4-2 Secure Service Framework on Mobile Ethernet

INOUE Daisuke and KURODA Masahiro

Diverse and highly-developed mobile services will arise on next generation wireless networks. Users on the networks will have to securely manage a lot of credentials in their mobile terminals. It is necessary to provide a secure and easy-to-use framework for managing credentials independent of mobile terminals. In the New Generation Mobile Network Project, we have designed a secure service framework that separates credentials from a mobile terminal and stores them into a tamper resistant smartcard. This paper describes an overview of the secure service framework and its prototype implementation.

Keywords
Secure service framework, Self-delegation, Contact-less smart card,
Service authentication, Mobile Ethernet

1 Introduction

With the popularization of wireless communication technologies such as third-generation cellular phone systems and wireless LANs, various types of services have come to be available in the mobile environment in addition to e-mail and web browsing: Blogs, SNS services, electronic commerce, electronic auctions, electronic public services, online banking, and more. Wireless communication technologies continue to progress toward more high-speed and wide-area accesses, such as ultra-fast wireless LANs, wireless MANs, and fourth-generation cellular systems, and so mobile services provided on next-generation wireless networks are also expected to employ more advanced technologies. However, from the viewpoint of the user, ease of use and flexibility of these services are not ideal. For example, authentication and payment methods vary among providers, and/or subscription services provided by one provider may not be accessed from another terminal. Therefore, in order to promote the popularization and development of mobile services in the future, it will be essential to create a service framework that can offer convenience, flexibility, and consistency.

On the other hand, terminals such as cellular phones, laptops, and PDAs, (all referred to below as “mobile terminals”) currently retain credentials for accessing mobile services (such as cached passwords, various certificates, and in the case of cellular phones, subscriber ID and its corresponding keys) within the individual unit. Further, several models of cellular phones with internalized FeliCa[1] functions have recently entered the market, and important personal and accounting information is now stored convergently on the mobile terminals. Many mobile terminals are equipped with user authentication functions based on password systems and biometrics installed within the individual terminal, but in many cases, users purposely deactivate these authentication functions due to the significant inconvenience of using the terminals with these functions activated. As a result, the damage incurred when mobile terminals are lost or stolen may be enormous. To prevent this risk, next-generation mobile services will require a secure framework that offers both convenience and security.

Taking the above considerations into account, research and development of a secure service framework—focusing on the basic technologies for next-generation mobile services offering high security and convenience—was carried out on the New Generation Mobile Network Project[2]. This article will give an overview of this secure service framework, and will also report on the prototype implementation of the framework on the Mobile Ethernet integrating various types of wireless systems.

2 Overview of the secure service framework

It is expected that a wide spectrum of mobile services will be popularized and developed on a next-generation wireless network, and we will most likely see an explosion in the number of service providers. Further, in the Mobile Ethernet[3] (an integrated heterogeneous wireless network developed in the course of the New Generation Mobile Network Project), a model has been proposed in which multiple operators (telecommunication carriers), both large and small, interoperate various wireless systems according to a standardized interface. In a next-generation wireless network, we may expect a situation in which the user eventually holds a great number of credentials both on the network and application levels, allowing seamless access and use of various mobile services over a network managed by multiple independent operators.

To address such a situation, we have designed a secure service framework composed of a mobile terminal, a contact-less smartcard, and a self-delegation protocol between the two devices, providing a balance between user-friendliness and overall system security (Fig. 1).

The features of the secure service framework are as follows.

(A) The credentials are separated from the mobile terminal and stored on a tamper-resistant[*1] contact-less smartcard.

(B) The mobile terminal is equipped with a contact-less smartcard reader and multiple wireless interfaces (and, as will be described later, a sensor for biometric

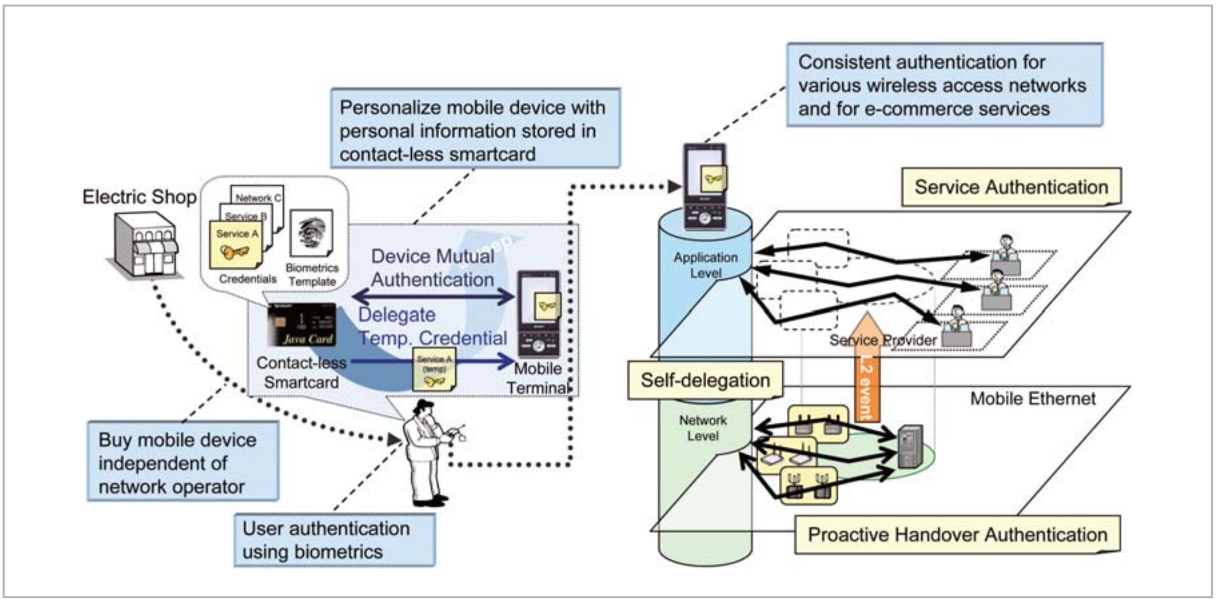

Personalize mobile device with personal information stored in contact-less smartcard
Consistent authentication for various wireless access networks and for e-commerce services
Electric Shop
Network C
Service B
Service A
Credentials
Biometrics Template
Device Mutual Authentication
Delegate Temp. Credential
Java Card
Contact-less Smartcard
Service A (temp)
Mobile Terminal
Buy mobile device independent of network operator
User authentication using biometrics
Service Authentication
Application Level
Service Provider
Self-delegation
L2 event
Mobile Ethernet
Network Level
Proactive Handover Authentication

Fig. 1 Outline of the secure service framework

[*1] Resistance to unauthorized analysis by external users of the internal structures of software or hardware and the information stored in these programs or devices.

authentication).

(C) When the user holds the contact-less smartcard up to the mobile terminal, a temporarily effective credential (referred to below as the "the temporal credential") will be calculated based on the credentials stored within the smartcard, and the temporal credential will be transmitted to the mobile terminal (self-delegation).

(D) The user will perform network and service authentications using the temporal credential that has been self-delegated to the mobile terminal, allowing access to the mobile services.

(E) The term of validity or the number of valid access operations for the temporal credential can be set flexibly according to each service.

The contact-less smartcard can store multiple credentials on both the network and application levels, and so it will be possible to centralize the credentials held by the user on a single smartcard. The user need only hold the smartcard against the mobile terminal to delegate the temporal credentials to the mobile terminal. The mobile terminal will not retain the original credential permanently; instead it will use the temporal credentials given by the smartcard to carry out the required network/service authentication. The term of validity or the number of valid access operations for the temporal credential can be set flexibly according to each service. For example, the temporal credential for network access may be valid for 24 hours, while that for network banking would be valid only once. This setup will mitigate the potential damage incurred by the user in the case of lost or stolen mobile terminals by preventing unauthorized use of services.

By separating the credentials from mobile terminals and storing them in contact-less smartcards in this manner, the user will have to keep the contact-less smartcard safely protected, instead of the mobile terminal. The contact-less smartcard is tamper-resistant, and also has the same shape as a bank card or credit card, so the users will likely be able to keep this card in relative safety compared to terminals. However, technical countermeasures will have to be devised against unauthorized use of the smartcard, and so biometric authentication was introduced to the present framework in addition to the above characteristics (A)-(E), as follows.

(F) The contact-less smartcard will store the user's biometric information.

(G) When the user attempts to use the contact-less smartcard, the biometric information provided by the user will be verified against that stored in the smartcard.

(H) After successful verification, the smartcard will be activated and ready for self-delegation.

The information required for biometric authentication will be stored in the contact-less smartcard, and this information will be used to check whether the user is the legitimate owner of the card. Unless the biometric authentication is successful, the smartcard will not be activated, and self-delegation will not be carried out, thus preventing unauthorized use of stolen smartcards. Note that we are assuming a case in which the sensors for biometric authentication are provided on the mobile terminals, and that verification will be made via the mobile terminal.

The present framework should allow the centralization of multiple credentials on a single contact-less smartcard. The user can carry out self-delegation to a mobile terminal by a simple motion, enabling access to the mobile services. The potential damage incurred by the user in the case of lost or stolen mobile terminals or smartcards will thus be greatly reduced through the use of this system.

3 Implementation of the secure service framework prototype

In this section we will describe an experiment for the implementation of the secure service framework prototype. The present prototype consists of a self-delegation protocol between the contact-less smartcard and mobile terminal and a service authentication protocol between service providers and mobile terminal. For more information on proactive handover authentication shown in the Fig. 1, a network authentication technology for the Mobile Ethernet, see reference[4].

In the framework described above, it is assumed that the mobile terminals are equipped with the sensors for biometric authentication and contact-less smartcard readers. However, technical considerations such the physical size of the card readers and the performance of the terminals (including power consumption) have hampered the creation of such a system for currently available commercial products. Thus, in order to conduct a prototype implementation test on current terminal models, the contact-less smartcard readers and biometric authentication sensors that will ultimately be installed in mobile terminals were placed on a structure separate from the terminals. This newly incorporated structure was referred to as the self-delegation unit.

Self-delegation of the temporal credential between the contact-less smartcard and the mobile terminal was made via the self-delegation unit (Fig. 2). The processing involved in authentication using biometric information stored on the contact-less smartcard was also performed on the self-delegation unit[*2]. Note that in the implementation of the present prototype, fingerprint authentication was used as the method for biometric authentication[*3].

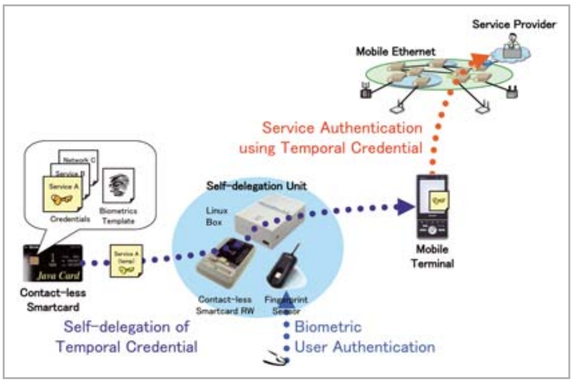


Fig.2 *Flow of self-delegation carried out via the self-delegation unit*

In the experiment, the mobile terminal accessed the service providers via the Mobile Ethernet, and the service provided was assumed to be a video-streaming distribution service. The protocol for the present prototype implementation consisted only of the symmetric crypto system. See references[4]and[6]for details regarding the self-delegation protocol using the asymmetric crypto system.

3.1 Hardware architecture

Figure 3 shows the hardware architecture for the present prototype.

(1) Mobile terminal

The mobile terminal employed was the Zaurus SL-6000W by Sharp Corporation. The SL-6000W features Bluethooth and IEEE802.11b wireless LAN technology, and a 3 G-network card (FOMA card) was inserted in its compact flash card slot. The mobile terminal was connected to the self-delegation unit via PAN using internal Bluetooth, and the internal wireless LAN and the FOMA card were used to establish connection to the Mobile Ethernet.

(2) Self-delegation unit

The self-delegation unit consisted of the Linux box and the OpenBlockS 266 manufactured by Plat'Home Co., Ltd. A Bluetooth card and a USB card were fixed to the two PCMCIA adapters added to the main body of the OpenBlockS 266. In addition, a contact-less smartcard reader and a fingerprint sensor were connected to the USB card. The self-delegation unit was connected to the mobile terminal via PAN using Bluetooth, and contact-less communication with ISO/IEC14443 Type B was established with the smartcard using a contact-less smartcard RW unit.

(3) Contact-less smartcard

The contact-less smartcard employed was

[*2] Insofar as the processing speed of the smartcard allows, the match-on-card method is the most favored method of verification since this entails biometric authentication without transmitting biometric information from the card.

[*3] Fingerprint authentication is commonly known to be vulnerable to forged fingerprints made of gelatin[5]. In the future, it will be necessary to consider the adoption of multi-modal biometric authentication, which combines multiple authentication methods (such as iris recognition and voice authentication).

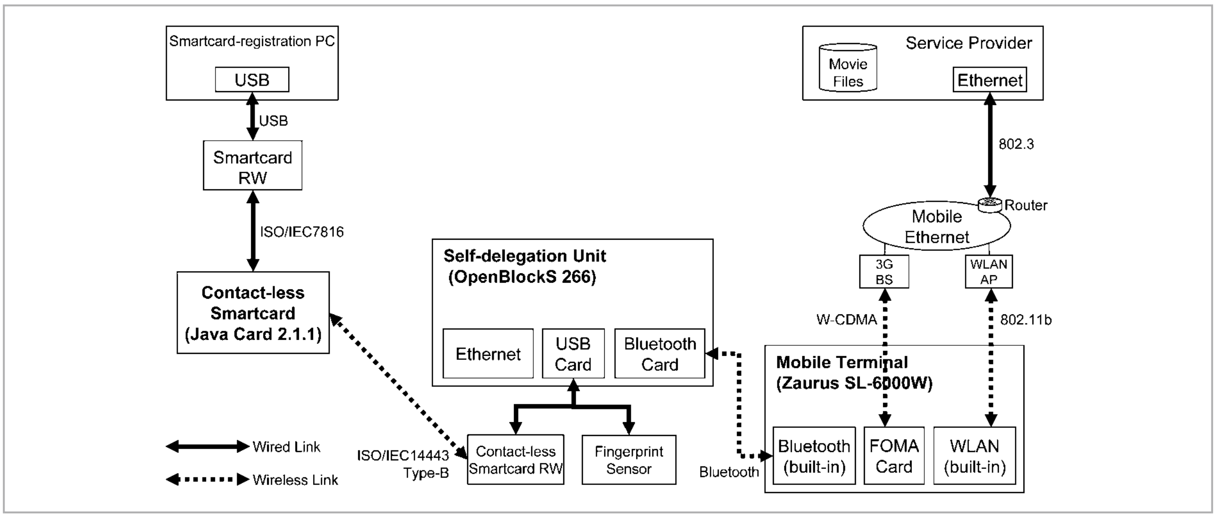


Fig.3 Hardware architecture

a smartcard by Sharp Corporation, in compliance with Java Card 2.1.1[7]. The smartcard has a built-in 16-bit CPU, 8-KB RAM, and a 1-MB flash memory. The card performs contact-less communication (ISO/IEC14443 Type B compliant) with the self-delegation unit using a built-in antenna, and also performs contact communication (ISO/IEC7816 compliant) with the PC for smartcard registration.

(4) Service provider

A general-purpose Linux PC was employed as the service provider. The service provider established a LAN connection via the Ethernet interface to the router connected to the Mobile Ethernet.

(5) Smartcard-registration PC

A general-purpose Windows PC was used as the smartcard-registration PC[*4]. The smartcard-registration PC featured a USB connection to the contact smartcard RW device, and performed contact communication (ISC/IEC7816 compliant) with the smartcard via the smartcard RW device.

Figure 4 shows the external view of the implemented prototype.

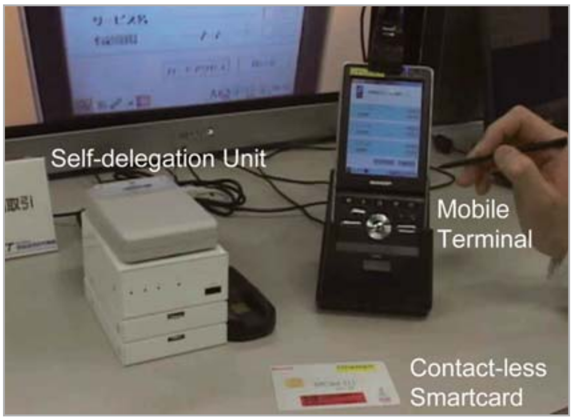


Fig.4 External view of the implemented prototype

3.2 Software architecture

Figure 5 shows the software architecture of the present prototype.

The details of the software architecture are omitted here; only the basic flow of the process will be given.

(0) In preparation, the credentials (key information) and biometric information (fingerprint template) are registered on the contact-less smartcard using the PC for smartcard registration.

(1) Device mutual authentication[*5] using a

[*4] A PC for registering credentials and biometric information on the contact-less smartcard. In actual applications, this function will be handled by the service provider.

[*5] This device mutual authentication is performed by the challenge and response method using 3DES. This process is not necessary when the mobile terminals perform the functions of the self-delegation unit. Note that the use of the 3DES is due to the restrictions in the version of Java Card employed here.

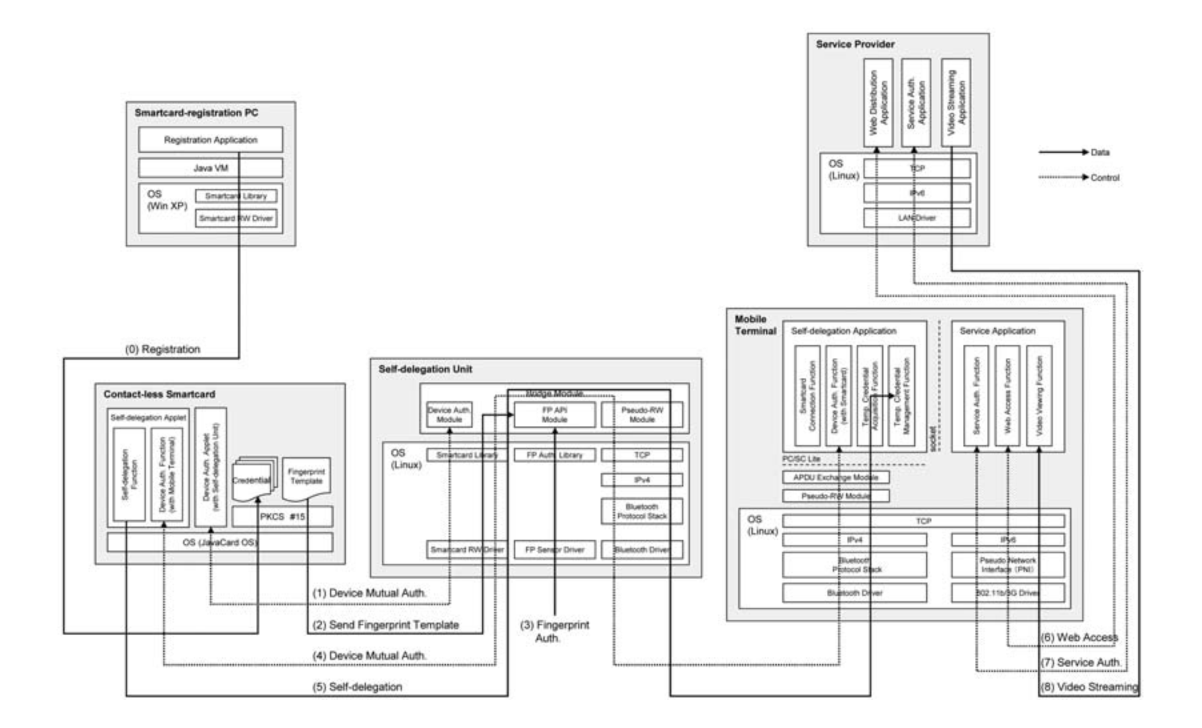


Fig.5 Software architecture

pre-shared key between the self-delegation unit and the smartcard is made by holding the smartcard against the self-delegation unit.

(2) The fingerprint template is transmitted from the smartcard to the self-delegation unit.

(3) Fingerprint authentication is carried out by having the user present the fingertip to the fingerprint sensor installed in the self-delegation unit. When authentication has been successfully completed, the self-delegation unit authorizes communication between the contact-less smartcard and the mobile terminal.

(4) Device mutual authentication[*6] is executed, using a pre-shared key between the contact-less smartcard and the mobile terminal.

(5) A temporal credential is calculated by the contact-less smartcard and self-delegated to the mobile terminal.

(6) Web browsing on the service provider site from the mobile terminal is performed via the Mobile Ethernet.

(7) When video content requiring service authentication is accessed, service authentication[*6] between the mobile terminal and the service provider is performed using a temporal credential as the key.

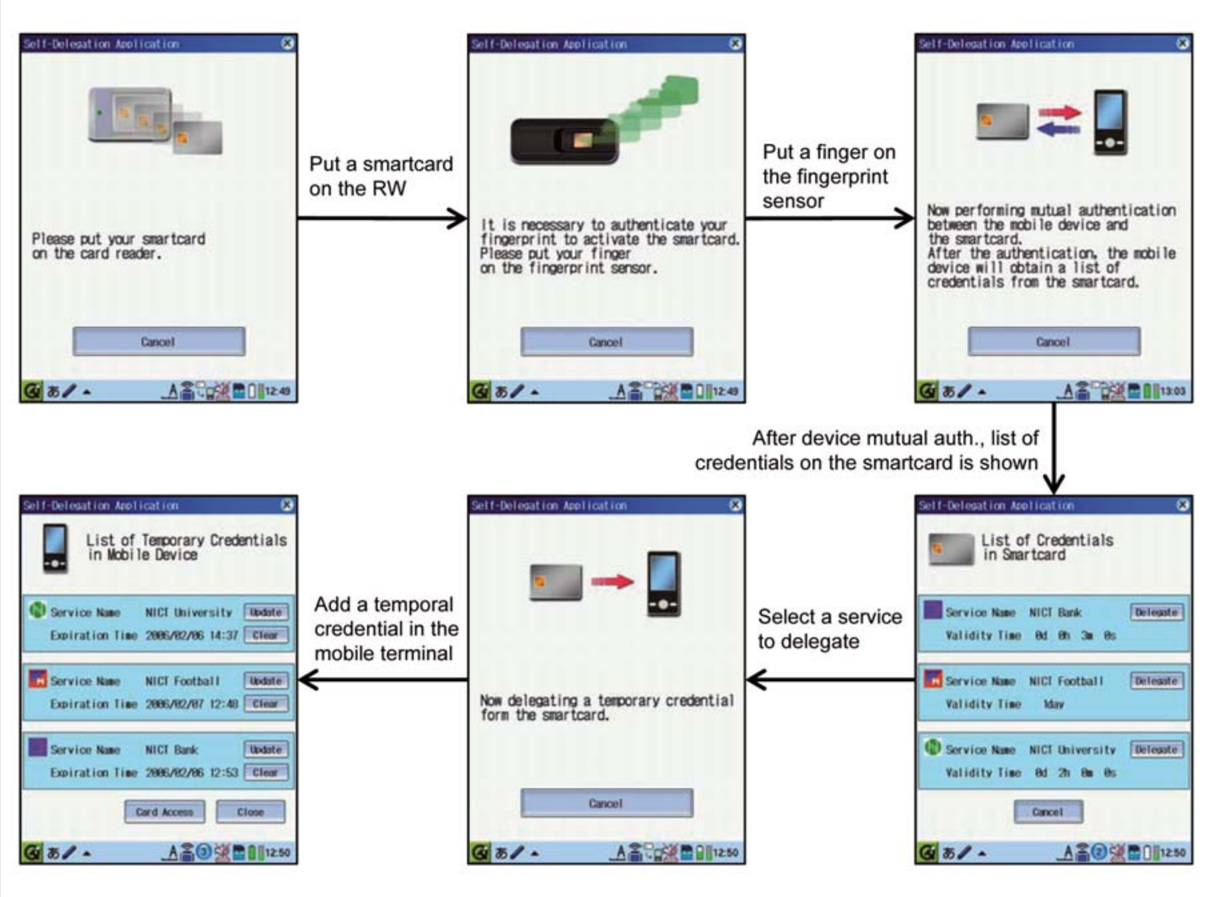
(8) When service authentication has been successfully performed, the service provider proceeds with the video distribution.

Figure 6 shows an example of the series of GUI screenshots for the self-delegation application (on the mobile terminal), corresponding to steps (1)-(5) above.

3.3 Self-delegation protocol and the service authentication protocol

Here we will discuss the details of the protocol for the self-delegation process between the contact-less smartcard and the mobile terminal, in addition to service authentication performed between the mobile terminal and the service provider.

[*6] This device mutual authentication is performed by the challenge and response method using HAMC SHA-1. Since collusions have been reported with SHA-1, it will be necessary to replace this with a different hash function, such as SHA-256.



Self-Delegation Application
Cancel
Put a smartcard on the RW
Put a finger on the fingerprint sensor
After device mutual auth., list of credentials on the smartcard is shown
Select a service to delegate
Add a temporal credential in the mobile terminal
Delegate
Update
Clear
Card Access
Close

Please put your smartcard on the card reader.

It is necessary to authenticate your fingerprint to activate the smartcard. Please put your finger on the fingerprint sensor.

Now performing mutual authentication between the mobile device and the smartcard. After the authentication, the mobile device will obtain a list of credentials from the smartcard.

List of Credentials in Smartcard

Service Name NICT Bank
Validity Time 0d 0h 3m 0s

Service Name NICT Football
Validity Time 1day

Service Name NICT University
Validity Time 0d 2h 0m 0s

Now delegating a temporary credential form the smartcard.

List of Temporary Credentials in Mobile Device

Service Name NICT University
Expiration Time 2006/02/06 14:37

Service Name NICT Football
Expiration Time 2006/02/07 12:48

Service Name NICT Bank
Expiration Time 2006/02/06 12:53

Fig.6 Example of the series of GUI screenshots for the self-delegation application (on the mobile terminal)

3.3.1 Definition of symbols

The definitions of the symbols used in the following sections are provided below.

PID	Contact-less smartcard (Personal Identity Device)
MT	Mobile terminal
SP	Service provider
SList	List of information on services provided by SP Service information (Sid, Uid, Sname, ValidTerm, SExpTime, K_{PS})
SListMT	Of the SList, those handed over to the mobile terminal (Sid, Sname, ValidTerm)
Sname	Name of service provided by SP
Sid	Service ID uniquely assigned to the service provided by SP
Uid	User ID uniquely assigned to the user signed up for the service provided by SP
R_n	Random number
ValidTerm	Term of validity for the temporal credential
CurtTime	Current time
ExpTime	Time of expiration of temporal credential ExpTime = CurtTime + ValidTerm
SExpTime	Time of expiration of service
h(k, m)	Computation of keyed hash function using key k corresponding to message m
E(k, m)	Encryption computation using key k corresponding to message m
MK	Master key of service provider
K_{PM}	Pre-shared key between PID and MT
K_{PS}	Pre-shared key (credential) between PID and SP $K_{PS} = h(MK_s, Uid)$
EK_{PM}	Encryption key used for cryptographic communication between PID and MT
TK_{MS}	Temporal shared key between PID and SP (temporal credential)
||	Data concatenation
⊕	Exclusive disjunction

3.3.2 Details of the protocol

Figure 7 presents the self-delegation and service authentication protocols. Steps (1)-(13) correspond to the self-delegation protocol, and steps (14)-(21) correspond to the service authentication protocol. It is assumed here that shared keys KPM and KPS have already been established between the contact-less smartcard and the mobile terminal and between the contact-less smartcard and the service provider, respectively. In addition, the protocol starts from the state in which biometric authentication has been completed by the user and the contact-less smartcard is activated and ready for self-delegation. (The self-delegation unit is omitted from Fig. 7 since it operates transparently between the smartcard and the terminal.)

Below is a summary of the processes carried out in each step.

Steps (1)-(7) : Device mutual authentica-

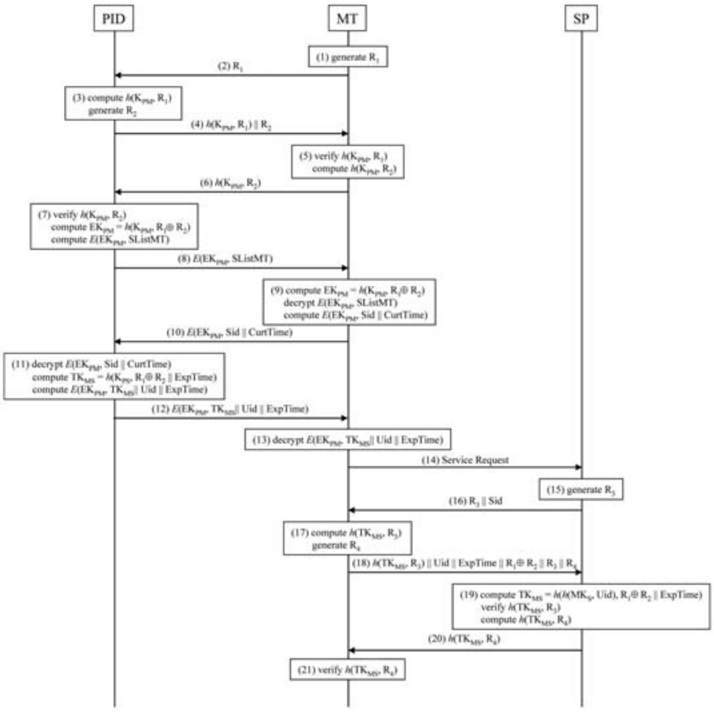

PID
MT
SP
(1) generate R_1
(2) R_1
(3) compute $h(K_{PM}, R_1)$
generate R_2
(4) $h(K_{PM}, R_1) \| R_2$
(5) verify $h(K_{PM}, R_1)$
compute $h(K_{PM}, R_2)$
(6) $h(K_{PM}, R_2)$
(7) verify $h(K_{PM}, R_2)$
compute $EK_{PM} = h(K_{PM}, R_1 \oplus R_2)$
compute $E(EK_{PM}, SListMT)$
(8) $E(EK_{PM}, SListMT)$
(9) compute $EK_{PM} = h(K_{PM}, R_1 \oplus R_2)$
decrypt $E(EK_{PM}, SListMT)$
compute $E(EK_{PM}, Sid \| CurtTime)$
(10) $E(EK_{PM}, Sid \| CurtTime)$
(11) decrypt $E(EK_{PM}, Sid \| CurtTime)$
compute $TK_{MS} = h(K_{PS}, R_1 \oplus R_2 \| ExpTime)$
compute $E(EK_{PM}, TK_{MS} \| Uid \| ExpTime)$
(12) $E(EK_{PM}, TK_{MS} \| Uid \| ExpTime)$
(13) decrypt $E(EK_{PM}, TK_{MS} \| Uid \| ExpTime)$
(14) Service Request
(15) generate R_3
(16) $R_3 \| Sid$
(17) compute $h(TK_{MS}, R_3)$
generate R_4
(18) $h(TK_{MS}, R_3) \| Uid \| ExpTime \| R_1 \oplus R_2 \| R_3 \| R_4$
(19) compute $TK_{MS} = h(h(MK_S, Uid), R_1 \oplus R_2 \| ExpTime)$
verify $h(TK_{MS}, R_3)$
compute $h(TK_{MS}, R_4)$
(20) $h(TK_{MS}, R_4)$
(21) verify $h(TK_{MS}, R_4)$

Fig.7 *Self-delegation/service authentication protocols*

tion and encryption key (EKPM) sharing between contact-less smartcard and mobile terminal

Steps (7)-(11) : Transmission of encrypted service information list (SListMT) from contact-less smartcard to mobile terminal, and transmission of the encrypted service ID for self-delegation (Sid) and current time from mobile terminal

Steps (11)-(13) : Computation of temporal credential (TKMS) in the contact-less smartcard and transmission of the encrypted temporal credential, user ID, and the term of validity of this ID, to the mobile terminal. The terminal decrypts the information and acquires the temporal credential.

Steps (14)-(21) : Service authentication between mobile terminal and service provider using the temporal credential (mutual authentication)

4 Performance evaluation

In this section, we will evaluate the performance of the self-delegation protocol between the contact-less smartcard and the mobile terminal, which is expected to have the most impact on the overall usability of the system. Figure 8 shows the details of the flow of the protocol, including the self-delegation unit (GW).

In Fig. 8, time measurements were performed 100 times for each step to calculate the average time for each. Note that since time measurements could not be performed for the contact-less smartcard, an applet that executes the target process for measurement was installed in the IC card to measure the time of communication when the given process was performed, which was compared to the time measured for an applet when the process was not. The difference between the two measurements was regarded as the time required for

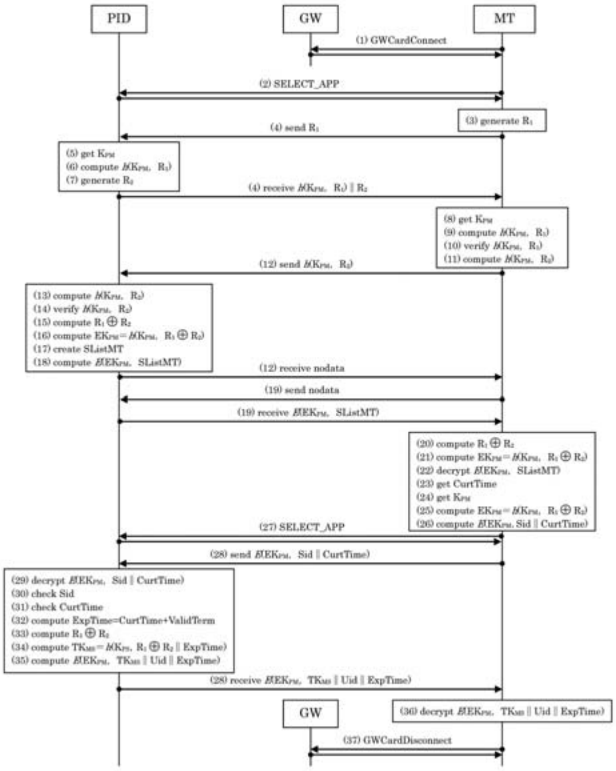

PID
GW
MT
(1) GWCardConnect
(2) SELECT_APP
(3) generate R_1
(4) send R_1
(5) get K_{PM}
(6) compute $H(K_{PM}, R_1)$
(7) generate R_2
(4) receive $H(K_{PM}, R_1) \| R_2$
(8) get K_{PM}
(9) compute $H(K_{PM}, R_1)$
(10) verify $H(K_{PM}, R_1)$
(11) compute $H(K_{PM}, R_2)$
(12) send $H(K_{PM}, R_2)$
(13) compute $H(K_{PM}, R_2)$
(14) verify $H(K_{PM}, R_2)$
(15) compute $R_1 \oplus R_2$
(16) compute $EK_{PM} = H(K_{PM}, R_1 \oplus R_2)$
(17) create SListMT
(18) compute $E(EK_{PM}, \text{SListMT})$
(12) receive nodata
(19) send nodata
(19) receive $E(EK_{PM}, \text{SListMT})$
(20) compute $R_1 \oplus R_2$
(21) compute $EK_{PM} = H(K_{PM}, R_1 \oplus R_2)$
(22) decrypt $E(EK_{PM}, \text{SListMT})$
(23) get CurtTime
(24) get K_{PM}
(25) compute $EK_{PM} = H(K_{PM}, R_1 \oplus R_2)$
(26) compute $E(EK_{PM}. \text{Sid} \| \text{CurtTime})$
(27) SELECT_APP
(28) send $E(EK_{PM}, \text{Sid} \| \text{CurtTime})$
(29) decrypt $E(EK_{PM}, \text{Sid} \| \text{CurtTime})$
(30) check Sid
(31) check CurtTime
(32) compute ExpTime=CurtTime+ValidTerm
(33) compute $R_1 \oplus R_2$
(34) compute $TK_{MS} = H(K_{PS}, R_1 \oplus R_2 \| \text{ExpTime})$
(35) compute $E(EK_{PM}, TK_{MS} \| \text{Uid} \| \text{ExpTime})$
(28) receive $E(EK_{PM}, TK_{MS} \| \text{Uid} \| \text{ExpTime})$
GW
(36) decrypt $E(EK_{PM}, TK_{MS} \| \text{Uid} \| \text{ExpTime})$
(37) GWCardDisconnect

Fig.8 Detailed flow of the self-delegation protocol

the process on the smartcard. Further, to measure the communication time between the smartcard and the mobile terminal, an applet was installed in the smartcard to receive data from the mobile terminal and to return data (of a predetermined size) for each evaluation item; here, the measured time was regarded as the communication time.

Table1 shows the results of the performance evaluation.

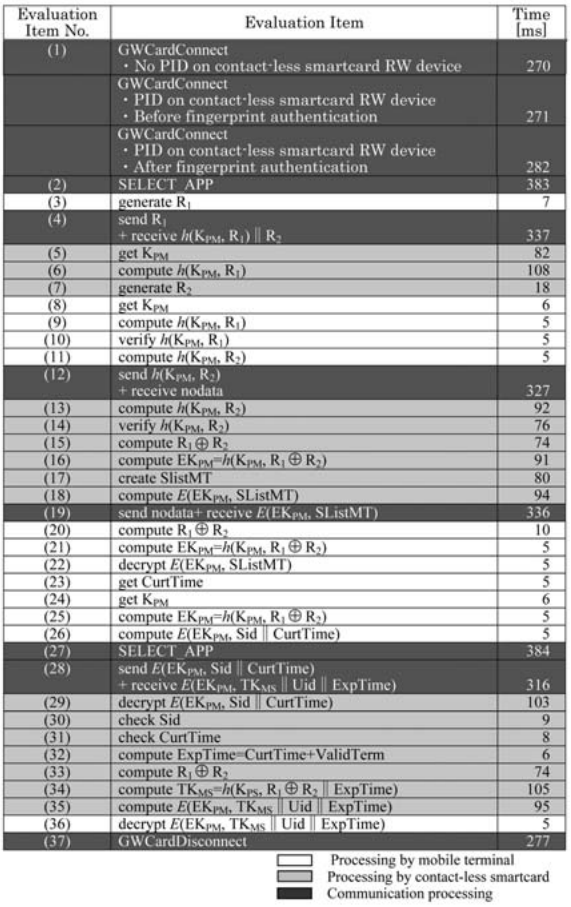
It can be seen from Table 1 that the time required for computation of the symmetric crypto system (3 DES with 2 keys) and the keyed hash function (HMAC SHA-1) in the mobile terminal were both approximately 5 ms, and that the total processing time required to complete self-delegation on the mobile terminal side was less than 80 ms.

In contrast, the time required for computation in the symmetric crypto system of the contact-less smartcard and the keyed hash function were both approximately 100 ms, and the total processing time required to complete self-delegation on the smartcard side exceeded 1,100 ms.

The process requiring the most time in the self-delegation protocol is transmission via the self-delegation unit; the round-trip time between the terminal and the smartcard exceeds 300 ms. A total communication time of more that 2,600 ms was required for completion of self-delegation, thus Bluetooth communication between the terminal and self-delegation unit represents the bottleneck in this protocol. However, as stated in Section **2**, if the contact-less smartcard reader and the fingerprint sensor can be installed in the mobile terminal, this communication time will be negligible.

The present prototype implementation experiment was constructed using the self-delegation protocol only for the symmetric crypto system, and so we have been able to achieve a degree of performance that would not impinge

Table 1 Results of performance evaluation



Evaluation Item No.	Evaluation Item	Time [ms]
(1)	GWCardConnect ・No PID on contact-less smartcard RW device	270
	GWCardConnect ・PID on contact-less smartcard RW device ・Before fingerprint authentication	271
	GWCardConnect ・PID on contact-less smartcard RW device ・After fingerprint authentication	282
(2)	SELECT_APP	383
(3)	generate R_1	7
(4)	send R_1 + receive $h(K_{PM}, R_1) \parallel R_2$	337
(5)	get K_{PM}	82
(6)	compute $h(K_{PM}, R_1)$	108
(7)	generate R_2	18
(8)	get K_{PM}	6
(9)	compute $h(K_{PM}, R_1)$	5
(10)	verify $h(K_{PM}, R_1)$	5
(11)	compute $h(K_{PM}, R_2)$	5
(12)	send $h(K_{PM}, R_2)$ + receive nodata	327
(13)	compute $h(K_{PM}, R_2)$	92
(14)	verify $h(K_{PM}, R_2)$	76
(15)	compute $R_1 \oplus R_2$	74
(16)	compute $EK_{PM}=h(K_{PM}, R_1 \oplus R_2)$	91
(17)	create SlistMT	80
(18)	compute $E(EK_{PM}$, SListMT)	94
(19)	send nodata+ receive $E(EK_{PM}$, SListMT)	336
(20)	compute $R_1 \oplus R_2$	10
(21)	compute $EK_{PM}=h(K_{PM}, R_1 \oplus R_2)$	5
(22)	decrypt $E(EK_{PM}$, SListMT)	5
(23)	get CurtTime	5
(24)	get K_{PM}	6
(25)	compute $EK_{PM}=h(K_{PM}, R_1 \oplus R_2)$	5
(26)	compute $E(EK_{PM}$, Sid ‖ CurtTime)	5
(27)	SELECT_APP	384
(28)	send $E(EK_{PM}$, Sid ‖ CurtTime) + receive $E(EK_{PM}, TK_{MS}$ ‖ Uid ‖ ExpTime)	316
(29)	decrypt $E(EK_{PM}$, Sid ‖ CurtTime)	103
(30)	check Sid	9
(31)	check CurtTime	8
(32)	compute ExpTime=CurtTime+ValidTerm	6
(33)	compute $R_1 \oplus R_2$	74
(34)	compute $TK_{MS}=h(K_{PS}, R_1 \oplus R_2$ ‖ ExpTime)	105
(35)	compute $E(EK_{PM}, TK_{MS}$ ‖ Uid ‖ ExpTime)	95
(36)	decrypt $E(EK_{PM}, TK_{MS}$ ‖ Uid ‖ ExpTime)	5
(37)	GWCardDisconnect	277

□ Processing by mobile terminal
▨ Processing by contact-less smartcard
■ Communication processing

upon overall usability. However, in order to realize a more flexible protocol—one that would not limit devices for self-delegation to those having pre-shared keys established in advance with the smartcard—it will be necessary to use an asymmetric crypto system that will place a significantly greater computational load on the system. Improvements in processing capability of the contact-less smartcard will therefore be crucial.

5 Conclusions

In this paper we described a secure service framework for a highly secure and convenient mobile service environment on a next-generation wireless network, and also presented the results of performance evaluation performed in a prototype implementation experiment. If the present framework can be developed further, it may help pave the way toward a ubiquitous environment in which the user will need only carry a contact-less smartcard bearing his/her own credentials, for temporal personalization of any pervasive device through self-delegation.

References

1 http://www.sony.co.jp/Products/felica/

2 H. Harada, M. Kuroda, H. Morikawa, H. Wakana, and F. Adachi, “The overview of the new generation mobile communication system and the role of software defined radio technology”, IEICE Trans, Commun., Vol.E86-B, No.12, Dec. 2003.

3 M. Kuroda, M. Inoue, A. Okubo, T. Sakakura, K. Shimizu, and F. Adachi, “Scalable Mobile Ethernet and Fast Vertical Handover”, Proc. IEEE Wireless Communications and Networking Conference 2004, Vol.2, pp.659- 664, Mar. 2004.

4 M. Kuroda, M. Yoshida, R. Ono, S. Kiyomoto, and T. Tanaka, “Secure Service and Network Framework for Mobile Ethernet”, Kluwer Wireless Personal Communications Special Issue on Security for Next Generation Communications, Vol.29, Issue 3-4, pp.161-190, Jun. 2004.

5 T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino “Impact of Artificial “Gummy” Fingers on Fingerprint Systems”, Proc. SPIE, Optical Security and Counterfeit Deterrence Techniques IV, Vol.4677, pp.275-289, Jan. 2002.

6 S. Kiyomoto, T. Tanaka, M. Yoshida, and M. Kuroda, “Design of Security Architecture for Beyond 3 G Mobile Terminals”, IPSJ Journal, Vol.45, No.8, pp.1856-1872, Aug. 2004.

7 http://java.sun.com/products/javacard/JC211SpecRelease.pdf

***INOUE Daisuke**, Ph.D.*

Researcher, Network Security Incident Response Group, Information Security Research Center

Information Security

***KURODA Masahiro**, Ph.D.*

Senior Researcher, Ubiquitous Mobile Communication Group, New Generation Wireless Communications Research Center

Ubiquitous Mobile Network and its Wireless Security