable for secret collaborative work and sensitive information sharing among group members.

(2.4) *All Different Random Numbers for Each Member and Each Session*: In SISS system, every member's group session key looks like private key because their group key has completely different values with different random numbers for each member. Nevertheless, the key plays a role of a group key, with which every group member can share their sensitive health information with other members in a group.

(3) Stronger authentication for login: The proposed model uses LGM (legitimate group member) for stronger authentication. Moreover, the authentication processes are mutual, so that the proposed model is secure against spoofing or masquerading attack.

(4) Scalability to other group project systems and mobile phone applicability: SISS is scalable to other group project systems on websites. Although application scenario is about patient networks on the web, SISS is extendable to other secure group projects. Furthermore, it is applicable to even LGM of mobile phones, because the next smartphone will feature a front-facing 3D laser scanner for facial recognition, which was the expectation of upcoming iPhone 8 in early 2018 [7] (but Apple released it without 3D laser scanner in late 2017).

(5) Privacy preserving system: SISS can meet the privacy requirements: pseudonymity (partial anonymity), unlinkability, and unobservability.

(6) Blinding: In every flow, SISS uses newly generated random numbers. This masking method does not allow an attacker to know or to guess real contents correctly.

2. Related Works and Application

2.1. Related Works. In this section, we introduce our main solution's research area of group key.

Many researchers have worked on group key in various views such as group key agreement, exchange, revocation, and multicast/broadcast. However, this paper only focuses on the group key application for sharing information among multiple users. In particular, SISS has a little different property from the general group key in that the security requirement of SISS is not backward secrecy (a joining member cannot know all the previous keys and information) but backward accessibility (a joining member can know all the previous keys and information). It is caused by the different application's goal from other group key systems, which can enable current group members to share all their information securely. That is the reason why SISS is related to the research area of keyword search schemes for multiuser setting.

At first, we address the researches about 'multiuser setting'. In [8], Park et. al firstly proposed privacy preserving keyword-based retrieval protocols for dynamic groups (multiuser setting) based on reversed hash key chain. In multiuser setting environments like companies and municipal offices, a server contains heterogeneous documents accessible by different groups or persons. A leaving member from a group should not have access to any documents of the group anymore, that is, forward secrecy. Because a newly joining member should perform the tasks of the group to which he belongs, all documents accessible to the group must be still available and he should be able to obtain all the group keys, that is, backward accessibility. They accomplished both security properties with reversed hash key chain. Thereafter, they made the formal definition for 'backward accessibility' firstly in [5], where they proposed two practical group search schemes in the respect of efficiency in cloud datacenter. As for the researches not based on reversed hash key chain, Wang et. al. [9] analyzed Park et. al's scheme[8] on various views including security and efficiency. They achieved backward accessibility for multiple users with public key and Pallier's cryptosystem instead of reversed hash key chain. The next year, Wang et. al proposed another scheme of keyword field-free conjunctive keyword searches on encrypted data in the dynamic group setting [10]. In [11], Li et al. suggested a new effective fuzzy keyword search in a multiuser system over encrypted cloud data. This system supports differential searching and privileges based on the techniques attribute-based encryption and Edit Distance.

In respect of key-updating and redistribution, there have been also many researches until now, because group key's rekeying and redistribution to all group members are complicated and hard tasks. Generally, this research area is divided into four categories: a centralized distribution scheme, a distributed key agreement scheme, a hybrid group key management scheme, and a self-healing group key distribution scheme (SGKD). The significant point of centralized key distribution schemes [12–20] is that a centralized key management center as a trusted party generates, updates, and distributes the group keys to all members. As to distributed key agreement schemes [21–28], all group members participate in generation, rekeying, and redistribution of their group keys. Hybrid group key management schemes [29–32] are about the best use of both schemes: a centralized distribution scheme and a distributed key agreement scheme. Lastly, a self-healing group key distribution scheme (SGKD) is for wireless networks including sensor networks [33–50]. In this paper, we focus on SGKD schemes because SISS's application includes both networks of wired and wireless and SISS has more relations with SGKD schemes than other schemes.

The main point of self-healing schemes is that group members can recover the missing session keys without retransmission of the missing messages from the GM (group manager). A SGKD scheme is largely divided into four categories again: polynomial based SGKD (P-SGKD) schemes [33–37], vector space secret sharing based SGKD schemes [38–41], bilinear pairings based SGKD schemes [42, 43], and exponential arithmetic based SGKD (E-SGKD) schemes [44–51]. P-SGKD and vector space secret sharing based SGKD schemes are generally known as "not efficient". A polynomial secret sharing scheme is the most common technique, where two types of polynomials are constructed: the revocation polynomial and the access polynomial. In [44], Rams et al. pointed out that almost all of the polynomial based SGKD schemes can be converted to E-SGKD schemes. P-SGKD schemes can be divided into two types based on using Lagrange Interpolation or not. In [33, 45–47] E-SGKD schemes are constructed from P-SGKD schemes with Lagrange Interpolation. The other P-SGKD schemes without Lagrange Interpolation can be classified into two types again based on using hash chains or not. Scheme 3 in [48] and Scheme 2 in [49] are E-SGKD schemes transformed from revocation polynomial based P-SGKD schemes without hash chains. In [50], Gou et. al first proposed E-SGKD scheme with high efficiency and backward secrecy by combining dual chains: a traditional hash chain and a key chain.

2.2. Entities and Application Scenario. SISS has three parties: users, SM (security manager), and SISS server. SM (security manager) is a kind of a client, which is granted a special role of a security manager. SM is assumed as a TTP (trusted third party) and it is located in front of the SISS server. SM controls group key and key-related information, all sensitive information, and all other events with powerful computational and storage abilities. Figure 1 shows the system configuration of SISS.

The main participants of SISS are group members who want to get help through information sharing. The information scope is health conditions and patient profile. Mostly, the health information could be shared but some secret personal data in patient profile should be revealed to the allowed people only.

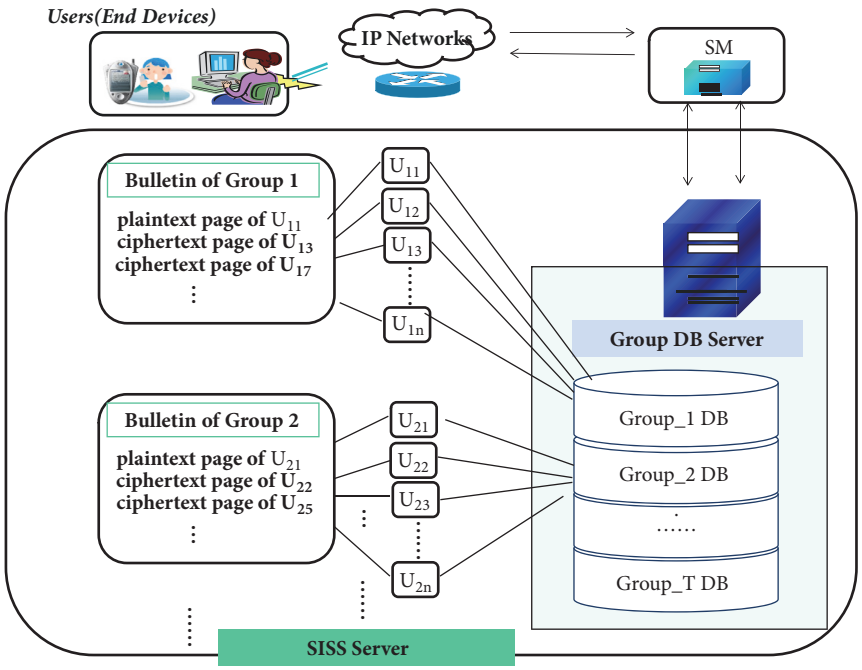


Figure 1: The configuration of SISS.

With the mobile devices or PC, Secure Information Sharing System (SISS) is constructed for online website patients with any disease all over the world. The needs are as follows: Many patients want to get in such kind of patient networks and to be helped more easily and securely. Each member uploads his/her conditions or information to his/her personal ciphertext pages (for sensitive data) or plaintext pages (for public data).

One more important thing is that the augmented reality (AR) technique of 3D image rendering is applied to external devices. The rendered image is selected from the contents-list consisting of the randomly repeated and rearranged human's facial expressions and gestures. As the first process, every user should register at SM; thereafter they should get through the authentication processes every session and then they start some actions. When some sensitive information is shared with other patients (it means that the ciphertext pages are generated), we know the page is encrypted by the group's encryption key. Only the legitimate users (who registered at SM and kept the information given by SM at their devices for authentication) can pass authentication processes and know the sharing information. In the last step of the authentication, 3-dimensional image is rendered. This image can be called a legitimate group member.

3. Preliminaries

3.1. Notations. The notations for SISS are explained in Table 1.

3.2. Algorithms for SISS Model. A SISS model consists of the following eight algorithms.

(1) $SysPrm(1^k)$: Parameter Generation algorithm $SysPrm$ takes as an input a security parameter k and produces a system parameter λ.

(2) $KeyGen(\lambda)$: Taking λ as input, Key Generation algorithm $KeyGen$ produces group session random numbers set RH, group member keys set K, and member pseudonym keys set P.

(3) $InfGenStr(\lambda, RH, K, P)$: Taking λ, RH, K, P as input, Information Generation and Storage algorithm $InfGenStr$ produces LGM (legitimate group member) and other information for authentication.

(4) $Q_r(RH, K, P)$: Given RH, K, P, Query algorithm produces Query Value Q_v.

(5) V_r-$Q_r(Q_v)$. Given Q_v, Verification algorithm verifies Q_v, and Query algorithm produces Query Value Q_v.

(6) $V_r(Q_v)$: Given Q_v, Verification algorithm verifies Q_v.

(7) $EncUp(M)$: This algorithm encrypts and uploads message M.

(8) $DnDec(M)$: This algorithm downloads and decrypts message M.

3.3. Security Building Blocks and Model

Definition 1 (one-way hash chain). It is generated by selecting the last value at random and applying it to one-way hash function h repeatedly. The initially chosen value is the last value of the key chain. Following are two properties of one-way hash chain.

Property 2. Anybody can deduce the earlier value k_i belonging to the one-way key chain with the later value k_j by checking $h^{j-i}(k_j)$ which equals k_i with the later value k_j.

Property 3. Given the latest released value k_i of one-way key chain, an adversary cannot find a later value k_j such that $h^{j-i}(k_j)$ equals k_i. Even when value k_{i+1} is released, the second preimage collision resistant property prevents an adversary from finding k'_{i+1} different from k_{i+1} such that $h(k_{i+1})$ equals k_i [6].

Remark. We call property 1 of one-way hash key chain 'backward accessibility' and property 2 'forward security'.

Definition 4 (PRF (pseudorandom function)). We say that '$F : K_f \times X \longrightarrow Y$ is (t, q, e)-secure pseudorandom function' if every oracle algorithm A making at most q oracle queries and with running time at most t has advantage $Adv_A < e$. The advantage is defined as $Adv_A = |Pr[A^{F_k} = 1] - Pr[A^{g} = 1]|$ where g represents a random function selected uniformly from the set of all maps from X to Y, and where the probabilities are taken over the choice of k and g [51].

Definition 5 (PRG G_r (pseudorandom generator)). We say that '$Gr : K_{Gr} \longrightarrow S$ is a (t, e)-secure pseudorandom generator' if every algorithm A with running time at most t has advantage $Adv_A < e$. The advantage is defined as $Adv_A = |Pr[A(Gr(U_{K_{Gr}})) = 1] - Pr[A(U_S) = 1]|$, where $U_{K_{Gr}}, U_S$ are random variables distributed uniformly on K_{Gr}, S [51].

Definition 6 (DDH (decisional Diffie-Hellman)). Let G be a group of prime order q and g a generator of G. The DDH problem is to distinguish between triplets of the form (g^a, g^b, g^{ab}) and (g^a, g^b, g^c), where a, b, c are random elements of $\{1, \ldots, q-1\}$.

Definition 7 (collusion resistance). The leaving member $U_i^l \in G^l$ $(1 \le i \le n, 1 \le l < l' \le s)$ colluding with the members in the after sessions $U_{i'}^j \in G^j$ $(1 \le i' \le n, j \ge l', i \ne i')$ cannot recover $K_{Gi}^{\ j}$ even knowing $\{\alpha_i^l, K_{Gi}^{\ l}\}$ and $\{\alpha_{i'}^j, K_{Gi'}^{\ j}\}$ [50], where G^l is a group G at l-th session.

Definition 8 (security game ICR-IS). (Indistinguishability of Ciphertexts from Random Bit Strings in Information Sharing)

Setup. The challenger C creates a ciphertext set B of m pages $\in \{CP_{it}\} \in \{B_t\}$ $(1 \le i \le n, 1 \le t \le T)$ and gives this to the adversary A. A chooses a polynomial number of subsets from B. This collection of subsets is called B^*; C runs algorithm $SysPrm, KeyGen, InfGenStr$ and encrypts each subset running algorithm $Q_r, V_rQ_r, V_r, EncUp$. Finally, C sends A all ciphertexts with their associated subsets.

Queries. A may request the encryption $EncUp(B^*))$ of any B and any verification V_r.

Challenge. A chooses a B_0 and its subsets such that none of the algorithms (Q_r, V_rQ_r, V_r) given in the step Queries distinguishes B_0 from $B_1 = \text{rand}(B_0)$. The challenger C chooses a random bit b and gives $EncUp(B_b)$ to A. A again asks for encrypted pages and their verifications with the restriction that A may not ask for the algorithm that distinguishes B_0 from B_1. The total number of ciphertexts and verifications is in k.

Response. A outputs $b_A \in \{0, 1\}$. If $b_A = b$, A is successful. In security game ICR-IS, adversary's advantage is defined as $Adv(1^k) = |Pr[b_A = b] - 1/2|$.

Table 1: Notations.

SM	Security Manager
K_G	the group keys set of group G
G_t	the *t-th* group
T	the total number of groups in a SISS server
mK_g	the master group key of group G
$CP_{i,t}^j$	the j-th ciphertext-page of a member i in a group t
n	the total number of group G's members
sn	session number
m	the total number of ciphertext-page CP
$U_{t,i}^j$	a member i of the *t-th* group in the *j-th* session
$K_{Gt,i}^{\ j}$	a group session key for each member i of *t-th* group G_t in the *j-th* session $sK_{i,1}^j, sK_{i,2}^j, sK_{i,3}^j, sK_{i,4}^j, sK_{i,5}^j$: five subkeys for a group session key $K_{Gi}^{\ j}$ of a member i in group G
α_i^j	random number of member i in the *j-th* session
p_i^j	pseudonym of member i in the *j-th* session
$h(\cdot)$	hash function
$h^t(x)$	the value hashed t-times for x
$f(\cdot)$	pseudorandom function
S_i^j	stereoscopic image information for 3D real model of member i in the *j-th* session
R_{S_i}	a rendered image of S_i
E	Encryption function
D	Decryption function
M	a message

3.4. Legitimate Group Member (LGM) . Every user (member) registers at SM with the contents-list which is the combination set for gestures and facial expressions of the user. All the gestures and facial expressions are randomly repeated and rearranged in the contents-list. Then, the user keeps the contents-list in their device for later authentication. $S_{t,i}^j$ is put as the stereoscopic image information for the gesture and facial expression of the member i of group t at the *j-th* session. Every session, SM selects one of the combinations of gestures and facial expressions from the contents-list and challenges the member of the group. Then, the member renders his own gesture and facial expression for $S_{t,i}^j$.

4. Construction of SISS Model

In this section, SISS is constructed by using the eight algorithms described before. This SISS model is divided largely

into four processes: system setting, registration, authentication for login, and action. The whole process is shown in Table 2 and the details are addressed in Section 4.1.

4.1. System Setting

4.1.1. $SysPrm(1^k)$ Construction. The basis of security system SISS is established.

(i) *Input*; k: a security parameter.

(ii) *Output*;$\lambda = f(\cdot), h(\cdot), G, g, G_r, n, j, i, E, D$: system parameters' set.

$f : \{0,1\}^k \times \{0,1\}^* \longrightarrow \{0,1\}^k$is a pseudorandom function and $h : \{0,1\}^* \longrightarrow \{0,1\}^k$ is a one-way hash function. G_r is a pseudorandom generator. G is a group of order q which is a large prime. g is a generator of a group G, n is the total members of group G, j is the session number, and i is each member of group G. E and D are encryption and decryption function.

4.2. Registration. The registration process consists of two algorithms: $KeyGen(\lambda)$, $InfGenStr(\lambda, RH, K, P)$.

4.2.1. $KeyGen(\lambda)$ Construction. Key materials are generated.

(i) *Input*; λ.

(ii) *Output*;RH(*group session random numbers set*), K(*group members keys set*), P(*pseudonym keys set*).

(1) Group Session Random Numbers RH: Reversed One-Way Hash Chain. It is assumed that the total number of sessions is s. For every member i, each different random number α_i^s $(1 \le i \le n)$ is generated for the last session. Here, each α_i^s is applied to one-way hash function $(s-1)$ times repeatedly to generate all sessions' random numbers and, respectively, for each user as follows.

$$\begin{gathered}
\alpha_i^s, \text{ (randomly generated)} \\
h(\alpha_i^s) = \alpha_i^{s-1} \\
h(\alpha_i^{s-1}) = \alpha_i^{s-2} = h^2(\alpha_i^s) \\
h(\alpha_i^{s-2}) = \alpha_i^{s-3} = h^3(\alpha_i^s) \\
\cdots \\
h(\alpha_i^4) = \alpha_i^3 = h^{s-3}(\alpha_i^s) \\
h(\alpha_i^3) = \alpha_i^2 = h^{s-2}(\alpha_i^s) \\
h(\alpha_i^2) = \alpha_i^1 = h^{s-1}(\alpha_i^s)
\end{gathered} \tag{1}$$

Therefore, the first session's random number of member i is α_i^1 and the *t-th* session's random number of member i is α_i^t : $h(\alpha_i^{t+1}) = \alpha_i^t = h^{s-t}(\alpha_i^s)$.

With these different random numbers, we can make all different group keys for each member and each session, respectively.

One-way hash function $h(\cdot)$ plays the important role in group information sharing. One-Way hash chain is generated by randomly selecting the last value, which is repeatedly applied to one-way hash function $h(\cdot)$. The initially selected value is the last value of the hash chain. One-way hash chain has two properties as mentioned in Definition 1 in Section 3. Therefore, the two properties make it possible that a leaving member cannot compute new keys after leaving the group and any newly joining member can obtain all previous keys and information through applying the current key to hash function $h(\cdot)$ repeatedly.

(2) Group Keys Set K. It is assumed that there are 'n' members of the group 'G', and the group session keys for each member i of group G are $\{K_{Gi}^j\}$ $(1 \le i \le n, 1 \le j \le s)$. Here, j is a session number and s is the last session. The each member i's group key K_{Gi}^j consists of totally five subkeys; $sK_{i,1}^j, sK_{i,2}^j, sK_{i,3}^j, sK_{i,4}^j, sK_{i,5}^j$. SM selects the master group key mK_g of group $G \in \{G_t\}$, $t \ge 1$ and generates a random number to blind the master key in five subkeys, which is the way to construct a group member's session key and, therefore, the last session group key K_{Gi}^s of user i.

$$\begin{aligned}
sK_{i,1}^s &= h(mK_g)\,\alpha_i^s, \\
sK_{i,2}^s &= h(mK_g)\, f_{mK_g}(mK_g)\,(1-\alpha_i^s), \\
sK_{i,3}^s &= g^{f_{mK_g}(mK_g)}, \\
sK_{i,4}^s &= -\left(h(mK_g) + \alpha_i^s\right), \\
sK_{i,5}^s &= f_{mK_g}(mK_g)\,\alpha_i^s
\end{aligned} \tag{2}$$

The generation principle of group keys is that every different random number for each member and each session ($n \times s$ random numbers) is combined to the master group key mK_g. Table 3 shows the random numbers and group keys for each member and each session which belong to the group G. The group G is one of the groups $\{G_t\}, t \ge 1$.

(3) Group Members' Pseudonym Keys Set P: Reversed One-Way Hash Chain. For stronger security and privacy, SISS uses each member's pseudonyms, which are generated with the reversed one-way hash chain in the same way as group session keys. Thus, each member has also s pseudonyms which are denoted as p_i^j (for each member i, $1 \le j \le s$).

$$\begin{gathered}
p_i^s, \text{ (randomly generated)} \\
h(p_i^s) = p_i^{s-1} \\
h(p_i^{s-1}) = p_i^{s-2} = h^2(p_i^s) \\
h(p_i^{s-2}) = p_i^{s-3} = h^3(p_i^s) \\
\cdots
\end{gathered}$$

$$h\left(p_i^4\right) = p_i^3 = h^{s-3}\left(p_i^s\right)$$
$$h\left(p_i^3\right) = p_i^2 = h^{s-2}\left(p_i^s\right)$$
$$h\left(p_i^2\right) = p_i^1 = h^{s-1}\left(p_i^s\right) \tag{3}$$

4.2.2. $InfGenStr(\lambda)$ Construction. The values to be used for authentication are generated and saved.

(i) *Input*; λ: system parameters' set.

(ii) *Output*; $S_{t,i}^j$, $h(E_{K_{Gi}^1}(p_i^1 \parallel S))$, $\{h(E_{K_{Gi}^j}(p_i^j))\}$, p_i^1, K_{Gi}^1, α_i^q, $\{h(p_i^j), p_i^j\}$ $(1 \le j \le s)$.

At the registration process, every user is given some information from SM and stores them in one's own device such as in smartphone or PC: $S_{t,i}^j$, p_i^1, K_{Gi}^1, $\{h(E_{K_{Gi}^j}(p_i^j))\}$, $h(E_{K_{Gi}^1}(p_i^1 \parallel S))$, $(1 \le i \le n, 1 \le j \le s)$. SM also stores some information for each member i: α_i^s, $\{h(p_i^j), p_i^j\}$, mK_g, $S_{t,i}^j$, $(1 \le j \le s)$.

As such, the output of $InfGenStr(\lambda)$ is the values created by the SM and each user during the registration process, which means the values are stored in advance for later use in the authentication process.

Additional Explanation for Encryption (E) and Decryption (D) with Group Keys. Here, $E_{K_{Gi}^j}(M)$ means ciphertext C with group members' group key '$K_G = \{K_{Gi}^j\}$' for a message M. The encryption with the master key mK_g is assumed $C = E_{mK_g}(M) = g^{h(mK_g)f(mK_g)}M$.

For simplicity, we put $sK_{i,1}^s, sK_{i,2}^s, sK_{i,3}^s, sK_{i,4}^s, sK_{i,5}^s$ as K_1, K_2, K_3, K_4, K_5, and $f_{mK_g}(mK_g)$ as $f(mK_g)$. Then, the encryption with each member's group key $\{K_{Gi}^j\}$, for example, in the last session (i.e., $j = s$, K_{Gi}^s) is as follows:

$$\begin{aligned} C &= E_{K_{Gi}^s}(M) = K_3^{K_1} g^{K_2} M \\ &= \left(g^{f(mK_g)}\right)^{h(mK_g)\alpha_i^s} g^{h(mK_g)f(mK_g)(1-\alpha_i^s)} M \\ &= g^{h(mK_g)f(mK_g)} M. \end{aligned} \tag{4}$$

The decryption method with the group master key 'mK_g' is $D = C \cdot g^{-h(mK_g)f(mK_g)} = M$. Then, the decryption method with each member's group key K_{Gi}^s in the last session is as follows:

$$\begin{aligned} D &= C \cdot K_3^{K_4} g^{K_5} \\ &= g^{h(mK_g)f(mK_g)} M \cdot \left(g^{f(mK_g)}\right)^{-(h(mK_g)+\alpha_i^s)} \cdot g^{f(mK_g)\alpha_i^s} \\ &= g^{h(mK_g)f(mK_g) - f(mK_g)h(mK_g) - f(mK_g)\alpha_i^s + f(mK_g)\alpha_i^s} \cdot M \\ &= M. \end{aligned} \tag{5}$$

We can check whether the result of encryption/decryption with the master group key 'mK_g' is the same as anything of each member's group key $K_{Gi}^j = \{sK_{i,1}^j, sK_{i,2}^j, sK_{i,3}^j, sK_{i,4}^j, sK_{i,5}^j\}$ $(1 \le i \le n, 1 \le j \le s)$. Because of the properties of this developed encryption and decryption algorithms, SISS has no need for rekeying processes whenever membership-changes happen.

4.3. Login by Authentication. The login process consists of four algorithms: $QrU1(RH, K, P)$, $VrSM1_QrSM2(QvU1)$, $VrU2_QrU3(Qv - SM2)$, and $VrSM3(R_{S_i^j})$.

4.3.1. $QrU1(RH, K, P)$ Construction. A member i makes login-request to SM with the stored information.

(i) *Input*; RH, K, P.

(ii) *Output*; $QvU1$ (1st querying value of a user).

(1) *Compute*: $f_{p_i^1}(K_{Gi}^1), h(p_i^1)$; with the stored value p_i^1, K_{Gi}^1, a member i computes $f_{p_i^1}(K_{Gi}^1), h(p_i^1)$.

(2) *A member i queries SM with QvU1*:

$$\begin{aligned} &QvU1 \\ &\quad = 1\,(sn), h\left(p_i^1\right), f_{p_i^1}\left(K_{Gi}^1\right), h\left(E_{K_{Gi}^1}\left(p_i^1 \parallel S_i^1\right)\right). \end{aligned} \tag{6}$$

Here, $h(E_{K_{Gi}^1}(p_i^1 \parallel S_i^1))$ is also the stored value at registration time. Because K_{Gi}^1 is the member i's group key in the first session, $E_{K_{Gi}^1}(p_i^1 \parallel S_i^1)$ means

$$\begin{aligned} C &= E_{K_{Gi}^1}\left(p_i^1 \parallel S_i^1\right) = K_3^{1\,K_1^1} g^{K_2^1}\left(p_i^1 \parallel S_i^1\right) \\ &= \left(g^{f(mK_g)}\right)^{h(mK_g)\alpha_i^1} g^{h(mK_g)f(mK_g)(1-\alpha_i^1)}\left(p_i^1 \parallel S_i^1\right) \\ &= g^{h(mK_g)f(mK_g)}\left(p_i^1 \parallel S_i^1\right). \end{aligned} \tag{7}$$

Here, for simplicity, K_1^1, K_2^1, K_3^1 are denoted as the member i's subkeys for its group key K_{Gi}^1 in the first session. K_4^1, K_5^1 are also the subkeys (for decryption) of K_{Gi}^1.

4.3.2. $VrSM1_QrSM2(QvU1)$ Construction. The SM verifies the login-request of member i and sends the next session information, the group key and the pseudorandom number, to member i.

(i) *Input*; $QvU1$.

(ii) *Output*; $QvSM2$ (2nd querying value of SM).

(1) *Find*: α_i^s, p_i^1.

SM checks $1(sn), h(p_i^1)$ and finds the corresponding values α_i^s, p_i^1 from its storage.

(2) *SM decrypts* with $p_i^1 : D(f_{p_i^1}(K_{Gi}^1)) = K_{Gi}^1$.

(3) *Compute and Verify*: $h^{s-1}(\alpha_i^s) = \alpha_i^{1\prime}, K_{Gi}^{1\prime} = K_{Gi}^1$.

Table 2: The whole process of SISS.

User	*SM*
1. System Setting *1.1. SysPrm*	
2. Registration *2.1. KeyGen* *2.2. InfGenStr*	
3. Log-in by Authentication	
3.1. QrU1 (1) Compute and Query: $f_{p_i^1}(K_{Gi}^1), h(p_i^1)$ $\xrightarrow{\{1(sn), h(p_i^1), f_{p_i^1}(K_{Gi}^1), h(E_{K_{Gi}^1}(p_i^1 \| S_i^1))\}}$	
	3.2. VrSM1_QrSM2 (1) Find: $1(sn), h(p_i^1) \longrightarrow \alpha_i^s, p_i^1$ (2) Decrypt: $D(f_{p_i^1}(K_{Gi}^1)) = K_{Gi}^1$ (3) Compute: $h^{s-1}(\alpha_i^s) = \alpha_i^{1'}$, $K_{Gi}^{1'} = \{K_1^1, K_2^1, K_3^1, K_4^1, K_5^1\}'$ Verify: $K_{Gi}^{1'} = K_{Gi}^1$ (4) Compute & Verify: $h(E_{K_{Gi}^1}(p_i^1 \| S_i^1))' =? h(E_{K_{Gi}^1}(p_i^1 \| S_i^1))$ (5) Compute: α_i^2, K_{Gi}^2 (6) Compute & Query: $\xleftarrow{f_{p_i^1}(K_{Gi}^2, p_i^2), f_{p_i^2}(p_i^1 \| S_i^1)}$
3.3. VrU2_QrU3 (1) Decrypt: $D(f_{p_i^1}(K_{Gi}^2, p_i^2)) = K_{Gi}^{2'}, p_i^{2'}$ (2) Compute & Verify: $h(E_{K_{Gi}^2}(p_i^2))' =? h(E_{K_{Gi}^2}(p_i^2))$, $h(p_i^{2'}) = p_i^1$ Then, $K_{Gi}^{2'} \longrightarrow K_{Gi}^2, p_i^{2'} \longrightarrow p_i^2$ (3) Decrypt: $D(f_{p_i^2}(p_i^1 \| S_i^1)) = p_i^1 \| S_i^1$ (4) Render at a page: $R'_{S_i^1}$	
	3.4. VrSM3 (1) Verify: $R'_{S_i^1} = R_{S_i^1}$
4. Action	
4.1. EncUp $[member - i]$ (1) Encrypt & Upload M: $C_i^1 = E_{K_{Gi}^1}(M)$	$\xrightarrow{=K_{i,3}^1{}^{K_{i,1}^1} \cdot g^{K_{i,2}^1} \cdot M = g^{h(K_G) f(K_G)} M}$
4.2. DnDec $[member - u]$ (1) Download from SISS Server: (2) Decrypt $C_i^1 : D = C_i^1 \cdot K_{u,3}^1{}^{K_{u,4}^1} \cdot g^{K_{u,5}^1} = M$	$\xleftarrow{C_i^1}$

Table 3: Random number and group key.

		Users		
		U_1	U_2	U_3...
session 1	random number	α_1^1	α_2^1	...
	group key	K_{G1}^1;	K_{G2}^1;	...
		$sK_{1,1}^1 = h(mK_g)\alpha_1^1$	$sK_{2,1}^1 = h(mK_g)\alpha_2^1$	...
		$sK_{1,2}^1 = h(mK_g)f_{mK_g}(mK_g)(1-\alpha_1^1)$	$sK_{2,2}^1 = h(mK_g)f_{mK_g}(mK_g)(1-\alpha_2^1)$	...
		$sK_{1,3}^1 = g^{f_{mK_g}(mK_g)}$	$sK_{1,3}^1 = g^{f_{mK_g}(mK_g)}$	...
		$sK_{1,4}^1 = -(h(mK_g)+\alpha_1^1)$	$sK_{2,4}^1 = -(h(mK_g)+\alpha_2^1)$	...
		$sK_{1,5}^1 = f_{mK_g}(mK_g)\alpha_1^1$	$sK_{2,5}^1 = f_{mK_g}(mK_g)\alpha_2^1$	...
session 2	random number	α_1^2	α_2^2	...
	group key	K_{G1}^2;	K_{G2}^2;	...
		$sK_{1,1}^2 = h(mK_g)\alpha_1^2$	$sK_{2,1}^2 = h(mK_g)\alpha_2^2$	...
		$sK_{1,2}^2 = h(mK_g)f_{mK_g}(mK_g)(1-\alpha_1^2)$	$sK_{2,2}^2 = h(mK_g)f_{mK_g}(mK_g)(1-\alpha_2^2)$	...
		$sK_{1,3}^2 = g^{f_{mK_g}(mK_g)}$	$sK_{1,3}^2 = g^{f_{mK_g}(mK_g)}$	...
		$sK_{1,4}^2 = -(h(mK_g)+\alpha_1^2)$	$sK_{2,4}^2 = -(h(mK_g)+\alpha_2^2)$	...
		$sK_{1,5}^2 = f_{mK_g}(mK_g)\alpha_1^2$	$sK_{2,5}^2 = f_{mK_g}(mK_g)\alpha_2^2$	...
session 3...	...	...	...	...

For the found value α_i^s, SM applies α_i^s to hash function repeatedly, up to $(s-1)$ times. If he obtains the result $\alpha_i^{1\prime}$, then SM computes $K_{Gi}^{1\prime}$;

$$\begin{aligned} K_1^{1\prime} &= h\left(mK_g\right)\alpha_i^{1\prime}, \\ K_2^{1\prime} &= h\left(mK_g\right)f_{mK_g}\left(mK_g\right)\left(1-\alpha_i^{1\prime}\right), \\ K_3^{1\prime} &= g^{f_{mK_g}(mK_g)}, \\ K_4^{1\prime} &= -\left(h\left(mK_g\right)+\alpha_i^{1\prime}\right), \\ K_5^{1\prime} &= f_{mK_g}\left(mK_g\right)\alpha_i^{1\prime}. \end{aligned} \tag{8}$$

Then, SM verifies $K_{Gi}^{1\prime} = K_{Gi}^1$ or not.

(4) *Compute and Verify* with K_{Gi}^1: $h(E_{K_{Gi}^1}(p_i^1 \parallel S_i^1))' = h(E_{K_{Gi}^1}(p_i^1 \parallel S_i^1))$.

Here, $E_{K_{Gi}^1}(p_i^1 \parallel S_i^1)$ is the stored value at the registration time and the encryption method is the same as the above 1.

(5) *Compute*: α_i^2, K_{Gi}^2.

SM applies α_i^s to hash function $(s-2)$ times and then computes K_{Gi}^2;

$$\begin{aligned} K_1^2 &= h\left(mK_g\right)\alpha_i^2, \\ K_2^2 &= h\left(mK_g\right)f_{mK_g}\left(mK_g\right)\left(1-\alpha_i^2\right), \\ K_3^2 &= g^{f_{mK_g}(mK_g)}, \\ K_4^2 &= -\left(h\left(mK_g\right)+\alpha_i^2\right), \\ K_5^2 &= f_{mK_g}\left(mK_g\right)\alpha_i^2. \end{aligned} \tag{9}$$

(6) *Compute and Query with QvSM2*:

$QvSM2 = f_{p_i^1}(K_{Gi}^2, p_i^2), f_{p_i^2}(p_i^1 \parallel S_i^1)$, where p_i^2 is also the stored value.

4.3.3. VrU2_QrU3(QvSM2) Construction. A member i verifies the received information from SM, then stores it for the next session, and renders the stereoscopic image on his page.

(i) *Input*; *QvSM2*.

(ii) *Output*; $R_{S_i^j}$ (rendering of S_i^j).

(1) *Decrypt*: i decrypts $D(f_{p_i^1}(K_{Gi}^2, p_i^2)) = K_{Gi}^{2\prime}, p_i^{2\prime}$ with the value p_i^1.

(2) *Compute and Verify*: $h(E_{K_{Gi}^2}(p_i^2))' = h(E_{K_{Gi}^2}(p_i^2))$, $h(p_i^{2\prime}) = p_i^1$.

With the decrypted values $K_{Gi}^{2\prime}, p_i^{2\prime}$, the group member i computes $h(E_{K_{Gi}^2}(p_i^2))'$ and verifies if this is the same as $h(E_{K_{Gi}^2}(p_i^2))$. Then, i hashes the value $p_i^{2\prime}$ and verifies $h(p_i^{2\prime}) = p_i^1$. If the verifications are successful, $K_{Gi}^{2\prime}$ and $p_i^{2\prime}$ become K_{Gi}^2 and p_i^2.

(3) *Decrypt* with p_i^2: $D(f_{p_i^2}(p_i^1 \parallel S_i^1)) = p_i^1 \parallel S_i^1$.

(4) *Render and Upload:* $R'_{S_i^1}$ at a page.

4.3.4. VrSM3($R_{S_i^1}$) Construction. SM verifies what the member i has rendered.

(i) *Input*; $R_{S_i^j}$.

(ii) *Output*; 1 or 0.

(1) *Verify*: $R'_{S_i^1} = R_{S_i^1}$*(3D facial expression and gesture)*.

In this process, a legitimate group member authentication is processed by rendering the decrypted S_i^1 with the member's external device. If SM's verification is successful (return message: 1), the member i can begin to act (login allowed). The action means uploading, downloading, and reading (decryption).

4.4. Action. The Action Process consists of two algorithms: $EncUp(M)$, $DnDec(M)$.

4.4.1. EncUp(M) Construction. A member i encrypts and uploads the sharing information.

(i) *Input*; M.

(ii) *Output*; C_i^1.

(1) *Encrypt and Upload M* by a member i.

$$C_i^1 = E_{K_{Gi}^1}(M) = K_{i,3}^{1\ K_{i,1}^1} \cdot g^{K_{i,2}^1} \cdot M = g^{h(mK_g)f(mK_g)}M \quad (10)$$

4.4.2. DnDec(M) Construction. Another member u downloads and decrypts the sharing information.

(i) *Input*; C_i^1.

(ii) *Output*; M.

(1) *Download*: C_i^1.

Another member u downloads C_i^1 from SISS bulletin (server).

(2) *Decrypt* C_i^1: $D = C_i^1 \cdot K_{u,3}^{1\ K_{u,4}^1} \cdot g^{K_{u,5}^1} = g^{h(mK_g)f(mK_g)}M \cdot (g^{f(mK_g)})^{-(h(mK_g)+\alpha_u^1)} \cdot g^{f(mK_g)\alpha_u^1} = g^{h(mK_g)f(mK_g)-f(mK_g)h(mK_g)-f(mK_g)\alpha_u^1+f(mK_g)\alpha_u^1} \cdot M = M$.

[The Second Session]. From the second session, most processes are similar to the first session. As the session is changed, the corresponding pseudonym keys and group session keys are also changed. As for the stereoscopic image information S for 3D real model, a member sends the information S^t kept from the first session to SM, and then SM challenges the member with the newly selected information S^{t+1} at (6) of the algorithm $VrSM1_QrSM2$. Lastly, the member renders 3D real model $R_{S^{t+1}}$ at his page. Action stage is also similar to the first session.

5. Security Analysis

The security requirements related to group key are as follows:

(1) Group Key Secrecy: It should be computationally impossible that a passive adversary discovers any secret group key.

(2) Forward Secrecy: Any passive adversary with a subset of old group keys cannot discover any subsequent (later) group key.

(3) Backward Secrecy: Any passive adversary with a subset of subsequent group keys cannot discover any preceding (earlier) group key.

(4) Key Independence: Any passive adversary with any subset of group keys cannot discover any other group key [4].

In this paper, the term negligible function refers to a function $\eta : N \longrightarrow R$ such that for any $c \in N$, there exists $n_c \in N$, such that $\eta(n) < 1/n_c$ for all $n \geq n_c$ [5].

The model SISS satisfies group information sharing secrecy as follows: (1) forward secrecy, (2) backward accessibility, (3) group key secrecy, and (4) collusion resistance.

Theorem 1 (forward secrecy). *For any group, an adversary A (including a participant $p \in G_t^j$) cannot know valid group key for (j+l)-th authentication when the adversary A knows group key K_{Gi}^j, where $p \notin G_t^{j+l}$ $(1 \leq j \leq s, 0 < l \leq s - j, 1 \leq t \leq T)$.*

Proof. By Property 3 of Definition 1, if the latest released group key is K_{Gi}^j, no one can know a later value K_{Gi}^l such that $h^{(l-j)}(K_{Gi}^l) = K_{Gi}^j$. Therefore, the probability that a participant $p \in G_t^j$ can generate valid group keys for the next *l-th* session is negligible, where $p \notin G_t^{j+l}$ $(j < l \leq s)$. It means that all leaving group members cannot access any of the next documents of the group anymore. □

Theorem 2 (backward accessibility). *For any group G_t, an adversary A (including a participant $p \in G_t^j$) can generate valid group key for (j-l)-th authentication when the adversary A knows group key K_{Gi}^j, where $p \notin G_t^{j-l}$ $(0 < l < j)$. Namely, all joining members to a group can access all of the previous information of the group.*

Proof. By Property 2 of Definition 1, if the latest released group key is K_{Gi}^j, anyone can deduce earlier values K_{Gi}^l $(0 < l < j)$ by applying the later value K_{Gi}^j to one-way hash key chain like this: $h^{(j-l)}(K_{Gi}^j) = K_{Gi}^l$. Therefore, the probability that a participant $p \in G_t^j$ can generate valid group keys for the earlier *l-th* session is $1 - \eta(n)$, where $p \notin G_t^{j-l}$ $(0 < l < j)$. Namely, all members joining a group can access all of the previous information of the group. □

Theorem 3 (group key secrecy). *For any group G_t, when a revelation of group key K_{Gi}^j happens, the probability that an adversary A (including a participant $p \in G_t^j$) can guess correctly the encrypted information message M of group G_t at the $j - th$ session is negligible.*

Corollary 4. *SISS is semantic secure according to the security game ICR -IS, if DDH is hard and the key material is chosen as described in the algorithm construction.*

The cryptographic elements for authentication and whole protocol are PRF (pseudorandom function, e.g., 128 bit-AES), PRG (pseudorandom generator, e.g., middle-square method, Naor-Reingold pseudorandom function, etc.), and hash function (HAS-160), generally known as secure cryptographic function. Through the cooperative processes of these elements, the final encryption is $C = E_{K_{Gi}^s}(M) = g^{h(mK_g)f(mK_g)}M$. Hence, we have only to show the security under the condition of 'DDH is hard'.

Proof (it is proved by contraposition) . A is assumed as an adversary that wins the security game ICR-IS with advantage ε. We construct an adversary Ω, which uses A as a subroutine and breaks the DDH with nonnegligible advantage.

(i) *Setup*. Algorithm Ω creates m message pages $\in \{M_{i,t}^j\} \in \{M_t\}$ $(1 \le i \le n, 1 \le t \le T, 1 \le j \le m)$ and gives this to the adversary A.

A chooses a polynomial number of subsets $\{M_{i,t}^j\} \in \{M_t\}$ from messages set M. This collection of subsets is called M^*. A sends them to Ω again. Ω invokes algorithm $SysPrm, KeyGen, InfGenStr$. After creating all ciphertext pages $\{CP_{i,t}^j\} \in \{CP_t\}$ for M^*, Ω gives them and their associated subsets to A.

Here, let $g^\delta, g^\tau, g^\gamma$ be a Diffie-Hellman triplet; the challenge is to determine $\gamma = \delta\tau$. Ω guesses a value CP^x for the page CP^y that A will choose in the game ICR-IS, by picking M^x uniformly at random in $\{M_{i,t}^j\}$ $(1 \le t \le T)$. Ω simulates the algorithm *EndUp* on $\{CP_{i,t}^j\}$ as follows. Ω maps every ciphertext page $\{CP_{i,t}^j\}$ to a random value $\{x_{i,t}^j\}$. For B= $\{B_t\}$=$\{CP_{i,t}^j\}$, Ω chooses random number γ_t and outputs the following.

$$\begin{aligned}
&\{B_t\}\left(1 \le i \le n, 1 \le t \le T, 1 \le j \le m\right) = \left\{CP_{i,1}^j\right\} \\
&\quad = \left\{g^{h(k_1)f(k_1)}\left(M_{i,1}^j\right)\right\} = \left\{g^{\gamma_1}\left(x_{i,1}^j\right)\right\} \\
&\quad = \left\{g^{\delta_1\tau_1}\left(x_{i,1}^j\right)\right\} = \left\{CP_{i,2}^j\right\} = \left\{g^{h(k_2)f(k_2)}\left(M_{i,2}^j\right)\right\} \\
&\quad = \left\{g^{\gamma_2}\left(x_{i,2}^j\right)\right\} = \left\{g^{\delta_2\tau_2}\left(x_{i,2}^j\right)\right\} \\
&\cdots \\
&\cdots \\
&\quad = \left\{CP_{i,T}^j\right\} = \left\{g^{h(k_T)f(k_T)}\left(M_{i,T}^j\right)\right\} = \left\{g^{\gamma_t}\left(x_{i,T}^j\right)\right\} \\
&\quad = \left\{g^{\delta_T\tau_T}\left(x_{i,T}^j\right)\right\}
\end{aligned} \tag{11}$$

(ii) *Queries*. If A queries for the message page $\{M_{i,t}^j\}$, Ω outputs the ciphertext page $\{CP_{i,t}^j\}$= $\{(g^{\gamma_t}(x_{i,t}^j)\}$.

(iii) *Challenge*. Finally, A selects a challenge page set $B_0 \in M_t$ at random and generates another page set B_1 from M. Next, A gives B_0, B_1 to Ω. Ω chooses $b \underset{\leftarrow}{R} \{0, 1\}$ and chooses random number γ_b. Ω returns to A the following ciphertext: In the case of $b = 0$, $CP^y = g^{\gamma_0}(x_{i,0}^j)$. If $b = 1$, Ω returns random value in reply to DDH challenge. If $\gamma = \delta\tau$, this is an encryption of $\{CP^x\}$; otherwise it is not. A is again allowed to ask for pages of the Board set with the restriction that A must not make a query to distinguish $\{CP_{i,0}^j\}$ from $\{CP_{i,1}^j\}$ = rand$(CP_{i,0}^j)$ where $\{CP_{i,0}^j\}$ means a DDH triplet and $\{CP_{i,1}^j\}$ is not a DDH triplet.

(iv) *Response*. A outputs a bit b'. If $b' = b$, Ω guesses that $g^\delta, g^\tau, g^\gamma$ constitute a DDH triplet. If $b' \ne b$, Ω guesses that $g^\delta, g^\tau, g^\gamma$ do not constitute a DDH triplet. Since the encryption will be random for the page $\{CP^x\}$ if and only if the challenge is not a DDH tuple, Ω solves the DDH challenge with the same advantage that A has in winning security game ICR-IS.

It is shown that Ω can solve the DDH problem $(\gamma = \delta\tau)$ with nonnegligible probability. Accordingly, the advantage of Ω in winning this experiment is as follows.

$$\begin{aligned}
Adv_\Omega &= Pr\left[Exp_\Omega^{DDH} = 1\right] = Pr\left[b' = b\right] \\
&= Pr\left[b' = b \mid b = 1\right] \cdot Pr\left[b = 1\right] \\
&+ Pr\left[b' = b \mid b = 0\right] \cdot Pr\left[b = 0\right] \\
&= Pr\left[b' = b \mid b = 1\right] \cdot \frac{1}{2} + Pr\left[b' = b \mid b = 0\right] \cdot \frac{1}{2} \\
&= Pr\left[b' = 1 \mid b = 1\right] \cdot \frac{1}{2} + Pr\left[b' = 0 \mid b = 0\right] \cdot \frac{1}{2} \\
&= Pr\left[b' = 1 \mid b = 1\right] \cdot \frac{1}{2} + \Big(1 \\
&- Pr\left[b' = 1 \mid b = 0\right]\Big) \cdot \frac{1}{2} = \frac{1}{2} \\
&+ \frac{1}{2}\Big(Pr\left[Exp_A^{ICR-IS-1} = 1\right] \\
&- Pr\left[Exp_A^{ICR-IS-0} = 1\right]\Big) = \frac{1}{2} + \frac{1}{2}Adv_A^{ICR-IS} = \frac{1}{2} \\
&+ \frac{1}{2}\varepsilon
\end{aligned} \tag{12}$$

□

Theorem 5 (collusion resistance). *For any leaving member $U_i^l \in G^l$ $(1 \le i \le n, 1 \le l < l' \le s)$ including any other adversaries, SISS is ns-collusion-resistant.*

Proof. For anyone U_i^l $(1 \le l < l' \le s)$ colluding with the legitimate member $U_{i'}^j$, the illegal member cannot compute K_{Gi}^j $(j \ge l')$. Although the compromised (illegal) member U_i^l knows $\{\alpha_i^l, K_{Gi}^l\}$ and $\{\alpha_{i'}^j, K_{Gi'}^j\}$, the illegal member cannot receive $\{\alpha_i^{l+1}, K_{Gi}^{l+1}\}$ of the next $(l + 1)$th session. Hence, they cannot compute K_{Gi}^j $(1 \le i \le n, 1 \le l < l' \le j \le s)$. One more important thing is that the illegal member cannot pass the verifiable process to render the real 3D image from the stereoscopic information S_i^j which consists of the member's own gesture and facial expression. Therefore, the illegal member cannot pass the authentication process of login. □

6. Performance Analysis

The main purpose of this paper is to design a prototype scheme for secure patient networks. In addition, we try to apply a new technology like the AR/VR technique of 3D model to the authentication process. However, the performance for whole protocol of SISS largely depends on the network condition. Hence, we experiment the performance of SISS with separate eight parts as follows: (1) the generation

Table 4: The specification of a server.

		Server
Hardware	CPU	Intel Core i7-4770K 3.5 GHz
	Memory	4 GB
	Disk	200 GB
OS	Ubuntu 14.04 LTS / Linux Kernel 3.19.0	
Development Tool	gcc 4.8	

Table 5: The specification of a client.

		Client (Nexus 5X)
Hardware	CPU	Snapdragon 808 1.8 GHz
	Memory	2 GB
	Disk	16 GB
OS	Android 6.0	
Development Tool	eclipse / Android SDK	

Table 6: Cryptographic parameters and libraries.

Pseudorandom function f	AES 128 bits
Hash function h	SHA-1
Group parameter	cyclic group
Modulus P	2048 bits
Order q	256 bits
Generator g	2048 bits
Time measurement function in a Server	clock()
Time measurement function in a Client	System.currentTimeMillis()
Crypto Library of a Server	MIRACLE
Crypto Library of a Client	Android OpenSSL & Java BigInteger

time in a server including storage time, (2) the time for a client including data transfer and storage in DB, (3) $QrU1$ of login, (4) $VrSM1_QrSM2$ of login, (5) $VrU2_QrU3$ of login, (6) $VrSM3$ of login, (7) *EncUp*, and (8) *DnDec*

6.1. Implementation and Experimental Environment. The experimental environments of a server and a client are addressed in Tables 4 and 5.

6.2. Cryptographic Parameters and Library. Cryptographic parameters and libraries are described in Table 6.

6.3. The Results of Implementation. Table 7 shows the performance of SISS divided into eight parts. The registration process which needs only once to join the website totally takes over 20 seconds. Considering that users should generate all their information to use it through all their sessions at this registration process in advance, the estimated time is understandable and applicable to a real world in general. Other processes such as login or actions take much less than 1 second (cf. in the implementation of VrU_QrU3, the rendering process can be skipped because there is no commercialized tools for rendering until now; we replace $R_{S_i^j}$ with 20-byte 3-dimensional image).

6.4. Comparison with Other Works. The related works' main goal was only focused on the methods of group key's rekeying, revocation, and redistribution, whereas SISS's main goal is to design the information sharing protocol with safety for application website. Hence, the Storage Overhead and Communication Overhead analyzed in the related works are obviously different from SISS because our proposed group key system is developed to eliminate the processes of rekeying and redistribution that constitute hard and complicated work with heavy overheads. To the best of my knowledge, SISS's group key is the first scheme without rekeying and redistribution; nevertheless it can guarantee the security requirements of group key.

Gou et al.'s paper [50] is the latest work to analyze the performances of current schemes until now; the minimal Storage Overhead is $\log_2 p$ (p: finite field's order) and Communication Overhead is $(n+2)j\log_2 q$ (n: maximum revoked users, j: session, q: multiplicative group's order). As for SISS, Storage and Communication Overheads are O(0). Thus, based on Gou et al.'s work, we only compare and analyze the security performances of group key because the proposed scheme SISS has no process itself for group key's rekeying and redistribution.

Table 8 shows that Staddon et al. and Liu et al.'s schemes can guarantee only forward secrecy, and Rams et al. and Guo et al.'s new scheme can meet all properties of forward secrecy, backward secrecy, and collusion resistance. The number of Revocation Limit is the maximum for Guo et al.'s new scheme and the proposed scheme SISS. And, SISS can guarantee all security properties of forward secrecy, backward accessibility, and collusion resistance, but not backward secrecy.

7. Discussion

7.1. The Differences from Other Group Key Structure

(1) The application and aim of our group key are different from the traditional group key. Rather, they are closer to 'keyword search schemes for multiuser setting with group keys', whose security requirements are forward secrecy and backward accessibility; the leaving members should not know the group's documents, and newly joining members should know the previous documents of the group to perform the group's tasks. In the sense of sharing information among group members, we used the term group key.

(2) The formation structure of the group key is completely different. General group key systems make every user share the same group key for the session. However, in SISS, on the basis of the master key, random numbers and other things are combined, where the random number has a different initial value for each user, which is hashed $(s-1)$ times for s sessions (total number of sessions is s). Finally, each user

Table 7: The result.

Operation		Process time (ms)
Registration (1000 Session)	The generation time in a server including storage time	6138 ms (6.1 s)
	The time for a client including data transfer and storage in DB	14499 ms (14.4 s)
Login	$QrU1$	16 ms (0.016 s)
	$VrSM1_QrSM2$	2 ms (0.002 s)
	$VrU2_QrU3$	56 ms (0.056 s)
	$DVrSM3$	skip
		(comparison between simple values)
EncUp		54 ms (0.054 s)
DnDec		32 ms (0.032 s)

Table 8: The comparison of group key security performance.

Scheme	Revocation Limit	Forward Secrecy	Backward Secrecy	Collusion Resistance
Staddon et al.	n	Yes	No	No
Liu et al.	n	Yes	No	No
Rams et al.	n	Yes	Yes	Yes
Guo et al.'s basic scheme	n	Yes	No	No
Guo et al.'s new scheme	ns	Yes	Yes	Yes
SISS	ns	Yes	Backward Accessibility	Yes

n: maximum revoked users, s: maximum session.

has different group keys for each session and does not share any key with any one, except for K_3 only. The important thing is that even members themselves do not know their group's master key because it is masked with random numbers. At registration, s hashed values generated through hash chain are stored in only SM's server. Each user has only $S_{t,i}^{j}$, p_i^1, K_{Gi}^{1}, $\{h(E_{K_{Gi}^{j}}(p_i^j))\}$, $h(E_{K_{Gi}^{1}}(p_i^1 \parallel S))$, $(1 \le i \le n, 1 \le j \le q)$ for the authentic information required at the start of the session as mentioned in the algorithm $InfGenStr$.

(3) The result of encryption/decryption with the group's master key and the result of encryption/decryption with the every member's group key are the same (refer to [Additional Explanation for Encryption (E) and Decryption (D) with Group Keys] in Section 4). This is because the developed equation (algorithm) is designed according to the principle that all random numbers attached before the computation should be removed after the computation. It makes rekeying and redistributing of the group key unnecessary for SISS. In SISS, members can upload only on their web pages, while download can be done on their own web pages and those of other users (valid users). Therefore, members encrypt the information that they want to share with the group key and encrypt the information that they want to be secret with their private keys.

(4) It can be said that the group key renewals for session changes are accomplished in the authentication processes of login for each member. In other words, the group key and pseudonym key for each user's next session are given by the SM at the end of the login process, which serve more as authenticators to pass the login process. If any member does not receive the group key and pseudonym key for the next session from the SM, the value can never be deduced. The reasons are as follows: (1) Group keys have an effect similar to a one-time password because they have completely different values for each member and for each session. (2) The master key and the random number cannot be inferred because of the combined characteristics (safety) between master key and random number such as DDH, DLP, and other cryptographic functions. (3) Due to the hash chain's one-wayness, which is the method of random number generation, we never know the random number of the next session, so we do not know the group key value of the next session

(5) The leave and revocation process of SISS is also different from the general group key because SISS does not have a rekeying and redistributing process. When SM receives the leave request from a member, the SM enters the revocation process, records the member's id in the leave-list, and deletes the user's hash chain and other additional information. Even if a member who has left a session tries to log in with the next session information which is received from the previous session, the member cannot pass the authentication because all information of the user has been removed from SM's server. And the member cannot receive the next session information any longer. In other words, a member can no longer log in to the group if the member leaves the group, so that the member should download all the previous information before requesting leave. The leaving members can never know the next subsequent information, while newly joining members can decrypt all the shareable information encrypted with the group key.

(6) The meaning of a session of SISS is different from other general group key systems that consider the session as the number of membership-changes, as SISS considers

the session as the number of logins for each member. If a member has performed a total of s logins, then the member can reconnect to SM and generate a new hash chain again as he did at the registration time. The total number of sessions, s, can be determined by the policies of the website or by the needs of individual members.

7.2. Legitimate Group Member. In the last step of the login authentication, 3-dimensional image R_S is rendered. R_S plays a role of a LGM (legitimate group member) which is decided with SM at the registration time. The goal is "improving authentication and security against conspiracy and compromise". If 3-dimensional image is inefficient in a real world, 2-dimensional image is recommendable.

In 2016, Google's project 'Tango' has been showcased with indoor mapping and VR/AR platform [52]. 'Tango' technology enables a mobile device to measure the physical world. Tango-enabled devices (smartphones, tablets) are used to capture the dimensions of physical space to create 3D representations of the real world. 'Tango' gives the Android device platform the new ability of spatial perception.

According to JPMorgan analyst Rod Hall [7], Apple expects that iPhone 8 would feature a front-facing 3D laser scanner for facial recognition. It can be also said that the facial recognition will potentially be more secure than Touch ID, and 3D laser scanner could eventually be used for other purposes such as augmented reality. Unfortunately, however, the iPhone 8 released in 2017 did not have the expected function of front-facing 3D laser scanner. Even though the released AR technique of iPhone 8 was different from the 3D laser scanner for facial recognition by Rod Hall [7], we can anticipate the generalized AR technique for the facial recognition in the near future. Therefore, we can say that the proposition of SISS is timely good to apply LGM to the real world keeping abreast of Tango and iPhone's AR/VR technique of mobile devices.

7.3. Privacy Preserving SISS. SISS can meet the privacy requirements as follows:

(1) Anonymity and Pseudonymity: In SISS, each member uses different pseudonymity for each session. Although perfect anonymity cannot be provided, pseudonymity can be provided instead.

(2) Unlinkability: Every session, users log in with different pseudonyms (P) and use different encryption keys (each member's group key). Consequently, SISS can achieve unlinkability and similar level of security to 'One-Time Encryption'.

(3) Unobservability: All information is encrypted by members' group keys, which have different values by being masked with the differently generated random numbers for each user and each session [53].

7.4. Mutual Authentication. An attacker may try to pretend to be a valid member to log in to the SIS system or masquerade as an SM server to extract users' information. This property is about spoofing attack.

The authentication between a member and the SM server is accomplished through the query and verification algorithms: Qr, Vr_Qr, Vr. Specifically, authentication processes for login consist of $QrU1(RH, K, P)$, $VrSM1_QrSM2(QvU1)$, $VrU2_QrU3(QvSM2)$, and $VrSM3(R_{S_i^j})$. In $QrU1(RH, K, P)$, a member queries the SM with $QvU1$ which is the computed values using the stored values at the registration time. Then, in $VrSM1_QrSM2(QvU1)$, the SM server verifies the value $QvU1$ with the stored values, too. After the successful verification, the SM server queries the member with $QvSM2$, which is also computed using the stored values and $QvU1$. To the last processes $VrSM3(R_{S_i^j})$, the member and the SM server authenticate each other using the stored values, respectively.

From a member to the SM server, if the SM server can obtain the corresponding rightly rendered image in the last authentication process, it means that the SM server is the real server to which a member wants to log in and the member is a valid user to be registered in advance.

8. Conclusion

SISS is the proposal for the patients from all over the world who want to get some help and share information through websites such as 'PatientsLikeMe' or 'CureTogether'. The proposed model SISS can guarantee security and privacy for the sensitive health and private information. As for the main method of group key management system, SISS addressed the hard problems of rekeying and redistribution, conspiracy, and backward accessibility with new ideas such as equations for encryption/decryption and LGM. Moreover, SISS is scalable to general group's project applications with safety. Therefore, it is clear that the problem of information sharing and the approaches between collaborative computing and security should be managed as Integrated Security Management (ISM).

Data Availability

No data were used to support this study. The main method is an encryption algorithm, so that the information in this paper is randomly generated.

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by Korea Government (Ministry of Education, NRF-2017R1D1A1B03029488). The author appreciates Ph.D. candidate Park, Jin Hyung for implementing performance.

References

[1] https://www.patientslikeme.com.
[2] https://curetogether.com.

[3] J. Eom, D. H. Lee, and K. Lee, "Patient-controlled attribute-based encryption for secure electronic health records system," *Journal of Medical Systems*, vol. 40, article no 253, 2016.

[4] Y. Kim, A. Perrig, and G. Tsudik, "Communication-efficient group key agreement," in *Proceedings of the International Information Security Conference*, vol. 65 of *IFIP Advances in Information and Communication Technology*, pp. 229–244, Springer, Paris, France, 2001.

[5] H.-A. Park, J. H. Park, and D. H. Lee, "PKIS: practical keyword index search on cloud datacenter," *EURASIP Journal on Wireless Communications and Networking*, vol. 84, no. 8, pp. 1364–1372, 2011.

[6] Y. Hu, A. Perrig, and D. B. Johnson, "Efficient security mechanisms for routing protocols," in *Proceedings of the NDSS'03*, pp. 57–73, February 2003.

[7] L. Rehm, 2017, https://www.dpreview.com/news/3278771312/iphone-8-expected-to-replace-touch-id-with-3d-facial-recognition.

[8] H. Park, J. W. Byun, and D. H. Lee, "Secure index search for groups," in *Proceedings of the TrustBus'05*, vol. 3592 of *Lecture Notes in Computer Science*, pp. 128–140, Springer, Berlin, Germany, 2005.

[9] P. Wang, H. Wang, and J. Pieprzyk, "Common secure index for conjunctive keyword-based retrieval over encrypted data," in *Proceedings of the SDM'07*, vol. 4721 of *Lecture Notes in Computer Science*, pp. 108–123, 2007.

[10] P. Wang, H. Wang, and J. Pieprzyk, "Keyword field-free conjunctive keyword searches on encrypted data and extension for dynamic groups," in *Proceedings of the CANS'08*, vol. 5339 of *Lecture Notes in Computer Science*, pp. 178–195, 2008.

[11] J. Li and X. Chen, "Efficient multi-user keyword search over encrypted data in cloud computing," *Computing and Informatics*, vol. 32, no. 4, pp. 723–738, 2013.

[12] D. Wallner, E. Harder, and R. Agee, "Key management for multicast: issues and architecture," IETF RFC 2627, 1999.

[13] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16–30, 2000.

[14] M. Zheng, J. Zhu, and G. Cui, "A hybrid group key management scheme for two-layered Ad Hoc networks," in *Proceedings of the 9th International Conference on Information Technology (ICIT'06)*, pp. 83-84, Bhubaneswar, India, 2000.

[15] A. Perrig, D. Song, and J. D. Tygar, "ELK, a new protocol for efficient large-group key distribution," in *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, pp. 247–262, USA, May 2001.

[16] X. S. Li, Y. R. Yang, M. G. Gouda, and S. S. Lam, "Batch rekeying for secure group communications," in *Proceedings of the 10th International Conference on World Wide Web, (WWW '01)*, pp. 525–534, Hong Kong, May 2001.

[17] Y. Chen, J. D. Tygar, and W. Tzeng, "Secure group key management using uni-directional proxy re-encryption schemes," in *Proceedings of the Conference on Computer Communications (INFOCOM '11)*, pp. 1952–1960, Shanghai, China, April 2011.

[18] J. A. Naranjo, N. Antequera, L. G. Casado, and J. A. Lopez-Ramos, "A suite of algorithms for key distribution and authentication in centralized secure multicast environments," *Journal of Computational and Applied Mathematics*, vol. 236, no. 12, pp. 3042–3051, 2012.

[19] P. Vijayakumar, S. Bose, and A. Kannan, "Centralized key distribution protocol using the greatest common divisor method," *Computers & Mathematics with Applications*, vol. 65, no. 9, pp. 1360–1368, 2013.

[20] L. Harn and C. Lin, "Authenticated group key transfer protocol based on secret sharing," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 59, no. 6, pp. 842–846, 2010.

[21] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, article no 197, pp. 644–654, 1976.

[22] M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system," in *Proceedings of the Advances in Cryptology, EUROCRYPT 1994*, vol. 950 of *Lecture Notes in Computer Science*, pp. 275–286, Springer, Berlin, Germany, 1994.

[23] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 8, pp. 769–780, 2000.

[24] A. Joux, "A one round protocol for tripartite Diffie-Hellman," *Journal of Cryptology*, vol. 17, no. 4, pp. 263–276, 2004.

[25] J. Katz and M. Yung, "Scalable protocols for authenticated group key exchange," in *Proceedings of the Advances in Cryptology, (CRYPTO '03)*, vol. 2729 of *Lecture notes in computer science*, pp. 110–125, Springer, Berlin, Germany, 2003.

[26] Q. Wu, Y. Mu, W. Susilo, B. Qin, and J. Domingo-Ferrer, "Asymmetric group key agreement," in *Proceedings of the EUROCRYPT'09*, vol. 5479 of *Lecture Notes in Computer Science*, pp. 153–170, Springer, Berlin, Germany, 2009.

[27] L. Zhang, Q. Wu, B. Qin, and J. Domingo-Ferrer, "Provably secure one-round identity-based authenticated asymmetric group key agreement protocol," *Information Sciences*, vol. 181, no. 19, pp. 4318–4329, 2011.

[28] X. Lv, H. Li, and B. Wang, "Group key agreement for secure group communication in dynamic peer systems," *Journal of Parallel and Distributed Computing*, vol. 72, no. 10, pp. 1195–1200, 2012.

[29] S. Mittra, "Iolus: a framework for scalable secure multicasting," in *Proceedings of the ACM SIGCOMM*, vol. 27, pp. 277–288, New York, NY, USA, 1997.

[30] L. R. Dondeti, S. Mukherjee, and A. Samal, "Scalable secure one-to-many group communication using dual encryption," *Computer Communications*, vol. 23, no. 17, pp. 1681–1701, 2000.

[31] A. T. Sherman and D. A. McGrew, "Key establishment in large dynamic groups using one-way function trees," *IEEE Transactions on Software Engineering*, vol. 29, no. 5, pp. 444–458, 2003.

[32] D.-W. Kwak and J. W. Kim, "A decentralized group key management scheme for the decentralized P2P environment," *IEEE Communications Letters*, vol. 11, no. 6, pp. 555–557, 2007.

[33] C. Blundo, P. D'Arco, A. De Santis, and M. Listo, "Design of self-healing key distribution schemes," *Designs, Codes and Cryptography. An International Journal*, vol. 32, no. 1-3, pp. 15–44, 2004.

[34] C. Blundo, P. D'Arco, and A. De Santis, "Definitions and bounds for self-healing key distribution schemes," in *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP '04)*, vol. 3142 of *Lecture Notes in Computer Science*, pp. 234–245, Turku, Finland, 2004.

[35] C. Blundo, P. D'Arco, and A. De Santis, "On self-healing key distribution schemes," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 52, no. 12, pp. 5455–5467, 2006.

[36] R. Dutta, "Anti-collusive self-healing key distributions for wireless networks," *International Journal of Wireless and Mobile Computing*, vol. 7, no. 4, pp. 362–377, 2014.

[37] H. Chen and L. Xie, "Improved one-way hash chain and revocation polynomial-based self-healing group key distribution schemes in resource-constrained wireless networks," *Sensors*, vol. 14, no. 12, pp. 24358–24380, 2014.

[38] R. Dutta, S. Mukhopadhyay, and M. Collier, "Computationally secure self-healing key distribution with revocation in wireless ad hoc networks," *Ad Hoc Networks*, vol. 8, no. 6, pp. 597–613, 2010.

[39] R. Dutta, S. Mukhopadhyay, A. Das, and S. Emmanuel, "Generalized self-healing key distribution using vector space access structure," in *Proceedings of the 7th International IFIP-TC6 Networking Conference on Ad Hoc and Sensor Networks*, vol. 4982 of *Lecture Notes in Computer Science*, pp. 612–623, Springer, Berlin, Germany, 2008.

[40] G. Saez, "On threshold self-healing key distribution schemes," in *Proceedings of the Cryptography and Coding, LNCS3796*, vol. 3796 of *Lecture Notes in Computer Science*, pp. 340–354, Springer, Berlin, Germany, 2005.

[41] B. Tian, S. Han, T. S. Dillon, and S. Das, "A self-healing key distribution scheme based on vector space secret sharing and one way hash chains," in *Proceedings of the 2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM'08)*, pp. 1–6, Newport Beach, Calif, USA, June 2008.

[42] X. Du, Y. Wang, J. Ge, and Y. Wang, "An ID-based broadcast encryption scheme for key distribution," *IEEE Transactions on Broadcasting*, vol. 51, no. 2, pp. 264–266, 2005.

[43] S. Han, B. Tian, Y. Zhang, and J. Hu, "An effieient self-healing key distribution scheme with constant-size personal keys for wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, pp. 1–5, Cape Town, South Africa, May 2010.

[44] T. Rams and P. Pacyna, "A survey of group key distribution schemes with self-healing property," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 820–842, 2013.

[45] J. Staddon, S. Miner, M. Franklin, D. Balfanz, M. Malkin, and D. Dean, "Self-healing key distribution with revocation," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 241–257, Berkeley, UK, May 2002.

[46] T. Rams and P. Pacyna, "Self-healing group key distribution with extended revocation capability," in *Proceedings of the International Conference on Computing, Networking and Communications, (ICNC '13)*, pp. 347–353, San Diego, Calif, USA, January 2013.

[47] T. Rams and P. Pacyna, "Long-lived self-healing group key distribution scheme with backward secrecy," in *Proceedings of the 1st International Conference on Networked Systems, (NetSys '13)*, pp. 59–65, Stuttgart, Germany, March 2013.

[48] D. Liu, P. Ning, and K. Sun, "Efficient self-healing group key distribution with revocation capability," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, pp. 27–31, Washington, D.C., USA, October 2003.

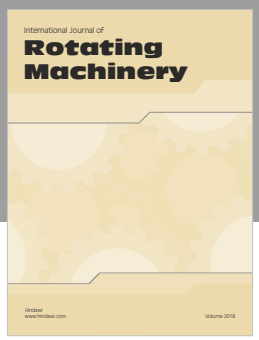
[49] D. Hong and J.-S. Kang, "An efficient key distribution scheme with self-healing property," *IEEE Communications Letters*, vol. 9, no. 8, pp. 759–761, 2005.

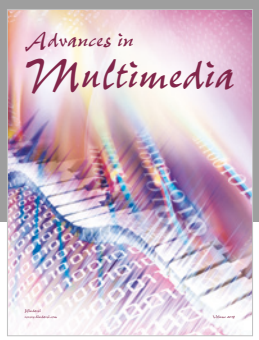
[50] H. Guo, Y. Zheng, X. Zhang, and Z. Li, "Exponential arithmetic based self-healing group key distribution scheme with backward secrecy under the resource-constrained wireless networks," *Sensors*, vol. 16, no. 5, article no 609, 2016.

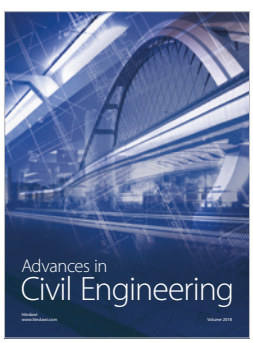
[51] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P '00)*, pp. 44–55, IEEE, Berkeley, Calif, USA, May 2000.

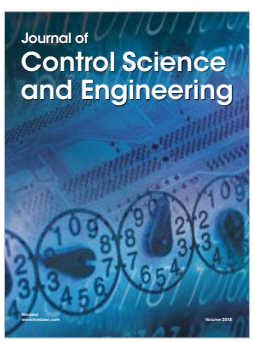
[52] G. Sterling, "Google to showcase project tango indoor mapping and VR/AR platform at Google I/O. The ambitious initiative brings the digital and physical even closer together," 2016, http://searchengineland.com/google-showcase-project-tango-indoor-mapping-vrar-platform-google-io-249629.

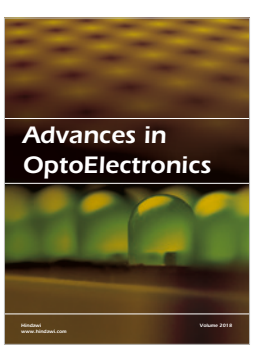
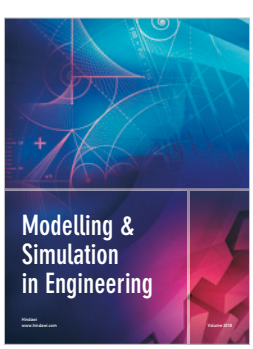
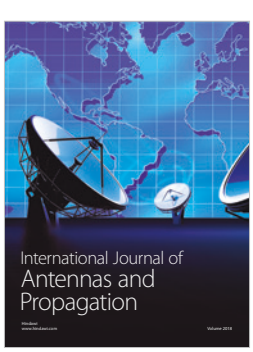
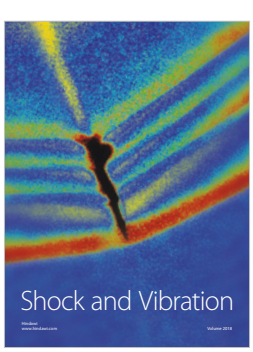
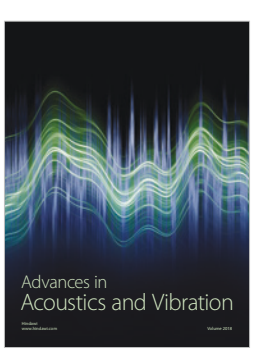
[53] H.-A. Park, J. Zhan, and D. H. Lee, "PPSQL: privacy preserving SQL queries," in *Proceedings of the 2nd International Conference on Information Security and Assurance, (ISA'08)*, pp. 549–554, Republic of Korea, April 2008.









Hindawi

Submit your manuscripts at
www.hindawi.com