An Efficient Authentication Protocol for Mobile Networks*

SHIUH-PYNG SHIEH, FU-SHEN HO AND YU-LUN HUANG
Department of Computer Science and Information Engineering
National Chiao Tung University
Hsinchu, Taiwan 300, R.O.C.
E-mail: {*ssp, fsho, ylhuang*}*@csie.nctu.edu.tw*

Many existing authentication protocols supporting inter-domain authentication on the Internet require their clients to communicate with every involved key distribution center (KDC) directly. This is inefficient and costly when the client side is a wireless mobile unit, for wireless transmission has relatively lower bandwidth, and a mobile unit is battery powered. In this paper, we present a mobile authentication protocol which only needs seven messages for inter-domain initial authentication regardless of the number of hops transited between the visited and home domains; four messages for subsequent authentication when the mobile user requests a different service provided by the visited domain; and two messages when the same service is requested again. With the enhanced version of BAN logic we propose, it is proved that our protocol can achieve more goals of authentication than those required by the original BAN logic.

Keywords: mobile networking, network security, authentication, mobile computing, inter-domain authentication

1. INTRODUCTION

In the last few years, developments in wireless LAN cards and the related data link layer protocol standard [8] have enabled portable units to access wired networks without actually being attached to wired lines while providing wired LAN equivalent data privacy for wireless data. Continuous and transparent network access for mobile units is also achieved by the presence of mobile network layer protocol standards, such as Mobile-IP [16], where authentication is responsible for verifying the identity of a mobile station, rather than the mobile users in that station. However, for upper layer protocols, user authentication is still necessary when a mobile user wants to request services provided by the service providers in the visited networks. Logging and auditing are possible only if a strong authentication scheme is supported.

Although the computing power of mobile units has become more powerful, most of them are still battery powered. The power consumption of mobile units directly affects their usage time. Furthermore, using the current technology, the wireless data rate is lower than the wired data rate. Thus, when designing an authentication protocol suitable for the mobile networking environment, an important goal is to reduce the number of messages needed for authentication, especially the number of messages issued by a mobile unit. The number of authentication messages issued by a mobile unit must be small because it takes much more time to transmit a message, and the power consumed in data transmission is

Received August 4, 1997; accepted March 10, 1998.
Communicated by Jieh Hsiang.

* This work was supported in part by the National Science Council, Taiwan, R.O.C., under Grant NSC 86-2622-E009-007R.

very costly. In addition to wireless bandwidth and power consumption, roaming across networks must also be considered in the design of an efficient mobile authentication protocol so that inter-domain authentication can be completed with less effort.

Many proposed authentication protocols [9, 11, 15, 18, 19] focus on the authentication of users registered within the same domain, but authentication of mobile users registered with other domains is not supported by these protocols. Among other protocols, Kerberos Version 4 [17] developed at MIT uses a symmetric key based authentication and key distribution protocol for open network systems to support inter-domain authentication, which uses a timestamp that depends on reliable synchronized clocks to assure the freshness of messages. As Gong [7] noted concerning the difficulty of recovering from a post-dated clock, an authenticator can be replayed when the time incorrectly recorded in the authenticator is reached. Kerberos is vulnerable to replay attacks [2] if the clocks of clients and servers cannot be at least loosely synchronized. In Kerberos V4, in order for principals in domain X to be authenticated by principals in domain Y, it is necessary for the authentication server in domain Y to be registered as a principal in domain X. This is not scalable because complete interconnection of domains requires the exchange of n^2 keys, where n is the number of domains. As an improvement, multi-hop inter-domain authentication is provided by Kerberos Version 5 [12] and V5-based systems, such as the DCE security service [10], where each domain shares a key with its parent and children in a domain hierarchy; therefore, fewer keys are exchanged. However, both Kerberos V4 and V5 require their clients to contact each of the related KDCs directly to get the ticket to the next hop. This is inefficient and costly when the client side is a wireless mobile unit, for wireless transmission has relatively lower bandwidth and a mobile unit is battery powered. Furthermore, if the chain of domains transited during inter-domain authentication contains many domains that are far apart, the mobile user has to wait for the responses from the remote KDCs. If the traffic in the inter-connection networks is heavy, the problem gets even worse.

KryptoKnight [10, 14], developed at IBM, is a compact and flexible authentication and key distribution system which is similar to Kerberos. Seven messages are required for intra-domain initial authentication, and three messages for intra-domain subsequent authentication. Like Kerberos V4, inter-domain authentication between two principals belonging to different domains is only possible if both KDCs share a common key. In addition to KryptoKnight, inter-domain authentication in Microsoft Windows NT [10] is also similar to Kerberos V4, allowing peer trust links only. Thus, when the scale of the organization becomes larger, management of peer trust links will be more difficult. Some practical issues related to deploying Kerberos in large organizations were discussed in [5].

In the original BAN logic [4], it is initially assumed that the shared session key is associated with its communication parties. However, BAN logic cannot prove the association between a session key and the principals who are going to use it to establish a secure communication channel. Unfortunately, it may be vulnerable to impersonation of a particular principal if the session key is not associated with its communicating parties in an authentication protocol. Thus, we have enhanced BAN logic to support proof of the association relationship between a session key and its communicating parties.

This paper is organized as follows. In section 2, a new authentication protocol with support for both initial and subsequent authentication within the same domain is proposed. In section 3, we extend our authentication scheme discussed in section 2 to satisfy the needs of inter-domain authentication in mobile networks. We also discuss possible attacks and the reasons why our protocols can resist these attacks in section 4. We compare our authentication protocols with other protocols in section 5. Finally, conclusions are given in section 6. In the appendix, we enhance original BAN logic [4] and use it to prove that our protocol can achieve more goals of authentication than those required by the original BAN logic.

2. THE PROPOSED INTRA-DOMAIN AUTHENTICATION PROTOCOL

In this section, we propose a nonce-based (nonce is a random number which is used only once) authentication and key distribution protocol, which supports both intra-domain initial authentication in four messages and subsequent authentication without the participation of the key distribution center in two messages. Both the initial and subsequent authentication achieve the goals of authentication and key distribution without the need for synchronized clocks. Based on the existence of continuous and transparent mobile network layer protocols, such as Mobile-IP, in our mobile environment, the proposed intra-domain authentication protocol can be further extended to an inter-domain authentication protocol, which satisfies the requirements of the mobile environment discussed in the previous section and supports efficient inter-domain subsequent authentication without the aid of the mobile user's home domain. The extended inter-domain authentication protocol is discussed in details in section 3.

2.1 Initial Authentication

In this intra-domain authentication protocol, there are three kinds of principals: mobile users, service providers and key distribution centers, where service providers provide services to mobile users registered in the same domain, and key distribution centers are trusted by both mobile users and service providers. A KDC is responsible for the distribution of session keys to the registered principals who want to establish a secure communication channel. Each registered principal (i.e., service and mobile user) shares a key with its KDC, and the key is securely stored in the database in the KDC which must be protected in a physically secure place.

After a user or service provider has registered with his KDC, he obtains a master key. The master key is shared with the KDC and should be delivered manually or using any other secure method in the initial phase.

The message flow of the intra-domain initial authentication protocol is shown in Fig 1, and the contents of each message are as follows :

Message 1	$U_a \rightarrow S_a$:	U_a, N_{Ua},
Message 2	$S_a \rightarrow KDC_a$:	U_a, S_a, N_{Ua}, N_{Sa},
Message 3	$KDC_a \rightarrow S_a$:	$\{U_a, N_{Sa}, K_{ss}, \{S_a, N_{Ua}, K_{ss}\}K_{Ua}\}K_{Sa}$,
Message 4	$S_a \rightarrow U_a$:	$\{S_a, N_{Ua}, K_{ss}\}K_{Ua}, \{U_a, VT_{Sa}, K_{ss}\}K_{Sa}, N_{Sa',}$
[Message 5]	$U_a \rightarrow S_a$:	$\{N_{Sa'}\}K_{ss}, \{S_a, VT_{Ua}, K_{ss}\}K_{Ua}$.

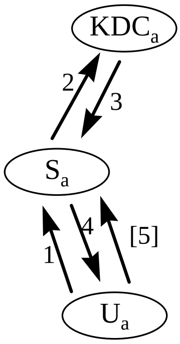


Fig.1. The intra-domain authentication protocol.

A user U_a initiates authentication by sending a service request to the service provider S_a. In Message 1, the identity of U_a and that of a challenge nonce N_{Ua} randomly generated by U_a are sent to the service provider. The service provider itself cannot authenticate the user, so it forwards the service request together with its name S_a and a challenge nonce N_{Sa} to its KDC (KDC_a). Then, both the user and service provider wait for the response from the KDC (Message 2).

Upon receiving the authentication request from S_a, the KDC queries its database and finds the service provider's master key K_{Sa} and the user's shared key K_{Ua}. After verifying that the two principals are still valid, KDC_a creates two credentials for both U_a and S_a, and encrypts the credentials using their own master keys. Each credential contains the identity of the principal's communication peer, the principal's challenge nonce, and a session key K_{ss} generated by KDC_a, which are used to assure that each principal believes the freshness of the session key and to associate the key with his communication peer. The credential for U_a is included in that for S_a. Finally, KDC_a sends back the credential for S_a as a response to the authentication request (Message 3).

Upon receiving the response from KDC_a, S_a decrypts its credential with its master key and derives a session key and the user's credential. After verifying the identity of U_a and the freshness of the nonce, S_a believes the freshness of the session key and its association with the other principal U_a. Then, S_a sends the user's credential, a *ticket* for the user and another challenge nonce $N_{Sa'}$ back to the user (Message 4).

In this authentication protocol, a ticket for a user is issued by the service provider at the end of an initial authentication. Such a ticket is used to perform efficient subsequent authentication solely between the service provider which issues this ticket and the user who receives the ticket. A ticket contains the identity of the issuer's communicating peer, the issuer's local time of validity (VT) and the session key. Since the ticket is encrypted with the master key of the issuer, it is only recognizable to the issuer.

Upon receiving Message 4, U_a decrypts the credential with his master key and also derives a session key. After verifying the freshness of the nonce, U_a can believe the freshness of the session key and its association with the other principal S_a. With the enhanced version of BAN logic [4] we propose, the proof of the association relationship will be presented in the appendix. Now, U_a can also assure that S_a has believed the session key, because the credential for U_a is included in the credential for S_a. Finally, U_a sends back the encrypted challenge nonce indicating that he also believes the session key K_{ss}, along with the ticket for S_a (Message 5). The advantage of issuing tickets to each other is that neither S_a nor U_a needs to keep the session key in memory when the session is completed. This reduces the risk that the session key may be stolen by reading the memory in the principal's machine. However, if mutual authentication between the user and the service provider is not required in the authentication phase, Message 5 can be eliminated so that only four messages are needed for the initial authentication, and only one message is issued by the mobile unit. When the service provider receives the first data message from the mobile unit, mutual authentication can still be achieved by verifying the correctness of the data. However, if Message 5 is not issued, the user must store the session key securely by himself at the end of a communication session in order to perform the subsequent authentication.

2.2 Subsequent Authentication

When U_a wants to request the same service from S_a again, he initiates the subsequent authentication protocol as follows :

Message 1'	$U_a \rightarrow S_a$:	$\{U_a, VT_{Sa}, K_{ss}\}K_{Sa}, N_{Ua}'$,
Message 2'	$S_a \rightarrow U_a$:	$\{N_{Ua}', K_{ss}'\}K_{ss}, \{S_a, VT_{Ua}, K_{ss}\}K_{Ua}, N_{Sa}''$,
[Message 3']	$U_a \rightarrow S_a$:	$\{N_{Sa}''\}K_{ss}'$.

In Message 1', U_a sends the ticket previously issued by the service provider S_a and challenge nonce N_{Ua}'. After verifying the validity of the ticket, S_a randomly generates a new session key K_{ss}' for this subsequent session and uses the old session key K_{ss} to encrypt the new session key and N_{Ua}'. Then, it sends back this encrypted message, the ticket issued by U_a and another nonce N_{Sa}'', and requests that U_a show that he is able to derive the new session key (Message 2'). Upon receiving it, U_a decrypts the ticket with his master key and checks the validity of the ticket. He uses the session key in the ticket to derive the new session key K_{ss}' generated by S_a. After checking the freshness of the nonce encrypted together with K_{ss}', U_a can trust the new session key and verify the identity of the service provider. Then, he uses the new session key to encrypt N_{Sa}'' and sends the encrypted nonce back to S_a (Message 3'). After S_a has sucessfully decrypted Message 3' with the new session key successfully and verified the freshness of the nonce, the identity of U_a is also verified. The use of a different session key in the new session prevents any attack involving replaying of old messages used in previous communication sessions, so this method can provide better security. Similarly, the last message in the subsequent authentication protocol can also be eliminated if mutual authentication is not needed in the authentication phase.

3. THE EXTENDED INTER-DOMAIN AUTHENTICATION PROTOCOL

In the previous section, we presented an authentication protocol which authenticates principals registered in the same domain. However, when the scale of the environment is large, it is inefficient and impractical for all the principals to be registered in a single security domain. Instead, users and service providers should register with their own KDCs. A common administrative structure of organizations is a hierarchical model, where every node in the hierarchy represents a domain, and where parent domains manage all their children domains. A natural and consistent naming convention of domains on the Internet is Domain Name System [13]. Fig. 2 shows an example of hierarchical domains, where the EDU.TW domain manages the NCTU.EDU.TW, NTU.EDU.TW, and the NTHU.EDU. TW domains, and the NCTU.EDU.TW domain manages the CSIE.NCTU.EDU.TW and CIS.NCTU.EDU.TW domains, and so on. In the proposed extended authentication protocol, every KDC must share a different secret key with all its ancestors in order to perform inter-domain authentication efficiently. Consequently, the root KDC needs a large database to store the shared keys for all descendant KDCs. Fortunately, the size of a key is small, and the root KDC is able to store all the keys.

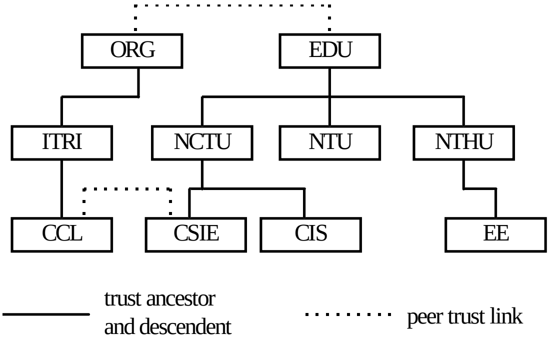


Fig.2. An example of hierarchical domains.

Once such registrations with its ancestors have been accomplished, any key distribution center in the hierarchy can securely and automatically change its inter-domain authentication key periodically, which can be achieved by simply providing a secure key-changing service in each domain. In addition to the normal trust relationship, a peer trust link can also be established if two domains register with each other (i.e., share a different key). Peer trust links are needed when there is no trustworthy administrative parent between two domains. Moreover, peer trust links can also gain performance benefits.

We have assumed that mobile users can know how to access the services provided by visited domains. This can be realized by periodically broadcasting the access points of available services on the wireless channel.

3.1 Initial Authentication

When a mobile user visits a remote domain and wants to request the service provided by the visited domain, he must first prove his identity to the service provider. The KDC in the visited domain cannot authenticate the identity of the mobile user because the user has not registered with it. To authenticate the users registered with other domains, the protocol presented in section 2 has to be extended. In our extended inter-domain authentication protocol, any principal registered in a domain obtains a *permanent principal name*, which is issued by its KDC and will not be changed as long as the principal is valid. A principal name contains two fields of information: a principal's identity registered in some domain and its certification path. A certification path of a principal is its fully qualified home domain name in the hierarchy. For example, if the whole hierarchy follows the Internet DNS naming convention, a user registered in the domain CSIE.NCTU.EDU.TW may obtain a principal name like JAMES@CSIE.NCTU.EDU.TW, which is very similar to his electronic mail address.

The message flow of the extended inter-domain initial authentication protocol is shown in Fig. 3 and the contents of each message are listed as follows :

Message 1”	$TU_x \rightarrow S_y$:	TU_x, N_{TUx};
Message 2”	$S_y \rightarrow KDC_y$:	TU_x, S_y, N_{TUx}, N_{Sy};
Message 3”	$KDC_y \rightarrow KDC_0$:	TU_x, S_y, N_{TUx}, KDC_y, N_{KDCy};
Message 4”	$KDC_0 \rightarrow KDC_x$:	$\{N_{KDCy}, K_{ss}, TU_y, K_{TUy}\}K_{KDCy}$, $\{TU_x, S_y, N_{TUx}, K_{ss}, TU_y, K_{TUy}, KDC_y, N_{KDCy}\}K_{KDCx}$;
Message 5”	$KDC_x \rightarrow KDC_y$:	$\{N_{KDCy}, K_{ss}, TU_y, K_{TUy}\}K_{KDCy}$, $\{N_{KDCy}, PU_a, INFO_{PUa}\}K_{TUy}$, $\{S_y, N_{TUx}, K_{ss}, TU_y, K_{TUy}\}K_{TUx}$;
Message 6”	$KDC_y \rightarrow S_y$:	$\{TU_x, N_{Sy}, K_{ss}, \{S_y, N_{TUx}, K_{ss}, TU_y, K_{TUy}\}K_{TUx}\}K_{Sy}$;
Message 7”	$S_y \rightarrow TU_x$:	$\{S_y, N_{TUx}, K_{ss}, TU_y, K_{TUy}\}K_{TUx}$, $\{TU_x, VT_{Sy}, K_{ss}\}K_{Sy}$, $Ns_{y'}$;
[Message 8”]	$TU_x \rightarrow S_y$:	$\{N_{Sy'}\}K_{ss}$, $\{S_y, VT_{TUx}, K_{ss}\}K_{TUx}$.

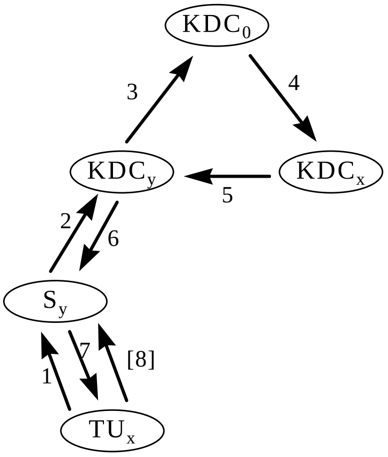


Fig.3. The extended inter-domain authentication protocol.

The first two and last three messages of this extended protocol are very similar to those of the original protocol, except that in this extended protocol, the user identity field uses a *temporary principal name*. When a user requests any services provided by the visited domain for the first time, he will be assigned a *temporary principal name* registered with the KDC of the visited domain and a *temporary authentication key* after successful of initial authentication so that all future subsequent authentication within the visited domain can be completed without the aid of the home KDC of the user. A temporary principal name is recognizable to the KDC of the visited domain, and its certification path field is the certification path of the visited domain. Usually, a temporary principal name has a shorter life time and fewer privileges than a permanent principal name. For each temporary principal name, there is a corresponding temporary authentication key to prove its identity. Typically, a mobile user who has visited different domains may have some such name-key pairs, which may be encrypted with the master key of the user and kept in a secure place (e.g., a smart card) to provide better security. A mobile user may choose the nearest ever-visited domain and use the temporary principal name issued by it to complete inter-domain authentication efficiently. If the temporary account has expired, the user can use another principal name issued by another domain to prove his identity and obtain a new temporary principal name from the currently visited domain. If a user only has his permanent principal name (i. e., the user never visits other domains or he has lost all the name-key pairs), he can use his permanent principal name as a temporary principal name.

When a mobile user (TU_x) who has registered in domain X, the certification path of which is <KDC_x, KDC_{x1}, ..., KDC_0, *same KDC list*>, arrives in a new visited domain (domain Y, the certification path of which is <KDC_y, KDC_{y1}, ..., KDC_0, *same KDC list*>) and wants to request the service (S_y) provided by domain Y, he first issues a service request (Message 1”) to the service provider. Since S_y cannot authenticate this user, it forwards the request message together with its own challenge nonce to its KDC (Message 2”).

Upon receiving the authentication request, KDC_y checks the user’s name and learns the certification path of the user. KDC_y compares the certification path with its certification path and tries to find a common point of trust. If KDC_y cannot find anything in common, the authentication will fail and the service request will also be rejected. KDC_y just issues an error message to S_y to inform it of the failure of authentication, and S_y also informs TU_x. If the most closely common point of trust can be found (e.g., KDC_0 in this case), KDC_y will continue to verify TU_x’s identity. It will send TU_x, S_y, N_{TUx}, its identity KDC_y and its nonce N_{KDCy} to KDC_0 (Message 3”).

When KDC_0 receives the authentication request from its child KDC (KDC_y), it looks up in its database, and finds that both KDC_y and the KDC of TU_x are its children. Thus, KDC_0 considers it an inter-domain authentication request. KDC_0 generates a session key K_{ss}, a temporary principal name TU_y, and a temporary authentication key K_{TUy} which will be used as the master key of TU_y. Then, KDC_0 generates the credentials for both KDC_y and KDC_x, and sends the two credentials to KDC_x to ask for the authentication of TU_x (Message 4”).

Upon receiving the instruction from its parent (KDC_0), KDC_x decrypts its own credential and finds that TU_x wants to request the service S_y in domain Y. KDC_x generates the credential for TU_x, including the nonce issued by TU_x, the session key and the new temporary authentication key K_{TUy}. Then, KDC_x sends the credential for TU_x, his permanent principal name PU_a and user related information (e.g., validity time and privileges) encrypted with K_{TUy} and the credential for KDC_y received in Message 4” to KDC_y (Message 5”). Upon receiving Message 5”, KDC_y verifies its credential issued by the common point of trust KDC_0 and obtains TU_y and K_{TUy}. Then, it continues to decrypt the response generated by KDC_x using K_{TUy}. After verifying the freshness of the response, KDC_y also gets the permanent principal name and his related information for the purpose of auditing. It creates a new temporary account named TU_y in its inter-domain principal database and uses K_{TUy} as the master key of the new account.

Basically, Messages 6”, 7” and 8” play the same roles as Messages 3, 4, and 5 described in section 2. However, in Messages 6” and 7”, TU_y and K_{TUy} are added to the credential for TU_x for the purpose of efficient inter-domain authentication for roaming users. Therefore, after verifying the legality and validity of Message 7”, TU_x becomes a registered principal in the new visited domain Y before the temporary account expires.

As discussed in section 2, the last message can be eliminated if mutual authentication is not needed in the authentication phase. Thus, only seven messages are needed for inter-domain initial authentication regardless of the number of hops between the visited and home domains. Furthermore, if the mobile user who wants to request the service provided by domain Y is registered in any of Y’s ancestors or in Y’s peer trust links, Message 4” is unnecessary and can be eliminated. Therefore, only six messages are needed in this case.

3.2 Subsequent Authentication

When a mobile user has obtained a temporary principal name and the corresponding authentication key from the KDC in the newly visited domain, the subsequent authentication protocol within the same visited domain will be exactly the same as the initial authentication protocol presented in section 2. Furthermore, if the user wants to request the same service which he requested previously in the same visited domain, the subsequent authentication protocol can be reduced to two messages, which is just the subsequent authentication protocol presented in section 2.

4. COUNTERING POSSIBLE ATTACKS

As cryptographic researchers [1, 3] have suggested some practical and systematic design principles of for an attack-resistant authentication protocol, careful and systematic design of our protocols can effectively prevent malicious attacks from intruders. In the following, we describe some possible attacks and why our protocols can prevent these attacks.

4.1 Trivial Substitution and Replay Attacks

Since all of our protocols are nonce-based instead of timestamp-based protocols and every credential in our protocols contains the verifier’s nonce used to verify the freshness of that credential, trivial substitutions and replays of old authentication messages can be easily prevented. In practice, like other nonce-based protocols, our protocol expects the destination to send a response whenever it successfully receives a challenge (that is, a nonce). Every time it sends a challenge, the sender starts a timer(eg., 5 minutes) and waits for a response. If the time expires before the response to the challenge is received, the sender assumes that the challenge or its response has been either lost or is corrupted and then issues a new challenge. Thus, the sender need not record all used nonces in a database. The only things that have to be recorded in the database are the nonces for the un-expired challenges still waiting for the responses.

4.2 Impersonation Attacks With Compromised Session Keys

In general, session keys are relatively easier to be compromised than master keys because they have to be stored in local memory, which might be insecure throughout the session. If an intruder U_i compromises an old session key used by user U_x and service provider S_y, he can try to make the service provider believe he is U_x by replaying some old messages and forging and substituting some credentials in the protocol run. Denning and Sacco [6] pointed out this kind of attack and suggested a method using timestamps, which requires synchronized clocks on all participants. Kao and Chow [9] included a temporary key along with the session key issued by the KDC, which is used to encrypt the responses of the challenges of mutual authentication. Since the temporary key is used just for the current authentication session and is discarded right away after authentication, an impersonation attack with a compromised session key can be prevented. In our intra-domain and inter-domain initial authentication protocols, we rely on neither timestamps nor temporary keys. Instead, due to the architecture of our protocols, this kind of attack can be detected easily by a service provider in Messages 3 and 6" by checking the freshness of the nonce in the service provider's credential. That is, if the intruder substitutes N_{Sa} in Message 2 and replays Message 3, the service provider can still detect that Message 3 is simply a replay by verifying the nonce in the credential, and there is no opportunity for the intruder to forge Message 5. Thus, the intruder will be rejected even if he holds a compromised session key.

4.3 Oracle Session Attacks

An oracle session attack was pointed out by Ray Bird et al. [3], in which an intruder starts two separate authentication sessions with two different service providers, such that he is able to take advantage of the messages in one authentication session to successfully impersonate a particular user in the other session. This kind of attack can be effectively prevented if the encrypted messages used in each run of the protocol are different from or logically linked with one another. In our intra-domain subsequent authentication protocol, the final message (Message 3') contains the new session key K_{ss}', which associates the last message with the previous message in the protocol. Therefore, an intruder cannot impersonate U_a to the service provider S_a using the oracle session attack because S_a will compare the received session key with the one it just issued in this subsequent authentication session and reject the intruder.

5. COMPARISONS

Comparing our proposed inter-domain mobile authentication protocol with other protocols described in previous sections, we get the following table (see Table 1).

In Table 1, the minimum number of messages for authentication, the number of messages for mutual authentication, the number of messages submitted by the user for authentication, and the number of messages submitted by the user for mutual authentication are compared. Among these protocols, our protocol requires a constant number of messages independent of the number of KDCs transited between the user's visited domain and home domain (denoted as *m*) whereas the number of messages in Kerberos V5 depends on *m*. If *m* is larger than two, Kerberos V5 requires more messages than our protocol. Although other protocols listed in Table 1 also require a constant number of messages, they do not support multi-hop authentication. In addition, due to the low data rate and battery power of a mobile unit, only one message is issued by a mobile user in our protocol while at least three messages are needed in other protocols. Therefore, when deploying our protocol in the mobile networking environment, the performance is better than that of other protocols discussed in this section.

Table 1. Inter-domain initial authentication.

	Minimum # of messages	# of messages for mutual authentication	# of messages submitted by user	# of messages submitted by user for mutual authentication	Type of trust	# of shared keys	User Mobility Support
Kerberos V4	7	8	4	4	peer	O(N^2)	No
Windows NT	7	-	2	-	peer	O(N^2)	No
Kerberos V5	2*m*+3	2*m*+4	*m*+2	*m*+2	peer & hierarchical	O(N)	No
Our protocol	7	8	1	2	peer & hierarchical	O(N)	Yes

m: number of KDCs transited between the user's visited domain and home domain.

N: number of domains.

The numbers of shared keys is also compared in Table 1. Consider a domain hierarchy which is a full *k*-ary tree with height *h*. The number of domains in such a hierarchy is $N = \sum_{i=0}^{h-1} k^i$. *h* is independent of the total number of transit KDCs, *m*. Whenever a user visits a domain, the height of the domain hierarchy tree is unchanged. Instead, the total number of transit KDCs changes according to the location of the visited domain in the hierarchy. The total number of shared keys in our protocol is:

$$\sum_{i=1}^{h-1} k^i = \frac{k(k^{h-1}-1)}{k-1} = O(k^{h-1}) = O(N),$$

which is the same as the number of shared keys in Kerberos V5. However, in our protocol, more keys must be stored than in Kerberos V5 to perform inter-domain authentication. Fortunately, the size of a key is small and the total amount of storage can be easily accommodated in a hard disk. In peer-trust protocols, such as Kerberos V4, the number of shared keys becomes $\frac{N(N-1)}{2}$, that is, O(N^2).

Our protocol provides better security than Kerberos. Every subsequent communication session in our protocol uses a different key. In contrast, Kerberos uses the same key for all sessions before a ticket expires. This makes Kerberos vulnerable to replay attacks.

6. CONCLUSIONS

In this paper, we first presented a simple nonce-based intra-domain authentication protocol, which efficiently supports both initial and subsequent authentication. Based on the existence of continuous and transparent mobile network layer protocols (e.g., Mobile-IP), the intra-domain authentication protocol has been extended to an inter-domain authentication protocol, which requires seven messages for initial authentication regardless of the number of hops transited between the visited and home domains, four messages for subsequent authentication when the mobile user requests a different service, and two if he requests the same service again. Since only one message is issued by the mobile unit for authentication, the protocol provides higher efficiency than other protocols due to their limited wireless data rate. With the enhanced version of BAN logic we propose, our intra-domain and inter-domain authentication protocols for mobile networks can achieve a larger number of goals of authentication than is recommended by BAN logic.

Since KDCs are isolated from users, only registered service providers can communicate with them directly, and users can only talk to intended service providers. This design agrees with the concept of the client-server model, which is the current trend in Internet applications. This architecture is also very suitable for mobile networks, where a mobile user only needs to connect to a visited service provider for authentication. Furthermore, with our protocols, illegal requests from users can be effectively rejected by service providers before they are forwarded to KDCs because access control lists will be consulted by the service providers to determine whether the requests should be rejected or not. Therefore, unnecessary authentication steps can be avoided, and the load on the KDC can be reduced.

REFERENCES

1. M. Abadi and R. Needham, “Prudent engineering practice for cryptographic protocols,” *IEEE Transactions on Software Engineering*, Vol. 22, No. 1, 1996, pp. 6-15.
2. S. M. Bellovin and M. Merritt, “Limitations of the kerberos authentication system,” *ACM Communications Review*, Vol. 20, No. 5, October 1990, pp. 119-132.
3. R. Bird, et al., “Systematic design of a family of attack-resistant authentication protocols, ” *IEEE Journal on Selected Areas in Communications*, Vol. 11, No. 5, 1993, pp. 679-693.
4. M. Burrows, M. Abadi and R. Needham, “A logic of authentication,” *ACM Transactions on Computer Systems*, Vol. 8, No. 1, 1990, pp. 18-36.
5. “Deploying kerberos for large organizations,” Technical Report, 94-47 Communication Department, CyberSafe Corporation, 1994.
6. D. E. Denning and G. M. Sacco, “Timestamps in key distribution protocols,” *Communications of the ACM*, Vol. 24, No. 8, 1981, pp. 533-536.
7. L. Gong, “A security risk of depending on synchronized clocks,” *ACM Operating Systems Review*, Vol. 26, No. 1, 1992, pp. 49-53.
8. IEEE Standard 802.11, “Wireless LAN medium access control (MAC) and physical layer (PHY) specifications,” *IEEE Draft Standard*, 1996.
9. I. L. Kao and R. Chow, “An efficient and secure authentication protocol using uncertified keys,” *ACM Operating Systems Review*, Vol. 29, No. 3, 1995, pp. 14-21.

10. C. Kaufman, et al., *Network Security: Private Communication in a Public World*, PTR Prentice Hall, Englewood Cliffs, New Jersey 07632, 1995.
11. A. Kehne, et al., “A nonce-based protocol for multiple authentication,” *ACM Operating Systems Review*, Vol. 26, No. 4, 1992, pp. 84-89.
12. J. Kohl and C. Neuman, “The kerberos network authentication service (V5),” *Internet Request for Comments 1510*, 1993.
13. P. Mockapetris, “Domain names: concepts and facilities,” *Internet Request for Comments 1034*, 1987.
14. R. Molva, et al., “KryptoKnight authentication and key distribution system,” in *Proceedings of 1992 European Symposium on Research in Computer Security*, 1992, pp. 155-174.
15. B. C. Neuman and S. G. Stubblebine, “A note on the use of timestamps as nonces,” *ACM Operating Systems Review*, Vol. 27, No. 2, 1993, pp. 10-14.
16. C. Perkins, “IP mobility support,” *Internet Request for Comments 2002*, 1996.
17. J. G. Steiner, B. C. Neuman and J. I. Schiller, “Kerberos: an authentication service for open network systems,” in *Proceedings of the Winter 1988 Usenix Conference*, 1988, pp. 191-201.
18. Paul Syverson, “On key distribution protocols for repeated authentication,” *ACM Operating Systems Review*, Vol. 27, No. 4, 1993, pp. 10-14.
19. S. P. Shieh and W. H. Yang, “An authentication and key distribution system for open network systems,” *ACM Operating Systems Review*, Vol. 30, No. 2, 1996, pp. 32-41.

APPENDIX. PROTOCOL ANALYSIS

In this appendix, we explain why our protocol can reach the goals of authentication for both initial and subsequent authentication.

A.1 Initial Authentication

The formal method, BAN logic, presented by Burrows et al. [4] states that authentication is complete between two parties, the user and the service provider, if there is a K_{ss} such that

$$\mathrm{TU}_x\,\textbf{believes}\ TU_x \xleftrightarrow{K_{ss}} S_y, \mathrm{S}_y\ \textbf{believes}\ TU_x \xleftrightarrow{K_{ss}} S_y.$$

The first two formulas state that each principal trusts K_{ss} as a secret session key shared with each other. Some protocols may provide more beliefs:

$$\mathrm{TU}_x\ \textbf{believes}\ \mathrm{S}_y\ \textbf{believes}\ TU_x \xleftrightarrow{K_{ss}} S_y, \mathrm{S}_y\ \textbf{believes}\ \mathrm{TU}_x\ \textbf{believes}\ TU_x \xleftrightarrow{K_{ss}} S_y.$$

These two formulas state that each principal believes that the other one currently trusts the session key. The functional objectives of the initial authentication are to prove (1) the presence of both parties to each other, and (2) the user's receipt of a ticket and a session key for subsequent authentication. At the end of the initial authentication without message 8”, we can achieve the following set of formalized goals:

TU$_x$ **believes** $TU_x \xleftrightarrow{K_{ss}} S_y$, (1)

S$_y$ **believes** $TU_x \xleftrightarrow{K_{ss}} S_y$, (2)

TU$_x$ **believes** S$_y$ **believes** $TU_x \xleftrightarrow{K_{ss}} S_y$. (3)

The protocol achieving this set of goals will not have undetected faults. In addition to the three formalized goals, our protocol can achieve one more formalized goal when message 8" is used or when the service provider receives the first data message from the user:

S$_y$ **believes** TU$_x$ **believes** $TU_x \xleftrightarrow{K_{ss}} S_y$**.** (4)

The proof is given as follows. When the server receives message 8", the annotation rules yield that

S$_y$ **sees** {N$_{Sy'}$, $TU_x \xleftrightarrow{K_{ss}} S_y$}K$_{ss}$.

Since we have formula (2), the message-meaning rule for shared keys applies and yields the following:

S$_y$ **believes** TU$_x$ **said** (N$_{Sy'}$, $TU_x \xleftrightarrow{K_{ss}} S_y$).

Since the nonce N$_{Sy'}$ is generated by S$_y$, we have the following hypothesis:

S$_y$ **believes fresh** (N$_{Sy'}$, $TU_x \xleftrightarrow{K_{ss}} S_y$).

The nonce-verification rule applies and yields

S$_y$ **believes** TU$_x$ **believes** (N$_{Sy'}$, $TU_x \xleftrightarrow{K_{ss}} S_y$).

The jurisdiction rule applies and yields

S$_y$ **believes** TU$_x$ **believes** $TU_x \xleftrightarrow{K_{ss}} S_y$**.**

A.2 Subsequent Authentication

Upon receipt of message 1', S$_a$ can decrypt this message and check VT$_{Sa}$. Since VT$_{Sa}$ has not expired, K$_{ss}$ is still good. Applying the message-meaning rule, the nonce-verification rule, and the jurisdiction rule, we get

S$_a$ **believes** U$_a$ **believes** $U_a \xleftrightarrow{K_{ss}} S_a$.

The service provider generates a new session key K$_{ss}$' for each subsequent session, which means that K$_{ss}$' will not be reused. Therefore, replay attacks can be prevented. The jurisdiction rule applies and yields

S_a **believes** $U_a \overset{K_{ss}^{1}}{\longleftrightarrow} S_a$,

Upon receipt of message 2’, U_a can decrypt this message to check N_{Ua}’. Since N_{Ua}’ is generated by U_a, we get

U_a **believes fresh** (N_{Ua}’, $U_a \overset{K_{ss}^{1}}{\longleftrightarrow} S_a$) ,

U_a **believes** S_a **believes** (N_{Ua}’, $U_a \overset{K_{ss}^{1}}{\longleftrightarrow} S_a$) .

The jurisdiction rule applies and yields

U_a **believes** $U_a \overset{K_{ss}^{1}}{\longleftrightarrow} S_a$.

Upon receipt of message 3’ or the first data message from the user, S_a can decrypt this message and verify N_{Sa}”. Since N_{Sa}” is generated by S_a, we get

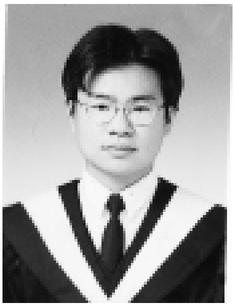
S_a **believes fresh** (N_{Sa}”, $U_a \overset{K_{ss}^{1}}{\longleftrightarrow} S_a$) ,

S_a **believes** U_a **believes** (N_{Sa}”, $U_a \overset{K_{ss}^{1}}{\longleftrightarrow} S_a$) .

The jurisdiction rule applies and yields

S_a **believes** $U_a \overset{K_{ss}^{1}}{\longleftrightarrow} S_a$.

Shiuh-Pyng Shieh (ÁÂÄÒ¥) received the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, in 1986 and 1991, respectively. He is currently an associate professor with the Department of Computer Science and Information Engineering, National Chiao Tung University. From 1988 to 1991, he participated in the design and implementation of the B2 Secure XENIX for IBM, Federal Sector Division, Gaithersburg, Maryland, U.S.A. He is also the designer of SNP (Secure Network Protocols). Since 1994, he has been a consultant for the Computer and Communications Laboratory, Industrial Technology Research Institute, Taiwan, in the area of network security and distributed operating systems. He is also a consultant for the National Security Bureau, Taiwan. Dr. Shieh has been on the organizing committees of a number of conferences, such as the International Computer Symposium and the International Conference on Parallel and Distributed Systems. Recently, he is the general chair of 1998 Network Security Technology Workshop, the program chair of 1999 Mobile Computing Conference and 1997 Information Security Conference (INFOSEC’97). His research interests include internetworking, distributed operating systems, and network security.



Fu-Shen Ho (¦óºÖºa) received the B.S. and M.S. degrees in computer science and information engineering from National Chiao Tung University in 1995 and 1997, respectively. He is currently a Ph.D. student in the same department. His research interests include operating systems design, distributed systems and network security.

Yu-Lun Huang (¶À¨|ºú) received her B.S. degree in computer science and information engineering from National Chiao Tung University in 1995. She is currently a Ph.D. candidate in the same department. Her research interests include electronic commerce, distributed systems, data hiding and network security. She is a member of the Phi Tau Phi Society.