Regular Paper

Design of Security Architecture for Beyond 3G Mobile Terminals

Shinsaku Kiyomoto,[†] Toshiaki Tanaka,[†] Mariko Yoshida[††]
and Masahiro Kuroda[†††]

Mobile services for cellular phones are now extending beyond phone services, and into several aspect of communication services. To support the various services, mobile terminals require more complicated architecture and sophisticated human interface. Furthermore, the cost of developing mobile terminals is increasing. Therefore, some open architecture, which is extensible for adding functions, will be required. In this paper, we assume a beyond 3G mobile terminal which has open and extensible platform, and we design security architecture for it. In the open architecture, the mobile terminal has basic functions and the user can easily extend the functions by adding external devices. The security architecture also dynamically changes the security policy of the terminal, when the functions are changed. Our security architecture can reduce the complexity of management of the security policy in the mobile terminal, and expand a variety of services. Although the open architecture has many advantages, it is prone to compromise in the system security. We discuss the threats against the open architecture and propose security functions to protect against these threats. Our architecture will be suitable for beyond 3G mobile terminals that have an open and flexible platform. We believe our research includes useful information for designing future mobile services.

1. Introduction

Mobile services for cellular phones are now extending beyond phone services to provide several communication services such as e-mail, web browser, PIM (Personal Information Manager), games, e-commerce, and so forth. Mobile terminals are also expected to support various types of wireless accesses, such as 3G, WLAN, and seamless handover among them in the future[43),44)]. To support various services, mobile terminals require more complicated architecture and sophisticated human interface. Furthermore, the cost of developing mobile terminals is increasing. Therefore, some open architecture, which is extensible for adding functions, will be required[22)]. A user can change the functions of his/her mobile terminal by attaching and removing external devices and applications. Various services will be provided in the architecture.

Generally, each service has its own security policy depending on the requirements to provide the service. The security architecture of the terminals is, therefore, carefully designed satisfying all the security requirements of the available services. For this reason, existing services are subject to restriction and closure. Thus, the mechanisms to realize sophisticated security architecture for the provision of flexible and unrestricted mobile services are an open issue.

In this paper, we assume a beyond 3G mobile terminal that has an open and extensible platform, and design a security architecture for the mobile terminal. In extensible architecture, the mobile terminal has the basic functions and the user can extend these functions easily by adding external devices. The proposed security architecture also dynamically changes the security policy of the terminal when the functions are changed. Our security architecture can reduce management complexity of the security policy in the mobile terminal, and expand the variety of services. An open platform expands the functions and services for mobile terminals, and reduces the cost of developing future mobile terminals and external devices. Although the open architecture has many advantages, it is prone to compromise in the system's security. We discuss threats to the open architecture and propose some countermeasures against such threats. Our architecture will be suitable for beyond 3G mobile terminals that have an open and flexible platform.

The rest of the paper is organized as follows: At first, we introduce related works in Section 2. Next, we present the assumed beyond 3G mobile terminals and discuss their security threats in Section 3. We propose a security architecture

[†] KDDI R & D Laboratories Inc.
[††] Mitsubishi Electric Corporation
[†††] National Institute of Information and Communications Technology

in Section 4. Finally, we discuss the security of the proposed architecture and conclude the paper in Section 5 and Section 6.

2. Related Work

The security architecture for several systems or networks were provided[23),24)]. Concerning mobile communication, several security architecture has been proposed. Defense Advanced Research Project Agency (DARPA) in America started a research project called GloMo[29)] in 1994. The GloMo provided mobile users access to a range of information services, but security technologies concentrated on confidentiality and key management of several communications. Bluetooth Specification[30)] also provides security functions for link-level privacy and entity authentication using a challenge-response scheme. A. Fox and S. Gribble provided an authentication scheme for mobile devices[25)]. The scheme is based on Kerberos scheme[26)] and provides a solution of indirect authentication for mobile nodes.

WAP Forum (now OMA)[40)] also provided security functions of communications for mobile Web applications such as WTLS protocol[42)]. L.B. Michael, et al. designed and proposed a secure download system for software defined radio terminals[27)]. However, very little research on the entire mobile services has been proposed. Especially, a security architecture model whole beyond the 3G mobile services is not discussed sufficiently. 3GPP[31)], and 3GPP2[32)] provided the specification documents for the third generation mobile terminals. In the organizations, the security for communication layers such as key agreement and encryption is established, but security for the application and services are still under discussion.

Mobile It Forum[39)] in Japan proposed only higher requirements for future mobile services such as mobile commerce. Mobile electronic Transactions (MeT)[41)] proposed limited architecture for mobile commerce, which define electronic ticket application and electronic payment application.

Current security architectures are designed assuming some closed and/or trusted environments such as trusted mobile terminals. Therefore, analysis of threats and design of the security architecture assuming beyond 3G mobile services is required. MIRAI and MIRAI+ project[21)] propose several technologies for beyond 3G mobile communications and services. This paper presents research of security threats against the beyond 3G mobile services, and the security architecture for the services, as part of the MIRAI project. Other technologies such as communications security including roaming and authentication will be presented in other papers.

In this proposed security architecture, we discuss mutual device authentication, an access control mechanism, and personalization scheme. Phoenix Technologies Ltd. provides a security framework called Phoenix FirstAuthority[38)] architecture, which is a device identification scheme based on PKI technologies. In the scheme, a server PC authenticates a client PC as a user authentication. However, no discussion as a mutual device authentication including PKI for mobile terminals and external devices is produced. We propose the mutual device authentication architecture including PKI and the revocation checking scheme.

MIDP 2.0[1)] security architecture has a role based access control mechanism for mobile terminals. We discuss the problem of the access control mechanism, and propose an improved access control mechanism in this paper. In PC environments, SELinux[33)] have been proposed. The SELinux has a secure file system that controls access by each process. TCPA[34)] is a hardware-based system to improve an access controll mechanism of PC.

Furthermore, we present an offline personalization scheme called self-delegation that is one efficient scheme of personalization. Online personalization (called On-the-air Service Provisioning) has been discussed in 3GPP2.

3. Design Concept of beyond 3G Mobile Services and Their Threats

We assume that a mobile terminal for new mobile services has higher computational power, a larger display, and some extensible slots, compared with current cellular phones. A design concept of the terminal is that the terminal is based on an open and extensible platform. Namely, it has only default functions at first, and the user can easily extend the functions by adding external devices, such as, a wireless-LAN card, e-commerce card, user identification card, and other application cards, which have standardized common interfaces. The mobile terminals can selectively access several types of mobile networks, by putting on and taking off extended devices. A user can purchase a mo-

Table 1 Threats against beyond 3G mobile terminals.

Malicious Entity	Purpose	Target Entity	Invalid action (example)
User	Invalid use of services	Terminal	(i) Forgery/Alteration of a terminal
		Terminal /Service	(ii) Masquerading
	Denial of service	Service /Terminal	(iii) Invalid operations (i) Forgery/Alteration of a terminal
	Steal secret Info.	Terminal	(iii) Invalid operations
Device Manufacturer	Steal secret Info. Get invalid benefit	User	(iv) Providing invalid terminal and/or devices
	Denial of service	Service	
Service Provider	Steal secret Info. Get invalid benefit	User /Terminal	(v) Masquerading (vi) Providing invalid program
	Denial of service	Terminal /Service	(vi) Providing invalid program (vii) Invalid operation

bile terminal from several shops such as electrical shops. These shops sell the mobile terminal and its extended devices, without installing user information like in existing mobile phones, but leaving the mobile terminal in its default condition. The user makes an agreement with the service provider providing the mobile communication services and other services for their mobile terminal. Users can install or download applications to their mobile terminal. The advantage of the open architecture is to reduce the cost of developing terminals and external devices.

Open architecture, however, is prone to compromise in the system security. This section analyses threats to the mobile terminal and proposes countermeasures against those threats. At first, we define players and their basic roles of the system as follows:

(a) Users: People who have a mobile terminal and make use of mobile services. Malicious users may try to use services by illegal procedures, and may impersonate other users. For example, the malicious user may forge a mobile terminal and/or masquerade as other users. Furthermore, a malicious user may try to steal some secret information from other user's terminals using his mobile terminal or external devices.

(b) Device manufacturers: Organizations that provide mobile terminals and/or external devices. Malicious manufacturers may provide invalid terminals and/or devices to steal a user's secret information, benefit them illegally to sell invalid devices, and prevent them from providing services.

(c) Service providers: Organizations who provide services such as mobile communication, e-commerce, and entertainment services, for mobile terminals. Malicious service providers may provide invalid programs and masquerade as a valid provider for the same purpose as malicious manufacturers. The malicious service providers also impede other service providers.

4. Security Architecture for beyond 3G Mobile Terminals

In the previous section, we discussed the service-level security requirements for beyond 3G mobile services including the provision of the mobile terminal. The threats in **Table 1** can be categorized into four groups.

The first involves invalid devices, which are provided by invalid manufacturers or are forged/altered by malicious users. Secondly, are invalid operations of malicious users or malicious programs provided by malicious service providers. The third groups involves the masquerading of users or service providers. The last group includes invalid programs.

An authentication scheme between devices is required to detect invalid devices. To protect against invalid operations, all environments have to be implemented with an access control mechanism and related system architecture. Masquerading is impossible by authentication of each other in service use. We propose an authentication method considering the properties of future, mobile services. A software verification method is needed to protect against installing invalid programs.

This section proposes new security functions protecting against the mutual device authentication, the access control based on privilege attribute certificates and secure system architecture, personalization and service authentication, and program verification.

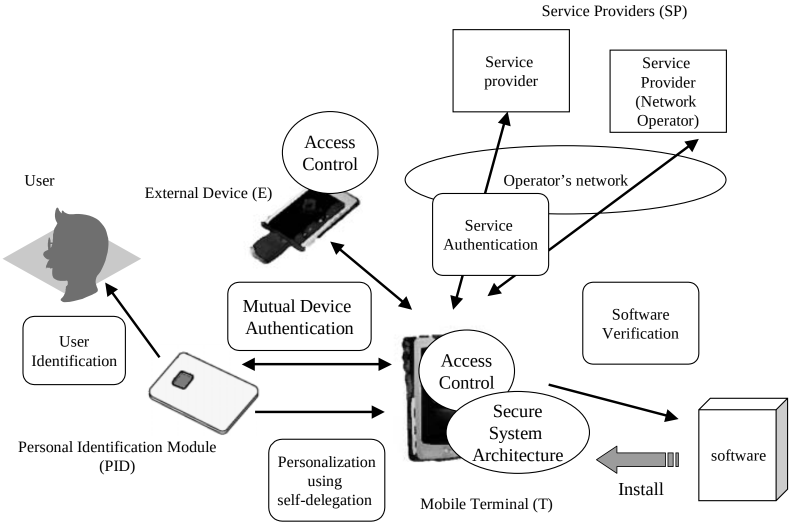

Fig. 1 Secure system architecture model.

4.1 Overview of Proposed Architecture

We propose security architecture for beyond 3G mobile terminals as follows (**Fig. 1**). The architecture consists of four key technologies.

- Mutual device authentication:
  Services for beyond 3G mobile terminals are based on an open architecture for the realization of flexible services, where many service providers and manufacturers exist. We assume terminals, and devices are not trustworthy. Namely, an arbitrary pair of terminals/devices does not trust each other at an initial phase. A mutual device authentication mechanism is used to verify the authenticity of each device, when a new device is attached.
- Access control and secure system architecture:
  The proposed mobile terminals and external devices have an access control mechanism and a secure system architecture, to protect against theft of secret information and illegal use of services. The access control mechanism protects against invalid access, and the secure system architecture protects secret information.
- Personalization, service authentication, and user identification:
  We assume users have some rights to use the services, which are securely stored in a tamper-resistant module called a personal identification device (PID). At first, a user has to personalize his/her terminal when he/she purchases it. The user can transfer the rights to his/her terminal using a self-delegation mechanism. The self-delegation mechanism creates time-limited information and delegates the information to protect primary secret information.
  The delegated information is used for service authentication. The service authentication is a mutual authentication between terminals (users) and service providers, which protects against the use of invalid rights and masquerading as each other. PID requires a personal identification mechanism to protect against invalid use. Therefore, PIDs should have some user identification mechanisms such as PIN (personal identification number) or biometrics technologies[20]. First, to activate PID, a user is requested to identify himself /herself, then he/she can use it.
- Verification of installed programs:
  A mobile terminal has a verification mechanism for programs. As we assume some existing techniques are used for the verification such as a digital signature, we will not discuss its mechanism in detail here.

4.2 Mutual Device Authentication

4.2.1 Basic Authentication Protocols

An initial mutual device authentication protocol is as follows:

$$T \rightarrow E : R$$
$$E \rightarrow T : D(Kpr_{(E)}, R||R'), R', Kpu_{(E)}$$
$$T \rightarrow E : D(Kpr_{(T)}, R'), Kpu_{(T)}$$

where a mobile terminal denotes T and an external device denotes E. $||$ represents a concatenation of data. R and R' indicate random numbers. $D(\cdot)$ indicates digital signature, $D(K,X)$ means that X is digitally signed using a private key K. $Kpr_{(I)}$ represents a private key of an entity I, and Kpu_I means a public key and its certificate (PKC) of an entity I. The public key and private key are defined for each device model. Each entity verifies a PKC, and verifies $D(\cdot)$ using a relative public key. We select algorithms based on discrete logarithm problems (DLP) such as ECDSA[2] for the algorithm $D(\cdot)$. Using the Diffie-Hellman key agreement[3], each entity can compute the shared secret key as follows. The same DLP algorithms as $D(\cdot)$ are selected for the key agreement algorithm, such as ECDH. $Ks_{(I,J)}$ indicates a shared key between I and J. $H(\cdot)$ indicates a hash function such as SHA-1[4].

$$T : Ks_{(T,E)} = H(R||R'||Kpu_{(E)}^{Kpr_{(T)}})$$
$$E : Ks_{(T,E)} = H(R||R'||Kpu_{(T)}^{Kpr_{(E)}})$$

General mutual authentication and key agreement protocols, such as IKE are also applicable for the mutual device authentication. For example, DH key agreement with a digital signature is applicable for the mutual device authentication, to sign public key and temporal information, which can be verified by each other such as time and sequential number. The proposed method is just one of the simple solutions available. (However, the IKE seems to be complex[18] for mobile terminals, indicating that we have to simplify the IKE so that it can be used on mobile terminals.) The shared key is stored in each entity. After the initial authentication, the authentication protocol can be simplified as follows, by using the shared key. $M(\cdot)$ represents a message authentication algorithm such as HMAC[5], and $M(K,X)$ means a message authentication code (MAC) of a message X using a key K.

$$T \rightarrow E : R$$
$$E \rightarrow T : M(Ks_{(T,E)}, R), R'$$
$$T \rightarrow E : M(Ks_{(T,E)}, R')$$

4.2.2 Public Key Infrastructure for the Mutual Device Authentication

Considering the open environment, the PKI (Public Key Infrastructure) model is used to create a trust relationship between strange devices. Verification of PKC in the mutual device authentication protocol is also required to establish a trust relationship. One of the verification methods for mobile environments requests verification to a Validation Authority (VA) using delegated validation protocols[6),7]. VA offers verification of certificates according to the requests of mobile terminals. However, a mobile terminal cannot use an external communication device until the mutual device authentication is successful. Therefore, the terminal has to locally verify the certificate of the external device at the initial phase. We propose three trust domain models between terminal and device manufacturers for the mutual device authentication, where one manufacturer manages one domain and his/her own CA. The first model is a cross certification model. One manufacturer's domain of mobile terminals exchanges cross-domain certificates with the other manufacturers' domain of external devices. Therefore, verification paths of the certificates are as follows:

Verification path of $Kpu_{(E)}$:
$Kpu_{(E)} \triangleright Kpu_{(ET)} \triangleright Kpu_{(Tm)}$

Verification path of $Kpu_{(T)}$:
$Kpu_{(T)} \triangleright Kpu_{(TE)} \triangleright Kpu_{(Em)}$

where $Kpu_{(Em)}$ is a manufacturer's CA certificate of external devices, and $Kpu_{(Tm)}$ is that of mobile terminals. $Kpu_{(ET)}$ indicates the cross-domain certificate from Tm to Em, and $Kpu_{(TE)}$ indicates that from Em to Tm. $A \triangleright B$ means a trust chain that A is authenticated by B. The right most entity of the trust chains is a trust anchor. The lengths of verification paths are 3.

The second model is a hierarchical model. A manufacturer of mobile terminals issued issues to manufacturers of external devices. Verification paths of the certificates are as follows. The lengths of the verification paths are 2 or 3.

Verification path of $Kpu_{(E)}$:
$Kpu_{(E)} \triangleright Kpu_{(Em)} \triangleright Kpu_{(Tm)}$

Verification path of $Kpu_{(T)}$:
$Kpu_{(T)} \triangleright Kpu_{(Tm)}$

The third model is also a hierarchical model using root CA (certificate authority) or bridge CA (BCA). A root CA (or BCA) issues PKC of manufacturers of mobile terminals and manufacturers of external devices. Verification paths of the certificates are as follows. $Kpu_{(CA)}$ is root CA (or BCA) and a trust anchor. The lengths of trust chains are 3.

Verification path of $\boldsymbol{Kpu_{(E)}}$:
$Kpu_{(E)} \triangleright Kpu_{(Em)} \triangleright Kpu_{(CA)}$

Verification path of $\boldsymbol{Kpu_{(T)}}$:
$Kpu_{(T)} \triangleright Kpu_{(Tm)} \triangleright Kpu_{(CA)}$

The second model is better than the other models, considering the lengths of the verification paths. However, in the second model, external devices have to manage many certificates that vary by manufacturers, which provide mobile terminals. The first model also requires management of many cross-domain certificates for mobile terminals and external devices. In the third model, each device manages only certificates of trust anchors (root CA or BCAs). The trust chains can be decreased using a list of trust certificates in the second verification and after. For example, a verification path of the third model is 2 ($Kpu_{(E)} \triangleright Kpu_{(Em)}$), where $Kpu_{(Em)}$ is already verified and listed in a trust certificate list that consists of hash values of trust certificates. Therefore, the third model is suitable for the proposed authentication model.

4.2.3 Verifying Revocation of Certificates

Each certificate has a validity period. However, the certificate may be revoked before it expires. Therefore, a relying entity should validate the status of a certificate to determine whether it has been revoked or not. Normally, CRLs (Certificate Revocation Lists) are used to check the revocation status of certificates[8]. A CRL is a list of the serial numbers of all unexpired certificates that a CA has revoked. CRL distribution gives rise to some problems however. The communication burden of downloading CRL is a serious problem in mobile environments. If a CA has revoked many certificates, the CRL for that CA is large, and transmission of it to mobile terminals consumes significant bandwidth on the network. Moreover, each entity may use only a small part of the information, and the burden of downloading the rest of the large CRL may be wasted. Therefore, suitable revocation mechanisms for the proposed domain model should also be discussed.

To decrease the size of CRLs, some alternative CRL distribution mechanisms are proposed, such as segmented CRLs, over-issued CRLs, delta-CRLs, and sliding window delta-CRLs[9),10]. The Online Certificate Status Protocol (OCSP)[11] is a verification mechanism different from the above mechanisms, and provides a status of requested certificates in real time. An OCSP mechanism is suitable for mutual device authentication, because the number of devices that a user operates will be much smaller than that of revoked devices. However, huge OCSP requests may cause a heavy processing load for an OCSP responder, if all devices use the OCSP. Therefore, decentralized OCSP responders which respond the statuses for OCSP requests should be constructed. The CA sends CRLs to the responders using an efficient distribution mechanism such as the delta-CRLs.

4.2.4 Two-phase Mutual Device Authentication

In mutual device authentication, the mobile terminal may access an OCSP responder through an untrustworthy device. For example, where the mobile terminal authenticates a communication card such as a cellular card, the terminal must use the communication card to check the revocation status of the card's certificate. Therefore, we propose a two-phase authentication.

In the first phase, both entities verify the peer certificate locally. In the second phase, both entities verify the status of the certificate. Each entity can only send an OCSP request after the first phase, and the entity can perform any action after the second phase. To protect against reuse of OCSP response messages, the validities of the messages defined by the "Validity" field should be short or any challenge, which is a random number, should be included in the OCSP response messages. If the mobile terminal and the external devices cannot access to the networks, user action or security policy that is previously defined determines whether the device is trustworthy. The mobile terminal and the external devices try to verify the status, when a network is available.

4.2.5 How to Trust OCSP Responders

OCSP response messages are digitally signed using the private key of the OCSP responder. A CA issues a certificate of the public key of the OCSP responder to guarantee validity of the responders. Therefore, a user can check validity of the message, if the user trusts the CA. If a private key of the responder is compromised, response messages from the responder cannot be trusted. However, the user cannot know the status of the responder's certificate. The trust level of OCSP responders is different from that of CA. The private key of OCSP responders is continuously exposed to risks of compromise, because the key is managed by on-line computers to create response messages[12].

How to trust an OCSP responder is an open issue. One simple solution to solve the above problem is to request several OCSP responders. A user requests a status of a certificate to several responders. We propose the "Majority Decision." Namely, the user judges whether the certificate is revoked or not by summing up the statuses, contained in all of the response messages, and the user picks up some of the trusting messages, and verifies the digital signature of the messages. The following is the detailed mechanism. We do not discuss DoS (Denial of Service) attacks against certificate verification of the devices. We may wish to protect modification of the above algorithm, and this work will be proposed in the future.

"Majority Decision" OCSP

s_i: Status of a requested certificate in message M_i
$s_i \in \{revoked = 1, notrevoked = 0\}$
n: a number of response messages
C: a number of revoked statuses
$C = \sum_n s_i$
M_i: a response message i
Sig_i: a digital signature of a message M_i
k: security parameter, k represents a number of responders whose private key is compromised. $k < n/2$.
$Verify(X)$: Verification of X using the public key of X, which includes verification of public key. $Verify(X) \in \{success = 1, fail = 0\}$

```
STEP1:
  $s_i, C =; 0, k$
  For $i = 1, ..., n$ do
    $C = C + s_i$
    If $C > k$ then the certificate is revoked
STEP2:
  $M_j \in \{M_i | c_i = 1\}$, $Sig_j$
  For $j = 1, \ldots, k + 1$ do
    $Verify(Sig_j)$
    If $Verify(Sig_j) = 0$ then the certificate is
    revoked
STEP3:
  Return the certificate is not revoked
```

4.3 Access Control Based on Privilege Attribute Certificates

4.3.1 Overview of Proposed Access Control Method

We propose an access control mechanism based on privilege attribute certificates (PACs). Generally, access control mechanisms are categorized into three mechanisms, namely, Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role-Based Access Control (RBAC)[15]. The DAC mechanisms are a means of restricting access to information based on the identity of users and/or membership of groups. The owner of information or any resource can make/change its permissions at his discretion. MAC mechanisms assign a security policy to all information and grant a security clearance to each user. In the RBAC mechanism, users or programs have various roles and the security policy can be defined for each role. RBAC provide more flexible access control than DAC and MAC, because the security policy of each role can be defined. However, these access control mechanisms require previously fixing a security policy for all user and all resources. The proposed access control mechanism is based on RBAC and MAC. In the proposed mechanism, security policy is dynamically changed to use the PAC, which defines privileges as security clearances.

A concept of PAC based access control is to design a dynamic creation of a security policy according to changes of the external devices and programs, and to decrease the management cost of security policies using pre-issued PACs. The PAC is a list of available functions to certify an owner's privilege, and it has been used in some access control methods[13),14]. We also apply the PAC to the access control method of the proposed security architecture. An access control module of each entity interprets its own policy and PACs offered by other entities, and dynamically creates new access control lists for each entity. In our service model, a terminal is considered an open platform to provide various services as described in Section 2.

The security policy is different depending on each device and each program and individually defined. Existing technologies such as MIDP 2.0[1] have some access control mechanisms. The mechanisms, however, do not apply to the above model because of the following reasons. A security policy of a mobile terminal is predefined, and it cannot change dynamically. If

Table 2 A Privilege Attribute Certificate (PAC).

Field name	Content
version	Version number
serialNumber	Serial number
issuer	Issuer and issuer URL
subject	Holder name
subjectCertificateURL	Holder's certificate URL
subjectCertificateHash	Hash value of holder's certificate
attribute	Description field of attributes
validity	Validity time
cRLInfo	Information for revocation verification
signature	Signature of the attribute certificate

a security policy is defined for each entity, the total data of the security policies stored in a terminal becomes large. Moreover, the management of the security policies may be incredibly complicated. On the other hand, our mechanism is scalable and flexible, because entities do not manage security policies for other entities but exchange "PAC" including information of access control for each entity. The entity can change dynamically depending on exchanged PAC by each entity.

4.3.2 Procedure of Access Control Method

A procedure of the proposed access control method is shown below:

(i) Exchange Privilege Attribute Certificate: Devices such as mobile terminals and external devices exchange their PACs with each other. Applications installed in a mobile terminal provide their PAC to the mobile terminal. Each entity verifies the PAC using a PKI framework.

(ii) Create Access Control List:
The policy engine parses the PAC, and merges it into a local policy, which is initially defined as a base policy, and dynamically creates a new Access Control List (ACL).

(iii) Control Access:
The access control module controls access operations to their own resources, depending on the dynamically created ACL.

4.3.3 Privilege attribute certificate

An attribute certificates in X.509 Recommendation[8] can be used as the PAC. We have to modify the format of attribute certificates, and remove redundant information in order to decrease the size of the certificates. The PAC includes information as illustrated in **Table 2**. An issuer is a device manufacturer or a service provider that provides services for mobile terminals. For example, if the manufacturer of the mobile devices has issued the PAC including a writing attribute to external devices, the external device can write some critical information stored in the mobile terminals. The subject is the holder name such as the model number of devices, the name of application providers, or users. A subjectCertificateURL and subjectCertificateHash are the URL of the holder's certificate and the hash value of the certificate, respectively. The URL refers to the certificate of an external device, application provider, or user. The attribute certificate is binding to the owner's certificate. An attribute field stores attribute information possessed by the holder. Validity is the expiration date. cRLInfo informs a contact point of a verifier to check the status of the attribute certificate. The signature is the issuer's signature.

4.3.4 Creation of Access Control List

In our method, ACL is dynamically created by the policy engine. An access control rule is generally described as a triplet (subject, object, action)[16],[17]. The subject is a principal who wants to use the object. The object is a resource or function that is used by the subject. The action means operation(s) for the object, which the subject is allowed to perform. We simplify the rule definition as a doublet (subject, object), and the action can be included in the object. The base policy (BP) of the accessed entity is described as $\{(\phi, O_i) \in BP\}$. S_k indicates a subject, and O_i indicates an object. ϕ means "not defined." If the entity should define a special rule for S_k, the rule is described as an additional rule (AR), which is described as $\{(S_k, O_i) \in AR\}$ or $\{(S_k, \neg O_i) \in AR\}$. The $\neg$ O_i means a subject S_k cannot use an object O_i. Privilege in PACs means a set of O_is, which are allowed in S_k. Therefore, the PAC of entity S_k means $\{(S_k, O_i) \in PAC\}$. The creation of the policy engine describes $\overset{PE}{\rightarrow}$, where the left side of the arrow is the input information and the right side of the arrow is the

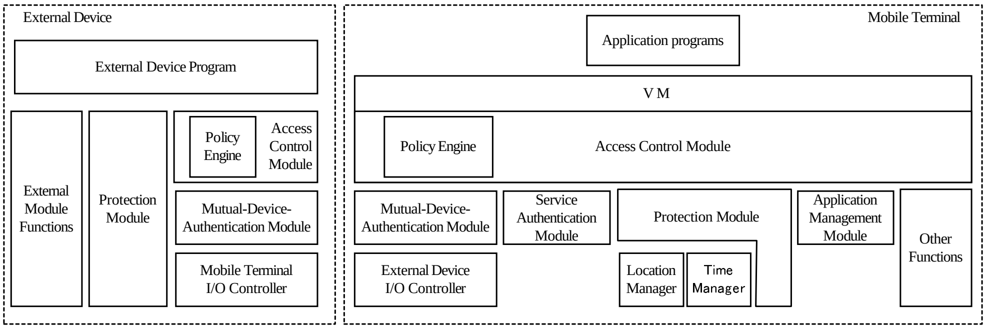

Fig. 2 Secure system architecture model.

output ACL. The ACL for S_k can be defined as a set of $\{(S_k, O_i) \in ACL\}$ $(i = 1, \ldots, n)$. (S_k, ϕ) means the function is not allowed in S_k. Where $\{(\phi, O_i) \in BP\}$, $\{(S_k, O_i) \in AR\}$, and $\{(S_k, O_i) \in PAC\}$ are given;

Policy Creation Algorithm $\forall O_i$ $(i{=}1, \ldots, n)$, S_k,
$\{(S_k, O_i) \mid (S_k, O_i) \in \{\{\{(\phi, O_i) \in BP\} \cup \{(S_k, O_i) \in PAC\}\} \cap \{(S_k, \phi) \in AR)\} \cup \{(S_k, O_i) \in AR\}\} \overset{PE}{\to} \{(S_k, O_i) \mid (S_k, O_i) \in ACL\}$

The ACL for S_K is derived as $\{(S_K, O_i) \in ACL\}$ $(i = 1, \ldots, n)$. Security policy is applied to the ACL, in the order of additional rules, the PAC, and the base policy. In the proposed method, O_is consist of function names that are hierarchically named. For example, the attribute of transferring data using serial I/F is described as $[Output] - [SerialI/F] - [Send/Receive]$. Moreover, time or location limited access controls are also accomplished using time or location information, where the definition is extended as $O_i = [function\ name]\&[time]\&[location]$.

4.3.5 Efficiency of Base Policy Management

In the proposed access control, the mobile terminal manages the base policy and the additional rules. In the existing methods, a mobile terminal has to manage $O(s * o)$ rules, where the number of all external devices is s and the functions of the terminal are o. However, in the proposed method, the mobile terminal manages $O(o + a * o)$ rules, where a is a number of external devices defined as additional rules. A number of external devices that a user may use defines e. Generally, $a < e \ll s$ can be assumed, because a user uses limited external devices, and only defines additional rules for limited external devices. The lack of information $O((s - a) * o)$ is provided by PACs, so that an entity based access control is accomplished. Therefore, the proposed method is more efficient compared with the existing methods, in terms of management of security policies in mobile terminals.

4.4 Secure System Architecture Model

We design a typical system architecture for the mobile terminals as shown in **Fig. 2**. Three important functions are included in the secure system architecture, namely, authentications, access control, and resource protection. External devices also have the same configuration except for the application management module.

4.4.1 Authentication Module

We designed three authentication modules, a mutual device authentication module, a service authentication module, and an application management module. The mutual device authentication module executes the mutual device authentication protocol and manages authentication statuses. The module provides an identifier with accessing entities to the access control module. The module also receives delegated information using self-delegation protocols. The service authentication module automatically executes service authentication protocols, when requested. The application management module authenticates and manages application programs installed in the mobile terminal. The module verifies a digital signature of the programs, when the programs are downloaded or installed locally. The module passes the result and a PAC to the access control modules. The application management module also manages a lifetime of application programs.

4.4.2 Access Control Module

The access control module of the mobile terminal controls any access from programs on the mobile terminal or external devices to programs or functions on the mobile terminal. The module of the mobile terminals is constructed into a Virtual Machine (VM) such as Java VM. The module of the external device controls from the mobile terminal or other devices to an external device program on the external devices. The modules include a policy engine that manages a security policy and creates an ACL from the BP, AR, and PACs. The modules can manage all accesses to resources including security modules.

4.4.3 Protection Module

The protection module protects some critical information such as the private key of mobile terminals, information for authentication, location, time, and root certificates, using cryptographic techniques or tamper-resistant techniques. To protect the critical information, the protection module has the function as secure data storage. One solution that realizes secure data storage is to control low-level file access by software such as operation systems and kernel programs or hardware techniques. Any program that has no privilege for the file access cannot even find the file. SELinux has a secure file system that controls access by each process. Hardware based approaches have been proposed such as TCPA. This approach may be more secure than the software basis. However, additional hardware increases the cost of producing mobile terminals.

The other solution is using cryptographic techniques. Blaze proposed a cryptographic file system for UNIX[35], and N. Provos presented encrypted virtual memory[36]. We consider these techniques are applicable for the protection module. For example, all files that are managed by the protection module, are automatically encrypted by a secret key, and stored as the encrypted data. The key issue to realize the cryptographic file system is management of the secret key. The secret key is a temporary key, and the secret key can be changed randomly by the protection module itself[37]. Therefore, the secret key appears on running memory area and has short lifetimes. However, protecting the running memory (including resume states) is still open issue, even though hardware based approaches have been proposed. We will discuss an actual solution for protecting the memory, and evaluate its implementation in future research.

The module manages the storage area for the critical information. Location information and time information are automatically set in the area by a location manager and time manager, respectively. Other security modules can directly access the protection module. For, example, the access control module can get time and location information from the module. Moreover, only programs permitted by the access control module can use the module. The module also manages the validity of stored data. If stored information has expired, the module erases the information automatically.

4.5 Personalization of the Mobile Terminal and Service Authentication

In the current mobile services, the mobile operator authenticates a subscriber using an identification stored in a mobile terminal, where the identification is newly assigned when a user subscribes. In such a case, the mobile operator does not identify the person but the mobile terminal he owns. For the next generation of mobile services, however, personal identification is needed to provide value added and fine grained services such as mobile commerce, mobile banking etc. One solution is that the mobile terminal has various personal identification mechanisms such as biometrics to activate the mobile terminal. However, the following drawbacks are seen.

- Sensitive information or mechanisms for personal identification are not securely managed on the mobile terminal because the mobile terminal is easily stolen and lost. To satisfy the above requirement, the mobile terminal should have an additional secure element such as an IC chip, which is not cost effective.
- Biometrics mechanisms[20] also require additional user action when authenticated. For example, the user has to put his finger on the CCD sensor to scan the fingerprint. If the mobile operator requires strict authentication, then the user is forced to perform such action every time he has access to various kinds of services.

To solve such problems, we apply a mechanism where the strict authentication is realized even if the mobile terminal is based on an off-the-shelf module and requires no tamper-resistant module. We use a basic concept called self-delegation. Goldreich, et al. proposed the self-delegation method with controlled propagation based on a non-interactive zero-knowledge proof[19]. The scheme, however, cannot be applied to mobile environments, considering the computational cost and communication cost.

We propose a practical approach for mobile environments. The basic idea is as follows: A user stores the information, which relates to strict authentication, into a tamper-resistant module, and the user keeps it in his/her home securely. Time limited authority is delegated into the mobile terminal by communicating with the tamper-resistant module on a local basis. After the delegation, the user can use the remote service by using the mobile terminal within a limited time. More precisely, our proposed mechanism uses both primary and secondary key information. Namely, the primary key information is stored in the tamper-resistant module at first. After the self-delegation, the secondary key information is derived from the primary key information by the tamper-resistant module and securely installed into the mobile terminal. Therefore, the user can be authenticated by using the secondary key in the mobile terminal. If the mobile terminal is stolen, a service provider can also authenticate the user using primary information stored in the tamper-resistant module. Therefore, the user can revoke delegated information.

The self-delegation can be widely applied to any services that strictly authenticate a user, such as e-commerce, identification, and other attractive services. Currently, a user tends to have many IC cards, which are issued by each service provider. Using the self-delegation, a user does not have to carry all his/her IC cards but only carries his/her mobile terminal, which has delegated information from the cards.

Delegated authority has validity, and if expired, the authority is of no use for authentication. The validity is described in Info data. A service provider checks the validity after authentication, and if expired, he/she does not allow the service to be used. The mobile terminal also checks the validity, so that the expired information is automatically deleted. Even if an attacker gets a mobile terminal or delegated information, he/she is only able to use it within a limited term. If the mobile terminal is stolen, the service provider authenticates the user using the primary information, and adds an identifier of delegated information into a revocation list. The revoked mobile terminal cannot use the service. If a contract between the service provider and the user has expired, primary information such as a user ID or a certificate of the user are also added into the revocation list.

4.5.1 Self-delegation Protocol

When the mutual device authentication is successful, the PID transfers delegated information to MT. A delegation protocol is selected according to primary information stored in the PID. U represents a user, and SP represents a service provider. If the primary key is a secret key $Ks_{(U)}$, a secondary secret key is delegated. The delegation protocol is as follows, where K_{auth} is an authenticator (secondary key), Did is a delegation ID which is defined each delegation, Uid is user ID which is provided by each service provider, and $Info$ is additional information such as validity. An attacker cannot trace Uid across service providers, because Uid of the user is different from each provider. Did is computed from a device ID of the mobile terminal and a random number which is generated for each self-delegation protocol, and the protection module securely manages the delegated information with the random number. We assume the device ID cannot be altered. Did is sent and stored to the PID. In the service authentication phase, Did is dynamically computed from the stored random number and the device ID by the service authentication module. Did can be untraceable because it is different from each self-delegation.

Secret key scheme

$$PID : K_{auth} = M(Ks_{(U)}, Did||R||Info)$$
$$PID \rightarrow T : K_{auth}, Uid, Info, R$$

If primary information is a public-private key pair $Kpr_{(U)}$ and $Kpu_{(U)}$, the PID issues a digital ticket and delegates it to T. The ticket is made beforehand as follows.

Ticket scheme

$$PID : R, Info, Ticket$$
$$\text{where, } Ticket = E(Kpu_{(SP)}, R),$$
$$D(Kpr_{(U)}, Info||E(Kpu_{(SP)}, R))$$

PID sends $Ticket$ and related information, when delegation is requested by T. The delegation protocol is as follows. $Ticket$ is computed previously, so that PID only computes K_{auth} in real time.

$$PID : K_{auth} = M(R, Did)$$
$$PID \rightarrow T : K_{auth}, Info, Ticket, Kpr_{(U)}$$

4.5.2 Service Authentication Protocol

A mobile terminal and a service provider authenticate each other using the following protocols, when the user of the terminal tries to use the service. We propose two authentication protocols, which are selected according to delegated information. R' and R'' indicate a random number. $Ks_{(SP)}$ is a master key of SP.

Secret key based protocol

$SP \rightarrow T : R'$
$T \rightarrow SP : Did, Uid, R, Info, R'',$
$\quad M(K_{auth}, R'||Did||Uid||R||Info||R'')$
$SP \rightarrow T : M(K_{auth}, R'')$

SP authenticates T as follows:
$\quad Check\ varidity$
$\quad Check\ revocation\ of\ Did\ and\ Uid$
$\quad M(Ks_{(SP)}, Uid) = Ks_{(U)}$
$\quad M(Ks_{(U)}, Did||R||Info) = K_{auth}$
$\quad Verify\ M(K_{auth}, R'||Did||Uid||R||Info$
$\quad ||R'')$
$\quad T$ authenticates SP verifying $M(K_{auth},$
$\quad R'')$

Ticket based protocol

$SP \rightarrow T : R'$
$T \rightarrow SP : Info, Ticket, Kpu_{(U)}, Did, R'',$
$\quad M(K_{auth}, R'||Info||Ticket||Kpu_{(U)}||Did$
$\quad ||R'')$
$SP \rightarrow T : M(K_{auth}, R'')$

SP authenticates T as follows:
$\quad Check\ varidity$
$\quad Check\ revocation\ of\ Did$
$\quad Verify\ Kpu_{(U)}$
$\quad Verify\ Ticket$
$\quad Decrypt\ E(Kpu_{(SP)}, R) = R$
$\quad M(R, Did) = K_{auth}$
$\quad Verify\ M(K_{auth}, R'||Info||Ticket||$
$\quad Kpu_{(U)}||Did||R'')$

The session key in both schemes is $M(K_{auth}, R'||R'')$. After authentication, all communication is protected using the session key.

4.5.3 Comparison of Secret Key and Ticket Based Schemes

Authentication protocols are classified into two types. One is an authentication scheme using an asymmetric key algorithm; the other is using a symmetric key algorithm. Many services have either an asymmetric or symmetric authentication scheme. Therefore, We design two delegation and authentication protocols. In this subsection, we evaluate the computational cost as **Table 3**, where public key encryption/decryption cost or cost of making/verification of a digital signature is P, a cost of calculating a hash value or calculating a message authentication code is H, and other computation is negligible. Both delegation protocols are lightweight protocols in a delegation phase, if the PID can pre-compute the ticket (which is defined as "Pre" in Table 3) before delegation. Both service authentication protocols can be applied for mobile terminals, because computational costs of a mobile terminal are small in the service authentication. However, a service provider has to compute three public key calculations. Therefore, the secret key scheme is suitable for fast and scalable authentication such as the authentication of network operators.

Table 3 Comparison of two delegation and authentication protocols.

	PID	T	SP
Delegation of a secret key	H	-	-
Delegation of a ticket	Pre:2P+H H	- -	- -
Auth. using a secret key	-	2H	4H
Auth. using a ticket	-	2H	3P+5H

Primary information such as a secret key and public-private key pair is stored into each PID. If one PID stores primary information for each service provider, we have to consider the cost of management of primary key information. The ticket based scheme is better than the secret key scheme in terms of the cost of key management, since a user uses one public-secret key pair to register for service providers. For example, the user may have a smart card such as an identification card, which stores a public-private key pair, and he/she may use it to register service providers as an initialization. We also consider the traceability of the public key. An attacker including malicious service providers try to trace user's activities using the public key. One solution is that the user makes different public-private key pairs for each service provider, even though the cost of key management increases.

4.6 Software Verification

A mobile terminal has a verification mechanism for installed programs. We assume some existing techniques are used for the verification such as a digital signature. For example, the digital signature of the service provider is attached to the program and the application management module checks validity of the signature such as MIDP 2.0 architecture. In the architecture, a PKI technology is used. Considering a simple PKI model, the trust anchor is the trusted third party. This model is the same as in the current services.

5. Discussion

In this section, we discuss the validity and feasibility of the proposed architecture on the following technical requirements.

(A) Mutual device authentication; Current technologies of security functions are designed assuming trustworthiness of all devices. Because, one manufacturer such as an operator or a few trusted manufacturer produce mobile devices in current mobile services. However, many device manufacturers exist assuming beyond 3G architecture is an open architecture. In the proposed device authentication scheme, the initial authentication is based on the public key scheme. The public key scheme is heavier than secret key schemes. It is unfeasible to share a secret key between all manufacturers in the open architecture. Furthermore, the secret key scheme has a problem when managing many secret keys.

Our scheme has two authentication schemes. In the initial authentication, the device authenticates an unspecified number of devices, so that the public key scheme is required. After the initial authentication, the secret key based authentication can be used. Because, the purpose of authentications after the initial authentication are the verification whether the partner device is authenticated previously, and the number of the partner devices that a user uses is limited. Therefore, the key management is feasible. The mutual device authentication verifies that each device is produced by valid manufacturers. The proposed mutual device authentication protects against invalid devices such as (i) and (iv) in Table 1.

Public key authentication schemes require a domain model of PKI. We compared three typical domain model of PKI for the mutual device authentication and confirm that the third model is suitable for the assumed environments. The third model need to construct a root CA. This research result suggests a governmental /non-governmental organization is required, which authorizes device manufacturers.

We also discussed and proposed efficient checking scheme of public key certificates revocation, assuming mobile environments. Furthermore, we presented one solution for the open issue that is how to trust an OCSP responder.

(B) Access control and secure system architecture; In Section 4.3, we proposed the PAC based access control mechanism. The characteristics of the proposed scheme are 1) the security policy for each program can be changed dynamically, 2) The management cost of the security policy in the mobile terminals is low. In beyond 3G mobile environments, many device and programs are attached and/or connected to the mobile terminal. Therefore, a flexible and variable security policy should be required. Furthermore, considering limited resources of mobile terminals, the management cost of the security policy should be small. Current mobile phones have a simple and fixed security policy because configuration of the security policy is very complex.

In the current application platform of the mobile terminal, the application is categorized into a few classes, and a user or service provider cannot change the security policy. For, example, Mobile Execution Environments (MExE) permission framework in 3GPP specifies four security domains. MIDP 2.0 is the next application platform for the mobile phones. The security policy of MIDP 2.0 can be changed, and defined for any entity of several granularities. However, the security policy has to be pre-defined, and management of the policy is still complicated.

The proposed access control mechanism provides flexible access control and can change the security policy dynamically depending on exchanged PAC by each entity. As shown in Section 4.3.5, the proposed scheme is also scalable in terms of management of security policies in mobile terminals. The proposed mechanism controls access to important resources to check the privilege described in the PAC, and protects the important resources against invalid operations. If a malicious service provider provides invalid programs without PAC, the invalid program cannot access any important resources. Therefore, the proposed access control mechanism protects against invalid operations by a malicious user and invalid programs such as (iii) and (vi) in Table 1.

Current mobile terminals have closed environments to maintain trustworthiness of the environments, so that the system architecture could be improved for beyond 3G mobile services, when considering open architecture. Recently, importance of a secure (tamper-proof) component that includes security functions and related information on a mobile device for future services has been discussed. Our design policy adheres fundamentally to the discussion. The method to make the mechanism tamper-proof depends on the implementation environments. We have proposed a typical architecture model.

The important design policy of the architecture model is separating security modules from other modules and protecting the security modules and its related critical information stored in a mobile terminal. In addition, we consider two security requirements that the function connecting to external devices should include. These are, an authentication mechanism of the partner devices, and time and location information protected against the alteration by malicious users. Secure time and location information are useful information for several services. In a mobile terminal of the proposed model, any programs without security modules can be run on a virtual machine that controls access to security modules and other local resources using an access control module. Each program and each external device is separated to security domains that are principals of a security policy. The access controller can control all access from programs according to the security policy which is dynamically created from a base policy and PACs. Therefore, security modules are protected. Furthermore, the protection modules protect critical information. The proposed model protects not only invalid operations and alteration of critical information such as (i), (iii), and (iv), but also the integrity of security functions.

One open issue is how to protect the running memory. Little research has been proposed[28)] to protect the running memory, but practical solutions are under consideration. We will design a detailed mechanism of the proposed secure system architecture and evaluate its implementation in future research.

(C) Personalization and service authentication; A cellular phone is currently assumed as a secure device. Information for service use is stored into the cellular directly, and a special device is required to store the information. If a user changes to new cellular, he/she has to go to shops that have a special device for personalization. One of the serious problems in terms of management of users and terminals is how to personalize a new terminal.

For example, in 3GPP2, Over-the-air Service Provisioning scheme (OTA-SP) is currently discussed. OTA-SP is an online personalization scheme using wireless networks, and information for authentication is downloaded from a server to a new mobile terminal. However, a secure channel between the server and the mobile terminal has to be constructed.

The other solution is using a removable SIM or UIM. SIM and UIM are tamper-proof devices and can be attached to and removed from the cellular. If the information is stored into the SIM or UIM, a user can transfer the information by removing and attaching the device. However, this scheme also has a problem. As stated in the discussion in 4.5, if the cellular is stolen or lost, the information is also lost or exploited.

The proposed scheme provides an offline personalization scheme, because all (primary) information is stored in the user's PID. If a user purchases a new mobile terminal, the user only delegates his/her own authority to a mobile terminal on local basis, using his/her PID. Therefore, the proposed self-delegation is secure and convenient. In our service authentication, we use two identifiers *Uid* and *Did*. However, an attacker cannot trace user action across service providers, because this information is different between each service provider. Self-delegation is an efficient authentication framework in terms of not only security but also user management.

The self-delegation provides strict authentication between a user and a service provider, and protects against masquerading such as (ii) and (v) in Table 1. A Kerberos based scheme proposed by A. Fox, et al. is one of the feasible solutions for indirect authentication in mobile environments. However, direct and real-time authentication is required by sensitive services such as e-commerce. The scheme proposed here can provide direct and strict authentication securely.

In Section 4.5.3, we showed that the computational costs for service authentication of a secret key based scheme and a public key based scheme are low in a mobile terminal. Especially, comparing with general public key authentication, the ticket based authentication scheme is required lower computational cost. Our self-delegation is applicable for the current authentication protocols such as SSL[45)] to modify the protocols slightly.

(D) Software verification; Software verification protects against invalid programs such as (vi) in Table 1. Security requirements for software verification can be satisfied by current technologies such as digital signature. As the purpose is detecting malicious programs provided by malicious service providers, we do not consider that valid service providers provide malicious programs. If we have to consider the above threat, we will have to discuss checking method of the program in real-time. The access control mechanism may be effective for the malicious programs. however, this problem is an open issue.

(E) Others; The other security requirements is the security between service providers such as impeding other service providers, which is (vii) in Table 1. These threats can be prevented using current security technologies for PCs.

From the above consideration, we conclude a proposed architecture satisfies the security requirements discussed in Section 3, and the architecture is feasible when compared with current technologies in assumed beyond 3G mobile services.

6. Conclusion

In this paper, we designed security architecture for beyond 3G mobile terminals, assuming future mobile environments. Firstly, we analyzed the security requirements of beyond 3G mobile services, and then proposed a new security architecture.

Our proposed architecture consisted of four security functions, which exhibited mutual device authentication, namely, PAC based access control and secure system architecture, self-delegation, and software verification. The mutual device authentication detects forged or invalid devices. The PAC based access control realized dynamic creation of a security policy according to changing functions. The access control mechanism and the secure system architecture protects against alteration of terminals and several invalid operations. The mechanisms also protect secret information that is stored in mobile terminals. The self-delegation makes mobile services more convenient and secure. A malicious user and malicious service provider cannot masquerade as other users or service providers, because they authenticate each other using service authentication protocols. Using the verification process of the programs precludes invalid programs.

The proposed architecture satisfies the security requirements discussed in Section 3, and the architecture is feasible when compared with current technologies in assumed beyond 3G mobile services. We believe our research is important for designing future mobile services. As for our future research, we will implement our security architecture using current mobile terminals, and evaluate its feasibility and scalability.

References

1) Sun Microsystems.: Mobile Information Device Profile (MIDP) 2.0, http://jcp.org/jsr/detail/118.jsp.

2) ANSI X9.62, Public Key Cryptography for the Financial Services Industry: Elliptic Curve Digital Signature Algorithm (ECDSA) (1999).

3) Diffie, W. and Hellman, M.: New Directions in Cryptography, *IEEE Trans. Inf. Theory*, IT-22, 6 (1976).

4) NIST, FIPS 180-1.: Secure Hash Standard (Apr. 1995).

5) Krawczyk, H., Bellare, M. and Canetti, R.: “HMAC” Keyed-Hashing for Message Authentication, RFC 2104 (1997).

6) Pinkas, D., et al.: Delegated Path Validation and Delegated Path Discovery Protocol Requirements, RFC3379 (2002).

7) Malpani, A., Housley, R. and Freeman, T.: Simple Certificate Validation Protocol, *IETF*, Internet draft (2002).

8) Draft revised ITU-T Recommendation X.509. ISO/IEC 9594-8: Public-Key and Attribute Certificate Frameworks (2000).

9) Cooper, D.A.: A Model of Certificate Revocation, *Proc. ACSAC '99* (Dec. 1999).

10) Cooper, D.A.: A More Efficient Use of Delta-CRLs, *Proc.2000 IEEE Symposium on Security and Privacy* (May 2000).

11) Myers, M., Ankney, R., Malpani, A., Galperin, S. and Adams, C.: Online Certificate Status Protocol—OCSP. Technical Report RFC2560, IETF (June 1999).

12) Micali, S.: NOVOMODO: Scalable Certificate Validation and Simplified PKI Management, *Proc. 1st Annual PKI Research Workshop* (2002).

13) European Computer Manufacturers' Association: Authentication and Privilege Attribute Security Application with related Key Distribution Functions, ECMA Standard 219 (Dec. 1994).

14) Jansen, W.: A Privilege Management Scheme for Mobile Agent Systems, *1st International Workshop on Security of Mobile Multiagent Systems, Autonomous Agents Conference* (2001).

15) The Open Web Application Security Project (OWASP).: A Guide to Building Secure Web Applications, Chapter 8, Access Control and Authorization, http://www.cgisecurity.com/owasp/ (2003).

16) Woo, T.Y.C. and Lam, S.S.: Authorization in Distributed Systems: A Formal Approach, *Proc. IEEE Symposium on Security and Privacy* (1992).

17) Jajodia, S., Samarati, P. and Subrahmanian, V.S.: A logical language for expressing authorizations, *Proc. IEEE Symposium on Security and Privacy* (May 1997).

18) Perlman, R. and Kaufman, C.: Analysis of the IPSec Key Exchange Standard, WETICE2001 (2001).

19) Goldreich, O., Pftzmann, B. and Rivest, R.L.: Self-Delegation with Controlled Propagation - or- What If You Lose Your Laptop, *Proc. Crypto 98*, Springer LNCS, Vol.1462, pp.153–168 (1998).

20) Zhang, D.D.: *Authenticated Biometrics Technologies and System*, Kluwer Academic Publishers (2000).

21) Harada, H., Kuroda, M., Morikawa, H., Wakana, H. and Adachi, F.: The Overview of The New Generation Mobile Communication System and The Role of Software Defined Radio Technology, *IEICE Trans. Comm.*, Vol.E86-B, No.12, (Dec. 2003).

22) Yoshida, M., et al.: A Secure Service Architecture for Beyond 3G Wireless Network, *Proc. WPMC 2003*, Vol.2, pp.579–583 (Oct. 2003).

23) Lacoste, G.: SEMPER; A Security Framework for the Global Electronic Marketplace, *Comtec—The magazine for telecommunications technology*, Vol.77, No.9, pp.56–63 (1999).

24) Foster, I., Kesselman, C., Tsudik, G. and Tuecke, S.: A Security Architecture for Computational Grid, *Proc. ACM Conference on Computer and Communication Security*, pp.88–92 (1999).

25) Fox, A. and Gribble, S.: Security On the Move: Indirect Authentication Using Kerberos, *Mobile Computing and Networking*, pp.155–164 (1996).

26) Kerberos.: The Network Authentication Protocol, http://web.mit.edu/kerberos/www/.

27) Michael, L.B., Mihaljevic, M.J., Haruyama, S. and Kohno, R.: Security Issues for Software Defined Radio: Design of a Secure Download System, *IEICE Trans. Comm.*, Vol.E85-B, No.12 (Dec. 2002).

28) Inamura, Y.: Cryptographic Memory System, *IPSJ SIG. Technical Report*, 2003-CSEC-22, pp.121–126 (2003).

29) Leiner, B.M., Ruth, R.J. and Sastry, A.R.: Goals and Challenges of the DARPA GloMo Program, *IEEE Personal Communications*, pp.34–43 (Dec. 1996).

30) The Official Bluetooth Membership Site.: Bluetooth Specification, http://www.bluetooth.org/spec/.

31) The 3rd Generation Partnership Project (3GPP), http://www.3gpp.org/.

32) The 3rd Generation Partnership Project 2 (3GPP2), http://www.3gpp2.org/.

33) National Security Agency.: Security-Enhansement Linux, http://www.nsa.gov/selinux/.

34) Trusted Computing Platform Alliance, http://www.trustedcomputing.org/home/.

35) Blaze, M.: A Cryptographic File System for Unix, *Proc. first ACM Conference on Communication and Computing Security* (*CCS*) (Nov. 1993).

36) Provos, N.: Encrypting Virtual Memory, *Proc. 9th USENIX Security Symposium* (Aug. 2000).

37) Blaze, M.: Key Management in an Encrypting File System, *Proc. 1994 USENIX Summer Technical Conference* (June 1994).

38) Phoenix Technologies Ltd.: Phoenix FirstAuthority, http://www.phoenix.com/.

39) Mobile IT Forum, http://www.mitf.org/.

40) WAP Forum, http://www.wapforum.org/.

41) Mobile electronic Transaction (MeT), http://www.mobiletransaction.org/.

42) WAP Forum.: Wireless Transpotation Layer Security, WAP Specification (2001).

43) Yumiba, H. and Yabusaki, M.: Mobile Service History and Future, *IEICE Trans. Comm.*, Vol.E85-B, No.10, pp.1878–1886 (2002).

44) Yamao, Y., Umeda, N., Otsu, T. and Nakajima, N.: Fourth Generation Mobile Communications System—Issues Regarding Radio System Technologies, *IEICE Trans. Comm.*, Vol.J83-B, No.10, pp.1364–1373, 2000.

45) Freier, A., Karlton, P. and Kocher, P.: The SSL Protocol Version 3.0, http://home.netscape.com/eng/ssl3/draft302.txt.

(Received December 2, 2003)
(Accepted June 8, 2004)

Shinsaku Kiyomoto received his B.E. in Engineering Sciences, and M.E. in Materials Science, from Tsukuba University, Japan, in 1998 and 2000 respectively. He joined KDD (now KDDI) and has been engaged in the research on stream cipher, cryptographic protocol, and mobile security. He is currently a research engineer of the Security Lab. in KDDI R & D Laboratories Inc. He received the Young Engineer Award from IEICE in 2004. He is a member of JPS, and IEICE.

Toshiaki Tanaka received the B.E. and M.E. degrees in communication engineering from Osaka University, Japan, in 1984 and 1986 respectively. He joined KDD (now KDDI) and has been engaged in the research on cryptographic protocol, mobile security, digital rights management, and intrusion detection techniques. He is currently a senior manager of the Security Lab. in KDDI R & D Laboratories Inc. He is a member of IEICE.

Mariko Yoshida received the B.S. degree from the Tokyo Institute of Technology, Japan, in 1994. She joined Mitsubishi Electric Corporation, Japan in 1994. Since then, she was engaged in business computer systems, mobile applications. Her current research interests includes wireless security and wireless applications.

Masahiro Kuroda received the M.E. degree in systems science from the Tokyo Institute of Technology, Japan, in 1980, the M.S. degree in computer science from University of California, Santa Barbara, CA, in 1989, and received the Ph.D. degree in computer science from Shizuoka University, Japan, in 2000. He joined Mitsubishi Electric Corporation, Kamakura, Japan in 1980. Since then, he was engaged in OS/network developments, mobile network computing R & D, and cellular Java standardizations. He is currently working as a group leader at National Institute of Information and Communications Technology, Yokosuka, Japan. His current research interests includes wireless network, wireless security, mobile systems, and next generation wireless systems architecture. He is a member of the IPSJ and IEEE Computer Society.