A New Delegation-Based Authentication Protocol for Use in Portable Communication Systems

Wei-Bin Lee and Chang-Kuo Yeh

***Abstract*—Portable communication systems (PCSs) provide a convenient means of communication; however, many problems arise relating to data security, user privacy, computational load, and communicational efficiency. To provide solutions for these problems, we introduce the concept of delegation into the wireless communication system. This new model makes our scheme an especially valuable improvement to portable communication systems.**

***Index Terms*—Cryptography, delegation, portable communication systems (PCSs), proxy signature.**

I. Introduction

PORTABLE communication systems (PCSs) do not require any physical circuits between subscriber and service provider. Radio waves being transmitted in space make it easy for anyone to eavesdrop on the contents of communication, so there are more security and privacy threats than with wire-line communication systems.

A secure communication system should possess four major features: secrecy, authenticity, integrity, and nonrepudiation. There is no doubt that the security mechanisms of PCS can benefit from the use of cryptography. There are two kinds of cryptosystem: the secret-key system and the public-key system. In the secret-key system, a single key can be used to encrypt and decrypt a message. In the public-key system, each person gets a pair of keys—a public key and a private key. The public key is assumed to be public while the private key is kept secret. In such a system, all communication involves only public keys, and no private key is ever transmitted or shared [14]. The most important development from work on public-key systems is the digital signature that can provide nonrepudiation service.

The speed of encryption/decryption of the secret-key system is faster than that of the public-key system, but it cannot provide the nonrepudiation feature. The public-key system possesses all of the four features, but it requires many more complicated calculations than do secret-key systems, and the public key must be changed periodically. It is not a problem to add extra hardware equipment to wire-line communication systems to support complicated computations, so recently public-key systems such as Rivest–Shamir–Adleman (RSA) [12] have been widely used in many commercial products. However, due to the hardware limitations of the portable handset in PCS, the mobile station cannot support encryption/decryption computations that are too complicated. Furthermore, the periodical changing of the public key is another problem because it is not practical in PCS. Thus, the cost-benefit analysis may prevent the current PCS such as Global System for Mobile Communication (GSM) [11] from adopting the public-key system in the first place. Clearly, if performance is not a major concern in PCS, RSA could be appropriate; otherwise, one needs to consider different alternatives.

According to the above analysis, we know that providing security services and making them work efficiently in the wireless environment is difficult. To provide a solution, we introduce the concept of delegation into our scheme. This new scheme cannot only provide security benefits such as user identity privacy (user location privacy is not our concern), nonrepudiation, and mutual authentication between user and service provider, but it can also provide efficient key management service. The new scheme does not increase the heavy computational loads for mobile stations and does not decrease the communicational efficiency.

In Section II, we briefly introduce the previous research results related to PCS and analyze the advantages and disadvantages of these results. In Section III, we introduce the concept of delegation and explain how the concept is applied to our model. Our scheme is described in detail in Section IV. The discussions and comparisons between our protocol and the others are stated in Section V. Finally, conclusions are given in Section VI.

II. Review of the Proposed Protocols

A. GSM Protocol

Fig. 1 illustrates GSM, the most widely used PCS system authentication protocol. Here, RAND is a random number generated by home location register (HLR), and A3 and A8 are one-way hash functions used to generate SRES and K_c, respectively. A5 is an encryption/decryption algorithm. K_i is a secret key shared by the mobile station (MS) and HLR. Initially, a user registers in HLR with a unique identity, international mobile subscriber identity (IMSI) and obtains the secret key K_i from HLR. The temporary mobile subscriber identity (TMSI) allocated by visited location register (VLR) is used to protect the user's IMSI from being exposed. The authentication process is carried out as follows:

Step 1) MS transmits IMSI to VLR.
Step 2) VLR passes IMSI to HLR.
Step 3) HLR examines whether IMSI exists. A set of authentication information (RAND, SRES, K_c) is sent back to VLR if IMSI exists; otherwise, the request is rejected.
Step 4) VLR forwards RAND received from HLR to MS.

Manuscript received November 28, 2001; revised August 27, 2002; accepted October 23, 2003. The editor coordinating the review of this paper and approving it for publication is Z. J. Haas.

The authors are with the Department of Information Engineering, Feng Chia University, Taichung 407, Taiwan, R.O.C. (e-mail: lwb@iecs.fcu.edu.tw).

Digital Object Identifier 10.1109/TWC.2004.840220

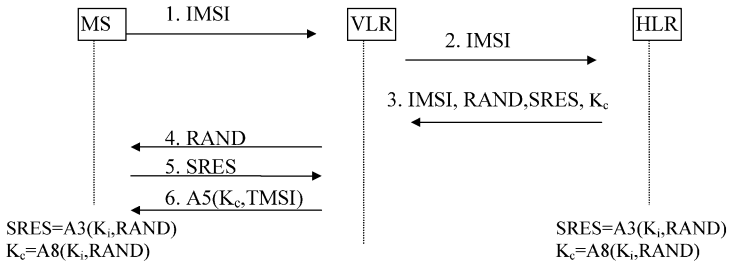

Fig. 1. GSM authentication protocol.

Step 5) MS uses K_i and RAND to generate SRES and sends it to VLR.

STep 6) VLR generates TMSI for MS if SRES is correct.

According to the above, we find that HLR and MS share a secret key, VLR uses the challenge/response technique to authenticate MS, and there is no security protection between VLR and HLR. The advantages of GSM are: 1) the secret-key system does not dramatically increase computational loads for the mobile station and 2) key management is simple since the long-term secret key K_i is permanently kept in the SIM card. Therefore, no periodic key updating would be necessary. However, GSM has the following problems: 1) it cannot provide the nonrepudiation feature, so a dishonest user may repudiate the calls he has made; 2) it cannot protect user identity privacy because the IMSI should be clearly transmitted where anyone can intercept it; 3) there is no security protection between VLR and HLR, so attackers can easily intercept the sensitive information transferred between HLR and VLR; and 4) MS cannot authenticate VLR.

B. MGSM Protocols

Many protocols try to enhance security or promote efficiency while maintaining the original architecture of GSM. For example, security protection is added between VLR and HLR in [5] to prevent an intruder from intercepting sensitive information transferred between HLR and VLR; in [1], HLR directly authenticates MS, and, in [4], the multicasting technique is applied, so they eliminate the steps of the authentication process. These protocols use the secret-key system to provide security services. Consequently, the limitations of the secret key system make nonrepudiation, for example, impossible. These protocols use the challenge/response technique to authenticate MS, but no security mechanism is provided for MS to authenticate VLR. We call these protocols Maintain the Architecture of GSM (MGSM) since they are still based on the secret-key system and use the challenge/response technique.

C. Public-Key System Protocols

In contrast, the public-key system-based protocol [2], [3], [8], [10] can provide more security features such as nonrepudiation and mutual authentication. However, these protocols suffer some drawbacks: 1) the public-key system requires much more complex computation than does the secret-key system, so it becomes a bottleneck for MS since there is limited computational power for its battery; 2) communicational efficiency might lessen if MS must always retrieve the most recent certificate revocation list (CRL) whenever a new certificate is received from the issuing certificate authority (CA); 3) the public key of MS should be updated periodically; however, this is impractical in the real world. This increases the difficulty of key management; 4) the real identity of MS is revealed because the public key is necessary for verification.

III. Concept of Delegation

In order to introduce our scheme, it is necessary to briefly describe the concept of delegation. Our method is inspired by the Proxy signature, which is the delegation of the power to sign messages [6], [9]. The following example illustrates the concept of delegation.

In a business corporation, the manager uses his private key to sign a document and his staff can verify the document based on his public key. If the manager cannot sign a document because he is away on business, he can delegate his signature authority to his trustworthy assistant to sign the document without giving the assistant his private key. His staff verifies that the document is still based on his public key.

This authorized signature technique is called proxy signature. This new type of digital signature gave us the inspiration for our model. The assistant is authorized to sign the document when the manager is absent, but the staff can still use the manager's public key to verify the document. This implies that, even if the staff can distinguish the signature of the assistant from that of the manager, the staff cannot know the real identity of the assistant. The manager cannot deny the signature if a dispute arises. Of course, the manager should have the ability to identify a dishonest assistant.

The previous example can be applied to our model. HLR gives MS the power to sign and VLR can verify the signature based on the public key of HLR. VLR can only verify the legality of MS but it does not know the real identity of MS. The model can provide user privacy and nonrepudiation features, and key management is easier than in the pure public-key system model since only the public key of HLR should be managed. Fig. 2 illustrates the concept of our model.

IV. Our Protocol

There are three phases in our model: initialization, registration, and authentication. First we briefly describe our model and employ proxy signature [6], [9] and hash chain techniques [13] to illustrate the model in Section IV-D.

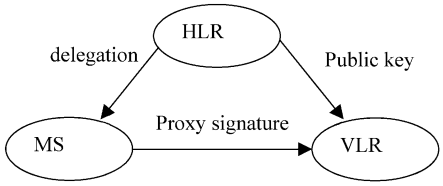

Fig. 2. Concept of our model.

A. Initialization

HLR generates its private key and the corresponding public key according to the underlying public-key cryptosystem.

B. Registration

When MS registers in HLR, HLR creates a proxy pair key that contains a pseudonym of MS and sends it to MS over a secure channel. The pseudonym is used to represent the real identity of MS in the network. The relation between the pair key and the corresponding real identity of MS are protected in a secure database located in HLR. No one except HLR can obtain any information about the real identity of MS.

C. Authentication

Authentication phase is divided into two parts: on-line authentication and off-line authentication. In the on-line authentication phase, the process requires that VLR must connect to HLR whenever MS demands authentication. However, without connecting to HLR to save authentication time and provide fault tolerance, off-line authentication is performed by VLR locally according to the parameters obtained from HLR in advance. Note that the first authentication request must be performed on-line, and the subsequent authentication requests can be continually performed off-line.

1) On-Line Authentication:

Step 1) MS sends his pseudonym to VLR.
Step 2) VLR makes responses to MS by sending a nonce and his identity.
Step 3) MS signs the nonce and the identity of VLR and then delivers the signature to VLR.
Step 4) VLR verifies the signature based on HLR's public key. If the verification is achieved, VLR sends a session key request that contains a pseudonym of MS to HLR.
Step 5) HLR checks whether the pseudonym of MS is legal according to the corresponding real identity located in his secret database. If MS is legal, HLR generates the session key based on the secret key shared with MS and then sends the key and an authenticator to VLR.
Step 6) VLR verifies the authenticator to check whether HLR is legal and then passes the authenticator to MS. When MS receives the information, he verifies the authenticator to check whether VLR is legal. If VLR passes the check, the authentication process is complete.

MS can also generate the same session key based on the same shared secret key, so the confidentiality of data transmission between VLR and MS can be guaranteed.

2) Off-Line Authentication: The authenticator leaves MS and VLR in possession of a secure token that can be used for subsequent authentication, avoiding the need to contact HLR repeatedly. MS uses the current session key to encrypt the secure token embedded in the authenticator and a new generated token and then sends the encrypted message to VLR. When VLR decrypts the message, it verifies the secure token is correct. The newly generated token is used for next authentication and the new session key is generated by using the old secure token and the current session key.

D. Implementation

We employ the proxy signature technique to illustrate how on-line authentication protocol works and employ the well-known hash chain technique to illustrate how off-line authentication protocol works.

Some notations should be explained here. $X \rightarrow \mathrm{Y} : \mathrm{Z}$ denotes that a sender X sends a message Z to a receiver $\mathrm{Y}, h(\,)$ denotes a one way hash function, $n1 \,||\, n2$ denotes a concatenation of data $n1$ and $n2, IDV$, and IDH denotes the identity of VLR and HLR, respectively. K_{HV} denotes the secret key shared by HLR and VLR, and $[M]N$ denotes a message M encrypted by key N.

1) Initialization: HLR generates parameters p (a 512-b prime number), q (a 160-b prime factor of $p-1$), and $g = h^{(p-1)/q} \bmod p$, where $h \in [1, p-1]$. Then HLR selects x (a number less than q) as the private key and calculates $\nu = g^x \bmod p$ as the corresponding public key certificated by a trusted Certificate Authority.

2) Registration HLR → MS : (σ, K): When MS subscribes to his home system (HLR), HLR will generate a random number k, compute $K = g^k \bmod p$, and further calculate $\sigma = x + kK \bmod q$, where σ is the secret key shared by MS and HLR and K is the pseudonym of MS. The relationship between the key pair (σ, K) and the real identity of MS must be protected by HLR in a secure database. After that, MS will obtain a SIM card with the key pair (σ, K) from HLR.

3) Authentication: Fig. 3 shows the process of the on-line authentication and Fig. 4 shows the ith process of the off-line authentication.

a) On-line authentication:

Step 1) $\mathbf{MS} \rightarrow \mathbf{VLR} : \boldsymbol{K}$.
Step 2) $\mathbf{VLR} \rightarrow \mathbf{MS} : \boldsymbol{n2}, \boldsymbol{IDV}$.
Step 3) $\mathbf{MS} \rightarrow \mathbf{VLR} : \boldsymbol{r}, \boldsymbol{s}, \boldsymbol{K}, \boldsymbol{n1}, \boldsymbol{IDH}, \boldsymbol{IDV}$.
Step 4) $\mathbf{VLR} \rightarrow \mathbf{HLR} : [\boldsymbol{n1} \,||\, \boldsymbol{n2} \,||\, \boldsymbol{K}]\boldsymbol{K}_{\mathrm{HV}}, \boldsymbol{IDH}, \boldsymbol{IDV}$.
Step 5) $\mathbf{HLR} \rightarrow \mathbf{VLR} : [[\boldsymbol{n1}, \boldsymbol{m_1}]\sigma \,||\, \boldsymbol{n2} \,||\, \boldsymbol{l} \,||\, \boldsymbol{C_1}]\boldsymbol{K}_{\mathrm{HV}}, \mathbf{IDH}, \mathbf{IDV}$.
Step 6) $\mathbf{VLR} \rightarrow \mathbf{MS} : [\boldsymbol{n1}, \boldsymbol{m_1}]\boldsymbol{\sigma}, \boldsymbol{IDV}$.

In Step 1), when an MS roams into a new VLR, he sends his pseudonym K to VLR.

In Step 2), VLR randomly generates $n2$ (a number less than q) and then sends $n2$ and his identity to MS.

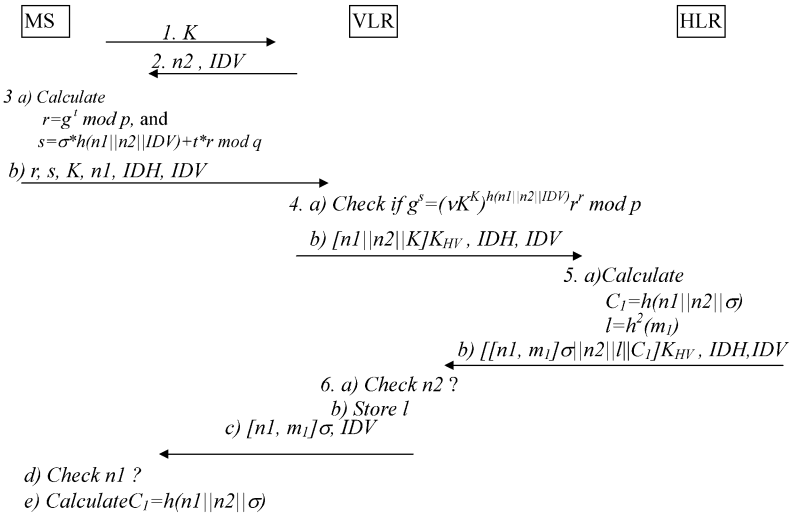

Fig. 3. On-line authentication process.

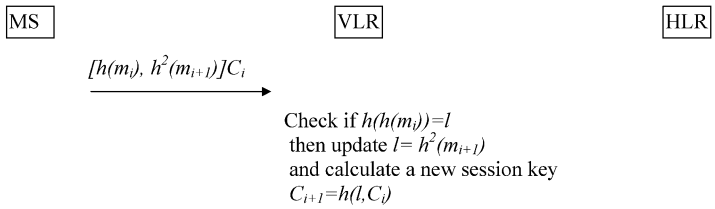

Fig. 4. ith off-line authentication process.

In Step 3), MS signs $n2$ and *IDV* received from VLR as follows:

$$r = g^t \bmod p \tag{1}$$

$$s = \sigma * h(n1 \,||\, n2 \,||\, \mathrm{IDV}) + t * r \bmod q \tag{2}$$

where t and $n1$ are random numbers less than q generated by MS. MS now sends $r, s, K, n1, IDH, IDV$ to VLR.

In Step 4), VLR verifies the received information by checking whether

$$g^s = (\nu K^K)^{h(n1\,||\,n2\,||\,IDV)} r^r \bmod p. \tag{3}$$

If the equation does not hold, the request is rejected; otherwise, the valid request implies that MS is authorized by HLR. VLR then sends $[n1 \,||\, n2 \,||\, K]K_{\mathrm{HV}}, IDH, IDV$ to HLR.

In Step 5), HLR decrypts the message $[n1 \,||\, n2 \,||\, K]K_{\mathrm{HV}}$ to get the pseudonym K and further search the corresponding σ in its secure database according to K. If not found, the authentication process is rejected; otherwise $C_1 = h(n1 \,||\, n2 \,||\, \sigma)$ and $l = h^2(m_1)$ are computed, where m_1 is a random number and $h^2(m_1)$ means that m_1 is hashed twice. After that, HLR sends

$$[[n1, m_1]\sigma \,||\, n2 \,||\, l \,||\, C_1]K_{\mathrm{HV}}, IDH, \text{and } IDV \textit{ back to } \text{VLR}.$$

In Step 6), VLR decrypts the message $[[n1, m_1]\sigma \,||\, n2 \,||\, l \,||\, C_1]K_{\mathrm{HV}}$ to get $[n1, m_1]\sigma, n2, l$ and C_1, where C_1 is the current session key used by VLR and MS, and l is prepared for VLR to verify MS for the off-line authentication. VLR checks whether $n2$ is the same as what was sent previously to HLR to prevent a replay attack. If $n2$ passes the check, C_1 and l are valid, and VLR forward $[n1, m_1]\sigma$ and IDV to MS. When MS receives the information, he decrypts the message $[n1, m_1]\sigma$ to get $n1$ and m_1 and checks whether $n1$ is the same as what he previously sent to VLR. If $n1$ passes the check, VLR is authenticated.

Because MS has the knowledge to compute the same session key $C_1 = h(n1 \,||\, n2 \,||\, \sigma)$, the confidentiality of data transmission between VLR and MS can be guaranteed.

b) Off-line authentication: In on-line authentication protocol, HLR sends a random number m_1 to MS and calculates $l = h^2(m_1)$ for VLR. The two parameters are key points for the off-line authentication protocol. For security considerations, it is not reasonable to perform off-line authentication all the time while the first on-line authentication is finished. Hence, a predefined constant n should be set to a reasonable constraint on the times to do off-line authentication. If MS wants to request the ith off-line authentication protocol, the process is illustrated as follows:

$$\mathbf{MS} \rightarrow \mathbf{VLR} : [\boldsymbol{h(m_i), h^2(m_{i+1})}]\boldsymbol{C_i}.$$

MS calculates $h(m_i)$ and $h^2(m_{i+1})$ and sends $[h(m_i), h^2(m_{i+1})]C_i$ to VLR, where m_i and m_{i+1} are random numbers generated by MS except m_1 that was generated by HLR and securely sent to MS in on-line authentication process. Furthermore, a new on-line authentication should be taken if $i + 1 = n$.

When VLR receives the message, he decrypts the message $[h(m_i), h^2(m_{i+1})]C_i$ to get $h(m_i)$ and $h^2(m_{i+1})$ and check whether $h(h(m_i))$ is equal to l. If not equal, the request is rejected; otherwise, the valid request implies MS is authenticated by VLR. Then l is updated by $h^2(m_{i+1})$ for the next off-

line authentication, and computes the new session key $C_{i+1} = h(l, C_i)$. Because MS has the knowledge to compute the same session key $C_{i+1} = h(l, C_i)$, the confidentiality of data transmission between VLR and MS can be guaranteed. Finally, i is added by one.

The off-line authentication processes have been performed until i is equal to n, and then the on-line authentication process should be started again if MS asks another authentication request.

V. Discussion and Comparison

It is reasonable to assume that HLR is trustworthy because we must register it with our private information to obtain the service. Hence, it is acceptable that the key pair (σ, K) and the real identity of MS are protected in a secure database located in HLR. Based on this hypothesis, we discuss and compare our protocol with the others described in Section II to show that our protocol is better.

A. User Identity Privacy

GSM, MGSM and public-key based protocols provide weak user identity privacy since MS must deliver his real identity to the network for authentication.

In our protocol, the real identity of MS is never transmitted over the entire network for authentication purposes. Because we use pseudonym K generated by HLR in the registration phase to represent the identity of MS in the network, no one except HLR can obtain any information about the identity of MS. Even VLR can only verify the legality of MS based on the public key of HLR, that is, (3). Equation (3) discloses nothing about the identity of MS but only implies that MS is authorized by HLR. Hence, our scheme provides stronger user identity privacy than GSM, MGSM and public-key based protocols.

B. Nonrepudiation

GSM and MGSM are based on the secret-key system, so they cannot provide the feature of nonrepudiation.

No doubt, public-key based systems can greatly benefit by the nonrepudiation feature of the public-key cryptosystem.

In our scheme, each MS gets a different pair key (σ, K) from HLR in the registration phase. The key implies the authorization from HLR. This authorization makes VLR transfer his trust in HLR to the requested legal pseudonym MS. Because only HLR has the ability to authorize MS to sign on his behalf, HLR cannot deny this in the event a disputation occurs. Of course, HLR has the ability to identify the misused MS. Thus, our scheme can also provide the feature of nonrepudiation.

C. Mutual Authentication Between MS and VLR

GSM and MGSM only provide the mechanism for VLR to authenticate MS, and the public-key based protocols can provide mutual authentication services.

In our scheme, it is easy for VLR to authenticate MS by verifying the proxy signature made by MS using the proxy key authorized by HLR. If (3) holds, MS is authenticated by VLR. On the other hand, MS can authenticate VLR by decrypting the message $[n1, m_1]\sigma$ received in Step 6) of the on-line authentication phase to get $n1$ and checking whether $n1$ is the same as what he sent to VLR in Step 3) of on-line authentication phase. Because HLR is trustworthy, only the legal VLR can decrypt the message $[[n1, m_1]\sigma \,||\, n2 \,||\, l \,||\, C_1]K_{\mathrm{HV}}$ received from HLR to get the correct $[n1, m_1]\sigma$. There is no way for an attacker to pretend to be a legal MS or VLR. Besides, without knowing the secret keys σ and K_{HV}, impersonating HLR is impossible. Thus, our protocol can provide mutual authentication service between MS and VLR.

Furthermore, MS gets a proxy pair key (σ, K) from HLR over a secure channel in the registration phase. The relation between the pair key and the corresponding real identity of the MS are protected in a secure database located in the HLR. In our scheme, MS uses the proxy key to sign and VLR is responsible to verify the MS according to (3) $g^s = (\nu K^K)^{h(n1\,||\,n2\,||\,\mathrm{IDV})} r^r \bmod p$. If a MS-A acquires MS-B's pseudonym K and impersonates B to ask authentication request, (3) cannot hold since MS A has no idea about B's proxy key σ. Therefore, this impersonating attack cannot succeed.

In the off-line authentication process, MS generates the message $[h(m_i), h^2(m_{i+1})]C_i$ and sends it to VLR. VLR decrypts the message to get $h^2(m_{i+1})$. It is very difficult to compute $h(m_{i+1})$ according to $h^2(m_{i+1})$, since $h(\,)$ is a one way hash function which is relatively easy to compute, but significantly harder to reverse [14]. If any attacker tries to replay this message to pass the authentication process, he cannot succeed since VLR will find out that the value $h(h(m_i))$ does not equal to l which is updated to $h^2(m_{i+1})$. If any attacker tries to forge this message to pass the authentication process, he cannot succeed since the message is encrypted by a session key C_i which is different from time to time and only known to MS and VLR. For security considerations, it is not reasonable to do off-line authentication all the time. Hence, a predefined constant n should be set to a reasonable constraint on the times to do off-line authentication.

According to the discussions of Sections V-A, V-B, and V-C, we find that the security of our protocol is based on the parameters $\sigma, K_{\mathrm{HV}}, n1, n2, C_i$ and $h(\,)$. Thus, in order to successfully pretend to be a legal mobile user or service provider, an attacker must forge some sensitive data to pass the authentication process. Fortunately, these attacks cannot work since all sensitive data is protected by σ, K_{HV} or C_i, which is unknown to attackers. The nonces $n1$ and $n2$ can prevent a replay attack because they are changed from time to time. The usage of a nonce guarantees the receipt of a fresh message.

D. Key Management

In public-key-system-based protocols, the verification of MS is based on the public key of MS. However, it is not easy to do so in practice. Because there are many mobile users in the system, the complexity of the public-key infrastructure (PKI) will be introduced into such protocols.

In our protocol, HLR authorizes MS to sign the message and VLR is merely needed to verify MS based on the public key of HLR. The number of HLR is much less than that of MS, so

TABLE I
NUMBERS OF COMPUTATIONS OF MS FOR THE PROPOSED PROTOCOLS

	hash function	secret-key computation (encryption/decryption)	public-key computation (signature/verification)
GSM	2(A3,A8)	1(A5)	0
MGSM	1(A8)	1(A5)	0
Public-key based scheme	0	0	2
Our scheme(on-line)	1	1	1*
Our scheme(off-line)	3	1	0

*: Can be reduced to 1 multiplication due to pre-computation

the complexity of PKI is dramatically reduced. Besides, theoretically, the key of HLR must be more strictly defined and protected than that of MS, and it should be a long-term key that can be used without being frequently updated. Key management will become easier than in the protocols based on public-key systems.

The key management of GSM and MGSM is also easy since secret key K_{i} is a long-term key kept permanently in the SIM card.

If HLR has to change its public/private key pair for some security reason, he should generate a new proxy key pair for each user and send the key to the corresponding user securely. This situation is the same as when CA changes its public/private key pair [16]; the old certificates, which CA signed before, should be collected back and destroyed, and the new certificates must be generated for the users. There seems to be no good solution to avoid this ugly situation. However, compared to the original public-key-based protocols, the key management of our protocol is much easier since the number of HLR's public key is much less than that of MS's in public-key-system-based protocols and the public key of HLR is not necessarily updated frequently.

E. Computational Load

Table I shows the numbers of computations of MS for all proposed protocols.

In GSM, MS needs two hash functions (A3 and A8) and one encryption/decryption (A5) of the secret-key system.

In MGSM [1], one hash function is saved since A3 can be computed off-line so the computational load would be a little lighter than GSM.

In public-key based schemes, two signatures/verifications of the public-key system are needed for MS.

In our protocol, one signature/verification of the public-key system, one encryption/decryption of the secret-key system, and one hash function are needed for MS. As we know that the speed of hash function is approximately 100 times faster than that of the encryption/decryption of the secret-key system and the speed of encryption/decryption of the secret-key system is about 100 times faster than that of the signature/verification of the public-key system [14]. Hence, the computational load of our protocol is lighter than public-key-based systems but heavier than GSM and MGSM. However, precomputed technology can be employed to our scheme to reduce the computational load. For example, to sign a message, MS should compute $r = g^t \bmod p$, and $s = \sigma * h(n1 \,||\, n2 \,||\, \mathrm{VLR}) + t * r \bmod q$. Since the most time-consuming parts r and $t * r \bmod q$ have nothing to do with the signed message, so MS can prepare them in advance for later authentication. Once MS asks for service, he retrieves the values r and $t * r \bmod q$ stored in the SIM card and further calculates s. Based on this precomputed technology, the computational load can be further reduced to one multiplication.

TABLE II
NUMBERS OF STEPS TO COMPLETE THE ENTIRE AUTHENTICATION PROCESS FOR THE PROPOSED PROTOCOLS

GSM	6 or 1*
MGSM	2
Public-key based scheme	≧6
Our scheme	6 or 1*

*: for off-line authentication

After the on-line authentication process is performed, the off-line authentication process will be implemented subsequently if MS asks another authentication request. In such situation, only three hash functions and one encryption of secret-key system are necessary. Fortunately, these are all not time-consuming computations so the computational load is very low for MS.

F. Communicational Load

In GSM, six steps are required to complete the entire authentication process (two between HLR and VLR, four between MS and VLR). GSM can also handle off-line user authentication [5], so if off-line authentication is performed, only one step is needed.

MGSM [1], designed to improve the communicational efficiency of GSM, needs only two steps.

In public-key based schemes, mutual authentication service between MS and VLR needs two steps. And MS/VLR should both take two steps to retrieve the public key of VLR/MS from the CA. At least six steps are involved to complete the entire authentication process for the public-key based schemes.

In our protocol, six steps are also needed to complete the entire authentication process. Hence, our protocol doesn't decrease communicational efficiency. If off-line authentication is performed, more communicational load will be decreased since only one step is needed.

Except for MGSM, six steps are necessary for the other proposed schemes. However, in public-key-based schemes, the

TABLE III
COMPARISON OF ALL PROPOSED PROTOCOLS

	Secret-key based scheme		Public-key based scheme	Our scheme
	GSM	MGSM		
Identity privacy	no	no	no	yes
Non-repudiation	no	no	yes	yes
Mutual authentication (between MS and VLR)	no	no	yes	yes
Easy key management	yes	yes	no	yes
Low computational load	yes	yes	no	yes
Good communicational efficiency	yes	yes	no	yes

extra cost for retrieving the most recent CRL to guarantee the correctness of the public key is also a cost for MS and VLR. This complexity might make the communicational efficiency of public-key-based protocols worse than GSM, MGSM and our protocol. Table II shows the number of steps to complete the entire authentication process for the proposed protocols.

Although our protocol has the same number of steps as the GSM protocol, the size of the messages is larger than the other proposed protocols. The probability of transmission error may be higher than the other protocols. Thus, a specific error correction algorithm should be included to guarantee accuracy of data transmission since not even one bit error can happen in any cryptographic protocol. For example, even parity is used for error correction in GSM. However, the problem of error correction is not our major concern in this paper, so we omit the discussion here and interested reader may refer to [17] for details.

G. Storage Capacity

Storage capacity should be taken into account when designing security protocols for mobile network environments since the mobile equipment has limited storage capacity. Considering the example we take in Section IV, the mobile station should store the parameters $p, q, K = g^k \bmod p, \sigma = x + kK \bmod q, n1, n2, t, IDV, r = g^t \bmod p$ and $s = \sigma * h(n1 \,||\, n2 \,||\, IDV) + t * r \bmod q$, where p is a 512-b prime number, q is a 160-b prime factor of $p-1, n1, n2$, and t are numbers less than q, and the length of IDV is 32 b, which is the same as the IMSI in GSM. Therefore, the total length of $(q, \sigma, n1, n2, t, s, p, K, r, \mathrm{IDV})$ is $160 * 6 + 512 * 3 + 32 = 2528$ b $= 316$ bytes.

The currently used SIM card consists of 16 k bytes of ROM, 256 bytes of RAM, and 8 k bytes of electrically erasable programmable ROM (EEPROM) [15]. Especially, EEPROM which contains subscription specific data such as IMSI and K_i is used for the nonvolatile memory. In summary, the capacity of EEPROM is large enough to accommodate the above parameters of our scheme.

Table III summarizes results relevant to evaluating PCS. This table illustrates the comparisons based on the service provided by the surveyed schemes. According to the discussion and analysis in Section V, our scheme is overall superior to all of the other schemes.

VI. CONCLUSION

In this paper, we propose a delegation-based authentication protocol to provide solutions to the problems of PCS. Proxy signature is the major technique used in our protocol. We compare our protocol with the other proposed protocols to show that the concept of delegation offers real benefits by providing more data security and user privacy, and it does not result in increased computational loads for mobile stations. Overall, our scheme is reasonably efficient.

ACKNOWLEDGMENT

The would like to thank the anonymous referees and the editor for his valuable suggestions that have resulted in the improvement of the correctness and readability of the paper. Their comments regarding our manuscript were extremely helpful to us in preparing a clearer version.

REFERENCES

[1] K. Al-Tawill, A. Akrami, and H. Youssef, "A new authentication protocol for GSM networks," in *Proc. 23rd Annu. IEEE Conf. Local Comput. Networks*, 1998, pp. 21–30.

[2] M. Aydos, B. Sunar, and C. K. Koc, "An elliptic curve cryptography based authentication and key agreement protocol for wireless communication," in *Proc. 2nd Int. Workshop Discrete Algorithms Meth. Mobile Comput. Commun.*, Dallas, TX, Oct. 30, 1998, pp. 1–12.

[3] M. J. Beller, L. F. Chang, and Y. Yacobi, "Privacy and authentication on a portable communications system," *IEEE J. Sel. Areas Commun.*, vol. 11, pp. 821–829, Aug. 1993.

[4] M. S. Hwang, Y. L. Tang, and C. C. Lee, "An efficient authentication protocol for GSM networks," in *Proc. EUROCOMM Inform. Syst. Enhanced Public Safety Security*, 2000, pp. 326–329.

[5] C. H. Lee, M. S. Hwang, and W. P. Yang, "Enhanced privacy and authentication for the GSM," *Wireless Networks*, vol. 5, pp. 231–243, 1999.

[6] W. B. Lee and C. Y. Chang, "Efficient proxy-protected proxy signature scheme based on discrete logarithm," in *Proc. 10th Conf. Inform. Security*, Taiwan, May 2000, pp. 4–7.

[7] H. Y. Lin, "Security and authentication in PCS," *Comput. Elect. Eng.*, vol. 25, no. 4, pp. 225–248, 1999.

[8] C. C. Lo and Y. J. Chen, "Secure communication mechanisms for GSM networks," *IEEE Trans. Consum. Electron.*, vol. 45, pp. 1074–1080, Nov. 1999.

[9] M. Mambo, K. Usuda, and E. Okamoto, "Delegation of the power to sign messages," *IEICE Trans. Fundamentals*, vol. E79-A, no. 9, pp. 1338–1353, Sep. 1996.

[10] J. H. Park and S. B. Lim, "Key distribution for secure VSAT satellite communications," *IEEE Trans. Broadcast.*, vol. 44, pp. 274–277, Sep. 1998.

[11] M. Rahnema, "Overview of the GSM system and protocol architecture," *IEEE Commun. Mag.*, pp. 92–100, Apr. 1993.

[12] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signature and public-key cryptosystem," in *Commun. ACM*, vol. 21, Feb. 1978, pp. 120–126.

[13] L. Lamport, "Password authentication with insecure communication," in *Commun. ACM*, vol. 24, 1981, pp. 770–772.

[14] RSA Laboratories' Frequently Asked Questions About Today's Cryptography, V4.0 [Online]. Available: http://www.rsasecurity.com/rsalabs/faq/

[15] European Digital Cellular Telecommunications System (Phase 2); Specification of the Subscriber Identity Module—Mobile Equipment (SIM—ME) Interface [Online]. Available: http://www.scia.org/knowledgebase/aboutSmartCards/specs.html

[16] WIDE ROOT CA Key Change Information [Online]. Available: http://www.wide.ad.jp/wg/moca/wide_root_key_change-e.html

[17] Characterization of Magnetic Recording Systems: A Practical Approach Written by Alexander Taratorin [Online]. Available: http://www.guzik.com/solutions/prmlbook/book.htm

Wei-Bin Lee received the B.S. degree from Chung-Yuan Christian University, Chungli, Taiwan, in 1991 and the M.S. degree in computer science and information engineering the Ph.D. degree from National Chung Cheng University, Chiayi, Taiwan, in 1993 and 1997, respectively.

He is currently an Associate Professor with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan. His research interests currently include information security, cryptography, computer communication, and digital watermarking.

Chang-Kuo Yeh received the B.S. degree from National Taiwan University, Taipei, in 1985 and the M.S. degree in computer information science from New Jersey Institute of Technology, Newark. He is currently working toward the Ph.D. degree in information engineering and computer science from Feng Chia University, Taichung, Taiwan.