Efficient Authentication Protocol for Mobile Networks*

Kee-Won Kim, Jun-Cheol Jeon, Kee-Young Yoo**

Department of Computer Engineering, Kyungpook National University

Abstract

The mobile devices are constrained to be low battery, and the mobile data channel is low data rate. Therefore, the computational cost on the client side of the authentication protocol should be low. In 2005, Hwang and Su proposed an efficient authentication protocol for mobile networks. Hwang-Su protocol is more efficient than other related protocols. In this paper, we propose a new efficient authentication protocol for mobile networks. The proposed protocol is more efficient than Hwang-Su's in computational respect although our protocol is nearly equal to Hwang-Su's in communication respect. It is suitable to apply in the mobile networks.

I. Introduction

With the rapid development of communication technology, mobile network technology has become more and more important, and has been widely used in personal communication. Accordingly, a lot of research has been devoted to the authentication protocols for mobile devices which enable the users to be authenticated by the service providers before consuming the requested services. The mobile devices are constrained to be low battery, and the mobile data channel is low data rate. Therefore, the computational cost on the client side of the authentication protocol should be low, and the number of interactions between the client and the service provider should be as less as possible.

Among many proposed authentication protocols, Kerberos [5, 4] is one of the most widely deployed protocols. Kerberos is based on the technology of timestamp and symmetric key cryptography. Kerberos requires the clients to contact each of the related Key Distribution Center (KDC) directly to get the tickets to the next hop. This is inefficient and costly.

Many efforts have been devoted to improve the security, the scalability, and/or the efficiency of Kerberos [3, 9, 1, 6, 11]. Some of them utilized public key to alleviate the burden of central secret key database while incurring more computational load on the client side [9, 6]. Some utilized the technique of session key certificate to reduce the possibility of session key compromise while keeping the central secret key databases [3, 11]. However, these protocols cannot withstand the known key attack once the session key is compromised.

Chien and Jan [2] pointed out that the Shieh-Ho-Huang protocol [11] based on the technique of session key certificate is vulnerable to the known key attack, and then proposed an authentication protocol for mobile networks based on public key cryptography, challenge-response and hash chaining. The proposed protocol consists of two sub-protocols, namely the intra-domain authentication protocol and the inter-domain authentication protocol, which are used depending on whether or not the user and the service provider are located in the same domain. In the intra-domain authentication protocol, the user and the service provider are located in the domain of the same KDC. In the inter-domain authentication protocol, it is assumed that each domain has a KDC and the KDC acts as the authority center for its domain. However, Tang and Mitchell [8] pointed out that both intra- and inter-domain authentication protocols of Chien and Jan [2] suffer from impersonation attacks.

Recently, Hwang and Su [10] proposed a efficient authentication protocol that is based on symmetric key, challenge-response and hash chaining. The proposed protocol consists of the intra-domain authentication protocol and the inter-domain authentication protocol. Both protocols are composed of two phases: the initial phase and the subsequent phase. They claimed that their protocols simultaneously possesses several practical

* This work was supported by the Brain Korea 21 Project in 2006.

** Corresponding author : Kee-Young Yoo (yook@knu.ac.kr)

good scalability, low communication and computational costs, and resistance to replay attacks, impersonation attacks, and known key attacks.

In this paper, we propose a new efficient intra-domain authentication protocol for mobile networks. The initial phase of the proposed protocol is based on symmetric encryption/decryption and hash function. The subsequent phase of the proposed protocol is based on only hash function. The proposed protocol is more efficient than Hwang-Su's in computational respect although our protocol is nearly equal to Hwang-Su's in communication respect.

The remainder of this paper is organized as follows. In Section 2, we propose a new efficient authentication protocol for mobile networks. In Section 3, we analyze the security of our protocol. In Section 4, the performance of the proposed protocol is discussed. Finally, we give a brief conclusion in Section 5.

II. The Proposed Protocol

We first list the notations used throughout this paper in the following.

- U, ID_U : a mobile user and his/her identity.
- S, ID_S : a service provider and his/her identity.
- KDC : a key distribution center.
- $h(\cdot)$: a one-way hash function.
- $f(\cdot)$: a secret key derivation function.
- $\oplus$: a bitwise XOR operation.
- $||$: a concatenation operation.
- K_C : a long-term secret key of KDC.
- K_{UC} : a shared key of U and KDC, $K_{UC}=f(K_C, ID_U)$.
- K_{SC} : a shared key of S and KDC, $K_{SC}=f(K_C, ID_S)$.
- $E_K(M)$: the result of encrypting M using the secret key K.

If the mobile user and the service provider registered in the same KDC, they process the intra-domain authentication protocol. The proposed intra-domain authentication protocol is composed of two phases: the initial phase and the subsequent phase. The mobile user who wants to access the service provider first performs the initial phase. Then the mobile user can get a session key. The initial phase achieves the purposes of authentication and key distribution. The mobile user performs the subsequent phase based on the session key to access the server n times. The subsequent phase achieves the purposes of authentication and key update between the mobile user and the service provider. The role players in this protocol are the mobile user (U), the KDC (C), and the service provider (S). The user and the KDC share a secret key K_{UC}. The service provider and the KDC share a secret key K_{SC}. The long-term secret key of KDC (C) is K_C.

2.1. The initial phase

The mobile user U and the service provider S registered in the same KDC. The mobile user wants to access the service provider. U firstly uses the initial phase to get the session key with the service provider. The mobile user, the service provider and the KDC authenticate one another in this phase. The initial phase is shown in Figure 1. The details are presented as follows:

1. $U \rightarrow S$: ID_U, N_U, $h(N_U, K_{UC})$

   U selects a random nonce N_U and computes $h(N_U, K_{UC})$, where K_{UC} is a shared key of U and KDC. Then U sends $\{ID_U, N_U, h(N_U, K_{UC})\}$ to S.

2. $S \rightarrow KDC$: ID_U, N_U, $h(N_U, K_{UC})$, ID_S, N_S, $h(N_S, K_{SC})$

   S selects a random nonce N_S, stores $\{N_S, ID_U, N_U\}$ and computes $h(N_S, K_{SC})$, where K_{SC} is a shared key of S and KDC. Then, S sends $\{ID_U, N_U, h(N_U, K_{UC}), ID_S, N_S, h(N_S, K_{SC})\}$ to KDC.

3. $KDC \rightarrow S$: $E_{K_{SC}}(N_S, h^n(a), n)$, N_C, M_U, V_U

   KDC authenticates the user by checking $h(N_U, K_{UC})$ based on $K_{UC}=f(K_C, ID_U)$ and the service provider by checking $h(N_S, K_{SC})$ based on $K_{SC}=f(K_C, ID_S)$. KDC chooses a random number a and computes $h^n(a)$, where $h^n(a)$ represents n iterations of hash function $h(\cdot)$ and n is the maximum number of times that U allowed to request the service of S. KDC selects a random nonce N_C and computes $M_U = h(N_C, K_{UC}) \oplus (a||n)$ and $V_U = h(M_U \oplus N_U, K_{UC})$. Then KDC sends $\{E_{K_{SC}}(N_S, h^n(a), n), N_C, M_U, V_U\}$ to S.

4. $S \rightarrow U$: N_C, M_U, V_U, W

   S decrypts the message using K_{SC} and authenticates KDC by the decrypted data N_S. S computes $K_n = h(n, h^n(a))$ as the session key and $W = h(N_C \oplus N_U, K_n)$ using K_n and $\{N_S, ID_U, N_U\}$ stored in Step 2. S keeps $\{ID_U, h^n(a), a\}$. Then S sends $\{N_C, M_U, V_U, W\}$ to U.

   Upon receiving $\{N_C, M_U, V_U, W\}$, U obtains a

and n by computing $M_U \oplus h(N_C, K_{UC})$ and authenticates KDC by checking V_U based on N_U and K_{UC}. U computes the session key $K_n = h(n, h^n(a))$ and authenticates S by checking W based on N_U and K_n. Then U keeps $\{a, n, K_n\}$ for requesting services in the future.

2.2 The subsequent phase

After the initial phase, the mobile user can request service from the service provider without involving the KDC as follows. The user is allowed to request the service n times. Without loss of generality, we assume the user is requesting the ith service now ($1 \le i \le n$). The subsequent phase is shown in Figure 2. The details are presented as follows:

1. $U \rightarrow S$: ID_U, $h^{n-i}(a) \oplus h^{n-i+1}(a)$

   U computes $h^{n-i}(a)$, $h^{n-i+1}(a)$, and $h^{n-i}(a) \oplus h^{n-i+1}(a)$. Then U sends $\{ID_U, h^{n-i}(a) \oplus h^{n-i+1}(a)\}$ to S.

1. $S \rightarrow U$: $h(h^{n-i}(a), K_{n-i})$

   S computes $h^{n-i}(a) \oplus h^{n-i+1}(a) \oplus h^{n-i+1}(a)$ to get $h^{n-i}(a)$ and checks whether $h(h^{n-i}(a))$ equals the stored hash value $h^{n-i+1}(a)$. If the check succeeds, S computes the new session key $K_{n-i} = h(n-i, h^{n-i}(a))$ and sends $h(h^{n-i}(a), K_{n-i})$ to U; otherwise rejects the request. Finally, S replaces $h^{n-i+1}(a)$ with $h^{n-i}(a)$ and stores i.

   Upon receiving $h(h^{n-i}(a), K_{n-i})$, U computes the new session key $K_{n-i} = h(n-i, h^{n-i}(a))$ and checks $h(h^{n-i}(a), K_{n-i})$. If the check succeeds, U believes that the right S has confirmed the new session key and keeps i.

III. Security Analysis

We show that the proposed protocol withstands four possible attacks: the relay attack, the impersonation attack, the known key attack and the oracle session attack.

3.1 Replay attack

In the initial phase, an adversary E intercepts the transmitted messages and replays the request $h(N_U, K_{UC})$, as U to S for authenticating. However, E does not posses the user's long-term secret key K_{UC}. E cannot get the correct secret information in Step 4. E cannot generate the correct session key. Our proposed initial phase is secure against replay attacks.

In the subsequent phase, an adversary E eavesdrops $\{ID_U, h^{n-j}(a) \oplus h^{n-j+1}(a)\}$ in the jth subsequent phase. In the ith subsequent phase, if E replays $\{ID_U, h^{n-j}(a) \oplus h^{n-j+1}(a)\}$ to S, where $j < i \le n$, S rejects it because $h(h^{n-j}(a) \oplus h^{n-j+1}(a) \oplus h^{n-i+1}(a) \neq h^{n-i+1}(a)$. In each run of the subsequent phase, the user has to provide the next hash value, and only the right user can compute this value. Therefore, replay attacks cannot work in our subsequent phase.

3.2 Impersonation attack

In the initial phase, an adversary E impersonates a legal user U to the service provider S, which results in S being cheated. But E does not have the user's long-term secret key K_{UC} and the freshness of the message is assured by the nonce, impersonation attacks fails in our initial phase.

In the subsequent phase, if an adversary E wants to impersonate the user U, E must compute a valid $h^{n-i}(a) \oplus h^{n-i+1}(a)$. Because E has no idea about n and a, E cannot forge a valid $h^{n-i}(a) \oplus h^{n-i+1}(a)$. Therefore, impersonation attacks fails in our subsequent phase.

3.3 Known key attack

In our initial and subsequent phase, even if the session key $K_{n-i+1} = h(n-i, h^{n-i}(a))$ is compromised, the known key attack still fails because an adversary can not get the $h^{n-i}(a)$. However, only the legitimate user can computes the hash value.

3.4 Oracle session attack

In our protocol, because an adversary E cannot get the keys K_{UC} and K_{SC}, E cannot get the correct a and n. Therefore, our protocol is secure against oracle session attacks.

IV. Performance analysis

Hwang-Su protocol is more efficient than Shieh-Ho-Huang [11] protocol and Chien-Jan protocol [2] in computation and communication load. The comparisons of the performance of the intra-domain protocol of our protocol and Hwang-Su protocol are shown in Table 1. Our subsequent phase is based on only hash function, not like Hwang-Su protocol that uses symmetric encryption/decryption and hash function.

In order to analyze the computation cost of these protocols, we assume that the time for performing a symmetric key encryption denotes T_S and the time for performing a hash function denotes T_H. Then, $T_S \approx 10 T_H$ [7]. In the initial phase and the subsequent phase, our protocol is more efficient than Hwang-Su protocol.

In order to analyze the communication cost of these protocols, we assume that identity (for example, U and S) is 32 bits, nonce is 64 bits, hash function digest is 160 bits (for SHA-1), and the bit length of hash chain number is 32 bits. In the initial phase of our protocol, the total sizes of communication messages are 1952 bits, while Hwang-Su protocol's protocol takes 1536 bits. But, the initial phase performs only one in entire protocol. In the subsequent phase of ours and Hwang-Su protocol, the total sizes of communications messages are 352 bits. Therefore, our protocol is more efficient than Hwang-Su's in computational respect although our protocol is nearly equal to Hwang-Su's in communication respect. The proposed protocol is suitable to apply in the mobile network.

Table 1. The comparisons of the computation and communication cost of intra-domain protocols

		Our protocol		Hwang-Su [10]	
		I.P.	S.P.	I.P.	S.P.
Hash function	U	n+4	n-i+2	n+2	n+i
	S	3	3	2	3
	C	n+4	0	n+4	0
Symmetric key encryption	U	0	0	0	1
	S	0	0	1	1
	C	1	0	2	0
Symmetric key decryption	U	0	0	2	1
	S	1	0	1	1
	C	0	0	0	0
Communication cost		1952 bits	352 bits	1536 bits	352 bits

V. Conclusions

In this paper, we have proposed a new efficient authentication protocol for mobile networks. The proposed protocol is more efficient than Hwang-Su protocol in computational respect. The proposed protocol withstands replay attacks, impersonation attacks, known key attacks, and oracle session attacks. The proposed protocol is efficient and suitable to apply to the mobile network.

[Reference]

[1] A. Fox and S.D. Gribble. Security on the move: Indirect authentication using kerberos. Proccedings of the Second Annual International Conference on Mobile Computing and Networking, MOBICOM, pp.155-164, 1996.

[2] H.Y. Chien and J.K. Jan. A hybrid authentication protocol for large mobile network. Journal of Systems and Software, Vol. 67, pp.123-137, 2003.

[3] I.-L. Kao and R. Chow. An efficient and secure authentication protocol using uncertified keys. ACM Operating Systems Review, Vol. 29, No. 3, pp.14-21, 1995.

[4] J. Kohl and C. Neuman. The kerberos network authentication service (v5). Internet Request for Comments, 99(1510), pp.1-100, January 1993.

[5] J.G. Steiner, B.C. Neuman, and J.I. Schiller. Kerberos: An authentication service for open network systems. Proceedings of the Winter Usenix Conference, pp.191-201, 1988.

[6] M.A. Sirbu and J.C.I. Chuang. Distributed authentication in kerberos using public key cryptography. Proceedings in IEEE Symposium on Network and Distributed System Security, pp.134-141, 1997.

[7] M.S. Hwang, I.C. Lin and L.H. Li. A simple micro-payment scheme. The Journal of Systems and Software, Vo. 55, pp.221-229, 2001.

[8] Q. Tang and C.J. Mitchell. Cryptanalysis of a hybrid authentication protocol for large mobile networks. Journal of Systems and Software, Available online at www.sciencedirect.com, In Press, June.

[9] R. Ganesan. Yaksha: Augmenting kerberos with public key cryptography. Proceedings of IEEE Symposium on Network and Distributed System Security, pp.132-143, 1995.

[10] R.-J. Hwang and F.-F. Su. A new efficient authentication protocol for mobile networks. Computer Standars & Interfaces, Vol. 28, pp.241-252, 2005.

[11] S.P. Shieh, F.S. Ho, and Y.L. Huang. An efficient authentication protocol for mobile networks. Journal of Information Science and Engineering, Vol. 15, pp.505-520, 1999.