Improving the Novikov and Kiselev User Authentication Scheme

Minho Kim[1] and Çetin Kaya Koç[2]
(Corresponding author: Minho Kim)
Department of Computer Science, Korea Air Force Academy[1]
Chungbuk, Cheongwon, Namil, Sangsu, 363-849, South Korea (Email: mhkim@lifetime.oregonstate.edu)
Information Security Research Center, Istanbul Commerce University[2]
Eminönü, Istanbul 34112, Turkey

(Received Mar. 2, 2006; revised and accepted May 7, 2006 & Aug. 8, 2006)

Abstract

Novikov and Kiselev [7] proposed an authentication method of a user from a remote autonomous object. Recently, Yang et al. [12] and Awasthi [1] have pointed out that the Novikov-Kiselev scheme is insecure against the man-in-the-middle attack. In this article, we propose an improved version of the Novikov-Kiselev scheme to overcome such vulnerability.

Keywords: Authentication, man-in-the-middle attack, remote autonomous object

1 Introduction

Whenever users want to access remote systems, they should be authenticated. Once authorized, they can then access the resources of the server. The scatter of remote systems in difference places allows more efficient and convenient access for geographically dispersed users. Lamport [5] proposed a remote password authentication scheme which authenticates remote users over an insecure channel. However, it suffers from the stolen-verifier attack if the adversary has the ability to obtain the stored verifier. It also has some practical implementation difficulties, such as the problems of high overhead and password resetting. Since then, many remote authentication schemes have been proposed [2, 4, 6, 9, 10, 11]. In 2000, Hwang and Li [3] proposed a remote user authentication scheme using smart cards. It was shown that no password table is required to keep in a system.

In general, the password authentication schemes use a pair of identity and password values to access the remote system. However, many researchers have attempted to create a remote authentication scheme without using passwords. In 2003, Novikov and Kiselev [7] proposed an algorithm of reliable authentication of the user from a remote autonomous object. Recently, Yang, Lee, and Hsiao [12] have pointed out that [7] is insecure against the man-in-the-middle attack. Awasthi [1] came to the same conclusion too, in addition to stating that [7] is vulnerable against the man-in-the-middle attack and the reflection attack.

In this paper, we propose a more secure scheme than the Novikov-Kiselev scheme. The rest of the paper is organized as follows. In Section 2, we will briefly review the Novikov-Kiselev Scheme. We will then show Yang-Lee-Hsiao's attack and Awasthi's attack in Sections 3 and 4. In Section 5, we will propose our improved scheme. We will analyze how our scheme can be secured against the man-in-the-middle attack in Section 6. Finally, we will conclude this paper in Section 7.

2 Review of the Novikov-Kiselev Scheme

2.1 Notations

- U_i, O, and E denote the user, the remote autonomous object, and the adversary.
- ID_i and K denote the user's identifier and the control command.
- (S_{PK_U}, S_{SK_U}) and (S_{PK_O}, S_{SK_O}) denote a pair of session keys of U_i and O.
- T_i denotes the time parameter.
- The expression $A \longrightarrow B : X$ means A sends the message X to B via a public communication channel.

The Novikov-Kiselev authentication scheme described in [7] comes in two stages described below.

2.2 The First Stage

The first stage is the pre-tuning of the parameters U_i and O. U_i produces ID_i and synchronizes T_i with the remote

object. This processing is executed just once. ID_i and T_0 are produced and stored in the operative memory of O by U_i.

2.3 The Second Stage

The second stage is the communication session between U_i and O. The procedures of this stage are as follows:

S1. $U_i \longrightarrow O : signal\ S$.
U_i sends start request signal S to O.

S2. $O \longrightarrow U_i :\ S_{PK_O}$.
O computes a pair of session keys S_{PK_O} and S_{SK_O} by using the RSA algorithm [8]. Next, O sends S_{PK_O} to U_i, and then turns on the timer to record the session beginning at time T_1.

S3. $U_i \longrightarrow O :\ E_{S_{PK_O}}(ID_i, S_{PK_U})$.
U_i generates a pair of session keys S_{PK_U} and S_{SK_U}, and then encrypts ID_i and S_{PK_U} with S_{PK_O} using the encryption function of the RSA algorithm. Next, he sends it to O.

S4. $O \longrightarrow U_i :\ E_{S_{PK_U}}(X)$.
O decrypts the received message with S_{SK_O} using the decryption function. O records T_2, which indicates the time that the message was received, and checks that $\Delta T = T_2 - T_1$. If $\Delta T \geq T_0$, then this communication is terminated. Otherwise, O checks received ID_i and stored ID_i in its own memory. If they are correct, O encrypts the message X which includes the command K with S_{PK_U}. Next, O sends it to U_i and records the time T_3.

S5. $U_i \longrightarrow O :\ E_{S_{PK_O}}(newID, K)$.
U_i decrypts the received message with S_{SK_U}, and then obtains X. He derives the command K from the message X, and then encrypts the command K and new identifier $newID$ with S_{PK_O}. Next, U_i sends it to O. After that, U_i records the value of $newID$ in his memory and destroys his pair of session keys (S_{PK_U}, S_{SK_U}) and S_{PK_O}.

S6. O checks $\Delta T = T_3 - T_2$. If $\Delta T \geq T_1$, then this communication is terminated. If it is valid, then O decrypts the received message with S_{SK_O} and obtains the command K. Next, O replaces $newID$ in his memory and destroys a pair of session keys (S_{PK_O}, S_{SK_O}). Finally, O executes the command K.

3 Yang-Lee-Hsiao Attack

The procedures of attack are briefly described as follows:

In Step S1, the adversary E intercepts the signal S.

In Step S2, E intercepts S_{PK_O}.

In Step S5, E intercepts $E_{S_{PK_O}}(newID, K)$ and replaces it with $E_{S_{PK_O}}(newID', K')$, and then sends it to O.

The adversary can attack by replacing the legal identifier of the user. Therefore, E can easily supplant the legal user.

4 Awasthi Attack

4.1 The First Stage

The user U_i sends ID_i and T_0 to the object O. On this communication, the adversary E intercepts the information and sends the tuple ID_i and T_* instead of the original one.

4.2 The Second Stage

AS1. $U_i \longrightarrow O : signal\ S$.
U_i sends start request signal S to O. E intercepts S and sends it to O.

AS2. $O \longrightarrow U_i :\ S_{PK_O}$.
The object O computes a pair of session keys S_{PK_O} and S_{SK_O} by using the RSA algorithm. Next, O sends S_{PK_O} to U_i, and then turns on the timer to record the session beginning at time T_1.

AS3. $E \longrightarrow U_i :\ S_{PK'_O}$.
E intercepts S_{PK_O} and sends a self generated $S_{PK'_O}$ to the user.

AS4. $U_i \longrightarrow O :\ E_{S_{PK'_O}}(ID_i, S_{PK_U})$.
U_i generates a pair of session keys S_{PK_U} and S_{SK_U}, and then encrypts ID_i and S_{PK_U} with $S_{PK'_O}$ using the encryption function of the RSA algorithm. Next, he sends it to O.

AS5. $E \longrightarrow O :\ E_{S_{PK_O}}(ID_i, S_{PK'_O})$.
E intercepts this encrypted message and decrypts it using $S_{SK'_O}$. He modifies the encrypted message as $E_{S_{PK_O}}(ID_i, S_{PK'_O})$ and sends it to O.

AS6. $O \longrightarrow U_i :\ E_{S_{PK'_O}}(X)$.
O decrypts the received message with S_{SK_O} using the decryption function. O records T_2, which indicates the time that the message was received, and checks that $\Delta T = T_2 - T_1$. If $\Delta T \geq T_*$, then this communication is terminated. O encrypts the message X which includes the command K with $S_{PK'_O}$. Next, O sends it to U_i.

AS7. $E \longrightarrow U_i :\ E_{S_{PK_U}}(X)$.
E intercepts this message and decrypts $E_{S_{PK'_O}}(X)$ using $S_{SK'_O}$. Next, he encrypts X with S_{PK_U} and sends it to U_i.

AS8. $U_i \longrightarrow O :\ E_{S_{PK'_O}}(newID, K)$.
U_i decrypts the received message with S_{SK_U}, and then obtains X. He derives the command K from the message X, and then encrypts the command K and the new identifier $newID$ with $S_{PK'_O}$. Next, U_i sends it to O.

AS9. $E \longrightarrow O:\ E_{S_{PK'_O}}(newID, K)$.
E intercepts the message $E_{S_{PK'_O}}(newID, K)$ and decrypts it with $S_{SK'_O}$. E can get the $newID$ and K. After that, E can make whatever modifications he wants.

5 Our Improved Scheme

We will make an improvement on the Novikov-Kiselev scheme by taking into account the man-in-the-middle attack of [12] and [1]. E monitors their communications and then eavesdrops the messages, such as S, S_{PK_O}, and $E_{S_{PK_O}}(newID, K)$. Next, he sends the replaced messages to O. Here we notice the weakness. Since O did not check the command K, E can obtain S_{PK_O} and replace $E_{S_{PK_O}}(newID, K)$ with $E_{S_{PK_O}}(newID', K')$ at any time. We modify the Novikov-Kiselev scheme as follows.

5.1 The Synchronization Phase

The user U_i sends ID_i to the object O. O computes a pair of session keys (S_{PK_O}, S_{SK_O}) by using the RSA algorithm and sends his session public key S_{PK_O} to U_i. Next, they synchronize according to time T_0. During this phase, these transactions are done via a secure communication channel. Next, O stores ID_i and T_0 in its memory.

5.2 The Authentication Phase

A1. $U_i \longrightarrow O: E_{S_{PK_O}}(ID_i,\ T_i)$.
U_i checks the current time T_i and encrypts ID_i and T_i by using session public key S_{PK_O}. Next, U_i sends start request signal to O with encrypted message $E_{S_{PK_O}}(ID_i,\ T_i)$.

A2. $O \longrightarrow U_i:\ E_{S_{SK_O}}(T_{j+1},\ X)$.
When O receives the message, as a first step, it records the current time T_j, and then decrypts the received message by using session private key S_{SK_O} and obtains ID_i and T_i. Next, O compares the time $\Delta T_x = T_j - T_i$ with T_0 that is stored in the memory at the synchronization phase. If $\Delta T_x \geq T_0$, then this communication is terminated. If they are valid, O turns on the timer and records the session beginning time T_{j+1}, and then sends $E_{S_{SK_O}}(T_{j+1},\ X)$ to U_i.

A3. $U_i \longrightarrow O:\ E_{S_{PK_O}}(T_{i+1},\ T^*_{j+1},\ T_{i+2},\ K')$.
When U_i receives the message, he records the current time T_{i+1} with priority, and then computes $D_{S_{PK_O}}[E_{S_{SK_O}}(T_{j+1},\ X)]$ to decrypt and obtain T^*_{j+1} and X. He can derive the command K' from the message X, and then encrypt T_{i+1}, T^*_{j+1}, T_{i+2} and K' with S_{PK_O}, where T_{i+2} is the new current time of U_i. Next, U_i sends it to O. After that, U_i destroys session public key S_{PK_O}.

A4. When O receives the message, it begins by recording the current time T_{j+2}, and then computes $D_{S_{PK_O}}[E_{S_{PK_O}}(T_{i+1},\ T^*_{j+1},\ T_{i+2},\ K')]$ to decrypt and obtain T_{i+1}, T^*_{j+1}, T_{i+2} and K'. Next, O checks that T^*_{j+1} equals to T_{j+1} that was sent by O in Step A2. If they are the same, then O compares the time $\Delta T_y = T_{j+1} - T_{i+1}$ and $\Delta T_z = T_{j+2} - T_{i+2}$ with T_0. If $\Delta T_y \geq T_0$ or $\Delta T_z \geq T_0$, then this communication is terminated. If both are valid, then O checks the received commands K' and K that were sent in Step A2. If they are correct, O executes the command K and destroys a pair of session keys (S_{PK_O}, S_{SK_O}).

5.3 The Change ID Phase

C1. - C2. First two Steps are the same as the authentication phase $A1$ and $A2$.

C3. $U_i \longrightarrow O$:
$E_{S_{PK_O}}(T_{i+1},\ T^*_{j+1},\ T_{i+2},\ K',\ old\ ID_i,\ new\ ID_i)$.
When U_i receives the message, he records the current time T_{i+1} with priority, and then computes $D_{S_{PK_O}}[E_{S_{SK_O}}(T_{j+1},\ X)]$ to decrypt and obtains T^*_{j+1} and X. He can derive the command K' from the message X, and then encrypts T_{i+1}, T^*_{j+1}, T_{i+2}, K', $old\ ID_i$ and $new\ ID_i$ with S_{PK_O}, where T_{i+2} is the new current time of U_i. Next, U_i sends it to O.

C4. $O \longrightarrow U_i:\ E_{S_{SK_O}}(T_k,\ new\ ID^*_i)$.
When O receives the message, it begins by recording the current time T_{j+2}, and then computes $D_{S_{SK_O}}[E_{S_{PK_O}}(T_{i+1}, T^*_{j+1}, T_{i+2}, K', old\ ID_i, new\ ID_i)]$ to decrypt and obtain T_{i+1}, T^*_{j+1}, T_{i+2} and K'. Next, O checks that T^*_{j+1} equals to T_{j+1} that was sent by O in Step C2. If they are correct, then O compares the time $\Delta T_y = T_{j+1} - T_{i+1}$ and $\Delta T_z = T_{j+2} - T_{i+2}$ with T_0. If $\Delta T_y \geq T_0$ or $\Delta T_z \geq T_0$, then this communication is terminated. If both are valid, then O checks the received commands K' and K that was sent in Step C2. If they are correct, O changes old ID_i to new ID_i. After that, O encrypts $new\ ID^*_i$ and T_k with S_{SK_O} and sends it to U_i, where T_k is the new current time of O. Next, O destroys a pair of session keys (S_{PK_O}, S_{SK_O}).

C5. When U_i receives the message, he records the current time T_{k+1} with priority, and then decrypts the message with S_{PK_O} and obtains T_k and $new\ ID^*_i$. Next, U_i compares the time $\Delta T_c = T_{k+1} - T_k$ with T_0 that was stored in the memory at the synchronization phase. If $T_0 \geq \Delta T_c$ and $new\ ID^*_i = new\ ID_i$, then U_i destroys session public key S_{PK_O}. If they are invalid, U_i considers that this communication was forged and he sets up $new\ ID'_i$ and a pair of new session keys (S'_{PK_O}, S'_{SK_O}) again through the synchronization phase.

6 Security Analysis

In this section, we briefly explain why the proposed scheme is secure against the man-in-the-middle attack and also more efficient.

In the synchronization phase, U_i and O transact the user's session public key S_{PK_O} and synchronize time T_0 via a secure communication channel. Even though E intercepts the start signal and $E_{S_{PK_O}}(ID_i,\ T_i)$ in Step A1, he cannot change or replace this information, since he neither knows nor is able to intercept S_{PK_O}. This can prevent the man-in-the-middle attack of [12] and [1].

In addition, we substitute the encrypted new ID_i and T_k with S_{PK_O} in our scheme in order to improve security. If E changes the value of newID_i or K to his own value newID_i^* or K^*, respectively, it should be known to U_i in Step C5. This is because U_i compares the time $\Delta T_c = T_{k+1} - T_k$ with T_0 that was stored in the memory at the synchronization phase. If $\Delta T_c \geq T_0$ or $new\ ID_i^* \neq\ new\ ID_i$, then this communication is terminated. Therefore, E cannot obtain new ID_i without S_{PK_O}.

In the Yang-Lee-Hsiao scheme, E can eavesdrop and replace the message $E_{S_{PK_O}}(newID, K)$ with $E_{S_{PK_O}}(newID', K')$ in Step S5. It is impossible to replace this message without knowing S_{PK_O} in our scheme. However, even though it occurs in our scheme, O checks the time difference $\Delta T_y = T_{j+1} - T_{i+1}$ and $\Delta T_z = T_{j+2} - T_{i+2}$ with T_0 in Step C4. If $\Delta T_y \geq T_0$ or $\Delta T_z \geq T_0$, then this communication is terminated. Moreover, U_i checks that the time $T_0 \geq \Delta T_c$ and $new\ ID_i^* =\ new\ ID_i$ in Step C5. If they are invalid, U_i will find out that this communication was forged and he will set up $new\ ID_i'$ and a pair of new session keys (S'_{PK_O}, S'_{SK_O}) again through the synchronization phase. E does not have enough time to eavesdrop and to perform replacement within ΔT_y and ΔT_z. Therefore, E cannot attack our scheme.

Thus, our proposed scheme is more secure against Yang-Lee-Hsiao's man-in-the-middle attack.

7 Cost Comparisons

We compare the computational cost of the Novikov-Kiselev scheme with our proposed scheme in Table **??**. We define some notations and show the comparative results as follows.

- DT: Data Transmission,
- E/D: Encryption/ Decryption,
- SK_O/SK_U: A pair of Session Keys of the Object/ User.

Since the command is formalized and O does not recheck command K' in [7], E is able to replace $E_{PK_O}(newID, K)$ with $E_{PK_O}(newID', K')$ at any time. Moreover, whenever U_i wants to use O, U_i not only has to change his old ID_i, but also has to create and record *new* ID_i in his memory. However, our scheme is able to reduce the time and the storage that were needed in the presence of $newID$, if U_i does not want to change his old ID_i. In the real world, not many people would want to change their IDs frequently. Furthermore, our scheme does not need to generate and store a pair of session keys for the users.

Table 1: Cost comparisons

Novikov-Kiselev Scheme	**Our Authentication Scheme**	**Our Change ID Scheme**
$5DT + 3E + 3D$	$3DT + 3E + 3D$	$4DT + 4E + 4D$
SK_O	SK_O	SK_O
SK_U	X	X

8 Conclusions

In this paper, we propose a new scheme that overcomes the weakness of the Novikov-Kiselev scheme. Our scheme is more efficient; it requires less time and storage space provided that the users do not change their IDs.

References

[1] A. K. Awasthi, "On the authentication of the user from the remote autonomous object," *International Journal of Network Security*, vol. 1, no. 3, pp. 166-167, Nov. 2005.

[2] M. S. Hwang, "Cryptanalysis of a remote login authentication scheme," *Computer Communications*, vol. 22, no. 8, pp. 742-744, 1999.

[3] M. S. Hwang, and L. H. Li, "A new remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28-30, 2000.

[4] W. S. Juang, "Efficient multi-server pasword authenticated key agreement using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 251-255, Feb. 2004.

[5] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770-772, 1981.

[6] C. C. Lee, L. H. Li, and M. S. Hwang, "A remote user authentication scheme using hash functions," *ACM Operating System Review*, vol. 36, no. 4, pp. 23-29, Oct. 2002.

[7] S. N. Novikov and A. A. Kiselev, "The authentication of the user from the remote autonomous object," *4th Siberian Russian Workshop and Tutorials EDM 2003: Section II, Erlagol, NSTU*, pp 137-138, 2003.

[8] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, Feb. 1978.

[9] H. M. Sun, "An efficient remote use authentication scheme with smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 958-961, Nov. 2000.
[10] K. Tan, and H. Zhu, "Remote password authentication scheme based on cross-product," *Computer Communications*, vol. 18, pp. 390-393, 1999.
[11] S. T. Wu and B. C. Chieu, "A note on a user friendly remote authentication scheme with smart cards," *IEICE Transactions on Fundamentals*, vol. E87-A, no. 8, pp. 2180-2181, Aug. 2004.
[12] C. Y. Yang, C. C. Lee, and S. Y. Hsiao, "Man-in-the-middle attack on the authentication of the user from the remote autonomous object," *International Journal of Network Security*, vol. 1, no. 2, pp. 81-83, Sep. 2005.

Minho Kim is currently an assistant professor of Computer Science at Korea Air Force Academy and has also been working in Electrical Engineering and Computer Science at Oregon State University. He received his Ph.D. degree from Oregon State University. Dr. Kim's research interests include algorithms and protocols for cryptography, computer arithmetic, computer and network security, and wireless communications.

Çetin Kaya Koç received his Ph.D. degree from University of California, Santa Barbara. Dr. Koç's research interests are in cryptographic engineering, algorithms and architectures for cryptography, computer arithmetic and finite fields, parallel algebraic computation, and network security. He has co-founded the Workshop on Cryptographic Hardware and Embedded Systems (CHES), and has been an Associate Editor of IEEE Transactions on Computers and IEEE Transactions on Mobile Computing. Dr. Koç has also been working as a consulting engineer with research and development interests in cryptographic engineering and embedded systems for several companies including Intel, RSA Security, and Samsung Electronics. Dr. Ko? is currently on leave from Oregon State University, working as the director of Information Security Research Center at Istanbul Commerce University, Istanbul, Turkey.