16

Security Architecture for Sensitive Information Systems

Xianping Wu, Phu Dung Le and Balasubramaniam Srinivasan
Monash University
Australia

1. Introduction

The use of information has become a pervasive part of our daily life; we have become "… an information society" (Gordon & Gordon, 1996). Employees use information to make personal choices and perform basic job functions; managers require significant amounts of it for planning, organizing and controlling; corporations leverage it for strategic advantage. Since the application of computers in administrative information processing began in 1954 (Davis & Olson, 1985), computers have become a key instrument in the development of information processing. The rapid development of information technology (IT) has helped to firmly establish the general attitude that information systems are a powerful instrument for solving problems.

Utilizing these emerging technologies, however, is not without problems. People start considering their sensitive information when it is transmitted through open networks; managers begin worrying about using forged information for business plans; and corporations worry about customer and investor confidence if they fail to protect sensitive information. Protecting sensitive information has consequently become a top priority for organizations of all sizes.

Despite this priority, the majority of existing sensitive information systems (Bacon & Fitzgerald, 2001; Bhatia & Deogun, 1998; Hong et al., 2007) focus on performance and precision of data retrieval and information management. A number of techniques are employed to protect information systems; however, in many cases, these techniques are proving inadequate. For example, while several information systems (Beimel et al., 1999; Cachin et al., 1999; Gertner et al., 1998; Gertner et al., 2000) use the add-ons security features to provide information confidentiality (which allow users to share information from a data media while keeping their channel private), these security measures are insufficient.

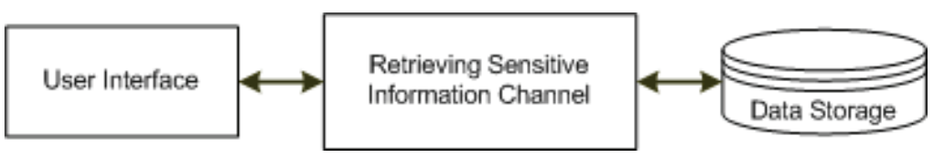

Fig. 1. The Architecture of Generic Sensitive Information Systems

As shown in Fig.1, generic sensitive information systems consist of - *communication channel*, *user interface* and *sensitive information storage* - three major components, and they are all

Source: Convergence and Hybrid Information Technologies, Book edited by: Marius Crisan,
ISBN 978-953-307-068-1, pp. 426, March 2010, INTECH, Croatia, downloaded from SCIYO.COM

potential targets for adversaries wanting to benefit from security weaknesses. Therefore, in following sections, existing approaches, main issues and limitations relating to sensitive information protection are investigated.

1.1 Related work and limitations

According to the process of sensitive information retrieving, several security aspects need to be studied. Firstly, securing *communication channel*, it applies cryptography and security tunnels to protect message between entities. Secondly, securing *user interface*, it uses authentication mechanisms to prevent unauthorized access to sensitive information. Thirdly, securing *sensitive information storage*, it uses cryptographic keys to encrypt all sensitive information before storing it.

1.1.1 Securing communication channel

In cryptography, a confidential channel is a way of transferring data that is resistant to interception, but not necessarily resistant to tampering. Conversely, an authentic channel is a way of transferring data that is resistant to tampering but not necessarily resistant to interception (Tienari & Khakhar, 1992). Interception and tampering resistance is best developed through communication channel.

In order to reach the interception resistance goal, all communication is scrambled into ciphered text with a predetermined key known to both entities to prevent an eavesdropper from obtaining any useful information. In order to achieve the tampering resistance goal, a message in a communication is assembled using a credential such as an integrity-check to prevent an adversary from tampering with the message.

In this section, the different approaches of securing communication channel are investigated, and their pros and cons are evaluated. The investigation is conducted by subdividing *communication channel* into unicast channel and multicast channel

Secure Communication in Unicast Channels: With the recent development of modern security tools to secure bidirectional communication between two entities, many protocols, such as Internet Protocol Security (IPsec) (Atkinson, 1995), Secure Sockets Layer (SSL), Transport Layer Security (TLS) (Dierks & Rescorla, 2008; Freier et al., 1996) and Secure Real-time Transport Protocol (SRTP) (Lehtovirta et al., 2007), have been proposed in the literature to address the problems and challenges of a secure unicast *communication channel*. One of the most important factors in unicast *communication channel* protection is the cryptographic key. The issues of key distribution and key type, therefore, determine the security of the unicast communication channel.

IPsec and SSL/TLS are the most famous, secure and widely deployed among all the protocols for protecting data over insecure networks. IPsec is a suite of protocols for protecting communications over Internet Protocol (IP) networks through the use of cryptographic security services. It supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection. However, in IPsec, communication is protected by session keys, and the security of a session key is guaranteed by long-term shared keys. Therefore, once the long-term keys are compromised, the security of IPsec is under threat. As Perlman and Kaufman (2001) indicated, IPsec is vulnerable to dictionary attack, due to the pre-shared long-term keys, and Ornaghi and Valleri (2003) demonstrated it in a BlackHat conference.

Moreover, in IPsec, the long-term shared keys involve into key exchange protocol to generate session keys. According to information entropy (Gray, 1990), the uncertainty of key

materials decreases when the use of the key materials in generation session keys is frequent. This leads to the key materials (that is, the long-term shared keys) being exposed.
SSL/TLS are cryptographic protocols that provide security and data integrity for unicast communications over insecure networks to prevent eavesdropping, tampering, and message forgery by employing pre-shared master secret (long-term shared key). That the master secret remains truly secret is important to the security of SSL/TLS. However, in the protocol design, the usage of master secret involves multiple phases, such as session key generation, certificate verification and cipher spec change (Wagner & Schneier, 1996).
On the top of the above concerns, the SSL/TLS protocols suffer from different types of flaws (Micheli et al., 2002): identical cryptographic keys are used for message authentication and encryption, and no protection for the handshake, which means that a man-in-the-middle downgrade attack can go undetected. Although a new design of SSL/TLS overcomes a few flaws, as (Bard, 2004; Wagner & Schneier, 1996) state, an attacker can use plaintext attacks to break SSL/TLS protocols due to the long-term shared identical cryptographic keys.
Secure Communication in Multicast Channels: As group-oriented communication systems become more widespread, sensitive information confidentiality is an issue of growing importance for group members. To achieve confidential communication in a multicast channel, cryptographic keys are employed to secure the multicasted contents. The keys (or the group key) must be shared only by group members. Therefore, group key management is important for secure multicast group communication. In modern group key management - Logical Key Hierarchy (LKH) (Harney & Harder, 1999), One-way Function Tree (OFT) (Sherman & McGrew, 2003), Iolus (Mittra, 1997) - for sensitive information systems requires group keys to have a number of characteristics: group key secrecy, backward secrecy, forward secrecy and group key independency. In addition, modern management also requires flexible and efficient rekeying operations and privacy for group members (Kim et al., 2004).
However, Challal and Seba (2005) imply that the major problems of group key management are confidentiality, authentication and access control. Also, there are no solutions to dedicate privacy protection for group members and confidentiality for sensitive information systems. Moreover, when a user joins a group, the new group keys are unicast to the user encrypted by a pre shared long-term key. It raises risks of sensitive information systems associated with the compromise of the long term key.

1.1.2 Securing user interface

The common security mechanism to protect *user interface* in sensitive information systems is authentication. Authentication is "the process of confirming or denying a user's claimed identity, which can be proved by knowledge, possession and property factors" (Meyers, 2002). This form of security uses measures such as security tokens (something users have), passwords (something users know) or biometric identifiers (something users are).
Kerberos (Steiner et al., 1988) is a representative knowledge factor based authentication protocol which allows individuals communicating over a non-secure network to prove their identity to one another in a secure manner. In the original design of Kerberos, session keys exchange used long-term shared keys. Although researchers (Erdem, 2003; Harbitter & Menascé, 2001; Sirbu & Chuang, 1997) proposed the use of public key cryptography to enhance security for key exchange and authentication, the long-term shared key is still a limitation of Kerberos-based information systems (Kohl et al., 1994). In 2008, Cervesato et

al. (2008) pointed out that man-in-the-middle attack can breach Kerberos-based information systems.

Other authentication factors based authentication protocols suffer security threats when the physical devices (security tokens and smart card) are lost or stolen or the biometric sources (fingerprint and retinal pattern) are compromised. Moreover, privacy is another concern; how biometrics, once is collected, can be protected.

By briefly investigating the extant authentication approaches in sensitive information systems, there is no proper technique to protect *user interface* in the process of sensitive information retrieving. Moreover, the extant authentication approaches are not able to manage dynamic group member authentication and authorization while allowing individuals to share their sensitive information without sacrificing privacy.

1.1.3 Securing sensitive information storage

Data encryption is a security mechanism to protect sensitive information at rest. It depends on a long-term shared key to cipher all critical information at rest (*sensitive information storage*). For example, IBM employs symmetric keys in z/OS to protect the sensitive information documents, and uses public keys to wrap and unwrap the symmetric data keys used to encrypt the documents. With this technique, IBM claims that many documents can be generated using different encryption keys (Boyd, 2007).

Similar mechanisms are also used for Oracle Database and Microsoft SQL Server, which conduct critical information protection via long-term shared keys. The security of the IBM mechanisms relies on public key infrastructure; if the public key pairs are disclosed, no matter how many different encryption keys are used to protect information, the whole information system will be compromised. In addition, the security of Oracle and Microsoft mechanisms depend on a long-term database master key; the sensitive information may be revealed if the database systems are breached.

It can be seen that no technique can ensure privacy protection, and also that the security of those techniques relies on long-term keys and public keys. Also, none of the existing approaches to protecting information storage can manage dynamic ownership of sensitive information (for example, in the case that a user loses the asymmetric key in z/OS or that the ownership of sensitive information is changed in a database).

1.1.4 Summary

Sensitive information systems consist of three major components: *communication channel*, *user interface* and *sensitive information storage*; the protection of these three components equates to the protection of sensitive information itself. Previous research in this area has been limited due to the employment of long-term shared keys and public keys. Currently, no complete security solution exists to help protect sensitive information in the three components. Issues such as dynamic sensitive information ownership, group authentication and authorization and privacy protection also create challenges for the protection of sensitive information systems.

In response to these limitations, we therefore propose a novel security architecture for sensitive information systems to tackle the problems and challenges.

1.2 Organization of the chapter and contributions

The rest of the chapter is organized as follows: Section 2 proposes formal security architecture for sensitive information systems. Section 3 details the components of the

proposed secure architecture. Section 4 gives a formal and thorough security analysis and discussion of the components. Section 5 concludes and provides future researcher directions.

Contributions: This research contributes to the development of the body of knowledge surrounding sensitive information protection. Its contributions include the following:

- Formal definition and cryptographic properties proofs of dynamic keys
  This thesis offered a first formal definition of dynamic keys with the following proved cryptographic properties: dynamic key secrecy, former key secrecy, key collision resistance and key consistency. The formal definition and the cryptographic properties can also be used as a guide to design new dynamic key generation algorithms. More importantly, the formal definition gives a distinct semantic notion to distinguish dynamic keys from other cryptographic keys, such as session keys, one-time pad and long-term keys.
- A new proposed security architecture for sensitive information systems
  This research proposed a novel security architecture by the employment of dynamic key and group key theories to overcome the security threats and concerns of sensitive information systems in the components of *communication channel*, *user interface* and *sensitive information storage*. The architecture can be applied to security applications all sectors, including the business, healthcare and military sectors, to protect sensitive information.

As a result of these contributions, we claim that the proposed security architecture for sensitive information systems protects *communication channel*, *user interface* and *sensitive information storage*. The architecture provides strong authentication and authorization mechanisms to conduct dynamic membership of groups and individuals to share or access sensitive information. It also prevents legal users accessing unauthorized sensitive information against internal security threats. The architecture achieves strong protection for sensitive information at rest in order to overcome security threats that compromise credentials of information systems. Furthermore, it is able to handle dynamic information ownership. Finally, the proposed architecture achieves privacy protection and includes a feature to detect and prevent intrusion.

2. Security architecture for Sensitive Information Systems (SecureSIS)

2.1 Dynamic key theory

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, which stands for number used once (Anderson, 2001). The use of dynamic keys introduces complications, such as key synchronization, in cryptographic systems. However, it also helps with some problems, such as reducing key distribution and enhancing key security. There are three primary reasons for the use of dynamic keys in sensitive information protection.

First, securing sensitive information by using long-term symmetric keys makes sensitive information systems more vulnerable to adversaries. In contrast, using dynamic keys makes attacks more difficult. Second, most sound encryption algorithms require cryptographic keys to be distributed securely before enciphering takes place. However, key distribution is one of the weaknesses of symmetric key algorithms. Although asymmetric key algorithms do not require key distribution, they are, in general, slow and susceptible to brute force key search attack. This situation can be improved by using asymmetric key algorithms once only

to distribute an encrypted secret. Dynamic keys can then be generated based on the secret and other key materials. This process can improve the overall security considerably. Last, but not least, security tokens can be generated by either long-term symmetric keys or nonce dynamic keys. Even though both methods generate variational tokens every time, the dynamic key method is more difficult to break than the long-term key method.

In accordance with the primary reasons for using dynamic keys in sensitive information protection, it is necessary to have an unambiguous and formal definition. The notion of a one-way function (Menezes et al., 1996) is used for reference. This is defined as "... 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 computationally infeasible to find any x such that $y = f(x)$." Formally, a function $f : \{0,1\}^* \to \{0,1\}^*$ 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 & Welsh, 2006). Therefore, dynamic keys can be defined as follows:

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

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

where,

$$\forall x, y(x \neq y), \neg(\exists f^i(x) = f^i(y)) \tag{2}$$

The special one-way function dynamic key generation scheme (Kungpisdan et al., 2005; Li & Zhang, 2004) has been proposed. However, the formal proofs have never been given; consequently, having formally defined dynamic keys, the cryptographic properties of dynamic keys are discussed and proved.

One of the most important security requirements of dynamic keys theory is key freshness. This means a generated dynamic key must be guaranteed to be new and able to be used only once. Furthermore, a dynamic key should be known only to involved entities. Therefore, four important security properties of dynamic keys (dynamic key secrecy, former key secrecy, key collision resistance and key consistency) are given.

Suppose that a set of dynamic keys is generated n times and the sequence of successive dynamic keys is $DK = \{dk_1, dk_2, ..., dk_n\}$ and $f(.)$ is a special one-way function to generate DK. The properties are:

Theorem 2.1 (Dynamic Key Secrecy) Dynamic key secrecy guarantees that it is computationally infeasible for an adversary to discover any dynamic key $\forall i \in \mathbb{N}, dk_i \in DK$.

Proof: From the definition it is apparent that the key generation algorithm is a one-way function. The dynamic key generation function therefore inherits the properties of the one-way function with the consequence that "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. □

Theorem 2.2 (Former Key Secrecy) Former key secrecy ensures that an adversary, who knows a contiguous subset of used dynamic keys (say $\{dk_0, dk_1 ... dk_i\}$), cannot discover any subsequent dynamic keys dk_j, where dk_j is the newest generated and $i < j$.

Proof: Assuming 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 that dk_j is compromised is $Pr(B_j \mid A) = \frac{Pr(B_j)Pr(A \mid B_j)}{\sum_{i=1}^{n} Pr(B_i)Pr(A \mid B_i)}$. According to the argument in the proof of Theorem 2.1, 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 this key being compromised is $Pr(A \mid B_j) = 0$, and $Pr(B_j \mid A) = 0$. Even if a contiguous subset of used dynamic keys becomes known, the security of subsequent fresh keys will not be affected. □

Theorem 2.3 (Key Collision Resistance) Key collision resistance means that given a dynamic key generation algorithm, $f(.)$, and two initial seeds, S_X and S_Y ($S_X \neq S_Y$), the probability of key collision is negligible.

Proof: Let λ be the probability of dynamic key collision with two different initial seeds. The probability of no key collision can then be characterized by a Poisson Distribution (Scheaffer, 1994): $Pr(y) = \frac{\lambda^y}{y!} e^{-\lambda}, y = 0,1,2...$. Where $y = 0$, no key collision event can occur and 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 $Pr(0)$ converges towards 1 and $\lambda \approx 0$. The value is negligible and completes the proof. □

Theorem 2.4 (Key Consistency) Key consistency guarantees to produce sequential, consistent, dynamic keys DK, if given the same $f(.)$ and an initial seed.

Proof: Given the same $f(.)$ and an initial seed, two entities P and Q can generate one set of dynamic keys. Let B denote the event of having distinct initial seeds for two entities. $\overline{B}$ is the complement of B, which has same initial seeds for both entities. Let A denote the event of producing the same output under $f(.)$. From Theorem 2.3, the probability of the two distinct inputs, S_X and S_Y, and the $f(.)$ producing the same output is negligible. The probability of producing the same output by a given $f(.)$ and two distinct seeds therefore converges towards 0. Hence, $Pr(B \mid A) \approx 0$. Since $\overline{B}$ is the complement of B, according to additive and multiplicative rules of probability, we have $Pr(A) = Pr(AB) + Pr(A\overline{B})$. Thus, $Pr(\overline{B} \mid A) = 1 - Pr(B \mid A)$. It follows $Pr(\overline{B} \mid A) \approx 1$. Therefore, given the same seeds and $f(.)$, the two entities can generate the same set of dynamic keys. □

2.2 Security architecture

Security architecture (SecureSIS) consists of four "tangible" components: dynamic key management (DKM) , user-oriented group key management (UGKM) (Wu et al., 2008b), authentication and authorization management (AAM) (Wu et al., 2009) and sensitive information management (SIM) (Wu et al., 2008a), and two "intangible" components: security agreement (SA) and security goals (Goals). DKM is the security foundation of

SecureSIS. It manages dynamic keys for other components to secure *communication channel*, *user interface* and *sensitive information storage* in the process of sensitive information retrieving.

In SecureSIS, two sets of dynamic keys are employed for engaging users (U) to protect their sensitive information and privacy. One is dynamic data key set DK_X, which is used to integrate with (encrypt) sensitive information at rest. Another is dynamic communication key set DK_Y, which is used to secure communication and generate tokens for authentication. In addition, there is no sensitive information at rest for "tangible" components. Hence, only one set of dynamic keys (component dynamic keys) conducts the security of communication channel among components.

UGKM is a membership management in SecureSIS. It is a novel hybrid group key management approach to govern dynamic membership and protect user privacy and multicast communication secrecy. Together with DKM, unicast communication channel for individuals and multicast communication channel for group members are protected.

AAM manages authentication and authorization for individuals and group members to protect user interface. The employment of DKM and UGKM makes the AAM secure and flexible to deal with group authorization, individual privacy protection.

SIM uses dynamic data keys to integrate with sensitive information at rest in order to protect sensitive information storage. It guarantees the breach of SIS does not have negative impact on the security of sensitive information itself. Also, SIM manages sensitive information ownership by applying UGKM to ensure the utility of sensitive information.

SA component guarantees the security of sensitive information in SecureSIS, if, and only if the sensitive information satisfies the agreement.

Goals component is security expectations of SecureSIS. According to the process of sensitive information retrieving, this component consists of user interface's goal, communication channel's goal and sensitive information storage's goal.

In order to protect sensitive information (called ***I***), the security architecture, SecureSIS, can be characterized as follows:

Definition 2.2 (SecureSIS) Security architecture is defined as a union of the following sets:

$$SecureSIS = [U, AAM, UGKM, SIM, DKM, SA, Goals] \tag{3}$$

where,

i. U is a set composed of engaged users who require sensitive information ***I***.
ii. AAM is a set of authentication and authorization management objects for verifying U and allowing U to delegate authorization in order to protect *user interface*.
iii. UGKM is a user-oriented group key management object for providing secure *communication channel* in order to secure ***I*** sharing among subsets of U.
iv. SIM is a set of sensitive information management objects for protecting *sensitive information storage*.
v. DKM is a set of dynamic key management objects for providing and managing dynamic keys of U, AAM, UGKM and SIM.
vi. SA stands for the security agreement associated with ***I***. It is a notional inner relationship between U and ***I***.
vii. Goals represents security goals of architecture regarding ***I*** protection.

To illustrate the conceptual architecture based on the definition of SecureSIS, AAM, UGKM, SIM and DKM can be thought as "tangible" objects to protect ***I***. These objects are therefore

components of SecureSIS architecture. In addition, SA and Goals are "intangible", thus, the tangible conceptual architecture is illustrated in Fig. 2.

Fig. 2. Tangible Conceptual Architecture of SecureSIS.

The set of engaged users, U, is a key component in SecureSIS. Every user owns or shares sensitive information. To protect sensitive information, the security of each single user needs to be scrutinized. In order to protect the privacy of each individual, U is classified into two categories: passive users, ω, and active users, ϖ. Passive user is inert and infrequently joins and leaves the system. In SecureSIS, ω does not share its own sensitive information with others, but accesses the sensitive information of ϖ. Active user is vigorously and frequently joins and leaves the system. ϖ needs to share sensitive information with ω therefore, it needs high privacy protection. Meanwhile, by a request, ω can be transformed into ϖ and vice versa ($\omega \bigcap \varpi = \varnothing$).

SecureSIS is split into several administrative areas. Each area has a local secure group controller (LSGC) associated with a subgroup to manage ***I*** sharing and accessing. The controllers together constitute a multicast group (UGKM) that maintains group key consistency by exchanging group information dynamically and securely. The communication structure of SecureSIS is shown in Fig.3.

In SecureSIS, the SA is the "contract" that governs the relationships between sensitive information ***I*** and owners (U) in a secured transaction (for example, information accessing and sharing). The SA classifies sensitive information into a number of levels following information classification, and then assigns access rules to each information object.

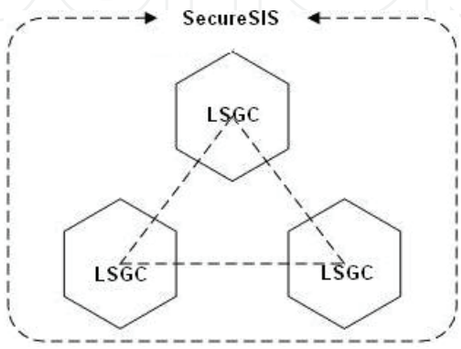

Fig. 3. The Structure of SecureSIS.

When designing security architecture for sensitive information systems, sensitive information protection is the primary consideration. Sensitive information must be stored safely (*sensitive information storage*), transmitted securely (*communication channel*) and made available only to authenticated and authorized (*user interface*) users. Such desires can be defined as security goals of SecureSIS.

User Interface's Goal (UIG): Sensitive information must only be disclosed to legitimate users with proper permissions and genuine sensitive information systems.

Communication Channel's Goal (CCG): Sensitive information must be identically maintained during transmission via open networks.

Sensitive Information Storage's Goal (SISG): Sensitive information must be stored securely and satisfy the requirement that only privileged users can understand and retrieve the information.

3. Security architecture components

3.1 Dynamic Key Management (DKM)

The reason for the employment of two sets of dynamic keys is that dynamic data keys are only used to integrate into sensitive information at rest (encryption), and dynamic communication keys are used only for token generation and commination protection. The two sets of dynamic keys are independent. According to the single-use nature and cryptographic properties of dynamic keys, the breach of one set of dynamic keys does not compromise the security of SecureSIS. Formally:

Definition 3.1 (Dynamic Key Management) Dynamic keys management is a quadruple $[DK_X, DK_Y, CDK, G(.)]$, where:

i. DK_X is a set composed of dynamic data keys $\{dk_{Xi} \mid i \in \mathbb{N}\}$ of users for securing sensitive information storage. Given $u_n \in U$, the dynamic data key set for user u_n is:

$$DK_X = \{dk_{Xi}.u_n \mid i \in \mathbb{N}\} \tag{4}$$

ii. DK_Y is a set composed of dynamic communication keys of users for protecting user interface and communication channel. Given $u_n \in U$, the dynamic communication key set for user u_n is:

$$DK_Y = \{dk_{Yj}.u_n \mid i \in \mathbb{N}\} \tag{5}$$

iii. CDK is a set composed of dynamic keys of each components for securing communication between DKM and AAM & SIM. Given $aam_m \in AAM, dkm_k \in DKM \text{ and } sim_n \in SIM$, the component dynamic key set for aam_m, dkm_k and sim_n is $\{cdk_i.aam_m \mid i \in \mathbb{N}\}$, $\{cdk_j.sim_n \mid i \in \mathbb{N}\}$ and $\{cdk_l.dkm_k \mid i \in \mathbb{N}\}$, respectively.

iv. $G(.)$ is a dynamic key generation scheme. It generates dynamic keys synchronously with U and other components in SecureSIS.

In order to make good use of dynamic key properties, the following agreements apply:

- For users, a user sharing DK_X and DK_Y with SecureSIS does not necessarily mean that the user has registered and is legitimate.
- For users, dynamic data keys do not involved in any communication. The keys are strictly used to wrap and unwrap sensitive information only.

- For both users and "tangible" objects, dynamic communication keys are used to generate security tokens and encipher communications.
- For objects, dynamic communication keys of users are generated via DKM, and transmitted securely via dynamic communication keys of objects.
- For both users and objects, a network failure caused by asynchronous dynamic communication keys will trigger a network fault heal event (Ngo et al., 2010). The event can be performed via negotiating dynamic key counters $\{Yj \mid j \in N\}$.

3.2 User-oriented Group Key Management (UGKM)

Every user in SecureSIS is managed via this component, and it applies a hierarchical structure to secure multicast communication channel. It is a top-down structure and consists of a root, subgroups (SG), clusters (C) and leaves (associated with users U).

The passive users ω are initially aggregated into clusters, at the upper level, called subgroups. Each cluster selects one of its members as the cluster leader to be the representative. The active users ϖ cannot join clusters, but virtual clusters. Each virtual cluster is a virtual container to accommodate involved ω and ϖ. When an active user joins, a member (passive user) of a closed cluster forms a virtual cluster under the same subgroup node. The member (passive user) is called virtual leader for the virtual cluster. The component is characterized as follows:

Definition 3.2 (User-oriented Group Key Management) User-oriented group key management is a septuple $[\omega, \varpi, C, VC, L, VL, Alg(U)]$, where:

i. VC (virtual cluster) is a set composed of virtual containers to accommodate involved ω and ϖ. An active user can only join (belong to) one virtual cluster; however, a passive user can belong to a subset of virtual clusters, such that,

$$\begin{aligned} &\forall \varpi_i \in \varpi, \exists! vc_j \in VC : \varpi_i \in vc_j \\ &\forall \omega_i \in \omega, \exists \textit{ at least one } vc_j : \omega_i \in \bigcup_{j \in N} vc_j \end{aligned} \tag{6}$$

ii. L (leader) is a set composed of leaders $L \subset \omega$ for authentication as representatives of clusters, used in AAM.

iii. VL (virtual leader) is a set composed of virtual leaders $VL \subset \omega$ for constructing virtual clusters and managing key operations.

iv. $Alg(U)$ is a suite of algorithms that manages U join and leave rekeying operations.

3.2.1 Key tree structure

As discussed in Section 1.1.1, since the drawbacks of the existing multicast communication channel approaches. The UGKM scheme must guarantee privacy protection for group members and confidentiality for sensitive information systems. It must also be suitable for groups with a large number of members. Therefore, UGKM is a two-tier hybrid group key management that focuses on privacy protection and confidentiality of sensitive information. Fig.4. depicts the logical structure of UGKM.

UGKM is divided into two levels: the passive user level (key tree distribution scheme) and the active user level (contributory group key management scheme). The passive user level consists only of passive users who participate in sensitive information sharing and accessing of other active users. As mentioned in Section 2.2, if a passive user wants to share its

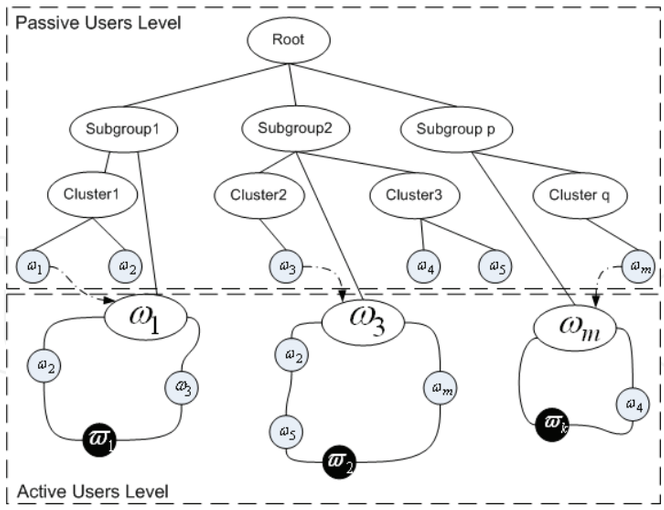

Fig. 4. Logical Structure of UGKM

sensitive information, the user must transform into an active user. When an active user joins the system, one of passive users will be promoted to leader to construct a dynamic virtual cluster under the subgroup.

3.2.2 Member join

In SecureSIS, users are categorized into passive and active users. Also, active users can only join virtual clusters. Therefore, there are three scenarios: an active user joins the system, a passive user joins a cluster and a passive user joins an existing virtual cluster.

Active User Joins. When an active user (ϖ_1 in Fig. 5) wishes to join the group, it applies the active user level key distribution agreement. Since a new virtual cluster is created, it does not need backward secrecy and the join procedure starts with an active user join request:

i. First, ϖ_1 contacts a LSGC, and the LSGC forwards the request to AAM for authentication via a secure unicast channel.
ii. After successful verification, one of the passive users (say ω_1) is selected as a leader. Then ω_1 constructs a dynamic virtual cluster $vc_i \in VC$ that connects all relevant members (say ω_2, ϖ_1 and ω_3).

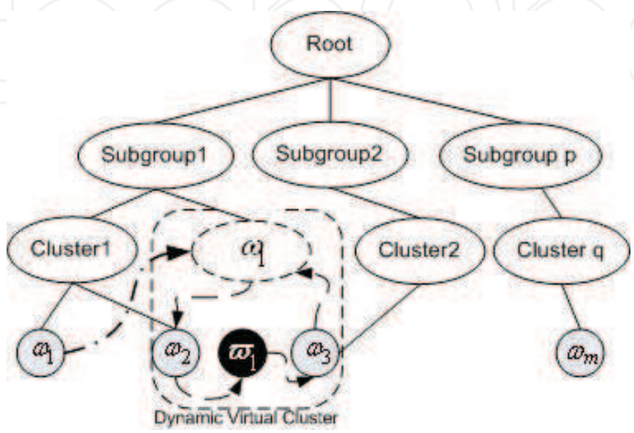

Fig. 5. User Join.

iii. All members of vc_i then start to contribute secrets and generate a virtual cluster key. The key is synchronized with a LSGC for sharing sensitive information among members based on virtual cluster key generation algorithm.

When an active user joins, a new virtual cluster is created and a virtual cluster key is contributed by all group members. The passive user (leader) has all relevant group keys and the LSGC knows the new virtual cluster key. Consequently, the rekeying operation does not take place. In other words, an active user join action does not affect whole group, and the virtual cluster leader takes responsibility for sensitive information forwarding.

Passive User Joins Cluster. When a passive user (for example, ω_m in Fig. 5.) wants to join the group, it applies the passive user level key distribution agreement. Backward secrecy must be guaranteed to prevent the new member from accessing previous group communications. The join procedure starts with passive user join request:

i. First, ω_m contacts the nearby LSGC, and the LSGC forwards the request to AAM for authentication via a secure unicast channel.
ii. After successful verification, the LSGC updates group keys for backward secrecy and unicast the new group keys for ω_m encrypted by the dynamic communication key of ω_m .

Passive User Joins Existing Virtual Cluster. If a passive user (ω_m in Fig. 5.) wants to join an existing virtual cluster, it needs to apply contributory group key management. For backward secrecy, the old virtual cluster key must be replaced with new contributed key:

i. First, ω_m contacts the nearby LSGC and the LSGC forwards the request to AAM for authentication via a secure unicast channel.
ii. After successful verification, a new virtual cluster key is generated by the leader and ω_m via the virtual cluster key generation algorithm.
iii. Once the new virtual cluster key is generated the leader broadcasts the new keys in the virtual cluster and informs the LSGC.

No matter whether the joining user is active or passive, if the user wishes to join a virtual cluster, contributory group key management is applied. Therefore, no rekeying operation occurs. To protect the privacy of active users, when a passive user wants to join an existing virtual cluster, the passive user needs access permission from the active user in the virtual cluster.

3.2.3 Member leave

Similar to the join operation, there are three scenarios for the member leave operation: an active user leaves the system, a passive user leaves the system or a passive user leaves an existing virtual cluster.

Active User Leaves. Suppose an active user (ϖ_1 in Fig. 5) wants to leave the system. It does not need forward secrecy, because virtual clusters are containers for active users. When the active user leaves, the virtual cluster is destroyed.

Passive User Leaves Cluster. If a passive user (for example, ω_m in Fig. 5) wants to leave cluster, it needs to apply a passive user level key distribution agreement. Forward secrecy must be guaranteed to prevent the leaving user from accessing future group communications. The leave operation begins with a passive user leave request:

i. First, ω_m sends a leave request to the LSGC.

ii. Upon receipt, the LSGC triggers a key update for other group members and unicasts new group keys to the involved cluster users with their dynamic communication keys.

Passive User Leaves Existing Virtual Cluster. If a passive user (for example, ω_3 in Fig. 5) wants to leave the virtual cluster, the virtual cluster will not be destroyed (which is the case should an active member leave). However, to ensure backward secrecy, the virtual cluster key needs to be updated. This action does not affect other group members.

i. First, ω_3 sends a leave request to the leader ω_1. ω_1 removes ω_3 from the member list and then updates LSGC.
ii. The LSGC then triggers the virtual cluster key generation algorithm to generate a new virtual cluster keys with existing members in the virtual cluster.

Passive users leaving several virtual clusters at the same time follow the procedure for this algorithm. However, when the passive user wants to leave the system, the procedure will apply group key tree management. Because the passive user does not "provide" sensitive information for virtual cluster members, the passive user does not have any impact on the virtual cluster. For forward secrecy, only a new virtual cluster key is required.

3.2.4 Periodic rekeying operation

The periodic rekeying operation is a process to renew group keys in the system for security purposes. It does not relate to either join or leave key operations. After a period of time, the group keys become vulnerable to key compromise and cryptanalysis attacks. This operation helps the system to reduce those risks. Because active users know virtual cluster keys rather than group keys, the periodic rekeying operation applies to passive users only.

3.3 Authentication and authorization management

Authentication and authorization are two interrelated concepts that form the security component of *user interface*. This component conducts security by co-operating with UGKM and DKM. It can be characterized as follows:

Definition 3.3 (Authentication and Authorization Management AAM) Authentication and authorization management is a quadruple $[U, EID, Proto, v(u_i, eid_j)]$, where:

i. EID is a set composed of enciphered identities for all registered users U.
ii. Proto is a set composed of protocols for authenticating the legitimacy of U and allowing U to delegate authorization in SecureSIS. (It consists of Initialization, Logon and AccessAuth, a suite of protocols).
iii. $v(u_i, eid_j)$ is a verification function that associates a Boolean value with a user $u_i \in U$ and an enciphered identity $eid_i \in EID$. Such checking defines the legitimacy of a user u_i with regard to the eid_j.

3.3.1 Initialization protocol

For every user registered in the system, the LSGC generates a unique random identity associated with the user. Separate from dynamic keys management, the unique identity generation takes place only in the LSGC. Given $aam \in AAM$ (an authentication and authorization management object) and $dkm \in DKM$ (a dynamic key management object), the protocol is described as follows:

i. A user $u_i \in U$ registers with the system, dkm generates a unique random identity id_i for the user u_i and two unique random secrets. (The two unique secrets are secretly

distributed to the user u_i for generating dynamic communication keys and dynamic data keys.)

ii. *dkm* uses the hash value of the first dynamic communication key and index i of the user to encipher the unique number as eid_i. Precisely:

$$EDI = \bigcup_{i=1}^{\mathbb{N}} \{id_i\} h(i, dk_{Y0}.u_i) \tag{7}$$

The generation of id_i can be varied depending on the security requirement. As suggested, multi-factor authentication provides stronger security for *user interface*. Therefore, we suggest that the id_i can be formed by a combination of a biometrics factor (fingerprint or DNA sequence), a possession factor (smart card) or a knowledge factor (passwords).

3.3.2 Logon protocol

Logon protocol is used as a first security shield to protect sensitive information systems. Once a user successfully verifies with a LSGC, the user is able to request and join a group. In other words, before joining a group, a user must be authenticated as a legitimate user. The protocol is depicted as follows:

i. First, a user sends a request to $aam \in AAM$, $\{logon\_request, h(i, dk_{Y(j-1)}.u_i)\} dk_{Yj}.u_i$.

ii. After understanding the received packet, *aam* uses $h(i, dk_{Y(j-1)})$ as a key K to decipher eid_i. If, and only if, the enciphered value is same as id_i, then the user is legitimate, and the user can make further requests, such as to join a group or to access sensitive information.

iii. Subsequently, *aam* sends back a challenge to verify itself to the user.

iv. When the user leaves the system, the current dynamic communication key of the user is used to generate a new key $K' = h(i, dk_{Y(j+n)}.u_i)$, and produce a new $eid_i^{'}$ to replace the old eid_i, where n is a natural number, indicating the number of messages performed by the user in the system ($eid'_i \leftarrow \{\{eid_i\} \sim K\} K'$).

3.3.3 AccessAuth protocol

The AccessAuth protocol offers an authentication and authorization mechanism for sensitive information sharing among groups and users. It enables privacy protection whereby owners can take full control of their sensitive information. The protocol also manages group-to-group, group-to-individual, individual-to-individual and individual-to-group authentication and authorization.

Before depicting the protocol, participant classification is given to clarify that participant p_m and p_n can be either a group or an individual. Formally:

Definition 3.4 (Participant Classification PC) PC is a triple, *[P,T,ς]*, where P is a set of participant objects and T is an enumeration of $\{single, group\}$, and $\varsigma : P \rightarrow T$ is the participant classification mapping.

When the classification type is $T : single$, P acts as an individual user $P \subseteq U$. When type is $T : group$, P is representative of a cluster $c_i \in C \bigcup VC$ where $P \subseteq L \bigcup VL$. In other words, P is a leader of c_i (a cluster or a virtual cluster). The protocol is described as follows:

i. p_m generates a token $h(I_n\_request, dk_{Y(j-1)}.p_m)$ and sends it together with a request (sensitive information of p_n) to the LSGC. Note that if p_m has the status of $T:group$, the p_m will be the representative (leader) of a group.
ii. After understanding the request and verifying the token, aam in the LSGC checks for permission based on the security agreement (SA) of sensitive information.
iii. After obtaining the token and query from aam, p_n can delegate permissions on each selective portion of information according to the query and generate a new token $h(I'_n\_response, dk_{Yj}.p_n)$. This token is sent back in the response message to aam to be ciphered by the next dynamic communication key.
iv. When aam receives and verifies the token from p_n, p_m is able to retrieve the sensitive data. If p_m has the status of $T:single$, the sensitive information will be unicast to p_m, otherwise, the sensitive information is multicast to the group and encrypted by the group key (either a cluster key or a virtual cluster key).

3.4 Sensitive information management

One of the most important technological challenges, that sensitive information systems facing today, is keeping sensitive content secure when it is shared among internal and external entities. In this component, dynamic keys are used to integrate with sensitive information ***I*** in order to help guard against the unauthorized disclosure of ***I***. The sensitive information is stored in a form of cipher (encrypted sensitive information, named EI), in another words, no plaintext is kept in SecureSIS. Also, each ***I*** is encrypted by a different dynamic data key, and all these dynamic data keys are encrypted by current dynamic data key (encrypted dynamic data keys, named EDK). Therefore, only the owner of sensitive information possesses the correct and latest dynamic data key. The privacy of owner thus is maintained in SecureSIS. The SIM component is formally characterized as follows:

Definition 3.5 (Sensitive Information System SIM) Sensitive information management is a quadruple $[RI, CI, EL, f(I)]$, where:

i. RI is a set composed of indices for collected critical information ***I***.
ii. CI is a union of sets of encrypted sensitive information (EI) and encrypted dynamic data keys (EDK), where, EI is produced using dynamic data keys of sensitive information owner u_n, $EI = \bigcup_{i,j\in\mathbb{N}} \{I_j\} dk_{Xi}.u_n, I_j \in I$ and, EDK is generated using current dynamic data keys of sensitive information owner to encrypt the keys used to encipher the information. It can be symbolized as: $EDK = \bigcup_{i,j\in\mathbb{N}} \{\{dk_{Xi}.u_n\} dk_{XC}.u_n, h(EI_j)\}, EI_j \in EI$. Meanwhile, $dk_{XC}.u_n$ is a current dynamic data key of u_n. It is specified in order to encrypt and decrypt the dynamic data keys (EDK). The encrypted keys are stored in the header of EI.
iii. EL stands for emergency list; a set of relationship objects O. Each $o_i \in O$ contains a user $u_i \in U$, a nominated cluster $c_n \in C$, an allocated auditing cluster $c_a \in C$ and an encrypted dynamic data key. At the cost of triggering an automatic audit, EL is used in an emergency to gain access to sensitive information ***I*** of users that would normally be inaccessible. $EL = \bigcup_{i\in\mathbb{N}^*} u_i, c_{n\mapsto i}, c_{a\mapsto i}, \{dk_{XC}.u_i\} K_{combine}$, where $K_{combine}$ is a combination key of

leaders l_n and l_a, which represent cluster $c_{n\mapsto i}$ and $c_{a\mapsto i}$ respectively, and $K_{combine} = h(h(n, dk_{Yj}.l_n), h(a, dk_{Yk}.l_a))$.

iv. *f(I)* is a symmetric cryptographic function that employs dynamic data key $dk_{Xi}.u_j$ to encipher/decipher sensitive data *I* and dynamic data keys.

3.4.1 SIM structure

Sensitive information management objects contain encrypted sensitive information and other supportive information. Each record or file of a user is enciphered with different dynamic data keys. Letting $ci_i.u_j \in CI$ be an object of CI, the structure of a SIM object is illustrated as in Fig. 6.

In regard to the architecture of SecureSIS, several administration areas form a multicast group (UGKM) and each area is managed by a LSGC associated with a subgroup. Also, RI, defined in SIM, is a set of indexes for collected sensitive information. The sensitive information of a user can therefore be stored in different SIM objects. In other words, fragmented sensitive information of a user can be transferred from different geographic locations and located by RI.

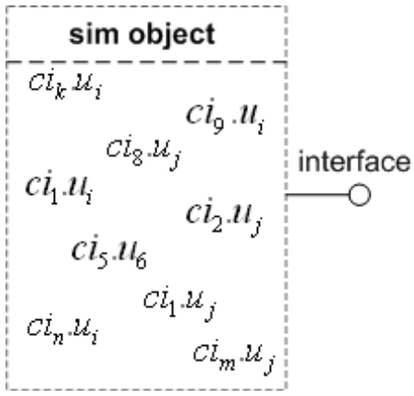

Fig. 6. Structure of a SIM Object.

3.4.2 Dynamic membership operations

When a user registers with the system, the user must agree and choose a trusted participant, either a joined cluster or a nominated cluster. The chosen participant will be added to the emergency list (EL). This confidentiality "overrides" rule allows an authenticated cluster in an emergency to gain access to sensitive information of users which would normally be inaccessible. The rule also solves the problem of information accessibility when a user permanently leaves the system. In other words, dynamic ownership of sensitive information is provided.

Meanwhile, the maintenance of the list EL is important. EL Update is an operation that updates the new nominated cluster or encrypted dynamic data keys to a relationship object $o_i \in O$. There are two events to trigger EL update. First, when a user requests a change of the nominated trust cluster, the system will allocate a new audit cluster and generate a new combination key by leaders of the new nominated cluster and the allocated audit cluster. Second, when the dynamic communication keys of the leaders are changed, the encrypted user dynamic data keys will be updated. The EL update operation ensures the list

is up-to-date in order for it to be used for authentication in emergency access situations or when the user permanently leaves.

Emergency Access. It is necessary when a user is not able to authenticate with the system and the user has authorized the nominated cluster as a trust participant. In an emergency circumstance, the user's sensitive information can be accessed via the attendant audit cluster.

Given $c_n \in C \bigcup VC$ as a nominated cluster for user $u_n \in U$ and $c_a \in C$ as an audit cluster, we have $l_n \in c_n$ and $l_a \in c_a$ as a leader of corresponding clusters. For an emergency access, the procedure is described as follows:

i. An emergency access event occurs.
ii. The leader of the nominated cluster sends a request to the system together with a token.
iii. The system looks at the EL and sends a request to the corresponding audit cluster in order to have a response and a token.
iv. After the system gathers two tokens from the nominated and audit clusters, the system will recover user u_n dynamic data key and encipher it with the dynamic communication key of l_n. The sensitive information of user u_n will then be sent to the nominated cluster c_n.

User Permanently Leaves. When a user permanently leaves the system, the user either removes selected owned sensitive information or leaves it as "orphan" information. When orphan information exists in the system, the nominated cluster takes control of the information.

The procedure is the same as in the emergency access procedure steps i-iii. The last step is to use the dynamic data key of the leader l_n to encipher the leaving user's dynamic data keys.

4. Security analysis and discussion on secureSIS

4.1 Security of DKM

Definition 3.1 demonstrates that two sets of dynamic keys are necessary to ensure security when protecting the sensitive information of users. The dynamic communication key set $\{dk_{Yj} \mid j \in \mathbb{N}\}$ protects communication channel and user interface, while the dynamic data key set $\{dk_{Xi} \mid i \in \mathbb{N}\}$ secures sensitive information storage.

Because dynamic keys possess dynamic key secrecy, former key secrecy and key collision resistance properties, a corollary can be made.

Corollary 4.1 Because SecureSIS uses two sets of dynamic keys, even if one set of dynamic keys were to be disclosed, the security of the proposed system would not be compromised.

Proof: Based on mutual information, $I(A;B) = \sum Pr(A;B) \log(\frac{Pr(A;B)}{Pr(A)P(B)})$, if $A = DK_X$ and $B = DK_Y$, then we have $I(DK_X;DK_Y) = \sum Pr(DK_X;DK_Y) \log(\frac{Pr(DK_X;DK_Y)}{Pr(DK_X)P(DK_Y)})$, and, according to key collision resistance, the probability of dynamic keys collision is negligible. In other words, generated two sets of dynamic keys with two independent unique seeds guarantee that DK_X is independent of DK_Y. Hence, according to probability theory, if, and only if A and B are independent, will $P(DK_X;DK_Y) = P(DK_X)P(DK_Y)$. If that is the case, then, we have $I(DK_X;DK_Y) = \sum Pr(DK_X;DK_Y) \log(\frac{Pr(DK_X;DK_Y)}{Pr(DK_X)P(DK_Y)}) = 0$, which is equivalent to

saying that one disclosed set of dynamic keys cannot reveal any information about another set of dynamic keys. □

Because a set of dynamic keys has no impact on another set of dynamic keys in DKM, a corollary can be claimed.

Corollary 4.2 The use of two sets of dynamic keys in SecureSIS can achieve intrusion detection and prevention.

Proof: Let A denote an adversary. By observing network traffic, A obtains a subset of used dynamic keys and a number of used tokens. According to dynamic key secrecy and former key secrecy, new dynamic keys are computationally infeasible based on obtained keys and tokens. Should A try to penetrate the system with obtained information, the action will be detected immediately, because dynamic keys can only be used once. In addition, although the actions of A compromise one set of dynamic keys, because of Corollary 4.1, the other set of dynamic keys will still be secure and unaffected. The security of the sensitive information is maintained and the proof is complete. □

4.2 Security of UGKM

Group key secrecy renders the discovery of any group key computationally infeasible for a passive adversary. In UGKM, group keys are generated by the key server (DKM) randomly in the passive user tier; this guarantees group key secrecy. However, in the active user tier, as defined, all active users belong to virtual clusters, and contributory group key management is applied to secure multicasting critical contents. The discussion in Section 3.2 on group keys gives an algorithm that generates virtual cluster keys for all involved members; a corollary can now be devised to show that UGKM also has a group key secrecy feature.

Corollary 4.3 The contributed virtual cluster key is computational infeasible.

Proof: Assume a virtual cluster $vc_n \in VC$ consists of one active user ϖ_m and n-1 passive users $vc_n \in VC, vc_n = \{\varpi_m, involved \sum \omega_i\}$. The virtual cluster key K_{vc} is formed by contributing the intermediate key $ik_i = f(dk_{yj}.u_i) \bmod p$ (the dynamic communication key) of each user $u_i \in vc_n$. Let K and IK be virtual cluster keys and intermediate key spaces respectively. Then, if an adversary obtains all intermediate keys $IK = \{ik_i \mid i \in \mathbb{N}\}$, the probability of breaching the contributed K_{vc} is:

$$Pr(K \mid IK) = Pr(K = K_{vc}; IK = ik_1) + Pr(K = K_{vc}; IK = ik_2) + \ldots + Pr(K = K_{vc}; IK = ik_n) \quad (8)$$

Thus we have, $Pr(K \mid IK) = \sum_{i=1}^{n} Pr(K = K_{vc}; IK = ik_i) = \sum_{i=1}^{n} Pr(K = K_{vc} \mid IK = ik_i) Pr(IK = ik_i)$. The contributed secret $dk_{yj}.u_i$ has all the cryptographic properties of dynamic keys and the special function $f(.)$ has the property of $\forall x, y(x \neq y), \neg \exists f(x) = f(y)$ (Definition 2.1). Therefore, the probability of generating each intermediate key $ik_i = f(dk_{yj}.u_i) \bmod p$ is $\frac{1}{p}$. In other words, the generated intermediate key is uniformly distributed over the interval $[0, p-1]$, and we have $Pr(IK = ik_i) = \frac{1}{p}$. Therefore, Eq. 8 is $\frac{1}{p} \sum_{i=1}^{n} Pr(K = f(ik_1 \ldots ik_n) \mid IK = ik_i)$.

There are n intermediate keys in vc_n, so, given an intermediate key, the probability of guessing $Pr(K = ik_1 ... ik_n \mid IK = ik_i) = \frac{1}{n}$. Thus, $Pr(K \mid IK) = \frac{1}{p}\sum_{i=1}^{n}\frac{1}{n} = \frac{1}{p}$. The contributed virtual cluster key $K = K_{vc}$ is therefore uniformly distributed over the interval $[0, p - 1]$. The contributed virtual cluster key is computationally infeasible; the proof is complete. □

Forward secrecy guarantees that knowledge of a contiguous subset of old group keys will not enable the discovery of any subsequent group keys. In other words, forward secrecy prevents users who have left the group from accessing future group communication. Forward secrecy is demonstrated in the active user tier by the member leave operation.

In the active user leave operation, each virtual cluster has only one active user and the existence of the active user determines the existence of the virtual cluster. When the active user leaves the virtual cluster, the cluster is destroyed. Operations involving active users consequently do not need forward secrecy. However, when a passive user leaves an existing virtual cluster, forward secrecy is necessary. As described in Section 3.2.3, a corollary can be made.

Corollary 4.4 Forward secrecy is guaranteed in virtual clusters.

Proof: Suppose ω_n is a former virtual cluster member. Whenever a leaving event occurs as a result of a passive user leaving an existing virtual cluster operation, a new K_{vc} is refreshed, and all keys known to leaving member ω_n will be changed accordingly. The probability of ω_n knowing the new K_{vc} is $Pr(new\ K_{vc} \mid K_{vc})$. According to Corollary 4.3, virtual cluster keys are uniformly distributed. The old K_{vc} and new K_{vc} are therefore independent and we have $Pr(new\ K_{vc}, K_{vc}) = Pr(new\ K_{vc})Pr(K_{vc})$, then $Pr(new\ K_{vc} \mid K_{vc}) = Pr(new\ K_{vc})$. Therefore, the probability of knowing the old K_{vc} and being able to use it to find the new K_{vc} is the same as finding the new K_{vc}. In other words, ω_n has the same level of information of the new virtual cluster key as an adversary. Forward secrecy is satisfied in operations involving virtual clusters; the proof is complete. □

Backward secrecy ensures that a new member who knows the current group key cannot derive any previous group key. In other words, backward secrecy prevents new joining users from accessing previous group content. Backward secrecy is achieved in the active user tier through the member join operation. In the active user join operation, when an active user joins the group, a new virtual cluster is created and consequently there are no previous virtual cluster keys to be taken into consideration; in this situation, backward secrecy is not a concern. However, when a passive user joins an existing virtual cluster operation, backward secrecy needs to be considered. As described in Section 3.2.4, a corollary can be made.

Corollary 4.5 Backward secrecy is guaranteed in virtual clusters.

Proof: Similar as Corollary 4.4.

4.3 Security of AAM

The proposed AAM manages the security of SecureSIS by adopting DKM and UGKM to protect user interface. It allows users to authenticate themselves to have fine-grain control over portions of their critical information. AAM offers secure authentication and flexible authorization for individuals and group members. AAM consists of an Initialization protocol, a Logon protocol and the AccessAuth protocol. In this section, the Logon protocol, as a representative, is examined to show the security in *user interface* protection.

In order to verify the security of each protocol, Spi calculus (Abadi, 1999; Abadi & Gordon, 1997) is used to evaluate the security of AAM. The approach is to test that a process $P(x)$ does not leak the input x if a second process Q cannot distinguish running in parallel with $P(M)$ from running in parallel with $P(N)$, for every M and N. In other words, $P(M)$ and $P(N)$ are indistinguishable for the process Q.

In order to investigate the Logon protocol, the protocol needs to be first abstracted into Spi calculus:

i. $u_i \rightarrow aam : \{logon\_req, h(i, dk_{Y(j-1)}.u_i)\}dk_{Yj}.u_i$ on $c_{ua}, c_{ua} \in C$.

ii. $aam \rightarrow dkm : \{key\_req, i\}cdk_l.aam$ on $v_{ad}, v_{ad} \in V$.

iii. $dkm \rightarrow aam : \{dk_{Yj}.u_i\}cdk_{l+1}.aam$ on $v_{da}, v_{da} \in V$.

iv. $aam \rightarrow u_i : \{logon\_req, h(logon\_req, dk_{Yj}.u_i)\}dk_{Y(j+1)}.u_i$ on $c_{au}, c_{au} \in C$.

It is assumed there are n users and each user has a public input channel (*C*). Informally, an instance of the protocol is determined by a choice of involved entities. More formally, an instance is a triple $[w,t,I]$ such that w and t are entities, such as users and SecureSIS component objects, and $\boldsymbol{I}$ is a message. Moreover, F is an abstraction representing the behaviours of any entities after receipt of the message from the protocol. Meanwhile, messages between aam and dkm occur in private communication channels (*V*) (steps ii and iii). The proof is the same as the public communication channels steps i and iv. Therefore, in this discussion, the proof of messages i and iv is given. In the Spi calculus description of the Logon protocol, given an instance (w, t, I), the following process corresponds to the role of users and the LSGC (AAM and DKM).

$$\begin{aligned} Send_{w,t} \triangleq \overrightarrow{c_{wt}}\left\langle \{logon\_req, h(w, dk_{Y(j-1)}.u_w)\}dk_{Yj}.u_w \right\rangle | \, c_{tw}(x_{cipher}).case\ x_{cipher}\ of \\ \{x, H(y_p)\}dk_{Y(j+1)}.u_w\ in\ let\ (x, y_{nonce}) = y_p\ in\ [x\ is\ logon\_req][y_{nonce}\ is\ dk_{Yj}.u_w]\ in\ F \end{aligned} \tag{9}$$

The process $Send_{w,t}$ describes one entity (users) processing an output message i) in parallel with an input message iv). It is a process parameterised by entities w and t. Formally, we view $Send_{w,t}$ as a function that map entities w and t to processes, called abstractions, and treat w and t on the left of $\triangleq$ as bound parameters. For the process $Recv_t$, it describes one entity (LSGC) processing an input message iv) in parallel with an output message i).

$$\begin{aligned} Recv_t \triangleq c_{wt}(y_{cipher}).case\ y_{cipher}\ of\ \{x, H(y_p^1)\}dk_{Yj}.u_w\ in\ let(x, y_{nonce}^1) = y_p^1 \\ in\ [x\ is\ w][y_{nonce}^1\ is\ dk_{Y(j-1)}.u_w] | \, \overrightarrow{c_{tw}}\left\langle \{logon\_req, h(logon\_req, dk_{Yj}.u_w)\}dk_{Y(j+1)}.u_w \right\rangle \end{aligned} \tag{10}$$

The processes $Sys(I_1 ... I_m)$ describes the whole protocol (message i and iv) with m instances. The channels c_{wt} and c_{tw} are public channels. The processes send a logon request under the dynamic communication key $dk_{Yj}.u_w$ and receive LSGC challenge information under the dynamic communication key $dk_{Y(j+1)}.u_w$. Besides, $(vdk_{Yj}.u_w)$ and $(vdk_{Y(j+1)}.u_w)$ achieve the effect that only entity w and t have the dynamic communication keys. Let $\bigcup_{x \in 1..m} P_x$ be $m-way$ composition $P_1 | ... | P_m$, and $\underline{(vdk_{Yj}.u_{wx})}\underline{(vdk_{Y(j+1)}.u_{wx})}$ stand for $(vdk_{Yj}.u_{w1})...(vdk_{Yj}.u_{wm})(vdk_{Y(j+1)}.u_{w1})...(vdk_{Y(j+1)}.u_{wm})$ we have:

$$Sys(I_1...I_m) \triangleq (c_{wt})(c_{tw})(\underline{vdk_{Yj}.u_{wx}})(\underline{vdk_{Y(j+1)}.u_{wx}})\{\bigcup_{x\in 1..m}(Send_{wx,tx}\,|\,!Recv_{tx})\} \quad (11)$$

The replication of the receiving processes $\bigcup_{x\in 1..m}!Recv_{tx}$ means that every entity is ready to play the role of receiver in any number of runs of the protocol in parallel. Therefore, the protocol can be simultaneous, even though same entity may be involved in many instances. We now examine one instance of the protocol. Let $\equiv$ be structural equivalence by combining Eq. 9 and 10, we have Eq. 11 rewritten as:

$$\begin{aligned} Sys \equiv\ & (vdk_{Yj}.u_w)(vdk_{Y(j+1)}.u_w)c_{wt}(y_{cipher}).case\ y_{cipher}\ of\ \{x,H(y_p^1)\}dk_{Yj}.u_w\ in\ let(x,y_{nonce}^1)=y_p^1 \\ & in\ (x\ is\ w)(y_{nonce}^1\ is\ dk_{Y(j-1)}.u_w)\,|\,\overrightarrow{c_{wt}}\left\langle\{logon\_req,h(w,dk_{Y(j-1)}.u_w)\}dk_{Yj}.u_w\right\rangle| \\ & c_{tw}(x_{cipher}).case\ x_{cipher}\ of\ \{x,H(y_p)\}dk_{Y(j+1)}.u_w\ in\ let\ (x,y_{nonce})=y_p \\ & in\ (x\ is\ logon\_req)(y_{nonce}\ is\ dk_{Yj}.u_w)\ in\ F\,|\,\overrightarrow{c_{tw}}\left\langle\{logon\_req,h(logon\_req,dk_{Yj}.u_w)\}dk_{Y(j+1)}.u_w\right\rangle \end{aligned} \quad (12)$$

Based on the reaction relation and reduction relation rules,

$$\begin{aligned} Sys &\mapsto (vdk_{Yj}.u_w)(vdk_{Y(j+1)}.u_w)F(logon\_req,h(logon\_req,dk_{Yj}.u_w),h(w,dk_{Y(j-1)}.u_w)) \\ &\mapsto F(logon\_req,h(logon\_req,dk_{Yj}.u_w),h(w,dk_{Y(j-1)}.u_w)) \end{aligned} \quad (13)$$

The processes have not revealed the information of *logon_req* and tokens. In the Logon protocol, the tokens are generated with the dynamic communication keys of users. According to the cryptographic properties of dynamic keys, the dynamic communication keys of users are equivalent to random numbers as well as the tokens. Consequently, a specification is given by revising the protocol. After applying reaction relation and reduction relation rules, we have $Sys_{spec} \mapsto F(logon\_req,random,random)$. This is equivalent to *Sys* (noted as $Sys(I_1...I_m) \simeq Sys(I_1...I_m)_{spec}$). In other words, $Sys(I_1...I_m)$ and $Sys(I_1...I_m)_{spec}$ are indistinguishable to an adversary. Thus this protocol has two important properties as proved:

- Authenticity: entity B always applies F to the message that entity A sends, and an adversary cannot cause entity B to apply F to other messages. In other words, $Sys(I_1...I_m) \simeq Sys(I_1...I_m)_{spec}$ for any message.
- Secrecy: The message cannot be read in transit from entity A to entity B, if, and only if F does not reveal the message, then the whole protocol does not reveal the message.

4.4 Security of SIM

The security of SIM is conducted by two sets of dynamic keys. The first set of dynamic keys (dynamic communication keys) is a security shield that is used to protect *communication channel* and *user interface*. The second set of dynamic keys (dynamic data keys) is the security core of SIM. This set only protects *sensitive information storage* and integrates with sensitive information stored in cipher form; it is never involved in the protection of *communication channel* and *user interface*.

According to Section 3.4, SIM offers the following security features:

- Every data entry operation yields different EI.
- Every transaction triggers EDK updates.
- Any data altered results in a new EI and a new set of EDK.

- Only the owner of sensitive data has the correct dynamic key to decipher the data.
- Only in an emergency circumstance is a nominated cluster, overseen by an auditing cluster, able to access the sensitive information of users.
- Any "orphan" sensitive information is managed by a nominated cluster overseen by an auditing cluster.

Intuitively, because the above facts protect sensitive information in storage, it would appear that sensitive information is secure and protected, even should the storage be breached. Therefore, a corollary can be made.

Corollary 4.6 Even if the security of one user is breached in SIM, the security of other users and sensitive information will not be compromised.

Proof: Suppose that S is a sample space possessing enciphered sensitive information. Events $B_1, B_2, ...B_n$ partition S, and we have $B_1 \bigcup B_2 \bigcup ... \bigcup B_n = S$. Due to SIM security features, the occurrence of events B_i and B_j are independent. Therefore, $B_i B_j = \varnothing$ for any pair i and j. Let B_j denote the event that disclosed information comes from user u_j and $Pr(B_i) > 0, i \in \mathbb{N}$. Let A denote the event that the sensitive information is compromised. According to the conditional probability of compromised information B_j given event A is one, $Pr(B_j \mid A) = 1$. Apply Bayes' law, we have:

$$Pr(B_j \mid A) = \frac{Pr(B_j)Pr(A \mid B_j)}{\sum_{i=1}^{n} Pr(B_i)Pr(A \mid B_i)} \tag{14}$$

and thus,

$$\begin{aligned} Pr(B_j)Pr(A \mid B_j) &= \sum_{i=1}^{n} Pr(B_i)Pr(A \mid B_i) \\ &= Pr(B_1)Pr(A \mid B_1) + Pr(B_2)Pr(A \mid B_2) + ...+ \\ &\quad Pr(B_j)Pr(A \mid B_j) + Pr(B_{j+1})Pr(A \mid B_{j+1}) + ...+ \\ &\quad Pr(B_{n-1})Pr(A \mid B_{n-1}) + Pr(B_n)Pr(A \mid B_n) \end{aligned} \tag{15}$$

and then,

$$Pr(B_1)Pr(A \mid B_1) + ... + Pr(B_{j-1})Pr(A \mid B_{j-1}) + Pr(B_{j+1})Pr(A \mid B_{j+1}) + ... + Pr(B_n)Pr(A \mid B_n) = 0 \tag{16}$$

Since, $\forall Pr(B_i) > 0$, then conditional probability of compromising sensitive information of others is zero. We have $Pr(A \mid B_1) + ... + Pr(A \mid B_{j-1}) + Pr(A \mid B_{j+1}) + ... + Pr(A \mid B_n) = 0$. Therefore, even when one user is compromised in SIM, the probability of breaching other sensitive information is zero; the proof is complete. □

4.5 SecureSIS goals discussion

Based on the proofs of Theorems 2.1-2.4 and Corollaries 4.1-4.6, the proposed security architecture satisfies the security requirements. By using the theorems and corollaries already presented, in this section, we prove that SecureSIS also meets its intended security goals.

Proof of User Interface's Goal. *User interface* is protected by a combination of AAM, DKM and UGKM. According to the discussion on AAM, any user $\forall u_i \in U$ can prove u_i to

SecureSIS by adopting dynamic communication keys securely. Also, for any sensitive information $\forall I_i \in I$, if the user u_i provides proof to SecureSIS with full permission to I_i, then the user u_i possesses the information. In addition, if user u_i possesses the information, then u_i has full control of it. Moreover, the Logon protocol in AAM, which guarantees that, as long as there is a correlated token (signature), SecureSIS will believe that the action is performed by user u_i. Furthermore, as was discussed in the Logon protocol on AAM a challenge-response message is returned by using the dynamic communication key of user u_i to generate a token in order to verify the genuineness of SecureSIS. According to the cryptographic properties of dynamic keys and the security of AAM, sensitive information is only disclosed to legitimate users with proper permissions and genuine SecureSIS. □

Proof of Communication Channel's Goal. The security of *communication channel* is managed by the use of dynamic communication keys (DKM) and group keys (UGKM). As discussed in Section 3.3.3, it ensures that $\forall u_i \in U$ believes received sensitive information is identically maintained in transit. Using the AccessAuth protocol, every message among entities is assembled with a unique token. Because of the features of DKM and UGKM, the keys needed to protect communication are secure. Every message received by SecureSIS can then be verified. Consequently, we have that sensitive information is identically maintained during transmission via open networks in SecureSIS. □

Proof of Sensitive Information Storage's Goal. The security of *sensitive information storage* is attained by SIM participating with DKM and UGKM. if $\forall u_i \in U$ possesses the information, the user has full control of it. In other words, the user can decipher EI_j. Hence u_i believes possessed sensitive information is genuine in sensitive information storage, and ensures that sensitive information is stored securely and only privileged users can understand and retrieve sensitive information in SecureSIS. □

5. Conclusion and future work

Protecting sensitive information is a growing concern around the globe. Securing critical data in all sectors, including the business, healthcare and military sectors, has become the first priority of sensitive information management. Failing to protect this asset results in high costs and, more importantly, can also result in lost customers and investor confidence and even threaten national security. The purpose of this research was to develop a security architecture able to protect sensitive information systems.

Sensitive information systems consist of three components: *communication channel*, *user interface* and *sensitive information storage*; the protection of these three components equates to the protection of sensitive information itself. Therefore, this research contributes to the development of the body of knowledge surrounding sensitive information protection. Its contributions include the following:

- Formal definition and cryptographic properties proofs of dynamic keys.
- A new proposed security architecture for sensitive information systems.

This research has opened up avenues for further work. These include i) investigation into the use of dynamic keys for intrusion prevention and detection; and ii) the design and development of new dynamic key algorithms.

This research has presented a security architecture that overcomes the limitations of existing security approaches in protecting sensitive information. The architecture has also

demonstrated the feature of intrusion prevention and detection by the employment of two sets of dynamic keys. This mechanism has yet to be studied formally and systematically. It could be further investigated and proposed as a new component for SecureSIS.

Another direction for future research could involve the design of new cryptographic algorithms in order to enhance the security of sensitive information systems. This current research has enabled the formal definition of dynamic keys and regulated the cryptographic properties of dynamic keys. Future work might involve the testing of these definitions to further demonstrate their appropriateness when guiding the design of new dynamic key generation algorithms.

6. References

Abadi, M. (1999). Secrecy by typing in security protocols. *Journal of the ACM, Vol: 46*, No. 5, pp, 749 - 786. ISSN: 0004-5411

Abadi, M, & Gordon, AD. (1997). A calculus for cryptographic protocols: the spi calculus. *Proceedings of* the *4th ACM conference on Computer and Communications Security*, pp.36-47, ISBN: 0-89791-912-2, Zurich, Switzerland, 1997, ACM, New York

Anderson, RJ. (2001). *Security Engineering: A Guide to Building Dependable Distributed Systems*, John Wiley & Sons, ISBN: 978-0-470-06852-6, New York

Atkinson, R. (1995). *Security Architecture for the Internet Protocol* (No. RFC 1825): Network Working Group, The Internet Engineering Task Force.

Bacon, CJ, & Fitzgerald, B. (2001). A systemic framework for the field of information systems. *ACM SIGMIS Database Vol: 32*, No. 2, pp, 46 - 67. ISSN: 0095-0033

Bard, GV. (2004). *The vulnerability of ssl to chosen-plaintext attack* (No. 2004/111): Cryptology ePrint Archive.

Beimel, A, Ishai, Y, Kushilevitz, E, & Malkin, T. (1999). One-way Functions are Essential for Single-Server Private Information Retrieval. *Proceedings of* the *31st Annual ACM Symposium on Theory of Computing*, pp.89-98, ISBN: 1-58113-067-8, Atlanta, Georgia, USA, 1999, ACM, New York

Bhatia, SK, & Deogun, JS. (1998). Conceptual clustering in information retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Vol: 28*, No. 3, pp, 427-436. ISSN: 1083-4419

Boyd, G. (2007). IBM Encryption Facility for z/OS. Retrieved 28 April, 2008, from ftp://ftp.software.ibm.com/common/ssi/rep_sp/n/ZSD01450USEN/ZSD01450USEN.pdf

Cachin, C, Micali, S, & Stadler, M. (1999). Computationally Private Information Retrieval with Polylogarithmic Communication, In: *Advances in Cryptology*, J Stern (Ed.), pp. 402-414, Springer Berlin / Heidelberg, ISBN: 978-3-540-65889-4, London, UK

Cervesato, I, Jaggard, AD, Scedrov, A, Tsay, J-K, Christopher, & Walstad. (2008). Breaking and fixing public-key Kerberos. *Information and Computation Vol: 206*, No. 2-4, pp, 402-424. ISSN: 0890-5401

Challal, Y, & Seba, H. (2005). Group Key Management Protocols: A Novel Taxonomy. *International Journal of Information Technology Vol: 2*, No. 1, pp, 105-118. ISSN: 1305-2403

Davis, GB, & Olson, MH. (1985). *Management Information Systems: Conceptual Foundations, Structure, and Development*, Mcgraw-Hill, ISBN: 0070158282, New York

Dierks, T, & Rescorla, E. (2008). *The Transport Layer Security (TLS) Protocol (V1.2)* (No. RFC 5246): Network Working Group, The Internet Engineering Task Force.

Erdem, OM. (2003). High-speed ECC based Kerberos Authentication Protocol for Wireless Applications. *Proceedings of* the *IEEE Global Telecommunications Conference*, pp.1440-1444, ISBN: 0-7803-7974-8, 2003, IEEE, Danvers, MA

Freier, AO, Karlton, P, & Kocher, PC. (1996). *The SSL Protocol (V3.0)*: Transport Layer Security Working Group.

Gertner, Y, Goldwasser, S, & Malkin, T. (1998). A Random Server Model for Private Information Retrieval, In: *Randomization and Approximation Techniques in Computer Science*, M Luby, JDP Rolim & MJ Serna (Ed.), pp. 200-217, Springer Berlin / Heidelberg, ISBN:3-540-65142-X, London, UK

Gertner, Y, Ishai, Y, Kushilevitz, E, & Malkin, T. (2000). Protecting data privacy in private information retrieval schemes. *Journal of Computer and System Sciences Vol: 60*, No. 3, pp, 592-629. ISSN: 0022-0000

Gordon, SR, & Gordon, JR. (1996). *Information Systems: A Management Approach*, The Dryden Press, Harcourt Brace College Publishers, ISBN: 9780471273189, Orlando, Florida

Gray, RM. (1990). *Entropy and Information Theory*, Spinger-Verlag, ISBN: 0-387-97371-0, New York

Harbitter, A, & Menascé, DA. (2001). The performance of public key-enabled kerberos authentication in mobile computing applications. *Proceedings of* the *8th ACM conference on Computer and Communications Security*, pp.78-85, ISBN: 1-58113-385-5, Philadelphia, PA, USA, 2001, ACM, New York

Harney, H, & Harder, E. (1999). *Logical Key Hierarchy Protocol*: Network Working Group, The Internet Engineering Task Force.

Hong, W-S, Chen, S-J, Wang, L-H, & Chen, S-M. (2007). A new approach for fuzzy information retrieval based on weighted power-mean averaging operators. *Computers & Mathematics with Applications Vol: 53*, No. 12, pp, 1800-1819. ISSN: 0898-1221

Kim, Y, Perrig, A, & Tsudik, G. (2004). Tree-based group key agreement. *ACM Transactions on Information and System Security, Vol: 7*, No. 1, pp, 60 - 96. ISSN: 1094-9224

Kohl, JT, Neuman, BC, & T'so, TY. (1994). The Evolution of the Kerberos Authentication System. *Proceedings of* the *Distributed Open Systems*, pp.78-94, ISBN: 0-8186-4292-0, 1994, IEEE Computer Society Press

Kungpisdan, S, Le, PD, & Srinivasan, B. (2005). A Limited-Used Key Generation Scheme for Internet Transactions. *Lecture Notes in Computer Science, Vol: 3325*, No., pp, 302-316. ISSN: 0302-9743

Lehtovirta, V, Naslund, M, & Norman, K. (2007). *Integrity Transform Carrying Roll-Over Counter for the Secure Real-time Transport Protocol (SRTP)* (No. RFC 4771): Network Working Group, The Internet Engineering Task Force.

Li, Y, & Zhang, X. (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*, pp.40-47, ISBN: 0-7695-2095-2, Washington, DC, USA, 2004, IEEE Computer Society

Menezes, A, Oorschot, PCV, & Vanstone, SA. (1996). *Handbook of Applied Cryptography*, CRC Press, ISBN: 9780849385230, California

Meyers, RA. (2002). *Encyclopedia of Physical Science and Technology*, Academic Press, ISBN: 9780122274107, Michigan

Micheli, AD, Brunessaux, S, Lakshmeshwar, S, Bosselaers, A, & Parkinson, D. (2002). *Investigations about SSL*: MATRA Systèmes & Information, NOKIA Research Centre, K.U.Leuven Research & Development and British Telecommunications.

Mittra, S. (1997). Iolus: A framework for scalable secure multicasting. *Proceedings of* the *ACM SIGCOMM Computer Communication Review*, pp.277-288, ISBN: 0146-4833, New York, 1997, ACM, New York

Ngo, HH, Wu, XP, Le, PD, & Wilson, C. (2010). Dynamic Key Cryptography and Applications. *Journal of Information System Security, Vol: 10*, No. 3, pp, 161-174 ISSN: 1816-3548

Ornaghi, A, & Valleri, M. (2003). Man in the middle attacks Las Vegas, NV,USA: Black Hat.

Perlman, R, & Kaufman, C. (2001). Analysis of the IPSec Key Exchange Standard. *Proceedings of* the *10th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2001*, pp.150-156, ISBN: 0-7695-1269-0, Cambridge, MA, USA, 2001, IEEE

Scheaffer, RL. (1994). *Introduction to Probability and Its Applications*, Wadsworth Publishing Company, Duxbury Press, ISBN: 0-534-23790-8, Washington

Sherman, AT, & McGrew, DA. (2003). Key Establishment in Large Dynamic Groups Using One-Way Function Trees. *IEEE Transactions on Software Engineering, Vol: 29*, No. 5, pp, 444-458. ISSN: 0098-5589

Sirbu, M, & Chuang, J. (1997). Distributed authentication in Kerberos using public key cryptography. *Proceedings of* the *Network and Distributed System Security*, pp.134-141, ISBN: 0-8186-7767-8, San Diego, CA, USA, 1997, IEEE Computer Society Washington, DC, USA

Steiner, J, Neuman, C, & Schiller, JI. (1988). Kerberos: An Authentication Service for Open Network Systems. *Proceedings of* the *Winter 1988 Usenix Conference*, pp.191-200, Dallas, Texas, 1988,

Talbot, J, & Welsh, D. (2006). *Complexity and Cryptography-An Introduction*, Cambridge Univeristy Press, ISBN: 9780521852319, New York

Tienari, M, & Khakhar, D. (1992). *Information network and data communication*, Amsterdam, Elsevier Science Pub. Co., ISBN: 9780444702142, Espoo, Finland

Wagner, D, & Schneier, B. (1996). Analysis of the SSL 3.0 protocol. *Proceedings of* the *2nd USENIX Workshop on Electronic Commerce*, pp.4-17, Oakland, California, 1996, USENIX Association

Wu, XP, Le, PD, & Srinivasan, B. (2008a). Dynamic Keys Based Sensitive Information System. *Proceedings of* the *9th International Conference for Young Computer Scientists*, pp.1895-1901, ISBN: 978-0-7695-3398-8, Zhang Jia Jie, China, 2008a, IEEE Computer Society

Wu, XP, Ngo, HH, Le, PD, & Srinivasan, B. (2008b). A Novel Group Key Management Scheme for Privacy Protection Sensitive Information Systems. *Proceedings of* the

International Conference on Security and Management, pp.93-99, ISBN: 1-60132-085-X, Las Vegas, Vevada,USA, 2008b, CSREA Press

Wu, XP, Ngo, HH, Le, PD, & Srinivasan, B. (2009). Novel Authentication & Authorization Management for Sensitive Information Privacy Protection Using Dynamic Key Based Group Key Management. *International Journal of Computer Science & Applications, Vol: 6*, No. 3, pp, 57-74. ISSN: 0972-9038

Convergence and Hybrid Information Technologies

Edited by Marius Crisan

ISBN 978-953-307-068-1
Hard cover, 426 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

Starting a journey on the new path of converging information technologies is the aim of the present book. Extended on 27 chapters, the book provides the reader with some leading-edge research results regarding algorithms and information models, software frameworks, multimedia, information security, communication networks, and applications. Information technologies are only at the dawn of a massive transformation and adaptation to the complex demands of the new upcoming information society. It is not possible to achieve a thorough view of the field in one book. Nonetheless, the editor hopes that the book can at least offer the first step into the convergence domain of information technologies, and the reader will find it instructive and stimulating.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen