Designing Authentication Protocols for Third Generation Mobile Communication Systems*

SHU-MIN CHENG, SHIUHPYNG SHIEH, WEN-HER YANG,
FU-YUAN LEE AND JIA-NING LUO
Department of Computer Science and Information Engineering
National Chiao Tung University
Hsinchu, 300 Taiwan
E-mail: ssp@csie.nctu.edu.tw

Security is considered an important issue for mobile communication systems. In particular, the design of authentication mechanisms has received considerable research interest recently. However, most of the current authentication schemes for mobile systems only have simple security functions and usually have some weaknesses, such as leakage of user identities and high update overhead of temporary identities. Moreover, these schemes cannot fulfill the security requirements specified in third generation mobile systems (IMT-2000, UMTS). In this paper, we propose a secure and flexible authentication framework for mobile communication systems. In the proposed framework, service providers can dynamically choose authentication mechanisms without the cooperation of network operators in visited domains. Based on the new framework, a secure authentication protocol is proposed. The proposed protocol can satisfy the security requirements of third generation mobile systems and is secure against network attacks, such as the replay attack and substitution attack. In short, our approach is secure and practical such that it can satisfy the security requirements of third generation mobile communication systems.

Keywords: network security, flexible authentication framework, authentication protocol, mobile networks, 3G communication systems

1. INTRODUCTION

With the development of communication and computer technologies, wireless network devices now provide people with a level of mobility that enables them to communicate with others anytime and everywhere. The combination of mobility and networking has led to the development of a whole new class of very interesting applications, but has also led to a whole new set of technical problems. One of the most challenging problems introduced by mobile networking is security. User mobility increases the risk of illegal users masquerading as legal users, and radio channels have become more vulnerable to eavesdroppers. Therefore, a secure and efficient authentication mechanism is especially needed for mobile communication systems. Current mobile communication systems, e.g., GSM and IS-41, provide simple security services, such as user authentication and traffic confidentiality. However, only mobile users are authenticated in these systems. Moreover,

Received October 7, 2002; revised June 27 & December 10, 2003; accepted March 3, 2004.
Communicated by Randy Y. C. Chow.
* This paper is sponsored in part by Institute for Information Industry, Lee and MTI Center for Internetworking Research (Global Crossing), Ministry of Education, and National Science Council.

in both systems, the confidentiality of messages is only guaranteed on the radio channel, rather than on the full path between the two communication participants.

Recently, several advanced mobile communication systems have become the standards for third-generation mobile communication systems. Such systems are currently being standardized by the International Telecommunications Union (ITU) [6] and European Telecommunications Standards Institute (ETSI). The systems are designed to provide access to a wide range of services, such as audio, video, speech, multimedia data and billing services. Various services operating on the hybrid mobile networks face new security issues. A number of security requirements for IMT-2000 [1, 7, 9, 10] have been identified with reference to the type of service, the way of network or service access, radio interface, mobile terminal, user identity module, network operation, and security management. Security requirements identified by previous work include *mutual authentication*, *anonymity*, *confidentiality*, *end-to-end security*, and *data integrity* [6, 7]. (Please also refer to Menezes *et al.* [14] for formal definitions of these terms.)

Some previous schemes have focused on the design of inter-domain authentication schemes, such as Kerberos [15, 21], Windows NT inter-domain access control systems and RADIUS [20]. However, all of these schemes focus on authentication issues related to conventional wired networks rather than mobile communication systems. For mobile communication systems, several authentication schemes have been presented in the literature. Theses schemes are designed to enhance the security of mobile communication systems. However, none of them can fulfill the security requirements of third generation mobile systems. In particular, those presented in [2, 4, 5, 11-13, 16] were not designed based on third generation mobile systems and may incur much computational overhead. Although [3, 17, 18] presented authentication mechanisms for third generation mobile systems, they did not address other security issues, such as end-to-end security, anonymity and confidentiality issues.

In 1997, the International Telecommunications Union (ITU) proposed three candidate authentication schemes [8] for IMT-2000 systems. The first one is based on the use of symmetric key cryptosystems and a challenge-response exchange. The second authentication mechanism is based on the unilateral use of a digital signature scheme and a challenge-response exchange. The third mechanism is also based on the use of a digital signature scheme, in which public key certificates and timestamps are combined to provide user identity confidentiality and unilateral entity authentication in a single mechanism. All three mechanisms provide only some security features and have some weaknesses. The first scheme requires too many authentication messages and does not ensure end-to-end security. In addition, it simply assumes that the channel between the network operator and service provider is secure, so messages transmitted through the channel are vulnerable. The second and third schemes do not provide mutual authentication or end-to-end security, and they incur high computation costs. Moreover, a common weakness of the three schemes is that the authentication mechanism is fixed such that the network operators of visited domains must be involved in the authentication procedure between roaming users and home service providers. Another authentication method [22] for UMTS systems [19] has also been proposed, in which service providers are allowed to choose among authentication schemes dynamically. However, this approach requires too many authentication messages, and network operators in visited domains still need to cooperate in the authentication procedure.

In this paper, a secure and flexible authentication framework for mobile communication systems is proposed. This new framework provides flexibility in authentication, which enables service providers to freely choose among authentication mechanisms without the need for cooperation among network operators in other network domains. This property gives service providers the flexibility to develop proprietary authentication protocols and dynamically adjust their own security policy. Based on this new authentication framework, a secure authentication protocol is also proposed. The proposed protocol can satisfy the security requirements of third generation mobile communication systems, such as mutual authentication, user anonymity, and end-to-end security. The proposed protocol is also secure against network attacks, such as the replay attack and substitution attack.

This paper is organized as follows. In section 2, an authentication framework is presented. Based on this authentication framework, a new authentication protocol is proposed in section 3. Security analysis and a comparison of the proposed authentication protocol with previous schemes will be discussed in section 4. Finally, conclusions are drawn in section 5.

2. AUTHENTICATION FRAMEWORK

In third generation mobile systems, many emerging services, such as the World Wide Web, stock quotes, e-mail account and multimedia, can be access through wireless link. When a mobile user roams far from his home domain and wants to access these services, he may intend to use the servers in a visited domain instead of the ones in his home domain. For example, suppose a roaming user wants to retrieve home pages from a web site; he must first connect to an Internet gateway to access the Internet. Certainly, he can use the Internet gateway in his home domain. However, if the Internet gateway he has connected to is located in a visited domain, he may benefit from better performance and lower connection charges because he is using a local connection to the Internet gateway.

To acquire mobile services in visited domains, mobile users must be authenticated. Normally, the service provider in a visited domain is unable to solely perform the authentication procedure without any prior knowledge of the roaming user; hence, the visited domain requires the participation of the home domain to authenticate the user. In fact, the service provider in the visited domain may simply forward the authentication request to the home domain and check the reply to see if the user has been successfully authenticated. In this way, the role that the service provider in the visited domain plays is more like that of an authentication proxy. From this point of view, the authentication server in the home domain can freely and dynamically choose the most appropriate authentication mechanism without the cooperation of the service provider in the visited domain. This makes the authentication framework flexible in terms of the dynamic selection of authentication mechanisms.

Here, we present a flexible authentication framework for mobile systems, where authentication servers have the capability to determine which authentication mechanism will be used for each authentication request. Basically, the proposed authentication framework consists of the following three parts:

- *C* is a set of mobile clients, which request network services and need to be authenticated. Mobile clients that may be mobile handsets, notebooks, PDAs, or other mobile entities. They are required to register in the authentication servers of home domains and may roam over mobile communication systems.
- *S* is a set of authentication servers, which are network hosts in the service provider and are responsible for authenticating *C*. In each network domain, there is always an authentication server in charge of registration and authentication. This server is trusted by the mobile clients and contains information about registered clients, including public and private data.
- *P* is a set of authentication proxies, which forward authentication requests from *C* to *S*, then wait for authentication responses and forward them back. An authentication proxy acts as an intermediator and has no ability to authenticate mobile clients. It is not involved in the authentication procedure between mobile clients and the authentication server. All it has to do is forward authentication messages and obtain the results.

Note that each client *c* is associated with an authentication server *s*. Client *c* registers with an authentication server *s*. If *c* is in his home domain, *s* acts as the authentication proxy and authentication server when *c* asks for authentication. That is, when *c* ask for authentication, *c* sends an authentication request to *s*. Now, *s* acts as the authentication proxy. Since *s* is able to authenticate *c*, *s* performs the function of the authentication server.

In this framework, the authentication request and response message flow between the three parties is defined as follows, where the authentication proxy *p* and authentication server *s* are associated with a client *c*:

- A $Request_c$ is a message generated by the client *c* when *c* asks for authentication. A $Request_c$ sent from *c* to *p* mainly contains the identity and authentication server information of *c* and authentication mechanisms.
- A *Forwarded* $Request_c$ is a message generated by the proxy *p*, and it consists of a $Request_c$, a proxy ID, and so on. After receiving the $Request_c$ from the client *c*, the proxy *p* sends the *Forwarded* $Request_c$ to the corresponding authentication server *s*.
- A $Response_c$ is a message generated by the authentication server *s* after the client *c* has been authenticated. A $Response_c$ contains *c*'s information and the result of authentication, which is sent to the proxy *p*.
- A *Forwarded* $Response_c$ is a message sent from the proxy *p* to the client *c*. After receiving a $Response_c$, *p* generates a *Forwarded* $Response_c$ which contains a $Response_c$ and other related information to be shared between the proxy *p* and client *c*.

The authentication framework is applicable to various services in mobile communication systems. For example, when the client requests a call setup service, the authentication proxy is the network operator in the visited domain, and the authentication server is the client's home service provider. If the requested service is access to the Internet, then the authentication proxy will be the Internet gateway, such as the WAP [23] gateway in current mobile systems.

2.1 The Operation Modes

In our framework, the authentication server maintains the profiles and privileges of registered clients. Thus, only the client's home authentication server has the ability to initially authenticate the client. Another entity, the authentication proxy, is mainly responsible for forwarding the client's authentication request to the authentication server. In the proposed framework, after initial authentication has been performed, the authentication proxy is then capable of authenticating the client when subsequent authentication is required. That is, the proposed authentication framework contains two operation modes for initial and subsequent authentication.

Initial Authentication

When the client *c* leaves his home domain and roams to a visited domain, the initial authentication shown in Fig. 1 is performed among the three parties. First, the request message *Request$_c$* is generated by the client *c* and sent to the authentication proxy *p* in the visited domain. Since the authentication proxy is unable to authenticate the client *c* by itself, it generates a *Forwarded Request$_c$* containing the *Request$_c$* and forwards it to the designated authentication server *s* in *c*'s home domain. The verification procedure is performed by the authentication server *s*, and a response message *Response$_c$* is generated corresponding to the authentication result. The authentication proxy forwards the *Forwarded Response$_c$* containing the *Response$_c$* to the client and decides whether or not to provide the service to the client according to the authentication result. Here, the authentication proxy caches some authentication information, which can be used in subsequent authentication. The response message *Response$_c$* lets the client *c* know whether the authentication was successful or not. After the initial authentication, both the proxy *p* and client *c* obtain the authentication result from the authentication server and share some secret information.

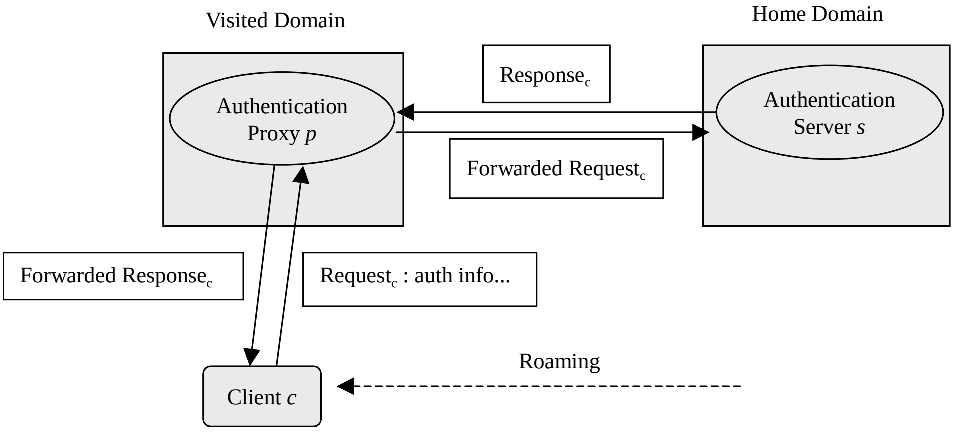


Fig. 1. Initial authentication.

Subsequent Authentication

After initial authentication, the authentication proxy *p* has the ability to authenticate the client *c* in subsequent communication. If the client *c* remains in the same visited domain and requests services, then the user should ask for subsequent authentication. Client *c* similarly generates an authentication request message $Request_c$. The $Request_c$ should contain the information shared between the client *c* and proxy *p*; the proxy *p* then use this information to authenticate the client *c*. As mentioned above, the proxy *p* has cached information needed to authenticate client *c*.

After authenticating the client, the proxy *p* sends a response message $Response_c$, which contains the authentication result, to the client *c*. The client *c* receives the response message $Response_c$ and learns whether the authentication was successful or not.

2.2 Message Format

To make the data flow in the proposed authentication framework clearer, the format of the message flow in the framework is explained in the following. There are two kinds of authentication messages: request messages and response messages. Request messages are sent from the client who needs to be authenticated by the authentication server. Therefore, the messages should contain the user's related information that the server can use to authenticate the user. The authentication server information is also included so that the authentication proxy can send the message to the correct server when the message is received. The rest of the request message consists of the authentication token. The authentication token is used by the authentication server to perform authentication, and it contains two fields. One field is a list of mechanism candidates, and the other is an authenticator, which is the mechanism content produced by the user using the selected authentication mechanism. The mechanism candidates are proposed by the user and then chosen by the authentication server. The user first indicates which mechanism is to be used by the default. If the server agrees, then the authentication server will continue to execute the subsequent authentication procedure. Otherwise, the authentication server will select a new authentication mechanism and notify the client. Then, the authentication procedure will restart. The proposed framework is capable of performing multiple authentications in one request message. As depicted in Fig. 2, an authentication request may contain several authentication tokens for different authentication servers. Hence, the number of authentication messages is reduced.

The authentication server sends response messages to the client. The response messages contain user and the authentication server information, and response tokens. Each response token consists of the authentication state, designated mechanism, and authenticator. The authentication state represents the result of authentication, such as "accept" or "reject." The authentication protocol that the server selects to perform this authentication is represented in the field of the designated mechanism. Also, the authenticator field is the mechanism content produced by the server using the selected authentication mechanism.

2.3 Analysis

Although the case presented in section 2.1 is that in which the client roams to a visited

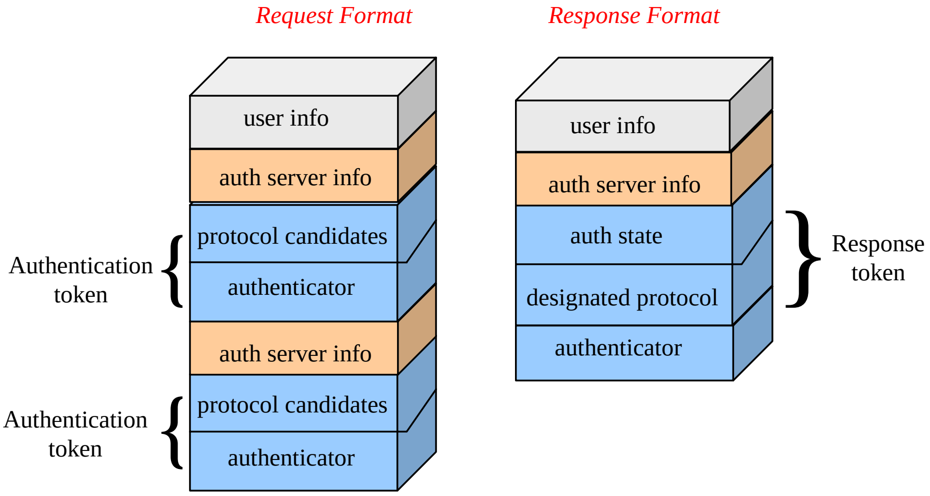


Fig. 2. Authentication message format.

domain, the proposed framework can also be applied to the case in which the client resides in his home domain. In this case, the proxy can be the same as the authentication server. The client can send the request directly to the authentication server, and the server can send the response directly to the client. Since the proxy is transparent to the client and the server knows that it can authenticate the client by itself, it is not necessary to modify the authentication procedure.

The mechanism used to perform authentication is proposed by the client and selected by the authentication server. They can periodically change authentication mechanisms, or they can use a different mechanism for each authentication. Therefore, the authentication framework provides flexibility in terms of the dynamic selection of authentication mechanisms. In addition, to ensure privacy of user information, the authentication token is encrypted so that no network entities, excluding the mobile client and its authentication server, can access its content. Note that the cryptographic algorithm used to protect the authentication token must be secure against common known attacks, such as the known ciphertext attack. Moreover, since the proxy does not participate in the authentication procedure between the authentication server and the client, it need not know which authentication mechanism is being used between the client and the authentication server. The proxy can only obtain the result of authentication, and decide whether to providing services to the client or not, depending on the result.

As depicted in Fig. 3, the three entities of our authentication framework, the authentication server, proxy and client, are the service provider, network operator and mobile user, respectively. When a mobile user roams to a visited domain, he may request services from the network operator of the visited domain. Hence, the mobile user is the client, and the network operator is the proxy in the authentication framework. In the following, we will use the WAP service to illustrate how the authentication framework works. (The WAP service, which is the Internet service for mobile systems, provides web contents and advanced services to cellular subscribers.) As the requested service from the

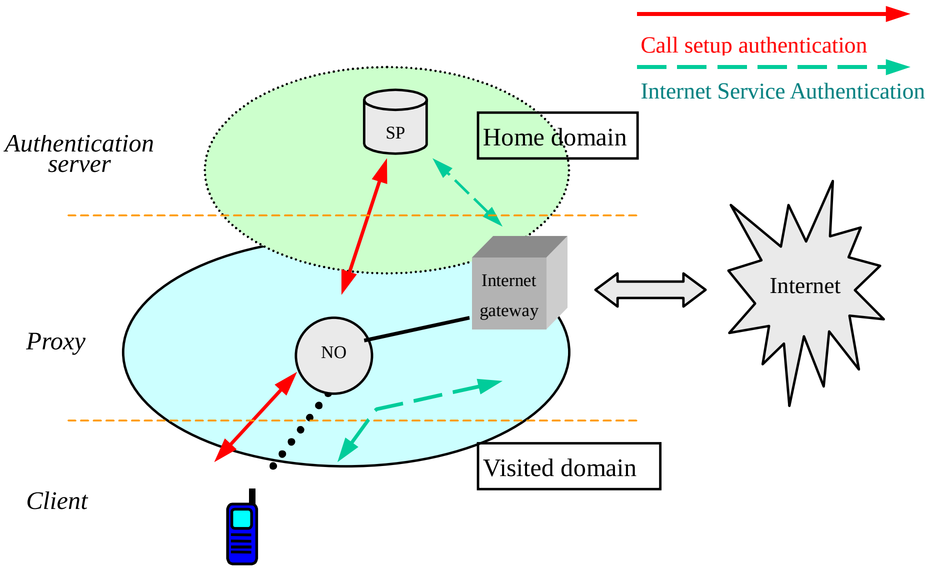


Fig. 3. The authentication roles in mobile services.

user is WAP service, the proxy of the authentication framework will be the WAP gateway. After receiving the authentication request from the user, the WAP gateway checks whether it can authenticate the user by itself or not. If it cannot, it delivers the authentication request to the user's home service provider by assuming that the WAP gateway trusts the authentication result received from the service provider. The service provider then authenticates the user, and the WAP gateway simply waits for the authentication result to arrive from the service provider. If authentication is successful, the WAP gateway provides WAP service to the user.

The current implemented authentication procedure for WAP service is as follows. The WAP gateway authenticates the user by itself. Each WAP gateway should privately maintain a user database and have its own authentication protocols. Therefore, the user needs to register in all the WAP gateways that he wants to use. This is inconvenient for users and WAP gateways. This approach suffers from the overhead involved in maintaining the user database. On the other hand, if our authentication framework is applied, the WAP gateway does not need to maintain a huge database of user information. The WAP gateway need not authenticate the users by itself and nor be involved in the authentication protocol.

3. PROPOSED AUTHENTICATION PROTOCOL

Based on the proposed framework, we will present a secure authentication protocol in this section. The proposed authentication protocol demonstrates how the proposed framework can be applied to enhance the flexibility of authentication in call setup services. The proposed protocol satisfies the security requirements of third generation mobile systems and enjoys the advantages of a flexible framework.

In the proposed authentication protocol, we assume that network operators and service providers have the public/private key pair and use asymmetric cryptosystems. In addition, there is public key infrastructure so that public keys can be correctly and efficiently distributed. This enables the network operators and service providers to mutually authenticate each other easily. The user and the service provider share a secret key. Message contents are represented according to the message format shown in Fig. 2, and the following notations are used to describe the proposed protocol.

IMUI	International mobile user identity.
TMUIs	Temporary user identity generated by service provider (SP)
TMUIn	Temporary user identity generated by network operator (NO)
Knp, Kns	Public/private key pair of network operator
Ksp, Kss	Public/private key pair of service provider
Kc	Session key shared by the user and network operator
Ku	Secret key shared by the user and service provider
N, Nk	Nonce numbers
CMn	Candidate mechanisms
T	Subscribed Service Period

Since the network operator is involved in the calling service, the proposed protocol considers the authentication of users both from the standpoint of the service provider and network provider. That is, the network operator is not only the proxy of the authentication procedure between the user and service provider but also is the authentication server of the user. Fig. 4 depicts the initial authentication procedure, and the authentication steps are described as follows.

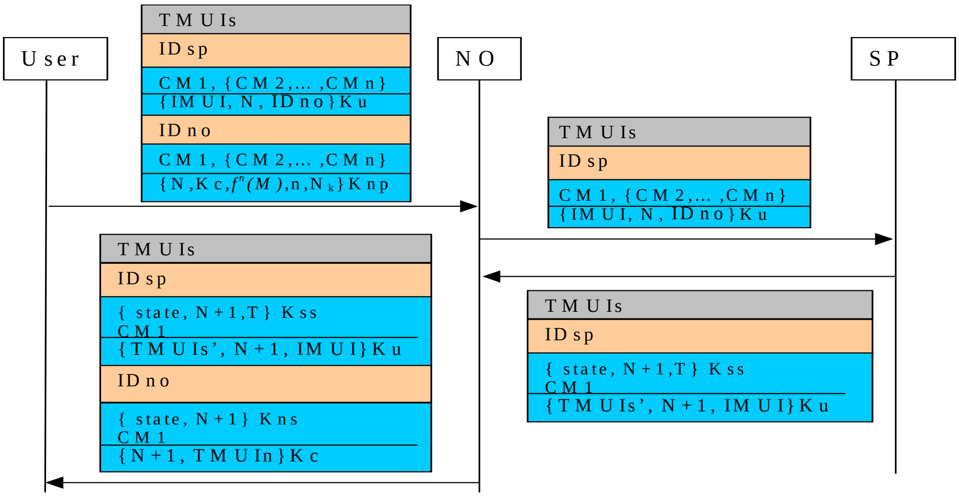


Fig. 4. Initial authentication procedure.

Step 1: While requesting a call setup service from NO in the visited domain, the user generates an authentication request message. The request message contains user information *TMUIs*, an authentication server identity *IDsp*, candidate authentication mechanisms {*CM*2, …, *CMn*} and the authenticator for SP. The first mechanism *CM*1 (in this case, *CM*1 is the mechanism we propose) is the default authentication method that the user chooses. The user generates a nonce *N* and puts it with *IMUI* as well as the NO identity *IDno* into the authenticator for SP. The authenticator is encrypted with *Ku* and becomes {*IMUI*, *N*, *IDno*}*Ku*.

Step 2: The user also generates a session key *Kc*. Candidate authentication mechanisms {*CM*2, …, *CMn*} and the authenticator {*N*, *Kc*, $f^n(M)$, *n*, *Nk*}*Knp* are put into the authentication token for NO, where $f^n(M)$ is a one-way hash function, *M* is secret information generated by the user and *n* represents the maximum number of services that the user can request after initial authentication. All of them are used to ensure non-repudiation, and *Nk* is used for session key generation.

Step 3: The user sends the request messages generated in steps 1 and 2 to NO.

Step 4: After receiving the request message, NO forwards the SP authentication token (mainly containing the authenticator {*IMUI*, *N*, *IDno*}*Ku*) to SP.

Step 5: Upon receipt of the authentication token, SP decides which candidate mechanism to use to authenticate the user. If SP agrees to use the same mechanism *CM*1 that the user chooses, it decrypts the authenticator using the user's secret key *Ku* associated with the *TMUIs* and verifies the unique identity *IMUI* of the user. According to the verification result, the authentication state will be "*accept*" or "*reject.*" The state is encrypted together with (*N* + 1) by SP's private key *Kss* so that no one can modify the state. After that, SP generates a new temporary identity *TMUIs*' for the user and uses *Ku* to encrypt *TMUIs*', *N* + 1, and *IMUI*. On the other hand, the response token indicates the mechanism that SP designates, and the state "*continue*" if SP does not choose the mechanism that the user chooses.

Step 6: SP uses the result produced in the previous step to generate a response message and sends it back to NO.

Step 7: While receiving the request message from the user in step 4, NO deciphers the authenticator by using its private key *Kns* to derive *N*, $f^n(M)$, *n*, *Nk* and *Kc* if NO agrees to use the same authentication mechanism *CM*1 that the user chooses. NO saves $f^n(M)$, *n*, and *Nk* for subsequent authentication and session key generation. Then NO waits to receive the authentication result from SP. If the state of the response from SP is not "*reject*" and if the value of (*N* + 1) is correct, NO generates a response to be sent to the user. The response authenticator of NO encrypted using *Kc* contains *N* and *TMUIn*', and the state is "*accept.*" In addition, NO keeps the subscribed service period for the mobile user. In the subsequent authentication phase, the subscribed service period is used to determine whether the service request is out-of-date or not. If the state of the response from SP is "*reject*," NO rejects the user's request. If the authentication protocol NO selected is different to *CM*1, then the state of the response message will be "*continue*," and NO will wait for the user to re-send the request message corresponding to the designated authentication mechanism.

Step 8: After generating the response message, NO sends its response message along with the one received from SP to the user.

Step 9: If the states of the two response messages are both "*accept*," then the user uses *Ku* to decrypt {*TMUIs*', *N* + 1, *IMUI*}*Ku* received from SP and uses *Kc* to decrypt {*N* + 1, *TMUIn*}*Kc* received from NO. The user then verifies the value of (*N* + 1). If it is correct and *IMUI* is correct, authentication is successful, and the user gets new temporary identities, *TMUIs*', as well as *TMUIn*. *Kc* becomes the shared key used by the user and NO. If the state of the response message is "*continue*," then the user re-sends the request message by using the designated mechanism to generate the authenticator, and the procedure restarts from step 1.

Authentication between the user and his service provider relies on the use of a shared secret key *Ku*. According to *TMUIs*, the service provider can find the associated secret key *Ku*. In this way, the service provider can authenticate the user, and the user can also authenticate the service provider. The authenticator that was sent to the network operator is encrypted using the public key of the network operator so that the user can authenticate the network operator. However, the message sent by the user can be produced by other malicious network entities; that is, the network operator cannot authenticate the user based only on the authentication result received from the service provider. Therefore, the network operator should verify whether the value of (*N* + 1) is correct or not. In this way, the network operator can make sure that the user is a legal user and that the request is fresh.

After the initial authentication, NO gets a secret key *Kc* that it shares with the user and subsequently can accomplish the authentication by itself. That is, subsequent authentication only happens between the user and NO using two message exchanges. Fig. 5 depicts the subsequent authentication procedure, and the authentication steps are described as follows.

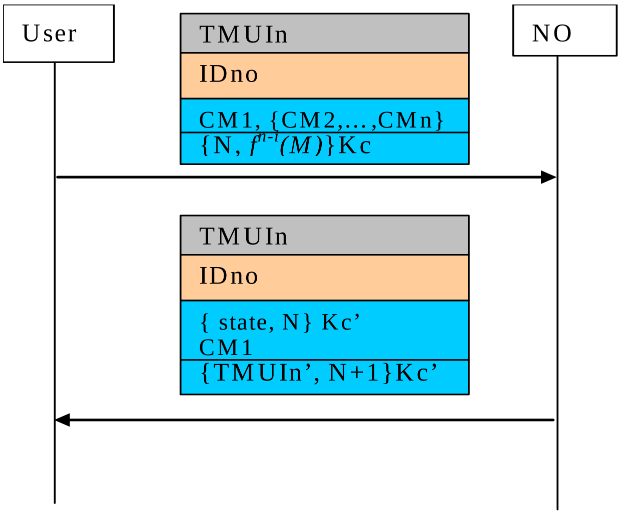


Fig. 5. Subsequent authentication procedure.

Step 1: When it requests a call delivery service from the visited domain after the initial authentication, the user generates a new session key *Kc*' = *h*(*Kc*, *N* + *Nk*), where *N* is a new generated nonce and *Kc* is the shared key. *Nk* is in the messages sent by the user to the network operator in the initial authentication procedure. Meanwhile, the user produces $f^{n-i}(M)$, where *i* is the number of services that have been requested, and *M* is the secret generated in the initial authentication.

Step 2: The user puts $\{N, f^{n-i}(M)\}Kc$ into the authenticator. Then the user sends the request containing *TMUIn* (user information), *IDno* (authentication server information), candidate authentication mechanisms, and the authenticator to NO.

Step 3: NO first checks the subscribed service period of the mobile user for the requested service. If the service request is not made within the valid subscribed service period, the service request is rejected. NO decides on the authentication mechanism the candidates {*CM*1, *CM*2, …, *CMn*}. If the protocol NO chooses is different from *CM*1, it specifies the designated protocol, and the response state is indicated as "*continue*." NO then sends the response to the user, and the user re-sends a request, in which the authenticator is produced according to the designated mechanism, to NO. The procedure then restarts from step 1. If NO chooses *CM*1, then it retrieves the shared session key *Kc* of the user according to *TMUIn*, decrypts {*N*}*Kc* and compares the result with *N*. Moreover, NO computes $f(f^{n-i}(M))$ to verify whether it is the same as the number, $f^{n-i+1}(M)$, which NO saved in the last authentication. If they are identical, the user has been authenticated successfully, and the response state is "*accept*."

Step 4: NO generates a new session key *Kc*' using the same function used by the user and then enciphers {*TMUIn*', *N* + 1} with *Kc*', where *TMUIn*' is the user's newly generated temporary identity. After the response token is generated, the authentication response is sent back to the user.

Step 5: Upon receipt of the response message, the user decrypts the authenticator using *Kc*' and gets the new identity *TMUIn*'.

The subsequent authentication procedure only contains two message exchanges. The nonce *N* transmitted between the user and network operator is used to refresh the session key. In this way, the encryption key used for every session is different. Except for the first session key, key generation is performed by both the user and network operator. The first session key is generated by the user and sent to the network operator. After that, the network operator can generate the following session key by itself. By using *Kc* and *Nk* + *N* as two inputs, the user and network operator can generate the same new session key if the inputs are identical.

End-to-End Security

After authentication is performed, the user can request a calling service to communicate with others. To provide data confidentiality and integrity throughout the entire connection path, a common encryption key must be negotiated between the participants to protect the communication contents. The key exchange between the end users is described as follows.

Step 1: Suppose user$_1$ originates the call; then, user$_1$ randomly generates an encryption key *Ke* and sends *N*, {{*Ke*}*Ks*1, {*N*}*Ke*}*Kp*2 to user$_2$, where *N*, *Ks*1, and *Kp*2 are nonce number, user$_1$'s private key and user$_2$'s public key, respectively.

Step 2: User$_2$ first decrypts the message by using his own private key *Ks*2 and derives the encryption key *Ke* by deciphering {*Ke*}*Ks*1 using user$_1$'s public key. He then uses *Ke* to decrypt {*N*}*Ke* and confirms the result with *N*.

Step 3: User$_2$ transmits the message {*N* + 1}*Ke* to user$_1$. After receiving the message sent by user$_2$, user$_1$ decrypts it to get *N* + 1 and then uses nonce *N* to verify that the information he received is correct. If it is correct, user$_1$ ensures that user$_2$ received the correct encryption key *Ke* needed to protect the communication contents.

The communication between the two parties in both wired and wireless paths is protected with the encryption key *Ke*. Therefore, end-to-end security is achieved in this way.

4. SECURITY ANALYSIS AND COMPARISON

In this section, we will discuss the security of the proposed protocol and compare it with other schemes. As mentioned previously, the security requirements of third generation mobile systems are mutual authentication, user anonymity, end-to-end security, non-repudiation, data integrity and data confidentiality. The proposed scheme can fulfill all of these requirements.

First, consider the authentication requirement. It is clear that the proposed authentication protocol can authenticate mobile users, service providers and network operators. Because the message sent to the service provider is encrypted using the shared secret key *Ku*, no one except for the home service provider can decrypt the message. Therefore, authentication between the user and the service provider can be achieved using *Ku*. The message sent to the network operator is encrypted using the public key of the network operator, so the user is assured that only the correct network operator can decrypt it. By comparing the two nonce numbers sent by the user and service provider, the network operator can authenticate the user. Consequently, mutual authentication is achieved.

Further, the proposed protocol can resist common attacks on authentication protocols. Our authentication protocol can resist the replay attack because the nonce is used to ensure the freshness of authentication sessions. Since the user and network operator must input *N* to generate a new session key, the session key can be refreshed for each communication. If an attacker replaces the *TMUI* of an intercepted authentication message and replays the authentication request, the attack will not succeed because the authenticator is encrypted using *Ku*. Next, consider the substitution attack. If an intruder replaces some fields of the authentication request, then the authentication will result in failure. For example, if an intruder replaces the authentication token {*N*, *Kc*, $f^n(M)$, *n*, N_k}*Knp* with {*N'*, *Kc'*, $f^n(M)$, *n*, N_k}*Knp* in the initial authentication, the network operator will find the nonce *N'* is different from the nonce *N* encrypted along with the authentication status sent by the service provider. Because the nonce used by the network operator is the same as the one used by the service provider, network operator can compare them to verify the user. Therefore, our protocol can resist the substitution attack. Moreover, even if an attacker gets the session key, he will not have the ability to generate new session keys. This is because session key generation involves N_k, and because only the user and network operator know N_k. The above analysis shows that our protocol can resist this kind of substitution attack. Thus, the proposed authentication protocol is secure and robust against attacks.

Next, consider user anonymity. To provide user anonymity, the permanent identity *IMUI* of users is never exposed in the plain-text mode. A cracker or even the network operator cannot get the real identity of a roaming user by eavesdropping on the authentication messages on both wireless and wired networks. After authentication is performed, all the network operator knows about the roaming user is a shared key and a temporary identity *TMUIn*, which unrelated to the permanent identity *IMUI*.

Consider the requirement of non-repudiation; our protocol can also satisfy this requirement. By using the one-way function, we can achieve non-repudiation. In the *i*-th session, the user provides $f^{n-i}(M)$ to ask for a connection. The network operator can verify the correctness of $f^{n-i}(M)$ by means of the one way function, but it can not derive $f^{n-i}(M)$ from $f^{n-i+1}(M)$. In this way, $f^{n-i}(M)$ can be used as a proof of the *i*-th connection. Whenever a random challenge occurs, the network operator can be required to show $f^{n-i}(M)$.

The requirement of end-to-end security is also addressed in the proposed protocol. When a user makes a call, the caller and callee negotiate a common encryption key to encrypt the data flowing over the channel. Since the full communication path is protected, data confidentiality and integrity are both achieved.

Table 1. A comparison of the proposed scheme with other schemes.

	Proposed scheme	Lee	Lin	Vanneste	M.1223-1	M.1223-2	M.1223-3
Mutal authentication	Yes	No	Yes	Yes	Yes	No	No
User anonymity	Yes	Yes	No	No	Yes	No	No
Non-repudiation	Yes	No	No	Yes	No	Yes	Yes
End-to-end security	Yes	Yes	No	No	No	No	No
Data confidentiality	Yes	Yes	Partial	Partial	Partial	Partial	Partial
Data integrity	Yes	Yes	Partial	Partial	Partial	Partial	Partial
Service independence	Yes	No	No	No	No	No	No
Protocol flexibility	Yes	No	No	Yes	No	No	No
Initial authentication message	4	5	4	14	5	?	?
Subsequent authentication messages	2	3	2	N/A	3	3	1

Compared to other schemes for mobile systems, our protocol is more secure and flexible. From the comparison shown in Table 1, we know that Lee's protocol [11] does not achieve mutual authentication and is not flexible. Although Lin's protocol [12] does authenticate all the communication parties, the protocol does not satisfy other security requirements, such as end-to-end security. The candidate security mechanism M.1223-1 of [8] also does not satisfy all the security requirements and is not flexible. The authentication protocol proposed in [22] achieves mutual authentication and is flexible, but it does not address the end-to-end security and user anonymity. Moreover, it involves too many authentication messages. Therefore, these protocols are not suitable for third-generation mobile systems. Since the proposed protocol is both secure and flexible, it can be efficiently applied to a variety of services in mobile systems.

5. CONCLUSIONS

In this paper, we have proposed an authentication framework that provides flexible authentication mechanisms for mobile systems. In the authentication framework, the authentication server can dynamically choose which authentication mechanism to use for each authentication request. Moreover, this property provides service providers with the ability to develop proprietary authentication mechanisms and adjust the security policy in run time. Based on the authentication framework, we have also presented a secure authentication protocol. The proposed protocol fulfills the security requirements of third generation mobile systems and requires fewer authentication messages than other protocols. In addition, the proposed authentication is secure against attacks, such as the replay attack and substitution attack. In short, the proposed authentication protocol satisfies the security requirements of third generation mobile communication systems. Moreover, it is efficient and robust.

REFERENCES

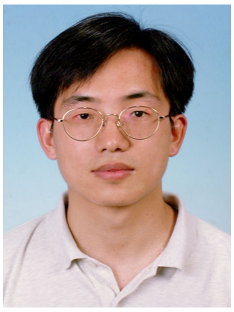
1. A. Barba, F. Recacha, and J. L. Melus “Security architecture in the third generation networks,” in *Proceedings of IEEE Singapore International Conference on Networks/International Conference on Information Engineering ’93*, Vol. 1, 1993, pp. 421-425.
2. T. G. Brutch and P. C. Brutch, “Mutual authentication, confidentiality, and key MANagement (MACKMAN) system for mobile computing and wireless communication,” in *Proceedings of the 14th Annual Computer Security Applications Conference*, 1998, pp. 308-317.
3. S. Dell’Uommo and E. Scarrone, “The mobility management and authentication/authorization mechanisms in mobile networks beyond 3G,” in *Proceedings of the 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2001, pp. 44-49.
4. L. Harn and H. Y. Lin, “Modification to enhance the security of the GSM protocol,” in *Proceedings of the 5th National Conference on Information Security*, 1995, pp. 74-76.
5. G. Horn, K. M. Martin, and C. J. Mitchell, “Authentication protocols for mobile network environment value-added services,” *IEEE Transactions on Vehicular Technology*, Vol. 51, 2002, pp. 383-392.

6. ITU-R Rec. M.687-2, "International mobile telecommunications-2000 (IMT-2000)," 1997.
7. ITU-R Rec. M.1078, "Security principles for international mobile telecommunications-2000 (IMT-2000)," 1997.
8. ITU-R Rec. M.1223, "Evaluation of security mechanisms for IMT-2000," 1997.
9. N. Jefferies, "Security in third-generation mobile systems," *IEE Colloquium on Security in Networks*, 1995, pp. 8/1-8/5.
10. J. Kim, M. Oh, and T. Kim "Security requirements of next generation wireless communications," in *Proceeding of International Conference on Communication Technology*, Vol. 1, 1998, pp. S2802-1-6.
11. C. H. Lee, M. S. Hwang, and W. P. Yang, "Enhanced privacy and authentication for the global system for mobile communications," *Wireless Networks*, Vol. 5, 1999, pp. 231-243.
12. C. T. Lin and S. P. Shieh, "Chain authentication in mobile communication systems," *Journal of Telecommunication Systems*, Vol. 13, 2000, pp. 213-240.
13. M. Looi, "Enhanced authentication services for internet systems using mobile networks," *IEEE Global Telecommunications Conference*, Vol. 6, 2001, pp. 3468-3472.
14. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, Boca Raton, FL, CRC Press, 1997.
15. S. Miller, C. Neuman, J. Schiller, and J. Saltzer, "Section E.2.1: Kerberos authentication and authorization system," M.I.T. Project Athena, Cambridge, Massachusetts, 1987.
16. R. Molva, D. Samfat, and G. Tsudik, "Authentication of mobile users," *IEEE Network*, Vol. 8, 1994, pp. 26-34.
17. S. Putz, R. Schmitz, and F. Tonsing, "Authentication schemes for third generation mobile radio systems," *The 9th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Vol. 1, 1998, pp. 126-130.
18. S. Putz, S. Putz and R. Schmitz, "Secure interoperation between 2G and 3G mobile radio networks," *The 1rst International Conference on 3G Mobile Communication Technologies*, 2000, pp. 28-32.
19. J. Rapeli "UMTS: targets, system concept, and standardization in a global framework," *IEEE Personal Communications*, Vol. 2, 1995, pp. 20-28.
20. C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote authentication dial in user service (RADIUS)," RFC2865, 2000.
21. J. Steiner, C. Neuman, and J. Schiller, "Kerberos: an authentication service for open network systems," *Usenix Conference*, 1988, pp. 191-202.
22. G. Vanneste *et al.*, "Authentication for UMTS: introduction and demonstration," *The 2nd International Distributed Conference on Network Interoperability*, 1997, pp. 715-721.
23. WAP Forum, "Wireless application protocol 2," WAP 2.0 Technical White Paper, http://www.wapforum.org/, 2002.

Shu-Min Cheng (鄭淑敏) was born in Taipei, Taiwan, R.O.C., in 1976. She received the B.S. and M.S. degrees from the Department of Computer Science and Information Engineering at National Chiao Tung University in 1998 and 2000, respectively. Her major research interest is network security.

Shiuhpyng Shieh (謝續平) is a Professor and former chairman of Department of Computer Science and Information Engineering of National Chiao Tung University. He is also, and the president of Chinese Cryptology and Information Security Association (CCISA), which is the largest and a highly respectable academic organization on information security research in Taiwan. He has worked as advisor to many institutes, such as National Security Bureau, GSN-CERT/CC, National Information and Communication Security Task Force. Before joining NCTU, Dr. Shieh participated in the design and implementation of the B2 Secure XENIX at IBM, Federal Sector Division, Gaithersburg, Maryland. He also designed and developed NetSphinx, a network security product, for Formosoft Inc., which is awarded 1999 network product of the year, Taiwan.

Dr. Shieh received the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Maryland, College Park. He is a senior member of IEEE, and an editor of ACM Transactions on Information and System Security, Journal of Computer Security, and Journal of Information Science and Engineering. He was on the organizing committees of numerous conferences, such as ACM conference on Computer and Communications Security, IACR Asiacrypt. Dr. Shieh published over a hundred academic articles, including papers, patents, and books. Recently he received the Outstanding Research Award from National Chiao Tung University for his academic achievement in research, and the Outstanding Achievement Award from Executive Yuan of Taiwan. His research interests include internetworking, distributed operating systems, and network security.



Wen-Her Yang (楊文和) received the M.S. and Ph.D. degrees in Computer Science from National Chiao Tung University in 1995 and 2000 respectively. He is currently a director of Network Security R&D Division of Formosoft International Inc. His research interests include operation systems, computer networks and network security.

Fu-Yuan Lee (李富源) received the B.S. degree in Computer Science from National Chiao Tung University in 1998. He is currently a Ph.D. student in the Department of Computer Science and Information Engineering at National Chiao Tung University. His research interests are in the areas of computer networks and network security.

Jia-Ning Luo (羅嘉寧) received the B.S. degree in Electrical Engineering and M.S. degree in Computer Science Engineering from Tatung University. He is currently a Ph.D. student in the department of Computer Science and Information Engineering, National Chiao Tung University. He is also a lecturer of the Overseas Chinese Institute of Technology. Jia-Ning Luo is interested in distributed systems and network security.