

ERRATA

- page 16, line 2: “state-of-the-art” for “state of the art”
 page 24, line 15: “Breaching the AES system...” for “Beaching the AES system ...”
 page 26, line 7: “In the early days of...” for “In the early day of...”
 page 62, line 3: “...is a drawback” for “...is drawback...”
 page 87, line 3: “After the literature on sensitive information storage protection has been reviewed” for “Having reviewed the literature on sensitive information storage protection”
 page 122, line 7: “3.2.7. Structure of SecureSIS” for “3.2.7. Structure in SecureSIS”
 page 181, line 1: “Other” for “other”
 Page 214, line 19: “5.5. SecureSIS Pentad Assessment” for “5.5. SecureSIS Panted Assessment”

ADDENDUM

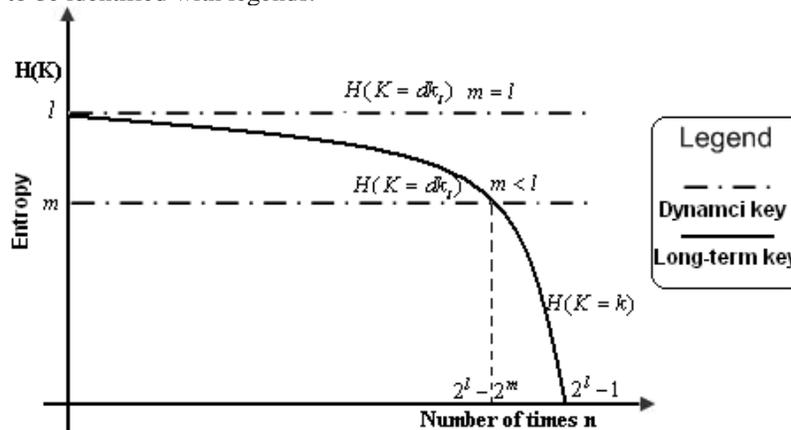
- 1.1. “In Section 3.1, the formalization of dynamic keys and the theorems used to prove the cryptographic properties appear to be a little bit straightforward, is it possible to improve the elaboration in this part by giving more evidences?”

The research described in this thesis is based on dynamic key theory to present a novel security architecture for sensitive information systems (SecureSIS). The architecture integrates dynamic keys into the *communication channel*, *user interface* and *sensitive information storage* to protect sensitive information. Therefore, the focus of the research is on how to protect sensitive information by the employment of dynamic keys based on their cryptographic properties. Although a number of dynamic key generation schemes are proposed in the literature [Ku05, KuLeSr05, LiZh04, NgWuLe10, RuWr02], for the first time in this thesis in Sections 2.1 and 3.1, formal characteristics are provided.

- 1.2. “The dynamic keys have many advantages over conventional symmetric keys and asymmetric keys and the thesis has provided qualified comparison. Is it feasible to conduct some quantified comparison between them? After all, disadvantages of them also exist and how to differentiate the application scenarios which are suitable for them respectively is meaningful.”

The comparison of conventional symmetric keys with dynamic keys is discussed in Section 2.1.1, and Table 2.1 which provides a comparison of these techniques with respect to lifetime, distribution, synchronization, storage and security properties. Also the comparison of the security properties are reiterated in Section 3.1.1 (Dynamic Keys versus Symmetric Cryptography) and in Section 3.1.2 (Dynamic Keys versus Asymmetric Cryptography). As the focus of this research is to provide an architecture solution for sensitive information protection, no specific “application (implementation) scenarios” have been discussed in the thesis.

- 1.3. “In the figure 3.1, the curve of entropy of dynamic keys and the curve of entropy of long term keys might need to be identified with legends.”



- 1.4. “In Section 5.1, concerning the proof part about the two sets of dynamic keys in DKM achieving intrusion detection and prevention, what if when two sets keys are both compromised although the chance is slim? How to theoretically prove the design of two sets is optimal, or not?”

The proposed SecureSIS employs two sets of dynamic keys to protect three components, *communication channel*, *user interface* and *sensitive information storage*, in the process of sensitive information retrieval. The reason for the employment of two sets of dynamic keys is that dynamic data keys are only used to integrate sensitive information itself (encryption), and dynamic communication keys are used only for token generation and communication protection. The two sets of dynamic keys are independent of each other. Also Since single-use nature and their cryptographic properties of dynamic keys, the probability of compromising dynamic keys is very small – see Theorem 3.1 on the Dynamic Key Secrecy. Hence, the probability of compromising two sets of independent dynamic keys is smaller than the individual ones (multiplicative law of probability).

It cannot be proved that the employment of two sets of dynamic keys is optimal. In fact, the greater the number of sets of dynamic keys, the less the probability of compromise. However, multiple sets of dynamic keys will increase the cost of key generation, management and storage overheads. On the other hand, a single set of dynamic keys cannot achieve intrusion detection. Hence, we need at least two sets of dynamic keys.

1.5. “In the comparison of group key management, strength of UGKM comes from the dynamic key and user categorization. How about the cost of management of join and leave? Is it feasible to make a comparison between those approaches too?”

The comparison of proposed work (UGKM) to scalable multi-subgroup key management (SMK) [WaLe05] and logical key hierarchy (LKH) [WaHaAg97, WoGoLa98] is summarized in the table. The entries indicate the number of messages, cryptographic operations and keys required, and the following notations are used:

- P,A,L : Passive users, Active users and Leaders, respectively.
- n : the number of members in the subgroup.
- m : the size of the cluster.
- x/y : ‘x’ denotes the cost of the requesting users to join or leave, and ‘y’ denotes the cost of other users.
- d : height of the tree, for the balanced tree, $d = \log_a n$, n is the number of members.
- a : the degree of the tree.
- n_s : the number of subgroups.
- H,D,E : one-way Hash, Decryption and Encryption functions, respectively.

Table. The Comparison of Complexities [WaLe05, WuNgLe08b].

		Communication				Computation				Storage	
		User Join		User Leave		User Join		User Leave		User	
		P	A	P	A	P	A	P	A	Single	Multi
SMK	AKC	O(1)		O(3)		Zero		$n_s E$		N/A	
	BS			$O(m-1+n/m+n_s)$		N/A		N/A		N/A	
	User	N/A		N/A		1D/3H		3D		4	j+3
LKH	AKC/ KDC	$O(d+n_s)$		$O((a-1)d+n_s)$		$2(d-1)E+(n_s-1)E$		$a(d-1)E+(n_s-1)E$		$(an-1)/(a-1)$	
	User					$(d-1)D/\frac{d}{d-1}D$		$\frac{d}{d-1}D$		$(\log_a n) + 1$	
UGKM	AKC	O(1)	0	O(3)	0	0	0	$n_s E$	0	N/A	
	BS	O(1)	0	$O(m-1+n/m+n_s)$	0	N/A		N/A		N/A	
	User	P	N/A	N/A	1D/3H	1D/1H	1D	1H	2	j+1	
	A	1D/1H				0	1H				
	L	1E/1H				1E	1H				

1.6. “SecureSIS pentad model is well presented to provide formal evaluation over the security architecture and component approaches, how about other formalized models to do such evaluation , is it feasible to make a comparison between these formalized models?”

The formalized extant security models - CIA Triad and Parkerian Hexad - have been discussed in Section 2.5. However, the lack of defined assessment properties is a major drawback. It is improper to use an existing model to assess sensitive information systems. Thus, the SecureSIS pentad is proposed, and it is the only model available in the literature for the assessment of sensitive information systems. Also, the SecureSIS pentad is a totally different architecture from the CIA Triad and Parkerian Hexad. Therefore, it is not meaningful to compare these with the SecureSIS pentad.

- 2.1. "Although a full implementation of the architecture will be beyond the scope of the current thesis, a general discussion on the how or what are the main implementation requirements to efficiently implement the SecureSIS architecture will add value to the thesis."

This research developed a novel security architecture for sensitive information systems which consists of dynamic key management (DKM), user-oriented group key management (UGKM), authentication and authorization management (AAM) and sensitive information management (SIM) in the collection of guarantees to provide security for sensitive information systems. Although a full implementation of SecureSIS is beyond the scope of this thesis, a brief description of its possible implementation will help significantly to put the architecture into effect.

Firstly, DKM manages dynamic keys of all involving entities in SecureSIS. The implementation can be carried out by the employment of a good hash function and a unique pre-shared secret to produce the first dynamic key. Following, the pre-shared key and the previous dynamic key are used to produce the next dynamic key. A good random number generator is essential to select a unique pre-shared secret for every user. DKM also needs to synchronize a dynamic key sequence with entities, such as network failures. The recovery procedure can then be followed [NgWuLe09a]. Moreover, the dynamic key agreement (Section 4.1.1) must be implemented fully. As well, DKM requires extra computational power for which all involved entities must be able to calculate dynamic keys.

Secondly, UGKM controls group members by categorizing them into active and passive users, thus, the implementation should follow the hybrid group key structure (Section 4.2.1). Conventional group key agreements [HaMu97b, KiPeTs01, KiPeTs04, WaLe05] can be applied for passive user level and contributory group key agreements [KoOh87, StStDi88, BeWi98, StTsWa96, StTsWa00] can be applied at the active user level. The execution can be done by defining a linked data structure (a doubly-linked list) to store a key node and member information. The key node can be generated by a hash function of member information in order to quickly locate members (tree traversal) and key updates.

Thirdly, AAM should employ multi-factor authentication as suggested in Section 4.3.2. The implementation of dynamic key generation (DKM) is essential as a knowledge factor. Other factors, such as smart card, hard token and biometrics, can be combined with dynamic keys to enhance the security of the AAM component.

Finally, the implementation of SIM can be done in a number of ways, since critical data can be presented in several formats, such as flat files and databases. However, a cryptographic mechanism must be implemented to encrypt the data by using dynamic data keys before it is written to storage. This procedure creates a performance bottleneck. The enhancement of SIM performance can be achieved via the employment of fast symmetric-key algorithms.

- 2.2. "Many definitions, theorems and corollaries are based on a simplistic assumption that the security attacks are random/ probabilistic events. That is not true in general. Corollary 5.1, Corollary 5.2 and Corollary 5.6 rely on these assumptions. A statement can clear the potential confusion."

Corollary 5.1, Corollary 5.2 and Corollary 5.6 are based on the assumption that every cryptographic key in SecureSIS can be attacked by the brute force technique. The probability that a key can be compromised by brute force is small due to the large key space. This is the general assumption in any cryptographic analysis.

- 2.3. "At number of places low probability of occurrence of an event is expressed by using the wrong phrase "impossible". Impossible means that the probability is exactly equals to 0. For example, see the statement of the Theorem 3.2. The proof is also an example of unnecessary trivial mathematics being included in the thesis (page 104). The student needs to identify and list those instances where this phrase (impossible) is used and mention an appropriate new phrase."

The phrase "impossible" does not appear in the thesis, but the term "infeasible" appears in Theorem 3.1 on page 101, Theorem 3.2 proof on page 102 and Corollary 5.2 proof on page 188. The probability

of computationally discovering a dynamic key is very small due to its cryptographic properties (see Section 3.1.1).

- 2.4. “A major issue that the student needs to address is to provide references to the results listed as theorems that are not candidates own work. I am not saying that the candidate has claimed the result as his, but that is a potential confusion. If the result is already published by another person the theorem should be stated but the proof can be omitted. The reader should be given reference to the publication. There are many such theorems- it is difficult for me to list all such incidences as a result of confusion about the “inventor”. Is it the candidate’s work or is he quoting a known result. I just list one such example; corollary 3.1 is an obvious result known for a long time but carries no reference or a statement that this is a well-known result. I suggest that the student provide a cross reference list for those axioms and theorems that have been proved by other researchers in the addendum of the thesis.”

All theorems and corollaries are listed in the following table to indicate that the research work was done either by the author (New) or by others (cross referenced CR) in order to eliminate any potential confusion.

Page		New / CR
101	Theorem 3.1 (Dynamic Key Secrecy) It guarantees that it is computationally infeasible for an adversary to discover any dynamic key.	New
102	Theorem 3.2 (Former Key Secrecy) It ensures that an adversary, who knows a contiguous subset of used dynamic keys, cannot discover any subsequent dynamic keys.	[KiPeTs04] [StTsWa00]
103	Theorem 3.3 (Key Collision Resistance) It means that given a dynamic key generation algorithm, and two initial seeds, S_x and S_y ($S_x \neq S_y$), the probability of key collision is negligible.	New
103	Theorem 3.4 (Key Consistency) It guarantees to produce sequential, consistent, dynamic keys, if given the same $f(.)$ and an initial seed.	New
106	Theorem 3.5 Dynamic keys are more secure than long-term shared keys to protect communication or sensitive information.	New
107	Corollary 3.1 If, and only if, $F(X)$ is a function of X , the entropy of $H(F(X) X)$ is zero.	[BrFo05]
109	Theorem 3.6 Asymmetric cryptography in protecting sensitive information is insecure.	New
187	Corollary 5.1 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.	New
188	Corollary 5.2 The use of two sets of dynamic keys in SecureSIS can achieve intrusion detection and prevention.	New
190	Corollary 5.3 The contributed virtual cluster key is computational infeasible.	New
193	Corollary 5.4 Forward secrecy is guaranteed in virtual clusters.	[StTsWa00]
194	Corollary 5.5 Backward secrecy is guaranteed in virtual clusters.	[StTsWa00]
195	Corollary 5.6 UGKM achieves collusion resistance.	New
209	Corollary 5.7 (Weak Security) Sensitive information interchange is secure in SIM.	New
210	Corollary 5.8 (Strong Security) Even though communication channel is breached, Sensitive information interchange is still secure in SIM.	New
211	Corollary 5.9 The breach of sensitive information storage does not threaten the security of sensitive information.	New
212	Corollary 5.10 Even if the security of one user is breached in SIM, the security of other users and sensitive information will not be compromised.	New

Security Architecture for Sensitive Information Systems

by

Xianping Wu
BCS, MBA, MNC

A Thesis Submitted in Fulfillment of
the Requirements for the Degree of
Doctor of Philosophy

Faculty of Information Technology
Monash University
Australia

2009

Abstract

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.

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. The research described in this thesis is based on dynamic key theory and group key theory to present a novel security architecture to enable sensitive information systems to overcome these challenges and meet the desired security goals for the three major components.

The proposed security architecture consists of dynamic key management, user-oriented group key management, authentication and authorization management and

sensitive information management, which guarantee the security of the three major components of sensitive information systems.

Because of the lack of the assessment properties of information security models, a new sensitive information security model is also presented in this thesis to evaluate the effectiveness of security architecture. This model proves that the security architecture satisfies the security goals. It can also be used to assess other security architectures, and thus makes a valuable contribution to the field of sensitive information systems security.

In summary, the proposed security architecture offers unique features necessary for the security of sensitive information systems. It also overcomes the limitations associated with existing security approaches and enables the complete protection of the three major components of sensitive information systems.

Copyright Notice A

Under the Copyright Act 1968, this thesis must be used only under the normal conditions of scholarly fair dealing. In particular no results or conclusions should be extracted from it, nor should it be copied or closely paraphrased in whole or in part without the written consent of the author. Proper written acknowledgement should be made for any assistance obtained from this thesis.

Copyright Notice B

I certify that I have made all reasonable efforts to secure copyright permissions for third-party content included in this thesis and have not knowingly added copyright content to my work without the owner's permission.

Xianping Wu



27 September 2009

Declaration

In accordance with Monash University Doctorate Regulation 17 / Doctor of Philosophy and Master of Philosophy (MPhil) regulations, the following declarations are made:

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Xianping Wu



08 June 2009

Dedications

Dedicated to my beloved wife, Sha Na, to my parents, Peixue Wu and Xiufen Li
and to the memory of grandpa

Acknowledgements

This thesis would not have been possible without the best efforts of many people. First of all, I would like to gratefully acknowledge my supervisors, Prof Balasubramaniam Srinivasan and Dr Phu Dung Le, for giving me this awe-inspiring opportunity to work on this research. I am grateful for their advice, encouragement and invaluable technical discussions. I very much appreciate their immense help during the research and for giving valuable feedback during the writing of this thesis. Without both of them, I would not have been able to complete this thesis.

I would like to specially express my gratitude to Osama Dandash. I much appreciate his financial aid to start my research life, and his encouragement when I felt under pressure during the research. I would also like particularly to acknowledge the contribution of Huy Hoang Ngo (Harry) and Dr Yi Ling Wang (Tony) for encouragement, discussions, brainstorming and cheering-up jokes, as well as other researchers Minh Le Viet, Dr Alex Tze Hiang Sim, Arun Mani, Xiya Fang, Minh Duc Cao, Abdulah Almuhaideb, Dr Samar Zutshi and Huamei Qi in Central South University.

I am thankful for the financial support (scholarship) from the Monash Research Graduate School. Special thanks to Julie Semon in Monash College for giving me the opportunity to teach; Nira Rahman in the Caulfield Library for thesis writing advice; and other library staff in resource finding. I thank Wei Wu, Snow, Dorin, Lin Zhang for warm friendship and support.

I thank and acknowledge the proofreading done by Megan Seen and my supervisors on thesis drafts. I also acknowledge the administrative support from John Sedgwick, Carmen Maestri, Chris Thomas, Michelle Ketchen, Allison Mitchell, Julie Austin, Katherine Knight, Diana Sussman, Duke Fonias and Akamon Kunkongkapun.

I cannot end without thanking my family, on whose constant encouragement and love I have relied throughout my research in completing this thesis. Many thanks are due to my wife, my soul mate, Sha Na, for her love, understanding, support, encouragement and her delicious Internet food. A million thanks are due to my parents and parents-in-law for their endless love and support in my educational pursuit.

Thank you all for letting me follow my dreams.

Table of Contents

ABSTRACT	I
COPYRIGHT NOTICE A	III
COPYRIGHT NOTICE B	IV
DECLARATION	V
DEDICATIONS	VI
ACKNOWLEDGEMENTS	VII
1. INTRODUCTION	1
1.1. INFORMATION SYSTEMS	1
1.2. SENSITIVE INFORMATION	3
1.2.1. <i>Characteristics of Sensitive Information</i>	4
1.2.2. <i>Protection of Sensitive Information</i>	5
1.3. SECURITY AND LIMITATIONS OF SENSITIVE INFORMATION SYSTEMS	6
1.3.1. <i>Retrieving Sensitive Information</i>	6
1.3.2. <i>Security Threats and Concerns of SIS</i>	7
1.4. MOTIVATIONS OF THE THESIS	8
1.5. OBJECTIVES OF THE THESIS	12
1.6. ORGANIZATION OF THE THESIS AND CONTRIBUTIONS	13
1.6.1. <i>Contributions of the Thesis</i>	16
2. SECURITY ISSUES OF SENSITIVE INFORMATION SYSTEMS	18
2.1. CRYPTOGRAPHIC SYSTEMS	19

2.1.1.	<i>Symmetric Cryptography</i>	20
2.1.2.	<i>Asymmetric Cryptography</i>	24
2.1.3.	<i>Summary</i>	27
2.2.	SECURING COMMUNICATION CHANNEL.....	27
2.2.1.	<i>Secure Communication in Unicast Channels</i>	28
2.2.2.	<i>Secure Communication in Multicast Channels</i>	38
2.2.3.	<i>Summary</i>	61
2.3.	SECURING USER INTERFACE	62
2.3.1.	<i>Proof by Knowledge</i>	62
2.3.2.	<i>Proof by Possession</i>	70
2.3.3.	<i>Proof by Property</i>	73
2.3.4.	<i>Authentication versus Authorization</i>	75
2.3.5.	<i>Summary</i>	76
2.4.	SECURING SENSITIVE INFORMATION STORAGE.....	79
2.4.1.	<i>Disk Encryption</i>	79
2.4.2.	<i>Database Encryption</i>	83
2.4.3.	<i>Summary</i>	87
2.5.	THE CURRENT MODELS FOR INFORMATION SECURITY	89
2.5.1.	<i>CIA Triad</i>	90
2.5.2.	<i>Parkerian Hexad</i>	92
2.5.3.	<i>Summary</i>	93
2.6.	CONCLUSION.....	94

3. SECURITY ARCHITECTURE FOR SENSITIVE INFORMATION SYSTEMS
..... **97**

3.1.	DYNAMIC KEY THEORY.....	99
3.1.1.	<i>Cryptographic Properties</i>	101
3.1.2.	<i>Dynamic Keys versus Symmetric Cryptography</i>	104

3.1.3.	<i>Dynamic Keys versus Asymmetric Cryptography</i>	107
3.2.	SECURITY ARCHITECTURE	110
3.2.1.	<i>Security Architecture Overview</i>	110
3.2.2.	<i>Engaged Users</i>	113
3.2.3.	<i>Dynamic Key Management</i>	115
3.2.4.	<i>User-oriented Group Key Management</i>	117
3.2.5.	<i>Authentication and Authorization Management</i>	118
3.2.6.	<i>Sensitive Information Management</i>	119
3.2.7.	<i>Structure of SecureSIS</i>	122
3.2.8.	<i>Entities Belonging</i>	123
3.2.9.	<i>Security Agreement</i>	124
3.2.10.	<i>Goals of SecureSIS</i>	125
3.3.	SENSITIVE INFORMATION SECURITY MODEL	127
3.3.1.	<i>SecureSIS Pentad</i>	128
3.3.2.	<i>Authenticity & Authority (AA)</i>	129
3.3.3.	<i>Integrity (IN)</i>	131
3.3.4.	<i>Non-repudiation (NR)</i>	132
3.3.5.	<i>Confidentiality (CO)</i>	133
3.3.6.	<i>Utility (UT)</i>	134
3.3.7.	<i>Summary on the SecureSIS Pentad</i>	135
3.4.	SUMMARY	136
4.	SECURITY ARCHITECTURE COMPONENTS.....	138
4.1.	DYNAMIC KEY MANAGEMENT	139
4.1.1.	<i>Dynamic Key Agreement</i>	139
4.1.2.	<i>Security Comparison</i>	141
4.2.	USER-ORIENTED GROUP KEY MANAGEMENT	142
4.2.1.	<i>Key Tree Structure</i>	143

4.2.2.	<i>UGKM Cryptographic Properties</i>	146
4.2.3.	<i>Group Keys</i>	147
4.2.4.	<i>Member Join</i>	148
4.2.5.	<i>Member Leave</i>	154
4.2.6.	<i>Periodic Rekeying Operation</i>	157
4.2.7.	<i>Security Comparison</i>	158
4.3.	AUTHENTICATION AND AUTHORIZATION MANAGEMENT	161
4.3.1.	<i>AAM Structure</i>	162
4.3.2.	<i>Initialization Protocol</i>	163
4.3.3.	<i>Logon Protocol</i>	164
4.3.4.	<i>AccessAuth Protocol</i>	165
4.3.5.	<i>Security Comparison</i>	168
4.4.	SENSITIVE INFORMATION MANAGEMENT	170
4.4.1.	<i>SIM Structure</i>	171
4.4.2.	<i>Data Operation</i>	174
4.4.3.	<i>Dynamic Membership Operations</i>	177
4.4.4.	<i>Security Comparison</i>	179
4.5.	SUMMARY	182
5.	SECURITY ANALYSIS AND DISCUSSION ON SECURESIS	185
5.1.	SECURITY OF DKM	186
5.1.1.	<i>Dynamic Keys in DKM</i>	187
5.1.2.	<i>Summary</i>	189
5.2.	SECURITY OF UGKM	189
5.2.1.	<i>Group Key Secrecy</i>	190
5.2.2.	<i>Forward Secrecy</i>	192
5.2.3.	<i>Backward Secrecy</i>	194
5.2.4.	<i>Collusion Resistance</i>	195

5.2.5. <i>Summary</i>	197
5.3. SECURITY OF AAM.....	198
5.3.1. <i>Introduction to the Spi Calculus</i>	198
5.3.2. <i>Logon Protocol</i>	200
5.3.3. <i>AccessAuth Protocol</i>	204
5.3.4. <i>Summary</i>	207
5.4. SECURITY OF SIM.....	208
5.4.1. <i>Security of Interchanging Sensitive Information</i>	209
5.4.2. <i>Security of Sensitive Information Storage</i>	210
5.4.3. <i>Summary</i>	214
5.5. SECURE SIS PENTAD ASSESSMENT.....	214
5.5.1. <i>Authenticity & Authority Discussion</i>	215
5.5.2. <i>Integrity Discussion</i>	218
5.5.3. <i>Non-repudiation Discussion</i>	221
5.5.4. <i>Confidentiality Discussion</i>	223
5.5.5. <i>Utility Discussion</i>	224
5.5.6. <i>SecureSIS Goals Discussion</i>	225
5.6. SUMMARY	227
6. CONCLUSION AND FUTURE WORK	230
6.1. REVISITING THE RESEARCH PROBLEM AND APPROACH	231
6.2. CONTRIBUTIONS	232
6.3. FUTURE WORK	235
REFERENCES.....	237
PUBLICATIONS	258

List of Figures

Figure 1.1. The Architecture of Generic Sensitive Information System.	7
Figure 1.2. Overview of Thesis Structure.	14
Figure 2.1. The Comparison of ESP and AH Protected IP Packet.	30
Figure 2.2. SRTP Session Key Derivation.....	37
Figure 2.3. CKDS (ING) Protocol.	41
Figure 2.4. GDH.1: An Example for Four Members.	43
Figure 2.5. GDH.2: An Example of Four Members.	44
Figure 2.6. GDH.3: An Example of Four Members.	46
Figure 2.7. LKH Key Tree.	49
Figure 2.8. LKH Member Joins the Group.	50
Figure 2.9. LKH Member Leaves the Group.	51
Figure 2.10. OFT Key Tree.....	53
Figure 2.11. OFT The Keys Known to a Group Member.....	54
Figure 2.12. OFT Member Join a Group.	55
Figure 2.13. OFT Member Leave a Group.	56
Figure 2.14. Subgroups and GSIs in Iolus Scheme.	58
Figure 2.15. EFS: File Encryption.	81
Figure 2.16. EFS: File Decryption.	82
Figure 2.17. Transparent Data Encryption Hierarchy.....	85
Figure 2.18. TDE in Oracle Database.	86
Figure 2.19. CIA Triad.....	91
Figure 2.20. Parkerian Hexad.	92
Figure 3.1. Entropy of Dynamic and Long-term Keys.	106

Figure 3.2. SecureSIS Core Component Overview.	110
Figure 3.3. Tangible Conceptual Architecture of SecureSIS.....	113
Figure 3.4. DKM Key Generation Flow.	116
Figure 3.5. AAM Process.....	119
Figure 3.6. Relationship between EI and EDK.....	121
Figure 3.7. The Structure of SecureSIS.	122
Figure 3.8. SecureSIS Pentad.....	129
Figure 3.9. Sensitive Information Integrity Triangle.	131
Figure 3.10. The Scope of Five Atomic Elements.....	136
Figure 4.1. Logical Structure of UGKM.....	145
Figure 4.2. User Join Operations.	149
Figure 4.3. Active User Join.	150
Figure 4.4. Passive User Join Cluster.	153
Figure 4.5. User Leave Operations.	154
Figure 4.6. Periodic Rekeying Timeline.	158
Figure 4.7. AccessAuth Protocol Logical Flow.....	166
Figure 4.8. Structure of a SIM Object.....	172
Figure 4.9. Retrieving Sensitive Information Flow Chart	173
Figure 4.10. Initial Status of SIM.	174
Figure 4.11. New Data Entry Status of SIM.....	175
Figure 4.12. Data Update Status of SIM.....	176
Figure 4.13. Data Deletion Status of SIM.....	177
Figure 4.14. Data Access Status of SIM.	177
Figure 4.15. Ownership Change of Sensitive Information.	179
Figure 5.1. The Organization of Security Analysis and Discussion.	186
Figure 5.2. Structure of the Logon Protocol.	201
Figure 5.3. Structure of the AccessAuth Protocol.	205

List of Tables

Table 1.1. Sensitive Information Levels of Classification in the U.S.....	3
Table 1.2. Sensitive Information Vulnerabilities.....	7
Table 2.1. Symmetric Keys Comparison.	21
Table 2.2. Comparison of CKDS, GDH.1, GDH.2 and GDH.3.	47
Table 2.3. Comparison of LKH and OFT.	57
Table 2.4. Advantages and Disadvantages of Multicast Communication Schemes.	60
Table 2.5. Advantages and Disadvantages of Knowledge, Possession and Property Factors.....	78
Table 2.6. Advantages and Disadvantages of Disk Encryption and Database Encryption.	88
Table 2.7. Problems in Sensitive Information Security.	95
Table 3.1. Applied SecureSIS Pentad with the Proposed SecureSIS.	137
Table 4.1. Key Managements Comparison.	141
Table 4.2. Security Comparison of Group Key Management.	159
Table 4.3. Security Comparison of AAM to Kerberos and its Successors.	168
Table 4.4. Security Comparison of SIM to Other Approaches.....	181
Table 4.5. SecureSIS Components vs. Goals.....	184

Chapter 1

Introduction

1.1. Information Systems

The use of information has become a pervasive part of our daily life; we have become “... an information society” [GoGo96]. 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 [DaOl85], 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 systems¹ are a powerful instrument for solving problems.

An information system (IS) is an organized set of components for collecting, transmitting, storing, and processing data in order to deliver information for action [Zw97]. It supports operations, management, and knowledge work in organizations. The use of information systems has increased due to economic and social issues.

¹ In this thesis, we use the term of information systems to represent computer-based information systems.

The functions of information systems include services which provide value to users or to other services via messages, which carry a meaning to users or services. Also, as IT becomes more sophisticated, the availability of these services and messages in organizations grows and spreads. The availability of IT shifts people from conducting business and communication in traditional to electronic ways. For example, people are able to access and manage their own bank account via online banking anytime and anywhere electronically, rather than physical banking, in which people have to wait in queues and undergo long verification processes in order to gain services. In addition, organizations issue electronic bills (e-bills) instead of paper bills in order to reduce the costs of paper bill delivery.

Recently, the use of information systems has obtained attention due to its high growth rate. An IDC (International Data Company) [Li08] study in 2007 noted that Internet banking in China had increased by 25.4% from the previous year and mobile banking by 19.3%. The IDC study predicted that online bank and mobile banking markets from 2008 to 2012 would increase rapidly with respective compound annual growth rates of 23.1% and 24.9%. Also, WinterGreen Research and Markets [CuEu08] forecast analysis indicates that the use of electronic medical record (EMR) systems is anticipated to increase to a rate of 63% by 2013. The rapid growth of information systems is not surprising. Compared to traditional information systems, the electronic information systems offer improved efficiency, process control, services and information process [DaOl85, GoGo96, Zw97].

1.2. Sensitive Information

The use of electronic information in organizations has raised problems. The importance of information protection reaches to the corporate boardroom, because failure to protect electronic information assets may result in lost customer and investor confidence. According to Parker [Pa98], information that has strategic value in organizations should be protected. This includes market-sensitive proprietary information, financial information, trade secrets, medical information, military information and human resources information. This information needs to be treated as sensitive information², that is, it needs to be recognized as “information or knowledge that might result in loss of an advantage or level of security if disclosed to others” [Pu07].

According to the U.S. government [Nc03], sensitive information is categorized into two classifications (shown in Table 1.1): non-classified and classified.

Table 1.1. Sensitive Information Levels of Classification in the U.S.

Non-classified	Sensitive Private Info	unauthorized disclosure could have a negative effect on its owner
	Confidential Business Info	public disclosure may harm a business
Classified	Restricted	public disclosure could have undesirable effects or do some harm
	Confidential	unauthorized disclosure could damage national security
	Secret	unauthorized disclosure could seriously damage national security
	Top Secret	unauthorized disclosure could severely damage national security
	Ultra Secret	unauthorized disclosure could existentially damage national security, international stability or wartime advantage

² In this thesis, sensitive information refers to digital critical information.

As argued by the U.S. government, “...loss, misuse, modification or unauthorized access to sensitive information can adversely affect the privacy of an individual, trade secrets of a business or even the security, internal and foreign affairs of a nation depending on the level of sensitivity and nature of the information” [Nc03].

1.2.1. Characteristics of Sensitive Information

Sensitive information has four primary characteristics [Pa98] that enables its comprehension:

- **Kind** – the type of information: for example, knowledge, descriptive, instructive, expository, factual, fictional, monetary, artistic, accounting or another type.
- **Representation** – the presentation of information: for example, in graphic images, coded symbols (digital text such as Unicode or ASCII code), digital sounds or videos.
- **Form** – the structure of information: for example, its style, language, syntax, encoding (encryption with a secret key), format and size.
- **Medium** – the physical manifestation of information: for example, electromagnetic pulses in space (radio waves) or electronic switches in computers (digital signals).

In addition, sensitive information has a number of other characteristics³ [LaBrHa85, Pa98, SoCh05] that help us determine the need for security.

- **Authenticity** – refers to the truthfulness of origins, attributions, commitments, sincerity, devotion, and intentions.

³ We hold over the other important characteristics, based on CIA Triad [Pe08] (confidentiality, integrity and availability), Parkerian hexad [Pa98] and DoD [LaBrHa85], for later discussion. Meanwhile, the Parkerian hexad adds three additional attributes to the three classic security attributes: utility, possession or authority and authenticity, and DoD adds Non-repudiation attribute to CIA Triad.

- **Confidentiality** – ensures that information is accessible only to those authorized to have access.
- **Possession (authority)** – refers to the ownership or control of information.
- **Integrity**-refers to the validity, trustworthiness and dependability of information.
- **Utility** – refers to the usefulness of information.
- **Non-repudiation** – refers to the un-deniability for entities to perform actions on sensitive information.
- **Availability** – refers to having timely access to information.

1.2.2. Protection of Sensitive Information

The Committee on National Security Systems [Cn92] in the USA defines information security as “...the protection of information systems against unauthorized access to or modification of information, whether in storage, processing, or transit, and against the denial of service to authorized users or the provision of service to unauthorized users, including those measures necessary to detect, document, and counter such threats.” In this sense, sensitive information is at risk, as every sensitive information breach impacts organizations negatively. Protecting sensitive data is a growing concern for organizations around the globe because of its financial implications; however data security is also necessitated by stringent industry and government regulations.

Sensitive information requires three types of safeguards. In addition to technical safeguards, to be secure, sensitive information also needs administrative safeguards. This is because, regardless of the technology used to lock or secure sensitive information, the way people work with one another and with information ultimately has the greatest impact on security. Finally, physical safeguards need to be considered.

Technical safeguards address topics such as authentication of users, audit logs, data integrity checks, and transmission security (encryption), while administrative safeguards address organizational controls such as policies and procedures, risk analysis and training. Physical safeguards cover issues such as access to buildings and workstations (locks and keys), disposal of computers and hard drives, and data backup and storage requirements. Technical safeguards have become the focus of research for sensitive information protection due to the increasing maturity of administrative and physical safeguards.

1.3. Security and Limitations of Sensitive Information Systems

The major reason behind sensitive information system's (SIS) lack of security is due to the inherent nature of IS which requires information collecting, processing, transmitting and storing in order to deliver information for action. If information were static and stationery, security would be less of an issue. The major processes involved in retrieving sensitive information, and security threats and concerns in SIS, are detailed as following sections.

1.3.1. Retrieving Sensitive Information

To describe the retrieval process, we use a simple and generic architecture as shown in Figure 1.1. First of all, before the retrieval process can be initiated, it is necessary to transform sensitive information into a logical view, which gives the view of how information is structured and organized. Once the logical view of the sensitive information is defined, the information manager can build an index of the sensitive information.

With the sensitive information indexed, the retrieving process can be initiated. The major components involved in the process are *communication channel*, *user interface* and *sensitive information storage*. Firstly, a legal user specifies a user need via a *user interface*, and the need is then processed to obtain the sensitive information from *sensitive information storage* through a *communication channel*.

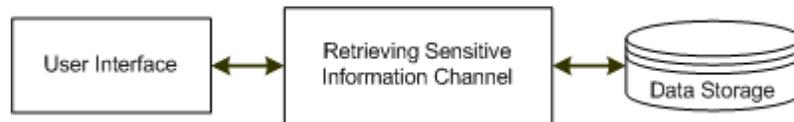


Figure 1.1. The Architecture of Generic Sensitive Information System.

1.3.2. Security Threats and Concerns of SIS

The three major components involved in the action of data retrieval - *communication channel*, *user interface* and *sensitive information storage* - are all potential targets for adversaries wanting to benefit from security weaknesses. According to [CIWi87, Pa98, SoCh05], security threats and concerns are raised against the key aspects⁴ of sensitive information, as shown in Table 1.2.

Table 1.2. Sensitive Information Vulnerabilities.

Characteristics	Target	Vulnerability
Authenticity	user interface	impersonation guessing spoofing
Confidentiality	communication channel information storage	eavesdropping intercepting
Possession	information storage	session hijacking
Integrity	communication channel information storage	falsification forgery
Utility	information storage	property damaging ⁵

⁴ Key aspects refer to characteristics of sensitive information.

⁵ Property damaging means accidentally lost the encryption key of encrypted the only copy of valuable information.

As reported [We05], British hacker, Gary McKinnon, caused nearly US \$1 million in damage due to breaking into US Navy, US Army, NASA and Pentagon systems. Also, according to [Id08], US \$3.2 billion has been lost as a result of internet identity theft in 2007 in the United States alone. With Asia's online population rapidly increasing, the global figure could easily be twice that in just a matter of years. These figures indicate the vulnerability of SIS. The U.S. Department of Defense Science Board has issued a report "Information Management for Net-Centric Operations" [Ds07]. This report stresses the need for extraordinary effort on information systems security, because "the threat to the information system will continue to evolve as globalization and the information revolution force changes in structure and technology". The report goes on to state that while the "...network approach and strategy enable new paradigms for sharing and using information, this capability also has the potential to significantly increase the nation's vulnerability to internal and external threats". It recommends an increase in current funding, funding for information systems over future years in defense programs, and that the programs focus on information assurance for the entire enterprise.

1.4. Motivations of the Thesis

With the development of network technology, the use of the Internet has pervaded everyday life. It is used for many services such as file transfers, internet payments, and viewing electronic documents. Meanwhile, the proliferation of electronically-accessible information has led to research and development in information systems to help users search for, fetch and share relevant and meaningful information.

The concept of information is closely related to notions of constraint, communication, control, data, form, instruction, knowledge, meaning, mental stimulus, pattern, perception, and representation. The concept has been developed rapidly in open network systems, typified by the Internet, to provide sufficient convenience for users, especially to group users to manage information for sharing, exchanging and using.

Advancement in information systems promises dramatic leaps forward in our daily life, especially in stock markets, financial institutions, and medical centres. For example, medical centres employ electronic medical record systems to share patients' records from other hospitals to rapidly diagnose the patients, and financial advisors can respond quickly to fluctuating stock markets by adopting information systems.

Utilising 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 organisations of all sizes.

Despite this priority, the majority of existing electronic information systems [BaFi01, BhDe98, HoChWa07, MeIlKa00] 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 [BeIsKu99, CaMiSt99, GeGoMa98, GeIsKu00] 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. As Bard [Ba04] states, the private *communication channel* is breakable due to the long-term shared identical cryptographic keys. Also, with the shared identical keys, adversaries can break the security of information systems via eavesdropping or intercepting.

Alternatively, cryptography techniques are employed to protect sensitive information storages rather than establishing private *communication channels*. These information systems [Bo07, Hs08, Na05] depend 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 [Bo07]. Similar mechanisms are also used for Oracle Database [Na05] and Microsoft SQL Server [Hs08], 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.

Securing the *user interface* to prevent unauthorized access to information systems is another approach to protecting sensitive information in organizations. This form of security uses measures such as security tokens, passwords or biometric identifiers. Kerberos is a representative 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 [Er03, HaMe01, SiCh97] 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 [KoNeTs94]. In 2008, Cervesato et al. [CeJaSc08] pointed out that man-in-the-middle attack can breach Kerberos-based information systems.

The existing approaches all have a common limitation: the employment of long-term shared keys or public keys. Among symmetric key encryption algorithms, only the one-time pad can be proven [Sh49] to be secure against any adversary, regardless of the amount of computing power available. Also, there is no asymmetric scheme with the one-time pad property, since all asymmetric schemes are susceptible to brute force key search attack [Ka67]. Therefore, once the keys are exposed, the protected SIS will be compromised.

In addition to above security threats and concerns relating to *communication channel*, *user interface* and *sensitive information storage*, the ownership of sensitive information presents another security concern. This concern evolves from simple organizational structure. A traditional approach to managing information ownership is to use access control [FeKuCh03]. However, this approach does not allow for dynamic ownership, whereby the owner of the information is likely to be changed, but the security characteristics of the information must be maintained.

The limitations of existing security measures can be summarised as follows:

- No proper authentication and authorization mechanisms to conduct dynamic membership of groups and individuals to share or access sensitive information.

- No prevention of legal users accessing unauthorized sensitive information against internal security threats.
- No proper critical information storage protection mechanism, which thwarts security threats of compromising credentials of information systems.
- No dealing with dynamic information ownership.

The above limitations in the existing body of knowledge motivate our research in order to eradicate these weaknesses and develop appropriate security architecture for SIS.

1.5. Objectives of the Thesis

This research aims to investigate the major security issues in current information systems, analyze these problems and then develop novel generic security architecture for SIS. The objectives of this thesis are:

- To develop general security architecture for various kinds of SIS. This architecture consists of a number of components to protect sensitive information. It defines characteristics and interactions among engaging entities.
- To develop a sensitive information security model to evaluate security architecture of SIS.
- To design practical and secure authentication and authorization protocols⁶ for individuals and group users to share sensitive information. The proposed protocols discard the use of long-term shared keys to achieve high security and tight access control.

⁶ It achieves security by confirming provenance & identity.

- To develop a new group key management component to handle dynamic information ownership and make sensitive information sharing more flexible and secure⁷.
- To develop a key generation management component to manage and deliver cryptographic keys for engaging components and users. This component defines key security properties to ensure that minimum security requirements are satisfied.
- To develop a new sensitive information management component for data storages⁸. This component protects sensitive information when information storage is compromised.
- To perform formal security analysis to illustrate that each proposed component has better security than existing approaches and to evaluate the architecture using the proposed information security model.

1.6. Organization of the Thesis and Contributions

This section provides an overview of the research presented in the following five chapters. Figure 1.2 provides a diagrammatic overview of the thesis structure. The key contributions made by each chapter are described in Section 1.6.1.

Chapter 2 provides a critical analysis of previous research for sensitive information protection used in SIS. Two main bodies of research are identified and reviewed: (i) security protections of the three major components in the process of sensitive information retrieving are studied and reviewed; and (ii) information security model is

⁷ It guarantees security of communication.

⁸ It safeguards sensitive information storage.

reviewed. Limitations in previous research of security SIS motivate us to do more research in this thesis.

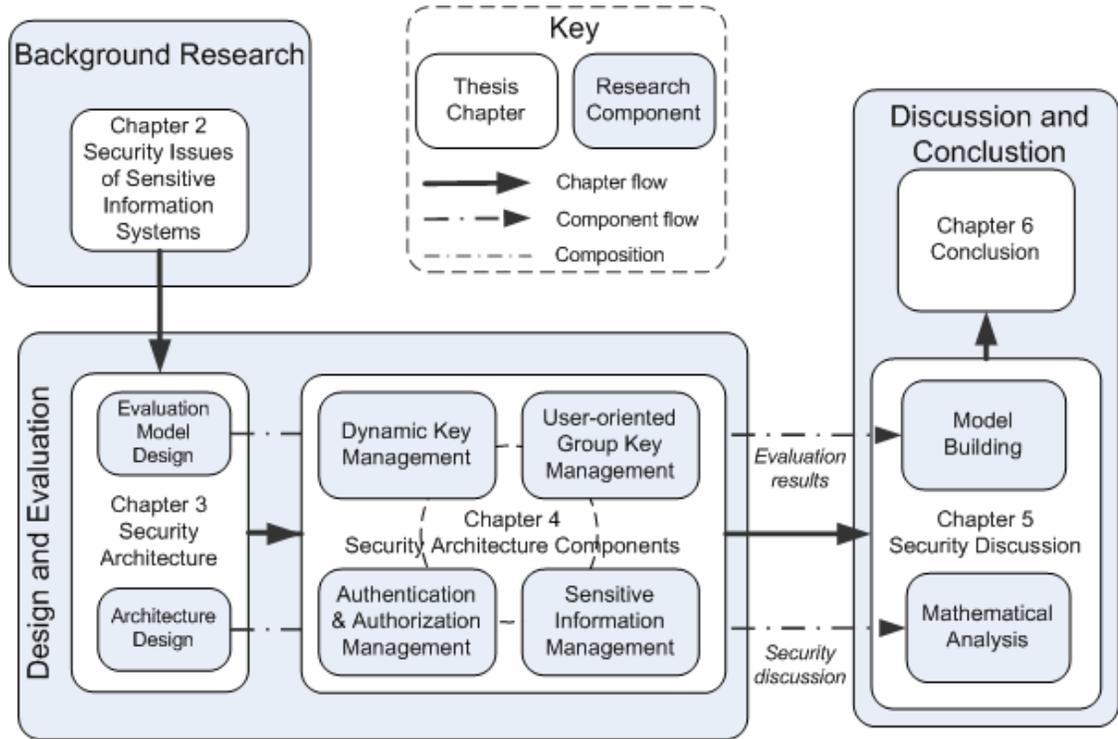


Figure 1.2. Overview of Thesis Structure.

Chapter 3 proposes formal security architecture for SIS, and also proposes an information security model to evaluate the security architecture for SIS. The architecture includes four components to support the model's security, later formalized in Chapter 4.

Chapter 4 details four components of the proposed secure architecture. In the first section, dynamic key theory is summarised and defined formally, and then the cryptographic properties of dynamic keys are discussed. Finally, we demonstrate how to apply the dynamic keys to other components (*communication channel, user interface and sensitive information storage*) in order to protect sensitive information.

In the second section, a new group key agreement is proposed which emphasizes the privacy of sensitive information owners. The agreement classifies group users into two categories to protect privacy of sensitive information while reducing rekeying complexities. We provide a number of algorithms for group member joining and leaving. The agreement satisfies the security of proposed architecture.

In the third section, an authentication and authorization management component is introduced. The component conducts a suite of protocols to achieve high security and tight access control to protect *user interface*. This component adopts the proposed dynamic key management model and new group key arrangement to manage information sharing security, and to achieve flexibility of authentication between groups and individuals.

In the fourth section, a sensitive information management component is proposed. This component integrates dynamic keys of users or groups with sensitive information to protect *sensitive information storage*. It precludes legal users accessing unauthorized information, and it also prevents information leakage from compromising *sensitive information storage*.

Chapter 5 discusses the substantive findings from the previous chapters. The discussion offers: (i) formal security analysis of the four components; and (ii) building and discussion of the proposed sensitive information security model.

Chapter 6 concludes the thesis by summarizing the main findings and contributions from the thesis. Limitations of the research and routes for further work are presented.

1.6.1. Contributions of the Thesis

This thesis makes a number of research contributions to state-of-the-art in sensitive information protection. These contributions are presented throughout the thesis, as follows:

- A novel security architecture and sensitive information security model for sensitive information is proposed in Chapter 3, and has been presented in Wu et al. [WuLeSr09].
- The formal description of dynamic key theory is tailored in Chapter 4, and the cryptographic properties of dynamic keys are given and proved [NgWuLe09a, WuNgLe09]. A dynamic key generation technique has been patented in Wu and Le [WuLe06].
- A user-oriented group key agreement [WuNgLe08b] is proposed in Chapter 4 in order to secure information sharing and protect privacy of individuals [ChWaWu06, NgWuLe08a].
- A secure and flexible authentication & authorization management scheme [WuNgLe08a, WuNgLe09] is designed in Chapter 4 to provide proper authentication and access control among individuals and groups [NgWuLe08b, NgWuLe09].
- Integrating dynamic keys with sensitive information [WuLeSr08] is introduced in Chapter 4 to enhance security of *sensitive information storage*.

As a result of these developments, we claim that the proposed security architecture for SIS 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 storage 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.

Chapter 2

Security Issues of Sensitive Information Systems

Goals. This chapter contains reviews of existing approaches, main issues and concepts relating to sensitive information protection. According to the process of sensitive information retrieving, we divide the study into - the protection of *communication channel*, *user interface* and *sensitive information storage* - three related work areas. We argue that the security threats and concerns of existing approaches in the sensitive information systems are long-term shared identical cryptographic keys and public keys. Also, there is no complete security architecture to help protect sensitive information in the retrieving process.

Besides, the security assessment properties of sensitive information are studied, we argue that the existing sensitive information security models lack of assessment properties to assess the security architecture, and it fails to address privacy concerns.

In Section 2.1, cryptographic systems (symmetric cryptography in Section 2.1.1 and asymmetric cryptography in Section 2.1.2) are reviewed primarily to help understand the following discussion. In Section 2.2, existing approaches for protecting *communication channel* (unicast channel in Section 2.2.1 and multicast channel in Section 2.2.2) are discussed to find security threats. In Section 2.3, authentication

factors (knowledge in Section 2.3.1, possession in Section 2.3.2 and property in Section 2.3.3) in securing *user interface* are reviewed to identify the weaknesses of existing approaches. In Section 2.4, existing cryptographic techniques (disk encryption in Section 2.4.1 and database encryption in Section 2.4.2) in protecting *sensitive information storage* are discussed to identify the problems of protecting sensitive information at rest⁹. In Section 2.5, information security models are studied to determine the insufficiency in sensitive information security. Finally, in Section 2.6, we conclude this chapter.

2.1. Cryptographic Systems

“Since the early stages of human civilization, there has been a need to protect sensitive information from falling into the wrong hands. To achieve such secrecy, mankind has relied on a branch of mathematics known as cryptography, which is the study of designing methods to securely transmit information over non-secure channels” [BrFo05]. One of the most important aspects of any cryptographic system is key management.

Therefore, cryptographic key management is reviewed in this section. According to RFC2828 [Sh00], the key management refers to “the process of handling and controlling cryptographic keys and related material (such as initialization values) during their life cycle in a cryptographic system, including ordering, generating, distributing, storing, loading, escrowing, archiving, auditing, and destroying the

⁹ Sensitive information at rest (Sensitive information storage) is a term that is used to refer to all data in computer storage while excluding data that is traversing a network or temporarily residing in computer memory to be read or updated [Bu06].

material”. As such, cryptographic keys and related material are the important element in key management. The keys are classified as symmetric and asymmetric.

2.1.1. Symmetric Cryptography

A symmetric key is a single cryptographic key (known as a secret key) that represents a shared secret between the sender and recipient. The key can be used to secure communication or derive other keys. According to the National Institute of Standards and Technology (NIST)¹⁰ [BaBaBu06], symmetric keys are cataloged into different types, and listed as followings¹¹:

- **Master keys** (long-term shared keys or static keys) – a master key is used to derive other symmetric keys using symmetric cryptographic methods. A master key can be used over a longer period of time to derive (or re-derive) multiple keys for the same or different purposes.
- **Session keys** (ephemeral keys) – a session key is a single-use symmetric key used for encrypting all messages in one communication session. Normally, it involves key negotiation and distribution.
- **One-time pad keys** (OTP keys) – an OTP key is a single-use symmetric key (pad) as long as the plaintext and used only once. No real-world implementation.
- **Key wrapping keys** (key encryption keys) – a key wrapping key is used to wrap (that is, encrypt) keying material that is to be protected and may be used to protect multiple sets of keying materials. The protected keying material is then transmitted or stored, or both.

¹⁰ NIST publishes Federal Information Processing Standards (FIPS) and NIST Recommendations that specify cryptographic techniques for protecting sensitive unclassified information.

¹¹ Only used symmetric key types in this thesis are listed.

- **Authentication keys** (tokens, credentials) – an authentication key is used with symmetric key algorithms to provide assurance of the integrity and source of messages, communication sessions or stored data.
- **Dynamic keys** (one-time keys) – are used once and then discarded, either to authenticate or encrypt a message.

Security Comparison of Symmetric Cryptographic Keys

Among the defined symmetric keys, key wrapping keys and authentication keys can be master keys, session keys or dynamic keys. We therefore compare long-term shared, session, one-time pad and dynamic keys. This comparison is presented in Table 2.1. The comparison criteria is selected based on Forouzan and Fegan [FoFe03].

Table 2.1. Symmetric Keys Comparison.

Symmetric Keys	Keys Comparison Criterion				
	Lifetime	Distribution	Sync	Storage	Security
Long-term Keys	indefinite	public key	no	one	low
Session Keys	session	hybrid	no	zero	moderate
One-time Pad (No real-world implementation)	once	physical	yes	indefinite	high
Dynamic Keys	once	public key	yes	one	high

Key Lifetime refers to the length of time the key can be used for encryption. The lifetime of long-term shared keys is indefinite, since the lifetime depends on the security policy and key size. Session keys, used for securing all messages in the one communication session, are also called ephemeral keys. Their lifetimes are less than long-term shared keys. The one-time pad and dynamic keys are used only once.

Among cryptographic keys, the one-time pad and dynamic keys have the smallest lifetime.

Key Distribution refers to the process of exchanging shared secrets for encryption. Strictly, long-term keys and secret dynamic keys employ public key cryptography to exchange secrets in order to overcome symmetric key distribution problems. Session keys can be distributed by using a shared long-term key or a public key starting at every communication session. A one-time pad is normally exchanged via physical devices. Distributing very long one-time pad keys is inconvenient and usually poses a significant security risk. The pad is essentially the encryption key, but unlike keys for modern ciphers, it must be extremely long and is consequently difficult for humans to remember.

Theoretically, the more frequently keys are exchanged, the more secure they are, because the adversary has less cipher text to work with for any given key. On the other hand, the distribution of keys delays the start of any exchange and places a burden on network capacity. Therefore, long-term keys and dynamic keys have advantages over others in key distribution. Nevertheless, in term of security, dynamic keys, unlike long-term keys, are used only once, and do not involve key distribution (only once for initial secret sharing). Dynamic keys are consequently more secure than long-term keys.

Key Synchronization refers to the process of ensuring that the key for encryption is the same for the two involved entities. Because long-term keys are shared and session keys are distributed for each transaction, these do not need key synchronization. However, for one-time pad keys and dynamic keys, both need to synchronize the key in order to ensure communication between entities. In this regard, one time pad keys

and dynamic keys are less convenient. Conversely, they do not need to synchronize keys before each transaction unless network failure occurs.

Key Storage is a measure of space consumption for storing keys. Long-term keys need only one key in an entity. According to the nature of dynamic keys, dynamic keys are used only once, and generated based on a form of shared secret. Therefore, they also require storage for only one key. Session keys are exchanged at beginning of each transaction, so there is no need to store any session keys. One-time pad keys require an unknown amount of storage, because all the keys are stored once, and the number of keys depends on the security policy.

The above comparison on the security of symmetric keys informally shows that dynamic keys has advantages over other cryptographic keys in terms of key lifetime, key distribution and key storage aspects, which provide higher security.

Symmetric Cryptography Overview

Historically, the first people to clearly understand the principles of cryptography and to elucidate the beginnings of cryptanalysis were the Arabs [Ka67]. The Arabs studied the art of unscrambling secret messages without knowledge of the secret key. The first modern symmetric key system invention, at the IBM Watson Research Lab in the 1960s under the leadership of Horst Feistel, is known as the Feistel cipher [Fe73]. Later, a modified version of the cipher originally known as Lucifer was published in the National Institute of Standards and Technology (NIST). Lucifer became the United States Data Encryption Standard (DES) [Nb88, Nb93]. DES has been in use for the last 20 years because of its short key size and reasonable security. However, it is nowadays possible in certain cases to conduct a brute-force attack on the entire key

space. Biham and Shamir [BiSh93] report the first theoretical attack on DES with less complexity than brute force, and Matsui [Ma94] demonstrates the first experimental cryptanalysis of DES using linear cryptanalysis. After that, DES was rewritten as Triple DES (TDES) [Nb99] to enhance its security. As TDES, the algorithm was believed to be practically secure, although there were theoretical attacks [Bi96, Lu98a].

In recent years, TDES has been superseded by the Advanced Encryption Standard (AES), which was developed by Joan Daemen and Vincent Rijmen [DaRi02, Nb01]. The standard uses the Rijndael block cipher, and specifies the key and block sizes that must be used. It has been analyzed extensively and is now used worldwide. It is fast for both software and hardware [ScWhWa00] and uses less memory. As a new encryption standard, it is currently being deployed on a large scale. However, despite its acceptance, a theoretical attack was announced by Nicolas Courtois and Josef Pieprzyk to indicate a potential weakness in the AES system [CoPi02]. The first successful attacks against AES implementation were side-channel attacks [OsShTr06]. Breaching the AES system was only a matter of time, as Shannon mathematically proved that among symmetric key encryption algorithms, only the one-time pad is secure against any attack. No other symmetric cryptography is information theoretically secure [Sh49]. Moreover, one of the most important drawbacks of symmetric cryptography is key distribution [Sa03].

2.1.2. Asymmetric Cryptography

An asymmetric key is a combination of two keys (known as public keys) commonly referred to as public and private keys. The public key and the private key are a matched set. According to NIST, the following asymmetric key types are given:

- **Private key** – is a cryptographic key, used with a public key cryptographic algorithm, that is uniquely associated with an entity and is not made public. In an asymmetric (public) cryptosystem, the private key is associated with a public key. Depending on the algorithm, the private key may be used to:
 - compute the corresponding public key,
 - compute a digital signature that may be verified by the corresponding public key,
 - decrypt data that was encrypted by the corresponding public key,
 - compute a piece of common shared data, together with other information.
- **Public key** – is a cryptographic key used with a public key cryptographic algorithm that is uniquely associated with an entity and that may be made public. In an asymmetric (public) cryptosystem, the public key is associated with a private key. The public key may be known by anyone and, depending on the algorithm, may be used to:
 - verify a digital signature that is signed by the corresponding private key,
 - encrypt data that can be decrypted by the corresponding private key, or
 - compute a piece of shared data.

Asymmetric Cryptography Overview

To put it in a historical perspective, asymmetric key systems were invented in the late 1970s. The first invention of asymmetric key algorithms was by James H.

Ellis, Clifford Cocks, and Malcolm Williamson at GCHQ¹² in the United Kingdom. The United States National Security Agency (NSA) also claims the first contribution to asymmetric key systems. The concept of asymmetric cryptography was created by Diffie, Hellman and Merkle (DH) [DiHe76b, He78]. Asymmetric cryptography does not provide “perfect secrecy” in the Shannon sense. However, from a practical point of view, it solves key distribution problem in symmetric key systems.

In the early days of public key systems, Merkle invented a public key algorithm called the “Knapsack algorithm” and his PhD thesis [Me79] influenced future public key systems. Rivest, Shamir, and Adleman [RiShAd78] at MIT invented RSA in 1977. RSA was the first algorithm known to be suitable for signing as well as encryption. It was believed to be secure given sufficiently long keys and the use of up-to-date implementation. However, although RSA was widely used in electronic commerce protocols, Peter Shor has shown that Shor's algorithm, when used with a quantum computer, can break RSA. Other researchers [Bo99, Co97, Wi90] have also reported the possibility of breaking RSA due to its short key size.

Another direction for public key systems was suggested by [Ko87] and [Mi86] based on the algebraic structure of elliptic curves over finite fields, known as elliptic curve cryptography (ECC). Although no mathematical proof of difficulty has been published for ECC, the U.S. NSA has endorsed ECC as recommended algorithms and allows its use for sensitive information protection up to the top secret category with 386-bit keys. Lenstra and Verheul's [LeVe01] study indicates that a 160-bit ECC key provides the same security as a 1024-bit RSA key.

¹² GCHQ: Government Communications Headquarters; a British intelligence agency responsible for providing signals intelligence (SIGINT) and information assurance to the UK government.

While it would appear that the security of ECC is stronger than that of Diffie-Hellman and RSA, ECC is still in its infancy, and has not undergone the kind of testing that has been applied to RSA and DH. A number of researchers [LaMo08, WiZu98] have proposed theoretical means of breaking ECC. As Kahn [Ka67] states “all asymmetric schemes are susceptible to brute force key search attack”.

2.1.3. Summary

In Section 2.1, cryptographic keys and historical background of cryptographic systems were explored. The following findings can be presented:

- Among all symmetric cryptographic keys, dynamic keys provide stronger security than others, comparable with long-term, session and one-time pad keys.
- Symmetric keys involve key distribution, which might compromise the security of cryptographic systems.
- Asymmetric cryptography is relatively computationally costly compared with symmetric cryptography.
- Asymmetric cryptography is susceptible to brute force key search attacks.

This section has primarily reviewed cryptographic systems to help understand cryptographic approaches in sensitive information protection. In the next three sections we explain the employment of the cryptographic systems in securing *communication channel*, *user interface* and *sensitive information storage*.

2.2. 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 [TiKh92]. 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.

2.2.1. 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) [At95], SEED¹³ [LeLeYo05], Secure Sockets Layer (SSL), Transport Layer Security (TLS) [DiRe08, FrKaKo96] and Secure Real-time Transport Protocol (SRTP) [LeNaNo07, OrMcBa04], 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. In addition, SRTP is a newly-

¹³ SEED, based on the Feistel network and developed by the Korean Information Security Agency, is used broadly throughout South Korean industry to replace 40 bit SSL.

proposed protocol for securing multimedia forms of sensitive information. SRTP provides encryption, message authentication and integrity and replay protection for both unicast and multicast channels. We therefore investigate the use of the cryptographic key in IPsec, SSL/TLS and SRTP in the next section.

IPsec

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. IPsec uses the following protocols to perform various functions [Ho05, ThDoG198]:

- Internet key exchange (IKE and IKEv2) to set up a Security Association (SA) by handling negotiation of protocols and algorithms and to generate the encryption and authentication keys to be used by IPsec.
- Authentication Header (AH) to provide connectionless integrity and data origin authentication for IP datagrams and to provide protection against replay attacks.
- Encapsulating Security Payload (ESP) to provide confidentiality, data integrity and data origin authentication of IP packets and also to provide protection against replay attacks.

The difference between AH and ESP is that an ESP packet includes enciphered data and authentication information whereas an AH packet only includes authentication information. This is illustrated in Figure 2.1. The encryption algorithms used with ESP can be DES, TDES or AES.

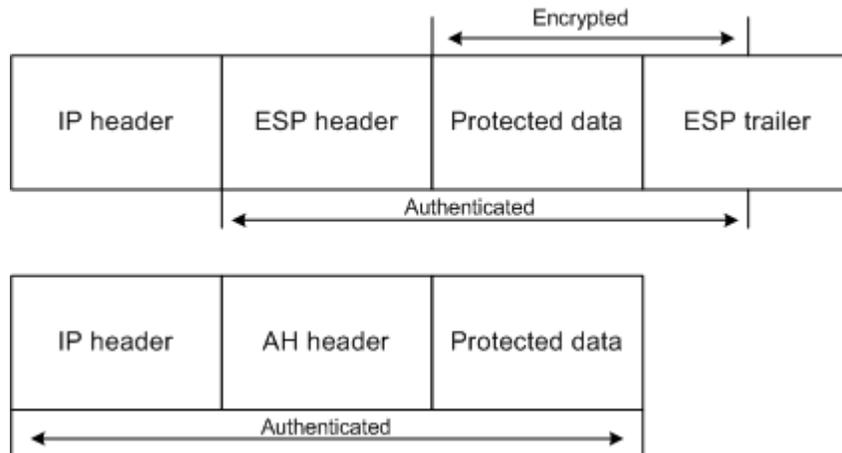


Figure 2.1. The Comparison of ESP and AH Protected IP Packet.

IPsec uses the concept of a security association (SA) as the basis for building security functions into IP. SA is the bundle of algorithms and parameters (such as keys) that is used to encrypt and authenticate a particular flow in one direction. Therefore, in normal bi-directional traffic, the flows are secured by a pair of SAs maintained by an SA database (SADB).

Since the security of IPsec provided by AH or ESP requires shared keys to perform authentication and/or confidentiality, the key distribution and key type determine the partial security of IPsec. As IPsec employs IKE or IKEv2 to set up a session secret (session key), the security of IKE bundles the security of IPsec.

IKE or IKEv2 uses a Diffie-Hellman (DH) key exchange to set up a shared session key from which cryptographic keys are derived. Public key techniques or, alternatively, a pre-shared key, are used to mutually authenticate the communicating parties. IKE/IKEv2 is a hybrid of the STS (Station to Station) [DiOoWi92], the Oakley [Or98] and SKEME (Versatile Secure Key Exchange Mechanism for Internet) [Kr96] protocols. IKE/IKEv2 operates inside a framework defined by the Internet Security

Association and Key Management Protocol (ISAKMP) [MaScSc98] as ISAKMP provides a framework to authenticate, exchange keys and eventually establish security associations.

IKE has a close relationship with ISAKMP, because ISAKMP typically is used with IKE for key exchange, although ISAKMP is designed to support many different key exchanges. Establishing an IPsec connection requires two phases:

- i) **Phase I:** parameter negotiation phase. It uses public key cryptography and runs a key management protocol to generate the initial shared SA called ISAKMP SA or Phase I SA. The secret keys in the SA are associated with symmetric cryptography to protect Phase II protocol in the key exchange phase.
- ii) **Phase II:** key exchange phase. It is under the protection of Phase I SA, and runs a key management protocol to generate more SAs called Phase II SA. Both phases are used to protect communication between communication entities.

Cryptographic keys play a key role in securing a unicast *communication channel*; hence, we will further explore Phase II to show how to exchange the cryptographic key used in protecting data traffic. Suppose after Phase I, three secret keys (K_d , K_a and K_e are used for deriving other keys, authenticating messages and encrypting messages, respectively) are shared exclusively between Initiator (**I**) and Responder (**R**). The protocol is also called QUICK mode and is described as follows:

- i) **I** wants to secure communicate with **R**. First, **I** generates a nonce N_I , and a token key using authentication key K_a and sends to **R**:

$$I \rightarrow R : HDR, \{SA, N_I, \tau_I\} K_e$$

where HDR represents ISAKMP header and $\tau_I = prf(K_a, SA, N_I)$;
meanwhile prf is a pseudorandom function (hash function) [BeCaKr96].

- ii) After received the message, **R** generates a nonce N_R and a new token key using shared K_a and sends to **I**:

$$R \rightarrow I : HDR, \{SA, N_R, \tau_R\} K_e$$

where $\tau_R = prf(K_a, SA, N_I, N_R)$.

- iii) After **I** verifies the message in ii), a new token is generated and sent to **R**:

$$I \rightarrow R : HDR, \{prf(K_a, N_I, N_R)\} K_e$$

After three messages, a shared session key $K_{Session} = prf(K_d, N_R, N_I)$ can be generated in two entities by employing the deriving key K_d and two other nonce.

It is notable that in IPsec, communication is protected by session keys. However, the security of a session key is guaranteed by the long-term shared keys K_d , K_a and K_e . Therefore, once the long-term keys (SA) are compromised, all QUICK mode negotiations protected by SA are disclosed. The security of IPsec is under threat. As Perlman and Kaufman [PeKa01] indicated, IPsec is vulnerable to dictionary attack, due to the pre-shared (in Phase I) long-term keys, and Ornaghi and Valleri [OrVa03] demonstrated it in a BlackHat conference.

Moreover, in IPsec Phase I, the long-term shared secrets (keys) evolve into key exchange protocol to generate session keys for Phase II. According to information entropy [Gr90], 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

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that provide security and data integrity for unicast communications over insecure networks. SSL protocol was originally developed by Netscape¹⁴. When version 2.0 was released it contained a number of security flaws which ultimately led to the design of SSL version 3.0. SSL is the basis for TLS version 1.0. [Ka04]. Many leading financial institutions have endorsed SSL for commerce over the Internet such as Visa¹⁵, MasterCard¹⁶ and American Express¹⁷.

SSL protocol allows mutual authentication between two entities. It also allows both entities to establish an encrypted connection, which requires all information sent between the entities to be encrypted in order to provide a high degree of confidentiality. SSL uses a combination of asymmetric and symmetric cryptography. An asymmetric key is used to perform mutual authentication and a symmetric key is used to secure communication. SSL/TLS consist of the following protocols [DiRe08, FrKaKo96]:

- Handshake protocol is used to perform authentication and key exchanges.
- Change cipher spec protocol is used to indicate that subsequent information will be protected under the agreement and keys.
- Alert protocol is used for signalling errors and session closure.

¹⁴ Netscape Communications Corporation is commonly known as Netscape. It is an US computer services company best known for its web browser.

¹⁵ http://usa.visa.com/merchants/risk_management/online_transaction_safet.html

¹⁶ <http://www.creditcardassist.com/mastercard/creditcards.html>

¹⁷ http://www.americanexpress.com/uk/legal/cs_security.shtml

- Application data protocol transmits and receives scrambled information.

As discussed in Section 2.1.1 (cryptographic systems), key exchange is important in determining the security of a protocol. Therefore, we study handshake protocol in SSL/TLS in detail. When a SSL client (**C**) and server (**S**) first start communicating, they agree on a protocol version, select cryptographic algorithms, optionally authenticate each other, and use public-key encryption techniques to generate shared secrets. These processes are performed in the handshake protocol, which can be summarized as follows:

Suppose a pre-master secret, K_{master} , is generated by the client and encrypted under the public key of the server, and sent the result to the server. By employing asymmetric keys, the pre-master secret, K_{master} , is thereby shared between two entities for securing the following messages.

- C** sends a client hello message to which **S** must respond with a server hello message. The client hello and server hello consists the following attributes: Protocol Version, Session ID, Cipher Suite, Compression Method and a checksum appended to messages and used to verify that the message contents have not been tampered. Additionally, two random values are generated and exchanged: ClientHello random (rnd_c) and ServerHello random(rnd_s).
- S** sends its certificate to **C** which is used to verify with a certification authority (CA). Following the certificate, **S** requests the certificate of **C**. **S** then sends the server hello done message, indicating that the hello-message phase of the handshake is complete.

- iii) After successful verification, **C** sends a response to **S** using asymmetric cryptography.
- iv) The key exchange message is now sent, and the content of that message depends on the public key algorithm selected between the client hello and the server hello.
- v) At this point, the handshake is complete and the **C** and **S** may begin to exchange application layer data.

After the handshake protocol, a session key is able to be produced by **S** and **C**, $K_{session} = h(K_{master}, h(K_{master}, rnd_C, rnd_S))$ by combining the master secret and two random numbers.

It is observable that the pre-shared master secret, K_{master} , is distributed by public key systems. As discussed in Section 2.1.1 (asymmetric cryptography), all asymmetric schemes are susceptible to brute force key search attack, which makes K_{master} vulnerable. In addition, all session keys are generated from K_{master} and the protection against tampering with the SSL handshake protocol relies heavily on the secrecy of K_{master} . That the master secret remains truly secret is important to the security of SSL/TLS. However, in the protocol design, the usage of K_{master} involves multiple phases, such as certificate verify, finished and change cipher spec [WaSc96].

On the top of the above concerns, the SSL/TLS protocols suffer from different types of flaws [MiBrLa02]: 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 [Ba04, WaSc96] state, an attacker can use

plaintext attacks to break SSL/TLS protocols due to the long-term shared identical cryptographic keys.

SRTP

SRTP, a profile of the Real-time Transport Protocol (RTP) [ScCaFr03], is a secure real-time protocol designed to protect sensitive information in the form of multimedia (such as video and voice) published in 2004. Its inventors claim that SRTP achieves high throughput and low packet expansion, and provides suitable protection for heterogeneous environments (a hybrid of wired and wireless networks). They also point out that IPsec or SSL/TLS could be used to protect RTP, but that these protocols lack dynamic allocation of sessions and do not address the need for an asymmetric cryptosystem. SRTP was developed to overcome these problems.

Before using SRTP to exchange any media, cryptographic keys need to be exchanged. SRTP relies on an external key management protocol to set up the initial master key. Two protocols specifically designed to be used with SRTP are ZRTP (published in January 2009 [ZiJoCa09]) and MIKEY (released in 2004 [ArCaLi04]). Both provide the necessary keying material and management mechanisms to maintain the security of multimedia sessions.

SRTP uses two types of keys, session keys and master keys, to secure multimedia communications. The master keys and other key materials in the cryptographic context are provided by key management protocols (ZRTP and MIKEY) external to SRTP. The session keys are derived in a cryptographically-secure way from the master keys. SRTP also requires a native derivation algorithm to generate session keys to secure the communication. The security of SRTP therefore relies on the security of the SRTP key

derivation algorithm and the master keys. Suppose the master keys K_{master} and other keying materials K_{master_salt} are secure in SRTP. The SRTP key derivation is illustrated in Figure 2.2.

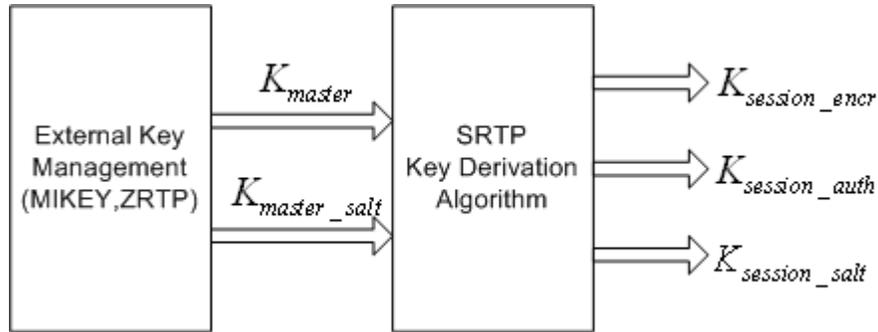


Figure 2.2. SRTP Session Key Derivation.

SRTP uses a secure pseudorandom function to generate encryption ($K_{session_encr}$), authentication ($K_{session_auth}$) and salt ($K_{session_salt}$) session keys from the master key and master salt. Session key derivation involves an 8-bit label (for example, 0x00, 0x01 and 0x02 labels for generating encryption, authentication and salt session keys respectively), master salt and other keying materials. If $x = (label, keying\ materials) \oplus K_{master_salt}$, then the session keys are generated as $prf(K_{master}, x)$.

As [GuSh07] point out, the security of the stream cipher-like encryption used in SRTP depends critically on the keystream never repeating. This is also emphasized several times in the specification [OrMcBa04]. Therefore, the master key and master salt must be unique in each session in order to produce unique session keys. But, according to the Gupta and Shmatikov study, “if the attacker ever succeeds in tricking an SRTP session into re-using previously used key material, the master key will

repeat". Under these circumstances, the confidentiality of sensitive information in multimedia form will have been breached.

In addition, as discussed in IPsec and SSL/TLS, the master key is involved in session key derivation. For multiple sessions and with the same sender involved, the session key may repeat. An adversary may consequently be able to reveal other session contexts. Also, by capturing enough packets and applying cryptanalysis, the adversary is able to breach SRTP. According to information entropy, the uncertainty of the master key will reach zero as it is used to generate session keys, since it participates in session key derivation.

This section has reviewed and discussed the security of unicast communication in sensitive information systems. By investigating - IPsec, SSL/TLS and SRTP - the most secure and widely deployed unicast communication protocols, we found that among these protocols, a common security drawback is the use of long-term shared keys. In the next section we will examine the security of multicast communication in protecting *communication channel*.

2.2.2. 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. Almost all the schemes discussed below use the notion of a central trusted authority, called a group controller (GC). The GC is used

to generate, distribute and update cryptographic keying material for group members to ensure multicast security through access control, data confidentiality and group authentication.

Historically, the first use of group keys was in the Second World War. Group keys were sent to groups of agents by the Special Operations Executive. These group keys allowed all the agents in a particular group to receive a single coded message [Ma99].

Modern group key management 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 [KiPeTs04].

In order to fulfill these requirements, substantial research work has been carried out over the last decade. Projects include Conference Key Distribution Systems (CKDS) [InTaWo82], Scalable Multicast Key Distribution [Ba96], Group Key Management Protocol (GKMP) [HaMu97a, HaMu97b], Logical Key Hierarchy (LKH) [WoGoLa98, WoGoLa00], Kronos [SeKoJa00], Distributed Logical Key Hierarchy [OhKeDa00], One-way Function Tree (OFT) [ShMc03], Contributory Key Agreement [KiPeTs04], VersaKey [WaCaSu99], Iolus [Mi97] and CLIQUES [StTsWa98]. Based on the way in which the group key is formed, these group key management systems can be classified into three approaches: contributory (distributed) key agreement, centralised key distribution and decentralized key distribution.

Contributory Key Agreement

Contributory key agreement (also called distributed key agreement) generates a group key via all group members' uniform contributions. These protocols are resilient to many types of attacks and are particularly appropriate for relatively small collaborative peer groups. Unlike most group key distribution protocols, contributory group key agreement protocols offer strong security properties such as key independence and perfect forward secrecy.

The first contributory key agreement proposal, CKDS (also known as ING), was developed by Ingemarsson et al., and based on a public key system (the Diffie-Hellman key exchange protocol [DiHe76a]). This was followed by IDCKD [KoOh87], STR protocol [StStDi88], Octopus Protocol [BeWi98], Group Diffie-Hellman (GDH) key exchange schemes [StTsWa96, StTsWa00] and Tree-based Group DH Key Management (TGDH) [KiPeTs04]. Zou and Ramamurthy [ZoRaMa05] point out that these agreements are all primarily different variations of n -party DH key exchange. We select the two most influential and prominent agreements, CKDS (ING) and GDH, for further investigation.

CKDS adopts the public key distribution system invented by Diffie and Hellman to generate a conference key for any group of stations to share in order to guarantee information security in multicast communication systems. CKDS consists of $n - 1$ rounds; group members are arranged in a cycle (Figure 2.3) and perform every round in synchronization. CKDS is illustrated in Figure 2.3.

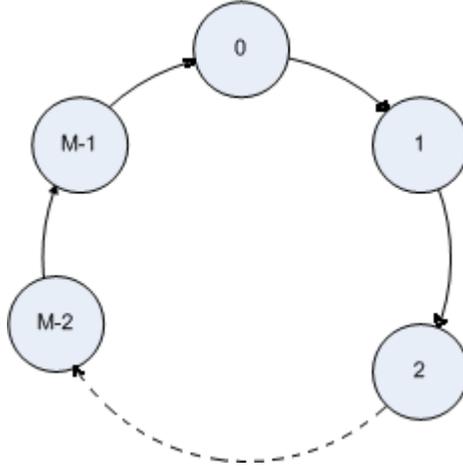


Figure 2.3. CKDS (ING) Protocol.

The net key (group key) can be generated as $m_i \xrightarrow{\alpha^{R_1 \dots R_i}} m_{i+1}$ for group members, where $\xrightarrow{\quad}$ indicates the sequential sending message in a cycle, and α is a primitive element in the field of integers modulo a prime number p . For example, if there are four members in a group, the net key can be generated in three rounds as follows:

- i) Each member generates a random number R_i and computes $\alpha^{R_i} \text{ mod } p$ and passes it:

$$m_i \xrightarrow{\alpha^{R_i} \text{ mod } p} m_{i+1}$$

The state of this round can be instantiated as:

$$m_0 \rightarrow m_1 : \alpha^{R_0} \text{ mod } p, m_1 \rightarrow m_2 : \alpha^{R_1} \text{ mod } p, \dots, m_3 \rightarrow m_0 : \alpha^{R_3} \text{ mod } p$$

- ii) After received the intermediate key, each member uses the key to compute a new value with the own $\alpha^{R_i} \text{ mod } p$. The state of this round is:

$$m_0 \rightarrow m_1 : \alpha^{R_0 R_3} \text{ mod } p, m_1 \rightarrow m_2 : \alpha^{R_1 R_0} \text{ mod } p, \dots, m_3 \rightarrow m_0 : \alpha^{R_3 R_2} \text{ mod } p$$

- iii) In the last round, each member repeatedly uses the received key with its own key to compute a new intermediate value. The state of the final round is:

$$m_0 \rightarrow m_1 : \alpha^{R_0 R_2 R_3} \bmod p, m_1 \rightarrow m_2 : \alpha^{R_1 R_0 R_3} \bmod p, \dots, m_3 \rightarrow m_0 : \alpha^{R_3 R_1 R_2} \bmod p$$

Therefore, after the final round, every member is in possession of the net key $\alpha^{R_0 R_2 R_3 R_4} \bmod p$. From the example, it is observable that each member in the CKDS starts synchronously, and requires $n - 1$ rounds to compute a net key. However, CKDS does not support dynamic membership operations, such as member join and leave, and has a high computational cost due to the $n - 1$ sequential modular exponentiations. In addition, the protocol falls into the class of natural DH extensions as defined in [StTsWa96]. Because the protocol has no natural group leader, it is difficult to use it as a foundation for auxiliary key agreement protocols [StTsWa00]. Furthermore, the CKDS only performs key distribution without authentication. Thus, the security of CKDS is breakable.

GDH protocol consists of three versions of the Group Diffie-Hellman key exchange schemes (GDH.1, GDH.2 and GDH.3) proposed by Steiner et al. in 1996 and 2000 [StTsWa96, StTsWa00]. The key generated by all three versions for n group members is $K = \alpha^{R_0 \cdot R_{n-1}}$, where α is a prime number, and R_i is a random number of member i .

I. **GDH.1** involves two stages ($n - 1$ rounds each): upflow and downflow. The upflow stage collects contributions from all group members. The downflow stage computes intermediate values and forwards them. The group key distribution protocol is defined as:

Stage 1 (Upflow): Round $i; i \in [1, n - 1]$

$$m_i \rightarrow m_{i+1} : \{\alpha^{\prod_{k \in [1, j]} R_k} \mid j \in [1, i]\}$$

Stage 2 (Downflow): Round $n - 1 + i; i \in [1, n - 1]$

$$m_{n-i} \rightarrow m_{n-i+1} : \{\alpha^{\prod_{k \in [j, n-i+1]} R_k} \mid j \in [1, n-i+1]\}$$

Finally, every member is in possession of the group key $K = \alpha^{R_0 \cdots R_{n-1}}$. Figure 2.4 illustrates an example of GDH.1 with four members.

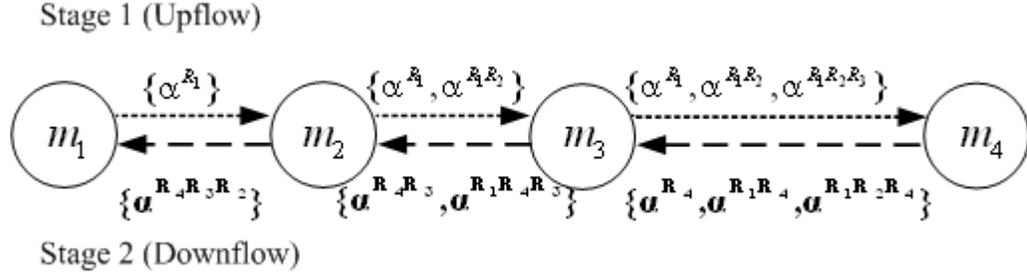


Figure 2.4. GDH.1: An Example for Four Members.

It is notable that GDH.1 has $2(n-1)$ rounds to compute the group key with total exponentiations $\frac{(n+3)n}{2} - 1$. Authors claim that the drawback of GDH.1 is its relatively large number of rounds. But on the other hand, GDH.1 protocol does not impose the special communication requirements, such as multicast, broadcast or synchronization, that CKDS does.

II. **GDH.2** reduces the number of rounds by collecting contributions from all members in upflow but broadcasts messages in stage 2. In the first stage, the upflow protocol in GDH.1 is modified by adding cardinal value $\alpha^{R_1 \cdots R_i}$. By the time the upflow reaches m_n , the cardinal value becomes $\alpha^{R_1 \cdots R_{n-1}}$. Therefore, the group member m_n is the first to compute the group key. Group member m_n also computes the last batch of intermediate values and broadcasts these to other members. The group key distribution protocol is defined as:

Stage 1 (Upflow): Round i ; $i \in [1, n-1]$

$$m_i \rightarrow m_{i+1} : \{\alpha^{\prod (R_k | K \in [1, i] \wedge K \neq j)} \mid j \in [1, i+1]\}, \alpha^{R_1 \dots R_i}$$

Stage 2 (Broadcast): Round n

$$m_n \Rightarrow m_i : \{\alpha^{\prod (R_k | k \neq i)} \mid i \in [1, n]\}$$

Finally, every member is in possession of group key $K = \alpha^{R_0 \dots R_{n-1}}$. Figure 2.5 illustrates an example of GDH.2 with four members.

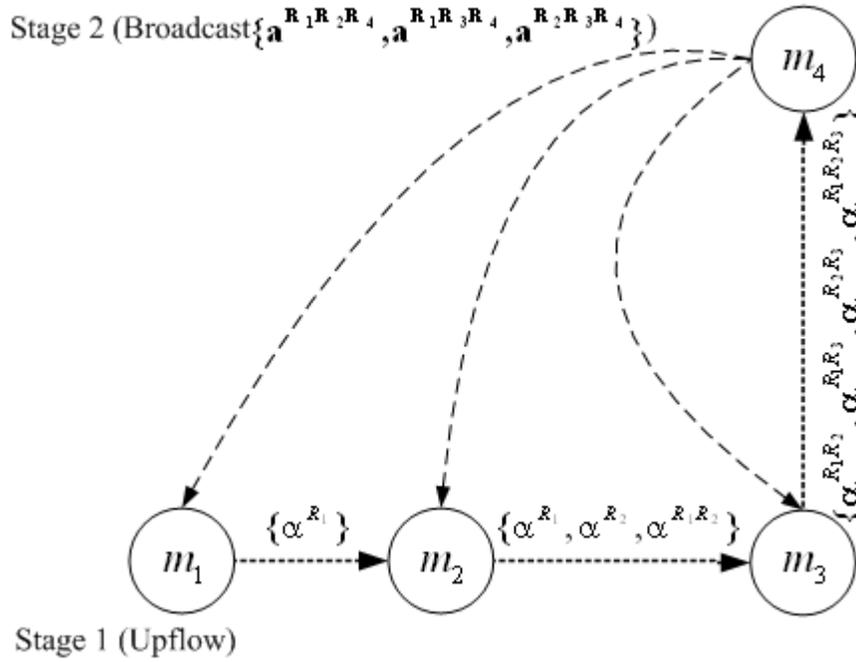


Figure 2.5. GDH.2: An Example of Four Members.

It can be observed from the definition and the example that GDH.2 has n rounds which has reduced $n-1$ rounds from GDH.1. But the total number of exponentiations remains the same as GDH.1.

III. **GDH.3** reduces the number of exponentiations and involves four stages. The first stage is similar to the upflow stage in GDH.1 and GDH.2 to collect contributions from all members with $n-2$ rounds. In the second stage, m_{n-1} broadcasts the

processed intermediate key obtained from stage 1 to all other members. In the third stage, every member extracts its own share R_i and sends the result to m_{n-1} . Finally, the fourth stage, m_{n-1} broadcasts recomputed values to members. After receiving values from m_{n-1} , every member can compute the group key $K = \alpha^{R_0 \cdots R_{n-1}}$. The group key distribution protocol is defined as:

Stage 1 (Upflow): Round $i; i \in [1, n-2]$

$$m_i \rightarrow m_{i+1} : \{ \alpha^{\prod (R_K | K \in [1, i])} \}$$

Stage 2 (Broadcast): Round $n-1$

$$m_{n-1} \Rightarrow m_i : \{ \alpha^{\prod (R_K | k \in [1, n-1])} \}$$

Stage 3 (Response): Round n

$$m_i \rightarrow m_n : \{ \alpha^{\prod (R_K | K \in [1, n-1] \wedge K \neq i)} \}$$

Stage 4 (Broadcast): Round $n+1$

$$m_n \Rightarrow m_i : \{ \alpha^{\prod (R_K | k \in [1, n] \wedge k \neq i)} \mid i \in [1, n-1] \}$$

Finally, every member is in possession of group key $K = \alpha^{R_0 \cdots R_{n-1}}$. Figure 2.6 illustrates an example of GDH.3 with four members.

It can be observed from the definition and the example that GDH.3 has $n+1$ rounds (that is, it has increased one round from GDH.2). But the total number of exponentiations has reduced to $5n-6$ from $\frac{(n+3)n}{2}-1$ in GDH.1 and GDH.2 (as its authors claimed).

In addition to group key generation, all three versions of GDH provide rekeying operations. This is the process of changing the group key and supporting keys and

sending them to group members. The group key must be updated when membership changes due to join or leave operations. The purpose of key updating is to enforce backward and forward secrecy.

IV. GDH Member Join. When a new member, m_{n+1} , wants to join a group, first m_n generates a new random number, R'_n , and computes $\alpha^{R'_n}$, and then sends the results to m_{n+1} . After that, m_{n+1} generates its own exponent $\alpha^{R_{n+1}}$ and computes the new group key. Finally, as in the normal protocol run, m_{n+1} broadcasts n intermediate keys to other group members. The purpose of this rekeying operation is to guarantee that GDH satisfies backward secrecy. Backward secrecy prevents new members from gaining the old group key and consequently accessing past group communication data.

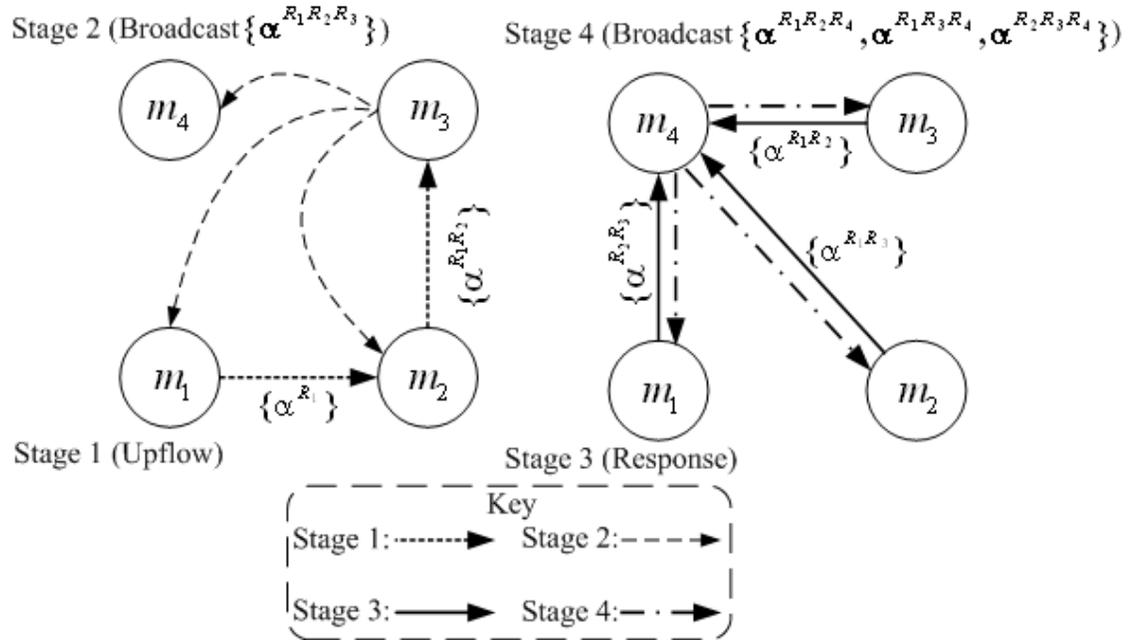


Figure 2.6. GDH.3: An Example of Four Members.

V. GDH Member Leave. When a member, m_p , wishes to leave a group, m_n first generates a new component α^{R_n} and computes $n - 2$ intermediate keys, excluding keying materials of m_p . Member m_n then broadcasts this key material to other members. Note that, since intermediate keys do not contain information for m_p , the excluded m_p is unable to compute the new group key. The purpose of this rekeying operation is to guarantee that GDH satisfies forward secrecy. Forward secrecy prevents leaving members from accessing future group communication.

Contributory Key Agreement Summary. By investigating the contributory key agreement in protecting multicast *communication channel*, and using CKDS and GDH as representatives, a comparative summary of CKDS, GDH.1, GDH.2 and GDH.3 is given in Table 2.2.

Table 2.2. Comparison of CKDS, GDH.1, GDH.2 and GDH.3.

Protocol	CKDS	GDH.1	GDH.2	GDH.3
Rounds	$n - 1$	$2(n - 1)$	n	$n + 1$
Message Sent per m_i	$n - 1$	2	1	2
Message Received per m_i	$n - 1$	2	2	3
Total EXPs	n^2	$\frac{(n + 3)n}{2} - 1$	$\frac{(n + 3)n}{2} - 1$	$5n - 6$
Special Member	no	no	m_{n-1}	m_{n-1} and m_{n-2}
Synchronization	yes	no		
Rekeying Operation	no	yes		
Group Key	$\alpha^{R_1 \dots R_{n-1}}$			

It is notable that the security of contributory key agreements relies on the security of nondeterministic polynomial time (NP) problems [TaWe06]. It is also notable that contributory key agreements do not employ a Group Controller (GC) to manage

rekeying operations. From the table, it can be seen that GDH.3 has the best performance. However, GDH.3 depends on a special member to perform rekeying. Therefore, it is not suitable for large groups.

Centralised Group Key Management

Centralised key distribution requires a GC (or a key distribution centre (KDC)) to generate the group key and distribute the key to all group members. The earliest centralised key distribution was a star-shaped scheme in which all group members adopted their own secret keys to encrypt the group key with the GC when a rekeying event occurred. This process was inefficient in terms of communication, although it provided both forward and backward confidentiality and ease of implementation. According to Zou [ZoRaMa05], “among all group key management protocols, the key tree scheme provides a very powerful approach in centralised group key management”.

Tree-based schemes have been independently proposed by several group researchers. The first such scheme was the logical key hierarchy (LKH) proposed by Wallner et al. [WaHaAg97, WoGoLa98]. A scheme similar to LKH was proposed by Caronni et al. [CaWaSu98] and Noubir [No98]. A more efficient scheme than LKH, based on the idea of a one-way function tree (OFT), was proposed by Sherman et al. [ShMc03]. Among the centralised key distribution schemes, LKH and OFT are the most popular.

LKH. In the LKH scheme, the associated binary tree is called a key tree. It is a virtual tree (see Figure 2.7). As discussed early, the group controller (GC) maintains the multicast group in the key tree. All members of the group are associated with leaf nodes of the tree. The nodes in the tree are assigned keys. The key assigned to the root

node is called Traffic Encryption Key (TEK). The key assigned to the other nodes is called the Key Encryption Key (KEK).

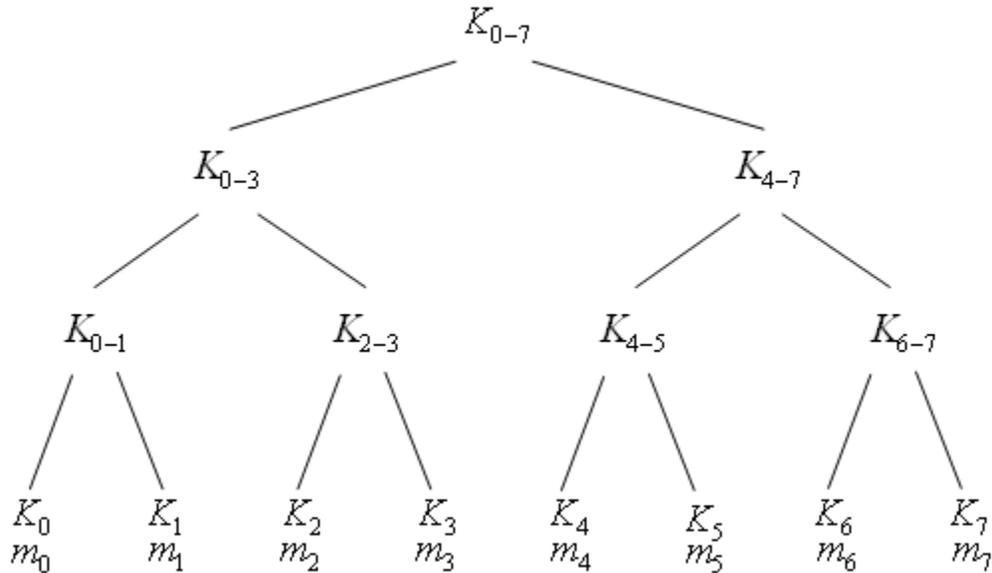


Figure 2.7. LKH Key Tree.

Each member in the key tree can recognise keys if there is a path from the member to the root. For example, m_3 knows the set of keys $\{K_3, K_{2-3}, K_{0-3}, K_{0-7}\}$. Also, each member holds a shared unique KEK (for example, m_3 holds K_3), only known between the member and the GC (KDC). The GC generates new group keys and distributes these to all group members when a member joins or leaves.

I. **LKH Member Join.** Suppose a current group consists of members m_0 to m_6 , and that member m_7 is about to join the group (see Figure 2.8). After m_7 is authenticated by the GC, the GC, to ensure backward secrecy, decides the location in the tree for m_7 and updates all the keys from the parent of the joining member m_7 to the root for backward secrecy. First, the GC generates a new set of keys

$\{K'_{6-7}, K'_{4-7}, K'_{0-7}\}$ and unicasts these to m_7 . The GC then multicast internal keys to other affected members. The rekeying messages are:

$$GC \rightarrow m_7 : \{K'_{6-7}, K'_{4-7}, K'_{0-7}\} K_7$$

$$GC \Rightarrow \{m_6\} : \{K'_{6-7}, K'_{4-7}, K'_{0-7}\} K_{6-7}$$

$$GC \Rightarrow \{m_4, m_5\} : \{K'_{4-7}, K'_{0-7}\} K_{4-7}$$

$$GC \Rightarrow \{m_1, m_2, m_3, m_4\} : \{K'_{0-7}\} K_{0-3}$$

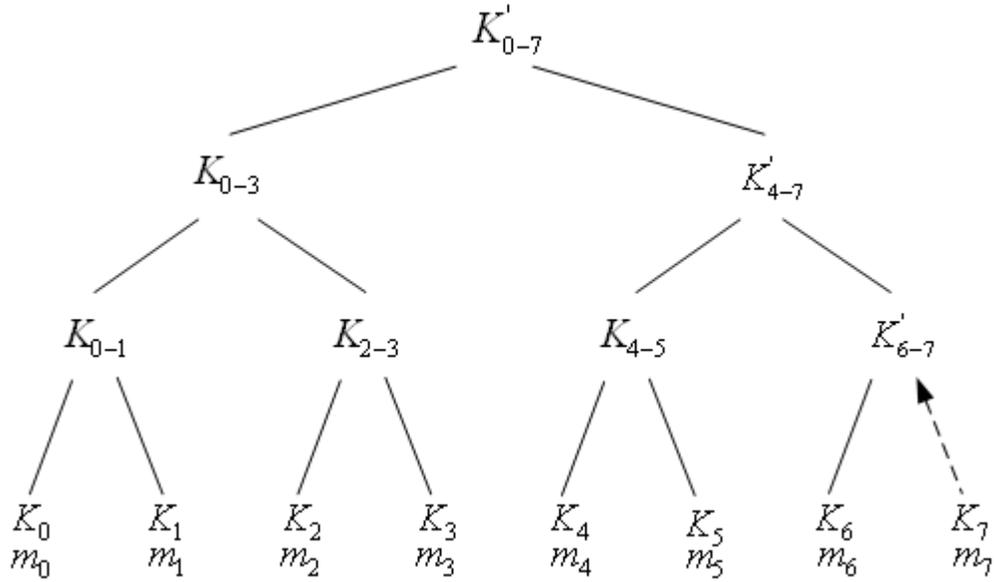


Figure 2.8. LKH Member Joins the Group.

In addition, suppose that m_8 is to join the group and that all slots are occupied.

In this situation, node m_0 becomes an internal node, m_{0-8} . A new level is thus established and more members can be allocated.

II. LKH Member Leave. When a member m_3 wishes to leave the group, all the keys that m_3 knows and shares with other members need to be changed to ensure

forward secrecy. The GC changes these keys from the bottom up. Figure 2.9 illustrates the rekeying operation.

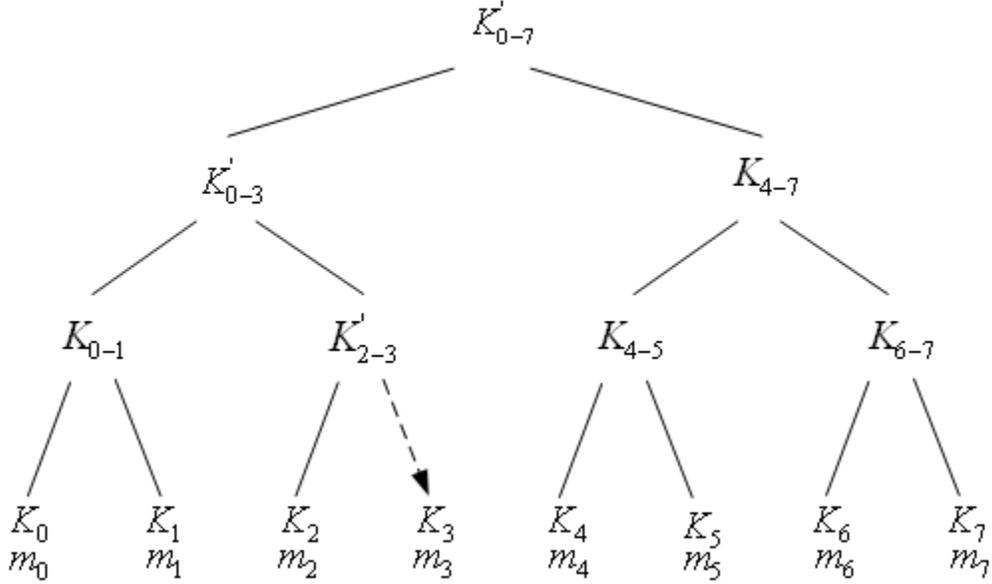


Figure 2.9. LKH Member Leaves the Group.

Because m_3 knows the set of keys $\{K_{2-3}, K_{0-3}, K_{0-7}\}$, these keys need to be regenerated by the GC and multicast to affected members. The rekeying messages are:

$$GC \Rightarrow \{m_2\} : \{K'_{2-3}, K'_{0-3}, K'_{0-7}\} K_{2-3}$$

$$GC \Rightarrow \{m_0, m_1\} : \{K'_{0-3}, K'_{0-7}\} K_{0-3}$$

$$GC \Rightarrow \{m_4, m_5, m_6, m_7\} : \{K'_{0-7}\} K_{4-7}$$

It is observable that the computational cost of the rekeying operation is logarithmic to the size of the group. Thus, the number of keys that needs to be changed in member join and leave operations is $\log(n)$, where n is the number of members in the multicast group. Also, this scheme requires a reliable multicast infrastructure, and it is scalable

for large group size. The LKH has been slightly improved in terms of performance by the VersaKey framework [WaCaSu99] and LKH+ [HaHa99].

OFT. In a one-way function tree scheme, the associated binary tree is the same as for LKH. The scheme assumes that there is a secure unicast channel between the GC and each group member. The main advantage of OFT over LKH is that it allows group members to compute group keys locally to reduce the communication and computation cost.

In an OFT, the GC maintains a logical key tree and the group members are assigned at leaf nodes (Figure 2.10). Each node associates with multiple keys; the KEK is a shared node secret K_i (an unblinded node key) between group members and the GC while the blinded node secret is the result of applying $K_{i_bk} = g(K_i)$, where $g(\cdot)$ is a one-way function. The node secret of the root is the group key, TEK. The GC computes the node secrets and blind node keys of all nodes in a bottom-up manner, beginning with the leaf nodes, level by level up to the root, by applying a mixing function $f(\cdot)$ (for example, XOR).

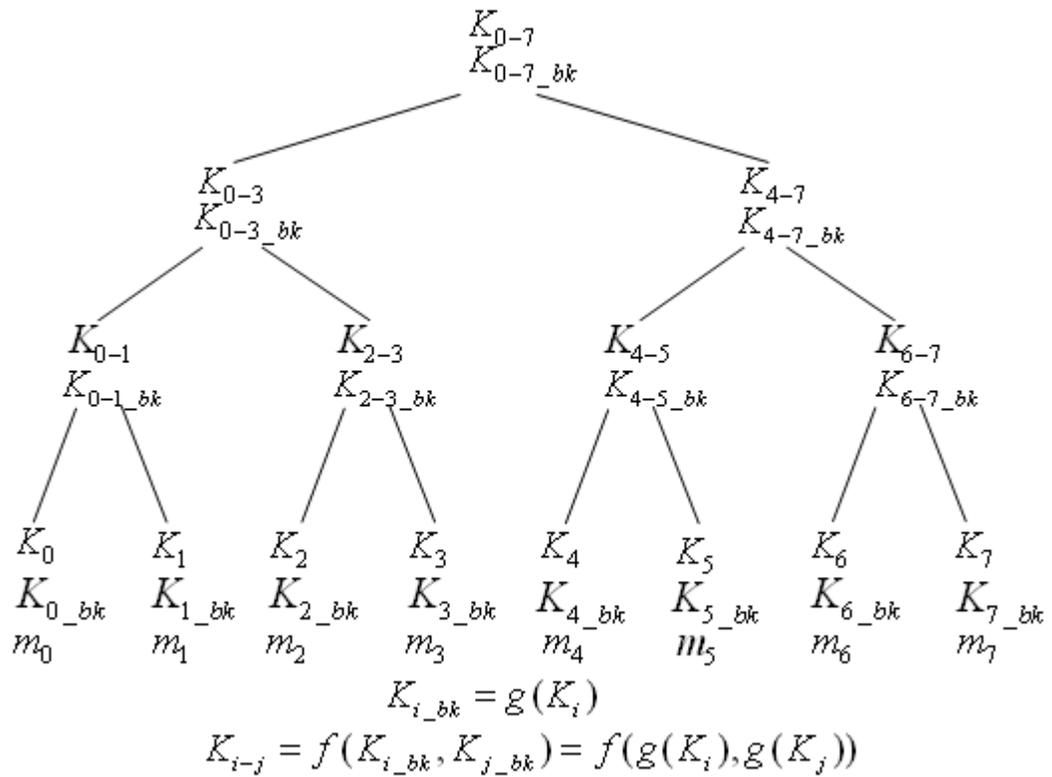


Figure 2.10. OFT Key Tree.

The security of the system depends on the fact that each member knows the unblinded node keys on the path from its node to the root, and knows the blinded node keys that are siblings on its path to the root, and no other blinded nor unblinded keys. The purpose of blinded and unblinded keys is to allow group members to compute higher-level keys from lower-level keys in order to reduce the number of rekeying operations conducted by the GC. Figure 2.11 shows an example of an OFT key tree, highlighting the blinded and unblinded keys that are known to a particular group member m_3 .

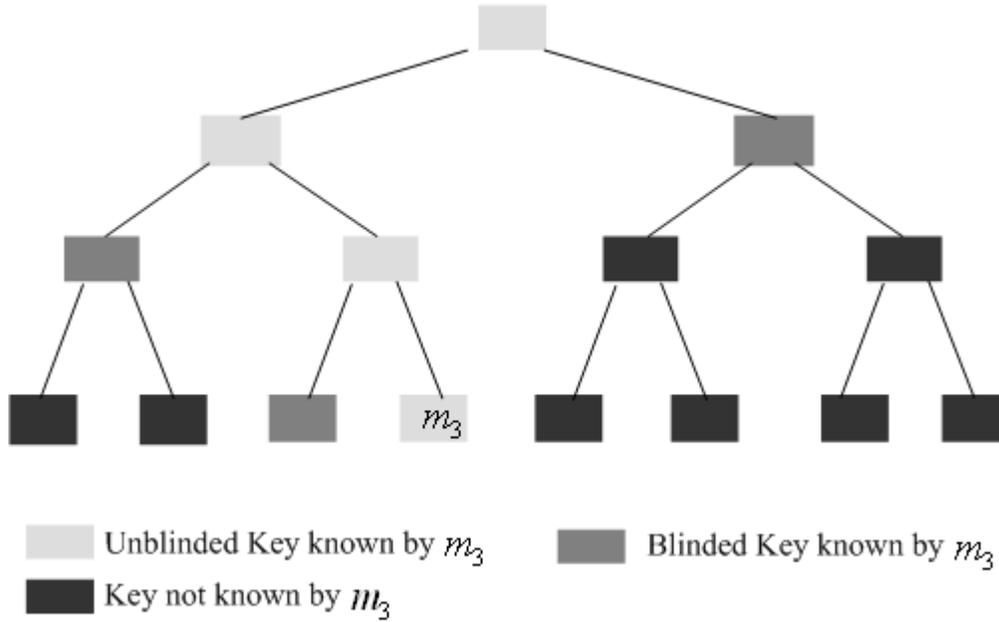


Figure 2.11. OFT The Keys Known to a Group Member.

- I. **OFT Member Join.** Suppose a current group consists of members m_0 to m_6 , and that member m_7 is about to join the group (see Figure 2.12). After authentication with the GC, the GC sends the blinded keys to m_7 that it is supposed to know:

$$GC \rightarrow m_7 : \{K_{6_bk}, K_{4-5_bk}, K_{0-3_bk}\} K_7$$

The GC then sends the blinded key of the new member, m_7 , to the member with the same parent node:

$$GC \rightarrow m_6 : \{K_{7_bk}\} K_6$$

Last, the GC broadcasts all changed blinded node keys to other affected members to ensure backward secrecy:

$$GC \Rightarrow \{m_4, m_5\} : \{K'_{6-7_bk}\} K_{4-5}$$

$$GC \Rightarrow \{m_0, m_1, m_2, m_3\} : \{K'_{4-7_bk}\} K_{0-3}$$

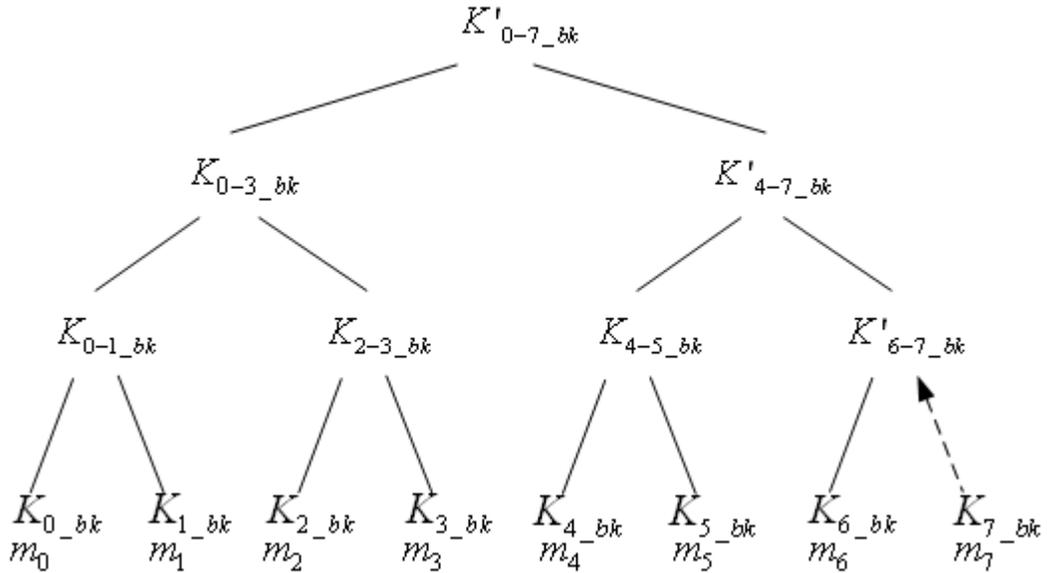


Figure 2.12. OFT Member Join a Group.

Each member in the group is therefore able to recompute the new group key from itself to the root.

- II. **OFT Member Leave.** Suppose that the member associated with the leaf m_3 wants to leave the group. When the member associated with the leaf m_3 leaves the group (Figure 2.13), the member assigned to the sibling of m_3 is reassigned to the parent of m_3 and given a new leaf key value. The new values of the blinded node keys that have changed are broadcast securely to the appropriate subgroups in order to ensure all current group members can recompute any keys that have changed as a result of the membership change. The message sent for rekeying operation can be briefly given:

$$GC \rightarrow m_2 : \{K'_{3_{bk}}\}K_2$$

$$GC \Rightarrow \{m_0, m_1\} : \{K'_{2-3_{bk}}\}K_{0-1}$$

$$GC \Rightarrow \{m_4, m_5, m_6, m_7\} : \{K'_{0-3_bk}\} K_{4-7}$$

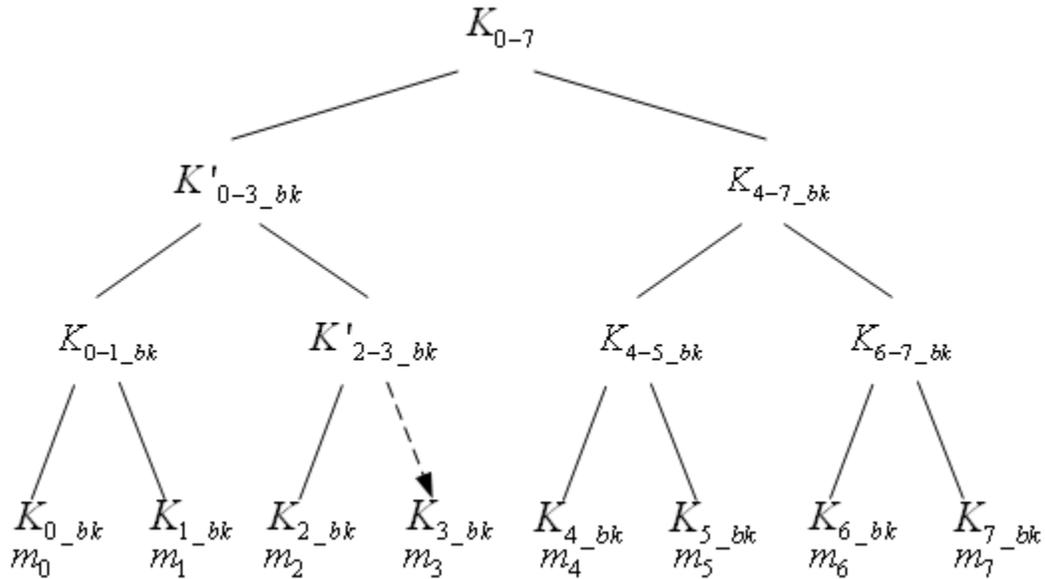


Figure 2.13. OFT Member Leave a Group.

The process requires the GC to multicast $\log_2 n$ (balanced tree) key updates, which is the height of the key tree. The major contribution of the OFT scheme is that it allows members to compute keys locally in order to reduce rekeying complexities. However, according to Horng and Ku et al.[Ho02, KuCh03], OFT is vulnerable to collusion attacks.

Centralised Key Distribution Summary. The two centralised key distribution schemes discussed, OFT and LKH, achieve similar performance; however, OFT offers two advantages:

- OFT reduces computation costs by allowing group members to perform local calculations to derive higher level keys.
- OFT reduces the number of messages required on a key update to $\log_2 n$.

Table 2.3 outlines the characteristics of the LKH and OFT schemes. OFT slightly outperforms LKH, but in terms of security, OFT is constrained by the limitations of its original design. Also, as all centralized key distribution schemes employ a group controller to recompute new group keys for members, this can pose a challenge for large groups with frequent rekeying operations. Moreover, single-point failure is the biggest drawback for such schemes in the case of a GC failure.

Table 2.3. Comparison of LKH and OFT.

Scheme	LKH	OFT
Number of Member Keys	$\log_2 n$	$2 \log_2 n$
Number of Rekeying Message	$2 \log_2 n$	$\log_2 n$
Computation Cost	$O((\log_2 n)^2)$	$O(\log_2 n)$
Vulnerable to Collusion	no	yes
Group Controller	yes	yes

Decentralized Group Key Management

Decentralized group key management is used to minimize the problems of centralized key distribution schemes, such as single-point failure. Under a decentralized system, key management is divided among hierarchically-structured managers, each with a small subgroup controller, in an attempt to concentrate the work in a single entity.

The earliest solution for decentralized key distribution was a core-based tree [Ba96]. Further research followed, such as Iolus [Mi97], MARKS [Br99], Kronos [SeKoJa00] and IGKMP [HaCaMo00]. Among these, the Iolus architecture is the most referenced scheme [ZoRaMa05]. It employs a multilayered management structure, and the scheme is discussed in details in the following:

Iolus is a high-level infrastructure for secure multicast. It divides a large group into a number of subgroups. When a member joins or leaves a group, only the key of the

subgroup to which the member belongs needs to be changed, while the keys of all other subgroups remain the same. Iolus relies on relay nodes for rekeying operations. The architecture is illustrated in Figure 2.14.

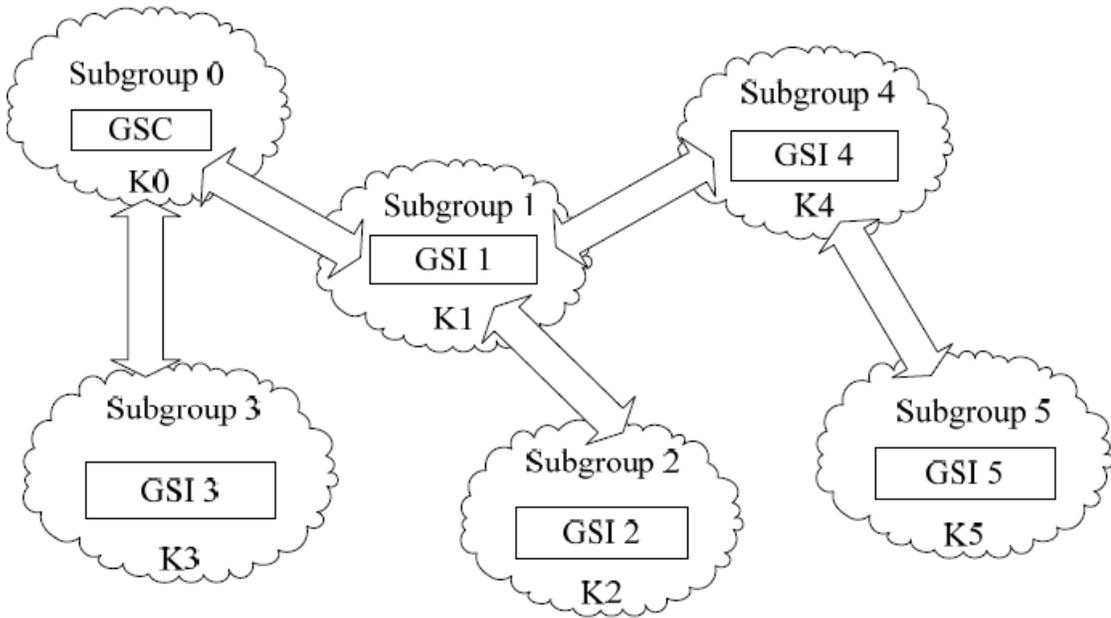


Figure 2.14. Subgroups and GSIs in Iolus Scheme.

In Iolus, each subgroup has a controller named the group security agent (GSA) or group security intermediates (GSI). Every subgroup has its own independent key κ_i . The GSI of the root subgroup is called the group security controller (GSC). A GSI is a bridge between its parent subgroup and its own subgroup, and it holds both subgroup keys (for example, GSI 2 holds κ_1 and κ_2). Because each subgroup uses a different key, the GSIs are responsible for translating data from one key to another and delivering it to other GSIs as appropriate.

I. **Iolus Member Join.** When a member wants to join the group, the GSI generates a new subgroup key and sends it to the member via a secure unicast channel. Then the GSI joins the next highest subgroup in the hierarchy.

II. **Iolus Member Leave.** When a member leaves the group, the GSI generates a new subgroup key and distributes the key to all remaining subgroup members. Should the member be the only member in the GSI, the GSI needs to contact its parent subgroup and remove itself from the secure distribution tree.

The major benefits of using a secure distribution tree are twofold. First, this architecture localizes the effect of group membership changes to one subgroup. Second, it overcomes the single-point failure effect. Should one GSI experience a system failure or security breach, only the breached subgroup loses services. The other subgroups continue to function.

Despite its advantages, Iolus suffers from several drawbacks [MoRaRo99]. The Mayer and Rao survey points out that Iolus requires a substantial resource overhead to manage a multicast group. Also, if the GSC fails, many of subgroups are cut off from each other. Decentralized key distribution schemes, such as Iolus, are therefore not suitable for large groups. In addition, the performance of such schemes is a challenge for multicast communication.

This section (2.2.2) has reviewed and discussed the security of multicast communication in sensitive information systems. By investigating the existing approaches (contributory, centralized and decentralized key agreement), we found that each key agreement has its own advantages and disadvantage (Table 2.4), and none of them is fulfilled the security requirements of sensitive information systems, such as privacy protection. In the next section, we will summarize the security threats and concerns of existing approaches to protect communication channel in sensitive information systems.

Table 2.4. Advantages and Disadvantages of Multicast Communication Schemes.

	Contributory key agreement	Centralized key distribution	Decentralized key distribution
Advantages	does not require a GC	large group orientation	membership change does not affect the entire group
		scalable	
	strongly collusion resistant	collusion resistant	operationally efficient
		operationally efficient	
Disadvantages	not scalable	likelihood of single point failure	communication expensive
	small group orientation; not suitable for a large group	single membership change affects the entire group	weak collusion resistance
	single membership change affects the entire group		
	no consideration of privacy protection	no consideration of privacy protection	no consideration of privacy protection
	unicast security on group key distribution	unicast security on group key distribution	unicast security on group key distribution
	operation efficiency		

2.2.3. Summary

In Section 2.2, we investigated secure communication in unicast and multicast channels. Having reviewed issues relating to securing unicast *communication channel*, a number of conclusions can be made:

- The use of long-term shared keys and public keys renders unicast *communication channels* vulnerable; hence they are not suitable for sensitive information protection.

Therefore, there is no proper approach to protect sensitive information in unicast *communication channel*. The use of long-term shared keys and public keys is the drawback which renders *communication channel* vulnerable. Also, a number of conclusions for securing multicast *communication channel* can be made by reviewing multicast approaches:

- Securing multicast *communication channel* approaches focus on rekeying operations (performances) in the group communication.
 - Contributory key agreement is not suitable for a large group even though it is flexible in terms of membership changes. It does not rely on a group controller.
 - Centralized key distribution suffers from the rekeying complexities associated with large groups.
 - Decentralized key distribution has large communication overheads.
- No multicast communication solutions exist that ensure privacy protection for group members and confidentiality for sensitive information systems.
- No unambiguous instructions on group key distribution to individuals.

Therefore, no proper approach helps protect sensitive information in multicast *communication channel*. The lack of privacy protection for group members and confidentiality for sensitive information systems is a drawback which threatens multicast *communication channel*. In next section, we will investigate the extant technical approaches in protecting *user interface*.

2.3. 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” [Me02]. It can be accomplished by using one or more of the following validation approaches: a knowledge factor¹⁸ (something users know), a possession factor (something users have) or a biometrics factor (something users are). In this section, we examine and review different authentication factors, analyse their advantages and disadvantages, and indicate the common problems facing each factor. We finish this section by distinguishing “authentication” from the closely related term “authorization”.

2.3.1. Proof by Knowledge

When using a knowledge factor for authentication, an entity proves its identity by providing knowledge of some secret information such as a password or a cryptographic key. This information may either be static or dynamically changing over time. Generally speaking, static information is used to implement weak authentication

¹⁸ A factor is a piece of information used to authenticate or verify a person's identity for security purposes.

mechanisms, whereas dynamic information is used when implementing strong authentication mechanisms.

Static Information

Initially, plain passwords were used to authenticate two communication entities by comparing them. However, it is possible for an adversary to guess a plain password [FeKa90, MoTh79]. In order to solve this problem, a plain password can be run through a one-way hash function, which would convert it into a random looking sequence of bytes. Nevertheless, the password database itself could still be vulnerable. Despite this weakness, this method of protection is still being used to this day, primarily for UNIX systems [FeKa90].

In 1992, Bellare and Merritt [BeMe92] introduced encryption key exchange (EKE¹⁹) protocol, which generates a session key between two authenticated entities to prevent guessing attacks. The EKE protocol was very influential and became the basis for much future work in this area, such as DH-EKE, SPEKE and A-EKE [BeMe93, Ja96a, Lu98b, Pa97, StTsWa95, Wu98]. The EKE protocol operates as follows:

Assume that two entities, **A** and **B**, wish to establish a secret (an authenticated session key). Initially, both entities share a password, say P , and agree with a base α and a modulus β for discrete exponentiation:

- i) **A** picks a random number, R_A , and calculates $\{\alpha^{R_A} \bmod \beta\}P$. **A** then sends the result, together with the identifier of **A**, to **B**

$$A \rightarrow B : A, \{\alpha^{R_A} \bmod \beta\}P$$

¹⁹ EKE is password authentication protocol, which can be categorized into either static information or dynamically changing information section. In this thesis, we regard it as both.

ii) Upon receipt, **B** picks a random number, R_B , and calculates $\alpha^{R_B} \bmod \beta$.

Because **B** knows P as well, **B** uses P to decipher $\{\{\alpha^{R_A} \bmod \beta\}P\} \sim P$ and calculate $\alpha^{R_A R_B} \bmod \beta$. **B** then derives a session key $K_{session}$ from the result, perhaps by selecting certain bits as agreed. Finally, **B** generates a random challenge $challenge_B$, and sends this to **A**.

$$B \rightarrow A : \{\alpha^{R_B} \bmod \beta\}P, \{challenge_B\}K_{session}$$

iii) After **A** uses P to understand $\alpha^{R_B} \bmod \beta$, **A** is then able to calculate $K_{session}$.

It, in turn, is used to decipher $\{\{challenge_B\}K_{session}\} \sim K_{session}$. Lastly, **A** generates its own random challenge, $challenge_A$, and sends this to **B**.

$$A \rightarrow B : \{challenge_A, challenge_B\}K_{session}$$

iv) Upon receipt, **B** deciphers $\{\{challenge_A, challenge_B\}K_{session}\} \sim K_{session}$ and verifies $challenge_B$. **B** then sends $challenge_A$ back.

$$B \rightarrow A : \{challenge_A\}K_{session}$$

v) Lastly, **A** deciphers and verifies that the challenge matches the original.

Despite the interest create by the EKE protocol, these protocols have not been proven secure and their conjectured security is based on heuristic arguments [GeLi06]. Gennaro and Lindell consider that the first rigorous treatment of the problem was provided by Halevi and Krawczyk [HaKr99]. They actually considered an asymmetric hybrid model to provide a password-based solution. But further examination of the risks of password authentication protocols [CoDiWa04], revealed that three types of attacks could compromise their security: technical attacks, discovery attacks and social

engineering attacks [BiK195, De89]. To counter these types of attacks, it is suggested of a need for password and system rules, and training and awareness.

Dynamically-changing Information

The idea behind using dynamically-changing information in authentication by a proof by knowledge is that each authentication process uses a unique piece of secret information once only. The secret information is not re-used. Consequently, if an adversary eavesdrops on an authentication process and obtains the relevant information, the adversary is not able to use the information in a replay attack. The information will not be valid a second time.

The use of dynamically-changing information is not a new idea. Transaction authentication numbers (TANs) [Op96] have been in use for some online banking services as a form of single use passwords to authorize financial transactions for a long time. A TAN is a piece of authentication information that can be used in a transaction. For example, a bank randomly creates a set of unique TANs for a user and delivers the set to the user securely. To perform a transaction, the user enters the request and "signs" the transaction by entering an unused TAN. The bank verifies the TAN submitted against the list of TANs issued to the user. If it is a match, the transaction is processed. If it is not a match, the transaction is rejected.

When the number of authentication processes or transactions is exhausted, the management of TANs will become difficult, since it is not scalable. Therefore, the use of cryptographic techniques [HaAt94] is necessary to solve the scalability problem. The following discussion gives a brief overview of current approaches.

One-time Password (OTP) Schemes. As its name implies, an OTP scheme employs a password can only be used once. Traditionally, static secret information can more easily be breached than dynamic information by an adversary given enough attempts and time. By constantly altering the secret information this risk can be reduced. The OTP schemes are very similar to TANs; however, the major difference is that, unlike the TAN schemes, the OTP schemes generate the secret information dynamically and deterministically, and they are scalable.

Generally speaking, there are three types of OTPs. The first type uses a mathematical algorithm to generate a new password based on the previous password. This type was originally proposed by Leslie Lamport in the early 1980s [La81]. In his scheme, two entities start with an initial seed (say s). A one-way function F is then used to generate a sequence of OTPs: $F(s), F(F(s)), F(F(F(s)))...$ as many times as necessary. If an infinite sequence of OTPs is needed, a new seed s' can be chosen after the s is exhausted. The scheme was developed at Bell Communication Research (Bellcore), now Telcordia Technologies²⁰, called S/KEY [Ha94]. It uses a cryptographic hash function as a one-way function to generate dynamic secret information. Haller [Ha94] stated that the S/KEY scheme “does not protect a network eavesdropper from gaining access to private information ...”, but claimed that the S/KEY scheme “is not vulnerable to eavesdropping / replay attacks”. In 1996, Mitchell and Chen [MiCh96] commented and proved that the scheme failed to provide this property. A similar system called One-Time Passwords in Everything (OPIE) [McAtMe95] was derived from S/KEY and was claimed “...to secure a system against replay attacks”. A number of offline OTPs techniques [LiZh04, RuWr02] were also

²⁰ <http://www.telcordia.com/>

proposed to enhance the security of such systems. However, all these schemes were vulnerable to phishing attacks [RoSa05]. Such systems were also breakable due to sharing long-term secret information [KuLeSr05].

The second type of OTP scheme is based on time synchronization between two entities. It usually relates to physical hardware tokens that generate an OTP via an accurate clock and synchronize with the clock on the authentication entity. The RSA SecurID tokens²¹ are the most widely-deployed OTP system in use today. Generally speaking, each SecurID token contains a cryptographic processor that generates an authentication code at fixed intervals (usually 30 or 60 seconds) using the built-in clock and an encoded random 64-bit secret key that encrypts the code with the key. The token offers a level of protection against password replay attacks, but it fails to provide adequate protection against man-in-the-middle attacks. At the RSA Conference in February of 2005, a live demonstration was conducted to defeat an RSA SecurID OTP Token [Tu07].

The last type of OTP scheme again uses a mathematical algorithm, but the new password is based on a challenge and a counter instead of being based on a previous password. The challenge type of OTP requires a user to provide a time-synchronized challenge to be properly authenticated. This kind of authentication will be discussed in detail in the next section.

Challenge Responses (CR) Mechanisms. OTP schemes use one-way authentication. They are simple and straightforward; an entity provides a piece of synchronized authentication information to another entity for validation. In contrast, CR mechanisms require both entities to interact (but not to be synchronized) and they involve two-way

²¹ <http://www.rsa.com/>

authentication, whereby both entities must each convince the other that they know the shared secret (the password), without this secret ever being transmitted in open networks. The first use of cryptography to achieve authentication was described by Feistel [Fe70] and applied to a network context by Branstad [Br73]. Diffie and Hellman [DiHe76a] and Kent [Ke77] developed it in more depth, and Needham and Schroeder [NeSc78] devised and improved the protocols. The Needham-Schroeder protocol aims to establish a session key between two entities on a network, performed as follows:

Assume that two entities, **A** and **B**, wish to establish a connection and that **S** is a server trusted by both entities. Initially, both entities share a secret key with the server, say K_{AS} and K_{BS} .

- i) **A** sends a message to **S**, requesting communication with **B**; meanwhile nonce challenge n_A ensures the message is fresh.

$$A \rightarrow S : A, B, n_A$$

- ii) Once received, **S** generates K_{AB} and sends it back to **A** copy encrypted under K_{BS} .

$$S \rightarrow A : \{n_A, K_{AB}, B, \{K_{AB}, A\}K_{BS}\}K_{AS}$$

- iii) Upon receipt, **A** forwards the key to **B**, thus authenticating the data.

$$A \rightarrow B : \{K_{AB}, A\}K_{BS}$$

- iv) **B** then generates the nonce challenge n_B , and sends it to **A** to show it has the key K_{AB} .

$$B \rightarrow A : \{n_B\}K_{AB}$$

v) Last, A performs a simple operation on the nonce n_B , and sends it back to verify that the same key K_{AB} is held with A .

$$A \rightarrow B : \{n_B - 1\} K_{AB}$$

The protocol is vulnerable to a replay attack. If an adversary records one run of this protocol, and then subsequently learns the value of K_{AB} , the adversary can then replay the message (iii) to B in which B is unable to tell freshness of the key. This flaw is fixed in the Kerberos protocol [ChGeRu90, StNeSc88] by the inclusion of a timestamp. The Kerberos protocol uses a Key Distributed Centre to authenticate users, and it distributes session keys to both users and servers. In the original design of Kerberos, session keys exchange used long-term shared keys. Kerberos has major drawbacks [KoNeTs94]:

- It depends on long-term symmetric encryption keys to generate session keys for key exchange.
- It requires clock synchronization among all entities.
- It requires continuous availability of a central server.
- Because the secret keys for all users are stored on the central server, a compromise of that server will compromise all users' secret keys.

Although the use of asymmetric cryptography [Er03, HaMe01, SiCh97] has been proposed to overcome these drawbacks, all asymmetric cryptography is susceptible to brute force key search attacks [Ka67]. One example of a more sophisticated CR mechanism is zero-knowledge password proof (ZKPP)²². This is an interactive method

²²ZKPP is not used in the cryptographic literature. In fact, it does not have much in common with Zero-knowledge proofs. It is a special kind of zero-knowledge proof of knowledge that addresses the limited size of passwords.

for one entity to prove to another entity that it knows the password without revealing that password. The first protocol to demonstrate ZKPP authentication was the EKE protocol (discussed in Section 2.2.1 static information).

ZKPP was later used as the basis for a new protocol named the secure remote password (SRP) protocol [Wu98], SRP combines techniques of zero-knowledge proofs [BeGoGo88] with asymmetric key exchange protocols. As claimed, it has a number of desirable properties:

- It allows a user to authenticate itself to a server.
- It is resistant to dictionary attacks mounted by an eavesdropper.
- It does not require a trusted third party.

However, as Wu [Wu98] mentioned, SRP has some security threats. These include key materials distribution via open networks and the possibility of an inappropriate password choice compromising security. However, by using such kinds of protocols, a sender can prove knowledge of a secret while revealing no information of the secret. It is possible, and likely, that zero-knowledge protocols will become more important and widely used in the future [Op01].

2.3.2. Proof by Possession

In a proof by possession, an entity proves its identity by proving ownership of some physical token, such as smart cards, USB tokens, magnetic stripe cards or identification cards. The token is used in addition to or in place of a password. It acts as an electronic key to access information; some may store cryptographic keys and even critical information of users. Proof by possession is most frequently used for hard token and smart cards authentication.

Hard Token Authentication

Hard token authentication is a form of authentication that requires something users have. These tokens are programmed to generate and display new passwords at certain time intervals. In order to access a system, an entity must provide the password displayed on the token, which is the “something users have” authentication factor. The algorithms of generating credentials are the four types earlier discussed in proof by knowledge (Section 2.2.1): static password, dynamically-changing password (OPT), asynchronous password and challenge response [Op01].

The security of token authentication is guaranteed by a constantly-changing password. The frequency of change makes it difficult for an adversary to use a password to gain malicious access. Even if the adversary successfully steals a password, by the time the adversary enters it into the system, the password will have already changed. Because the mechanism of generating credentials relies on the first factor (that is, proof by knowledge), it suffers the same security threats and concerns.

Smart Cards

The best-known example of proof by possession is smart card authentication, which is based on a credit card-sized plastic card embedded with an integrated circuit chip. Subscriber Identity Module (SIM) cards are a smart card used in mobile phones to authenticate users with service centres. Smart cards provide not only memory capacity, but also computational capability [CoBr93, WaZhZh06].

The smart card chip was invented by German rocket scientist Helmut Gröttrup and his colleague Jürgen Dethloff in 1968; the patent was finally approved in 1982. The first mass use of the cards was for payment in French pay phones, starting in 1983.

The second use was with the integration of microchips into all French debit cards completed in 1992. The major boom in smart card use came in the 1990s, with the introduction of the smart-card-based SIM used in GSM mobile phone equipment in Europe.

Smart cards are highly secure by design. Should unauthorized users try to tamper with the contained data, a security mechanism will destroy all information stored in the card. Smart card authentication can utilize this security mechanism to store a user's sensitive data. Some smart cards have separate cryptographic coprocessors that support different algorithms such as RSA, ECC and triple-DES.

Smart cards contain unique features that bring many benefits for users [IsSu01]. Smart cards provide a portable, easy-to-use form factor that many are familiar with using. Smart cards are also capable of processing, and not just storing, information. Also, secret key information is stored tamperproof on the card. Secret key operations are performed directly on the card; hence, no spy attacks on the secrets are possible. Moreover, high security is achieved when running cryptographic operations in the cards.

Smart cards have been advertised as suitable for personal identification tasks, because they are engineered to be tamper resistant. The embedded chip of a smart card usually implements some cryptographic algorithm. The security of smart card-based authentication relies on the security of smart cards, and also on the secret of the cryptographic algorithm, the keys stored, and the access control inside the smart card – all of which can become the targets of attackers.

Chan [Ch97] reviewed techniques to attack smart cards. He reported that logical attacks are possible (based on the fact that electrically-erasable programmable read

only memory (EEPROM) write operations can be impacted via unusual voltages and temperatures) and that data can be trapped by raising or dropping the supplied voltage to the microcontroller. This process, known as differential power analysis, is also reported by Ross and Markus [AnKu96]. Another problem is that smart cards can be physically disassembled by using acid, abrasives, or some other technique to obtain direct, unrestricted access to the on-board microprocessor. Although such techniques obviously involve a fairly high risk of permanent damage to the chip, they permit much more detailed information (for example, photomicrographs of encryption hardware) to be extracted. Smart cards authentication therefore suffers security threats when the physical cards are lost or stolen.

2.3.3. Proof by Property

In a proof by property (inherence) authentication process, an entity proves its identity by proving biometric characteristics. The biometric characteristics are measured and compared with a reference pattern. The biometrics offers greater security and convenience than traditional methods of personal recognition. In some applications, biometrics can replace or supplement the existing technology. Formally, the biometrics verification can be described as follows [JaRoPr04]:

$$(ID, X_{Bio}) \in \begin{cases} B_1, & \text{if } S(X_{Bio}, X_p) \geq t \\ B_2, & \text{otherwise} \end{cases} \quad (2.1)$$

For the entity, given an input feature X_{Bio} , extracted from the biometric data, and a claimed identity ID , it is possible to determine if (ID, X_{Bio}) belongs to class B_1 (genuine) or B_2 (imposturous). Function S measures the similarity between X_{Bio} and

X_p ; t is a predefined threshold. It is notable that biometric measurements of the same entity taken at different times are almost never identical; thus the need for a threshold.

Biometric characteristics can be divided in two main classes: physiological traits, which are related to the shape of the body, such as fingerprints, retinal pattern, DNA sequence and hand geometry, and behavioural traits, which are related to the behaviour of a person, such as a signature, keystroke dynamics and voice. According to Prabhakar & Pankanti et al., [PrPaJa03], any trait can serve as a biometric characteristic as long as it satisfies the following criteria:

- **Universality** – each person should have the characteristic.
- **Distinctiveness** – any two persons should be sufficiently different in terms of the characteristic.
- **Permanence** – the characteristic should not vary over a period of time.
- **Collectibility** – the characteristic should be quantitatively measurable.

Historically, the first physiological biometric characteristics were fingerprints. A fingerprint is an impression of the friction ridges of all parts of the finger. It is the oldest form of biometric verification, and also the best example of a proof by property. Persian official and physician Rashid-al-Din Hamadani comments on using fingerprints to identify people in China: "Experience shows that no two individuals have fingers exactly alike" [Co03]. The first behavioural biometric characteristic to be used – and is still widely used today – is the signature. Although signatures require contact with a writing instrument and an effort on the part of the user, they are accepted in government, legal and commercial transactions as a method of verification.

Biometrics as a commercial, modern technology has been around since the early 1970s when the first commercially-available device was brought to market. Shearson

Hamil, a Wall Street company, installed a finger-measurement device to serve as a time-keeping and monitoring application [WoOrHi02]. Since then, biometrics has improved tremendously in ease of use and diversity of applications. The advancement of biometrics has been driven by low-cost increased computing power, better algorithms and the cheaper storage mechanisms available today [WoOrHi02]. The fundamental operation of a biometrics authentication mechanism follows biometrics acquisition, biometrics classification and biometrics matching [DuJuKo02].

Similarly to the many interesting and powerful developments of technology, there are also concerns about biometrics. The biggest concern is the fact that once a fingerprint or other biometric source has been compromised, it is compromised for life. Also, noise in sensed data can result in a user being incorrectly rejected (for example, in the case of a fingerprint with a scar or a voice altered by a cold). Moreover, privacy is another concern; how biometrics, once is collected, can be protected.

2.3.4. Authentication versus Authorization

Authentication is the process of verifying that credentials are genuine. Authorization is the process of checking if a validated user is permitted to access a particular resource. More precisely, as defined in Khare [Kh06], authentication is the process of verifying that a claim made by a subject should be treated as acting on behalf of a given principal (for example, person, computer or smart card), while authorization is the process of verifying that an authenticated subject has the authority to perform a certain operation. Therefore, authentication heads authorization. Also, authorization cannot occur without authentication. Consequently, the terms authentication and authorization are frequently used together.

2.3.5. Summary

In Section 2.3, the role of authentication factors (knowledge, possession and property) in protecting *user interface* in sensitive information systems were investigated and the relationship between authentication and authorization was discussed. The knowledge factor²³ was investigated from a technical perspective, while possession and property factors were briefly reviewed. The following findings can be presented from the existing literature:

- Protocols (such as EKE and its successor) that use static information (such as passwords) provide weak authentication.
- Protocols (such as OTP, CR and ZKP) that use dynamic information can provide strong authentication. Despite this, extant techniques cannot secure *user interface* in sensitive information systems, due to the employment of long-term shared and public keys.
- Lost or stolen physical devices are the biggest concern relating to the possession factor.
- The property factor can be permanently compromised.
- No group authentication and authorization protocols to protect sensitive information systems and handle dynamic membership.

Table 2.5 presents a comparison of the three factors. The table lists the advantages and disadvantages of proof by knowledge, possession and property. From the comparison, it can be seen that no single factor satisfies the security requirements of sensitive information systems.

²³ The knowledge factor was reviewed soundly in terms of its technical aspects because the other two factors relate to hardware-based technology.

Also, by 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. Therefore, a new and proper authentication and authorization approach is required. In the next section, we will thoroughly examine the existing approaches in *sensitive information storage* protection.

Table 2.5. Advantages and Disadvantages of Knowledge, Possession and Property Factors.

	Knowledge	Possession	Property
Advantages	simple administration	stronger user authentication mechanism	stronger user authentication mechanism
	strong authentication (dynamic changing information)		convenient (password always at hand)
	inexpensive method of user authentication		hard to duplicate cannot be shared or forgotten
Disadvantages	relatively weak security	more expensive than knowledge factor	more expensive than knowledge factor
		possibility of lost or stolen devices	accuracy concern (noise in sensed biometric data)
		extra device	extra device
		management matters: need to issue hard tokens or smart card, and track	extra dependency on software and hardware compromised for life privacy concerns

2.4. Securing Sensitive Information Storage

Protection of critical data in sensitive information systems involves three components: *communication channel*, *user interface* and *sensitive information storage*. Previous sections have reported on approaches to *communication channel* and *user interface* protection mechanisms. In this section, the techniques of *sensitive information storage* protection are reviewed.

According to an Enterprise Strategy Group (ESG) estimation, annual sensitive information growth reaches 50-60% for many organizations [Wh09]. Ensuring the security of sensitive data at rest is thus a worthwhile endeavour. Clark et al. [ClChCh08] suggest that cryptographic techniques for *sensitive information storage* protection can be divided into two categories: disk encryption and database encryption. In the following sections, these two storage protection techniques are investigated.

2.4.1. Disk Encryption

Disk encryption is a special case of data-at-rest protection where the storage media is a sector-addressable device, such as a hard disk or a flash card. Two high-level terms describe implementations of disk encryption:

- **Full disk encryption** (FDE) or “whole disk encryption” encrypts every bit of data on a disk. This encryption is performed through software such as PGPDisk from PGP Corporation [Pg08], BitLocker Drive Encryption from Microsoft Corporation [Hy08] or McAfee Endpoint Encryption (Safeboot Device Encryption) [Mc07]. The encryption is usually performed on a sector-by-sector basis. A filter driver is loaded into memory at bootup, encrypts every file as it is

written to disk and decrypts any file and data that is moved off the disk. The process is transparent to end users.

- **Filesystem level encryption (FLE)** or “file or folder encryption” is a form of disk encryption where individual files or directories are encrypted by the file system itself. Alternatively a third-party software package, such as Encrypting File System (EFS) for Microsoft Windows [Co06], IBM Encryption Facility for z/OS [Bo07] or EVFS for HP-UX (Encrypted Virtual File System) [Hp07] may perform the encryption.

FDE does not replace FLE, because FDE generally uses the same long-term key for encrypting the whole disk and all data are decipherable when the system runs. Although some FDE solutions use multiple keys for encrypting different partitions, if an adversary gains access to the computer at run time, the adversary has access to all files. In contrast, FLE allows different keys for different folders. An adversary is thus prevented from extracting information from still-encrypted files and folders. In addition, most FDE schemes are vulnerable to a cold-boot attack, in which encryption keys can be stolen by cold-booting a machine already running an operating system, then dumping the contents of memory before the data disappears [HaScHe08].

Overall, for sensitive information protection, FLE offers greater relative security than FDE and among the FLE software, the EFS and IBM encryption facilities are the most secure and widely deployed. These will now be discussed in more detail.

Encrypting File System (EFS)

EFS is a file system driver that provides security for Microsoft Windows (2000 or later). It enables files to be transparently encrypted onto Windows NT file systems to

protect confidential data from attackers with physical access to the computer. The EFS employs a bulk symmetric key, known as the File Encryption Key (FEK), with symmetric algorithms, such as AES and TDES, to secure sensitive information. The FEK is encrypted with a public key that is associated with the user who encrypted the file. This encrypted FEK is stored in the file header. This mechanism can be described as follows, and is illustrated in Figure 2.15 and 2.16.

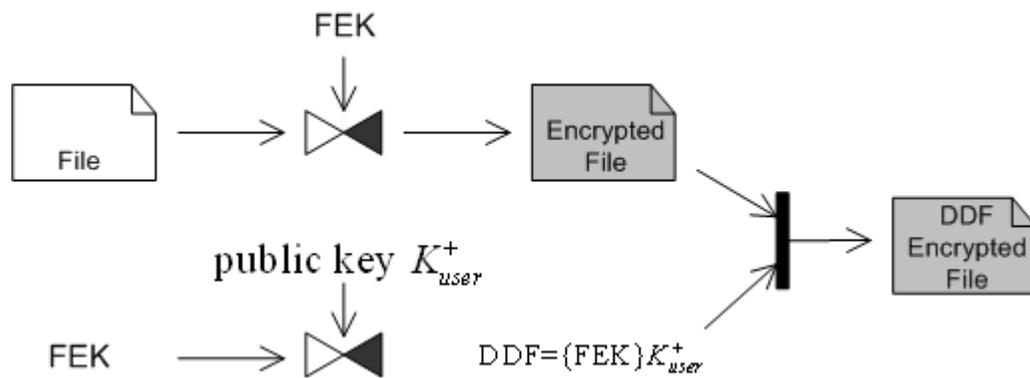


Figure 2.15. EFS: File Encryption.

Encrypting a File with EFS. When a user encrypts an existing file, the following process occurs:

- i) A file encryption key (FEK) is randomly generated.
- ii) A data decryption field (DDF) is created to contain the encrypted FEK.
- iii) The existing file is encrypted by the FEK using AES, 3DES, or DESX algorithms, depending on the version of the operating system and the effective security policy.
- iv) The FEK is encrypted by the public key of the user.
- v) The encrypted FEK is added to the header of the file (DDF).

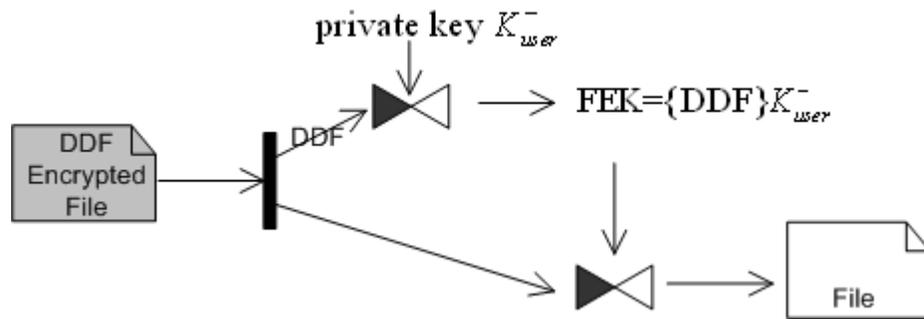


Figure 2.16. EFS: File Decryption.

Decrypting a File with EFS. When a user decrypts an encrypted file, the following process occurs:

- i) The file system retrieves the DDF from the file.
- ii) EFS retrieves the private key of the user to decrypt the DDF and obtain the FEK.
- iii) EFS uses the obtained FEK to decrypt the file.

It is notable that the security of EFS relies on the security of the asymmetric cryptosystem. As discussed on Section 2.1, asymmetric cryptography suffers from brute force key search attacks; therefore, once the user is breached, the sensitive information is disclosed. In addition, the method of using EFS in the MS Windows' family leads to security concerns, because an adversary can log in as that user (or recovery agent) and gain access to the RSA private key which can decrypt all files²⁴.

IBM Encryption Facility

The IBM encryption facility for z/OS, first introduced in 2005, is a host-based software solution designed to encrypt sensitive data before transferring it to tape for archival purposes or business partner exchange,. The encryption services feature is

²⁴ <http://home.eunet.no/pnordahl/ntpasswd/>

similar to EFS in that it supports TDES triple-length keys or 128-bit AES keys in z/OS to protect the sensitive information. In addition, IBM uses RSA keys to wrap and unwrap the AES and TDES data keys to encrypt the file. With this technique, IBM claims that many files can be generated using different encryption keys. However, the problem with this solution is that its security relies on an asymmetric key infrastructure. If the public key pairs are disclosed, no matter how many different encryption keys are used to protect data, the whole data system will be compromised.

2.4.2. Database Encryption

A database is a structured collection of records or data that is stored in a computer system. Based on a database model (such as a relational, hierarchical or object model), the structure is achieved by organizing the data. The relational model (as seen in Microsoft SQL Server or Oracle Database) is the most common today.

Because security has become a major issue in recent years, many commercial database vendors provide built-in encryption mechanisms. Data is encoded natively into the tables and deciphered "on the fly" when a query comes in. Most relational database management systems (RDMS) conduct database security by applying database encryption that uses cryptographic keys to encipher the sensitive data [Gu06]. The security of the RDMS therefore relies on the security of the cryptographic keys, and the key management in database management systems plays an important role in sensitive information protection.

Key Management in SQL Server

SQL Server uses encryption keys to help secure data and credentials stored in a server database. In SQL Server, encryption keys include a combination of symmetric

and asymmetric keys that are used to protect sensitive data. The symmetric key (database encryption key DEK) is randomly generated for each user at the first start of the service and is stored in SQL Server. The DEK is protected by a pair of asymmetric keys created by the operating system. The private key of the asymmetric key pair is secured by a symmetric key (database master key DMK), which is under protection of the service master key (SMK). At the root of the cryptographic key hierarchy is the Windows Data Protection Application Programming Interface (DPAPI), which secures the key hierarchy at the machine level and is used to protect the service master key (SMK) for the database server instance [Hs08]. This architecture of key management is named transparent data encryption (TDE). TDE is a new encryption feature introduced in Microsoft SQL Server 2008, and is designed to provide protection for the entire database at rest without affecting existing applications. Figure 2.17 shows the full encryption hierarchy.

It is notable that the security of SQL Server relies on the security of DPAPI. The mechanism of encrypting data is same as EFS. The breaches of DPAPI result in operating systems being compromised. The disclosure of DMK compromises all sensitive information in the database. The exposure of a DEK leads to the leakage of sensitive information. In other words, the use of long-term keys causes the possibility of a security breach in SQL Server. In addition, such architecture does not stress the privilege of the database administrator, who is able to access all user data. In short, in SQL Server, the privacy of users cannot be guaranteed.

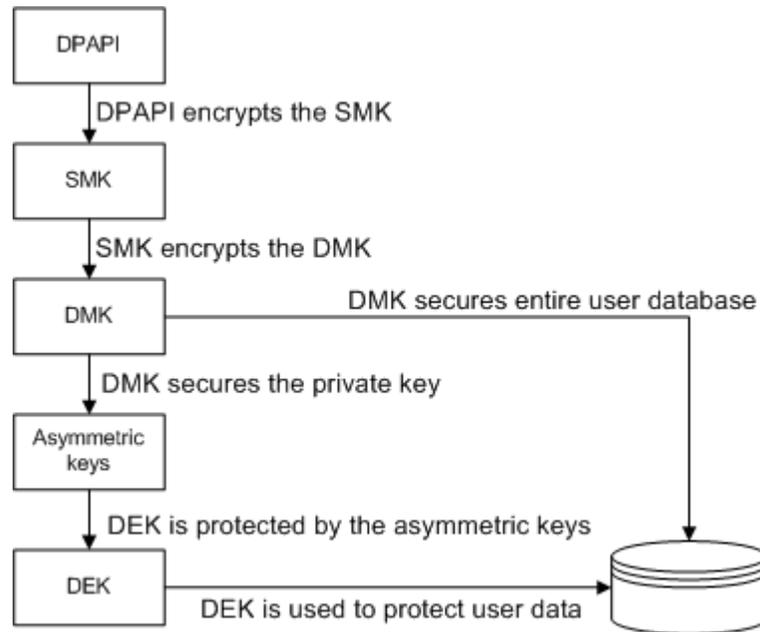


Figure 2.17. Transparent Data Encryption Hierarchy.

Key Management in Oracle Database

Unlike SQL Server where database administrators (DBAs) have total privileges and are able to access all data, Oracle Database [Na05] supports limited partitioning of DBA privileges, systems operator (SYSOPER) and all DBA privileges (SYSDBA). However, the partitioning does not solve the root cause of the DBA privilege problem.

Oracle Database also provides advanced security transparent data encryption (TDE)²⁵. TDE provides built-in key management and complete transparency for encryption of sensitive application data. TDE encrypts data before it is written to disk and decrypts data before it is returned to the application. The key management in Oracle Database is illustrated in Figure 2.18. TDE generates an encryption key randomly, called table key K_{table} , to secure table columns. If there is more than one column in a single table, the same key is used for all columns. Each table key is stored

²⁵ Oracle uses the same term as SQL Server.

in the Oracle data dictionary and is encrypted using the TDE master encryption key (MEK). The MEK is stored outside the database, using Oracle Wallet and a hardware security module such as a smart card.

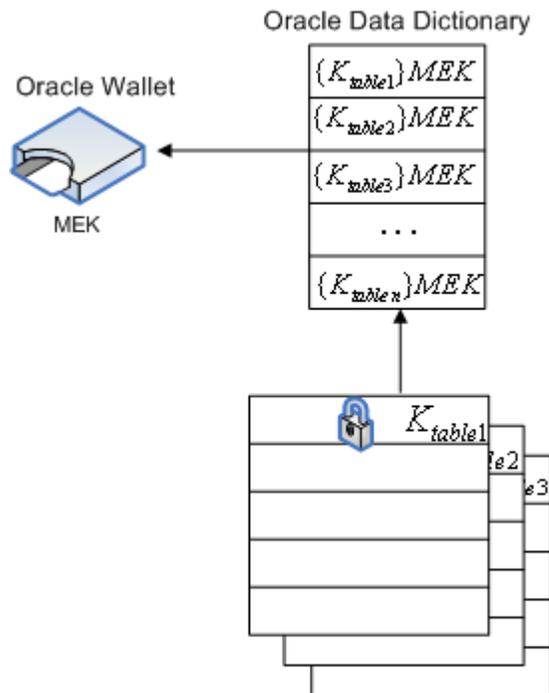


Figure 2.18. TDE in Oracle Database.

It is observable that the security of Oracle has been improved theoretically compared to SQL Server, because Oracle employs table keys to encipher each table whereas SQL Server uses one DEK to secure all sensitive data of one user. However, the security of Oracle Database is similar to SQL Server, which relies on the security of the MEK. In other words, a breach of the MEK exposes all sensitive information in the database. The MEK is stored in Oracle Wallet, and, as discussed in Section 2.2.2 proof by possession, hardware security modules, such as smart cards, can be lost or stolen. Also, damage to the hardware security module can cause the loss of sensitive information.

2.4.3. Summary

In Section 2.3, protecting sensitive information at rest was investigated by categorizing disk encryption and database encryption. After the literature on *sensitive information storage* protection has been reviewed, the following findings can be presented:

- The security of disk encryption relies on long-term asymmetric or symmetric keys. Once these keys are compromised, the sensitive information in the disk is disclosed.
- The security of database encryption relies on a number of long-term symmetric and asymmetric keys. The compromise of the master encryption key results in exposure of all sensitive data.
- The DBA in database encryption has full privileges and can access all information in the database, which can lead to a breach of sensitive information systems.
- Neither disk encryption nor database encryption can ensure privacy protection.

The pros and cons of disk and database encryption for *sensitive information storage* protection are presented in Table 2.6. From the table, it can be seen that neither technique can ensure privacy protection, and also that the security of both 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). Therefore, a new technique in *sensitive information storage* protection is necessary.

Table 2.6. Advantages and Disadvantages of Disk Encryption and Database Encryption.

	Disk Encryption	Database Encryption
Advantages	transparent data encryption	weak privacy protection
	various authentication processes available to secure the disk password, such as hard token, soft token and pass phrase	transparent data encryption
		different tables can be encrypted with different keys
Disadvantages	security relies on long-term keys	security relies on long-term keys
	one key for all sensitive information	key management has to be sophisticated
	corrupted unique recovery key loses the utility of sensitive information	
	increase in data access times	requires tight integration with the database
	lack of privacy protection	DBA has full privileges of the database
	disk administrator has full privileges of the disk	

2.5. The Current Models for Information Security

In the previous sections (2.2 securing communication channel, 2.3 securing user interface and 2.4 securing sensitive information storage), the extant technical approaches in protecting sensitive information have been investigated. In this section, according to Section 1.2.1 (characteristics of sensitive information²⁶), we review the existing information security models²⁷ used as security assessment of sensitive information systems.

According to the discussion in Section 1.2 (sensitive information), sensitive information systems inherit the properties of information systems. Availability is thus an essential principle for both information systems and sensitive information systems. The availability attribute is not necessary in the security assessment of sensitive information systems as it is already included in information systems. Moreover, for sensitive information, the attributes such as confidentiality, integrity, authenticity, authority and possession are more important than the attribute of availability, since the loss of the availability does not harm sensitive information itself. Therefore, the security assessment properties for sensitive information systems are **A**uthenticity, **A**uthority, **I**Ntegrity (IN), **N**on-**R**epudiation (NR), **C**Onfidentiality (CO) and **U**Tility (UT). Moreover, as discussed in Section 2.3.4 (authentication versus authorization), the term of authentication and authorization is always used together. We thus discuss authenticity and authority (AA) together for sensitive information systems.

²⁶ Characteristics of sensitive information are authenticity, confidentiality, possession (authority), integrity, non-repudiation, utility and availability.

²⁷ A security model is a framework that can be used to guide the design of a sensitive information system or to evaluate the security of a sensitive information system.

As historical aspects of information security models, the traditional model, the CIA (confidentiality, integrity and availability) Triad [Cn92, Pe08], has been used as the principle of information security since computers were introduced. Using the CIA Triad as a foundation, many groups have proposed security frameworks for information security, including the Organisation for Economic Cooperation and Development (OECD) [Oe92]. The frameworks include the Generally Accepted System Security Principles (GASSP) [Po99], developed by the International Information Security Foundation (I²SF) Sponsored Committee, and the British Code of Practice proposed by the UK Department of Trade and Industry (DTI) [BS93].

With the development of IT, a new type of model was required. The US Department of Defence (DoD) recognized the limitation of the CIA Triad by adding two additional elements (authenticity and non-repudiation). The Trusted Computer System Evaluation Criteria [LaBrHa85] (commonly known as the Orange Book), became “...one of the most renowned publication on computer security, and has had a profound influence in encouraging computer manufacturers to include security in their products for many years” said Parker [Pa98]. He also argued the shortcomings of the CIA foundation of information security and proposed a new model, named the Parkerian Hexad, by adding three extra elements (utility, authenticity and possession). In the following section, two profound and widely deployed information security models - the CIA Triad and Parkerian Hexad - are discussed.

2.5.1. CIA Triad

For over twenty years information security has held confidentiality, integrity and availability (known as the CIA Triad; shown in Figure 2.19) as its core principles.



Figure 2.19. CIA Triad.

Confidentiality is the property of preventing disclosure of information to unauthorized individuals or systems, such as a credit card transaction on the Internet and personal medical records in healthcare. Integrity ensures that data is an accurate and unchanged representation of the original secure information, such as transaction continuity and completeness in the business. Availability ensures that the information concerned is readily accessible to the authorised viewer at all times, such as preventing denial-of-service attacks in Internet banking systems.

While the CIA model was appropriate when computing environments were simple, it is not suited to larger and more complex systems such as those associated with electronic business (e-business), electronic medical record (EMR) systems and electronic government (e-gov). As recognised by the US DoD, non-repudiation and authenticity characteristics of current information development were not included in the CIA model. Nor did the CIA model address another current area of concern: information possession. For example, making unauthorized copies of copyrighted software constitutes theft, but does not breach the tenets of the CIA model, and the issue is one of possession rather than a loss of confidentiality, integrity and availability [Pa98].

2.5.2. Parkerian Hexad

Because of the limitations of the CIA Triad, the Parkerian Hexad (see Figure 2.20) was proposed by Donn B. Parker. Incorporating six elements of information security, the Parkerian Hexad adds three additional attributes to the CIA Triad. The six elements are:

- **Confidentiality** – the limited observation and disclosure of knowledge.
- **Possession** – the holding, control and ability to use information.
- **Integrity** – the completeness, wholeness and readability of information and quality being unchanged from a previous state.
- **Authenticity** – the validity, conformance and genuineness of information.
- **Availability** – the usability of information for a purpose.
- **Utility** – the usefulness of information for a purpose.

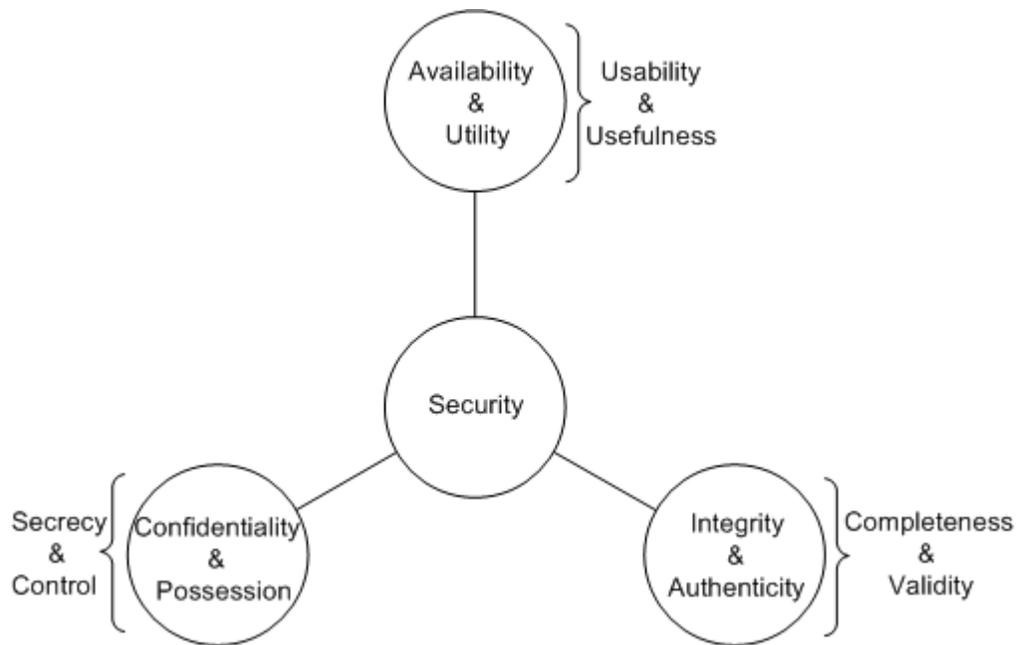


Figure 2.20. Parkerian Hexad.

The Parkerian Hexad is non-overlapping. This means each principle (attribute) is absolutely necessary to ensure that security is maintained. However, the principle can

be relationally linked to each of the three components of the traditional CIA model. The new model can also be used to evaluate the security of information systems. However, it has limitations. Like the CIA Triad, the Parkerian Hexad also lacks a non-repudiation attribute (where, in the case of Internet banking transactions, a legal entity denies the actions in such systems).

Another concern is that privacy is not addressed by the Parkerian Hexad. Privacy may be implied by confidentiality, but privacy goes beyond confidentiality. The sensitive information of users must be protected, and the protection is just part of privacy. Protection also requires users to manage their information, such as delegating permission on partially-sensitive information, in order to protect assets. For example, in a healthcare system, a patient should be able to control his or her sensitive medical information in terms of authorizing who can access the information and what information can be retrieved. In this regard, the Parkerian Hexad fails to address information authority.

2.5.3. Summary

In Section 2.5, information security models were investigated for sensitive information systems evaluation. As a result of reviewing the CIA Triad and the Parkerian Hexad, the following findings can be presented:

- The CIA Triad is not suitable for modern information systems.
- The Parkerian Hexad fails to address privacy concerns, and it is not suitable for modern sensitive information systems, since the lack of assessment properties (non-repudiation and authority).

As a result of these findings, it would appear that a new security model is required for sensitive information assessment. The new model should contain the attributes of authenticity, authority (AA), integrity (IN), non-repudiation (NR), confidentiality (CO) and utility (UT). Availability is not essential for sensitive information security assessment, because, in an extreme scenario, such as national threats, the property of availability of sensitive information can be constrained for the public.

2.6. Conclusion

In this chapter, we investigated existing techniques to protect sensitive information through the three components of sensitive information systems *communication channel*, *user interface* and *sensitive information storage*. The security issues and concerns are highlighted in Table 2.7. The table summarizes the problems and challenges of sensitive information protection with the current approaches.

From the literature, it is apparent that the existing approaches are technically not able to protect sensitive information. For example, IPsec and SSL/TLL (Section 2.2.1) cannot protect sensitive information at rest while multicast communication (GDH, LKH and Iolus) (Section 2.2.2) can only secure multicast communication. Challenge response (CR) mechanisms, smart cards and biometrics (Section 2.3) cannot guarantee the security of *communication channel* and *sensitive information storage* while encrypting file systems and database approaches (Section 2.4) do not guarantee the security of *communication channel*.

Table 2.7. Problems in Sensitive Information Security.

Security Problems and Challenges		
Communication Channel	Unicast	<ul style="list-style-type: none"> • Master key secrecy (long-term shared key) • Session key distribution • Public key secrecy
	Multicast	<ul style="list-style-type: none"> • Privacy protection • Group key unicast secrecy (long-term shared key)
User Interface		<ul style="list-style-type: none"> • Group authentication and authorization • Long-term shared key (knowledge factor) • Privacy and accuracy concerns (property factor) • Lifetime compromise (possession factor)
Sensitive Information Storage		<ul style="list-style-type: none"> • Encryption key secrecy (long-term shared key) • Public key secrecy • Privacy protection • Dynamic sensitive information ownership • Group access privilege management issue

Moreover, in addition to the above limitations, each approach can only address one particular security aspect. The lack of a complete architecture for protecting sensitive information therefore results in an ad hoc, rather than an integrated, approach. However, the employment of long-term keys and public keys constrains the security of existing approaches. Issues, such as group sensitive information sharing (*communication channel*), group authentication and authorization (*user interface*), sensitive information ownership (*sensitive information storage*) and privacy protection

(three components) also create challenges for the protection of sensitive information systems.

This chapter has also examined information security models for the security assessment of sensitive information systems. Due to their lack of assessment properties, the existing models, such as the CIA Triad and the Parkerian Hexad, are suited neither to evaluating the security of sensitive information systems nor to guiding the design of sensitive information architecture.

In response to these limitations, we therefore propose a novel security architecture for sensitive information systems (presented in Chapter 3) to tackle the problems and challenges. In addition, due to the lack of assessment properties within the existing information security models, a novel sensitive information security model is also presented in Chapter 3 to assess the proposed security architecture.

Chapter 3

Security Architecture for Sensitive Information Systems

Goals. Based on the previous discussion of the drawbacks of existing approaches in sensitive information protection, we concluded that the employment of long-term shared keys and public keys results in security threats and concerns in sensitive information systems. This is because the entropy (uncertainty) of the keys decreases when the keys are involved in frequent communication; they face a greater risk of exposure.

In addition, we also concluded, in Chapter 2, that the issues of privacy protection, sensitive information ownership, dynamic sensitive information sharing and group authentication and authorization create challenges for the protection of sensitive information systems. More importantly, currently, no complete solution exists to help protect sensitive information in the three components of *communication channel*, *user interface* and *sensitive information storage*.

The security comparison of symmetric keys in Section 2.1.1 has shown that dynamic keys provide stronger security than long-term shared, session, or one-time pad keys. Therefore, in this chapter we formally propose and define a new security

architecture as a complete solution for the protection of sensitive information systems. The architecture features dynamic keys to eliminate the security threats and concerns caused by the employment of long-term shared keys and public keys.

To introduce the security architecture, this chapter starts by defining and verifying dynamic keys and their properties (Section 3.1.1). It then moves on to analyse long-term shared keys (Section 3.1.2) and public keys (Section 3.1.3) in order to formally argue for the employment of dynamic keys in the security architecture. The arguments show that dynamic keys provide stronger security than long-term shared keys and public keys in sensitive information protection.

Using the arguments, dynamic key theory is applied in the security architecture (Section 3.2.3) to protect sensitive information. By combining the dynamic keys, a new group key agreement is proposed and defined formally (Section 3.2.4) to tackle the issues such as group sensitive information sharing and privacy protections. In the proposed security architecture, a new group authentication and authorization approach (Section 3.2.5) and a new sensitive information at rest protection approach (Section 3.2.6) are also defined to protect *user interface* and *sensitive information storage* in order to solve the dynamic membership, sensitive information ownership and group access privilege management issues in sensitive information systems.

Subsequently, a number of definitions are given to describe the components of the security architecture and relationships among the components. We state the security goals of the security architecture in order to show the security expectations for the proposed architecture. In Section 3.3, we propose a novel sensitive information security model that compensates for the lack of assessment properties in the CIA Triad and the Parkerian Hexad. (We later build the model in Chapter 5 to assess whether the

proposed security architecture satisfies the goals after the security statements of each component have been verified and proved.) The chapter concludes with Section 3.4.

3.1. *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 [AnAn01]. 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 SIS 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 (as discussed in Section 2.3.1) 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 method²⁸.

In accordance with the primary reasons for using dynamic keys in sensitive information protection, it is necessary to have an unambiguous and formal definition. In addition, the idea of dynamic keys is derived from TAN [Op96] and its successors mentioned in Section 2.3.1. Therefore, the notion of a one-way function [Me96] 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 $f : \{0,1\}^* \rightarrow \{0,1\}^*$ 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 [TaWe06]. Therefore, dynamic keys can be defined as follows:

Definition 3.1 (Dynamic Keys) 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). Concisely:

$$DK = \{f^i(\text{forms of } s) \mid i \in \square\} \quad (3.1)$$

where

$$\forall x, y (x \neq y), \neg(\exists f^i(x) = f^i(y)) \quad (3.2)$$

²⁸ Due to the limitations of long-term shard keys, once the key is compromised, the security of the generated token is breached.

The special one-way function dynamic key generation scheme [Ku05, KuLeSr05, LiZh04, RuWr02] 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 in the next section.

3.1.1. Cryptographic Properties

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 based on Definition 3.1 as follows:

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 special one-way function to generate DK . The properties are:

Theorem 3.1 (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 $x \in_R \{0,1\}^{|\mathcal{X}|}$, is negligible". Thus, it is computationally infeasible for an adversary to discover any dynamic key. \square

Theorem 3.2 (Former Key Secrecy) Former key secrecy ensures that an adversary, who knows a contiguous subset of used dynamic keys (say $\{dk_0, dk_1 \dots dk_i\}$), 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 $Pr(B_j | A) = \frac{Pr(B_j)Pr(A | B_j)}{\sum_{i=1}^n Pr(B_i)Pr(A | B_i)}$.

According to the argument in the proof of Theorem 3.1, 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 $Pr(A | B_j) = 0$, and $Pr(B_j | A) = 0$. Even if a contiguous subset of used dynamic keys becomes known, the security of subsequent fresh keys will not be affected. \square

Theorem 3.3 (Key Collision Resistance) Key collision resistance 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 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

Distribution²⁹ [Sc94]: $Pr(y) = \frac{\lambda^y}{y!} e^{-\lambda}$, $y = 0, 1, 2, \dots$. 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. \square

Theorem 3.4 (Key Consistency) Key consistency guarantees to produce sequential, consistent, dynamic keys DK , if given the same $f(\cdot)$ and an initial seed.

Proof: Given the same $f(\cdot)$ 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. \bar{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 $f(\cdot)$. From Theorem 3.3, the probability of the two distinct inputs, s_x and s_y , and the $f(\cdot)$ producing the same output is negligible. The probability of producing the same output by a given $f(\cdot)$ and two distinct seeds therefore converges towards 0.

²⁹ 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.

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 $f(\cdot)$, the two entities can generate the same set of dynamic keys. □

This section discussed the cryptographic properties of dynamic keys. Using these properties, the next section will argue that dynamic keys provide stronger security than other symmetric keys.

3.1.2. Dynamic Keys versus Symmetric Cryptography

Suppose that $G(\cdot)$ is a dynamic key generation algorithm and it satisfies dynamic key cryptographic properties. Let $DK = \{dk_1, dk_2, \dots, dk_n\}$ be the sequence of successive dynamic keys with m bits length; its key space is then 2^m . Also, as was proved in Theorem 3.3, the probability of dynamic key collision is negligible. Therefore an adversary must traverse the whole key space to determine the current dynamic key. In

other words, the probability of finding the key is $\frac{1}{2^m}$. This verdict supports Theorem 3.1, which demonstrates it is computationally infeasible to discover a dynamic key. Moreover, the initial seed, s_x or s_y , never participates in the information transaction. In this case, the uncertainty or entropy³⁰ of the seeds $H(seed)$ is its length because $length = \log(2^{length})$ ³¹. Hence, an adversary who knows a contiguous subset of used dynamic keys (say $\{dk_0, dk_1, \dots, dk_i\}$) cannot discover any subsequence dynamic keys. This argument supports Theorem 3.2.

In addition to the above discussion, as discussed in Section 2.1.1 and 2.3.1, a one-time pad has the same security level as dynamic keys. However, the distribution of the one-time pad is inconvenient and usually poses a significant security risk. Also, unlike modern ciphers, the pad length must be extremely long and the number of pads must be large. One time pads are rarely employed in modern sensitive information systems.

More to the point, the difference between session keys and long-term shared keys is their lifetimes. In terms of information theory, entropy of session keys and long-term shared keys is the same if, and only if, the key length is same. Also, the entropy of keys declines when the keys are involved in communication. We therefore discuss long-term shared keys and dynamic keys in some detail. By combining dynamic key cryptographic properties, the following result can be given.

³⁰ Entropy (refers to Shannon entropy) is a measure of the difficulty in guessing a random variable. It is measured in Shannon bits. For example, a random 10-letter English text has estimated entropy of around 15 Shannon bits, meaning that on average, it has 26^{10} possible combinations, and its Shannon bit is $H(26^{10}) = \log(26^{10}) \approx 15$.

³¹ In this thesis, log invariably means log to the base 2.

Theorem 3.5 Dynamic keys are more secure than long-term shared keys to protect communication or sensitive information.

Proof: Let K be key space for dynamic or long-term keys. Owing to dynamic key cryptographic properties and their features, by observing $K = dk_i$, the uncertainty is represented by $H(K = dk_i)$; thus, the entropy of any new dynamic key is:

$$H(K = dk_i) = \log(2^m) = m$$

For the long-term key $K = k_i$, assume the key size is l (normally, $l > m$). When the long-term key is fresh, the uncertainty of the key is:

$$H(K = k_i) = l$$

However, after using the key n times, the uncertainty of the key is reduced to:

$$H(K = k_i) = \log(2^l - n)$$

The entropy of long-term keys and dynamic keys is illustrated in Figure 3.1.

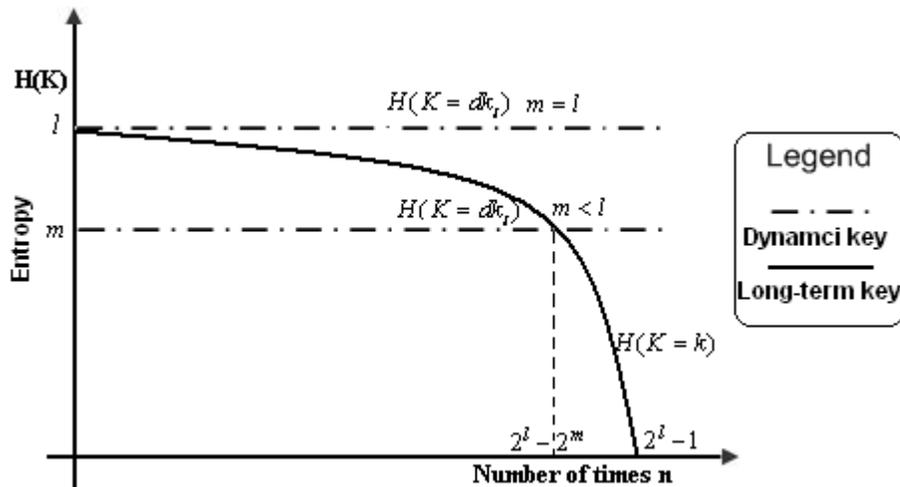


Figure 3.1. Entropy of Dynamic and Long-term Keys.

As shown in Figure 3.1, in the case that l is greater than m , after $2^l - 2^m$ times³² the entropy of the long-term key is the same as the dynamic key, and after $2^l - 1$ times the entropy of the long-term key is zero. In the case of $l = m$, the entropy of the long-term key declines by involving in communication. However, in both cases, the entropy of dynamic keys remains the same at value m . Therefore, dynamic keys are more secure than a long-term key and are better able to protect communication or sensitive information. \square

This section has argued fresh dynamic keys provide stronger security than long-term keys, including session keys. The next section will compare dynamic keys to asymmetric keys to show that asymmetric cryptography is insecure in sensitive information protection.

3.1.3. Dynamic Keys versus Asymmetric Cryptography

According to Theorem 3.5, as the long-term keys are repeatedly used, the entropy of the keys converges towards 0; that is, $\int_0^{n=2^l-1} H(K = k_i) dn = 0$. However, the entropy of the dynamic keys remains the same due to the single-use nature and cryptographic properties of dynamic keys. Therefore, a corollary can be induced.

Corollary 3.1 If, and only if, $F(X)$ is a function of X , the entropy of $H(F(X) | X)$ is zero.

³²The entropy of the long-term shared key is $H(K = k_i) = \log(2^l - n)$, after using the key n times. Therefore, after the use of the key $2^l - 2^m$ times, the entropy of the long-term shared key is $H(K = k_i) = \log(2^l - (2^l - 2^m)) = \log(2^m) = m$. The entropy is the same as dynamic key. $H(K = dk_i) = \log(2^m) = m$.

Proof: Based on conditional information entropy [Gr90], we have:

$$H(F(X) | X) = H(F(X), X) - H(X)$$

and, for X and $F(X)$ with outcomes $x_i \in X$ and $f(x_i) \in F(X)$, according to the

definition of information entropy $H(X) = -\sum_{i=1}^n Pr(x_i) \log Pr(x_i)$, we have:

$$H(F(X) | X) = \left\{ -\sum_{i=1}^n Pr(f(x_i), x_i) \log Pr(f(x_i), x_i) \right\} - \left\{ -\sum_{i=1}^n Pr(x_i) \log Pr(x_i) \right\}$$

and, following the multiplicative rule of probability [Sc94], we have:

$$H(F(X) | X) = \left\{ -\sum_{i=1}^n Pr(f(x_i) | x_i) Pr(x_i) \log Pr(f(x_i) | x_i) Pr(x_i) \right\} - \left\{ -\sum_{i=1}^n Pr(x_i) \log Pr(x_i) \right\}$$

Since $f(x_i)$ is a function of mapping x_i to $f(x_i)$, hence, by giving x_i , the probability of working out that $f(x_i)$ is 1, such that $Pr(f(x_i) | x_i) = 1$. Therefore we have:

$$H(F(X) | X) = \left\{ -\sum_{i=1}^n Pr(x_i) \log Pr(x_i) \right\} - \left\{ -\sum_{i=1}^n Pr(x_i) \log Pr(x_i) \right\}$$

It follows that:

$$H(F(X) | X) = 0$$

Therefore, if $F(X)$ is a function of X , the entropy of $H(F(X) | X)$ is zero, and the proof is completed. \square

The corollary proves that a cryptosystem provides only computational security.

Thus, it can be extended to the following statement.

Theorem 3.6 Asymmetric cryptography in protecting sensitive information is insecure.

Proof: Suppose C is cipher text and M is plain text in an asymmetric cryptosystem. If f is the decryption function, we have:

$$f(C) = M$$

Since $H(F(X) | X) = 0$, let $X = C$, $F(X) = f(X)$, we have:

$$H(f(C) | C) = 0$$

Therefore, $H(M | C) = 0$. In other words, all the uncertainty of the plain text is stored in the cipher text, which is known. The proof is completed. \square

From Corollary 3.1 and Theorem 3.6 it can be concluded that an asymmetric cryptosystem provides only computational security. That is, given sufficient time and resources, any asymmetric cryptosystem is breakable. However, the security of using dynamic keys does not rely on cryptographic functions, but on their cryptographic properties.

Section 3.1.2 has identified that dynamic keys provide stronger security than long-term shared keys or session keys, and this section has proved that asymmetric keys are insecure in sensitive information protection. Therefore, in the next section, we will present a novel security architecture for sensitive information systems by applying dynamic key theory.

3.2. Security Architecture

We have previously discussed that dynamic keys provide stronger security than long-term shared keys and public keys in sensitive information protection. Therefore, in this section, we apply dynamic keys theory to propose a novel security architecture.

To present the security architecture, this section starts with an overview of the architecture and then gives a formal description of the architecture. Following that, engaged users and each component are defined formally. Finally, the expected security goals of the architecture are formalized in Section 3.2.10.

3.2.1. Security Architecture Overview

Security architecture (SecureSIS) consists of four “tangible” components (Figure 3.2): dynamic key management (DKM), user-oriented group key management (UGKM), authentication and authorization management (AAM) and sensitive information management (SIM), 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.

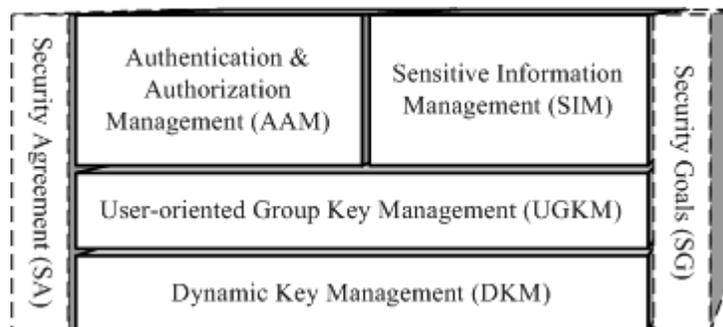


Figure 3.2. SecureSIS Core Component Overview.

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.

SG 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 3.2 (SecureSIS) Security architecture is defined as a union of the following sets:

$$SecureSIS = [U, AAM, UGKM, SIM, DKM, SA, Goals] \quad (3.3)$$

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 Figure 3.3.



Figure 3.3. Tangible Conceptual Architecture of SecureSIS.

3.2.2. Engaged Users

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 . Formally:

Definition 3.3 (Users U) U is a duple $[\omega, \varpi]$,

where:

- i) ω is a set of passive users in the system, that 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 .

- ii) ϖ is a set of active users in the system, that is vigorously and frequently joins and leaves the system. In SecureSIS, 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:

$$\omega \cap \varpi = \emptyset \quad (3.4)$$

In SecureSIS, U is able to send action requests, $AR = \{\text{Create, Delete, Retrieve, Append, Modify}\}$, to create, collate, annotate, modify, disseminate, use and delete authorized sensitive information. Also, ω is able to send a transformation request, $TR = \{\text{Tran_Req}\}$, to transform its state from a passive user to an active user in order to manage sensitive information, such as in the case of healthcare system, a doctor (passive user) can send TR to be a patient (active user) to share his/her medical records with other doctors (passive users), and vice versa. To protect the sensitive information, U needs to have the following properties:

- For every user, TR is a request to transform a user from one category to another.

$$\forall \omega_i \in \omega, TR(\omega_i) = \varpi_i, \text{ and vice versa} \quad (3.5)$$

- For every user, an AR allows U to take control of managing sensitive information.

The predicate is true if the U has privilege to access I .

$$\forall u_i \in U, AR(I_j \in I, a \in ACTION) = true \text{ iff } u_i \text{ has permission } a \text{ of } I_j \quad (3.6)$$

This section defined engaging users into passive users and active users, also the properties of engaging users were given. The next section, DKM component is formalized.

3.2.3. Dynamic Key Management

In this thesis, dynamic key management is proposed so that dynamic key theory safeguarding can be applied to *communication channel*, *user interface* and *sensitive information storage* in order to keep sensitive information I secure within SecureSIS. The security architecture employs two sets of dynamic keys for U and one set of dynamic keys for each component of SecureSIS.

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 communication 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 3.4 (Dynamic Key Management DKM) Dynamic keys management is a quadruple $[DK_x, DK_y, CDK, G(\cdot)]$,

where:

- i) DK_x is a set composed of dynamic data keys $\{dk_{x_i} \mid i \in \square\}$ of users for securing sensitive information storage. Given $u_n \in U$, the dynamic data key set for user u_n is:

$$DK_x = \{dk_{x_i}.u_n \mid i \in \square\} \quad (3.7)$$

- 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:

$$DK_Y = \{dk_{y_j}.u_n \mid i \in \square\} \quad (3.8)$$

- iii) *CDK* is a set composed of dynamic keys of each components for securing communication between DKM and AAM & SIM. Given $aam_m \in AAM$, $dkm_k \in DKM$ and $sim_n \in SIM$, the component dynamic key set for aam_m , dkm_k and sim_n is $\{cdk_i.aam_m \mid i \in \square\}$, $\{cdk_j.sim_n \mid i \in \square\}$ and $\{cdk_l.dkm_k \mid i \in \square\}$, respectively.
- iv) $G(\cdot)$ is a dynamic key generation scheme. It generates dynamic keys synchronously with U and other components in SecureSIS.

Note that DKM generates dynamic keys synchronously with all involved entities, and there is no key distribution between users and the DKM entity. There is one important exception: when users communicate with other components such as $aam_m \in AAM$ and $sim_n \in SIM$, a $dkm_k \in DKM$ generates corresponding dynamic keys of the users and securely transmits to the involved components³³. DKM is illustrated in Figure 3.4. The figure shows that the relationship of dynamic key sets and key generation scheme is not invertible.

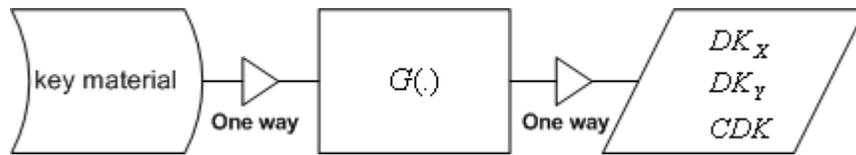


Figure 3.4. DKM Key Generation Flow.

This section defined the component of DKM. The next section will define UGKM.

³³ Details see Sections 4.2, 4.3 and 4.4.

3.2.4. User-oriented Group Key Management

Every user in SecureSIS is managed via this component, and it applies a hierarchical structure to secure multicast *communication channel*. The first key tree has been suggested in [WaHaAg97] for centralized group key distribution systems. As discussed in Chapter 2, centralized group key management is large group oriented, scalable and operation efficient, thus, this component adopts and extends the key tree T . 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. Formally:

$$T = \{root\} \cup SG \cup C \cup U \quad (3.9)$$

The component is characterized as follows:

Definition 3.5 (User-oriented Group Key Management UGKM) 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,

$$\begin{aligned} \forall \omega_i \in \omega, \exists ! vc_j \in VC : \omega_i \in vc_j \\ \forall \omega_i \in \omega, \exists \text{ at least one } vc_j : \omega_i \in \bigcup_{j \in N} vc_j \end{aligned} \quad (3.10)$$

- 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 $L \subset \omega$ for constructing virtual clusters and managing key operations.
- iv) $Alg(U)$ is a suite of algorithms that manages U join and leave rekeying operations.

This section defined the UGKM component, it delineated an active user can only belong to one virtual cluster, and only passive users can join clusters. Therefore, the privacy of active users is protected. In the next section, AAM will be defined by using DKM and UGKM to protect *user interface*.

3.2.5. 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 3.6 (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.
- 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 $eid_j \in EID$. Such checking defines the legitimacy of a user u_i with regard to the eid_j .

In Figure 3.5, U verifies itself to an $aam_m \in AAM$ in order to gain permission to execute a particular protocol. The protocol is performed if, and only if, the form of enciphered identities matches corresponding identities; otherwise, the process is terminated.

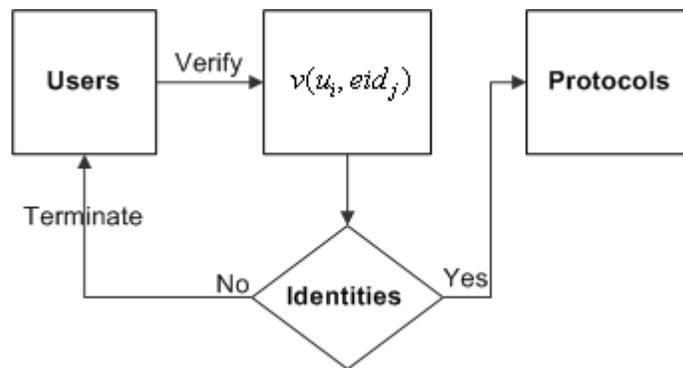


Figure 3.5. AAM Process.

This section defined the AAM component to protect *user interface*. In the next section, the SIM component will be defined to protect sensitive information at rest.

3.2.6. Sensitive Information Management

One of the most important technological challenges that SIS 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 I 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.

In addition, the usefulness of sensitive information at rest is a challenge as well for SIS. Since sensitive information must remain its usefulness, then it is useful for users. In the case of an emergency circumstance, the cryptographic key is lost. The form of the sensitive information is useless. Therefore, to overcome these challenges, the SIM component is formally characterized as follows:

Definition 3.7 (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),

$$CI = EDK \cup EI \quad (3.11)$$

where, EI is produced using dynamic data keys of sensitive information owner u_n ,

$$EI = \bigcup_{i, j \in \square} \{I_j\} dk_{xi}.u_n, I_j \in I \quad (3.12)$$

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:

$$EDK = \bigcup_{i,j \in \square} \{ \{ dk_{xi}.u_n \} dk_{xc}.u_n, h(EI_j) \}, EI_j \in EI \quad (3.13)$$

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. The relationship is illustrated in Figure 3.6. In addition, $h(EI_j)$ is used to ensure integrity of sensitive information I_j .



Figure 3.6. Relationship between EI and EDK.

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.

$$EL = \bigcup_{i \in \square} u_i, c_{n \rightarrow i}, c_{a \rightarrow i}, \{ dk_{xc}.u_i \} K_{combine} \quad (3.14)$$

where $K_{combine}$ is a combination key of leaders l_n and l_a , which represent cluster $c_{n \rightarrow i}$ and $c_{a \rightarrow i}$ respectively:

$$K_{combine} = h(h(n, dk_{yj}.l_n), h(a, dk_{yk}.l_a)) \quad (3.15)$$

- 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 .

This section defined the SIM component to protect sensitive information at rest. It also considered emergency situations, such as the loss of cryptographic key and the change of sensitive information ownership. In the next section, the structure of SecureSIS is given to show the relationship of four “tangible” components.

3.2.7. Structure of SecureSIS

SecureSIS is split into several administrative areas. Each area has a local secure group controller (LSGC) associated with a subgroup ($sg_i \in SG$) 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 structure of SecureSIS is shown in Figure 3.7.

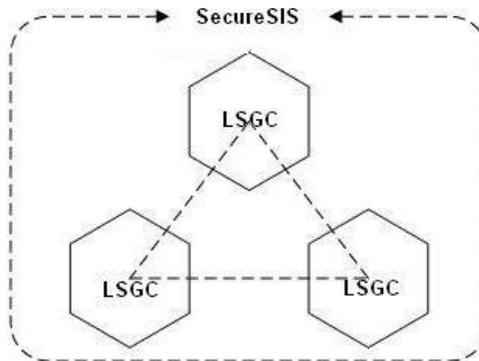


Figure 3.7. The Structure of SecureSIS.

A LSGC comprises an object of AAM, DKM and SIM, respectively. Formally:

$$LSGC = \{aam_m \in AAM, dkm_n \in DKM, sim_k \in SIM\} \quad (3.16)$$

3.2.8. Entities Belonging

According to Equation 3.10, it can be seen that an active user $\varpi_n \in \varpi$ belongs only one virtual cluster while a passive user $\omega_n \in \omega$ can join multiple virtual clusters. The reason behind this is that active users share sensitive information with passive users. For privacy reasons, active users only belong to virtual clusters, and only one active user is allowed in one virtual cluster. However, passive users can belong to multiple clusters and virtual clusters. This ensures that sensitive information of the active user is secure and not disclose to other active users. Precisely:

$$\begin{aligned} \forall \varpi_i \in \varpi, \exists ! vc_j \in VC : \varpi_i \in vc_j \wedge \neg(\exists c_k \in C), \varpi_i \in c_k \\ \forall \omega_i \in \omega, \exists \text{ at least one } vc_j \in VC \wedge \exists \bigcup_{k \in N} c_k \in C : \omega_i \in vc_j \cup c_k \end{aligned} \quad (3.10A)$$

In addition to sensitive information, each critical information object has an owner ($u_n \in U$ or a group of users $\{u_i \mid u_i \in U\}$) and the object is fully controlled by the owner. Precisely:

$$\begin{aligned} \forall I_j \in I, \exists u_i \in U : u_i := I_j \\ u_i := I_j : AR(I_j, *) \equiv true \end{aligned} \quad (3.17)$$

Meanwhile, $:=$ stands for possession and $*$ is a wildcard for action request. In the scenario of that sensitive information owner permanently leaves the system, the sensitive information is orphan sensitive information, referring to *EL* (Equation 3.14), new ownership³⁴ is assigned to the orphan information. Precisely:

$$\forall I_j (\neg \exists u_i (u_i := I_j)) \Rightarrow c_{n \rightarrow i} := I_j \quad (3.18)$$

³⁴ The ownership will be assigned to the nominated cluster; details in Chapter 4.

This section clarified the entity belongings. In the next section, one of the “intangible” components (SA) will be defined to show the relationship between sensitive information and users.

3.2.9. Security Agreement

In SecureSIS, the security agreement 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 security agreement classifies sensitive information into a number of levels following information classification³⁵, and then assigns access rules to each information object. Formally:

Definition 3.8 (Security Agreement SA) Let SR be a set of security rules (for example ACCEPT, DENY and NEGOTIATE). A security agreement is a triple, $[I, UL, \lambda]$,

where:

- i) I is a set of sensitive information objects labelled with information security classification.
- ii) UL is a set of user lists and each list consists of a number of users.
- iii) $\lambda : I \cup UL \rightarrow SA$ is security agreement mapping.
- iv) SA:ACCEPT is a access control flag that allows sensitive information to be disclosed.
- v) SA:DENY is a access control flag that restricts users from accessing sensitive information.

³⁵ This refers to Table 1.1. Sensitive information levels of classification in the US.

vi) SA:NEGOTIATE is an access control flag. Initially, it limits users (as in DENY) and then allows users to negotiate with owners of sensitive information to request permission.

It is defined that each critical information object has security rules (SR), and every security rule is assigned to a number of UL. Users in the list of NEGOTIATE are allowed to request permissions of accessing and sharing sensitive information from information owners. In this section, the agreement has been defined for SecureSIS. In the next section, the security expectations will be attempted to give in order to ensure the proposed SecureSIS satisfies the security requirements of sensitive information.

3.2.10. Goals of SecureSIS

When designing security architecture for SIS, 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 goals of SecureSIS. Formally:

$$Goals = \{UIG, CCG, SISG\} \quad (3.19)$$

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

$$\begin{aligned} &\forall u_i \in U \forall I_j \in I (u_i \text{ CanProve } u_i \text{ to SecureSIS} \wedge \\ &u_i \text{ CanProve } AR(I_j, *) \equiv \text{true to SecureSIS}) \\ &\Rightarrow u_i := I_j \end{aligned} \quad (3.20)$$

$$\forall u_i \in U (\text{SecureSIS CanProve Genuine to } u_i) \quad (3.21)$$

The Equation 3.20 implies that u_i must be able to prove to SecureSIS that it is a legitimate user, and also u_i must be able to prove that u_i can have privileged access to information I_j . According to Definition 3.7, sensitive information is enciphered with dynamic data keys of owners. Therefore, without proper privileges to information I_j , u_i cannot understand the information, even though u_i has the information I_j in the form of $ei_j \in EI$.

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

$$\forall u_i \in U \exists I_j \in I (\text{iff } u_i := I_j \Rightarrow u_i \text{ CanVerify } I_j \text{ is Genuine}) \quad (3.22)$$

$$\forall I_j \in I (\text{SecureSIS CanVerify } I_j \text{ is Genuine}) \quad (3.23)$$

Equation 3.22 indicates that if, and only if, u_i satisfies Equation 3.20, u_i is able to verify the authenticity of received I_j . On the other hand (Equation 3.23), SecureSIS is able to prove and verify that received information, I_j , is genuine.

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.

$$\forall EI_j \in EI \exists u_i \in U (\text{iff } u_i := EI_j \Rightarrow u_i \text{ CanUnderstand } I_j) \quad (3.24)$$

According to Equation 3.6, the predicate is true only if u_i has privilege to I_j . Because sensitive information is stored as cipher form³⁶, Equation 3.6 is transformed into:

$$\forall u_i \in U, AR(EI_j \in EI, a \in ACTION) = true \text{ iff } u_i \text{ has permission } a \text{ of } EI_j \quad (3.25)$$

Sensitive information in storage is consequently only disclosed and understandable to legitimate u_i with proper permissions.

This section attempted to give the security expectations of SecureSIS. In the next section, a novel sensitive information security model will be presented in order to tackle the lack of assessment properties in existing information security models.

3.3. Sensitive Information Security Model

The defined security architecture should express the means for sensitive information protection. As the models discussed in Section 2.5 – the CIA triad and the Parkerian hexad – have limitations, they are not a valid basis for sensitive information protection.

In this section, a comprehensive new information security model is presented that solves the problems of the existing models. Five core elements discussed and argued which are all essential to sensitive information security: authenticity and authority, integrity, non-repudiation, utility and confidentiality, are used to replace the CIA triad and Parkerian hexad in the new security model. By defining the new security model, we introduce the logic associated with each atomic element to evaluate SecureSIS. In addition, the proposed security model can be used as a guide for designing SIS.

³⁶ Refers to Equation 3.12.

3.3.1. SecureSIS Pentad

The SecureSIS pentad is a set of five elements of information security with attributes: authenticity & authority (AA), integrity (IN), non-repudiation (NR), confidentiality (CO), and utility (UT). These attributes of sensitive information are atomic in that they are not broken down into further constituents and they are non-overlapping in that they refer to unique aspects of information. Any information security breach can be described as affecting one or more of these fundamental attributes of information. Formally:

Definition 3.9 (SecureSIS pentad) The SecureSIS pentad is a sensitive information security model. It comprises five elements to guarantee the security of sensitive information.

$$SecureSIS\ pentad = \{AA, IN, NR, CO, UT\} \quad (3.26)$$

where:

- i) AA is the act of confirming provenance and identity as genuine and the act of delegating shared information. It measures the security of verification of individuals and group users. It also measures the security of access control in terms of privacy protection of individuals. In short, the use of AA is the measurement of user interface and sensitive information storage.
- ii) IN refers to the validity of data. It measures the security of communication channel.
- iii) NR is an atomic element measuring that a user in a dispute cannot repudiate, or refute the validity of what has done in the system.

- iv) CO is the property of preventing disclosure of information to unauthorized individuals or group users in SIS. It is an overall measurement of communication channel, user interface and sensitive information storage.
- v) UT means usefulness for accessing and sharing sensitive information to authorized users. It measures the usefulness of sensitive information when its ownership is changed or when owners of critical information lose the decryption key or in an emergency.

The SecureSIS pentad is used to evaluate the security of SIS as a basis of assessment. It is applied to SecureSIS in order to validate its security in this thesis. To build a model, the five security attributes are used to satisfy the Goals of SecureSIS. The relationship of the SecureSIS pentad to SecureSIS is shown in Figure 3.8.

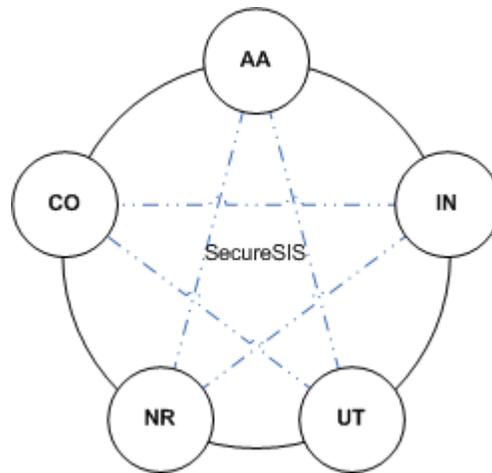


Figure 3.8. SecureSIS Pentad.

3.3.2. Authenticity & Authority (AA)

In information security, authentication is the process of verifying a claim made by an entity, while authorization is the process of verifying that an authenticated entity has the authority to perform a certain operation. Authentication, therefore, must precede

authorization. Since authorization cannot occur without authentication, thus, to assess the security of SIS, authenticity and authority are both defined in the SecureSIS pentad

Authentication is defined in relation to message authentication and entity authentication. Message authentication provides the identity of the sender P of a message to a given recipient Q . Entity authentication provides an identification of an entity in a communication. An important difference between these two types of authentication is that message authentication is not limited to a certain time period, while entity authentication is limited to the duration of the communication over interval $[t_0, t_1]$. Therefore, to achieve proper message authentication, the requirement must be satisfied as follows:

$$\forall I_i \in I, P := I_i : \frac{P \rightarrow Q : I_i, token}{Q \text{ believes } P \text{ said } I_i} \quad (3.27)$$

For entity authentication, precisely:

$$\int_{t_0}^{t_1} \frac{P \text{ claims to } Q}{Q \text{ believes } P} dt \quad (3.28)$$

Authorization is the concept of allowing only permitted users access to resources. More formally, authorization is a process that protects sensitive resources by only allowing a granted authority to use them. In other words, unauthorized access is restricted. Precisely:

$$\forall I_i \in I, P := I_i : \frac{\text{iff } Q : AR(I_i, *) == false}{Q \neq I_i} \quad (3.29)^{37}$$

Authenticity and authority determine the security of *user interface* in SIS. The combination of Equations 3.20 to 3.25 describes that sensitive information must only

³⁷ \neq means it is not the case of $:=$.

be disclosed to genuine SIS and legitimate users with proper permissions. As one of SecureSIS goals, AA is able to be applied to evaluate UIG³⁸.

3.3.3. Integrity (IN)

The integrity of critical information prevents an unauthorized user from altering the asset in a *communication channel* or *sensitive information storage*. It is “completeness, wholeness and readability of sensitive information and quality unchanged from a previous state”[Pa98]. Concisely, it is the assurance that sensitive information is consistent, correct, and accessible (shown as Figure 3.9).



Figure 3.9. Sensitive Information Integrity Triangle.

According to [CIWi87, Pa98], sensitive information integrity is essential to SIS, and is able to promise information consistency and accuracy; thus, it is defined in the SecureSIS pentad. In order to achieve sensitive information integrity, any sensitive information transmitted between entities via *communication channel* must be consistent. It is also compulsory for SIS that sensitive information be kept correctly in *sensitive information storage*. Formally:

³⁸ UIG refers to user interface's goal.

$$\forall I_i \in I, \frac{P \rightarrow Q : I_i}{Q \text{ receives } I_i \wedge Q \text{ believes } I_i \equiv I_i} \quad (3.30)$$

$$\forall I_i \in I, \frac{P := I_i}{P \text{ believes } I_i} \quad (3.31)$$

Integrity ensures the security of two components in the sensitive information retrieval process mentioned in Chapter 1³⁹. The Equations above emphasize the goal of CCG⁴⁰ and SISG⁴¹; thus IN can be used to assess the goals partially.

3.3.4. Non-repudiation (NR)

Non-repudiation is the property that binds an entity to sensitive information. A complete non-repudiation service must ensure both non-repudiation of origin and non-repudiation of receipt [Zh01]. It is an essential element for SIS due to the limitations of using the CIA triad and the Parkerian hexad discussed in Chapter 2. Non-repudiation differs from authentication. The former provides evidence of an identity that can be shown to an adjudicator, while the latter assures only that the recipient is convinced of the identity of the sender. In addition, non-repudiation of receipt ensures the sender has evidence that the recipient received previously-sent information.

For the purpose of securing SIS to achieve non-repudiation, any transaction occurring in SIS needs to be verified and identified, comprising actions in *sensitive information storage* and transactions via *communication channel*. Also, it is mandatory that non-repudiation of receipt must be sent to the sender and recipient along with the evidence for the previous transaction. This bi-directional non-repudiation is named in SecureSIS pentad as mutual non-repudiation. Formally:

³⁹ Refers to Figure 1.1 and Section 1.3.1.

⁴⁰ CCG refers to communication channel's goal.

⁴¹ SISG refers to sensitive information storage's goal.

$$\frac{P \text{ believes } \text{fresh}(\text{sign}), P \text{ sees } Q \text{ performs an action with a sign}}{P \text{ believes } Q \text{ performs the action with the sign}} \text{ vice versa} \quad (3.32)$$

Mutual non-repudiation involves all three components of the sensitive information retrieval process. It guarantees undeniable proof of entity actions in SIS. Equation 3.32 is thus able to appraise the goals of SecureSIS.

3.3.5. Confidentiality (CO)

Confidentiality preserves authorized restrictions on sensitive information access and disclosure. Breaches of confidentiality involve authenticity, authority, integrity and non-repudiation. Consequently, it is the most important property among the SecureSIS pentad. Conceptually, confidentiality covers two properties: sensitive information confidentiality and privacy.

Sensitive information confidentiality ensures that critical information is not made available or disclosed to unauthorized users while privacy ensures that individuals control or influence their own sensitive information. For SIS, critical information must only be disclosed or accessible to authorized users (same as Equations 3.29, 3.30 and 3.31) and information owners must have fine-grain control over their assets. Formally:

$$\forall I_i \in I, P := I_i \wedge Q : AR(I_i, *) \equiv \text{false} \frac{P \text{ authorizes } Q}{Q : AR(I_i, *) \equiv \text{true}} \quad (3.33)$$

Confidentiality measures the security of all three components *communication channel*, *user interface* and *sensitive information storage*. It also covers privacy or secrecy of information owners. In addition, the scope of confidentiality overlaps AA

and IN. Consequently it can be used to evaluate the security of SIS and to assess the goals of SecureSIS.

3.3.6. Utility (UT)

Utility is the property that indicates the usefulness of sensitive information. While it is not one of the core principles of the CIA triad, nevertheless it is a core principle of sensitive information security. In the scenario of sensitive information ownership change or a lost unique encryption key, the sensitive information is still available, but in a form that is not useful. While the information's authenticity and authority, integrity and non-repudiation are unaffected, and its confidentiality is greatly improved, the information cannot be used. Therefore, utility is essential and is included in the SecureSIS pentad to measure the usefulness of sensitive information.

In the interests of achieving sensitive information utility in SIS, a change of sensitive information ownership must not affect the usefulness of sensitive information. In an emergency, where the owner of sensitive information is unable to manage the asset, the usefulness of sensitive information must not be compromised. Formally:

$$\forall I_i \in I, P := I_i \frac{\text{orphan } I_i \wedge P \text{ authorizes } Q}{Q := I_i} \quad (3.34)$$

$$\forall I_i \in I, P := I_i \frac{\text{emergency}}{P, Q := I_i} \quad (3.35)$$

Utility does not measure the security of *communication channel*, *user interface* and *sensitive information storage*, but it does impact security. Without the property of sensitive information utility, the goals of *communication channel*, *user interface* and *sensitive information storage* cannot be reached. Sensitive information utility is therefore used as a baseline security assessment.

3.3.7. Summary on the SecureSIS Pentad

The SecureSIS pentad consists of five atomic elements. It is important to identify the differences between these elements.

First, authenticity deals with entity verification while authority governs the legitimate entities that have permission to perform certain operations. Therefore both are combined as AA and used to assess the security of restricted, sensitive or valuable information in SIS. IN also deals with the intrinsic condition of information but it does not involve the meaning of information. In contrast, AA is concerned with genuineness and control of information. NR handles the evidence of entities for an adjudicator, while AA addresses the credentials of entities. CO deals with disclosure of sensitive information, whereas AA is more concerned with the process of preventing disclosure. Last, but not least, UT is a baseline security assessment, since without usefulness, all other properties are valueless.

Determining the appropriate security by applying the five elements depends on the four primary characteristics of information⁴². The five elements are unique and independent, and often require different security controls. For example, maintaining the IN of sensitive information does not necessarily mean that the information is valid; it only indicates the information's wholeness and completeness. Sensitive information can be invalid (that is, lacking authenticity), but it can be identical to the original, and thus possess IN. Similarly, maintaining the NR of sensitive information does not necessarily maintain its AA, and vice versa, as they are two different elements.

With the exception of CO, the elements can be violated without affecting the other elements. CO is an exception because the loss of CO can have a negative impact on

⁴² Primary characteristics refer to kind, representation, form and medium. See Section 1.2.1.

the other elements. The disclosure of sensitive information makes IN, AA and NR invalid, whereas UT remains the same. In contrast, the loss of UT can improve CO because the form of sensitive information becomes useless. The scope of the five atomic elements is illustrated in Figure 3.10. The figure shows UT as a baseline; with UT assured, the security of SIS can be guaranteed. The figure also shows how the scope of CO overlaps IN, NR and AA.

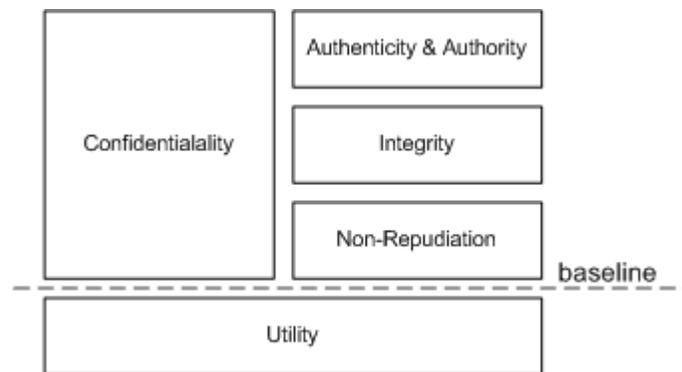


Figure 3.10. The Scope of Five Atomic Elements.

3.4. Summary

In this chapter, dynamic key theory has been formalized, and its cryptographic properties have been proved. Also, it has proved that dynamic keys provide stronger security than long-term shared keys and public keys. Therefore, SecureSIS is presented by applying dynamic key theory to govern critical information in three phases: *communication channel*, *user interface* and *sensitive information storage*. It has claimed that the proposed architecture employs following techniques to overcome the security threats and concerns:

- The use of two sets of dynamic keys to protect sensitive information in the process of sensitive information retrieving.

- The use of the proposed user-oriented group key management to deal with dynamic information ownerships, and to overcome confidentiality and integrity threats.
- The use of the proposed authentication & authorization management to conduct dynamic membership of groups and individuals to share or access sensitive information in order to handle authenticity and authority concerns.
- The use of the proposed sensitive information management to protect sensitive information at rest in order to thwart security threats of compromising credentials of SIS.

In addition to the proposed architecture, a security model - the SecureSIS pentad - was proposed to evaluate SecureSIS. It consists of five atomic elements that can be used to assess the security of the proposed security architecture. The model is able to evaluate the security of SecureSIS as shown in Table 3.1.

Table 3.1. Applied SecureSIS Pentad with the Proposed SecureSIS.

Components of Sensitive Information System	Goals of SecureSIS (Section 3.2.10)	SecureSIS Pentad	
		Elements	Baseline
User Interface	UIG	AA,NR,CO	UT
Communication Channel	CCG	IN,NR,CO	
Sensitive Information Storage	SISG	IN,NR,CO	

In the following chapter, each “tangible” object - DKM, UGKM, AAM and SIM - of SecureSIS is developed. The SecureSIS pentad is used as a guide to maximize the security of sensitive information in SIS.

Chapter 4

Security Architecture Components

Goals. In chapter 2, the technical background of this thesis was examined. In chapter 3, the limitations of employing long-term shared keys and public keys in SIS to protect sensitive information necessitated the formalisation of dynamic key theory. The cryptographic properties of dynamic keys provide stronger security than other keys. The SecureSIS architecture was consequently defined formally by applying dynamic key theory together with the expected security goals. The SecureSIS pentad (a sensitive information security model) was also proposed in order to assess the security of SecureSIS and guide in the design of security architecture.

In this chapter, we elaborate on all four “tangible” components of SecureSIS, Dynamic Key Management (DKM), User-oriented Key Management (UGKM), Authentication and Authorization Management (AAM) and Sensitive Information Management (SIM), guided by the SecureSIS pentad. In addition to giving a comprehensive understanding of the proposed SecureSIS, we demonstrate how to use dynamic key theory in DKM (Section 4.1), and then apply it to UGKM (Section 4.2), AAM (Section 4.3) and SIM (Section 4.4) to reach the security goals of

communication channel, user interface, and sensitive information storage. The chapter concludes in section 4.5.

4.1. Dynamic Key Management

In cryptography, dynamic key management is related to the generation, storage, synchronization, safeguarding, replacement and use of keys. Appropriate and successful dynamic key management is critical to the security of SIS. Therefore, the security of dynamic key management leads the security of sensitive information systems.

In this section, by applying dynamic key theory to SecureSIS, a security agreement is addressed for all entities in SecureSIS. The section finishes with a summary of desired cryptographic properties.

4.1.1. Dynamic Key Agreement

The cryptographic properties of dynamic keys help with security enhancement when protecting sensitive information. Each entity in SecureSIS must have shared dynamic key sets. The initial seeds and dynamic keys generation schemes take place in the function $f(.)$. In this thesis, the use and management of dynamic keys are emphasized.

Definition 3.4 shows two sets of dynamic keys employed (dynamic data key set DK_x and dynamic communication key set DK_y) to conduct the security of SecureSIS. DK_x is a set composed of dynamic data keys for securing *sensitive information storage*. DK_y is a set composed of dynamic communication keys for protecting *communication channel and user interface*. Note that DK_x and DK_y are only applied to users. Since involved “tangible” objects, such as $aam_m \in AAM$, $dkm_k \in DKM$ and

$sim_n \in SIM$, do not possess sensitive information, component dynamic keys (CDK) are used for protecting *communication channel* among these entities.

As discussed in Section 3.2.7, SecureSIS conducts the security of SIS via local secure group controllers (LSGC). Each LSGC administers sensitive information in and outbound, and consists of objects aam_m , dkm_k and sim_n . Also, all LSGC form a multicast group⁴³ to maintain group keys consistency. Meanwhile, dynamic key sets of users are synchronized among DKM. Therefore, once a user has shared key sets with SecureSIS and successfully registered⁴⁴, the user can join any local administration. 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 [NgWuLe09a]. The event can be performed via negotiating dynamic key counters⁴⁵ $\{Y_j \mid j \in N\}$.

⁴³ The multicast group is discussed in Section 4.2 UGKM.

⁴⁴ User registration - see Section 4.3.2 initialization protocol.

⁴⁵ Negotiating dynamic key counters can be performed via pervious successful dynamic communication key and current counter j .

4.1.2. Security Comparison

In this section (4.1), dynamic key management was introduced based on Definition 3.4. By applying the nature of dynamic keys⁴⁶, if the agreements (Section 4.1.1) are followed, the security of the proposed architecture is guaranteed. The proposed dynamic key management provide stronger security than other existing approaches in sensitive information protection, and comparable with *communication channel* (unicast and multicast), *user interface* and *sensitive information storage* protection. This comparison is presented in Table 4.1. The comparison criteria are selected based on the discussion in Chapter 2.

Table 4.1. Key Managements Comparison.

Criteria	Key Management Approaches				
	Communication Channel		User Interface	Sensitive Information Storage	DKM
	Unicast	Multicast			
Key Type	long-term	group long-term	long-term public	long-term public	dynamic
Key Distribution	yes	yes	yes/no	yes/ no	no
Key Lifetime	indefinite	moderate indefinite	indefinite	indefinite	once
Security Breach Detection	no	no	no	no	yes

Key Type refers to the type of keys employed in key managements. As discussed in Chapter 2, long-term (master) and public keys are mainly adopted in extant approaches of sensitive information protection. However, in the proposed security architecture, dynamic keys are adopted to in DKM. DKM has the advantage over others due to the nature of dynamic keys.

⁴⁶ The informal security comparison with other symmetric keys was presented in Section 2.1.1, and formal discussion (proof) on security comparison with symmetric and asymmetric keys was conducted in Section 3.1.2 and 3.1.3.

Key Distribution refers to the process of exchanging shared secrets for encryption. According to the discussion in Section 2.1.1, long-term shared keys involve key distribution, but not public and dynamic keys. Therefore, the risk of public and dynamic keys compromising is reduced.

Key Lifetime refers to the length of time the key can be used for encryption. As discussed and compared in Section 2.1.1 (Table 2.1), dynamic key is used only once, which make the key management stronger comparable with others.

Security Breach Detection refers to the ability of key managements in detecting the breach of sensitive information systems. As discussed in Section 2.2, 2.3 and 2.4, when the cryptographic keys (long-term shared and public keys) are breached, the sensitive information will be disclosed. In DKM, dynamic keys are employed, and each key is used only once. Any attempt to reuse an invalidated dynamic key can therefore be detected. Also, two sets of dynamic keys can guarantee that the breach of one set of dynamic keys does not compromise the security of sensitive information systems (DKM)⁴⁷.

The proposed DKM has advantages over other key managements (discussed in Chapter 2) in terms of key type, key distribution, key lifetime and security breach detection. In following sections, the details of the use of DKM are introduced.

4.2. User-oriented Group Key Management

Along with the popularity of group-oriented communication systems, sensitive information sharing has brought substantial convenience for users. However, sensitive information confidentiality is rising as an important issue for group members. To

⁴⁷ The formal proof is conducted in Section 5.1.1.

achieve confidentiality in group (multicast) communication, group key management for sensitive information systems requires key secrecy, backward secrecy and forward secrecy. In addition, it also requires flexible and efficient rekeying operations. Privacy for users in sensitive information systems remains a challenge for group key management.

In this section, user-oriented group key management (UGKM) is formally introduced. A security comparison is then conducted to show the advantages of the proposed group key management.

4.2.1. Key Tree Structure

As discussed in Section 2.2.2, since the drawbacks of the existing multicast *communication channel* approaches. The UGKM scheme (defined in Definition 3.5) 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.

In order to protect the privacy of individuals in sensitive information systems, UGKM categorizes group members U into active users⁴⁸ ω and passive users ω . Also, several new concepts are introduced. Meanwhile, each virtual cluster is formed by a passive user as a leader, and each virtual cluster is able to contain only one active user but one, or more than one, passive user.

In addition, in accordance with Equation 3.9 (key tree) and Definition 3.5, UGKM is a two-tier hybrid group key management that focuses on privacy protection and confidentiality of sensitive information. Figure 4.1 depicts the logical structure of

⁴⁸ Active & passive users refer to Definition 3.3.

UGKM. It is divided into two levels: the passive user level and the active user level. The passive user level consists only of passive users who participate in sensitive information sharing and accessing of other active users. As mentioned in Equation 3.5, if a passive user wants to share its sensitive information, the user must transform into an active user. The active user level employs a group key tree distribution scheme; it is formed by one active user and several passive users. Meanwhile, one passive user is promoted to leader to construct a virtual cluster. As defined in Definition 3.5 (Equation 3.10), each virtual cluster has only one active user, and a passive user can belong to multiple virtual clusters. The key management of this level is conducted by a contributory group key management scheme.

According to the structure in SecureSIS⁴⁹, each LSGC associates with a subgroup to manage sensitive information sharing, and also all LSGC together constitute a multicast group UGKM. Moreover, each LSGC consists of an object from AAM, DKM, and SIM. Therefore, each LSGC can simultaneously perform as a key server, an authentication and authorization server and sensitive information server.

⁴⁹ The structure refers to Section 3.2.7.

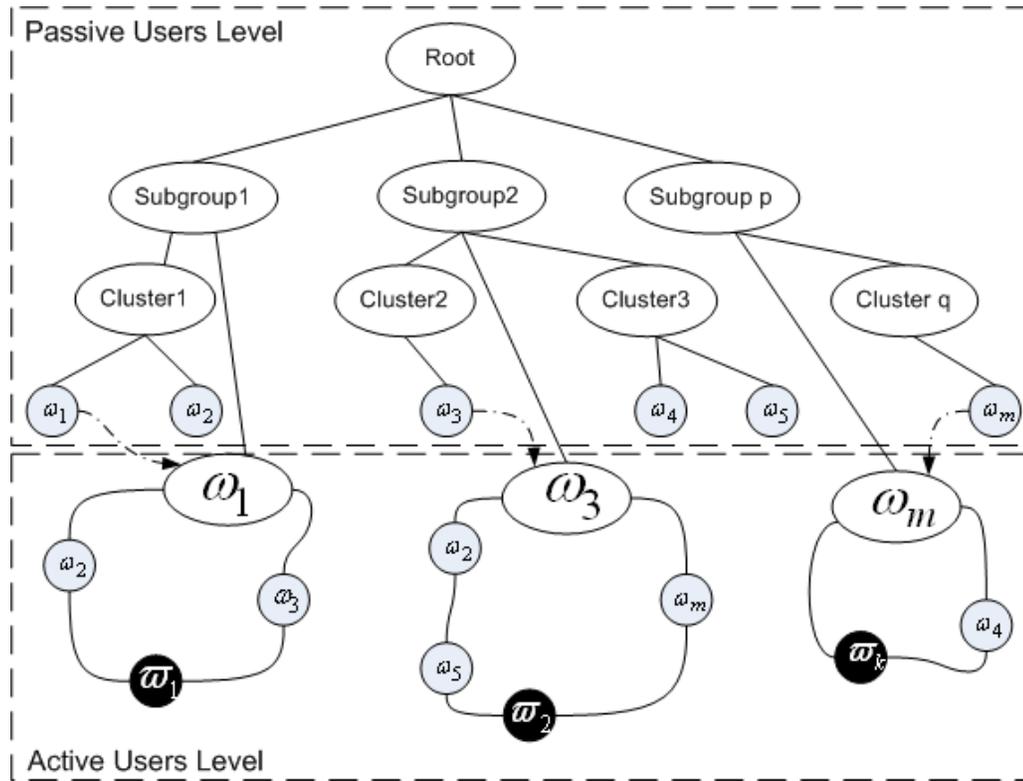


Figure 4.1. Logical Structure of UGKM.

UGKM is presented as tree-based group key management. However, when a $\omega_j \in \omega$ joins the system, one of $\omega_i \in \omega$ will reconstruct a dynamic virtual cluster under the subgroup. The logical structure of UGKM can be built in two steps:

- i) During the group initiation phase, a key tree is set up for the passive user level. Passive members are assigned into this level.
- ii) After the passive user level initialization is completed, active users are assigned into the active user level and a key ring is built for each active user. Meanwhile, a leader is selected from the passive users and assigned to the virtual cluster.

In this section, the key structure of UGKM was presented. In the next section, the security properties of UGKM will be given.

4.2.2. UGKM Cryptographic Properties

A comprehensive group key agreement solution must handle adjustments to group secrets subsequent to all membership change operations in the underlying group communication system. In order to guarantee the security of multicasting content, the proposed UGKM must have desired properties. As discussed in Section 3.1.1, key freshness is one of the most important requirements of dynamic keys management. Key freshness also applies to group key management as well together with following properties extended from Kim and Perrig et al. [KiPeTs04]:

- **Group Key Secrecy** – guarantees that it is computationally infeasible for a passive adversary to discover any group key.
- **Forward Secrecy** – guarantees that a passive adversary who knows a contiguous subset of previous group keys cannot discover subsequent group keys.
- **Backward Secrecy** – guarantees that a passive adversary who knows a contiguous subset of group keys cannot discover preceding group keys.
- **Key Independence** – guarantees that a passive adversary who knows any proper subset of group keys cannot discover any other group key.

Since the proposed UGKM is a hybrid group key management scheme, the passive user level employs group key tree management while the active user level adopts contributory group key management. The following notable features are associated with all protocols:

- Each passive group member receives a group key. The key is computed and distributed under the same protocol.

- Each active group member contributes an equal secret to a group key (virtual cluster key). The key is computed as a function of all current group members' secrets.
- For active group members, each secret is private and is never revealed to other members.
- All protocol messages are sequence-numbered.

This section gave the security properties of group keys. Next section, the generation of group keys in UGKM will be discussed.

4.2.3. Group Keys

Group keys are used to secure communications in SIS. The proposed hybrid group key management adopts contributory and group key tree agreements. Therefore, two algorithms are needed to generate group keys. For a passive user, group key tree management is applied, and a LSGC generates random group keys for members. However, for an active user, contributory key management is applied.

As defined⁵⁰, active users can only belong to virtual clusters. Thus, the algorithm for generating a key for a virtual cluster requires all involved entities to contribute their secrets. Moreover, referring to the dynamic key agreement, each individual has two sets of dynamic keys. One feature of dynamic communication keys, $dk_{y_j} \in DK_y$, is used as contributed secrets. The virtual cluster key generation algorithm is described in more detail as follows:

⁵⁰ It refers to Definition 3.5.

Assume a virtual cluster $vc_n \in VC$ consists of one active user and $m-1$ passive users (say $vc_n = \{\omega_n, involved \sum \omega_i\}$), and $f(x)$ is the special one way function used in dynamic key management.

- i) All users in vc_n form a network topology and ω_n is a leader and distributes a large prime number p to all members in vc_n .
- ii) Every user $u_i \in vc_n$ contributes a secret $s_i = f(dk_{v_j}, u_i) \bmod p$ to the leader.
- iii) ω_n gathers key materials and broadcasts intermediate values to other group users depending on the network topology in order to make all users $u_i \in vc_n$ generate a virtual cluster key $K_{vc} = f(s_1 \dots s_m) \bmod p$.

This section presented a virtual cluster key generation algorithm for UGKM. In the next section, member key operations, such as join and leave, will be discussed.

4.2.4. Member Join

Join is the procedure invoked by a user who wishes to become a member of a multicast group. 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. These scenarios are illustrated in Figure 4.2.

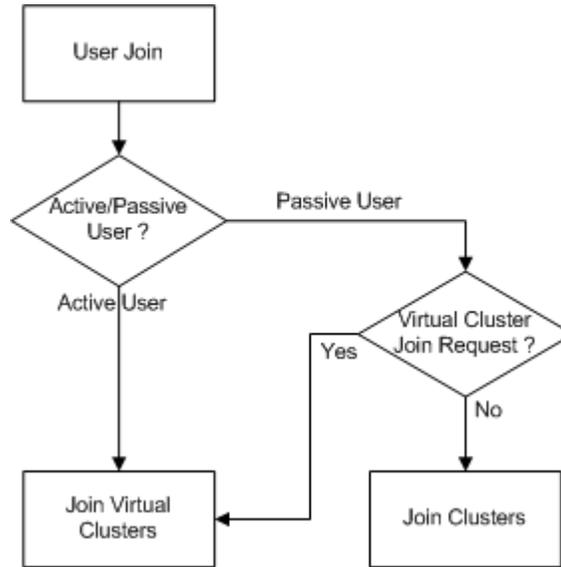


Figure 4.2. User Join Operations.

Active User Joins. When an active user (ω_1 in Figure 4.3) wishes to join the group, it applies the active user level key distribution agreement. According to Definition 3.5, it does not need backward secrecy and the join procedure starts with an active user join request.

- i) First, ω_1 contacts a LSGC, and the LSGC forwards the request to AAM for authentication⁵¹ via a secure unicast channel. Precisely:

$$\omega_1 \rightarrow LSGC : \{Active_user_join_request\}$$

- 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, ω_1 and ω_3).

⁵¹ Authentication & authorization will be discussed in Section 4.3, in member join section, message communication is discussed.

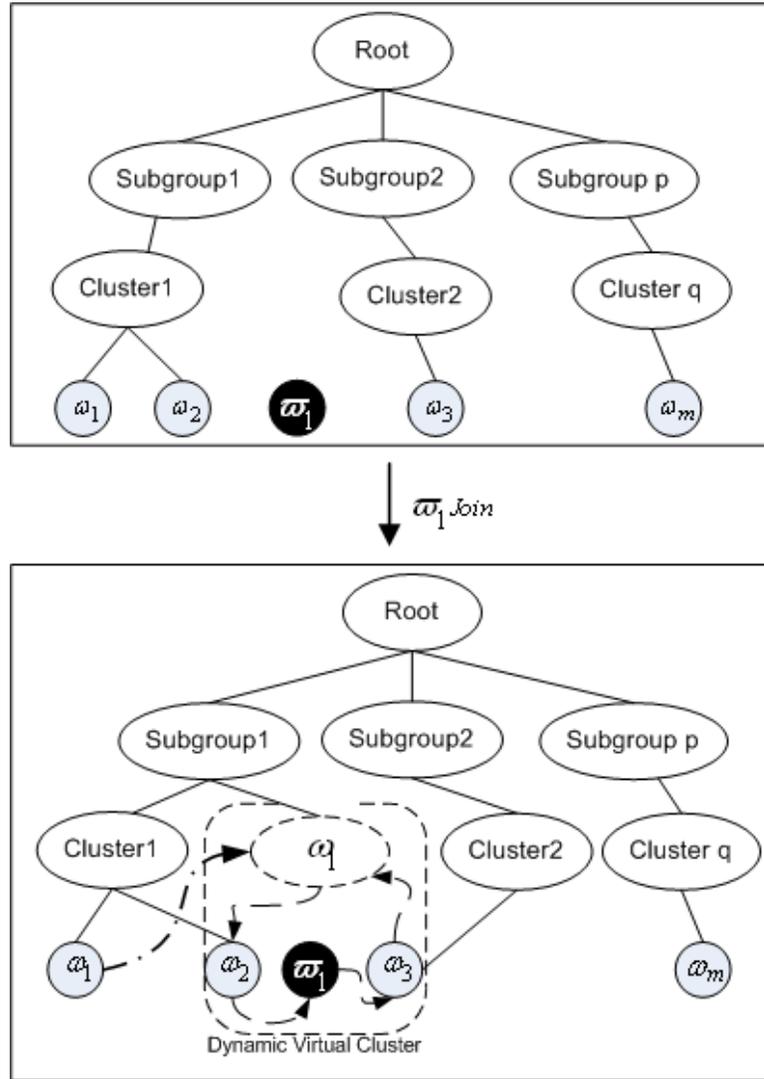


Figure 4.3. Active User Join.

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 algorithm⁵². Precisely:

$$leader(\omega_1) \rightarrow LSGC : \{K_{virtual_cluster}\} dk_{y_j} \cdot \omega_1$$

⁵² It refers Section 4.2.4.

When an active user joins, a new virtual cluster is created and a virtual cluster key is contributed by all group members. Also, a passive user is chosen as the leader of the created virtual cluster. The passive user (leader) has all relevant group keys (for example, ω_1 has subgroup1 key $K_{subgroup1}$ and root K_{root}). Furthermore, 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_4 in Figure 4.4) 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_4 contacts the nearby LSGC, and the LSGC forwards the request to AAM for authentication via a secure unicast channel. Precisely:

$$\omega_4 \rightarrow LSGC : \{Passive_user_join_request\}$$

- ii) After successful verification, the LSGC updates group keys for backward secrecy (for example, ω_4 is assigned to cluster 2, c_2 ⁵³). Precisely:

$$\forall \omega_j \in c_k, k \neq 2, LSGC \Rightarrow \omega_j : \{Join_key_update\}$$

$$\forall \omega_i \in c_2, LSGC \Rightarrow \omega_i : \{K_{new_root}, K_{new_subgroup2}, K_{new_cluster2}\} K_{cluster2}$$

$$LSGC \rightarrow \omega_4 : \{K_{new_root}, K_{new_subgroup2}, K_{new_cluster2}\} dk_{y_j} \cdot \omega_4$$

⁵³ c_2 refers to Section 3.2.4 (Equation 3.9).

When a passive user joins the group, it triggers a group key tree management scheme and a rekeying operation is incurred. However, as discussed in Chapter 2 (Section 2.2.1 and 2.2.2), in order to overcome the security threats of long-term shared secrets between individual members and the system, in UGKM, dynamic communication keys are used for replacing the long-term shared keys to secure *communication channel*. The same process applies to passive users joining multiple and virtual clusters.

Passive User Joins Existing Virtual Cluster. If a passive user (ω_m in Figure 4.3) wants to join an existing virtual cluster vc_n , 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. Precisely:

$$\omega_m \rightarrow LSGC : \{Passive_user_join_virtual_cluster_request\}$$

- ii) After successful verification, a new virtual cluster key is generated by the leader and ω_m via the virtual cluster key generation algorithm. For example, it shifts one bit of the former virtual cluster key and the contributed secret of ω_m . Precisely:

$$s_m = f(dk_{y_j} \cdot \omega_m)$$

$$K_{new_virtual_cluster} = f(h(cyclic_bit_shift_of(K_{virtual_cluster})) \cdot s_m)$$

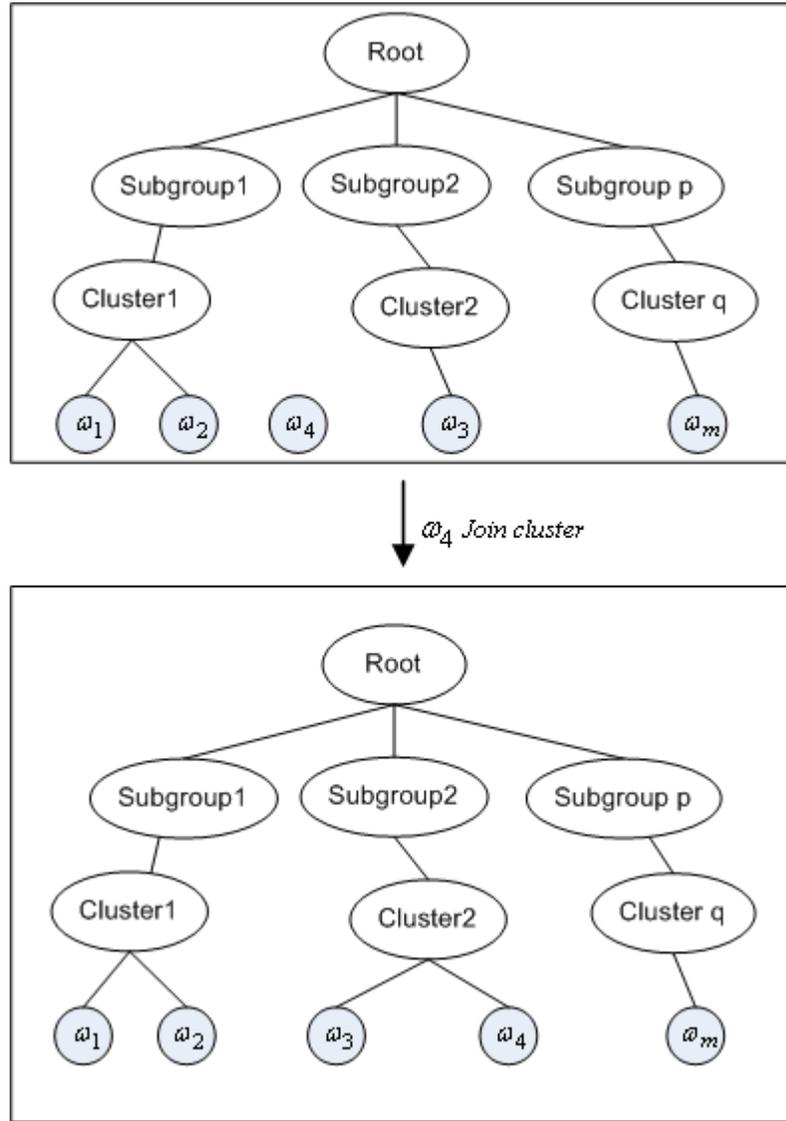


Figure 4.4. Passive User Join Cluster.

iii) Once the new virtual cluster key is generated the leader ω_1 broadcasts the new keys in the virtual cluster and informs the LSGC. Precisely:

$$\forall u_i \in vc_n, leader(\omega_1) \Rightarrow u_i : \{K_{new_virtual_cluster}\} K_{virtual_cluster}$$

$$leader(\omega_1) \rightarrow LSGC : \{K_{new_virtual_cluster}\} dk_{y_j} \cdot \omega_1$$

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. These details are discussed in Section 4.3.

4.2.5. Member Leave

Leave is the operation invoked by a group member who wishes to leave the multicast group. 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. These scenarios are illustrated in Figure 4.5.

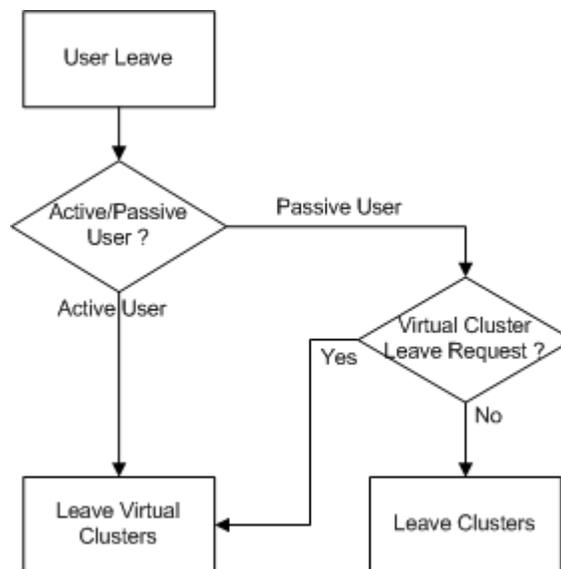


Figure 4.5. User Leave Operations.

Active User Leaves. Suppose an active user (ϖ_1 in Figure 4.3) wants to leave the system. It does not need forward secrecy, because virtual clusters are containers for

active users (Definition 3.5). When the active user leaves, the virtual cluster $vc_n \in VC$ is destroyed. The leave procedure starts with active user leave request.

- i) First, ω_1 sends a leave request to the virtual cluster leader (say ω_1), and the leader forward the request to the LSGC to remove the virtual cluster.

Precisely:

$$\omega_1 \rightarrow \omega_1 : \{Active_user_leave_request\}$$

$$\omega_1 \rightarrow LSGC : \{Active_user_leave_request\}$$

- ii) The leader then broadcasts to all members that the virtual cluster has been invalidated and is no longer available and that the virtual cluster key will be removed from each member. Precisely:

$$\forall u_i \in vc_n, \omega_1 \Rightarrow u_i : \{virtual_cluster_invalid\}K_{virtual_cluster}$$

The concept of virtual clusters focuses on active users. Each virtual cluster has only one active user and it is the existence of the active user that determines the virtual cluster. Virtual clusters allow active users to frequently visit their sensitive information and share their information with authorized passive users. When the active user leaves the virtual cluster, the cluster is destroyed.

Passive User Leaves Cluster. If a passive user (for example, ω_4 in Figure 4.4) wants to leave cluster 2 (say c_2), 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_4 sends a leave request to the LSGC. Precisely:

$$\omega_4 \rightarrow LSGC : \{Passive_user_leave_request\}$$

- ii) Upon receipt, the LSGC triggers a key update for other group members and unicasts new group keys to cluster c_2 users. Precisely:

$$\forall \omega_i \in c_k, k \neq 2, LSGC \Rightarrow \omega_i : \{Group_key_update\}$$

$$\forall \omega_i \in c_2, LSGC \rightarrow \omega_i : \{K_{new_root}, K_{new_subgroup2}, K_{new_cluster}\} dk_{y_j}.\omega_i$$

When a passive user leaves a cluster, it triggers a group key tree management scheme and a rekeying operation takes place. For forward secrecy, the new group keys are unicast to the involved cluster members in c_2 via dynamic communication keys to secure key materials. The security of UGKM is therefore guaranteed.

Passive User Leaves Existing Virtual Cluster. If a passive user (for example, ω_3 in Figure 4.3) wants to leave the virtual cluster vc_n , 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 vc_n member list and then updates LSGC. Precisely:

$$\omega_3 \rightarrow \omega_1 : \{Passive_user_leave_virtual_cluster_request\}$$

$$\omega_1 \rightarrow LSGC : \{vc_n - \omega_3\}$$

- ii) The LSGC then triggers the virtual cluster key generation algorithm to generate a new virtual cluster keys with existing members in vc_n .

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.

This section introduced member leave operations for passive and active users in UGKM. From a security aspect, group keys are threatened if no rekeying operation occurs in a particular period. The next section therefore introduces periodic rekeying operations to overcome this security threat.

4.2.6. 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. It also employs group key tree management. For example, if the last rekeying operation⁵⁴ occurred at time t and a passive user has a life cycle $[t_1, t_2]$, then $t_1 \leq t < t_2$, as illustrated in Figure 4.6. Let t_0 be the time period for the security parameter depending on security levels (requirements). The periodic rekeying algorithm is shown below:

- i) When the last rekeying operation occurred, the LSGC marks the time as t .
- ii) The LSGC monitors whether $t_2 - t \geq t_0$; if so it triggers a rekeying operation.

⁵⁴ Last rekeying event refers to a passive user join or leave operation.

iii) The LSGC then updates t to $t + t_0$ ($t \leftarrow t + t_0$). If $t < t_2$, the LSGC repeats step ii).

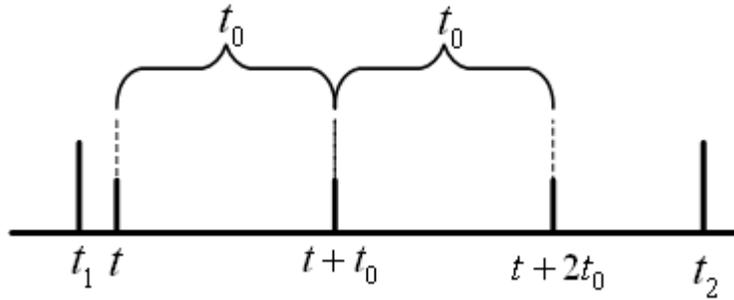


Figure 4.6. Periodic Rekeying Timeline.

4.2.7. Security Comparison

In this section (4.2), the UGKM key tree was introduced based on Definition 3.5. Its cryptographic properties were also discussed, together with four notable features. The group key generation algorithm and the rekeying operation were then introduced for UGKM. Meanwhile, all unique long-term shared keys between individuals and group key servers were replaced by dynamic communication keys; hence the security of a multicasting group is better than that of groups⁵⁵ which conduct the security of *communication channel* based on a long-term shared key.

Group members (users) are divided into two categories: passive users, who do not share their sensitive information but access the information of others; and active users, who share sensitive information with passive users. Also, each active user, when combined with passive users, constructs a virtual cluster. This prevents active users from accessing the sensitive information of other active users. These features of UGKM guarantee the privacy of each active user. Although performance is not one of

⁵⁵ Security is discussed in detail in Chapter 5.

major design goals of the architecture, this is also improved because active users joining the system and passive users joining existing virtual clusters do not trigger rekeying operations. The security features in tree-based group key distribution, contributory group key distribution and UGKM are shown in Table 4.2. The comparison criteria is based on the cryptographic key required for unicast, the cryptographic key required for multicast, the security of the rekeying operation and the security of privacy of individuals.

Table 4.2. Security Comparison of Group Key Management.

		Unicast key	Multicast key	Privacy Protection
Tree-based Group Key		long-term shared key	group key	no
Contributory Group Key		long-term shared key	contributory key	no
UGKM	Passive Users	dynamic key	group key	yes
	Active Users		virtual cluster key	

The Unicast Key is a cryptographic key in group key management that is used to secure the launch of information packets to a single destination. When a group key server unicasts new group keys or information to a user, a long-term shared key is employed to secure the communication in tree-based group key and contributory group key management. However, for our proposed UGKM, only dynamic keys are used. As shown in Table 4.1, UGKM provides more secure unicast communication than others.

The Multicast Key is a cryptographic key in group key management that protects the delivery of information to a group of destinations. For tree-based group key management, a random key (group key) is used to protect sensitive information. Since the key is used until a rekeying operation occurs, and the lifetime of a group key is that of a session key, the security of group keys is equivalent to session keys. For

contributory group key management, a contributory key is generated among all group members. Thus, the security of the key is weakened, and the security of keys lies between the security of long-term shared keys and the security of session keys. As defined in UGKM, active users nominate group members⁵⁶ and each member contributes a dynamic key to form a virtual cluster key. Also, the lifetime of a virtual cluster is much less than the lifetime of normal groups. Therefore, the security of a UGKM virtual cluster key lies between the security of session keys and the security of dynamic keys.

Privacy Protection is the ability of individuals in group key management to seclude or reveal their own sensitive information selectively. As discussed in Chapter 2 that neither tree-based nor contributory group key managements consider privacy protection for individuals. All members share information securely, and individuals cannot manage their own sensitive information in systems. However, UGKM employs virtual clusters to secure the sensitive information of active users, and allows only one active user in one virtual cluster. Compared to other key management schemes, UGKM provides greater privacy protection.

As discussed in this section, the proposed hybrid UGKM solved the privacy problem by adopting contributory key agreement, and also solved the scalability problem of contributory key agreement by using tree-based group key management. Also, by applying dynamic key theory to UGKM for unicast communication, it enhanced the security of UGKM. Therefore, unicast *communication channel* and multicast *communication channel* are fully protected. The next section will introduce AAM for the protection of *user interface*.

⁵⁶ Nominating members are discussed in details in Section 4.4.

4.3. Authentication and Authorization Management

Confidentiality is the most crucial requirement in security for sensitive information systems. Confidentiality relates to the authentication and authorization processes that are responsible for the security of *user interface*. These processes guarantee that sensitive information is only accessed by intended authorized users. Currently, a number of approaches discussed in Section 2.3 (authenticate and authorize users in sensitive information systems). These approaches are not sufficiently flexible to allow users to negotiate for access control of the resource. Nor do they focus on the privacy of the information owner, especially in health and military information systems.

Another problem is that these approaches do not provide verification for group members. Although group key management is a solution to provide secure authentication for group members, the approaches do not possess the ability to delegate access control for and from users in sensitive information systems. As mentioned, these approaches have a common limitation of employing long-term shared keys. Therefore, once the keys are exposed, a sensitive information system will be compromised.

In this section, a formal authentication and authorization management scheme is introduced. This scheme allows users to authenticate themselves to have fine-grain control over portions of their records. It focuses on privacy protection and offers secure authentication and flexible authorization for individuals and group members. Last, the proposed authentication and authorization management is compared with others to show its security advantages.

4.3.1. AAM Structure

As defined in Section 3.2.7, each LSGC consists of an object of AAM, SIM⁵⁷ and DKM. Meanwhile, the AAM object manages and performs system verification and access control. It allows a user or group users to share or access sensitive information of others. Also, it allows users to have fine-grain control over delegating access to portions of their information to others. Referring to Figure 3.5, the logical workflow of AAM can be described as follows:

- i) U (a user or group of users) requests sensitive information of other U (a user or group of users) from a LSGC.
- ii) After successful verification, the LSGC processes the request based on the security agreement (SA)⁵⁸ of sensitive information. In the SA:NEGOTIATE scenario, a particular protocol⁵⁹ is applied to U in order to retrieve the sensitive information.

As defined in Section 3.2.5, *Proto* consists of Initialization, Logon and AccessAuth, a suite of protocols⁶⁰. Initialization protocol is a preliminary setting for all users who are registered in the system. Logon protocol is a procedure used when a user wants to join the system. It is notable that joining a system is different from joining a group. Before a user can join a group, the user must be authenticated to the system. In other words, without successfully verifying with the system, a user cannot join a group. The AccessAuth protocol is an authentication process for users or group users delegating their sensitive information.

⁵⁷ SIM will be introduced in next section.

⁵⁸ The Security Agreement refers to Definition 3.8.

⁵⁹ The protocol refers to Section 4.3.4.

⁶⁰ Refer to Definition 3.6.

4.3.2. 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.
- ii) 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.)
- iii) 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:

$$EDI = \bigcup_{i=1}^{\square} \{id_i\}h(i, dk_{y_0}.u_i) \quad (4.1)$$

Meanwhile, 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, iris or DNA sequence⁶¹), a possession factor (smart card or token) or a knowledge factor (passwords).

Two unique random secrets could also be generated by the combination of the three factors. When multiple factors are combined to generate id_i and secrets, AAM can

⁶¹ Deoxyribonucleic acid (DNA) [Sa88] is a nucleic acid that contains the genetic instructions used in the development and functioning of all known living organisms and some viruses.

guarantee that only genuine users will join the system. AAM also guarantees non-repudiation, as users will not be able to deny system activity. In addition, because id_i is generated with multi factors of user u_i , and the id_i is only stored in one of DKM objects, the id_i can be considered as the signature of user u_i . For a higher security requirement, id_i and eid_i can be stored separately. Then the EDI can be denoted precisely:

$$EDI = \bigcup_{i=1}^{\square} \{id_i\}h(i, dk_{y_j}, u_i) \quad (4.2)$$

Meanwhile, j is an index of a corresponding dynamic communication key of user u_i . This means that when a user leaves the group, the EDI needs to be updated by regenerating it with a current dynamic communication key of the user u_i . In other words, when a passive user leaves a cluster algorithm or an active user leave algorithm is invoked, an EDI update event will be triggered in order to synchronize index j with a user for the next logon to the sensitive information system.

4.3.3. 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) When a user sends a request to $aam \in AAM$.

$$\forall u_i \in U, u_i \rightarrow aam : \{logon_request, h(i, dk_{y_{(j-1)}}, u_i)\} dk_{y_j}, u_i; i, j \in \square^*$$

ii) aam then requests a dynamic communication key of the user from dkm .

Note that the communication between aam and dkm takes place internally, although component dynamic keys are used to prevent internal attacks.

$$\begin{aligned} aam &\rightarrow dkm : \{key_request, i\}cdk_l.aam \\ dkm &\rightarrow aam : \{dk_{y_j}.u_i\}cdk_{l+1}.aam, l \in \square \end{aligned}$$

iii) 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. Note that, for a high security requirement, id_i can be stored in a different place. Then aam needs to send the deciphered value to verify id_i .

$$v(u_i, eid_j) \leftarrow id_i == \{eid_i\} \sim K ? true : false$$

iv) Subsequently, aam sends back a challenge to verify itself to the user.

$$aam \rightarrow u_i : \{logon_request, h(logon_request, dk_{y_j}.u_i)\}dk_{Y(j+1)}.u_i$$

v) 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.

$$eid'_i \leftarrow \{\{eid_i\} \sim K\}K'$$

4.3.4. 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. The mechanism is illustrated in Figure 4.7.

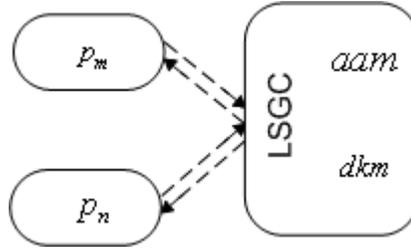


Figure 4.7. AccessAuth Protocol Logical Flow.

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 4.1 (Participant Classification PC) PC is a triple, $[P, T, \varsigma]$, 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 $c_i \in C \cup VC$ where $P \subseteq L \cup VL$. In other words, P is a leader of c_i (a cluster or a virtual cluster). Given $p_m, p_n \in P$ (that is, two participants), $I_n \in I$ (the information object of p_n), $aam \in AAM$ (the authentication and authorization object) and $dkm \in DKM$ (a dynamic key management object.), suppose p_m wants to share or access sensitive information I_n belonging to p_n . The protocol is described as follows:

- i) p_m generates a token $\tau = h(I_n - request, dk_{Y(j-1)} \cdot p_m)$ and sends it together with a request 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:

$$p_m \rightarrow LSGC : \{I_n - request, \tau\} dk_{Y_j} \cdot p_m$$

- ii) After understanding the request $I_n - request$ and verifying the token, aam in the LSGC checks for permission based on the security agreement⁶² of I_n . If p_m is on the list of SA:DENY on I_n , the request is rejected. In the opposite case, if p_m is the owner of the request information or on the list of SA:ACCEPT, the process moves to step iv. If neither of the above situations exist, p_m is on the list of SA:NEGOTIATE. When p_n is assigned to a group, the request is forwarded to p_n including the new token $\tau' = h(I_n - request, dk_{Y(i-1)} \cdot p_n)$ that was generated with a dynamic communication key of p_n . Precisely:

$$dkm \rightarrow aam : \{dk_{Y(j-1)} \cdot p_n, dk_{Y_j} \cdot p_n\} cdk_l \cdot aam$$

$$LSGC(aam) \rightarrow p_n : \{I_n - request, \tau'\} dk_{Y_j} \cdot p_n$$

- 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 $\tau'' = h(I'_n - response, dk_{Y_j} \cdot p_n)$. This token is sent back in the response message to aam to be ciphered by the next dynamic key. Note that because $I'_n - response \subseteq I_n - request$, p_n has full control of its own sensitive information:

⁶² Security agreement refers to Definition 3.8.

$$p_n \rightarrow LSGC : \{I'_n\text{-response}, \tau\} dk_{Y(j+1)} \cdot p_n$$

iv) When *aam* receives and verifies the token τ from p_n , p_m is able to retrieve the sensitive data I_n based on $I'_n\text{-response}$. If p_m has the status of $T : single$, the sensitive information will be unicast to p_m :

$$LSGC \rightarrow p_m : \{I_n, h(I_n, dk_{Y(j)} \cdot p_m)\} dk_{Y(j+1)} \cdot p_m$$

Otherwise, when p_m has the status of $T : group$, the sensitive information is multicast to the group where $p_m \in c_m$ and encrypted by the group key (either a cluster key or a virtual cluster key):

$$\forall u_i \in c_m; LSGC \Rightarrow u_i : \{I_n, h(I_n, K_{c_m\text{-}Or\text{-}vc_m})\} K_{c_m\text{-}Or\text{-}vc_m}$$

4.3.5. Security Comparison

In this section (4.3), a novel authentication and authorization management using a dynamic key-based UGKM is proposed to handle the security of *user interface*. The approach consists of Initialization, Logon and AccessAuth protocols.

A number of factors enhance the security of authentication. First, the use of dynamic keys in the authentication and authorization mechanism improves the security of SecureSIS. AAM also achieves group-to-group, group-to-individual, individual-to-group and individual-to-individual verification. The use of UGKM gives the proposed AAM the ability to handle dynamic member authentication. The security features of UGKM enable privacy protection of each individual and the AAM allows sensitive information owners to take full control on their assets by delegating access permissions. These strengths of our proposed AAM are detailed in Table 4.3 and compared to Kerberos and its successors based on the discussion in Chapter 2.

Table 4.3. Security Comparison of AAM to Kerberos and its Successors.

Comparison Criterion	Kerberos & its Successors	AAM
Authentication Factors	one or multiple	multiple
Processing Requirement	clock synchronization & availability of central server	no
Group Authentication	no	yes
Privacy Protection	no	yes
Access Negotiation	no	yes
Credentials Lifetime	predefined lifetime	one message
Key Distribution	shared session key	no
Trust	central server/ certificate authority	self
Keys	long-term shared key, session key and public keys.	dynamic keys

The table above summarizes the advantages of AAM against Kerberos and its successors [Er03, HaMe01, NeYuHa05]. It lists the features of AAM such as authentication factors, group authentication, privacy protection, access negotiation and keys. The security aspects of processing requirement, credentials lifetime, key distribution and trust are further discussed below:

Processing Requirement. As discussed in Chapter 2, Kerberos and its successors require clock synchronization among all entities in order to avoid replay attacks. However, AAM employs dynamic keys and is immune to replay attacks due to the nature of dynamic keys which use each key only once. In addition, while Kerberos requires the continuous availability of a central server, AAM does not because the LSGCs all form a multicasting group to maintain the consistency of group keys and other key materials. Should one LSGC fail, the authentication process can take place remotely.

Credentials Lifetime. In Kerberos and its successors, credentials have a pre-defined lifetime; a user can have one credential for one time period. However, in AAM, all

credentials (tokens) are used only once. Therefore, in order to perform actions, only genuine user has a legitimate credential.

Key Distribution. As described in Chapter 2, session keys are used in Kerberos to guarantee the security of information systems. Session key distribution is always involved. The security of such Kerberos-based information systems can therefore be compromised. AAM uses dynamic keys to conduct its security. Because of its cryptographic properties, no key distribution is necessary. In this regard, the security of AAM is better than Kerberos.

Trust. To use Kerberos, a trusted central server is necessary. All users need to request a credential to access information. However, in AAM, the combination of the use of a dynamic key and the challenge response mechanism⁶³ solves the trust problem because only genuine entities can produce a nonce token or key. The properties of dynamic keys guarantee that each entity only needs to trust its own dynamic keys.

In this section, AAM has been presented to protect *user interface* in SIS. By comparing it with other widely-used authentication techniques, AAM has been shown to provide stronger security and flexible access control. It can also deal with group authentication and authorization. In the next section, by applying DKM and UGKM, the management of sensitive information at rest will be introduced.

4.4. Sensitive Information Management

Protecting sensitive information is a growing concern for everyone around the world. Failing to protect sensitive information may result in high costs, such as losing

⁶³ The challenge response mechanism was discussed in Chapter 2. The use of challenge response refers to logon protocol.

customers in business and affecting investor confidence. Emerging technologies ensure that sensitive information protection is vulnerable to security threats. Especially in regard to protecting *sensitive information storage*, failure to secure the storage results in all sensitive information at rest being disclosed completely. In other words, no matter the sophistication of the security techniques employed, as soon as a breach of *sensitive information storage* occurs, sensitive information is disclosed.

A number of approaches (discussed in Chapter 2) have been proposed to protect *sensitive information storage*. However, the majority of solutions consist of prevention of unauthorized alteration of storage, prevention of unauthorized reading of storage areas and encryption of sensitive information. The use of long-term shared keys or public keys is a common technique for the above approaches; unfortunately, these keys have limitations. A better alternative is the use of dynamic keys that can eliminate the security threats associated with employing long-term shared keys or public keys.

In this section, we examine two approaches (database encryption and disk encryption) used by existing information systems in protecting sensitive information. By highlighting security concerns, sensitive information management is introduced formally according to Definition 3.7 (SIM). It integrates the dynamic data keys of users with sensitive information. The section finishes with a security comparison of SIM to other approaches to informally show the security of SIM.

4.4.1. 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 (Definition 3.7). Letting $ci_{i,u_j} \in CI$ be an object of CI, the structure of a SIM object $sim \in SIM$ is illustrated as in Figure 4.8.

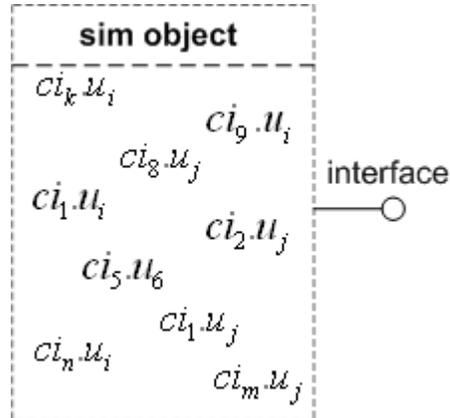


Figure 4.8. Structure of a SIM Object.

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 $sg_i \in SG$. 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.

In addition, sensitive information is enciphered by different dynamic data keys. Therefore, no encryption and decryption action is required between LSGCs while fragmented information needs to be transferred. Figure 4.9 depicts a scenario in which a user u_n has encrypted sensitive information stored in three SIM objects (three LSGCs).

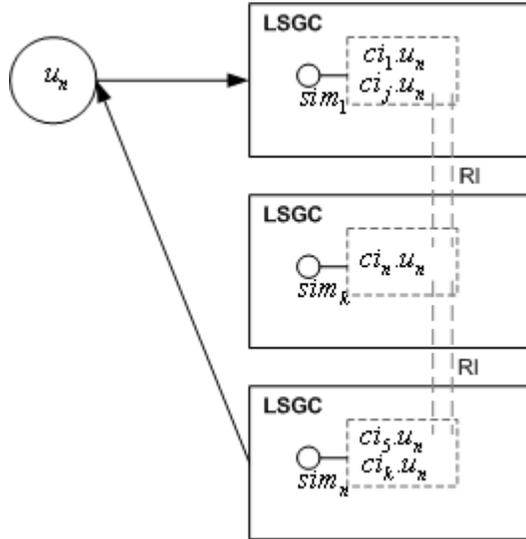


Figure 4.9. Retrieving Sensitive Information Flow Chart

Suppose user u_n (active user ω_n in a virtual cluster or ω_n in a cluster) has sensitive information stored in SIM objects sim_1 , sim_k and sim_n . Currently u_n has joined a LSGC which contains a sim_1 object. However, the sensitive information $ci_k.u_n$ in sim_n object is required. The retrieving sensitive information flow is described as follows:

- i) u_n requests $ci_k.u_n$ from LSGC (sim_1).
- ii) Since all LSGC constitute a multicast group, RI will return the location of $ci_k.u_n$.
- iii) With the owner's permission, LSGC (sim_n) unicasts the sensitive information⁶⁴ to u_n — this procedure refers to AccessAuth protocol step iv (Section 4.3.4).

⁶⁴ The sensitive information $ci_k.u_n$ refers to Section 4.3.4, where $I_n = ci_k.u_n$.

4.4.2. Data Operation

The definition of SIM shows there is no unprocessed plain information stored in the system. To describe the process of securing and managing sensitive information in SIM, an example is illustrated with a user $u_n \in U$ who owns sensitive information I_a , I_b and $I_c \in I$. Let $ci \in CI$ be stored information for the user u_n . First of all, before the enciphering, each data operation, such as data entry and data update, triggers SIM to build an index (RI) of the information for fast locating over large volumes of data. Then the sensitive information is enciphered and stored. The example shows how the dynamic data keys are used to integrate with sensitive information I .

Initial Stage, assume u_n has only information I_a , and based on SIM definition, the enciphered data is stored in a SIM object. Before it takes place, u_n needs to apply Initialization protocol with an AAM object. Then u_n has two sets keys: dynamic data keys and dynamic communication keys in order to manipulate sensitive information. The initial ci for u_n is shown in Figure 4.10.

$$\left. \begin{array}{l} \{I_a\}dk_{xi} \\ \{dk_{xi,u_n}\}dk_{xi} \end{array} \right\} ci$$

Figure 4.10. Initial Status of SIM.

Data Entry refers to user u_n having new critical information that needs to be processed and stored in the system. In this operation, it is assumed that u_n needs to process I_b and later I_c . The change of ci is shown in Figure 4.11.

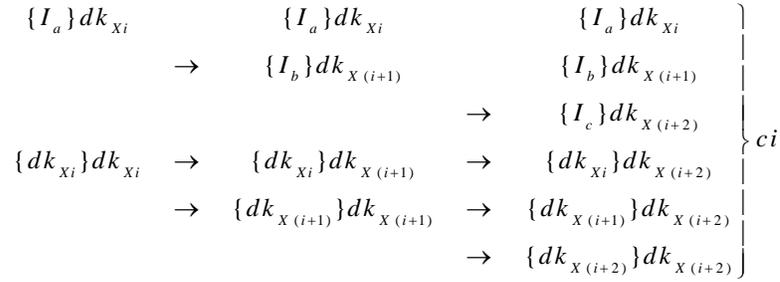


Figure 4.11. New Data Entry Status of SIM.

When a data entry event occurs, the following procedures take place. In Figure 4.11, \rightarrow is used to emphasize that the change of EDK and EI. Without symbol \rightarrow , no change has occurred.

- i) if u_n is not in the system, the user needs to authenticate with the system via the Logon protocol and join a group.
- ii) u_n sends sensitive information I_b to the system.

$$u_n \rightarrow LSGC : \{I_b, h(I_b, dk_{Y(j-1)}, u_n)\} dk_{Yj} \cdot u_n$$

- iii) The sensitive information I_b will be indexed and then enciphered with the current dynamic data key of the user.

$$sim \leftarrow ci : \{I_b\} dk_{X(i+1)}$$

- iv) Finally, all data keys will be rewrapped with the current data key.

Data Update refers to the manipulation of information to bring critical data up to date. In this scenario, it is assumed that the user wants to update I_b to I_b^* . The procedure is described as follows and the change of ci is shown in Figure 4.12.

- i) u_n needs to authenticate with the system via the Logon protocol and join a group.

- ii) Once successful log into a group (cluster or virtual cluster), u_n requests I_b via AccessAuth protocol. Meanwhile, $I_n = \{I_b\}dk_{X(i+1)} \wedge \{dk_{X(i+1)}\}dk_{X(i+2)}$.
- iii) After the sensitive information update, the update will invoke the data entry procedure.

$$\left. \begin{array}{l}
 \{I_a\}dk_{X_i} \\
 \{I_b\}dk_{X(i+1)} \\
 \{I_c\}dk_{X(i+2)} \\
 \{dk_{X_i}\}dk_{X(i+2)} \\
 \{dk_{X(i+1)}\}dk_{X(i+2)} \\
 \{dk_{X(i+2)}\}dk_{X(i+2)}
 \end{array} \right\} \rightarrow \left. \begin{array}{l}
 \{I_a\}dk_{X_i} \\
 \{I_b^*\}dk_{X(i+3)} \\
 \{I_c\}dk_{X(i+2)} \\
 \{dk_{X_i}\}dk_{X(i+3)} \\
 \{dk_{X(i+3)}\}dk_{X(i+3)} \\
 \{dk_{X(i+2)}\}dk_{X(i+3)}
 \end{array} \right\} ci$$

Figure 4.12. Data Update Status of SIM.

Data Deletion is an operation involving data erasure. When u_n wants to erase the sensitive information I_a , the procedure is listed as follows and the change of ci is shown in Figure 4.13:

- i) u_n needs to authenticate with the system via the Logon protocol and join a group.
- ii) Once successfully logged into a LSGC, u_n sends the request by using the AccessAuth protocol for step i) only.
- iii) The LSGC removes the sensitive information and correlative encrypted data key based on the request.
- iv) The remaining dynamic data keys will be rewrapped by the current dynamic data key of the user u_n .

$$\left. \begin{array}{l}
\{I_a\}dk_{X_i} \quad \rightarrow \\
\{I_b^*\}dk_{X(i+3)} \quad \rightarrow \quad \{I_b^*\}dk_{X(i+3)} \\
\{I_c\}dk_{X(i+2)} \quad \rightarrow \quad \{I_c\}dk_{X(i+2)} \\
\{dk_{X_i}\}dk_{X(i+3)} \quad \rightarrow \\
\{dk_{X(i+3)}\}dk_{X(i+3)} \quad \rightarrow \quad \{dk_{X(i+3)}\}dk_{X(i+4)} \\
\{dk_{X(i+2)}\}dk_{X(i+3)} \quad \rightarrow \quad \{dk_{X(i+2)}\}dk_{X(i+4)}
\end{array} \right\} ci$$

Figure 4.13. Data Deletion Status of SIM.

Data Retrieval. Data retrieval is a simple process in which u_n applies the AccessAuth protocol to fetch the critical data without modification. No data needs to be re-indexed and enciphered. Only dynamic data keys need to be rewrapped. The change of ci is shown in Figure 4.14.

$$\left. \begin{array}{l}
\{I_b^*\}dk_{X(i+3)} \quad \rightarrow \quad \{I_b^*\}dk_{X(i+3)} \\
\{I_c\}dk_{X(i+2)} \quad \rightarrow \quad \{I_c\}dk_{X(i+2)} \\
\{dk_{X(i+3)}\}dk_{X(i+4)} \quad \rightarrow \quad \{dk_{X(i+3)}\}dk_{X(i+5)} \\
\{dk_{X(i+2)}\}dk_{X(i+4)} \quad \rightarrow \quad \{dk_{X(i+2)}\}dk_{X(i+5)}
\end{array} \right\} ci$$

Figure 4.14. Data Access Status of SIM.

4.4.3. 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 ⁶⁵ is important. EL Update is an operation that updates the new nominated cluster $c_n \in C$ 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. Emergency access 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 $c_n \in C \cup VC$ 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 $\tau_n = h(n, dk_{y_{(j-1)}}.l_n)$.

$$l_n \rightarrow LSGC : \{request, \tau_n, h(request, \tau_n, dk_{y_{(j-1)}}.l_n)\} dk_{y_j}.l_n$$

- 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 $\tau_a = h(a, dk_{y_j}.l_a)$.

⁶⁵ Definition of EL refers to Definition 3.7.

$$LSGC \rightarrow l_a : \{request, h(request, dk_{Y(j-1)}.l_a)\} dk_{Yj}.l_a$$

$$l_a \rightarrow LSGC : \{response, \tau_a, h(response + \tau_a, dk_{Yj}.l_a)\} dk_{Y(j+1)}.l_a$$

- 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 .

$$LSGC \rightarrow l_n : \{I_n\} dk_{Xi}.u_n, \{dk_{Xi}.u_n\} dk_{XC}.u_n, \{dk_{XC}.u_n\} dk_{Xj}.l_n$$

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. The change is shown in Figure 4.15. Suppose user u_n owns sensitive information I_n . After u_n permanently leaves the system without removing I_n , the ownership will be changed to nominated cluster c_n .

$$\left. \begin{array}{l} \{I_n\} dk_{Xi}.u_n \\ \{dk_{Xi}.u_n\} dk_{XC}.u_n \end{array} \right\} \rightarrow \left. \begin{array}{l} \{I_n\} dk_{Xi}.u_n \\ \{dk_{Xi}.u_n\} dk_{XC}.l_n \end{array} \right\} ci$$

Figure 4.15. Ownership Change of Sensitive Information.

4.4.4. Security Comparison

In this section (4.4), sensitive information management is proposed based on Definition 3.7. This management scheme integrates dynamic data keys with the

sensitive information of users in order to protect *sensitive information storage*. It also proves a suite of data operations, consisting of data entry, data update, data deletion and data retrieval to manipulate sensitive information among group users. Moreover, the proposed component supports dynamic membership, which allows an authenticated cluster in an emergency to gain access to the sensitive information of users which would normally be inaccessible. The scheme also supports dynamic ownership, which allows an authenticated cluster to take control of “orphan” sensitive information created when the owner permanently leaves the system.

Because a dynamic data key encrypts only one record or file in the system, the security of sensitive information is maximized, even should *sensitive information storage* be breached. Also, by adopting dynamic keys in SIM, the privacy of sensitive information owners has been protected, because the owner manages assets with a current dynamic data key. Table 4.4 shows the security features of the proposed SIM compared to other existing security mechanisms for protecting *sensitive information storage*.

Table 4.4. Security Comparison of SIM to Other Approaches.

Comparison Criterion	Database Encryption (SQL)	Disk Encryption (IBM z/OS)	SIM
Security Technique	multilevel security to classify information	different keys to encrypt sensitive files	integrates dynamic keys with sensitive information
Security Key Type	long-term shared key	symmetric and asymmetric keys	dynamic keys
Privacy Protection	no	yes	yes
Dynamic Ownership	no	no	yes

Security Technique. Using a database approach, Microsoft SQL employs multilevel security to separate information based on its security classification. The technique is effective because not all data are visible to all users. However, the technique does not prevent internal attacks in the case of a database manager having permission to view all information.

Using a cryptography approach, IBM uses different symmetric keys to secure files and wrap the keys by users' asymmetric keys. The technique improves upon the security of the database approach, but a breach of the asymmetric key leads to the disclosure of sensitive information.

Using a SIM approach, dynamic keys are used integrated with sensitive information. Because of dynamic and former key secrecy, the security of the SIM approach is better than that of the cryptographic approach.

Privacy Protection. As mentioned in the discussion on security techniques, the database approach is susceptible to internal attacks and does not protect the privacy of sensitive information owners. In other words, an adversary with higher privileges is able to “oversee” the sensitive information of others. In contrast, the cryptography and

SIM approaches both provide a privacy protection mechanism by using different keys to encrypt sensitive information. Even a number of compromised encryption keys do not threaten all the sensitive information of a user.

Dynamic Ownership. Database and cryptography approaches do not take dynamic ownership into consideration. In contrast, the SIM approach provides dynamic ownership and membership operations to deal with emergency situations and the occurrence of “orphan” information in the system.

4.5. Summary

In this chapter, four “tangible” components of SecureSIS were proposed and formally described. A security comparison for each component with existing techniques was made. The comparisons show that the proposed security architecture (the four “tangible” components) is able to overcome the security concerns and minimise the threats to *communication channel*, *user interface* and *sensitive information storage*. In order to design the four components, the SecureSIS pentad was used as a guide. The proposed four components have contributed the following achievements:

- DKM-based UGKM enhances the security of SecureSIS and it allows sensitive information sharing among group members while protecting the privacy of individuals.
- AAM provides multifactor authentication and achieves high security and tight access control among individuals and group members based on DKM and UGKM. It also gives flexibility to sensitive information owners while protecting their privacy.

- SIM integrates dynamic keys with critical information to protect sensitive information. It guarantees that a breach of the credentials of one user cannot compromise the security of other users.
- The use of DKM, UGKM and AAM in SIM is able to solve dynamic information ownerships problem.
- The use of two sets of dynamic keys in SecureSIS is able to achieve intrusion detection and prevention based on their cryptographic properties.

Goals Discussion on SecureSIS. Sensitive information protection is the first priority of SecureSIS. The proposed four components of SecureSIS satisfy the goals of SecureSIS, whereby only legitimate users with proper permissions are able to access sensitive information; transmitted sensitive information is identically maintained among involved entities; and only privileged users are able to understand and access sensitive information.

UIG is satisfied by AAM. The use of dynamic communication keys and multifactor authentication can guarantee that only genuine users and systems are able to understand requests and responses and generate identical tokens offline. Authenticity and authority are thus guaranteed, and Equations. 3.20 and 3.21 are met.

CCG is satisfied by DKM and UGKM. By using dynamic communication keys and virtual cluster keys, sensitive information is distributed securely among group users. Using the cryptographic properties of dynamic keys and the AccessAuth protocol, all messages are embedded in a unique token to guarantee information integrity, and hence, Equations. 3.22 and 3.23 are met.

SISG is satisfied by SIM. The combination of UGKM and AAM used in SIM ensures that not only can privileged users understand and retrieve information, but also in emergency circumstances, dynamic information ownership is enabled. In addition, the use of dynamic data keys guarantees the secure storage of sensitive information. Equation 3.24 is therefore met.

Informally, by applying the SecureSIS pentad, the design of the four components satisfies the goals of SecureSIS (shown in Table 4.5).

Table 4.5. SecureSIS Components vs. Goals.

Goals of SecureSIS (Section 3.2.10)	Components (Definition 3.2)	SeucresIS Pentad (Definition 3.9)	
UIG	AAM,DKM,UGKM	AA,NR,CO	UT
CCG	DKM, UGKM	IN,NR,CO	
SISG	SIM, DKM,UGKM	IN,NR,CO	

In the following chapter, the formal security of each component is discussed, and then the SecureSIS pentad model is built to evaluate the security of SecureSIS in order to prove that the proposed security architecture satisfies the goals of SecureSIS.

Chapter 5

Security Analysis and Discussion on SecureSIS

Goals. In the previous chapter, four “tangible” components of SecureSIS were proposed and described formally in order to demonstrate their role in protecting *communication channel*, *user interface* and *sensitive information storage* of sensitive information systems. The components have also been compared, informally, with existing mechanisms in term of security.

In this chapter, a formal and thorough security analysis and discussion of the four components are given. Information theory and probability theory are used to demonstrate the security of DKM (Section 5.1), UGKM (Section 5.2) in protecting *communication channel*, and the security of SIM (Section 5.4) in protecting *sensitive information storage*; Spi calculus is adopted to evaluate the security of AAM (Section 5.3) in protecting *user interface*. Based on these results, we build the SecureSIS pentad model⁶⁶ in Section 5.5 to assess the security of SecureSIS in order to show that the proposed security architecture satisfies authenticity and authority, integrity, non-repudiation, confidentiality and utility security properties. With these properties, the

⁶⁶ The SecureSIS pentad is defined in Section 3.3.1.

security goals of SecureSIS (Section 3.2.10) are met (discussed in Section 5.5.6). The organization of this chapter is illustrated in Figure 5.1 as a signpost to show the security analysis and discussion logic in this chapter. The chapter concludes in Section 5.6.

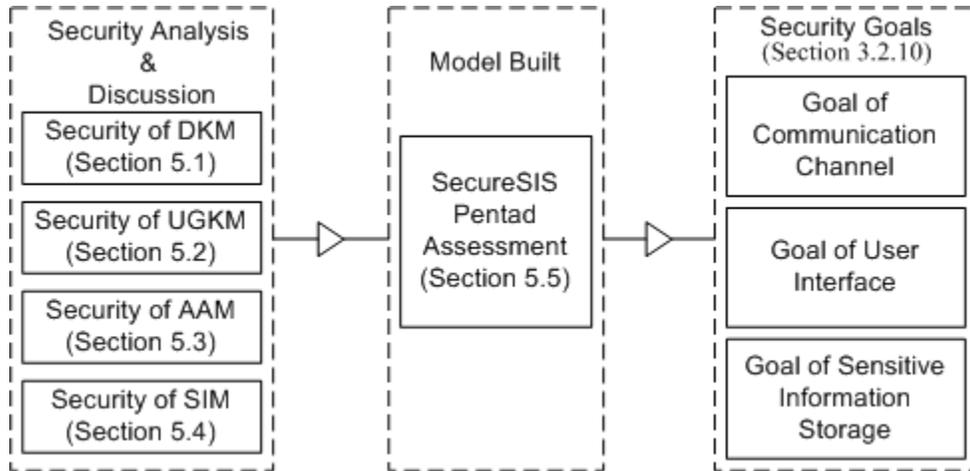


Figure 5.1. The Organization of Security Analysis and Discussion.

5.1. Security of DKM

DKM handles the security of sensitive information systems by employing dynamic key theory. It adopts two sets of dynamic keys to protect the sensitive information of users. It also employs one set of dynamic keys to secure communication between components should other components (such as AAM, SIM or UGKM) require the dynamic communication keys of users in order to process user requests. DKM has the dynamic key cryptographic properties of dynamic key secrecy (Theorem 3.1), former key secrecy (Theorem 3.2), key collision resistance (Theorem 3.3) and key consistency (Theorem 3.4) features. These properties guarantee the security of SecureSIS.

The use of the cryptographic properties of dynamic keys to provide a security foundation to support the security of DKM is discussed in Section 5.1.1. The section finishes with a summary restating the contribution of dynamic keys to SecureSIS.

5.1.1. Dynamic Keys in DKM

Definition 3.4 (DKM) in Section 3.2.3 and the Dynamic Key Agreement in Section 4.1.1 demonstrate that two sets of dynamic keys are necessary to ensure security when protecting the sensitive information of users. Theorems 3.5 and 3.6 prove that the use of dynamic keys improves the security of sensitive information systems. Dynamic keys offer more security than long-term keys and, in comparison, asymmetric cryptosystems are insecure. Therefore, dynamic keys are adopted rather than long-term shared keys and public keys. The dynamic communication key set $\{dk_{y_j} \mid j \in \square\}$ protects *communication channel* and *user interface*, while the dynamic data key set $\{dk_{x_i} \mid i \in \square\}$ 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 5.1 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 information⁶⁷, $I(A; B) = \sum Pr(A; B) \log\left(\frac{Pr(A; B)}{Pr(A)Pr(B)}\right)$, if

$A = DK_x$ and $B = DK_y$, then we have:

⁶⁷ Mutual information is a measure of the amount of information can be obtained about one by observing another [Gr90].

$$I(DK_x; DK_y) = \sum Pr(DK_x; DK_y) \log\left(\frac{Pr(DK_x; DK_y)}{Pr(DK_x)P(DK_y)}\right) \quad (5.1)$$

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 $P(A; B) = P(A)P(B)$, thus:

$$P(DK_x; DK_y) = P(DK_x)P(DK_y) \quad (5.2)$$

If that is the case, then:

$$I(DK_x; DK_y) = \sum Pr(DK_x; DK_y) \log\left(\frac{Pr(DK_x; DK_y)}{Pr(DK_x)P(DK_y)}\right) = 0 \quad (5.3)$$

which is equivalent to saying that one disclosed set of dynamic keys cannot reveal any information about another set of dynamic keys. \square

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

Corollary 5.2 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 5.1, 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. \square

5.1.2. Summary

Based on the mathematical proofs and discussion already presented in this thesis, the use of dynamic keys has the following security factors to contribute in SecureSIS:

- Dynamic keys have dynamic key secrecy, former key secrecy, key collusion resistance and key consistency properties.
- Dynamic keys are more secure than long-term shared keys and session keys, and more convenient than a one-time pad.
- According to Corollary 3.1, asymmetric keys are not sufficiently secure to protect sensitive information.
- The breach of one set of dynamic keys does not compromise the security of SecureSIS.
- The use of two sets of dynamic keys in SecureSIS achieves intrusion detection and prevention.

5.2. Security of UGKM

The security of UGKM is enhanced by the use of dynamic keys. The privacy protection of individuals in UGKM is also enhanced by categorizing group members into passive users and active users. In this section, a comprehensive discussion on the security of UGKM is presented. As described in Section 4.2, the proposed UGKM is a two-tier hybrid group key management approach. The passive user tier applies key tree

group key management. This form of security has been discussed and evaluated by [KiPeTs04, StTsWa98, WaHaAg97, WaLe05, WoGoLa00].

In the next section the cryptographic properties of group key management are discussed in order to guarantee the security of multicasting contents. Group key secrecy is discussed in Section 5.2.1, followed by forward and backward secrecy and key independence in Sections 5.2.2, 5.2.3 and 5.2.4 respectively. Collusion resistance is deliberated in Section 5.2.5. The summary restates the contributions of the proposed UGKM.

5.2.1. Group Key Secrecy

Group key secrecy, as defined in Section 4.2.2, 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 4.2.3 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 5.3 The contributed virtual cluster key is computational infeasible.

Proof: Assume a virtual cluster $vc_n \in VC$ consists of one active user ω_m (Definition 4) and $n-1$ passive users $vc_n \in VC, vc_n = \{\omega_m, involved \sum \omega_i\}$. The virtual cluster key K_{vc} is formed by contributing the intermediate key $ik_i = f(dk_{y_j}, u_i) \bmod p$ (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 $IK = \{ik_i | i \in \square\}$, the probability of breaching the contributed K_{vc} is:

$$Pr(K | IK) = Pr(K = K_{vc}; IK = ik_1) + Pr(K = K_{vc}; IK = ik_2) + \dots + Pr(K = K_{vc}; IK = ik_n) \quad (5.4)$$

Thus we have:

$$Pr(K | IK) = \sum_{i=1}^n Pr(K = K_{vc}; IK = ik_i) \quad (5.5)$$

According to probability theory, $Pr(A; B) = Pr(A | B) \cdot Pr(B)$, so:

$$Pr(K | IK) = \sum_{i=1}^n Pr(K = K_{vc} | IK = ik_i) Pr(IK = ik_i) \quad (5.6)$$

The contributed secret $dk_{y_j}.u_i$ has all the cryptographic properties of dynamic keys and the special function $f(\cdot)$ has the property of $\forall x, y(x \neq y), \neg \exists f(x) = f(y)$ (Definition 3.1). Therefore, the probability of generating each intermediate key $ik_i = f(dk_{y_j}.u_i) \bmod p$ is $\frac{1}{p}$. In other words, the generated intermediate key is

uniformly distributed over the interval $[0, p - 1]$, and we have:

$$Pr(IK = ik_i) = \frac{1}{p} \quad (5.7)$$

Combined with (5.6), we have:

$$Pr(K | IK) = \frac{1}{p} \sum_{i=1}^n Pr(K = K_{vc} | IK = ik_i) \quad (5.8)$$

Also, because $K_{vc} = f(ik_1 \dots ik_n) \bmod p$, so:

$$Pr(K | IK) = \frac{1}{p} \sum_{i=1}^n Pr(K = f(ik_1 \dots ik_n) | IK = ik_i) \quad (5.9)$$

There are n intermediate keys in vc_n , so, given an intermediate key, the probability of guessing $K = ik_1 \dots ik_n$ is $\frac{1}{n}$.

$$Pr(K = ik_1 \dots ik_n | IK = ik_i) = \frac{1}{n} \quad (5.10)$$

However, when the special one-way function $f(\cdot)$ is applied, this makes it harder for an adversary to work out the $K_{vc} = f(ik_1 \dots ik_n)$, so:

$$Pr(K = f(ik_1 \dots ik_n) | IK = ik_i) \leq \frac{1}{n} \quad (5.11)$$

Thus, combining (5.9) and (5.11), we have:

$$Pr(K | IK) \leq \frac{1}{p} \sum_{i=1}^n \frac{1}{n} = \frac{1}{p} \quad (5.12)$$

Because the large prime number p is the key space of K_{vc} , the maximum security of $Pr(K | IK)$ is $\frac{1}{p}$, thus:

$$Pr(K | IK) = \frac{1}{p} \quad (5.13)$$

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. \square

5.2.2. Forward Secrecy

Forward secrecy, as defined in UGKM cryptographic properties (Section 4.2.2), 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 (described in Section 4.2.5).

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 4.2.5, a corollary can be made.

Corollary 5.4 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:

$$Pr(new K_{vc} | K_{vc}) \quad (5.14)$$

According to Corollary 5.4, virtual cluster keys are uniformly distributed. The old K_{vc} and new K_{vc} are therefore independent and we have:

$$Pr(new K_{vc}, K_{vc}) = Pr(new K_{vc})Pr(K_{vc}) \quad (5.15)$$

Since $P(A; B) = P(A | B)P(B)$, then (5.14) can be written as:

$$Pr(new K_{vc} | K_{vc}) = \frac{Pr(new K_{vc}, K_{vc})}{Pr(K_{vc})} \quad (5.16)$$

Taking (5.15) into (5.16):

$$Pr(new K_{vc} | K_{vc}) = Pr(new K_{vc}) \quad (5.17)$$

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. \square

5.2.3. Backward Secrecy

Backward secrecy, as defined in UGKM cryptographic properties (Section 4.2.2), 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 (described in Section 4.2.4).

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 4.2.4, a corollary can be made.

Corollary 5.5 Backward secrecy is guaranteed in virtual clusters.

Proof: Suppose ω_n is a new member about to join a virtual cluster. When a passive user joins an existing virtual cluster operation, a new K_{vc} is contributed by ω_n and the old K_{vc} will be updated for all existing members. The probability of ω_n knowing the old K_{vc} is:

$$Pr(old K_{vc} | new K_{vc}) \quad (5.18)$$

According to Corollary 5.3, virtual cluster keys are uniformly distributed. Because the old K_{vc} and the new K_{vc} are independent, we have:

$$Pr(old K_{vc}, new K_{vc}) = Pr(old K_{vc})Pr(new K_{vc}) \quad (5.19)$$

Since $P(A; B) = P(A | B)P(B)$, (5.18) can be written as:

$$Pr(old K_{vc} | new K_{vc}) = \frac{Pr(old K_{vc}, new K_{vc})}{Pr(new K_{vc})} \quad (5.20)$$

and, taking (5.19) into (5.20):

$$Pr(old K_{vc} | new K_{vc}) = Pr(old K_{vc}) \quad (5.21)$$

The probability of using the new K_{vc} to find old K_{vc} is therefore the same as finding the old K_{vc} . In other words, ω_n cannot use the new K_{vc} to gain access to previous group content, and an adversary is in the same situation. Backward secrecy is guaranteed in virtual clusters; the proof is complete. \square

5.2.4. Collusion Resistance

Collusion attack refers to a situation where any set of departing members work together to regain the current group key by applying the old keying materials known by them. Collusion resistance in UGKM ensures that previous virtual cluster passive users cannot collude and determine the current virtual cluster keys. The privacy of current active users of the virtual cluster is protected because the previous virtual cluster users cannot collude to identify the current key. Therefore, a collusion resistance corollary for UGKM can be made.

Corollary 5.6 UGKM achieves collusion resistance.

Proof: Suppose a virtual cluster $vc_n \in VC$ and k previous passive users⁶⁸ want to collude to identify the new virtual cluster key of vc_n . Let every $\omega_i, 1 \leq i \leq k$ hold key materials (such as intermediate keys described in Section 4.2.3), and denote this as $k_{materials} \cdot \omega_i$. The uncertainty of the new virtual cluster key for k previous passive users is:

$$H(new K_{vc} | \sum_{i=1}^k k_{materials} \cdot \omega_i) \quad (5.22)$$

According to the chain rule of information entropy, (5.22) can be transformed into the equation:

$$H(\sum_{i=1}^k k_{materials} \cdot \omega_i, new K_{vc}) = H(\sum_{i=1}^k k_{materials} \cdot \omega_i) + H(new K_{vc} | \sum_{i=1}^k k_{materials} \cdot \omega_i) \quad (5.23)$$

Because previous passive users know their key materials, the uncertainty of their key materials is:

$$H(\sum_{i=1}^k k_{materials} \cdot \omega_i) = 0 \quad (5.24)$$

and the uncertainty of the new cluster key and their key materials is:

$$H(\sum_{i=1}^k k_{materials} \cdot \omega_i, new K_{vc}) = H(new K_{vc}) \quad (5.25)$$

Then, taking (5.24) and (5.25) into (5.23) we have:

$$H(new K_{vc} | \sum_{i=1}^k k_{materials} \cdot \omega_i) = H(new K_{vc}) \quad (5.26)$$

⁶⁸ According to the active user leave operation, when an active user leaves a virtual cluster, the virtual cluster is destroyed. Collusion attack is therefore not a security concern for situations involving active users.

The uncertainty of knowing the new virtual cluster key through former key materials of k passive users is same as the uncertainty of the new virtual cluster key. According to Corollary 5.4, forward secrecy is guaranteed in virtual clusters and the new virtual cluster key is secure. An adversary can no more gain access to the current virtual cluster key than can the k previous passive users; the proof is complete. \square

5.2.5. Summary

This section formally proved the security of UGKM following on from the discussion in Section 4.2.2. The proposed UGKM has the following security factors to contribute to SecureSIS:

- Group key secrecy is satisfied by proving that the contributed virtual cluster key is uniformly distributed over the key space. It is computationally infeasible.
- Forward secrecy is satisfied by proving that a previous passive user, knowing a contiguous subset of old virtual cluster keys, cannot gain information concerning any subsequent virtual cluster keys.
- Backward secrecy is satisfied by verifying that a user, knowing a contiguous subset of virtual cluster keys, cannot gain information concerning the preceding virtual cluster keys.
- Collusion resistance is achieved by certifying that a number of previous passive users cannot collude to find subsequent virtual cluster keys.

This section has used mathematical proofs to show the suitability of UGKM for inclusion in the SecureSIS architecture. The next section presents an evaluation of AAM for the same purpose.

5.3. 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. The latter two protocols involve sensitive information transmission. Therefore, in this section, the Logon and AccessAuth protocols are examined to show the security in *user interface* protection. The section finishes by restating the contributions of the proposed AAM to SecureSIS.

In order to verify the security of each protocol, Spi calculus [Ab99, AbGo97] 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 . To start verifying the security of AAM by Spi calculus, the features of the Spi calculus are essential.

5.3.1. Introduction to the Spi Calculus

In this section, we briefly introduce the Spi calculus [Ab99] syntax and semantics. In the Spi calculus, the simplicity of the calculus lies in the dual role that names play as communication channels and variables. Letting x and y range over variables, we assume that C is a set composed of public communication channels $\{c_{xy} \in C\}$ and V

is a set of private communication channels $\{v_{xy} \in V\}$ established between entities x and y . Spi calculus has following process constructs:

- **Concurrency** – written $P \mid Q$, behaves as processes P and Q running in parallel.
- **Communication** – the basic computation and synchronisation mechanism in the Spi calculus is interaction, in which a term N is communicated from an output process to an input process via a named channel, c_{xy} or v_{xy} .
 - An output process, $\overline{c_{xy}} \langle N \rangle . P$, indicates that term N is communicated on channel c_{xy} and then process P runs.
 - An input process, $c_{xy}(x).P$, describes a process waiting for a term N that was sent on a communication channel named c_{xy} before proceeding as P .
- **Replication** – written $!P$, behaves as an infinite number of copies of P running in parallel.
- **Match** – written $[M \text{ is } N]P$, behaves as P provided that terms M and N are the same, otherwise the process stalls.
- **Encryption** – written $\{M\}K$, represents the cipher text obtained by encrypting the term M under the shared key K using a symmetric algorithm.
- **Decryption** – written $\text{case } L \text{ of } \{x\}K \text{ in } P$, attempts to decrypt the term L with the shared key K . If L is a cipher text of the form $\{M\}K$, then the process behaves as $P[M/x]$ ⁶⁹, otherwise the process stalls.

⁶⁹This denotes the outcome of replacing each free occurrence of x in process P with the term M .

- **Restriction** – written $(\nu n)P$, makes a new private name n , which may occur in P , and then behaves as P .

The basic security property of Spi calculus is secrecy, which is based on the indistinguishability of processes. By using this property to evaluate the security of cryptographic protocols, a few additional notions need to be presented:

- **Reduction relation** – written \mapsto , is defined as the least relation closed under a set of reduction rules. The main reduction rule that captures the ability of processes to communicate through channels is:

$$\overline{c_{xy}} \langle N \rangle . P \mid c_{xy}(x) . Q \mapsto P \mid Q[N / x] \quad (5.27)$$

- **Reaction relation** – written $P \mapsto P'$, indicates that there is a reaction amongst the sub-processes of P , if P can perform a computation step, following which it is now P' .

5.3.2. Logon Protocol

In order to investigate the Logon protocol, the protocol needs to be first abstracted into Spi calculus. Figure 5.2 depicts the structure of the protocol, and informally, the protocol is written as follows:

- i) $u_i \rightarrow aam : \{logon_req, h(i, dk_{y(j-1)}, u_i)\} dk_{y_j}, u_i$ 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) $dkm \rightarrow aam : \{dk_{y_j}, u_i\} cdk_{l+1}, aam$ on $v_{da}, v_{da} \in V$.
- iv) $aam \rightarrow u_i : \{logon_req, h(logon_req, dk_{y_j}, u_i)\} dk_{y(j+1)}, u_i$ on $c_{au}, c_{au} \in C$.

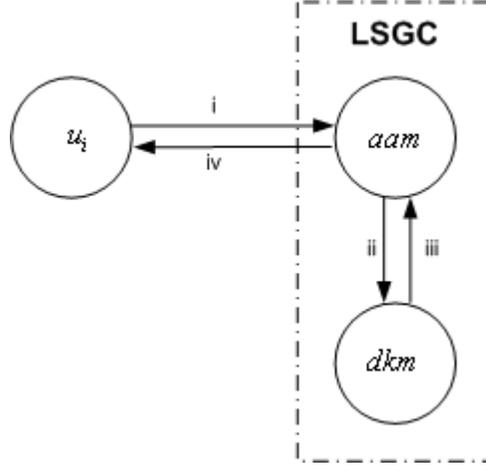


Figure 5.2. Structure of the Logon Protocol.

It is assumed there are n users and each user has a public input channel. 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 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 (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).

$$\begin{aligned}
 \text{Send}_{w,t} \square \overline{c_{wt}} \left\langle \{ \text{logon_req}, h(w, dk_{Y(j-1)}.u_w) \} dk_{Y_j}.u_w \right\rangle / c_{tw}(x_{cipher}). \text{case } x_{cipher} \text{ of} \\
 \{ x, H(y_p) \} dk_{Y(j+1)}.u_w \text{ in let } (x, y_{nonce}) = y_p \text{ in} \\
 [x \text{ is logon_req}] [y_{nonce} \text{ is } dk_{Y_j}.u_w] \text{ in } F
 \end{aligned} \tag{5.28}$$

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 \square 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).

$$\begin{aligned}
Recv_t \square c_{wt}(y_{cipher}).case\ y_{cipher}\ of\ \{x, H(y_p^1)\}dk_{Y_j}.u_w\ in\ let(x, y_{nonce}^1) = y_p^1 \\
in\ [x\ is\ w][y_{nonce}^1\ is\ dk_{Y_{(j-1)}}.u_w]\ | \\
\overline{c_{tw}}\langle\{logon_req, h(logon_req, dk_{Y_j}.u_w)\}dk_{Y_{(j+1)}}.u_w\rangle
\end{aligned} \tag{5.29}$$

The processes $Sys(I_1 \dots I_m)$ 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 $dk_{Y_j}.u_w$ and receive LSGC challenge information under the dynamic communication key $dk_{Y_{(j+1)}}.u_w$. Besides, $(vdk_{Y_j}.u_w)$ 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 $P_1 | \dots | P_m$, and $\underline{(vdk_{Y_j}.u_{wx})} \underline{(vdk_{Y_{(j+1)}}.u_{wx})}$ stand for $(vdk_{Y_j}.u_{w1}) \dots (vdk_{Y_j}.u_{wm}) (vdk_{Y_{(j+1)}}.u_{w1}) \dots (vdk_{Y_{(j+1)}}.u_{wm})$ we have:

$$Sys(I_1 \dots I_m) \square (c_{wt})(c_{tw}) \underline{(vdk_{Y_j}.u_{wx})} \underline{(vdk_{Y_{(j+1)}}.u_{wx})} \{ \bigcup_{x \in 1..m} (Send_{wx,tx} \mid !Recv_{tx}) \} \tag{5.30}$$

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 Equations 5.28 and 5.29, we have Equation 5.30 rewritten as:

$$\begin{aligned}
Sys \equiv & (vdk_{y_j}.u_w)(vdk_{y_{(j+1)}}.u_w)c_{wt}(y_{cipher}).case\ y_{cipher}\ of \\
& \{x, H(y_p^1)\}dk_{y_j}.u_w\ in\ let(x, y_{nonce}^1) = y_p^1\ in\ (x\ is\ w)(y_{nonce}^1\ is\ dk_{y_{(j-1)}}.u_w)\ | \\
& \overline{c_{wt}} \langle \{logon_req, h(w, dk_{y_{(j-1)}}.u_w)\}dk_{y_j}.u_w \rangle\ | \\
& c_{tw}(x_{cipher}).case\ x_{cipher}\ of\ \{x, H(y_p)\}dk_{y_{(j+1)}}.u_w\ in\ let(x, y_{nonce}) = y_p \\
& in\ (x\ is\ logon_req)(y_{nonce}\ is\ dk_{y_j}.u_w)\ in\ F\ | \\
& \overline{c_{tw}} \langle \{logon_req, h(logon_req, dk_{y_j}.u_w)\}dk_{y_{(j+1)}}.u_w \rangle
\end{aligned} \tag{5.31}$$

Based on the reaction relation and reduction relation rules (Equation 5.27),

$$\begin{aligned}
Sys & \mapsto (vdk_{y_j}.u_w)(vdk_{y_{(j+1)}}.u_w)F(logon_req, h(logon_req, dk_{y_j}.u_w), \\
& h(w, dk_{y_{(j-1)}}.u_w)) \\
& \mapsto F(logon_req, h(logon_req, dk_{y_j}.u_w), h(w, dk_{y_{(j-1)}}.u_w))
\end{aligned} \tag{5.32}$$

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 (discussed in Section 3.1 and Theorem 3.5), 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.

$$\begin{aligned}
Send_{spec(w,t)} & \sqcap \overline{c_{wt}} \langle \{logon_req, random\}dk_{y_j}.u_w \rangle\ | \\
& c_{tw}(x_{cipher}).case\ x_{cipher}\ of\ \{x, random\}dk_{y_{(j+1)}}.u_w \\
& in\ [x\ is\ logon_req]\ in\ F
\end{aligned} \tag{5.33}$$

$$\begin{aligned}
Recv_{spec(t)} & \sqcap c_{wt}(y_{cipher}).case\ y_{cipher}\ of\ \{x, random\}dk_{y_j}.u_w\ in\ [x\ is\ w]\ | \\
& \overline{c_{tw}} \langle \{logon_req, random\}dk_{y_{(j+1)}}.u_w \rangle
\end{aligned} \tag{5.34}$$

$$\begin{aligned}
& Sys(I_1 \dots I_m)_{spec} \sqsubseteq (c_{wt})(c_{tw})(\underline{vdk_{Y_j} \cdot u_{wx}})(\underline{vdk_{Y_{(j+1)}} \cdot u_{wx}}) \\
& \quad \left\{ \bigcup_{x \in 1 \dots m} (Send_{spec(wx, tx)} \mid !Recv_{spec(tx)}) \right\}
\end{aligned} \tag{5.35}$$

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 $Sys(I_1 \dots I_m) \sqsubseteq Sys(I_1 \dots I_m)_{spec}$). In other words, $Sys(I_1 \dots I_m)$ and $Sys(I_1 \dots I_m)_{spec}$ 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, $Sys(I_1 \dots I_m) \sqsubseteq Sys(I_1 \dots I_m)_{spec}$ 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.

5.3.3. AccessAuth Protocol

The AccessAuth protocol is designed to perform identity verification and access control management that allows individuals and group users to share sensitive information. Similar to the Logon protocol, the AccessAuth protocol needs to be informally transformed into the following (the structure of the protocol is depicted in Figure 5.3):

- i) $p_m \rightarrow LSGC : \{I_n - req, h(I_n - req, dk_{Y_{(j-1)}} \cdot p_m)\} dk_{Y_j} \cdot p_m$ on c_{mL}
- ii) $LSGC \rightarrow p_n : \{I_n - req, h(I_n - req, dk_{Y_{(i-1)}} \cdot p_n)\} dk_{Y_j} \cdot p_n$ on c_{Ln}
- iii) $p_n \rightarrow LSGC : \{I'_n - res, h(I'_n - res, dk_{Y_j} \cdot p_n)\} dk_{Y_{(j+1)}} \cdot p_n$ on c_{nL}

iv) $LSGC \rightarrow p_m : \{I_n, h(I_n, dk_{y_j} \cdot p_m)\} dk_{y_{(j+1)}} \cdot p_m$ on c_{Lm}

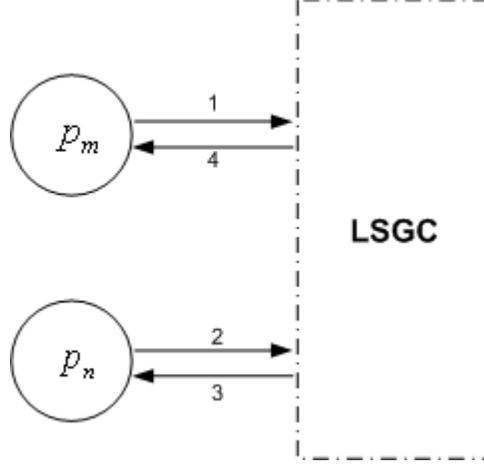


Figure 5.3. Structure of the AccessAuth Protocol.

Meanwhile, c_{mL} , c_{Ln} , c_{nL} and c_{Lm} are public communication channels among participants and the LSGC. In the Spi calculus description of the AccessAuth protocol, given an instance (w, t, D) , the following process corresponds to the role of participants and the LSGC (AAM and DKM).

$$\begin{aligned}
 Send_{w,t} \square \overline{c_{wL}} \langle \{I_{t-req}, h(I_{t-req}, dk_{y_{(j-1)}} \cdot p_w)\} dk_{y_j} \cdot p_w \rangle / \\
 c_{Lw}(x_{cipher}).case\ x_{cipher}\ of\ \{x_t, H(y_p)\} dk_{y_{(j+1)}} \cdot p_w\ in \\
 let\ (x_t, y_{nonce}) = y_p\ in\ [x_t\ is\ I_t][y_{nonce}\ is\ dk_{y_j} \cdot p_w]\ in\ F
 \end{aligned} \tag{5.36}$$

$$\begin{aligned}
 Recv_t \square c_{Lt}(y_{cipher}).case\ y_{cipher}\ of\ \{x_t^1, H(y_p^2)\} dk_{y_j} \cdot p_t\ in\ let\ (x_t^1, y_{nonce}^2) = y_p^2\ in \\
 [x_t^1\ is\ request][y_{nonce}^2\ is\ dk_{y_{(a-1)}} \cdot p_t]\ | \\
 \overline{c_{tL}} \langle \{I'_{t-res}, h(I'_{t-res}, dk_{y_j} \cdot p_t)\} dk_{y_{(j+1)}} \cdot p_t \rangle
 \end{aligned} \tag{5.37}$$

The sending and receiving processes are described in detail in AAM. The LSGC controls the forwarding and assembling of messages among participants. The LSGC is the same for all instances.

$$\begin{aligned}
& LSGC \sqcap c_{wL}(z_{cipher}).case\ z_{cipher}\ of\ \{x_t^1, H(y_p^1)\}dk_{y_j \cdot p_w}\ in\ let\ (x_t^1, y_{nonce}^1) = y_p^1\ in \\
& [x_t^1\ is\ request][y_{nonce}^1\ is\ dk_{Y(a-1) \cdot p_w}].\overline{c_{Lt}} \\
& \langle \{I_t - req, h(I_t - req, dk_{Y(i-1) \cdot p_t})\}dk_{y_j \cdot p_t} \rangle | \tag{5.38}
\end{aligned}$$

$$\begin{aligned}
& c_{tL}(w_{cipher}).case\ w_{cipher}\ of\ \{x_t^3, H(y_p^3)\}dk_{Y(j+1) \cdot p_t}\ in\ let\ (x_t^3, y_{nonce}^3) = y_p^3\ in \\
& [x_t^3\ is\ response][y_{nonce}^3\ is\ dk_{Y_a \cdot p_t}].\overline{c_{Lw}} \langle \{I_t, h(I_t, dk_{y_j \cdot p_w})\}dk_{Y(j+1) \cdot p_w} \rangle
\end{aligned}$$

$$\begin{aligned}
& Sys(I_1 \dots I_m) \sqcap (\overline{vdk_{y_j \cdot p_x}})(\overline{vdk_{Y(j+1) \cdot p_x}}) \\
& \{ \bigcup_{x \in 1 \dots m} Send_{w_x, t_x}(I_x) \mid !LSGC \mid !Recv_{t_x} \} \tag{5.39}
\end{aligned}$$

The replication of the server $!LSGC$ and the receiving processes $!Recv_{t_i}$ means that every participant is ready to play the role of receiver in any number of runs of the protocol in parallel. By combining Equations 5.36, 5.37 and 5.38, we have Equation 5.39 rewritten as:

$$\begin{aligned}
& Sys \equiv (\overline{vdk_{y_j \cdot p_w}})case\ \{I_t - req, h(I_t - req, dk_{Y(j-1) \cdot p_w})\}dk_{y_j \cdot p_w} \\
& of\ \{x_t^1, H(y_p^1)\}dk_{y_j \cdot p_w}\ in\ let\ (x_t^1, y_{nonce}^1) = y_p^1\ in \\
& (\overline{vdk_{y_j \cdot p_t}})case\ \{I_t - req, h(I_t - req, dk_{Y(i-1) \cdot p_t})\}dk_{y_j \cdot p_t} \\
& of\ \{x_t^1, H(y_p^2)\}dk_{y_j \cdot p_t}\ in\ let\ (x_t^1, y_{nonce}^2) = y_p^2\ in \\
& (\overline{vdk_{Y(j+1) \cdot p_t}})case\ \{I'_t - res, h(I'_t - res, dk_{y_j \cdot p_t})\}dk_{Y(j+1) \cdot p_t} \\
& of\ \{x_t^3, H(y_p^3)\}dk_{Y(j+1) \cdot p_t}\ in\ let\ (x_t^3, y_{nonce}^3) = y_p^3\ in \\
& (\overline{vdk_{Y(j+1) \cdot p_w}})case\ \{I_t, h(I_t, dk_{y_j \cdot p_w})\}dk_{Y(j+1) \cdot p_w} \\
& of\ \{x_t, H(y_p)\}dk_{Y(j+1) \cdot p_w}\ in\ let\ (x_t, y_{nonce}) = y_p\ in\ F \tag{5.40}
\end{aligned}$$

Based on the reaction relation and reduction relation rules (Equation 5.27), we have:

$$\begin{aligned}
& Sys \mapsto (\overline{vdk_{y_j \cdot p_w}})(\overline{vdk_{Y(j+1) \cdot p_w}})(\overline{vdk_{y_j \cdot p_t}}) \\
& (\overline{vdk_{Y(j+1) \cdot p_t}})F(I_t - req, I_t - req, I'_t - res, I_t) \\
& \mapsto F(I_t - req, I_t - req, I'_t - res, I_t) \tag{5.41}
\end{aligned}$$

Thus the processes have no disclosure of the critical information I_t and its intermediate values $I_t - req$ and $I'_t - res$. In the protocol, dynamic communication

keys are employed in forming tokens. In accordance with the cryptographic properties of dynamic keys, a specification is devised as follows:

$$\begin{aligned} Send_{spec(w,t)} \sqsubseteq \overline{c_{wL}} \left\langle \{I_t\text{-req}, random\} dk_{y_j \cdot p_w} \right\rangle / c_{Lw}(x_{cipher}) . case\ x_{cipher} \\ of\ \{x_t, random\} dk_{y_{(j+1)} \cdot p_w} \text{ in } [x_t \text{ is } I_t] \text{ in } F \end{aligned} \quad (5.42)$$

$$\begin{aligned} Recv_{spec(t)} \sqsubseteq c_{Lt}(y_{cipher}) . case\ y_{cipher} \text{ of } \{x_t^1, random\} dk_{y_j \cdot p_t} \\ in\ [x_t^1 \text{ is request}] / \overline{c_{tL}} \left\langle \{I_t\text{-res}, random\} dk_{y_{(j+1)} \cdot p_t} \right\rangle \end{aligned} \quad (5.43)$$

$$\begin{aligned} LSGC_{spec} \sqsubseteq c_{wL}(z_{cipher}) . case\ z_{cipher} \text{ of } \{x_t^1, random\} dk_{y_j \cdot p_w} \\ in\ [x_t^1 \text{ is request}] . \overline{c_{Lt}} \left\langle \{I_t\text{-req}, random\} dk_{y_j \cdot p_t} \right\rangle / \\ c_{tL}(w_{cipher}) . case\ w_{cipher} \text{ of } \{x_t^3, random\} dk_{y_{(j+1)} \cdot p_t} \\ in\ [x_t^3 \text{ is response}] . \overline{c_{Lw}} \left\langle \{I_t, random\} dk_{y_{(j+1)} \cdot p_w} \right\rangle \end{aligned} \quad (5.44)$$

$$\begin{aligned} Sys(I_1 \dots I_m)_{spec} \sqsubseteq (vdk_{y_j \cdot p_x})(vdk_{y_{(j+1)} \cdot p_x}) \\ \left\{ \bigcup_{x \in 1..m} (Send_{spec(wx,tx)}(I_x)) / LSGC_{spec} / Recv_{spec(x)} \right\} \end{aligned} \quad (5.45)$$

After applying the reduction relation rules, we have $Sys(I_1 \dots I_m) \sqsubseteq Sys(I_1 \dots I_m)_{spec}$. In other words, $Sys(I_1 \dots I_m)$ and $Sys(I_1 \dots I_m)_{spec}$ are indistinguishable to an adversary. Thus, similar to the Logon protocol, the AccessAuth protocol also has two important properties:

- Authenticity: $Sys(I_1 \dots I_m) \sqsubseteq Sys(I_1 \dots I_m)_{spec}$ for any message.
- Secrecy: $Sys(I_1 \dots I_m) \sqsubseteq Sys(I_1 \dots I_m)_{spec}$ if $F(message) \sqsubseteq F(random)$ for any message.

5.3.4. Summary

This section formally discussed the security of AAM by using Spi calculus. It proved that by using dynamic keys, Logon and AccessAuth protocols the proposed AAM does

not leak any sensitive information, and sensitive information and random numbers are indistinguishable to an adversary. Also, both the Logon and AccessAuth protocols have authenticity and secrecy properties. The proposed AAM thus has the following security factors to contribute to SecureSIS:

- Logon and AccessAuth protocols are secure; they do not reveal sensitive information in transit between entities.
- AAM has authenticity and secrecy properties.

5.4. 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*⁷⁰ and *user interface*⁷¹. 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 Tipton and Krause [TiKr07], data interchange and storage present a major problem for the management of security information. Therefore, in this section, the security of interchanging sensitive information is examined in Section 5.4.1 and is followed by a discussion on the security of *sensitive information storage* in SIM. The contributions are restated in the summary.

⁷⁰ Communication channel is discussed in UGKM and AAM.

⁷¹ User interface is discussed in AAM.

5.4.1. Security of Interchanging Sensitive Information

As described in Section 4.4 on SIM, sensitive information is stored in a form of ciphers, and the ciphers (sensitive information) can be kept in multiple SIM objects. The information interchange occurs when data operations (Section 4.4.2) are triggered. Referring to Figure 4.9, suppose u_n joins an LSGC (SIM object, denoted $LSGC^1$) and wants to manage its sensitive information $ci_k.u_n$, which is located in another LSGC (denoted $LSGC^2$). Informally, the message flow can be redescribed as follows by combining Section 4.3.4 (the AAM AccessAuth protocol) and Section 4.4.1 (the SIM structure):

- i) $u_n \rightarrow LSGC^1 : \{I_n - req, h(I_n - req, dk_{Y(j-1)}.u_n)\} dk_{Yj}.u_n$
- ii) $LSGC^2 \rightarrow u_n : \{ci_k.u_n, h(ci_k.u_n, dk_{Yj}.u_n)\} dk_{Y(j+1)}.u_n$

The $LSGC^1$ in the message flow refers to the group u_n joined, while the $LSGC^2$ represents the location of the sensitive information $ci_k.u_n$. According to Definition 3.7 (Equations 3.11, 3.12 and 3.13), $ci_k.u_n$ can be expanded into $\{I_k\} dk_{xi}.u_n, \{\{dk_{xi}.u_n\} dk_{xc}.u_n, h(\{I_k\} dk_{xi}.u_n)\}$. Intuitively, sensitive information I_k is protected in transit between the user and the $LSGC^1$ and the $LSGC^2$ by the dynamic data key and the dynamic communication key. Hence, we can make a corollary:

Corollary 5.7 (Weak Security) Sensitive information interchange is secure in SIM.

Poof: According to the Spi calculus proof in AAM, the AccessAuth protocol does not reveal sensitive information in transit among entities. The $ci_k.u_n$ is thus secure,

and an adversary cannot distinguish between information ci_{k,u_n} and a random number; the proof is complete. \square

According to the weak security corollary and the cryptographic properties of dynamic keys, it is presumed that by using dynamic data keys in SIM, a strong security in information interchange can be achieved, we can thus make a corollary.

Corollary 5.8 (Strong Security) Even though communication channel is breached, Sensitive information interchange is still secure in SIM.

Proof: Suppose an adversary A breaches the communication channel and A understands the communication key and has the content of messages (say ci_{k,u_n}). According to Corollary 5.1, the compromised dynamic communication key does not affect any dynamic data key. The content ci_{k,u_n} is under the protection of the dynamic data key. In addition, dynamic key secrecy guarantees that it is computationally infeasible to find dynamic keys. Therefore, the content ci_{k,u_n} is secure; the proof is complete. \square

5.4.2. Security of Sensitive Information Storage

The security aspect of most concern in SIM is sensitive information security. Definition 3.7, SIM data operation and dynamic membership operation, 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 5.9 The breach of *sensitive information storage* does not threaten the security of sensitive information.

Proof: Suppose an adversary A breaches the security of *sensitive information storage*. In other words, A has access to all of sensitive information I in the form of cipher CI but lacks the keys to decipher CI . According to Definition 3.7, we note:

$$A \leftarrow CI \quad (5.46)$$

Thus, the probability of revealing sensitive information through the given CI for A is:

$$Pr(I | CI) \quad (5.47)$$

Let M , C and K denote plain text, cipher text and an encryption key set respectively. Symbols $+$ and $-$ are symmetric encipher and decipher operations. We have:

$$\begin{aligned} C &= M + K \\ M &= C - K \end{aligned} \quad (5.48)$$

According to conditional probability rules, (5.47) is rewritten:

$$Pr(I | CI) = \frac{Pr(I, CI)}{Pr(CI)} \quad (5.49)$$

Since (5.46), then the probability of revealing CI is one for A , $Pr(CI) = 1$, then:

$$Pr(I | CI) = Pr(I, CI) \quad (5.50)$$

Drawing on Definition 3.7 and applying it to (5.48):

$$Pr(I | CI) = Pr(I, CI) = Pr(CI - DK_x, CI) \quad (5.51)$$

Thus, the probability of revealing sensitive information through a given CI to A is the probability of knowing all dynamic data key set DK_x :

$$Pr(I | CI) = Pr(DK_x) \quad (5.52)$$

According to Theorems 3.1 , 3.2 and 3.3, the probability of A knowing the set of dynamic keys is zero. In other words, the dynamic keys are infeasible to compute.

Thus:

$$Pr(I | CI) = Pr(DK_x) = 0 \quad (5.53)$$

Hence, although *sensitive information storage* is breached, sensitive information is still secure and protected; the proof is complete. \square

Since the security of *sensitive information storage* is guaranteed by above proof, a strong claim is made.

Corollary 5.10 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, \dots, B_n partition S , and we have $B_1 \cup B_2 \cup \dots \cup B_n = S$. Due to SIM security features, the occurrence of events B_i and B_j are independent. Therefore, $B_i B_j = \emptyset$ for any pair i and j , where \emptyset denotes a null set.

Let B_j denote the event that disclosed information comes from user $u_j \in U$ and $Pr(B_i) > 0$, where $i = 1, 2, \dots, n$. 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:

$$Pr(B_j | A) = 1 \quad (5.54)$$

Apply Bayes' law, we have:

$$Pr(B_j | A) = \frac{Pr(B_j)Pr(A | B_j)}{\sum_{i=1}^n Pr(B_i)Pr(A | B_i)} \quad (5.55)$$

and, take (5.54) into (5.55), thus:

$$Pr(B_j)Pr(A | B_j) = \sum_{i=1}^n Pr(B_i)Pr(A | B_i) \quad (5.56)$$

and, expand (5.56):

$$\begin{aligned} Pr(B_j)Pr(A | B_j) &= Pr(B_1)Pr(A | B_1) + Pr(B_2)Pr(A | B_2) \\ &\dots + \\ &Pr(B_j)Pr(A | B_j) + Pr(B_{j+1})Pr(A | B_{j+1}) \\ &\dots + \\ &Pr(B_{n-1})Pr(A | B_{n-1}) + Pr(B_n)Pr(A | B_n) \end{aligned} \quad (5.57)$$

and then:

$$\begin{aligned} Pr(B_1)Pr(A | B_1) + \dots + Pr(B_{j-1})Pr(A | B_{j-1}) + Pr(B_{j+1})Pr(A | B_{j+1}) + \dots + \\ Pr(B_n)Pr(A | B_n) = 0 \end{aligned} \quad (5.58)$$

Since, $\forall Pr(B_i) > 0$, where $i = 1, 2, \dots, n$, then conditional probability of compromising sensitive information of others is zero. We have:

$$Pr(A | B_1) + \dots + Pr(A | B_{j-1}) + Pr(A | B_{j+1}) + \dots + Pr(A | B_n) = 0 \quad (5.59)$$

Therefore, even when one user is compromised in SIM, the probability of breaching other sensitive information is zero; the proof is complete. \square

5.4.3. Summary

This section has proved that the proposed SIM offers secure sensitive information interchange and *sensitive information storage* by giving Corollaries 5.7, 5.8, 5.9 and 5.10. The corollaries show that sensitive information in a cipher form is secure in transit among entities. Also, sensitive information itself is secure and protected in storage. The use of the two sets of dynamic keys in SIM contributes the following security features:

- Weak security: sensitive information interchange is secure in SIM.
- Strong security: Even should a communication channel be breached, sensitive information interchange is still secure in SIM
- The breach of *sensitive information storage* does not threaten the security of sensitive information.
- If the security of one user is breached in SIM, the security of other users is not compromised; nor is other sensitive information threatened. In other words, if *sensitive information storage* is compromised, and also one or more users are breached, the sensitive information of others in SIM is still secure and protected.

5.5. SecureSIS Pentad Assessment

In this section, we build the SecureSIS pentad model to evaluate the security of the proposed security architecture for sensitive information systems. The five elements of the SecureSIS pentad are discussed and proved to show that the proposed SecureSIS

satisfies the security requirements of the SecureSIS pentad model and the security goals of *communication channel*, *user interface* and *sensitive information storage*.

5.5.1. Authenticity & Authority Discussion

AAM offers authenticity and authority (discussed in the security of AAM (Section 5.3)). In summary, an adversary cannot distinguish $\{I_a, h(I_a, dk_{Y(i-1)})\}dk_{Y_i} \cdot p_n$ from $\{random, random\}random$ by sniffing networks. The communication between users and the LSGC is secure. In other words, an adversary can have no knowledge of conversations between entities, and only legitimate users and genuine entities are able to understand conversations.

According to the requirements of AAM (Equations 3.27, 3.28 and 3.29), a number of proofs can be given to show that SecureSIS has the property of AAM in protecting sensitive information.

Axiom 5.1 For any message $I_i \in I$, entity Q believes entity P said I_i in SecureSIS.

Proof: Suppose a user $u_i \in U$ sends a request to an LSGC. Let u_i be the entity P and LSGC be the entity Q :

$$u_i \rightarrow LSGC : \{I_req, h(I_req, dk_{Y(j-1)} \cdot u_i)\}dk_{Y_j} \cdot u_i \quad (5.60)$$

Then:

$$P := I_req, h(I_req, dk_{Y(j-1)} \cdot u_i) \quad (5.61)$$

Theorem 3.4, key consistency property of dynamic keys, states that both entities have a correlative dynamic communication key. So:

$$Q \text{ sees } I_req, h(I_req, dk_{Y(j-1)} \cdot u_i) \quad (5.62)$$

According to the proof of security of AAM, an adversary cannot distinguish between meaningful messages and random messages. The cryptographic properties of dynamic keys (Theorems 3.1, 3.2 and 3.3) also contribute to message security. Consequently, the message is secure:

$$Q \text{ believes } I_req, h(I_req, dk_{Y(j-1)}.u_i) \quad (5.63)$$

Then:

$$Q \text{ computes new token } h(I_req, dk_{Y(j-1)}.u_i) \quad (5.64)$$

If the new token is the same as $Q \text{ believes } h(I_req, dk_{Y(j-1)}.u_i)$, combining (5.61) and (5.62), we have:

$$\forall I_i \in I, P := I_i : \frac{P \rightarrow Q : I_i, token}{Q \text{ believes } P \text{ said } I_i} \quad (5.65)$$

Therefore, Equation 3.27 is satisfied in SecureSIS, and the proof is complete. \square

Axiom 5.2 Entity Q believes the claim of entity P during time interval $[t_0, t_1]$ in SecureSIS.

Proof: Suppose a user $u_i \in U$ wants to login to an LSGC. Let u_i be the entity P and the LSGC be the entity Q :

$$u_i \rightarrow LSGC : \{logon_req, h(i, dk_{Y(j-1)}.u_i)\} dk_{Y_j}.u_i \quad (5.66)$$

According to Axiom 5.1:

$$Q \text{ believes } logon_req, h(i, dk_{Y(j-1)}.u_i) \quad (5.67)$$

The entity Q uses $h(i, dk_{Y(j-1)}, u_i)$ as a key to decipher eid_i ⁷². If, and only if, the value is the same as the unique id_i then is the entity P genuine as claimed:

$$Q \text{ believes } P \quad (5.68)$$

However, according to AAM and DKM, the trust of P is only available for one message. Letting t denote the time of processing a message, then we have:

$$\int_{t_0}^{t_1} \frac{P \text{ claims to } Q}{Q \text{ believes } P} dt \quad (5.69)$$

Therefore Equation 3.28 is satisfied in SecureSIS, and the proof is complete. \square

The security of authenticity takes advantage of the properties of dynamic keys. The same dynamic key cannot be used for authentication twice. If the same key is used twice, an intrusion detection mechanism (Corollary 5.2) will be triggered. Also, as suggested, the unique id_i can be generated from biometrics, such as a fingerprint, DNA or an iris to further enhance the security of SecureSIS.

Axiom 5.3 Entity Q possesses sensitive information I of entity P , if and only if, the predicate $AR(I, *)$ is true in SecureSIS.

Proof: Assuming the AccessAuth protocol, and letting entity Q have full permission to I_i , then:

$$Q : AR(I_i, *) \equiv true \quad (5.70)$$

According to the properties of engaging users (Equations 3.6 and 3.17):

$$Q := I_i \quad (5.71)$$

Applying the summation rule:

⁷² Refer to Definition 3.6 and details in Section 4.3.

$$Q : AR(\sum_{i \in N} I_i, *) \equiv true \Rightarrow Q := \sum_{i \in N} I_i \quad (5.72)$$

Applying not operation:

$$Q : \neg(AR(\sum_{i \in N} I_i, *) \equiv true) \Rightarrow \neg(Q := \sum_{i \in N} I_i) \quad (5.73)$$

Then:

$$AR(\sum_{i \in N} I_i, *) \equiv false \Rightarrow Q := \sum_{i \in N} I_i \quad (5.74)$$

Thus:

$$\forall I_i \in I, P := I_i : \frac{\text{iff } Q : AR(I_i, *) == false}{Q := I_i} \quad (5.75)$$

Therefore Equation 3.29 is satisfied in SecureSIS, and the proof is complete. \square

The Axiom 5.3 is also indicated in SIM component, assume user u_n has permission to access only sensitive information I_a of user u_m . Then, u_n sees message $\{I_n, h(I_n, dk_{Y_j} \cdot p_n)\} dk_{Y_{(j+1)}} \cdot p_n$. According to definition of SIM, $I_n = \{I_a\} dk_{X_i} \cdot u_m \wedge \{dk_{X_i} \cdot u_m\} dk_{Y_j} \cdot u_n$. Therefore I_a is available for u_n . However, say I_b is unauthorized sensitive information for u_n , u_n cannot understand $\{I_b\} dk_{X_{(i+x)}} \cdot u_m$, although u_n has the key $dk_{X_i} \cdot u_m$ from previous transaction⁷³.

5.5.2. Integrity Discussion

Integrity deals with the intrinsic condition of sensitive information. In SecureSIS, the use of the hash function ensures sensitive information integrity. When the data is

⁷³ It applies cryptographic properties of dynamic keys.

changed, the hash function yields a different result. In SecureSIS, every assembled message has a fresh token to guarantee the sensitive information integrity property in *communication channel*. SecureSIS also guarantees that sensitive information is secure when in storage. We conclude the following axioms.

Axiom 5.4 Entity Q believes received sensitive information from entity P is identically maintained via *communication channel* in SecureSIS.

Proof: Assume all message communications such that $P \rightarrow Q : \{I_n, h(I_n, dk_{y_j} \cdot p_m)\} dk_{y_{(j+1)}} \cdot p_m$ in the AccessAuth protocol.

According to Axiom 5.1:

$$Q \text{ believes } I_n, h(I_n, dk_{y_j} \cdot p_m) \quad (5.76)$$

For group users $Q = \{Q_1 \dots Q_n\}$ sharing sensitive information, according to the proof of security of UGKM and Axiom 5.1:

$$Q_1 \dots Q_n \text{ believe } I_n, h(I_n, dk_{y_j} \cdot p_m) \quad (5.77)$$

Letting the segment of message $I'_n = I_n$, then:

$$Q \text{ believes } Q \text{ receives } I'_n \quad (5.78)$$

According to Theorem 3.4, entities Q and P have identical sets of dynamic communication keys, so:

$$Q := h(I'_n, dk_{y_j} \cdot p_m) \quad (5.79)$$

When comparing with the received token, if, and only if, both are the same, then:

$$Q \text{ believes } I'_n \equiv I_n \quad (5.80)$$

The entity Q therefore believes that I_n has not been maliciously or accidentally altered⁷⁴. Thus, combining (5.78) with (5.80), we have:

$$\forall I_i \in I, \frac{P \rightarrow Q : I_i}{Q \text{ receives } I_i \wedge Q \text{ believes } I_i \equiv I_i} \quad (5.81)$$

Thus Equation 3.30 is satisfied in SecureSIS, and the proof is complete. \square

Axiom 5.5 Entity P believes possessed sensitive information I is genuine in *sensitive information storage* and secure in SecureSIS.

Proof: According to Definition 3.7, for entity P possessing sensitive information I , we have:

$$P := I \quad (5.82)$$

Sensitive information I is stored in a form of a cipher ci :

$$ci = \{I\}dk_{xi}.P \wedge \{dk_{xi}.P, h(\{I\}dk_{xi}.P)\}dk_{xc}.P \quad (5.83)$$

According to the cryptographic properties of dynamic keys and UGKM:

$$P \text{ believes } SIM := ci \quad (5.84)$$

and:

$$P \text{ believes } SIM \text{ sees } h(\{I\}dk_{xi}.P) \wedge \{I\}dk_{xi}.P \quad (5.85)$$

The entity P is able to compute a new hash value of $\{I\}dk_{xi}.P$ to compare with $h(\{I\}dk_{xi}.P)$, if, and only if, the values are matched. Then:

$$P \text{ believes } SIM \text{ believes } I \quad (5.86)$$

⁷⁴ Malicious altering means an adversary alters or forges sensitive information. Accidental altering indicates a network transmission error or a data storage crash.

According to the discussions on the security of SIM (Corollaries 5.7, 5.8, 5.9 and 5.10):

$$P \text{ believes } I \quad (5.87)$$

Therefore, in SecureSIS, for any sensitive information stored, we have:

$$\forall I_i \in I, \frac{P := I_i}{P \text{ believes } I_i} \quad (5.88)$$

Thus Equation 3.31 is satisfied in SecureSIS, and the proof is complete. \square

5.5.3. Non-repudiation Discussion

Non-repudiation is achieved in AAM through the use of dynamic keys. When a user sends a request to share the sensitive information of others, or the user gives permission for others to access the user's sensitive information, a token needs to be generated and sent to the LSGCs. The token is constructed by a unique dynamic key, known only to the user and system. This knowledge ensures the user is unable to deny issuing permissions or sending requests. In addition, the token is dynamically generated based on the user's dynamic communication key and is only used once. The token thus eradicates the security threat of sniffing attacks.

In addition, as described, when a user registers with the system, a unique secure ID id_i is generated based on the security levels by either biometrics or a secure random number. Therefore, the user has a lawful identity in the system to be representative of the user, and the id_i is enciphered by the combination of a dynamic communication key and its index value i . Whenever a user wishes to logon to the system, only the correct combination is able to verify the legitimacy of the user. Based on the security of DKM and former key secrecy, it is computationally infeasible to discover any

dynamic key; thus, the id_i is considered as a “signature” of the user in the system to achieve high assurance. Therefore, an axiom can be given.

Axiom 5.6 Entity P believes actions performed by entity Q with signature in SecureSIS.

Proof: Because of the Initialization and Logon protocols, we know the id_i of entity Q has never been involved in any transactions via either public channels or private channels. Meanwhile, P denotes the LSGC. Thus:

$$Q \text{ believes } fresh(id_i) \wedge P \text{ believes } fresh(id_i) \quad (5.89)$$

Also, only the genuine entity Q has the correct dynamic communication key. Q generates a correct token and sends to P . According to Axioms 5.1, 5.2 and 5.4:

$$P \text{ believes } Q \text{ said token} \quad (5.90)$$

Taking the token into Equation 4.2 to have a fresh id_i , if, and only if, the fresh id_i is same as the lawful identity in system:

$$P \text{ believes } Q \quad (5.91)$$

Because all actions performed by Q in SecureSIS can be seen by P and, according to security of AAM:

$$P \text{ sees } Q \text{ performs actions with token} \quad (5.92)$$

Then, combining (5.89), (5.91) and (5.92) and letting tokens denote with signs, we have:

$$\frac{P \text{ believes } fresh(token), P \text{ sees } Q \text{ performs an action with a token}}{P \text{ believes } Q \text{ performs the action with the token}} \quad (5.93)$$

Only the legitimate entity Q can have a token to generate a fresh signature, and only legitimate entity Q has a correct dynamic communication key set. According to Theorems 3.1, 3.2, and 3.3, all generated tokens by Q are secure. Therefore, a token is equivalent to a signature:

$$token \sqsubseteq sign \quad (5.94)$$

Taking (5.94) into (5.93):

$$\frac{P \text{ believes } fresh(sign), P \text{ sees } Q \text{ performs an action with a sign}}{P \text{ believes } Q \text{ performs the action with the sign}} \text{ vice versa (5.95)}$$

Therefore Equation 3.32 is satisfied in SecureSIS, and the proof is complete. \square

5.5.4. Confidentiality Discussion

Confidentiality is the property of preventing disclosure of sensitive information to unauthorized individuals or group of users in SecureSIS. Confidentiality emphasizes the secrecy of *communication channel*, *user interface* and *sensitive information storage*. The confidentiality of sensitive information is guaranteed by relying on the security of AAM to protect *user interface*, the security of UGKM and DKM to secure *communication channel* and the security of SIM to defend *sensitive information storage*.

As suggested in Section 3.3.5, confidentiality relates not only the security of the above three components, but also to the privacy or secrecy of sensitive information owners. Thus, in addition to Axioms 5.1-5.6, the privacy of owners of sensitive information should be protected. In other words, sensitive information owners should have fine-grain control over their assets.

Axiom 5.7 Entity P , the owner of sensitive information, has full control of its assets in SecureSIS.

Proof: Suppose entity Q wants to access the sensitive information $I_i \in I$ of entity P . According to the security of AAM and SIM, initially, entity Q has no knowledge of I_i and also has no permissions relating to I_i . Precisely:

$$\forall I_i \in I, P := I_i \wedge Q : AR(I_i, *) \equiv false \quad (5.96)$$

According to the security of the AccessAuth protocol, entity Q cannot distinguish communication among entities. Also, the security of DKM and UGKM guarantees that the “signature” is infeasible to compute (Theorem 3.1). Moreover, according to the security of SIM, without proper permissions, entity Q cannot understand the form of the sensitive information, even though Q is able to obtain the cipher form. Thus entity Q must request permission from the owner P in order to have access to sensitive information I_i . Combined with (5.96), we have:

$$\forall I_i \in I, P := I_i \wedge Q : AR(I_i, *) \equiv false, \frac{P \text{ authorizes } Q}{Q : AR(I_i, *) \equiv true} \quad (5.97)$$

Therefore Equation 3.33 is satisfied in SecureSIS, and the proof is complete. \square

5.5.5. Utility Discussion

Utility Utility relates to information usefulness. It is a baseline for the other four elements (discussed in Section 3.3.7). Utility impacts dynamic membership and emergency situations. The proposed SecureSIS employs UGKM to handle information sharing. UGKM enables a particular segment of information to be available to group of users. If the segment of sensitive information belongs to a group of users, the

dynamic data keys of the leader of the group are used to secure the data, thus enabling all group users to access the information. In addition, when a data owner permanently leaves the system, the ownership of data will be assigned to the leader of the nominated cluster.

At times, a data owner may not be able to provide a key to retrieve information, and the information may be required urgently (perhaps for medical reasons). This lack of access represents a breach of utility; the information is controlled, integral and authentic— but the information is not useful in its inaccessible form. In SIM, the use of *EL* solves the problem. *EL* contains the nominated cluster $c_n \in C$, an allocated auditing cluster $c_a \in C$ and an encrypted dynamic data key that enables access to the sensitive information of users normally inaccessible. This feature guarantees the usefulness of information. Thus Equations 3.34 and 3.35 are satisfied in SecureSIS.

5.5.6. SecureSIS Goals Discussion

Based on the proofs of Axioms 5.1-5.7 and the discussion on utility, the proposed security architecture satisfies the criteria of the SecureSIS pentad. Meeting these criteria ensures that SecureSIS has authenticity and authority, non-repudiation, integrity, confidentiality and utility properties to protect sensitive information. By using the theorems, corollaries and axioms already presented, in this section, we prove that SecureSIS also meets its intended 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 (Axioms

5.1 and 5.2). 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 (Axiom 5.3). In addition, in the discussion on CO, axiom 5.7 stated that if user u_i possesses the information, then u_i has full control of it. Moreover, the discussion of NR mentions 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 . Thus we have:

$$\begin{aligned} & \forall u_i \in U \forall I_j \in I (u_i \text{ CanProve } u_i \text{ to SecureSIS} \wedge \\ & u_i \text{ CanProve } AR(I_j, *) \equiv \text{true to SecureSIS}) \\ & \Rightarrow u_i := I_j \end{aligned} \quad (5.98)$$

Furthermore, as was discussed in the Logon protocol on AAM (Section 4.3.4) 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, we have:

$$\forall u_i \in U (\text{SecureSIS CanProve Genuine to } u_i) \quad (5.99)$$

Thus Equations 3.20 and 3.21 are proved ensuring that sensitive information is only disclosed to legitimate users with proper permissions and genuine SecureSIS. \square

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 on IN, it ensures that $\forall u_i \in U$ believes received sensitive information is identically maintained in transit (Axiom 5.4).

$$\forall u_i \in U \exists I_j \in I (\text{iff } u_i := I_j \Rightarrow u_i \text{ CanVerify } I_j \text{ is Genuine}) \quad (5.100)$$

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:

$$\forall I_j \in I (\text{SecureSIS CanVerify } I_j \text{ is Genuine}) \quad (5.101)$$

Thus Equations 3.22 and 3.23 are proved and ensure that sensitive information is identically maintained during transmission via open networks in SecureSIS. \square

Proof of Sensitive Information Storage's Goal. The security of *sensitive information storage* is attained by SIM participating with DKM and UGKM. As discussed in Section 3.3.3 on IN, $\forall u_i \in U$ believes possessed sensitive information is genuine in *sensitive information storage* (Axiom 5.5). We have:

$$\forall EI_j \in EI \exists u_i \in U (\text{iff } u_i := EI_j \Rightarrow u_i \text{ CanUnderstand } EI_j) \quad (5.102)$$

According to the discussion on CO and NR (Axioms 5.6 and 5.7), 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 we have:

$$\forall EI_j \in EI \exists u_i \in U (\text{iff } u_i := EI_j \Rightarrow u_i \text{ CanUnderstand } I_j) \quad (5.103)$$

Thus Equation 3.24 is proved and ensures that sensitive information is stored securely and only privileged users can understand and retrieve sensitive information in SecureSIS.

5.6. Summary

In this chapter, the security aspects of the four components of SecureSIS were formally proved and discussed according to the description of Chapter 4. The

SecureSIS pentad model was built based on Definition 3.9 in Chapter 3 to evaluate the security characteristics of sensitive information in SecureSIS. The evaluation shows the proposed security architecture satisfies the security requirements of sensitive information. According to the SecureSIS pentad evaluation, SecureSIS meets the user interface's goal, communication channel's goal and sensitive information storage's goal. The discussion and mathematical proofs demonstrate that the proposed security architecture makes the following contributions to the protection of sensitive information:

- The use of dynamic keys in SecureSIS ensures greater security than the long-term shared symmetric keys and the asymmetric keys of previous security models.
- The use of two sets of dynamic keys in SecureSIS guarantees that:
 - DKM achieves privacy protection and sensitive information systems intrusion detection and prevention.
 - UGKM satisfies group key secrecy, forward secrecy, backward secrecy and collusion resistibility.
 - AAM ensures authenticity and secrecy, and also ensures that an adversary cannot distinguish between sensitive information and random text in transit between entities.
 - SIM has sensitive information interchange secrecy and sensitive information storage secrecy properties.

By evaluating SecureSIS against the SecureSIS pentad model, we demonstrated that in SecureSIS the characteristic(s) of:

- authenticity and authority ensure that sensitive information is securely shared either among a group of users or can be retrieved by individuals,
- integrity ensures that sensitive information is identically maintained in communication and in storage,
- non-repudiation guarantees that entities are unable to deny performing actions on sensitive information,
- confidentiality ensures sensitive information in any form is protected and secure, and
- utility ensures that, in any circumstance, sensitive information is useful.

The theorems, corollaries and axioms presented in this chapter have also demonstrated that SecureSIS meets its intended goals:

- Sensitive information is only disclosed to legitimate users with proper permissions and genuine SecureSIS (the user interface's goal).
- Sensitive information is identically maintained during transmission via open networks (the communication channel's goal).
- Sensitive information is stored securely and only privileged users can understand and retrieve the information (the sensitive information storage's goal).

Chapter 6

Conclusion and Future Work

Goals. This thesis has investigated sensitive information security in SIS. The limitations of extant security approaches (caused by the employment of long-term shared and public keys) and the resulting issues relating to dynamic sensitive information ownership, group authentication and authorization and privacy protection motivated us to propose a new security architecture. SecureSIS eliminates the limitations and issues of current approaches by applying dynamic key and group key theories.

In addition to the new security architecture, a new sensitive information security model, the SecureSIS pentad, was also proposed in this thesis. This model overcomes the lack of the assessment properties of extant information security models. The SecureSIS pentad also demonstrates that SecureSIS meets its security goals.

In this chapter, the aims and methodology of the research are reviewed in Section 6.1. The contributions offered over the previous five chapters are restated in Section 6.2. Finally, future work possibilities are identified in Section 6.3.

6.1. Revisiting the Research Problem and Approach

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. As discussed in Chapter 2, previous research in this area has been limited. After assessing the state of prior research, the objectives of this research were defined as follows:

- To develop a general security architecture for various kinds of sensitive information systems that enables the protection of the three components *communication channel*, *user interface* and *sensitive information storage*. The architecture should be able to:
 - handle dynamic membership of groups and individuals and enable the appropriate sharing or accessing of sensitive information,
 - prevent legal users accessing unauthorized sensitive information to prevent internal security threats,
 - manage dynamic ownership of sensitive information, and
 - govern the security of information storage should the sensitive information system be breached.

- To develop methodological recommendations for security evaluation of sensitive information systems.

To achieve these aims, a research methodology was structured on a three-stage process: development of a formal architecture, component design, and security discussion and assessment.

- Development of a formal architecture (Chapter 3): a new security architecture for sensitive information systems (SecureSIS) was formally proposed as the first step in protecting the three major components (*communication channel*, *user interface* and *sensitive information storage*) and to achieve the security goals of the security architecture. A sensitive information security model (SecureSIS pentad) was suggested to assess the security of the proposed architecture.
- Component design (Chapter 4): as per the proposed and defined formal architecture (SecureSIS), each component was designed and guided by the proposed SecureSIS pentad model.
- Security discussion and assessment (Chapter 5): a security discussion on each designed component was formally conducted, and then the proposed sensitive information security model was built to assess the proposed architecture and to prove that the security goals were met.

6.2. 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 (Definition 3.1) with the following proved cryptographic properties: dynamic key secrecy (Theorem 3.1), former key secrecy (Theorem 3.2), key collision resistance (Theorem 3.3) and key consistency (Theorem 3.4). These theorems were used to prove the correctness of two presumptions (Theorem 3.5 and 3.6) in sensitive information protection: i) that dynamic keys are more secure than long-term shared keys; and ii) that, compared with dynamic keys, asymmetric keys are insecure. 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 thesis proposed a novel security architecture, SecureSIS, 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.

- A new proposed sensitive information security model for sensitive information systems

This thesis proposed a new sensitive information security model, the SecureSIS pentad, to assess the security of SecureSIS. The SecureSIS pentad can also be used to assess other security architectures, thus making a valuable contribution to the field of sensitive information system security.

- Development of dynamic key management (DKM)

This thesis developed a dynamic key management approach (discussed in Section 5.1.1) that employs two sets of dynamic keys (dynamic data keys and dynamic communication keys) to guarantee the security of sensitive information. Using this approach, even if one set of dynamic keys were to be compromised, the security of SecureSIS would not be breached. This approach is also able to detect and prevent intrusion in sensitive information systems.

- Development of user-oriented group key management (UGKM)

This thesis developed a hybrid group key agreement (UGKM) to deal with dynamic group membership in sensitive information sharing and privacy protection. The agreement adopts the properties of dynamic keys to guarantee the security of sensitive information in transit among entities (*communication channel*). The agreement enables group key secrecy (Section 5.2.1), forward secrecy (Section 5.2.2), backward secrecy (Section 5.2.3) and collusion resistance (Section 5.2.4).

- Development of authentication and authorization management (AAM)

This thesis developed an authentication and authorization management approach to protect *user interface*. AAM achieves high security and tight access control when dealing with dynamic membership of groups and individuals that share or access sensitive information. AAM enables dynamic ownership of sensitive information, flexible access control and strong identity verification.

- Design of sensitive information management (SIM)

This thesis applied DKM and UGKM to design a new approach for the protection of sensitive information at rest. SIM integrates dynamic keys with sensitive information stored in the form of a cipher to avoid leaking sensitive information in the case of unauthorized access. SIM guarantees that sensitive information interchange is secure should the *communication channel* and *user interface* components be compromised. That is, a breach of *sensitive information storage* will not threaten the security of sensitive information, and the security of other users and sensitive information will not be compromised should the security of some users be breached.

6.3. Future Work

This research has opened up avenues for further work. These include i) investigation into the use of dynamic keys for intrusion prevention and detection; ii) the design and development of new dynamic key algorithms; and iii) the amelioration of the sensitive information security model (SecureSIS pentad).

This thesis 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. We have begun some work in this direction [DaWuWa08, WuLeSr09].

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. We have designed one realization (a dynamic key generation algorithm) and it is being patented [WuLe06].

The usefulness of the SecureSIS pentad in assessing other security architectures also needs to be firmly established. Where possible, we suggest the adoption of methodologies similar to those used in the earlier studies that evaluated the CIA Triad and the Parkerian Hexad, but with the substitution of the SecureSIS pentad. A comparison of the results would yield insight and enable further finetuning of the SecureSIS pentad.

In conclusion, SecureSIS overcomes the limitations associated with existing security approaches and enables the complete protection of the three components of sensitive information systems. The results from our study are both a catalyst and a justification for further research in this area to increase the body of scientific knowledge concerning sensitive information protection.

References

- [Ab99] Abadi, M. (1999). Secrecy by typing in security protocols. *Journal of the ACM*, Vol. 46 (5), pp. 749 - 786.
- [AbGo97] Abadi, M., and Gordon, A. D. (1997). *A calculus for cryptographic protocols: the spi calculus*. In Proceedings of the Computer and Communications Security, pp. 36-47.
- [AnKu96] Anderson, R., and Kuhn, M. (1996). *Tamper Resistance - a Cautionary Note*. In Proceedings of the 2nd Usenix Workshop on Electronic Commerce, pp. 1.
- [AnAn01] Anderson, R. J., and Anderson, R. (2001). *Security Engineering: A Guide to Building Dependable Distributed Systems* John Wiley & Sons.
- [ArCaLi04] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and Norrman, K. (2004). *MIKEY: Multimedia Internet KEYing* (No. RFC 3830): Network Working Group, The Internet Engineering Task Force.
- [At95] Atkinson, R. (1995). *Security Architecture for the Internet Protocol* (No. RFC 1825): Network Working Group, The Internet Engineering Task Force.
- [BaFi01] Bacon, C. J., and Fitzgerald, B. (2001). A systemic framework for the field of information systems. *ACM SIGMIS Database* Vol. 32 (2), pp. 46 - 67.
- [Ba96] Ballardie, T. (1996). *Scalable Multicast Key Distribution* (No. RFC 1949): Network Working Group, The Internet Engineering Task Force.
- [Ba04] Bard, G. V. (2004). *The vulnerability of ssl to chosen-plaintext attack* (No. 2004/111): Cryptology ePrint Archive.

- [BaBaBu06] Barker, E., Barker, W., Burr, W., Polk, W., and Smid, M. (2006). *Recommendation for Key Management*. The National Institute of Standards and Technology.
- [BeWi98] Becker, K., and Wille, U. (1998). *Communication Complexity of Group Key Distribution*. In Proceedings of the 5th ACM Conference on Computer and Communications Security pp. 1-6.
- [BeIsKu99] Beimel, A., Ishai, Y., Kushilevitz, E., and Malkin, T. (1999). *One-way Functions are Essential for Single-Server Private Information Retrieval*. In Proceedings of the 31st Annual ACM Symposium on Theory of Computing, pp. 89-98.
- [BeCaKr96] Bellare, M., Canett, R., and Krawczyk, H. (1996). *Pseudorandom functions revisited: the cascade construction and its concrete security*. In Proceedings of the 37th Annual Symposium on Foundations of Computer Science pp. 514-523.
- [BeMe92] Bellare, S. M., and Merritt, M. (1992). *Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks*. In Proceedings of the IEEE Symposium on Research in Security and Privacy, pp. 72-84.
- [BeMe93] Bellare, S. M., and Merritt, M. (1993). *Augmented Encrypted Key Exchange: a Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise*. In Proceedings of the 1st ACM Conference on Computer and Communication Security, pp. 244-250.
- [BeGoGo88] Ben-Or, M., Goldreich, O., Goldwasser, S., Håstad, J., Kilian, J., Micali, S., et al. (1988). *Everything Provable is Provable in Zero-Knowledge*. In Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology, pp. 37-56.
- [BhDe98] Bhatia, S. K., and Deogun, J. S. (1998). Conceptual clustering in information retrieval. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 28 (3), pp. 427-436.
- [Bi96] Biham, E. (1996). *How to Forge DES-Encrypted Messages in 2^{28} Steps* (No. CS 884): Technion-Israel Institute of Technology.

- [BiSh93] Biham, E., and Shamir, A. (1993). *Differential Cryptanalysis of the Data Encryption Standard* New York. Springer-Verlag.
- [BiKl95] Bishopa, M., and Klein, D. V. (1995). Improving System Security via Proactive Password Checking. *Computers and Security*, Vol. 14 (3), pp. 233–249.
- [Bo99] Boneh, D. (1999). Twenty Years of Attacks on the RSA Cryptosystem. *Notices of the AMS*, Vol. 46 (2), pp. 203-213.
- [Bo07] 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
- [Br73] Branstad, D. K. (1973). *Security aspects of computer networks*. In Proceedings of the AIAA Computer Network Systems Conference, pp. 73.
- [Br99] Briscoe, B. (1999). *MARKS: Zero Side Effect Multicast Key Management using Arbitrarily Revealed Key Sequences*. In Proceedings of the 1st International Workshop on Networked Group Communication, pp. 301-320.
- [BrFo05] Bruen, A. A., and Forcinito, M. A. (2005). *Cryptography, Information Theory, and Error-Correction: a Handbook for the 21st Century*. Wiley-Interscience.
- [BS93] BSI. (1993). *Code of Practice for Information Security Management (CoP)* (No. PD 0003): British Standards Institute.
- [Bu06] Burg, T. (2006). Data at rest encryption on the NonStop platform. *SunTUG*, Retrieved 31 March 2009, from http://www.suntug.org/Articles/2006-06_comForte_data-at-rest_Encryption.pdf
- [CaMiSt99] Cachin, C., Micali, S., and Stadler, M. (1999). Computationally Private Information Retrieval with Polylogarithmic Communication In *Advances in Cryptology* Vol. 1592/1999, pp. 402-414: Springer Berlin / Heidelberg.

- [CaWaSu98] Caronni, G., Waldvogel, M., Sun, D., and Plattner, B. (1998). *Efficient communication for large and dynamic multicast groups*. In Proceedings of the 7th IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 376-383.
- [CeJaSc08] Cervesato, I., Jaggard, A. D., Scedrov, A., Tsay, J.-K., Christopher, and Walstad. (2008). Breaking and fixing public-key Kerberos. *Information and Computation* Vol. 206 (2-4), pp. 402-424.
- [ChGeRu90] Champine, G. A., Daniel E. Geer, J., and Ruh, W. N. (1990). Project Athena as a distributed computer system. *Computer*, Vol. 23 (9), pp. 40-51.
- [Ch97] Chan, S. C. C. (1997). An overview of smart card security. *Break IC* Retrieved 22 December, 2008, from <http://www.break-ic.com/topics/attack-microcontroller.asp>
- [ChWaWu06] Chen, Y. J., Wang, Y. L., Wu, X. P., and Le, P. D. (2006). *The Design of Cluster-based Group Key Management System in Wireless Networks*. In Proceedings of the International Conference on Communication Technology, pp. 1-4.
- [ClChCh08] Clark, C., Chaffin, L., Chuvakin, A., Fogie, S., Schiller, C., Paladino, S., et al. (2008). *InfoSecurity 2008 Threat Analysis* Burlington, Massachusetts, USA. Syngress Publishing, Inc.
- [ClWi87] Clark, D. D., and Wilson, D. R. (1987). *A comparison of commercial and military computer security policies*. In Proceedings of the IEEE Symposium on Security and Privacy pp. 184-194.
- [Cn92] CNSS. (1992). *National Training Program for Information Systems Security (INFOSEC) Professionals*, National Security Telecommunications and Information Systems Security Committee.
- [Co03] Cole, S. A. (2003). *SUSPECT IDENTITIES: A History of Fingerprinting and Criminal Identification*. Harvard University Press.
- [CoDiWa04] Conklin, A., Dietrich, G., and Walz, D. (2004). *Password-Based Authentication: A System Perspective*. In Proceedings of the 37th Annual Hawaii International Conference on System Sciences, pp. 1-10.

- [CoBr93] Cooke, J. C., and Brewster, R. L. (1993). *The Use of Smart Cards in Personal Communication Systems Security*. In Proceedings of the 4th IEE Conference on Telecommunications, pp. 246-251.
- [Co97] Coppersmith, D. (1997). Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities. *Journal of Cryptology*, Vol. 10 (4), pp. 233-260.
- [Co06] Corio, C. (2006). First Look: New Security Features in Windows Vista. *TechNet Magazine*, Retrieved 2 January, 2009, from <http://technet.microsoft.com/en-us/magazine/cc160980.aspx>
- [CoPi02] Courtois, N. T., and Pieprzyk, J. (2002). Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. In *Advances in Cryptology*, Vol. 2501, pp. 267-287: Springer Berlin / Heidelberg.
- [CuEu08] Curtiss, E. T., and Eustis, S. (2008). *Physician Office Electronic Medical Record Market Strategies Shares, and Forecasts 2007 to 2013*. Ireland: WinterGreen Research, Inc.
- [DaRi02] Daemen, J., and Rijmen, V. (2002). *The Design of Rijndael: AES- The Advanced Encryption Standard* Berlin Heidelberg. Springer-Verlag.
- [DaWuWa08] Dandash, O., Wu, X. P., Wang, Y. L., Le, P. D., and Srinivasan, B. (2008). Security Analysis for Internet Banking Models. *Special Issue of the International Journal of Computer and Information Science*, Vol. 9 (2), pp. 1-10.
- [DaOl85] Davis, G. B., and Olson, M. H. (1985). *Management Information Systems: Conceptual Foundations, Structure, and Development* New York. Mcgraw-Hill.
- [De89] Dehnad, K. (1989). A simple way of improving the login security. *Computers and Security*, Vol. 8 (7), pp. 607-611.
- [DiRe08] Dierks, T., and Rescorla, E. (2008). *The Transport Layer Security (TLS) Protocol (V1.2)* (No. RFC 5246): Network Working Group, The Internet Engineering Task Force.

- [DiHe76a] Diffie, W., and Hellman, M. (1976). *Multiuser Cryptographic Techniques*. In Proceedings of the National Computer Conference, pp. 109-112.
- [DiHe76b] Diffie, W., and Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, Vol. 22 (6), pp. 644-654.
- [DiOoWi92] Diffie, W., Oorschot, P. C. V., and Wiener, M. J. (1992). Authentication and Authenticated Key Exchanges. *Designs, Codes and Cryptography*, Vol. 2 (2), pp. 107-125.
- [Ds07] DSB. (2007). *Information Management for Net-Centric Operations*. Washington, D. C.: Office of the Under Secretary of Defense For Acquisition, Technology, and Logistics.
- [DuJuKo02] Dugelay, J. L., Junqua, J. C., Kotropoulos, C., Kuhn, R., Perronnin, F., and Pitas, I. (2002). *Recent advances in biometric person authentication*. In Proceedings of the IEEE conference on Acoustics, Speech, and Signal Processing, pp. 4060-4063.
- [Er03] Erdem, O. M. (2003). *High-speed ECC based Kerberos Authentication Protocol for Wireless Applications*. In Proceedings of the IEEE Global Telecommunications Conference, pp. 1440-1444.
- [Fe70] Feistel, H. (1970). *Cryptographic coding for data-bank privacy* (No. RC 2827). Yorktown Heights, New York: International Business Machines Corporation.
- [Fe73] Feistel, H. (1973). Cryptography and Computer Privacy. *Scientific American*, Vol. 228 (5), pp. 15-23.
- [FeKa90] Feldmeier, D. C., and Karn, P. R. (1990). *UNIX Password Security - Ten Years Later*. In Proceedings of the Advances in Cryptology, pp. 44-63.
- [FeKuCh03] Ferraiolo, D. F., Kuhn, D. R., and Chandramouli, R. (2003). *Role-Based Access Control* Norwood, USA. Artech House Publishers.

- [FoFe03] Forouzan, B. A., and Fegan, S. C. (2003). *Data Communications and Networking*. McGraw-Hill Science/Engineering/Math.
- [FrKaKo96] Freier, A. O., Karlton, P., and Kocher, P. C. (1996). *The SSL Protocol (V3.0)*: Transport Layer Security Working Group.
- [GeLi06] Gennaro, R., and Lindell, Y. (2006). A framework for password-based authenticated key exchange. *ACM Transactions on Information and System Security*, Vol. 9 (2), pp. 181-234.
- [GeGoMa98] Gertner, Y., Goldwasser, S., and Malkin, T. (1998). A Random Server Model for Private Information Retrieval In *Randomization and Approximation Techniques in Computer Science*, Vol. 1518/1998, pp. 200-217: Springer Berlin / Heidelberg.
- [GeIsKu00] Gertner, Y., Ishai, Y., Kushilevitz, E., and Malkin, T. (2000). Protecting data privacy in private information retrieval schemes. *Journal of Computer and System Sciences* Vol. 60 (3), pp. 592-629.
- [GoGo96] Gordon, S. R., and Gordon, J. R. (1996). *Information Systems: A Management Approach* Orlando, Florida. The Dryden Press, Harcourt Brace College Publishers.
- [Gr90] Gray, R. M. (1990). *Entropy and Information Theory* New York. Springer-Verlag.
- [Gu06] Guimaraes, M. (2006). *New challenges in teaching database security*. In Proceedings of the 3rd annual conference on Information security curriculum development, pp. 64-67.
- [GuSh07] Gupta, P., and Shmatikov, V. (2007). *Security Analysis of Voice-over-IP Protocols*. In Proceedings of the 20th IEEE Computer Security Foundations Symposium, pp. 49-63.
- [HaScHe08] Halderman, J. A., Schoen, S. D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J. A., et al. (2008). *Lest We Remember: Cold Boot Attacks on Encryption Keys*. In Proceedings of the 17th USENIX Security Symposium, pp. 45-60.

- [HaKr99] Halevi, S., and Krawczyk, H. (1999). Public-key cryptography and password protocols. *ACM Transactions on Information and System Security*, Vol. 2 (3), pp. 230-268.
- [Ha94] Haller, N. (1994). *The S/KEY One-Time Password System*. In Proceedings of the ISOC Symposium on Network and Distributed System Security, pp. 151-157.
- [HaAt94] Haller, N., and Atkinson, R. (1994). *On Internet Authentication* (No. RFC 1704): Network Working Group, The Internet Engineering Task Force.
- [HaMe01] Harbitter, A., and Menascé D. A. (2001). *The performance of public key-enabled kerberos authentication in mobile computing applications*. In Proceedings of the 8th ACM conference on Computer and Communications Security pp. 78-85.
- [HaCaMo00] Hardjono, T., Cain, B., and Monga, I. (2000). *Intra-Domain Group Key Management Protocol*: Network Working Group, The Internet Engineering Task Force.
- [HaHa99] Harney, H., and Harder, E. (1999). *Logical Key Hierarchy Protocol*: Network Working Group, The Internet Engineering Task Force.
- [HaMu97a] Harney, H., and Muckenhirn, C. (1997). *Group Key Management Protocol (GKMP) Architecture* (No. RFC 2094): Network Working Group, The Internet Engineering Task Force.
- [HaMu97b] Harney, H., and Muckenhirn, C. (1997). *Group Key Management Protocol (GKMP) Specification* (No. RFC 2093): Network Working Group, The Internet Engineering Task Force.
- [He78] Hellman, M. E. (1978). An Overview of Public Key Cryptography. *IEEE Communicaiton Society Magazine*, Vol. 16 (6), pp. 24-32.
- [Ho05] Hoffman, P. (2005). *Cryptographic Suites for IPsec* (No. RFC 4308): Network Working Group, The Internet Engineering Task Force.

- [HoChWa07] Hong, W.-S., Chen, S.-J., Wang, L.-H., and Chen, S.-M. (2007). A new approach for fuzzy information retrieval based on weighted power-mean averaging operators. *Computers & Mathematics with Applications* Vol. 53 (12), pp. 1800-1819.
- [Ho02] Horng, G. (2002). Cryptanalysis of A Key Management Scheme for Secure Multicast Communications. *IEICE Transactions on Communications* Vol. E85-B (5), pp. 1050-1051.
- [Hp07] HP. (2007). *Encrypted Volume and File System v1.0 (EVFS v1.0) Administrator's Guide HP-UX 11i v2 Update 2* (No. 5991-7466): Hewlett-Packard Development Company.
- [Hs08] Hsueh, S. (2008). Database Encryption in SQL Server 2008 Enterprise Edition. *SQL Server Technical Article* Retrieved 28 April, 2008, from [http://msdn2.microsoft.com/en-us/library/cc278098\(SQL.100\).aspx](http://msdn2.microsoft.com/en-us/library/cc278098(SQL.100).aspx)
- [Hy08] Hynes, B. (2008). Advances in BitLocker Drive Encryption. *TechNet Magazine*, (6).
- [Id08] IDTheftProtect. (2008). ID THEFT PROTECT:News and Reviews. *Global: BoxSentry and Internet Identity join forces to fight online fraud* Retrieved 23 February, 2009, from http://www.id-theftprotect.com/news.php?news_id=261
- [InTaWo82] Ingemarsson, I., Tang, D. T., and Wong, C. K. (1982). A Conference Key Distribution System. *IEEE Transactions on Information Theory*, Vol. 28 (5), pp. 714-720.
- [IsSu01] Ismadi, A., and Sukaimi, Y. B. (2001). *Smart Card- An Alternative to Password Authentication: SANS Security Essentials*.
- [Ja96a] Jablon, D. P. (1996). Strong password-only authenticated key exchange. *ACM SIGCOMM Computer Communication Review*, Vol. 26 (5), pp. 5-26.
- [JaRoPr04] Jain, A. K., Ross, A., and Prabhakar, S. (2004). An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14 (1), pp. 4-20.

- [Ka04] Kaeo, M. (2004). *Designing Network Security* Indianapolis, Indiana, USA. Cisco Press.
- [Ka67] Kahn, D. (1967). *The Codebreakers: The Story of Secret Writing* New York. Macmillan Pub Co.
- [Ke77] Kent, S. T. (1977). *Encryption-based protection for interactive user/computer communication*. In Proceedings of the 5th symposium on Data communications, pp. 5.7 - 5.13.
- [Kh06] Khare, R. (2006). *Network Security and Ethical Hacking* Beckington, UK. Luniver Press.
- [KiPeTs04] Kim, Y., Perrig, A., and Tsudik, G. (2004). Tree-based group key agreement. *ACM Transactions on Information and System Security*, Vol. 7 (1), pp. 60 - 96.
- [Ko87] Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation*, Vol. 48 (1), pp. 203–209.
- [KoNeTs94] Kohl, J. T., Neuman, B. C., and T'so, T. Y. (1994). *The Evolution of the Kerberos Authentication System*. In Proceedings of the Distributed Open Systems, pp. 78-94.
- [KoOh87] Koyama, K., and Ohta, K. (1987). *Identity-based Conference Key Distribution Systems*. In Proceedings of the Theory and Applications of Cryptographic Techniques on Advances in Cryptology, pp. 175-184.
- [Kr96] Krawczyk, H. (1996). *SKEME: A Versatile Secure Key Exchange Mechanism for Internet*. In Proceedings of the Symposium on Network and Distributed System Security, pp. 114-127.
- [KuCh03] Ku, W. C., and Chen, S. M. (2003). *An improved key management scheme for large dynamic groups using one-way function trees*. In Proceedings of the International Conference on Parallel Processing Workshops, pp. 391- 396.

- [Ku05] Kungpisdan, S. (2005). *Modelling, Design, and Analysis of Secure Mobile Payment Systems*, PhD Thesis, Monash Univeristy, Melbourne, Australia.
- [KuLeSr05] Kungpisdan, S., Le, P. D., and Srinivasan, B. (2005). A Limited-Used Key Generation Scheme for Internet Transactions. *Lecture Notes in Computer Science*, Vol. 3325, pp. 302-316.
- [La81] Lamport, L. (1981). Password authentication with insecure communication. *Communications of the ACM* Vol. 24 (11), pp. 770 - 772.
- [LaBrHa85] Latham, D. C., Brand, S. L., Hammonds, G., Tasker, P. S., Edwards, D. J., and Schell, R. R. (1985). *Department of Defense Trusted Computer System Evaluation Criteria*, DoD 5200.28, US National Security Institute.
- [LaMo08] Lavasani, A., and Mohammadi, R. (2008). Implementing a feasible attack against ECC2K-130 certicom challenge. *ACM Communications in Computer Algebra*, Vol. 42 (1), pp. 61-62.
- [LeLeYo05] Lee, H. J., Lee, S. J., Yoon, J. H., Cheon, D. H., and Lee, J. I. (2005). *The SEED Encryption Algorithm* (No. RFC 4009): Network Working Group, The Internet Engineering Task Force.
- [LeNaNo07] Lehtovirta, V., Naslund, M., and 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.
- [LeVe01] Lenstra, A. K., and Verheul, E. R. (2001). Selecting cryptographic key sizes. *Journal of Cryptology*, Vol. 14 (4), pp. 255–293.
- [Li08] Li, Q. (2008). China: eBanking IT Solutions Market 2008-2012 Forecast and Analysis. Retrieved 25 August, 2008, from <http://www.marketresearch.com/product/display.asp?productid=1869884&xs=r&SID=35378328-424089109-459063949&curr=USD&kw=&view=abs>

- [LiZh04] Li, Y., and Zhang, X. (2004). *A Security-Enhanced One-Time Payment Scheme for Credit Card*. In Proceedings of the 14th International Workshop on Research Issues on Data Engineering: Web Services for E-Commerce and E-Government Applications, pp. 40-47.
- [Lu98a] Lucks, S. (1998). Attacking Triple Encryption. In *Fast Software Encryption*, Vol. 1372, pp. 239-253: Springer Berlin / Heidelberg.
- [Lu98b] Lucks, S. (1998). *Open key exchange: How to defeat dictionary attacks without encrypting public keys*. In Proceedings of the Workshop on Security Protocols, pp. 79-90.
- [Ma99] Marks, L. (1999). *Between Silk and Cyanide: A Codemaker's War, 1941-1945*. HarperCollins.
- [Ma94] Matsui, M. (1994). The First Experimental Cryptanalysis of the Data Encryption Standard. In *Advances in Cryptology*, Vol. 839, pp. 1-11: Springer Berlin / Heidelberg.
- [MaScSc98] Maughan, D., Schertler, M., Schneider, M., and Turner, J. (1998). *Internet Security Association and Key Management Protocol (ISAKMP)* (No. RFC 2408): Network Working Group, The Internet Engineering Task Force.
- [Mc07] McAfee. (2007). *McAfee Endpoint Encryption*: McAfee, Inc.
- [McAtMe95] McDonald, D. L., Atkinson, R. J., and Metz, C. (1995). *One time passwords in everything (OPIE): experiences with building and using stronger authentication*. In Proceedings of the 5th conference on USENIX UNIX Security Symposium, pp. 16-27.
- [MeIIKa00] Mena, E., Illarramendi, A., Kashyap, V., and Sheth, A. P. (2000). OBSERVER: An Approach for Query Processing in Global Information Systems Based on Interoperation Across Pre-Existing Ontologies *Distributed and Parallel Databases*, Vol. 8 (2), pp. 223-271.
- [Me96] Menezes, A. (1996). *Handbook of Applied Cryptography*. CRC Press.

- [Me79] Merkle, R. C. (1979). *Secrecy, Authentication, and Public Key Systems*, PhD Thesis, Stanford University, California, United States.
- [Me02] Meyers, R. A. (2002). *Encyclopedia of Physical Science and Technology* the University of Michigan. Academic Press.
- [MiBrLa02] Micheli, A. D., Brunessaux, S., Lakshmeshwar, S., Bosselaers, A., and Parkinson, D. (2002). *Investigations about SSL: MATRA Syst èmes & Information*, NOKIA Research Centre, K.U.Leuven Research & Development and British Telecommunications.
- [Mi86] Miller, V. S. (1986). *Use of elliptic curves in cryptography*. In *Proceedings of the Advances in cryptology*, pp. 417-426.
- [MiCh96] Mitchell, C. J., and Chen, L. (1996). Comments on the S/KEY user authentication scheme. *ACM SIGOPS Operating Systems Review*, Vol. 30 (4), pp. 12-16.
- [Mi97] Mitra, S. (1997). *Iolus: A framework for scalable secure multicasting*. In *Proceedings of the ACM SIGCOMM*, pp. 277-288.
- [MoTh79] Morris, R., and Thompson, K. (1979). Password security: a case history. *Communications of ACM*, Vol. 22 (11), pp. 594-597.
- [MoRaRo99] Moyer, M. J., Rao, J. R., and Rohatgi, P. (1999). A survey of security issues in multicast communications. *IEEE Network* Vol. 13 (6), pp. 12-23.
- [Na05] Nanda, A. (2005, 28 April 2008). Encrypt Your Data Assets: Build a flexible infrastructure to protect sensitive data. *Oracle Magazine*.
- [Nb88] National.Bureau.of.Standards. (1988). *Data Encryption Standard (DES)*. FIPS PUB 46-1, U.S. Department of Commerce & National Institute of Standards and Technology, Federal Information Processing Standards Publications.
- [Nb93] National.Bureau.of.Standards. (1993). *Data Encryption Standard (DES)*. FIPS PUB 46-2, U.S. Department of Commerce & National Institute of

Standards and Technology, Federal Information Processing Standards Publications.

- [Nb99] National.Bureau.of.Standards. (1999). *Data Encryption Standard (DES)*. FIPS PUB 46-3, U.S. Department of Commerce & National Institute of Standards and Technology, Federal Information Processing Standards Publications.
- [Nb01] National.Bureau.of.Standards. (2001). *Advanced Encryption Standard(AES)*. FIPS PUB 197, U.S. Department of Commerce & National Institute of Standards and Technology, Federal Information Processing Standards Publications.
- [Nc03] NCSTSD. (2003). *Classified National Security Information: Final Rule*. 183, National Archives and Records Administration, Federal Register.
- [NeSc78] Needham, R. M., and Schroeder, M. D. (1978). Using encryption for authentication in large networks of computers. *Communications of the ACM*, Vol. 21 (12), pp. 993-999.
- [NeYuHa05] Neuman, C., Yu, T., Hartman, S., and Raeburn, K. (2005). *The Kerberos Network Authentication Service (V5)* (No. RFC 4120): Network Working Group, The Internet Engineering Task Force.
- [NgWuLe08a] Ngo, H. H., Wu, X. P., and Le, P. D. (2008). *A Group authentication model for wireless network services based on group key management*. In Proceedings of the International Conference on Enterprise Information Systems, pp. 182-188.
- [NgWuLe08b] Ngo, H. H., Wu, X. P., Le, P. D., and Wilson, C. (2008). *A Method for Authentication Services in Wireless Networks*. In Proceedings of the 14th Americas Conference on Information Systems, pp. 1-9.
- [NgWuLe09a] Ngo, H. H., Wu, X. P., Le, P. D., and Wilson, C. (2009). Dynamic Key Cryptographic and Applications. *To Appear for Journal of Information System Security*.
- [NgWuLe09] Ngo, H. H., Wu, X. P., Le, P. D., and Wilson, C. (2009). *Package-Role Based Authorization Control Model for Wireless Network Services*. In

Proceedings of the 4th International Conference on Availability, Reliability and Security, pp. 475-480.

- [No98] Noubir, G. (1998). *Multicast security: Performance Optimisation of Interner Protocol Via Satellite* (No. 20): European Space Agency.
- [Oe92] OECD. (1992). *Organization for Economic Cooperation and Development (OECD) Guidelines for the Security of Information Systems*: Organization for Economic Cooperation and Development.
- [Op96] Oppliger, R. (1996). *Authentication Systems for Secure Networks* Norwood, MA , USA. Artech House Publishers.
- [Op01] Oppliger, R. (2001). *Internet and Intranet Security* Norwood, MA , USA. Artech House Publishers.
- [OrMcBa04] Oran, D., McGrew, D., Baugher, M., Naslund, M., Carrara, E., Norman, K., et al. (2004). *The Secure Real-time Transport Protocol (SRTP)* (No. RFC 3711): Network Working Group, The Internet Engineering Task Force.
- [Or98] Orman, H. (1998). *The OAKLEY Key Determination Protocol* (No. RFC 2412): Network Working Group, The Internet Engineering Task Force.
- [OrVa03] Ornaghi, A., and Valleri, M. (2003). *Man in the middle attacks Las Vegas, NV,USA*: Black Hat.
- [OsShTr06] Osvik, D. A., Shamir, A., and Tromer, E. (2006). *Cache Attacks and Countermeasures: The Case of AES*. In *Topics in Cryptology – CT-RSA 2006*, Vol. 3860, pp. 1-20: Springer Berlin / Heidelberg.
- [Pa98] Parker, D. B. (1998). *Fighting Computer Crime: A new Framework for Protecting Information* New York. Wiley Computer Publishing, John Wiley & Sons, Inc.
- [Pa97] Patel, S. (1997). *Number theoretic attacks on secure password schemes*. In *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 236-247.

- [PeKa01] Perlman, R., and Kaufman, C. (2001). *Analysis of the IPsec Key Exchange Standard*. In Proceedings of the 10th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2001, pp. 150-156.
- [Pe08] Perrin, C. (2008). The CIA principle. *IT Security* Retrieved 12 January, 2009, from <http://blogs.techrepublic.com.com/security/?p=488>
- [Pg08] PGP.Corporation. (2008). *PGP® Whole Disk Encryption 9.9*: PGP Corporation.
- [Po99] Poore, R. S. (1999). *Generally Accepted System Security Principles (GASSP) Version 2.0*: The Institute of Internal Auditors.
- [PrPaJa03] Prabhakar, S., Pankanti, S., and Jain, A. K. (2003). Biometric recognition: Security and privacy concerns. *IEEE Security Privacy (1)*, 33-42.
- [Pu07] Purpura, P. (2007). *Security and Loss Prevention: An Introduction* New York. Butterworth-Heinemann, Oxford.
- [RiShAd78] Rivest, R., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, Vol. 21 (2), pp. 120-126.
- [RoSa05] Robertson, S., and Salmond, T. (2005). Phishing attack targets one-time passwords: Scratch it and weep. *The Register* Retrieved 4 November, 2007, from http://www.theregister.co.uk/2005/10/12/outlaw_phishing/
- [OhKeDa00] Rodeh, O., Birman, K. P., and Dolev, D. (2000). *Optimized Group Rekey for Group Communication Systems*. In Proceedings of the Network and Distributed System Security, pp. 39-48.
- [RuWr02] Rubin, A. D., and Wright, R. N. (2002). *Off-Line Generation of Limited-Use Credit Card Numbers*. In Proceedings of the 5th International Conference on Financial Cryptography, pp. 196-209.
- [Sa88] Saenger, W. (1988). *Principles of Nucleic Acid Structure* New York. Springer.

- [Sa03] Salomon, D. (2003). *Data Privacy and Security* New York. Springer-Verlag.
- [Sc94] Scheaffer, R. L. (1994). *Introduction to Probability and Its Applications* Washington. Wadsworth Publishing Company, Duxbury Press.
- [ScWhWa00] Schneier, B., Kelsey, J., Whiting, D., Wagner, D., Hall, C., Ferguson, N., et al. (2000). *The Twofish Team's Final Comments on AES Selection: AES Round 2 public comment*.
- [ScCaFr03] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V. (2003). *RTP: A Transport Protocol for Real-Time Applications* (No. RFC 3550): Network Working Group, The Internet Engineering Task Force.
- [SeKoJa00] Setia, S., Koussih, S., and Jajodia, S. (2000). *Kronos: A scalable Group Re-Keying Approach for Secure Multicast*. In Proceedings of the IEEE Symposium on Security and Privacy, pp. 215-228.
- [Sh49] Shannon, C. (1949). Communication Theory of Secrecy Systems. *Bell System Technical Journal*, Vol. 28 (4), pp. 656-715.
- [ShMc03] Sherman, A. T., and McGrew, D. A. (2003). Key Establishment in Large Dynamic Groups Using One-Way Function Trees. *IEEE Transactions on Software Engineering*, Vol. 29 (5), pp. 444-458.
- [Sh00] Shirey, R. (2000). *Internet Security Glossary* (No. RFC 2828): Network Working Group, The Internet Engineering Task Force.
- [SiCh97] Sirbu, M., and Chuang, J. (1997). *Distributed authentication in Kerberos using public key cryptography*. In Proceedings of the Network and Distributed System Security, pp. 134-141.
- [SoCh05] Solomon, M. G., and Chapple, M. (2005). *Information Security Illuminated* Sudbury. Jones and Bartlett Publishers, Inc.
- [StStDi88] Steer, D. G., Strawczynski, L., Diffie, W., and Wiener, M. J. (1988). *A Secure Audio Teleconference System*. In Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology, pp. 520-528.

- [StNeSc88] Steiner, J., Neuman, C., and Schiller, J. I. (1988). *Kerberos: An Authentication Service for Open Network Systems*. In Proceedings of the Winter 1988 Usenix Conference, pp. 191-200.
- [StTsWa95] Steiner, M., Tsudik, G., and Waidner, M. (1995). Refinement and extension of encrypted key exchange. *ACM SIGOPS Operating Systems Review*, Vol. 29 (3), pp. 22-30.
- [StTsWa96] Steiner, M., Tsudik, G., and Waidner, M. (1996). *Diffie-Hellman Key Distribution Extended to Group Communication*. In Proceedings of the 3rd ACM Conference on Computer and Communications Security, pp. 31-37.
- [StTsWa98] Steiner, M., Tsudik, G., and Waidner, M. (1998). *CLIQUEs: A New Approach to Group Key Agreement*. In Proceedings of the 18th International Distributed Computing Systems, pp. 380-387.
- [StTsWa00] Steiner, M., Tsudik, G., and Waidner, M. (2000). Key Agreement in Dynamic Peer Groups. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11 (8), pp. 769-780.
- [TaWe06] Talbot, J., and Welsh, D. (2006). *Complexity and Cryptography-An Introduction* New York. Cambridge Univeristy Press.
- [ThDoGl98] Thayer, R., and Glenn, N. D. R. (1998). *IP Security Document Roadmap* (No. RFC 2411): Network Working Group, The Internet Engineering Task Force.
- [TiKh92] Tienari, M., and Khakhar, D. (1992). *Information network and data communication* Espoo, Finland. Amsterdam, Elsevier Science Pub. Co.
- [TiKr07] Tipton, H. F., and Krause, M. (2007). *Information Security Management Handbook* New York, USA. CRC Press, Taylor & Francis Group.
- [Tu07] Tubin, G. (2007). *The Perfect Storm:Man in the Middle Phishing Kits, Weak Authentication and Organized Online Criminals* (No. 021807): Protecting Online Identity, TriCipher.

- [WaSc96] Wagner, D., and Schneier, B. (1996). *Analysis of the SSL 3.0 protocol*. In Proceedings of the 2nd USENIX Workshop on Electronic Commerce, pp. 4-17.
- [WaCaSu99] Waldvogel, M., Caronni, G., Sun, D., Weiler, N., and Plattner, B. (1999). The VersaKey Framework: Versatile Group Key Management. *IEEE Journal on Selected Areas in Communications* Vol. 17 (9), pp. 1614-1631.
- [WaHaAg97] Wallner, D. M., Harder, E. J., and Agee, R. C. (1997). *Key Management for Multicast: Issues and Architectures* (No. RFC 2627): Network Working Group, The Internet Engineering Task Force.
- [WaZhZh06] Wang, X., Zhang, J., Zhang, W., and Khan, M. K. (2006). *Security Improvement on the Timestamp-based Password Authentication Scheme Using Smart Cards*. In Proceedings of the IEEE International Conference on Engineering of Intelligent Systems, pp. 1-3.
- [WaLe05] Wang, Y., and Le, P. D. (2005). Scalable multi-subgroup key management in wireless networks. *International Journal of Computer Science and Network Security*, Vol. 5 (11), pp. 95-106.
- [We05] Webb, A. (2005). Briton arrested in military hacking Computer expert faces extradition. Retrieved 2 Sep, 2008, from http://www.accessmylibrary.com/coms2/summary_0286-17055599_ITM
- [Wh09] Whitehouse, L. (2009). Storage in 2009: Data protection. Retrieved 6 January, 2009, from <http://searchstorage.techtarget.com.au/articles/28252-Storage-in-2-9-Data-protection>
- [Wi90] Wiener, M. J. (1990). Cryptanalysis of Short RSA Secret Exponents. *IEEE Transactions on Information Theory*, Vol. 36 (3), pp. 553-558.
- [WiZu98] Wiener, M. J., and Zuccherato, R. J. (1998). *Faster Attacks on Elliptic Curve Cryptosystems*. In Proceedings of the Selected Areas in Cryptography, pp. 190-200.
- [WoGoLa98] Wong, C. K., Gouda, M., and Lam, S. S. (1998). *Secure Group Communications Using Key Graphs*. In Proceedings of the ACM Special Interest Group on Data Communication, pp. 68-79.

- [WoGoLa00] Wong, C. K., Gouda, M., and Lam, S. S. (2000). Secure Group Communications Using Key Graphs. *IEEE/ACM Transactions on Networking*, Vol. 8 (1), pp. 16-30.
- [WoOrHi02] Woodward, Jr., J. D., Orleans, N. M., and Higgins, P. T. (2002). *Biometrics: Identity Assurance in the Information Age* Berkeley, California, USA. McGraw-Hill Osborne Media.
- [Wu98] Wu, T. (1998). *The Secure Remote Password Protocol*. In Proceedings of the Internet Society Symposium on Network and Distributed System Security, pp. 97–111.
- [WuLe06] Wu, X. P., and Le, P. D. (2006) A Dynamic key Generation Scheme and Symmetric Cryptography. China Patent No. 200710175938.7. S. I. P. Office.
- [WuLeSr08] Wu, X. P., Le, P. D., and Srinivasan, B. (2008). *Dynamic Keys Based Sensitive Information System*. In Proceedings of the 9th International Conference for Young Computer Scientists, pp. 1895-1901.
- [WuLeSr09] Wu, X. P., Le, P. D., and Srinivasan, B. (2009). Security Architecture for Sensitive Information Systems. In *Convergence and Hybrid Information Technologies*. Vienna, Austria: IN-TECH.
- [WuNgLe08a] Wu, X. P., Ngo, H. H., Le, P. D., and Srinivasan, B. (2008). *Novel Authentication Protocol for Sensitive Information Systems Using Dynamic Key Based Group Key Management*. In Proceedings of the International Conference on Convergence and Hybrid Information Technology (ICCIT 2008), pp. 1113-1119.
- [WuNgLe08b] Wu, X. P., Ngo, H. H., Le, P. D., and Srinivasan, B. (2008). *A Novel Group Key Management Scheme for Privacy Protection Sensitive Information Systems*. In Proceedings of the International Conference on Security and Management, pp. 93-99.
- [WuNgLe09] Wu, X. P., Ngo, H. H., Le, P. D., and 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 (3), pp. 57-74.

- [Zh01] Zhou, J. (2001). *Non-repudiation in Electronic Commerce* Norwood, MA, USA. Artech House Publishers.
- [ZiJoCa09] Zimmermann, P., Johnston, A., and Callas, J. (2009). *ZRTP: Media Path Key Agreement for Secure RTP* (No. Draft 13): Network Working Group, The Internet Engineering Task Force.
- [ZoRaMa05] Zou, X., Ramamurthy, B., and Magliveras, S. S. (2005). *Secure Group Communications over Data Networks* New York, USA. Springer Science/Business Media, Inc.
- [Zw97] Zwass, V. (1997). *Foundations of Information Systems* New York. Irwin/McGraw-Hill Companies, Inc.

Publications

Patent

Wu, X. P., and Le, P. D. (2006) A Dynamic Key Generation Scheme and Symmetric Cryptography. China Patent No. 200710175938.7. S. I. P. Office.

Book Chapters

Wu, X. P., Le, P. D., and Srinivasan, B. (2009). Security Architecture for Sensitive Information Systems. **To Appear** for Book: Convergence and Hybrid Information Technologies. Vienna, Austria: IN-TECH.

Journal

Wu, X. P., Ngo, H. H., Le, P. D., and 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 (3), pp. 57-74.

Dandash, O., Wu, X. P., Wang, Y. L., Le, P. D., and Srinivasan, B. (2008). Security Analysis for Internet Banking Models. Special Issue of the International Journal of Computer and Information Science ,Vol. 9(2), pp. 1-10.

Ngo, H. H., Wu, X. P., Le, P. D., and Wilson, C. (2010). Dynamic Key Cryptography and Applications. International Journal of Network Security, Vol. 10 (3), pp. 161-174.

Qi H.M., Chen Z.G. , Wu X.P (2009), A Calculus for end-to-end delay bounds in WMNs by Min-plus Algebra, **To Appear** for the journal of High Technology Letters

Ngo, H. H., Wu, X., Le, P. D., and Srinivasan, B. (2009). An Authentication and Authorization Model for Wireless Network Services. Submitted to IEEE Transactions on Systems, Man, and Cybernetics (SMC).

Conferences

Wu, X. P., Ngo, H. H., Le, P. D., and Srinivasan, B. (2008). A Novel Group Key Management Scheme for Privacy Protection Sensitive Information Systems. In Proceedings of the 2008 International Conference on Security and Management (SAM'08), pp. 93-99.

Wu, X. P., Ngo, H. H., Le, P. D., and Srinivasan, B. (2008). Novel Authentication Protocol for Sensitive Information Systems Using Dynamic Key Based Group Key Management. In Proceedings of the 2008 International Conference on Convergence and Hybrid Information Technology (ICCIT 2008), pp. 1113-1119.

Wu, X. P., Le, P. D., and Srinivasan, B. (2008). Dynamic Keys Based Sensitive Information System. In Proceedings of the 9th International Conference for Young Computer Scientists, pp. 1895-1901.

Wu, X. P., Ngo, H. H., Le, P. D., and Srinivasan, B. (2008). Design & Implementation of a Secure Sensitive Information System for Wireless Mobile Devices. In Proceedings of the Australasian Telecommunication Networks and Applications Conference, pp. 45-50.

Ngo, H. H., Wu, X. P., and Le, P. D. (2008). A Group Authentication Model for Wireless Network Services based on Group Key Management. In Proceedings of the International Conference on Enterprise Information Systems (ICEI08), pp. 182-188.

- Ngo, H. H., Wu, X. P., Le, P. D., and Wilson, C. (2008). A Method for Authentication Services in Wireless Networks. In Proceedings of the 14th Americas Conference on Information Systems (AMCIS'08), pp. 1-9.
- Ngo, H. H., Wu, X. P., Le, P. D., and Wilson, C. (2009). Package-Role Based Authorization Control Model for Wireless Network Services. In Proceedings of the 4th International Conference on Availability, Reliability and Security, pp. 475-480.
- Chen, Y. J., Wang, Y. L., Wu, X. P., and Le, P. D. (2006). The Design of Cluster-based Group Key Management System in Wireless Networks. International Conference on Communication Technology (ICCT '06) pp. 1-4.