A Consolidated Authentication Model in Cloud Computing Environments

Jaejung Kim* and Seng-phil Hong**

**Department of Computer Science, Sungshin University, Seongbuk-gu, Seoul, Korea*
jajukim@hotmail.com
***Department of Computer Science, Sungshin University, Seongbuk-gu, Seoul, Korea*
philhong@sungshin.ac.kr

Abstract

Due to increasing needs of Internet access through smart phones and smart pads, it is essential to have service provider systems, which allows to access services through a variety of devices. In particular, this system is required to protect credential and personal information saved in each device, is need a more efficient and secure consolidated authentication model (CAM) in order to authenticate a user and devices.

This paper analyzes the current user authentication model for both user and device authentication and securely available credential (SACRED) standards. Furthermore, it is also our intention to design N-screen based consolidated user authentication model that meets framework and protocol requirement of credentials and privacy protection requirements in a cloud computing environments.

Keywords: *User Authentication, Privacy, PKI, Credentials, N-screen*

1. Introduction

At each site, to deracinate a disclosure, abuse, and misuse of personal information, Privacy Act was enacted and enforced. [1] In addition, although accredited certificate is the most commonly used method for user authentication [2] in various e-governments and online based financial sector, the risk of using it has been coming on the rise due to spreading of malicious program such as virus, spyware or malware. In order to solve this problem, there has been active discussion over ways to come up with enhanced safety measures. In this regard, it is necessary to develop new technologies which enable the accredited certificate to utilize in various smart devices and cloud computing environment. [3]

In this paper, it is our purpose to suggest a safe and convenient user authentication model that mobile device users can effortlessly use credentials in cloud computing environments. After this brief introduction, the reminder of this paper is organized as follows: section 2 discusses the security issues in clouds and M2M (Machine to Machine) environments, the analysis of current user authentication model, and securely available credentials. Then we propose consolidated user authentication with framework architecture and protocol framework is shown in section 3. Section 4 describes overall prototyping of proposed model. Section 5 presents the comparison and verification of our architecture. Finally we conclude the paper and suggest the future research work in section 6.

2. Related Work

2.1. Security issue for clouds and M2M

Identity and access management one of key security and privacy issues in cloud computing have become increasingly an area of concern for organizations.[4] The identity proofing and authentication aspects of identity management entail the use, maintenance, and protection from users. Preventing unauthorized access to information resources in the cloud is also a major consideration. One recurring issue is that the organizational identification and authentication framework may not naturally extend into a public cloud and extending or changing the existing framework to support cloud services may prove difficult. [5] [6]

Security issues such as the standardization of smart device authentication, a common security monitoring and control services, human-centric privacy, the security of M2M communication between the various smart devices are occurred in the communication model of society based on smart devices. There are the absence of device authentication standard as the various types of new smart devices emerge and the absence of a standardized security technology for secure mobile commerce and financial services such as online banking in open market environment. To remedy this, the standardization of open security platform technology for the effective management of a variety of security technologies using the communication environment, smart devices, and services is needed. The consolidated user authentication between smart devices, data encryption/decryption, etc. is required at the M2M communication between smart devices. To this end, the establishment of total security infrastructure for smart devices is needed in all areas of the smart devices, networks, and services. [7]

2.2. Current user authentication model

The current user authentication model as below is a model which enables user mobile devices to perform user authentication by using user certificate stored in the smart devices. This model does not need to install additional software such as plug-in or ActiveX in order to perform user authentication. It simply requires web browser such as Internet Explorer, Opera, Safari, Firefox, Chrome etc. Furthermore, this model is designed not to reveal the sensitive data such as digital signature information to Auth Server through end-to-end encryption which ensures its secure delivery between service provider and smart device. [8]

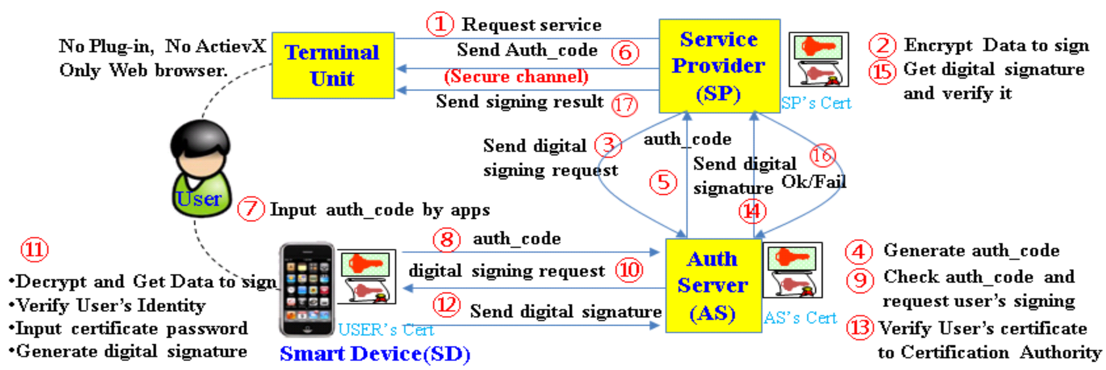

Figure 1. Current user authentication model

In this user authentication model, credentials are stored in a specific smart device to generate a digital signature. However, since cloud computing environments should support a variety of devices, a more efficient and secure user authentication model is required. In addition to that, tight security system needs to be equipped in dealing with the life cycle of personal information which ranges from collection, storage, use, transfer to disposal.

2.3. Securely available credentials

The SACRED Working Group is working on the standardization of a set of protocol for securely transferring credentials among devices. The international standards of SACRED Working group consist of RFC 3157(Requirements) [9], RFC 3760 (Credential Server Framework) [10], and RFC 3767(Protocol) [11].

Problems and limitations of existing SACRED standards are as below.

First, although it defines the framework and protocol requirements with respect to securely available credential, the existing SACRED standards does not provide a detail implementation guideline. Second, SACARE defines the upload and download protocol for credentials, but does not define a protocol to create a proxy signature from Signing Server using uploaded credential by Client.

To solve these problems, let us redefine a credential framework that meets the framework and protocol requirements of SACRED and design a credential protocol based on ASN.1. In addition, we would like to define a protocol of credential roaming and proxy signature which fits in cloud computing environments.

3. Consolidated Authentication Model (CAM)

3.1. Overview

The CAM consists of consolidated authentication mechanism and policy compliance mechanism. The consolidated authentication mechanism is guaranteed to the use's consolidated authentication using security technology. The policy compliance mechanism is supported the systemically policy not only managing but also controlling the system during interoperation process.

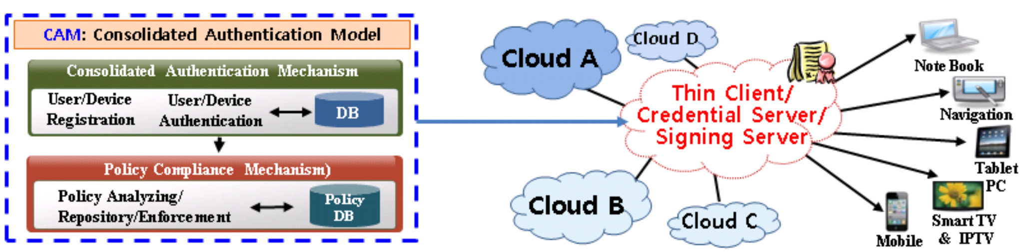

Figure 2. CAM Architecture

3.2. Framework architecture

Client, Credential Server, Signing Server in framework perform the following roles. A Client uploads or downloads credentials from Credential Server through a variety of devices such as PC, smart pad, and smart phone and generates the digital signature from Signing Server. The Credential Server (CS) downloads secure credentials and uploads them from the client. The Credential Store is the repository for secured credentials. The Signing Server (SS) creates a digital signature by the Client's request.

3.3. Secure credentials design

Credentials are information that can be used to establish the identity of an entity, or help that entity communicate securely. Credentials include such things as private keys, trust roots, tickets, or the private part of a Personal Security Environment (PSE) [RFC2510]. [9] Several standardized formats for the representation of credentials exist e.g., PKCS#12[12], PKCS#15[13]. Secure Credentials is a set of one or more credentials that have been cryptographically secured, e.g., encrypted/MACed with a passkey [11].

3.4. Protocol framework

Consolidated authentication mechanism consists of account management module (AMM), credential roaming module (CRM), and proxy signature module (PSM).

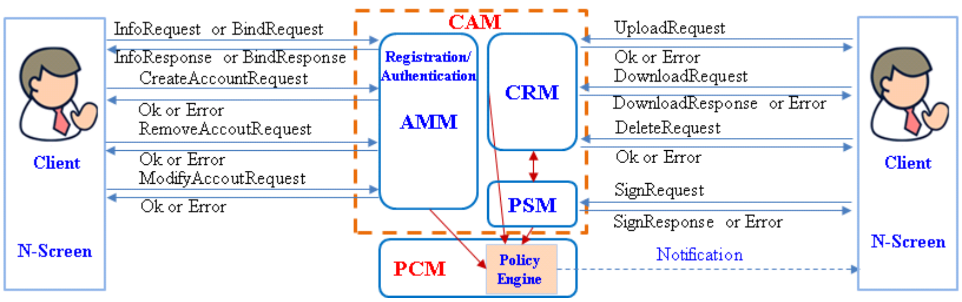

Figure 3. Protocol Framework

Notations and abbreviation for protocol design is as follows:

Table 1. Notations and Abbreviation

Symbol: Description	Symbol: Description	Symbol: Description
SN: Serial Number	TIME: Signed Time	KDF2: Key Derivation Function 2[14]
SD: Signed Data	PV: Password Verifier	RC, RS: Random Number
ID: Identification Number	C: Credential	SC: Secure Credential, K: Key for SC
H(): Hash Function	E(): Encryption	S(): Generate Signature
=?: Compare with	D(): Decryption	V(): Verify Signature

3.4.1. Initialization and key sharing operations

1) BindRequest/BindResponse Protocol

If Client has his/her own digital certificate, he/she creates a digital signature using their private key and passes it to credential server.

① *Client:* $I=H(ID)$, $N=H(SN)$, $SD=S_{client\_key}(I|N|TIME)$

② *Client → CS: BindRequest(SD, Client_Cert)*

③ *CS:* $I|N|TIME= V_{client\_cert}(SD)$, $ERS=E_{client\_cert}(RS)$

④ *CS → Client: BindResponse(ERS, CS_cert)* ⑤ *Client:* $RS=D_{client\_key}(ERS)$

2) InfoReqeust/InfoResponse Protocol

If Client doesn't have the certificate, Credential Server sends a session key (RS) after the server creates secure channel through SSL/TLS [15] or DH key exchange [16].

① Client → CS: InfoRequest *② CS → Client: InfoResponse(RS, CS_cert)*

3.4.2. Account management operations

1) Create Account Protocol

When Client creates user account, Credential Server registers hash value which contains both client's unique ID such as Resident Registration number and unique device information such as serial number or MAC address that is used by the Client PC, smart pad, smart phone, etc.

① Client → CS: BindRequest(SD, Client_Cert) ② CS → Client: BindResponse(ERS, CS_cert)

③ Client: <u>ERC</u>=E_{RS}(RC), I=H(ID), N=H(SN), K=KDF2(I, PW), PV=H(H(ID, K)),
HI=H(I|PV|RS|RC|N), <u>EI</u>=E_{RC}(I|PV|HI|N)

④ Client → CS: CreateAccountRequest(ERC, EI)

⑤ CS:RC=D_{RS}(ERC)),I|PV|HI|SN=D_{RC}(EI),HI'=H(I|PV|RS|RC|SN,I|PV|HI|SN=?I|PV|HI'|SN

⑥ CS → Client: CreateAccountResponse(Ok or Error)

2) Modify Account Protocol and Remove Account Protocol

Client can register new device information or modify the registered device information. If the Client no longer uses the account, a registered account can be removed.

① Client → CS: BindRequest(SD, Client_Cert) ② CS → Client: BindResponse(ERS, CS_cert)

③ Client: <u>ERC</u>=E_{RS}(RC), I=H(ID), N=H(SN),K=KDF2(I, PW), PV'=H(ID, K),
HI=H(I|PV'|RS|RC|N), <u>EI</u>=E_{RC}(I|PV'|HI|N)

④ Client → CS: {Modify,Remove}AccountRequest(ERC,EI)

⑤ CS: RC=D_{RS}(ERC)), I|PV'|HI|N=D_{RC}(EI), HI'=H(I|PV'|RS|RC|N),
I|H(PV')|HI|N=?I|PV|HI'|N

⑥ CS → Client: {Modify, Remove}AccountResponse (Ok or Error)

3.4.3. Credential roaming operations

1) Credential Upload Protocol

The registration process that Client uploads credentials to Credential Server is as below.

① Client → CS: BindRequest(SD, Client_Cert) ② CS → Client: BindResponse(ERS, CS_cert)

③ Client: <u>ERC</u>=E_{RS}(RC),I=H(ID), N=H(SN), K=KDF2(I, PW), PV=H(H(ID, K)),
SC=E_K(C),HI=H(I|PV|SC|RS|RC|N), <u>EI</u>=E_{RC}(I|PV|SC|HI|N)

④ Client → CS: UploadRequest(ERC, EI)

⑤ CS: RC=D_{RS}(ERC), I|PV|SC|HI|N=D_{RC}(EI), HI'=H(I|PV|SC|RS|RC|N),
I|PV|SC|HI|N=?I|PV|SC|HI'|N ⑥ CS → Client: UploadResponse(Ok or Error)

2) Credential Download Protocol from Credential Server

In order to use the credential in a variety of environments, the download procedure of credentials from the credential server is as follows.

① Client → CS: InfoRequest ② CS → Client: InfoResponse(RS, CS_cert)

③ Client: <u>ERC</u>=E_{RS}(RC), I=H(ID), N=H(SN),K=KDF2(I, PW), PV' =H(ID, K),
HI=H(I|PV'|RS|RC|N), <u>EI</u>=E_{RC}(I|PV'|HI|N)

④ Client → CS: DownloadRequest(ERC, EI)

⑤ CS: RC = D_{RS}(ERC), I|PV'|HI|N=D_{RC}(EI), HI'=H(I|PV|RS|RC|N),
I|H(PV')|HI|N=? I|PV|HI'|N, <u>ESC</u>=E_{RC}(SC)

⑥ CS → Client: DownloadResponse(ESC) ⑦ Client: SC=D_{RC}(ESC), C =D_K(SC)

3) Credential Download Protocol from direct solutions

The way to deliver credential among different devices is through PKCS#12, which is currently supported by most browsers. Credential is double-protected by the password of private key and that of PKCS#12.

① *Device 1: PKCS#12Export(data or file(*.pfx or *.p12))*
② *Device1 → Device 2: Transfer PKCS#12 data*
③ *Device 2: PKCS#12Import(data or file(*.pfx or *.p12))*

3.4.4. Proxy signature operations

1) Proxy Signature [17] Protocol

The signing process of Client using Signing Server is as follows:

① *SS → CS: BindRequest(SD, SS_cert)* ② *CS → SS: BindResponse(ERS, CS_cert)*
③ *Client → SS: InfoRequest* ④ *SS → Client: InfoResponse(RS, SS_cert)*
⑤ Client: $\underline{ERC}=E_{RS}(RC), I=H(ID), N=H(SN), K=KDF2(I,PW), PV'=H(ID,K), \underline{D}=H(M),$
$HI=H(I|PV'|D|RS|RC|N), EI=E_{RC}(I|PV'|HI|N), \underline{ED}=E_{RS}(ERC|EI), \underline{K}=E_{RS}(K)$
⑥ *Client → SS: SignRequest (ED, D, EK, ERC)* ⑦ SS: $ERC|EI=D_{RS}(ED)$
⑧ *SS → CS: DownloadRequest(ERC, EI)*
⑨ CS: $RC=D_{RS}(ERC), I|PV'|HI|N=D_{RC}(EI), HI'=H(I|PV|RS|RC|N),$
$I|H(PV')|HI|N=?\ I|PV|HI'|N, \underline{ESC}=E_{RC}(SC)$ ⑩ *CS → SS: DownloadResponse(ESC)*
⑪ SS: $RC=D_{RS}(ERC), SC=D_{RC}(ESC), K=D_{RS}(EK), C=D_K(SC), \underline{SD}=S_{client\_key}(D)$
⑫ *SS → Client: SignResponse(SD)* ⑬ Client: $D=H(M), D'=V_{client\_cert}(SD), D=?D'$

4. Prototyping

4.1. Implementation

To demonstrate the feasibility of our architecture, we implemented a prototype system which provides consolidate user authentication for secure system. This system is developed using JSP, JAVA, iPhone development toolkit technologies. [18] This table below shows CAM's add account, signing procedure using iPhone's Application and user certificate saved in server.

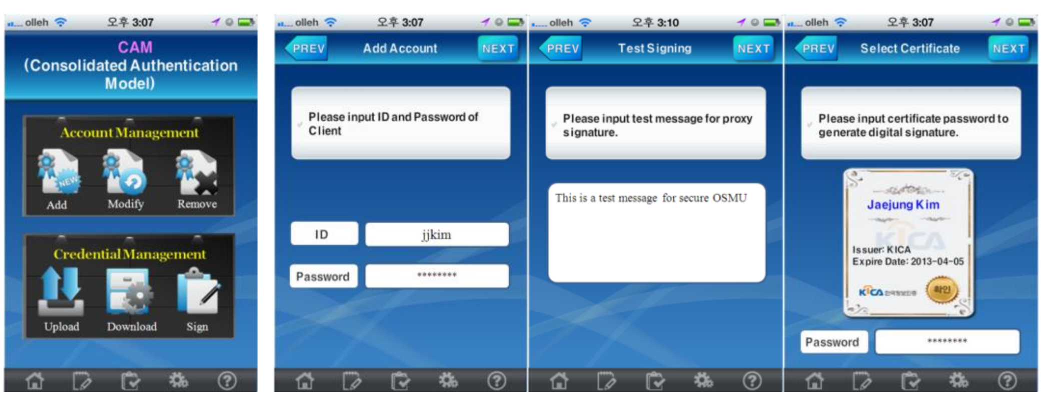

Figure 4. iOS User Interface of CAM

4.2. Simulation

The simulation platform and certificate to show performance difference between CAM and conventional PKI is as follows;

Table 2. Simulation Environments

Category	Description
Platform	PC (Intel dual core, 3.2GHz), Smart Phone (iPhone 4), UNIX (SUN Fire V240 1.5GHz*2ea)
certificate	User certificate, Device certificate: RSA2048bit/SHA256

A general scenario of CAM and PKI is as follows:

1) CAM: User's device requests proxy signature to Signing server and service provider verify that result. 2) PKI: User's device generates digital signature and service provider verify it. We simulated RSA signing and verification for each platform. The CAM provide more a time saving and enhanced security than traditional authentication using PKI by providing consolidated authentication for user, device and contents.

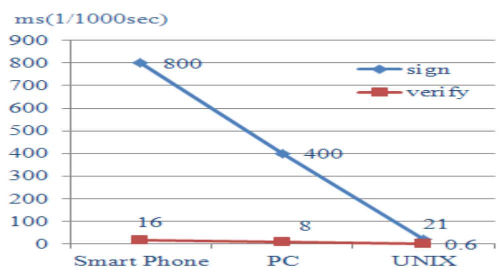

Figure 5. Result of RSA signing and verify

Table 3. Simulation Result

Category	PKI	CAM
User authentication	Slow (PC, Smart Phone)	Fast (UNIX)
Authentication method	Provided by each service	Centralized management

5. Comparison and verification

We demonstrate that CAM architecture can solves the existing problems by satisfying framework requirements, protocol requirements, privacy protection requirements, and by comparing with the current user authentication model.

The CAM satisfies framework requirements as follow:

Table 4. Framework Requirements

No	Description	CAM
F1	The framework must support both "credential server" and "direct" solutions.	O
F2	The "credential server" and "direct" solutions should use the same technology.	O
F3	The framework must allow for protocols which support different user authentication schemes.	O
F4	The details of the actual credential type or format must be opaque to the protocol.	O
F5	The framework must allow use of different transports.	O

The CAM satisfies protocol requirements as follow:

Table 5. Protocol Requirements

No	Description	CAM
G1	Credential transfer both to and from a device must be supported.	O
G2	Credentials must not be forced by the protocol to be present in cleartext at any device other than the end user's.	O
G3	The protocol should ensure that all transferred credentials be authenticated in some way.	O
G4	The protocol must support a range of cryptographic algorithms.	O
G5	The protocol must allow the use of various credential types and formats.	O
G6	One mandatory to support credential format must be defined.	O
G7	One mandatory to support user authentication scheme must be defined.	O
G8	The protocol may allow credentials to be labeled with a text handle.	O
G9	Full I18N support is required (via UTF8 support)	O
G10	The protocol is able to support privacy, that is, anonymity for the client.	O
G11	Transferred credentials may incorporate timing information.	O

The CAM compare with the old user authentication model as follow:

Table 6. Comparison between Old model and CAM

Category	Framework requirements	Protocol requirements	Privacy protection requirements	Proxy signature
Old model	△	△	△	X
CAM	O	O	O	O

O: provided △: partially provided X: not provided

6. Conclusion and Future Work

In this paper, we discuss the security and privacy issues of the current user authentication model that are not able to provide credential roaming in cloud computing environments due to the absence of securely available credential protocol in consolidated user authentication method. In order to solve this problem, we proposed the secure CAM architecture so that one credential is applicable to various mobile devices in cloud computing environments.

Following is contributions of N-screen based consolidated user authentication model for internet services that meets framework and protocol requirement of credentials and privacy protection requirements in a cloud computing environments. We designed the secure CAM architecture in cloud computing environments, which not only provides more flexible authentication framework but also leads to safer credential management in operating various mobile devices such as smart phone, smart pad, etc. We define framework architecture, credential profile, protocol framework for consolidated authentication mechanism in order to provide an appropriate user authentication model for a cloud computing environments.

The future study will continue to focus the design and implement of our suggested model, and we will expand to new devices and environments.

References

[1] D. H. Bae, "A Study on the Revision of the 'Personal Information Protection Act' and its Related Acts", Research of IT and Law, vol. 6, **(2012)** February, pp. 1-261
[2] J. Kim and S. Hong, "A Method of Risk Assessment for Multi-Factor Authentication", Journal of Information Processing Systems (JIPS), **(2011)** April, Page 187-198.
[3] H. Kharche, D. S. Chouhan, "Building Trust in Cloud Using Public Key Infrastructure", International Journal of Advanced Computer Science and Applications, vol. 3, no. 3, **(2012)**.
[4] Y. Wei, "Recommended Industries for Foreign Investment-Cloud Computing Industry", ITRI IEK Report, **(2010)**.
[5] K. D. Kadam, S. K. Gajre and R. L. Paikrao, "Security issues in Cloud Computing", IJCA Proceedings on National Conference on Innovative Paradigms in Engineering and Technology (NCIPET 2012) ncipet(11): **(2012)**, pp. 22-26.
[6] J. Wayne and G. Timothy, "Guidelines on security and privacy in public cloud computing", NIST Special Publication 800-144, **(2011)** December**.**
[7] J. Y. Ahn, J. G. Song, D. J. Hwang and S. S. Kim, "Trends in M2M Application Services Based on a Smart Phone", Communications in Computer and Information Science, vol. 117, **(2010)**, pp. 50-56.
[8] J. Kim and S. Hong, "One-Source Multi-Use System having Function of Consolidated User Authentication", YES-ICUC 2011, **(2011)**.
[9] IETF RFC 3157, Securely Available Credentials-Requirements, **(2001)** August.
[10] IETF RFC 3760, Securely Available Credentials-Credential Server Framework, **(2004)** April.
[11] IETF RFC 3767, Securely Available Credentials-Securely Available Credentials Protocol, (**2004)** June.
[12] RSA Lab. PKCS #12 v1.0: Personal Information Exchange Syntax, **(1999)** June.
[13] RSA Lab. PKCS #15 v1.1: Cryptographic Token Information Syntax Standard, **(2000)** June.
[14] ITEF RFC 6070, PKCS #5: Password-Based Key Derivation Function 2 (PBKDF2), **(2011)** January.
[15] ITEF RFC 5246, the Transport Layer Security (TLS) Protocol Version 1.2, August **(2008)**
[16] ITEF RFC 2631, Diffie-Hellman Key Agreement Method, **(1999)** June.
[17] C. Popescu, "A Secure Proxy Signature Scheme with Delegation by Warrant", Studies in Informatics and Control, vol. 20, no. 4, **(2011)** December.
[18] M. Sinkinson, "The Complete iPhone Development Toolbox", **(2010)** March 2.

Authors

Jaejung Kim

Jae-Jung Kim received his BS degree in Computer Science from Chungnam University in 1997 and MS degree in Information Security from Korea University in 2003, respectively. Since 1997, he stayed in LG-CNS Systems and Korea Information Certification Authority Inc. to develop PKI solutions. And now he is undertaking a doctorate course as a member of the information security lab at Sungshin University. His research interests include Public Key Infrastructure (PKI), security architecture and protocol, cross certification, anonymous authentication, and device authentication.

Seng-phil Hong

Professor Seng-phil Hong received his BS degree in Computer Science from Indiana State University, and MS degree in Computer Science from Ball State University at Indiana, USA. He researched the information security for PhD at Illinois Institute of Technology from 1994 to 1997, He joined the Research and Development Center in LG-CNS Systems, Inc since 1997, and he received Ph.D. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST) in Korea. He is actively involved in teach and research in information security at The Sungshin Women's University, Korea. His research papers appeared in a number of journals such as ACM Computing, Springer-Verlag's Lecture Notes in Computer Science, etc. His research interests include access control, security architecture, Privacy, Smart Device Security and e-business security.