

**NOVEL AUTHENTICATION & AUTHORIZATION MANAGEMENT FOR  
SENSITIVE INFORMATION PRIVACY PROTECTION USING DYNAMIC KEY  
BASED GROUP KEY MANAGEMENT**

XIANPING WU<sup>i</sup>

*Caulfield School of Information Technology, Monash University  
Melbourne, Victoria, 3145, Australia  
xpwu1@student.monash.edu.au*

HUY HOANG NGO

*Caulfield School of Information Technology, Monash University  
Melbourne, Victoria, 3145, Australia  
hhngo1@student.monash.edu.au*

PHU DUNG LE

*Caulfield School of Information Technology, Monash University  
Melbourne, Victoria, 3145, Australia  
phu.dung.le@infotech.monash.edu.au*

BALASUBRAMANIAM SRINIVASAN<sup>ii</sup>

*Clayton School of Information Technology, Monash University  
Melbourne, Victoria, 3145, Australia  
bala.srinivasan@infotech.monash.edu.au*

This paper presents a secure authentication and authorization management mechanism for protecting privacy in sensitive information systems. It allows involved individuals and group participants to achieve high security levels and tight authorization control. The need of sharing long term secrets to authenticate individuals and group users is eradicated in the proposed protocol by dynamic keys. It overcomes the secrets compromising during authentication via open networks. Furthermore, it also offers an ability allowing information owners to have fine-gained control of their critical information. Finally, the paper gives a formal analysis to demonstrate how secure the proposed work together with discussions of security issues. It is argued that the proposed work achieves strong authentication and authorization, and solves the involved participants' plausible deniability issues.

*Keywords:* Privacy protection; sensitive information; dynamic key; group key; authentication; authorization.

## **1. Introduction**

Along with advancements of network technology, the use of internet has become an important part of our daily life. It has been evidenced for many services such as information sharing and internet banking. Meanwhile, the huge proliferation of

<sup>i</sup> PO Box 197 Caulfield East, Melbourne, Victoria, 3145, Australia

<sup>ii</sup> Monash University, Clayton campus, Wellington Road, Clayton, Victoria, 3800, Australia

electronically accessible information has led to a great deal of research and development in information system to help people search and quickly fetch or share relevant and meaningful information. Utilizing these emerging technologies, however, also experience with security issues. Among these security issues, confidentiality for sensitive information of users is the most concerns. While sharing information with others, users want to protect their privacy. Also they yearn to gain fine control of their sensitive information in order to delegate authority for individual or group users. On the other hand, for information service providers, protecting sensitive information is a growing concern. The importance of securing critical business data and customer information reaches to the corporate boardroom, because failure to protect these assets may result in losing customer and investor confidence.

Meanwhile, confidentiality is the most crucial requirement in security for sensitive information systems. Confidentiality refers to authentication and authorization. They guarantee that sensitive information is only accessed by intended authorized users. Currently, there are a number of approaches to authenticate and authorize users in sensitive information systems. The most common approaches are combining authentication methods such as Kerberos [Neuman and Steiner,(1988)] and EAP [Edney and Arbaugh,(2003)] and common authorization methods such as access control list [Loscocco, Smalley et al.,(1998; Loscocco and Smalley,(2001)] and Role Based Access Control [Osborn, Sandhu et al.,(2000; Sandhu, Ferraiolo et al.,(2000)].

Nevertheless, these approaches aren't flexible enough to allow users to negotiate for access control on the resource. They also do not stress on privacy of information owner, especially in health information system and military information system. The analysis in [Mao and Boyd,(1994)] points out that authentication using long term key encryption like Kerberos and EAP can lead to the vulnerability of the systems under cryptanalysis. Furthermore, these authentication schemes do not provide authentication and authorization verification for group members. Although Group Key Management [Mitra,(1997; Amir, Nita-Rotaru et al.,(2001; Bresson, Chevassut et al.,(2004; Kim, Perrig et al.,(2004; Ngo, Wu et al.,(2008)] is a solution to provide secure authentication for group members, this approach does not consider the ability to delegate access control for/from participants in sensitive information systems.

Besides, these approaches and solutions for protecting sensitive information have a common limitation of employing long term shared keys or public keys to secure the authentication. Among symmetric key encryption algorithms, only the one-time pad can be proven [Shannon,(1949)] to be secure against any adversary, no matter how much computing power is available. Also, there is no asymmetric scheme with this property, since all asymmetric schemes are susceptible to brute force key search attack [Kahn,(1967)]. Therefore, once the keys are exposed, the protection for sensitive information systems will be compromised.

In this paper, we propose a new secure authentication and authorization protocol for protecting privacy in sensitive information systems. It allows users authenticate themselves to have fine-gained control to the portions of their records. It stresses on privacy protection and offers secure authentication and flexible authorization for individuals and group members.

The rest of paper is organized as follows. Section 2 briefly discusses the background which relates to our proposed works. The proposed secure authentication & authorization management for sensitive information systems is presented in session 3. Section 4

analyses the security of the management. Section 5 concludes and explores possible future research directions.

## **2. Background**

In this section, we describe two techniques, group key management and dynamic key management used in our proposed works. They are two fundamental components for our authentication and authorization management for sensitive information systems.

### **2.1. Group key management**

Along with the popularity of group-oriented communication systems, sensitive information sharing has brought huge convenient for users. However, sensitive information confidentiality is rising as an important issue for group members. To achieve confidentiality in group communication, according to [Kim, Perrig et al.,(2004)], the group key management for sensitive information systems is required group key secrecy, backward secrecy and forward secrecy. Besides that, it also requires the flexible and efficient rekeying operation. Meanwhile, privacy for users in sensitive information systems is a big challenge for group key management.

In cryptography, a group key is a cryptographic key that is shared between a group of users. Thus, the management of the group key plays a critical role in secure multicast. Group key management is a component that generates, distributes and updates cryptographic keying material for group members to ensure the multicast security, such as access control, data confidentiality and group authentication.

There have been many proposed works on designing group security solutions in order to protect sensitive information multicasting. These include secure multicast routing [Shields and Garcia-Luna-Aceves,(1999)], reliable group rekey message delivery [Wong and Lam,(2000)], and cryptographic schemes [Neuman and Steiner,(1988)]. For an overview of the issues in multicast security, it has been discussed in [Moyer, Rao et al.,(1999)]. In [Challal and Seba,(2005)], Challal and Seba imply that the major problems of group key management are confidentiality, authentication, access control and watermarking. Meanwhile, forward secrecy and backward secrecy are the most important for group communication confidentiality.

- (1) Group key secrecy guarantees that it is computationally infeasible for a passive adversary to discover any group key.
- (2) Forward secrecy guarantees that a passive adversary who knows a contiguous subset of previous group keys cannot discover subsequent group keys.
- (3) Backward secrecy guarantees that a passive adversary who knows a contiguous subset of group keys cannot discover preceding group keys.

To meet the security requirements, group key management uses rekeying operation when a member leaves or joins to a group.

- (1) Member Join, when a user wishes to join into a group, rekeying operation is performed to update group keys in order to achieve backward secrecy.
- (2) Member Leave: when a user leaves or is evicted from a group, rekeying operation is triggered to preserve forward secrecy.

## 2.2. Dynamic key management

When a user joins a group, for backward secrecy, a new group key is generated, encrypted by a shared unique key and sent to the user. In order to prevent the group key from risks associated with the compromise of long term unique shared cryptographic keys, dynamic keys are used to overcome the threats.

A dynamic key is a single-use symmetric key used for generating tokens and encrypting messages in one communication flow. Each key is a nonce. The use of dynamic keys introduces complication in cryptographic systems. However, it also helps with some problems. There are three primary reasons for dynamic keys used in our proposed work.

Firstly, the use of long term share keys makes sensitive information systems vulnerable for adversaries, however, by using dynamic keys, it makes attacks more difficult. Secondly, most sound encryption algorithms require cryptographic keys to be distributed securely before enciphering takes place. However, key distribution is one of the drawbacks of symmetric key algorithms. Although asymmetric key algorithms do not require key distribution, but they are slow and susceptible to brute force key search attack. Therefore, the use of asymmetric key algorithms to distribute an encrypted secret for another is only once. Then dynamic keys are generated based on the secret and other key materials. It can improve the overall security considerably. Last but not least, security token can be generated by either long term symmetric keys or nonce dynamic keys. Even though both methods generate variational tokens every time, dynamic key method is more difficult to break than long term key method.

In accordance with the primary reasons for using dynamic keys in our proposed work, it is necessary to have an unambiguous definition. In addition, the idea of dynamic keys is derived from TAN [Oppliger,(1996)]. Therefore, the notion of one way function [Menezes,(1996)] is used for reference. It is defined that “one way function is a function  $f$  such that for each  $x$  in the domain of  $f$ , it is easy to compute  $f(x)$ ; but for essentially all  $y$  in the range of  $f$ , it is computational infeasible to find any  $x$  such that  $y = f(x)$ .” Formally, a function  $f : \{0,1\}^* \rightarrow \{0,1\}^*$  is one way, if and only if,  $f$  is polynomial time computable, and for any probabilistic polynomial time algorithm  $A$ , the probability that  $A$  successfully inverts  $f(x)$ , for random  $x \in_R \{0,1\}^{|x|}$  is negligible [Talbot and Welsh,(2006)]. Therefore, dynamic keys can be defined,

**Definition 2.1** *Dynamic keys  $DK = \{dk_i \mid i \in N\}$  are synchronously offline generated by a special one way function  $f(\cdot)$  in two entities  $P$  and  $Q$  based on a form of pre-shared secret  $s$ .*

$$DK = \{f^i(\text{forms of } s) \mid i \in N\} \quad (1)$$

where,

(1) *Special one way function means that it is computational infeasible to find any distinct input that has the same output as any specified input. Concisely,*

$$\forall x, y(x \neq y), \neg \exists f(x) = f(y) \quad (2)$$

The special one way function, precisely, dynamic key generation scheme [Rubin and Wright,(2002; Li and Zhang,(2004; Kungpisdan, Le et al.,(2005; Dandash, Wang et al.,(2007)] has been proposed and the mechanisms of generating dynamic keys have already in place. Therefore, in this paper, the use of dynamic keys in group key management to improve security of sensitive information systems is discussed emphatically. The cryptography using those dynamic keys can secure communication as good as one time pad [Kahn,(1967)]. By applying dynamic keys to group key management, it eliminates shared long term unique keys between group members and key controllers.

### 3. The proposed work

In this section, we briefly describe the architecture of our proposed sensitive information system. We will then introduce our proposed dynamic key based group key management (DKGK). Finally, we will present the proposed protocol based on DKGK

#### 3.1. Secure Sensitive Information (SIS) Architecture

The proposed secure SIS is split into several administrative areas (see Fig. 1) based on geographical locations. Each area has a local secure group controller (LSGC) to manage sensitive information sharing and accessing. The LSGCs together constitute a multicast group that maintains group key consistency by exchanging group information dynamically.

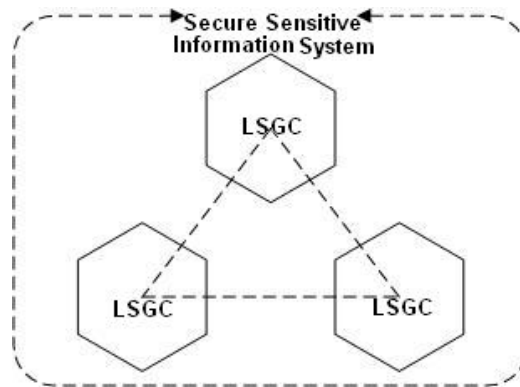


Fig. 1 Secure SIS architecture

A LSGC (see Fig. 2) comprises of a strong authentication server (**SAS**), a key server (**KS**), an access control server (**ACS**) and a record tracing server (**RTS**) to manage user/users (**U**) joining and leaving. Also, it authenticates and authorizes **U** to securely access the data from secure SIS via open networks. Moreover, it adapts a dynamic key generation technique to generate one time keys instead of **U**'s unique key encryption key (**KEK**) to enhance security of SIS.

Among the entities in the LSGC architecture, **SAS** plays a key role. It controls the verification for legitimate **U** to access the sensitive information system. For value added security feature, **U** can use either cryptographic smart cards or biometrics to authenticate **U** itself to the system.

**KS** manages both group keys and dynamic keys generation, and distributes them. It only communicates with **SAS** to dispatch keys, and then **SAS** forwards the keys to **U**. **RTS** records all inbound and outbound transaction, such as who, when and from where has access to the system and what information has been accessed. The records help the system to trace back what **U** has done in the system to achieve information non-repudiation. **ACS** performs authorization process based on security policies.

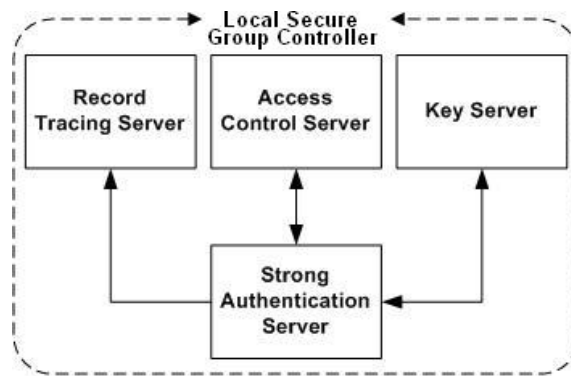


Fig 2. LSGC architecture

### 3.2. Notations

The following notations will be used throughout the rest of the paper:

Table 1: Notations

$DK_i$	dynamic key
$GK$	a list of group keys depends on key trees.
$U_{ID}$	user identity.
$G_{ID}$	group identity.
$KS$	key server
$ID_P$	identity of a participant
$I_P$	information of participant P, which can be request queries, notifications or status.
$D_{SAS}$	sensitive data from SIS
$P_X$	participant X, it can be either a user or a group
$h(X, Y)$	a hash function of message X using key Y
$\rightarrow$	unicast
$\Rightarrow$	multicast or broadcast

### 3.3. Dynamic key based group key management (DKGK)

The group key management and dynamic key theories are employed to manage keys operations and secure sensitive information communication as parts of **KS**. In traditional authentication mechanism [Oppliger,(1996)], **U** shares a unique key with **KS** as authentication and communication key. In the proposed architecture, this long term shared key is replaced by dynamic keys to improve security of systems. As security aspects, we concentrate on how to use dynamic keys with group key management protocols.

#### 3.3.1. Member Join

Join is the procedure invoked when **U** wishes to be a member of a group to access group sensitive information. Note that, **U** is able to be a member of multiple groups.

- (1) First, **U** sends a join request to **SAS** with authentication token  $h(G_{ID} + U_{ID}, DK_i)$ . Meanwhile,  $DK_i$  is only known between **U** and **SAS**.

$$U \rightarrow SAS : \{join\_req, G_{ID}, U_{ID}, h(G_{ID} + U_{ID}, DK_i)\} \quad (3)$$

- (2) After **U** authenticates to **SAS** with the token, rekeying operation is performed for backward secrecy, **SAS** multicasts the new group keys to the group members.

$$\forall U_n \in G, SAS \Rightarrow U_n : \{Key\_Update, GK_{new}\}GK_{current} \quad (4)$$

- (3) Finally, the group keys are unicasted to **U** encrypted with  $DK_{i+1}$ .

$$SAS \rightarrow U : \{GK_{new}\}DK_{i+1} \quad (5)$$

#### 3.3.2. Member Leave

Leave is the operation invoked when **U** wants to leave the group. The rekeying operation takes place for forward secrecy.

- (1) First, **U** sends a message to notify leave request, it also specifies the leaving group.

$$U \rightarrow SAS : \{leave\_req, G_{ID}\} \quad (6)$$

- (2) Then, **SAS** multicasts new updated group keys to the each group member except the leaving **U** encrypted with each remained number's dynamic key  $DK_{i+1}$

$$\forall U_i \in G, SAS \Rightarrow U_i : Key\_Update, \sum \{GK_{new}\}DK_{i+1} \quad (7)$$

Besides the join and leave rekeying operations, the employed group key management also conducts a periodic rekeying operation to increase the security of SIS.

### 3.3.3. Periodic Rekeying Operation

The periodic rekeying operation is a process to renew group keys in the system for security purpose. It does not relate to either join or leave key operations. After a period time, the group keys become vulnerable to key compromise and cryptanalysis attacks. This operation helps the system to reduce those risks.

Since active users are not necessary to know group keys but virtual cluster keys, thus, periodic rekeying operation applies to passive users only, and also it employs group key tree management. Suppose last rekeying operation occurred at time  $t$ , and a passive user with a life cycle  $[t_1, t_2]$ , then  $t_1 \leq t < t_2$ , as illustrated in fig. 3. Let  $t_0$  be the period time of security parameter depending security levels. The periodic rekeying algorithm is shown below.

- (1) First, when last rekeying operation occurs, LSGC marks the time as  $t$ .
- (2) Second, LSGC monitors if  $t_2 - t \geq t_0$ , it triggers rekeying operation.
- (3) Last, updates  $t$  to  $t + t_0$  ( $t \leftarrow t + t_0$ ). If  $t < t_2$ , it repeats step 2.

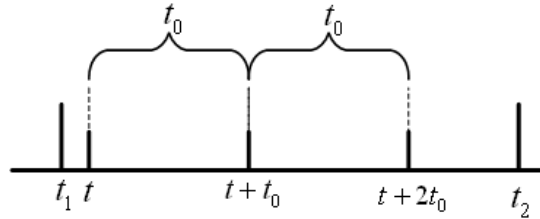


Fig 3. Periodic rekeying timeline

### 3.4. Authentication and authorization management (AAM)

The proposed SIS conducts individuals  $\mathbf{U}$  and groups'  $\mathbf{G}$  authentication and authorization mechanism using group key management to share sensitive information among group members. In the mechanism showing in Fig 4, participant  $P_X$  and  $P_Y$  can be either  $\mathbf{U}$  or  $\mathbf{G}$ . This mechanism also allows  $\mathbf{U}$  to share or access sensitive information of others. Moreover, the mechanism is able to allow  $\mathbf{U}$  to have fine-grained control on delegating access to portions of their information to others. It consists of iAuth and gAuth protocols. iAuth is the authentication and authorization protocol to individuals while gAuth is to groups.



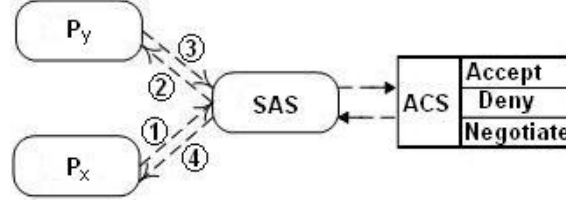


Fig 4. Authentication and Authorization Mechanism

The proposed AAM conducts individuals and groups' authentication and authorization mechanism using DKGK to secure sensitive information among group members. It can be modeled as follows.

**Definition 3.1** AAM model is a quadruple  $[U, EID, P, A(u_i, eid_i)]$ ,

where

- (1)  $U$  is a set composed of engaged users.
- (2)  $EID$  is a set composed of enciphered identities for all registered users.
- (3)  $P$  is a set composed of protocols for users to delegate authorization in the system.
- (4)  $A(u_i, eid_i)$  is a checking function which associates a Boolean value with a user  $u_i \in U$  and an identity  $eid_i \in EID$ . Such checking defines legalization of a user with regard to the  $eid_i$ .

AAM allows user or group users to share or access sensitive information of others. Also, it is able to allow users to have fine-gained control on delegating access to portions of their information to others. It consists, of initialization, logon, iAuth and gAuth, a suite of protocols. Initialization protocol is a preliminary setting for all users who registers in the system. Logon protocol is a procedure when a user wants to join the system. iAuth and gAuth protocols are authentication processes for users or group users delegating their sensitive information.

### 3.4.1. Initialization Protocol

For every user registers in the system, **KS** generates a unique random identity associating with the user. Note that, it is apart from dynamic keys management, the unique identity generation takes place in **KS**. The protocol is listed as follows.

- (1) When a user  $u_i \in U$  registers in the system.
- (2) **KS** generates a unique random identity  $id_i$  for the user  $u_i$
- (3) **KS** encrypts the unique identity by hash value of the dynamic key and  $i$  as  $eid_i$

$$eid_i = \{id_i\}h(i, DK_0) \quad (8)$$

Initially, the user only knows the value  $i$ , and **KS** holds  $id_i$ ,  $eid_i$  and  $i$ . For a high security requirement,  $id_i$  can be generated based on users' biometric identification and

stored separately from  $eid_i$ . Therefore, in **KS**, only  $eid_i$  and  $i$  will be saved, and the  $EID$  is  $\sum_{i,j \in \mathbb{N}^+} \{id_i, h(i, DK_j)\}$ , and  $j$  is an index of a corresponding dynamic key.

### 3.4.2. Logon Protocol

When a user wishes to logon the system, the following protocol is applied.

- (1) When a user sends a request to LSGC.

$$u_i \rightarrow LSGC : \{\text{logon\_request}, h(i, DK_{(j-1)})\} DK_j \quad (9)$$

- (2) After understanding the received packet, **KS** uses token as a key  $K = h(i, DK_{(j-1)})$  to decipher  $eid_i$ . If only if the enciphered value is same as  $id_i$ , then the user is legitimate, and the user can do further request, such as joining groups or requesting sensitive information. Note that, for high security requirement  $id_i$  can be stored in a different place. Then **KS** needs to send the deciphered value to verify  $id_i$ .

$$A(u_i, eid_i) \leftarrow id_i == \{eid_i\} \sim K ? true : false \quad (10)$$

- (3) Subsequently, **KS** uses the previously received dynamic key to generate a new key  $K' = h(i, DK_j)$ , and produce a new  $eid'_i$  to replace the old  $eid_i$ .

$$eid'_i = \{\{eid_i\} \sim K\} K' \quad (11)$$

Note that,  $\forall u_i \in U$ , when the dynamic key of a user is changed, step 3 will be triggered automatically in order to guarantee that  $EID$  is up-to-date.

### 3.4.3. iAuth Protocol

This protocol offers authentication and authorization for a group to an individual. It allows group members or single member of the group to share and access information of  $U_y$ . This protocol emphasizes that **U** takes control of its sensitive data. The protocol is described as follows to perform strong secure authentication and authorization.

- (1) If  $U_x$ ,  $U_x \in G_x$ , wants access to the information of  $U_y$ , he can make the query as a single user or representative of  $G_x$ .  $U_x$  generates a token  $h(I_x, DK_{xi-1})$  using previous dynamic key and  $I_x$ . Then the token is sent to **SAS** encrypted by  $DK_{xi}$  (The dynamic key is only known between  $U_x$  and **SAS**). Meanwhile,  $I_x$  contains information request query and type, where  $type \in \{single, group\}$ . Precisely,

$$U_x \rightarrow SAS : ID_x, \{I_x, h(I_x, DK_{xi-1})\} DK_{xi} \quad (12)$$

- (2) After **SAS** decrypts the request and verifies the token, **SAS** checks permissions of  $U_x$  or  $G_x$  with **ACS** depends on the type. If  $U_x$  or  $G_x$  has pre-defined **DENY** permission on  $I_x$ , **SAS** sends a rejected response. In the opposite case, if  $U_x$  or  $G_x$  is owner of the request information or has pre-defined **ACCEPT**

permission, SAS proceeds to step 4 to response the requested information. Otherwise, if there is no predefined permission in ACS, SAS will negotiate with the owner of the information  $U_Y$ . When  $U_Y$  is active as a member of a group, SAS forwards the query including the new token  $h(I_X, DK_{Y_{i-1}})$  to  $U_Y$ , and encrypted by  $DK_{Y_i}$  which is the dynamic key is known between SAS and  $U_Y$ . Precisely,

$$SAS \rightarrow U_Y : ID_{SAS}, \{I_X, h(I_X, DK_{Y_{i-1}})\}DK_{Y_i} \quad (13)$$

- (3) After obtaining the token and query from SAS,  $U_Y$  can decide permission on each selective portion of information query  $I_Y$ , and issue a new token  $h(I_Y, DK_{Y_i})$ . This token is sent back in the response message to SAS ciphered by the dynamic key  $DK_{Y_{i+1}}$ . Note that, because  $I_Y \subseteq I_X$ ,  $U_Y$  has full control of its sensitive information. Precisely,

$$U_Y \rightarrow SAS : ID_Y, \{I_Y, h(I_Y, DK_{Y_i})\}DK_{Y_{i+1}} \quad (14)$$

- (4) When SAS receives the token from  $U_Y$ , SAS retrieves the sensitive data  $D_{SAS}$  based on  $I_Y$ . If *type* is *single*, SAS unicasts  $D_{SAS}$  to  $U_X$ , encrypted by  $DK_{X_{i+1}}$ . Precisely,

$$SAS \rightarrow U_X : \{D_{SAS}, h(D_{SAS}, DK_{X_i})\}DK_{X_{i+1}} \quad (15A)$$

Otherwise, when *type* is *group*, SAS multicasts  $D_{SAS}$  to  $G_X$ , encrypted by  $GK_X$ .

$$\forall U_i \in G_X; SAS \Rightarrow U_i : \{D_{SAS}, h(D_{SAS}, GK_X)\}GK_X \quad (15B)$$

#### 3.4.4. gAuth Protocol

This protocol offers authentication and authorization for a group to another group. It allows group members or single member of the group to share and access information of another group  $G_Y$ . The protocol is described as follows:

- (1) If  $U_X$ ,  $U_X \in G_X$ , wants the access of the information of group  $G_Y$ , he can make the query as a single user or representative of  $G_X$ .  $U_X$  that sends a sharing request to SAS about joining  $G_Y$ . Meanwhile  $I_X$  consists of sharing request and type, where *type*  $\in \{single, group\}$ . Precisely,

$$U_X \rightarrow SAS : ID_X, \{I_X, G_Y, h(G_Y, DK_{X_{i-1}})\}DK_{X_i} \quad (16)$$

- (2) SAS checks with ACS about the pre-defined permission of  $I_X$  as same as iAuth. When  $U_X$  or  $G_X$  has *ACCEPT* permission, SAS then checks the request

type. If *type* is *single*,  $U_X$  joins  $G_Y$  and performs the rekeying operation. From now on,  $U_X$  belongs to both  $G_X$  and  $G_Y$  as shown in Fig 5.

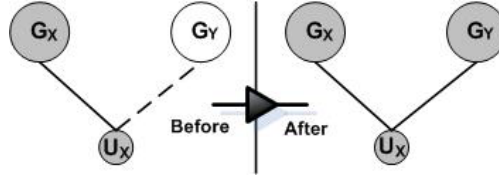


Fig 5. Individuals and Groups Authentication

Otherwise, when *type* is *group*, SAS simply merges  $G_X$  and  $G_Y$  as a group.

- (3) SAS then retrieves and broadcasts the sensitive data  $D_{SAS}$  based on  $I_X$  as follows. Precisely,

$$\forall U_i \in \{G_X, G_Y\}; SAS \Rightarrow U_i : \{D_{SAS}\}GK \quad (17)$$

#### 4. Security Analysis and Discussion

In this section, we first discuss the security of dynamic key based group key management, and then analyze the proposed work to verify their security. Finally, the AAM is compared with other authentication protocols in protecting sensitive information.

##### 4.1. Security of DKGK

In order to evaluate the security of the DKGK, the study by Kim et al. [Kim, Perrig et al.,(2004)] suggests verifying security requirement of group key management should base on group key secrecy, forward secrecy and backward secrecy. Meanwhile, forward and backward secrecies can be achieved from rekeying operation during join and leave events. Before discussion, the cryptographic properties of dynamic key should be analyzed.

One of the most important security requirements of dynamic keys management is key freshness. It means a generated dynamic key must be guaranteed to be new and used only once. Furthermore, the dynamic key should be known only to involved entities. Suppose that a set of dynamic keys is generated  $n$  times and the sequence of successive dynamic keys is  $DK = \{dk_1, dk_2, \dots, dk_n\}$ , and  $f(\cdot)$  is a one way function to generate  $DK$ . The properties are:

**Theorem 1 (Dynamic Key Secrecy)** *It guarantees that it is computationally infeasible for an adversary to discover any dynamic key  $\forall i \in \mathbb{N}^*, dk_i \in DK$ .*

**Proof.** It is obvious from the definition that the key generation algorithm is a one way function. Therefore, dynamic key generation function inherits the properties of one way function. It leads to “for any probabilistic polynomial time algorithm  $A$ , the probability that  $A$  successfully inverts  $f(x)$ , for random  $x \in_R \{0,1\}^{|x|}$  is negligible”. Thus, it is computationally infeasible for an adversary to discover any dynamic key.  $\square$

**Theorem 2 (Former Key Secrecy)** *It ensures that an adversary knows a contiguous subset of used dynamic keys (say  $\{dk_0, dk_1, \dots, dk_i\}$ ) cannot discover any subsequent dynamic keys  $dk_j$ , where  $dk_j$  is the newest generated and  $i < j$ .*

**Proof.** Assume  $n$  dynamic keys, let  $B_i$  denote the event of selecting a dynamic key from dynamic key  $i$  ( $dk_i$ ). Notice that,  $\sum_{i=1}^n B_i$  form a partition of the sample space for the experiment of selecting a dynamic key. Let  $A$  denote the event that the selected dynamic key is compromised. Therefore, based on Bayes' rule, the probability of that  $dk_j$  is compromised, is  $Pr(B_j | A) = \frac{Pr(B_j)Pr(A | B_j)}{\sum_{i=1}^n Pr(B_i)Pr(A | B_i)}$ . According to the argument in

the proof of Theorem 1, it is computationally infeasible for an adversary to discover any dynamic key. In other words, given a fresh dynamic key  $dk_j$ , the probability of compromising is  $Pr(A | B_j) = 0$ . Then,  $Pr(B_j | A) = 0$ . So it can be seen that knowing a contiguous subset of used dynamic keys does not affect security of subsequent fresh keys.  $\square$

**Theorem 3 (Key Collision Resistance)** *It means that given a dynamic key generation algorithm  $f(\cdot)$  and two initial seeds  $S_x$  and  $S_y$  ( $S_x \neq S_y$ ), the probability of key collision is negligible.*

**Proof.** Let  $\lambda$  be the probability of dynamic key collision with two different initial seeds. Then the probability of no key collision can be characterized by Poisson Distribution<sup>1</sup> [Scheaffer, (1995)].  $Pr(y) = \frac{\lambda^y}{y!} e^{-\lambda}$ ,  $y = 0, 1, 2, \dots$ , where  $y = 0$ , it means no

key collision event occurrence. Therefore, we have  $Pr(0) = \frac{\lambda^0}{0!} e^{-\lambda} = e^{-\lambda}$ . Since  $f(x)$  is a special one way function, then the probability of  $Pr(0)$  is converging towards 1. Therefore,  $\lambda \approx 0$ . It is negligible and completes the proof.  $\square$

**Theorem 4 (Key Consistency)** *It guarantees to produce sequential consistent dynamic keys  $DK$ , if given a same  $f(\cdot)$  and an initial seed.*

**Proof.** Given a same  $f(\cdot)$  and an initial seed, two entities  $P$  and  $Q$  can generate two sets of dynamic keys. Let  $B$  denote the event of having distinct initial seeds for two entities, and  $\bar{B}$  is the complement of  $B$ , which has same initial seeds for both entities. Let  $A$  denote the event of producing same output under  $f(\cdot)$ . From Theorem 3, it has the property of that the probability of two distinct inputs  $S_x$  and  $S_y$ , and the  $f(\cdot)$  producing

<sup>1</sup> In probability theory and statistics, the Poisson distribution is a discrete probability distribution that expresses the probability of a number of events occurring in a fixed period of time if these events occur with a known average rate and independently of the time since the last event

same output is negligible. Therefore, the probability of producing same output by given a  $f(.)$  and two distinct seeds is converging towards 0.

Hence,

$$Pr(B | A) \approx 0$$

Since,  $\bar{B}$  is the complement of  $B$ , according to additive and multiplicative rules of probability, we have,

$$Pr(A) = Pr(AB) + Pr(A\bar{B})$$

and

$$Pr(AB) = Pr(B)Pr(A | B)$$

Therefore, we have,

$$Pr(\bar{B} | A) = 1 - Pr(B | A)$$

It follows that

$$Pr(\bar{B} | A) \approx 1$$

Therefore, given the same seeds and  $f(.)$ , the two sets of dynamic keys are consistent.  $\square$

According the cryptographic properties of dynamic keys, the use of dynamic keys in group key management can improve the security of sensitive information systems. From DKGK description, when an adversary leaves a group, new group keys are generated from **KS** without using any previous keys information and distributed to all group members except the adversary. Therefore it is infeasible for him to discover the new group keys. However, when no rekeying operation occurs after a period of time, the group keys become less secure. Periodic rekeying operation helps to reduce risks of compromising group keys. These promise group key secrecy.

Group joining operation requires a long term unique key sharing between users and key servers. An adversary may compute the key after capturing enough messages from group key operations. The adversary then can use the key to masquerade as the user to access unauthorized sensitive information. Therefore, dynamic key technique is employed in group key management to prevent from keys compromising problem. In addition, if group joining operation is conducted by key exchange instead of sharing a long term key, group members need to establish a secure tunnel with a key server to obtain group keys. It will decrease system performance when members frequently join and leave. Hence, using dynamic key technique in group key management can overcome those problems on both aspects.

## 4.2. Security of AAM

To verify security of the proposed AAM, we take measures of confidentiality, data integrity and non-repudiation.

### 4.2.1. Confidentiality

In order to satisfy confidentiality of information security, our proposed mechanism must meet the requirement that only authorized users are able to retrieve sensitive information from the system. Therefore, to prove confidentiality of our protocols, following goal is set:

**Goal:** SAS believes U sends a legitimate token. Then U gains privilege to access the system.

**Proof:** from iAuth protocol, when  $U_X$  requests confidential information from SAS, it sends a request and a token to SAS, take step 1 as an example,  $U_X \rightarrow SAS : ID_X, \{I_X, h(I_X, DK_{X_{i-1}})\}DK_{X_i}$ . First, SAS believes it shares same dynamic key set  $\{DK_{X_i} | i \in \mathbb{N}^*\}$  with  $U_X$ . Then SAS can decrypt and see the received message and the token  $\tau (h(I_X, DK_{X_{i-1}}))$ . Last, SAS uses previous key  $DK_{X_{i-1}}$  and received  $I_X$  to generate a new token  $\tau' = h(I_X, DK_{X_{i-1}})$ . Then SAS compares the generated token  $\tau'$  with the received one  $\tau$ . Only if the two tokens are same  $\tau \equiv \tau'$ ,  $U_X$  is authenticated.  $\square$

The proof of confidentiality is successful, and proof of gAuth protocol is same as iAuth.

#### 4.2.2. Data Integrity

Data integrity refers to the validity of data. It can be compromised through malicious altering and accidental altering. In AAM, hash functions are used to promise data integrity. When the data is changed, the hash function yields a different result.

#### 4.2.3. Non-repudiation

Non-repudiation can be achieved in AAM with the use of the dynamic key generation technique. When U sends requests for sharing others sensitive information, or U gives permission for other U or G to access his sensitive information, a token needs to be generated and sent to SAS. The token is constructed by a unique dynamic key, which is only known between SAS and U. Therefore, it eliminates U denying issuing permissions or sending requests.

In addition, the token is dynamically generated and only used once. So it eradicates security threat of sniffing attacks. Furthermore, in our system, RTS records every transaction occurred in the system. Therefore, U also cannot deny what he has done in the system based on the nonce token

### 4.3. Comparison of Techniques

In this section, the Kerberos, EAP, Iolus and Limited-Used Key Generation Scheme (LKG) are compared with the proposed protocol in protecting sensitive information to evaluate the security of AAM.

Our proposed AAM supports for both individual and group authentication based on group key management and dynamic key technique. The security of sensitive information system is greatly improved in comparing to the other techniques. Besides security aspect, AAM also allows individual and group users to negotiate and delegate the access permission with owners of sensitive information. It emphasizes privacy protection, which the owners take full control of their sensitive information. It also supports non-repudiation by adapting dynamic key technique and RTS. In summary, the comparison in Table 2 demonstrates that AAM has advantages over other mechanisms in protecting sensitive information.

Table 2: Techniques Comparison of Protecting Information Mechanism

Criterion		Kerberos	EAP	LKG	AAM
Authentication	Individual	Yes	Yes	Yes	Yes
	Group	N/A	N/A	No	Yes
Cryptographic Keys		Session key & long term key	Session key & long term key	Dynamic Key	Group key & Dynamic Key
Delegation Support		No	No	No	Yes
Non-Repudiation		Yes	Yes	Yes	Yes

## 5. Conclusions

In this paper, we have pointed out the problems and limitations of current authentication protocols for sensitive information systems. We then proposed a secure authentication & authorization protocol using dynamic key based group key management. We have shown that the proposed protocol is secure against key compromise and it allows owners to have fine-grained control to their sensitive information. Our proposed protocol is flexible for individuals and groups authenticating and authorizing while also protect privacy of information owners.

In future work, we aim to adopt AAM protocol and merge dynamic keys and sensitive data in order to enhance security of sensitive information.

## 6. Acknowledgements

The authors wish to thank the anonymous reviewers and the editor for their constructive criticism and insightful comments, and also thank Mr Minh The Ngo for proofreading, which helped to improve the paper

## References

- Amir, Y., C. Nita-Rotaru, et al. (2001). Framework for Authentication and Access Control of Client-Server Group Communication Systems. Proceedings of the Third International COST264 Workshop on Networked Group Communication, Springer-Verlag London, UK.pp: 128 - 140.
- Bresson, E., O. Chevassut, et al. (2004). "Mutual authentication and group key agreement for low-power mobile devices." Computer Communications (Security and Performance in Wireless and Mobile Networks) V 27(I 17), pp: 1730-1737
- Challal, Y. and H. Seba (2005). "Group Key Management Protocols: A Novel Taxonomy." International Journal of Information Technology V 2(I 1), pp: 105-118.
- Dandash, O., Y. Wang, et al. (2007). A new Dynamic Key Generation Scheme for Fraudulent Internet Payment Prevention. Proceedings of the International Conference on Information Technology, Las Vegas, Nevada, USA, IEEE Computer Society Washington, DC, USA.pp: 83-88.
- Edney, J. and W. A. Arbaugh (2003). Real 802.11 Security: Wi-Fi Protected Access and 802.11i Addison-Wesley Professional.
- Kahn, D. (1967). The Codebreakers: The Story of Secret Writing New York, Macmillan Pub Co.



- Kim, Y., A. Perrig, et al. (2004). "Tree-based group key agreement." *ACM Transactions on Information and System Security (TISSEC)* V 7(I 1), pp: 60 - 96.
- Kungpisdan, S., P. D. Le, et al. (2005). "A Limited-Used Key Generation Scheme for Internet Transactions." *Lecture Notes in Computer Science* V 3325(I, pp: 302-316.
- Li, Y. and X. Zhang (2004). A Security-Enhanced One-Time Payment Scheme for Credit Card. *Proceedings of the 14th International Workshop on Research Issues on Data Engineering: Web Services for E-Commerce and E-Government Applications (RIDE'04)*, Washington, DC, USA, IEEE Computer Society. pp: 40 - 47.
- Loscocco, P. A. and S. D. Smalley (2001). Meeting Critical Security Objectives with Security-Enhanced Linux. *Proceedings of 2001 Ottawa Linux Symposium*, Ottawa Canada.
- Loscocco, P. A., S. D. Smalley, et al. (1998). The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments. *Proceedings of 21st National Information Systems Security Conference*, Hyatt Regency Crystal City Arlington, Virginia, USA. pp: 303-314.
- Mao, W. and C. Boyd (1994). On strengthening authentication protocols to foil cryptanalysis. *Computer Security — ESORICS 94*, Springer Berlin / Heidelberg, pp: 193-204.
- Menezes, A. (1996). *Handbook of Applied Cryptography*, CRC Press.
- Mitra, S. (1997). Iolus: A Framework for Scalable Secure Multicasting. *Proceedings of ACM Special Interest Group on Data Communication (SIGCOMM) Cannes, French Riviera, FRANCE*. pp: 277-288.
- Moyer, M. J., J. R. Rao, et al. (1999). "A survey of security issues in multicast communications." *IEEE Network* V 13(I 6), pp: 12-23.
- Neuman, C. and J. G. Steiner (1988). Authentication of Unknown Entities on an Insecure Network of Untrusted Workstations. *Proceedings of the Usenix Workshop on Workstation Security*. pp: 1-11.
- Ngo, H. H., X. Wu, et al. (2008). A Group authentication model for wireless network services based on group key management. *proceeding of International Conference on Enterprise Information Systems (ICEI08)*, Spain. pp: 182-188.
- Oppliger, R. (1996). *Authentication Systems for Secure Networks*. Norwood, MA , USA Artech House Publishers.
- Osborn, S., R. Sandhu, et al. (2000). "Configuring role-based access control to enforce mandatory and discretionary access control policies." *ACM Transactions on Information and System Security (TISSEC)* V 3(I 2), pp: 85 - 106.
- Rubin, A. D. and R. N. Wright (2002). Off-Line Generation of Limited-Use Credit Card Numbers. *Proceedings of the 5th International Conference on Financial Cryptography*, London, UK, Springer-Verlag. pp: 196 - 209.
- Sandhu, R., D. Ferraiolo, et al. (2000). The NIST model for role-based access control: towards a unified standard. *Proceedings of the fifth ACM workshop on Role-based access control*, Berlin, Germany, ACM New York, NY, USA. pp: 47-63.
- Scheaffer, R. L. (1995). *Introduction to Probability and Its Applications*, Duxbury Press, Wadsworth Publishing Company.
- Shannon, C. (1949). "Communication Theory of Secrecy Systems." *Bell System Technical Journal* V vol.28(I 4), pp: p 656-715.

- Shields, C. and J. J. Garcia-Luna-Aceves (1999). KHIP—a scalable protocol for secure multicast routing. Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication, Cambridge, Massachusetts, United State, ACM New York, NY, USA.pp: 53 - 64.
- Talbot, J. and D. Welsh (2006). Complexity and Cryptography-An Introduction. New York, Cambridge Univeristy Press
- Wong, C. K. and S. S. Lam (2000). Keystone: a group key management service. Proceedings of International Conference on Telecommunications, ICT, Acapulco, Mexico.