# SMS – a Security Management System with LDAP

# 整合 LDAP 的安控管理系統


蔡宗易 Tsung-Yi Tsai
交通大學資訊工程學系
tytsai@csie.nctu.edu.tw

王嘉宏 Jia-Horng Wang
中華電信
wch96823@ms11.hinet.net

蔡文能 Wen-Nung Tsai
交通大學資訊工程學系
tsaiwn@csie.nctu.edu.tw



### *Abstract*

*In a heterogeneous environment, different computer hosts may not have the same machine type and may also require different login procedures. When a user uses a certain host, he has to come up with the right password and login with the right procedure. However, this information might be lost and result in the difficulty of management. In this paper, we investigated two representative authentication services and proposed a similar scheme integrated with LDAP, which we named it as "Security Management System (SMS)". In this system, the user logins merely once by using a Smart Card to provide Single Sign On authentication, and then the system manages the identities with the help of directory service.*



***Keywords:*** *Kerberos, SESAME, Electronic certificate, Directory Service*



### 中文摘要

在一個異質性系統環境中,所有主機的系統類型和登入方式不盡相同， 使用者要登入不同的主機,必須使用不同組的帳號密碼,對使用者而言,記憶這麼多組不同的帳號密碼,這些資訊很有可能被遺忘。而且無法確認每一部主機的帳號實際擁有者是誰？"使用者安全管理系統"這個解決方案就是要減少記憶密碼的困擾,以及確認每一個帳號之實際擁有者。

在本篇論文中，我們研究一些重要的認證服務,並且提出一套類似的系統,利用電子憑證與目錄服務之技術,當使用者以智慧卡登入系統時,以簽章方式確認使用者身份，而系統採用"目錄服務"來管理所有使用者,使得帳號管理更為明確,避免不明帳號出現於系統內。



**關鍵詞**：Kerberos, SESAME, 電子憑證，目錄服務


## 1. Introduction

For the past few decades, ids and passwords are used to verify the identity of a user. However, we now need a more comprehensive authentication system to address several issues, including security, single sign on and heterogeneous issues. An integrated authentication system should have the following characteristics: (1) Authentication (2) Confidentiality (3) Information Integrity (4) Authorization (5) Access Control and (6) Auditing. The Kerberos [5] system, which is part of the Athena Project started by MIT from 1983, is the best-known authentication system for a distributed system with high industry and research acceptance. However, it has no mechanisms to provide authorization service. The SESAME architecture [14][15], on the other hand, is based upon the Kerberos with several improvements added, such as logon, which uses digital signatures to prevent off line dictionary attacks; and cross-realm authentication, which uses PKI and Role-Based Access Control.

However, password-based authentication still faces several problems in the distributed heterogeneous environment. The IT organizations must manage a lot of user information, such as user name, password and access-level authorizations. This information is often stored in several different locations; thus, any change in the user status or user access rights requires modifications on multiple locations. This situation will complicate the management of user identity and result in more inaccuracy, redundancy and inconsistency. According to the research of META Group [9], a consolidate user management system could result in increases in consistency by 44%, accuracy by 36% and actual security by 33%.

In this paper, we proposed a user identity management system, the "SMS", to address the following issues:

(1) Reduce the burden of memorizing pairs of id and password.
(2) Provide integrity check between user account and user identification.
(3) Provide easy-to-use, web-based management interface.

(4) Build a small certification authority (CA) to manage certificates of users and provide mutual authentication between users and computer systems.
(5) Store the user's private information in a Smart Card, including the public key, private key and the certificate.
(6) Log every operation in an audit system to provide post disaster tracking.

This paper is organized as follows. In section 2, we will give a brief overview of Kerberos and SESAME. Section 3 will describe in detail the architecture of our proposed system. In section 4, we make a complete evaluation of our implemented system, and compare the pros and cons among Kerberos, SESAME and SMS. Section 5 presents the conclusion and future works.

## 2. Related Works

The most representative works about the area of authentication might be the Kerberos system and the SESAME system. In this section, we introduce these two systems to bring in the concept of how authentication can work securely, as well as some related terminologies.

### 2.1. Kerberos Authentication Protocol

Kerberos is part of the Athena Project leaded by MIT [6]. The Kerberos Version 5 is now a standard document in RFC1510 [5]. The purpose of the Athena project is to provide an easy access environment to computing resource for all students in MIT. The Kerberos sub-program was designed to address the issue of impersonation problems for those untrustful public workstations using the ticket-based two-way authentication.

Kerberos provides the following features:
(1) **Trusted Third-Party**. Kerberos separates out the responsibility of authentication to a dedicate authentication server (AS).
(2) **Bi-directional Authentication**. Not only the request side has to be authenticated, but also the service part has to identify itself.
(3) **Conventional Cryptosystem**. Kerberos uses traditional encryption algorithms to encrypt and decrypt transmitted data to provide confidentiality and decrease system overhead.
(4) **Cross Realm**. Users can access resource cross the realm resided in.

As shown in Figure 1, there are four major components in the Kerberos system model. The Authentication Server (AS) is responsible for checking the identification of each client, which is intentional to access the resource on the network. Every authenticated client will acquire a Ticket Granting Ticket (TGT) from AS to demonstrate its validity of identification. As long as the TGT held is not expired, the client can use it to request a passport to the server that provides desired service. This work is done by the Ticket Granting Server (TGS). After the client possesses the Service Grating Ticket (SGT) from TGS, it can use it to do mutual authentication with server and request the service finally. In this way, clients do not need to type in his/her password every time he/she wants to access the service. This can dramatically decrease the possibility of the password being stolen by attackers eavesdropping on the Internet.

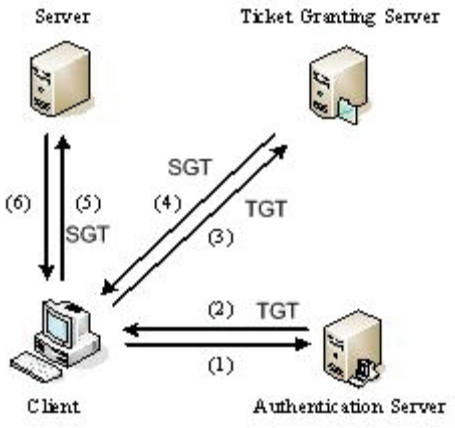

Figure 1 System Model of Kerberos.

When a user resided in one local realm (say, RealmA) wants to access the resource on the other realm (say, RealmB), the local TGS will issue a passport to the remote TGS for this user. With this ticket to the TGS on RealmB, the user can request the service following the same steps of (3) ~ (4) as if he/she is a trusted user in RealmB. However, in this peer-to-peer architecture, there is a need to maintain an order of $n^2$ key pairs when the number of related realms is n.

### 2.2. Kerberos extension: PKINIT, PKCROSS

In order to take the advantages of PKI, there are still two ongoing Kerberos extensions proposed by IETF, the PKINIT (Public Key cryptography for INITial authentication in kerberos) [8] and PKCROSS (Public Key cryptography for CROSS-realm authentication in kerberos). With PKINIT, users have the alternative to use public key certificate and digital signature to identify themselves and obtain an ordinary TGT as usual. With PKCROSS, Kerberos can benefit from PKI to transmit cross-realm session key between local realm TGS and remote realm TGS without the burden of maintaining $O(n^2)$ peer-to-peer relationships.

PKINIT is an Internet Draft that the IETF (Internet Engineering Task Force) is still formulating [1][8]. This specification formulates a method that enables the user to use the Public Key Cryptography during the initial authentication. The method is shown as the following: When a user uses the

Kerberos system, he or she may choose to use between the Public Key Cryptography and the Private Key Cryptography during the initial authentication. If the user uses the Public Key Cryptography, the system will then insert the user's electronic certificate and digital signature into the pre-authentication data fields. After authenticated, the user will be given a ticket-granting ticket in order to proceed in further steps. The ticket-granting ticket uses either the Diffie-Hellman derived key or the user's public key for encryption. Also, the feedback message must be signed by the authentication server, that is, it must use the public key of the authentication server.

PKCROSS is abbreviated for Public Key cryptography for CROSS-realm authentication in kerberos. When a user is being cross-realm authenticated, the PKCROSS formulates a method that enables the user to use the Public Key Cryptography: the user can use the authentication server's public key or cross-realm private key for encryption. The PKCROSS specification defines how these messages are to be transmitted. Without the Public Key Cryptography, a manager will then have to pre-establish an information interchange key for each connection between all of the realms, resulting in a need of $2\times\binom{N}{2} = N\times(N-1)$ keys. The manager could use a hierarchy structure to reduce the amount of pre-establish keys, but will cause an unwanted net flow increase. To diminish these management loads, the PKCROSS uses the convenience of the Public Key Cryptography's basic structure. Therefore, the manager does not have to deal with all the keys at once, but instead, establish a shared key only when a user requires for cross-realm authentication. This method does not affect the users, for only the authentication server is modified while the users' operating process remains the same. This method implements the Public Key Cryptography's distributed confidential management, and also retains the realm's confidential managing characteristics. The basic scenario of PKCROSS is shown in Figure 2.

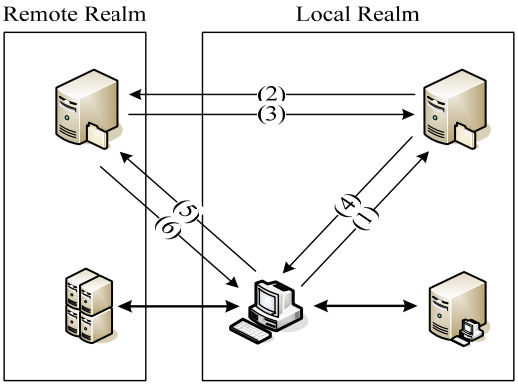


Figure 2 PKCROSS scenario.

## 2.3. SESAME Authentication Protocol

SESAME (Secure European System for Applications in a Multivendor Environment) is an authentication system designed by the ECMA [14] (European Computer Manufacturers Association). The original motivation of this project was to demonstrate the possibility and implementation result of ECMA's works based on ISO security architecture – ISO-749802. This project was then under the auspices of European Commission RACE program and released its version V2 and V3 in 1994 and 1995 respectively. The latest version is SESAME V4.

SESAME is base on Kerberos, so it has many similarities compared with Kerberos. However, there are still several features that SESAME focuses on to address the limitations of Kerberos [12] [13]. First, SESAME adopts the asymmetric cryptography and introduces PKI. Therefore, it can leverage the advantages provided by public key cryptography. Second, SESAME is designed to be compatible with Kerberos. Third, SESAME has both identity-based and role-based access controls to provide a more delicate granularity of access control management [10]. Forth, SESAME offers a platform-independent library, which is called GSSAPI (Generic Security Services Application Programming Interface). With GSSAPI, SESAME provides the application programmer with a uniform library of functions that implement security in a client-server scenario without knowing the details of how every security function is actually implemented.

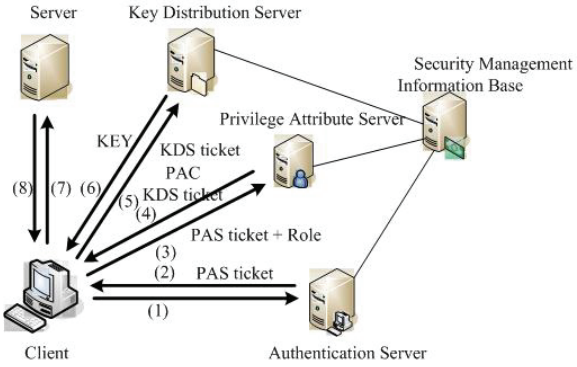


Figure 3 System model of SESAME.

As shown in Figure 3, there are six major components in the SESAME system model.

(1) **Security Management Information Base** (SMIB). All security information used by SESAME components will be stored in the SMIB, including user identity, user information, authorization information and key information.

(2) **Authentication Server** (AS). The AS is responsible to verify the identity of request user as the AS does in the Kerberos system, what is

different is it returns a PAS (Privilege Attribute Server) ticket to the authenticated user.

(3) **Privilege Attribute Server** (PAS). In order to provide a more delicate access control, the PAS will use the information stored in the SMIB and desired role from requesting user to generate a PAC (Privilege Attribute Certificate) for this user. The most important filed of the PAC is the privileges information of possessed user, including access identity, role attribute, primary group and secondary group. All applications in the SESAME system can use a PAC to determine the access control policy with more privilege information instead of identity only. Besides the PAC, the PAS will issue a KDS (Key Distribution Server) ticket to user for accessing the service of the KDS.

(4) **Key Distribution Server** (KDS). Before user can access resource on each server, he/she has to obtain the key information of target server for future communication. The KDS is responsible for key management of all target servers.

(5) **Initiator.** Either a user or a machine can request for service and start the sequence from authentication to service granting. In this client-server model, we view the initiator as a client.

(6) **Target Server**. When a user has passed through authentication step and obtained the entire requisites, he can send the service ticket and the PAC to the target server that provides desired resource. On each target, there is a special piece of code, which is called the PAC Validation Facility (PAF) that will perform the necessary checks on the PAC.

## 3. SMS System Architecture

In this paper, we proposed a "Secure Management System - SMS" to address several issues that both Kerberos and SESAME do not concern. First, we integrate a smart card to store the key information and certificate [3][1][16]. This can omit our key and password transmitting on the network, and reduce the possibility of the password being stolen. Second, we introduce LDAP to assist in the management of user information. LDAP is a lightweight directory access protocol compared with X.500. It is designed to provide a request-and-reply model between client and backend information storage. It assumes information is seldom modified once been set up in storage. Based on this assumption, it can provide fast access of information through it. Third, we design a centralized monitoring server to send manage commands to each managed servers and report exception warning from those monitored agents. In this section, we are going to give a detail description of our proposed management model, including components of this system and the interaction between them.

### 3.1. SMS System Model

As shown in Figure 4, there are eight major components in the SMS's system model. In this model, we store every manager's identity and authority in the **LDAP server**. When certain manager want to login to the **Management Server**, he/she has to obtain his/her own digital certificate from super administrator signed by the **Certificate Authority**. With valid certificate, each manager can login to management server; issue commands to **managed servers** or examine warning messages from those servers. Following are detail descriptions of each component.

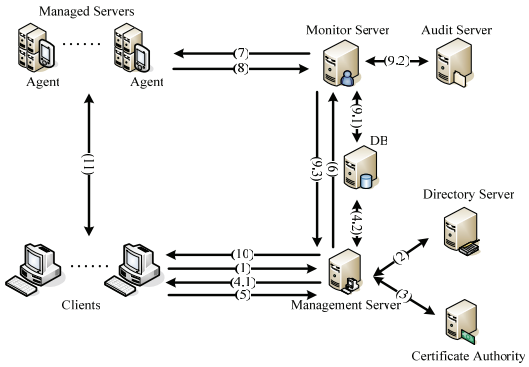


Figure 4 System Model of SMS

1. The **Certificate Authority Server** (CA): The CA is used to sign digital certificate for each manager in this system.
2. The **Lightweight Directory Access Protocol Server** (LDAP): The LDAP is used to store all basic information and authorities of users in this system. It provides a fast access interface in request-and-reply model.
3. The **Management Server** (MaS): The MaS is the major component in the SMS system. With this security gateway, managers can make their own commands or supervise account information of all managed servers. When there are changes in the LDAP system, the MaS server can issue update commands automatically to agents on related servers for accounting consistency.
4. The **Monitor Server** (MoS): It plays an intermediary role between the MaS and other components in this system. On one side, it will receive commands issued from managers on the MaS and forward them to remote agents for execution. On the other side, it will receive unusual account warnings from managed servers and try to compare account status stored both on the managed server and the database, including ids and encrypted passwords. Once there are any inconsistencies, the MoS will ask related agents to do recovery and report this incident to the audit server for future inspection.

5. The **Audit Server** (AS): An independent auditing server that stores incidents happened in this system.
6. The **Database Sever** (DB): It is used to store manager information used in this system, including id, encrypted password and those servers managed by this manager. The MaS and the MoS servers will get information from the DB to do validation.
7. The **Client** (C): Each manager is considered as a client.
8. The **Agent**: In order to provide a uniform interface to the system user, we design a cross-platform agent program installed on each managed server to execute management commands and report account status to our system.

### 3.2. SMS System Flow

In this section, we describe how the system works as the steps shown in Figure 4.
1. **C→MaS**: When client logins to system in the first time, he has to request for permission to use this system. To do so, the client has to provide the desired login account and usage purpose to MaS.
2. **MaS → (LDAP || CA)**: When MaS receiving request from the client, it will ask for validation of this client's identity to LDAP server. If result is positive, the super administrator of this system can used his certificate signed by CA to produce a unique key for this client. All information (User information and related unique key) will be stored in the DB. From now on, this client can use its account, password and unique key to login to the SMS system.
3. **MaS → MoS**: When the client logs in the system with the correct id, password and key, it can register to manage certain server. Once been approved by the super administrator for this registration, it can issue commands to the MoS.
4. **MoS → Agent**: When the MoS Server receives commands from the MaS, it will forward this command to the remote agent via secure channel. The agent program will turn in execute commands for the manager.
5. **Agent → MoS**: After execution of commands, the agent program will return both execution results and status of the managed server to the MoS server via the secure channel. If the managed server changes its account information with unauthorized request, the agent program has to report such exception to the MoS server. Once the MoS server receives this warning, it will compare the account information between unusual servers and the DB to do possible recovery action until both are in consistent state.
6. **MoS → MaS**: The MoS will in turn send back the execution result to the MaS, and report unusual incidents to the audit server.
7. **MaS → C**: Finally, the MaS will show result in web page format to client.

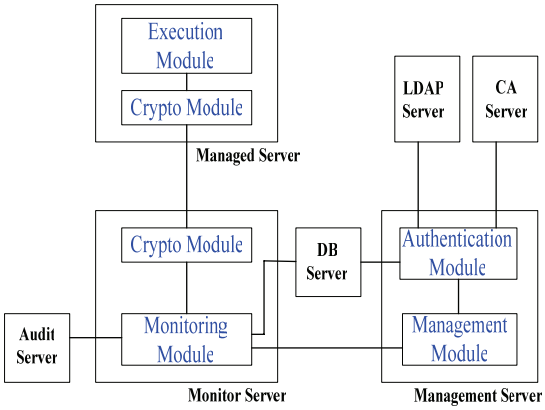


Figure 5 Architecture of SMS system

### 3.3. SMS Module Description

To fulfill the system requirements of the SMS's operation model, we implement it in five software modules as shown in Figure 5. Each module takes responsible for certain tasks and each system component may contain several modules to make up corresponding functionality.

1. The **Authentication Module**: When clients apply their registration to the MaS server in the very first time, the authentication module will first check identities stored in the LDAP server against supplied ones. Afterwards, it will use a hashing algorithm to generate a random value with its digital certificate if the identification is valid. This random hash value will be used as one key to validate the user identity in future login.
2. The **Management Module**: This module is used to provide a web-based configuration interface for authenticated users. By using this interface, managers can send commands to monitoring module to manage those servers that they are responsible to.
3. The **Monitoring Module**: This module is resident in the MoS server to receive commands issued by managers. It will forward those commands to execution module faithfully and return back execution result to ensure each command is done completely as required. In the meanwhile, when the system information of managed server is changed illegally, the monitoring module will report this event to management module for further processing.
4. The **Crypto Module**: In order to protect transmission data, both the monitoring module and execution module will use crypto module to build up a secure communication channel. Either the command or execution result will be encrypted and decrypted at both nodes to provide confidentiality and integrity.
5. The **Execution Module**: It is used to execute the received commands sent by managers. Moreover, it plays the role of monitoring the system information of each managed servers in periodic manner. Any

illegal modification should be reported to the monitoring module immediately to assure the integrity and security of system information.

## 4. Function Evaluation and Comparison

In this section, we first verity that our proposed system is indeed satisfied several security requirements. Then we will give a comparison between some well-known security systems and ours with concern of functionalities.

### 4.1. Function Evaluation

In our proposed system, it satisfied five security requirements:

(1) **Authentication**: In this system, it provides two-steps authentication. When the client wants to setup connection to this system, he/she has to apply a registration request to the super administrator through the management interface. In this step, we can first check the identification of this user through the LDAP server. When this user passes the check, we can do second step inspection by the super administrator in man-made manner. This kind of inspection procedure should be careful to make it impossible for malicious registration. After passing these two authentication steps, the client can obtain a legal account and connect to the management server.
The super administrator will have a certificate signed by a CA for management purpose. This certificate is stored in the smart card. It contains several components to represent unforgible identity, including the public-key and the signature. The super administrator can issue another digital certificate to applied users using this signed certificate. With the assumption that users do not leak out their certificates, we can verify the user's identity with higher accuracy than traditional password-based authentication, since any forged certificated will fail at validation.

(2) **Confidentiality**: In our proposed system, we should protect the transmission data among users and system components. To achieve this requirement, we can use traditional symmetric encryption system to encode and decode the data. However, it will be exposed under attacked if the session key is simple even disclosed. In our proposed system, it is secure in login step because we use certificates in cooperate with passwords to provide confidentiality to user identity. It is also secure in communication phase since every data is transmitted in secure channel based on SSL confidentiality.

(3) **Integrity**: To defense against forged data, we store all the important information of the managed server with a message authentication code (MAC). The agent program will periodically compute the MAC code and compare it with the stored ones to ensure data integrity. Besides the integrity of stored data, we provide integrity to transmission data by computing the MAC code and encoding both message and the MAC code before sending on network.

(4) **Access Control**: When client wants to connect to our system, he has to register first and then this user would be added in the access list. All the authentication and authorization information are stored in the DB. Both the MaS and the MoS components will use them to verity the user identity and legality of issued commands.

(5) **Auditing**: We will record the login method, login time and remote IP address when user login to our system. If the user issue commands to modify account configuration, the agent program will report this behavior. If the agent program finds that this modification request is illegal, it will retrieve original accounting data in secure channel and restore them.

In addition to security requirements, SMS also has another two advantages compared with other security systems:

(1) **Customization**: Our proposed system is especially customized to fulfill the requirements of enterprise. Compared with other authentication system, other systems are mostly designed and implemented without satisfaction to enterprise specific requirements.

(2) **LDAP Integration**: It will be a great help to the management information when using the LDAP server. The LDAP server provides an efficient request and reply usage model. Moreover, this simple interface makes it easy to integrate enterprise resource. Based on the benefits of the LDAP, our system provides a well-defined interface to connect to other existing working systems in enterprise. This makes it a really useful system.

### 4.2 Function Comparison

Based on the functionality analysis discussed above and related references in [11][12], we put a comparison between SMS and existing security systems, as shown in. Table 1.

| System / Function | Kerberos | SESAME | SMS |
|---|---|---|---|
| Authentication | Y | Y | Y |
| Confidentiality | Y | Y | Y |
| Integrity | Y | Y | Y |
| Audit | - | Y | Y |
| Access Control | - | Y | Y |
| Customization | - | - | Y |
| Integrate LDAP | - | - | Y |

Table 1 Functional comparison result

In this table, each column represents one security system which is Kerberos, SESAME and SMS respectively. In the followings, we list seven functionalities to examine each system, which are authentication, confidentiality, integrity, auditing, access control, customization and LDAP integration. In each entry, "Y" means the corresponding system has that functionality, and "-" means otherwise.

According to this result, it shows that every security system would have three basic functionalities – the authentication to provide identification, the confidentiality to protect transmitted data, and the integrity to ensure correctness. Therefore, our proposed SMS system follows such two great security systems and put these as our basic requirements. To supplement the lack of other functionalities of Kerberos, we enhance the audit and access control parts on SMS. Finally, we integrated the LDAP and build up a well-defined interface to link up with existing enterprise application systems. The most advantage benefited from the LDAP server is the instantaneity reaction of personnel modification. Once the personnel department has changed the configuration of account, this event will be adopted immediately on management server to reduce the risk of authority abuse.

## 5. Conclusion

In our proposed Security Management System (SMS), we use the LDAP server to decrease the complexity of identity management. We eliminate the burden of administrators to memorize pairs of passwords. Instead, we use PKI and put certificate in smart cards for authentication, reducing the risk of identification counterfeit of administrators. We store all user information in a database, including the relationship between managers and the responsible servers. In this way, there is no need to worry about account inconsistency in every server. The monitoring server will keep an eye on any unusual modification of account and report to users. In our system, we setup a web-based configuration interface to let managers control their servers through this unified interface. The management server will take over the jobs faithfully and return execution result to user in web pages.

## References


[1] Brian Tung, Clifford Neuman, "Public Key Cryptography for Initial Authentication in Kerberos," Internet-Draft, 2004 http://www1.ietf.org/proceedings_new/04nov/IDs/draft-ietf-cat-kerberos-pk-init-21.txt

[2] C. C. Chang and S. J. Hwang, "Using Smart Cards to Authentication Remote Passwords", Computer Mathematics with Applications, Vol.26, No.7, pp. 19 ~ 27, 1993.

[3] C. K. Chan and L. M. Cheng, "Cryptanalysis of a Remote User Authentication Scheme Using Smart Cards", *IEEE Transactions on Consumer Electronics*, Vol.46, No.4, pp. 992 ~ 993, Nov 2000.

[4] Gerald Carter, *LDAP System Administration*, O'Reilly & Associates, Inc., 2003.

[5] J. Kohl, C. Neuman, "The Kerberos Network Authentication Service (V5)." *Request For Comments: 1510, Internet Engineering Task Force*, September 1993.

[6] Josina M, Arfman, "Project Athena: supporting distributed computing at MIT", *IBM Systems Journal*, Sep 1992.

[7] J. T. Kohl, B. C. Neuman, T. Y. T'so, "The Evolution of the Kerberos Authentication System." *In Distributed Open System, pages 78-94. IEEE Computer Society Press*, September 1994.

[8] L. Zhu, K. Jaganathan, etc., "OCSP Support for PKINIT," Internet-Draft, 2005. http://www.ietf.org/internet-drafts/draft-ietf-krb-wg-ocsp-for-pkinit-06.txt

[9] META Group, "The Value of Identity Management: How securing identity management provides value to enterprise." August 2002.

[10] M. Vandenwauver, R. Govaerts, J. Vandewalle, "Role Based Access Control in Distributed Systems", *Communications and Multimedia Security*, volume 3, pp. 169 ~ 177, 1997.

[11] M. Vandenwauver, R. Govaerts, and J. Vandewalle. "Overview of Authentication Protocols." In *Proceedings of 31st Annual IEEE Carnahan Conference on Security Technology*, pages 108-113, 1997.

[12] P. Ashley, "Authentication For a Large Heterogeneous Multi-Domain System", *Australian Unix and Open Systems Group National Conference*, pp. 159 ~ 169, 1997.

[13] P. V. McMahon, "SESAME V2 Public Key and Authorization Extensions to Kerberos", *in Proceedings of the Symposium on Network and Distributed system Security*, IEEE Computer Society Press, pp. 114 ~ 131, Feb 1995.

[14] SESAME. http://www.esat.kuleuven.ac.be/cosic/sesame/

[15] T. Parker and D. Pinkas. "SESAME V4-OVERVIEW." *SESAME*, December 1995.

[16] W. H. Yang, S. P. Shieh, "Password authentication schemes with smart cards", Computer and Security, Vol. 18, No.8, pp. 727 ~ 733, 1999.