The Improvement of YSYCT Scheme for Imbalanced Wireless Network

Jung-Wen Lo

Department of Information Management, National Taichung Institute of Technology
129 Sec. 3, San-min Rd., Taichung, Taiwan 404, R.O.C. (Email: asalo@ntit.edu.tw)

(Received July 18, 2005; revised and accepted Aug. 16, 2005)

Abstract

Recently, Yeh et al. proposed an improved password authenticated key exchange scheme (YSYCT scheme) which is secure against undetectable on-line password guessing attacks and provides the explicit key authentication. In this article, readers can understand that the YSYCT scheme still is insecure and the user's password can be exposed by man-in-the-middle attack. Besides, an improved protocol is proposed to avoid this attack.

Keywords: Authentication, information security, key exchange, password, RSA

1 Introduction

The symmetric encryption is the best choice to improve the efficiency of encryption because of its encrypting/decrypting speed, but the key distribution and key management in the symmetric encryption are the problems. In 1976, Diffie and Hellman proposed a key exchange scheme allowing two participants to establish a shared secret key over an insecure network [4]. This scheme offered a way to solve the problems. However, the scheme was vulnerable to the man-in-the-middle attack because of the lack of the participants' authentication.

After Bellovin-Merritt proposed password-based key exchange scheme in 1992 [2], the password authenticated key exchange (PAKE) techniques are widely discussed [3, 5, 8, 10, 11]. Luck proposed an open key exchange scheme based on RSA cryptography in 1997 [6], but his scheme was insecure against e-residue attack. Later, MacKenzie et al. improved the drawback with large prime [7]. However, it resulted in a heavy computation, so Zhu et al. proposed a password-based authenticated key exchange protocol based on RSA scheme for imbalanced wireless network [11]. Later, Yeh et al. found Zhu's scheme was insecure against undetectable on-line password guessing attacks and did not provide the explicit key authentication which could guarantee the exchanged key being computed by both participants. Therefore, they proposed an improved scheme to solve the problems [10]. However, their scheme is insecure against off-line password guessing attack which was discussed in paper [3] and [8]. In this article, we shows that their scheme is still insecure against man-in-the-middle attack, and an improved scheme is proposed.

The organization of the remainder of this paper is described as follows. The brief review of the YSYCT protocol and the weakness of the scheme are stated in Section 2. In Section 3, an improved scheme was proposed to keep the attack off. In Section 4, the discussions of security and efficiency improvement are given. The last section is the conclusion.

2 The Weakness of YSYCT Scheme

In this section, the YSYCT scheme is first reviewed [10] and the weakness of their scheme is shown next. Some notations used throughout this article are shown in Table 1.

2.1 The YSYCT Scheme

The YSYCT protocol is shown in Figure 1 and the steps of the protocol are briefly described as follows:

Step 0: Both A and B share a password pw.

Step1: After A generates a RSA public key pair and selects a random number r_A, A sends (n, e, r_A) to B.

Step 2: B checks the validity of the public key by using an interactive protocol through sending out N messages and verifying the return results. If any return message is invalid, B rejects the connection. Otherwise, B selects a number s_B and computes $\pi = E_{pw}(ID_A, ID_B, r_A, s_B)$. Then, B sends $z = \pi^e \bmod n$ to A.

Table 1: Notations to be used throughout in this article

A	a server
B	a low power client
ID_A and ID_B	the identities of A and B, respectively
pw	a password shared between A and B
(n, e)	a RSA public key
m_i	the i_{th} testing message for interactive protocol
N	the total number of testing messages for interactive protocol
E_K, D_K	the symmetric encryption and decryption algorithms defined by a symmetric key K
H, G_1, G_2, h	the distinct hash functions

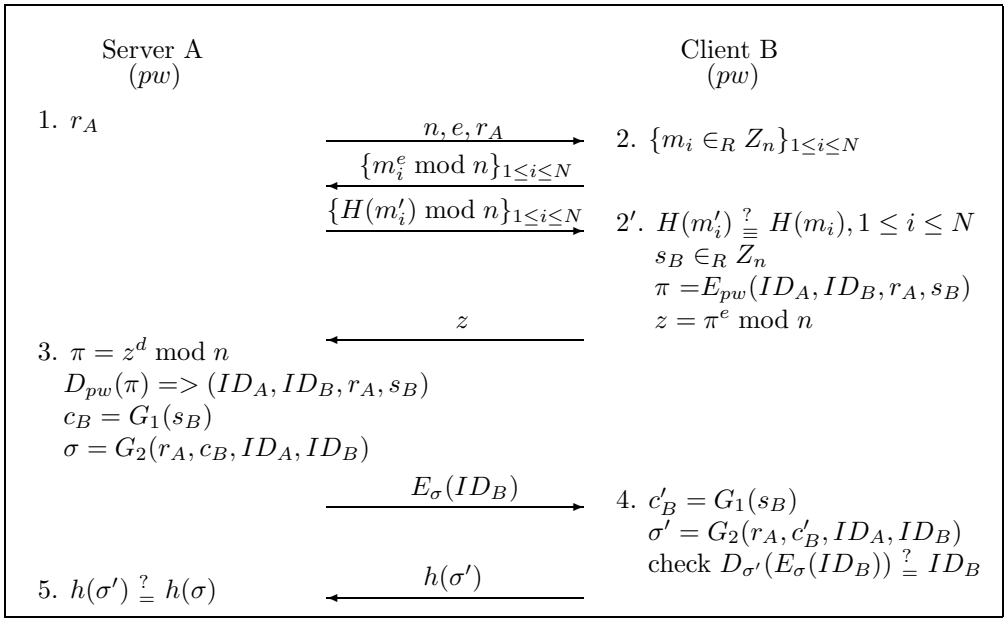

Figure 1: The YSYCT protocol

Step 3: A decrypts z to obtain π and then computes the value of s_B by decrypting π. Next, A computes its session key contribution $c_B = G_1(s_B)$ and the session key $\sigma = G_2(r_A, c_B, ID_A, ID_B)$ afterward. Then, A sends the cipher $E_\sigma(ID_B)$ to B.

Step 4: After computing $c'_B = G_1(s_B)$ and the session key $\sigma' = G_2(r_A, c'_B, ID_A, ID_B)$, B decrypts the $E_\sigma(ID_B)$ by key σ' and checks if it contains B's identity ID_B. B rejects the connection if it is false. Otherwise, B sends $h(\sigma')$ to A.

Step 5: A accepts the connection only if the $h(\sigma)$ computed by A is identical to the incoming data $h(\sigma')$.

2.2 The Man-in-the-middle Attack of YSYCT Scheme

Assuming an attacker C can not only listen the communication between server A and client B, but also delete the message or modify the message transmitted between A and B. C can imitate A or B to send out message. Herein, the attacker C finds out the common password shared between A and B was described.

Step 0: A password pw is shared between A and B but the attacker C does not have any knowledge about it.

Step 1: Attacker C intercepts the RSA public key (n, e) and random number r_A from the message which is sent from A to B. In the meanwhile, C generates a new RSA public key (n', e') and delivers them with the number r_A to B.

Step 2: B begins to execute the interactive protocol with sending out the first message m_1 encrypted by key (n', e') to check the validity of the keys of RSA. Attacker C intercepts the message $\{m_1^{e'} \bmod n'\}$ and decrypts the message to obtain m_1. Next, C encrypts m_1 by key (n, e) before sending the cipher to A. In the meanwhile, A does not detect the

message was falsified by C so A will send out the corresponding hashed message $H(m_1)$ to C. Finally, C passes $H(m_1)$ to B directly for finishing the first round. After performing the interactive protocol with message m_i where $i = 1$ to N, B selects a number s_B and computes $z = \pi^{e'} \bmod n'$ where $\pi = E_{pw}(ID_A, ID_B, r_A, s_B)$. Then, B sends z to C.

Step 3: C obtains π from the equation $z^{d'} \bmod n'$ and then sends $z' = \pi^e \bmod n$ to A. Server A executes the same procedures as Step 3 of Yeh et al.'s scheme. After receiving the message $E_\sigma(ID_B)$, attacker C can proceed the password guessing as follows.

1) Pick up a password pw' from the password pool.
2) Compute $D_{pw'}(\pi)$ to obtain s'_B.
3) Compute the key $\sigma'' = G_2(r_A, c'_B, ID_A, ID_B)$ where $c'_B = G_1(s'_B)$
4) Execute the equation $D_{\sigma''}(E_\sigma(ID_B))$ to obtain a value ID'_B.
5) If ID'_B equals to the B's identity ID_B, the pw' is the common password shared between A and B. Otherwise, C picks up another password from the password pool and repeats above steps till finding out the password.

Because the password is memorial and limited in a small character pool, attacker C can figure out the correct one off-line easily.

Step 4: To prevent A and B from recognizing the existence of C, C passes the message $E_\sigma(ID_B))$ to B and then passes the $h(\sigma')$ to A to finish the protocol.

In this main-in-the-middle attack, attacker C imitates two roles: one is the server A during communicating with client B and the other is the client B during communicating with server A so that server A and client B do not recognize the attacker existed. Also, attacker C can obtain the password shared between A and B easily because the password pool is too small for modern technology.

2.3 The Off-line Guessing Attack of YSYCT Scheme

[3, 8] indicated that attacker C could pose as server A and find out the common password of the server A and client B. The details are described in the following statements.

Step 0: A password pw is shared between A and B but the attacker C does not have any knowledge about it.

Step 1: C generates a RSA public key pair (n', e') and selects a random number r'_A. Then, C sends n', e', r'_A to B.

Step 2: B performs an interactive protocol to check the validity of the keys of RSA. B rejects the connection if any message is incorrect. Otherwise, B selects a number s_B and then computes $z = \pi^{e'} \bmod n'$ where $\pi = E_{pw}(ID_A, ID_B, r'_A, s_B)$. Next, B sends z to C.

Step 3: C computes the corresponding π with $z^{d'} \, mod \, n'$ and obtains the values of $(ID'_A, ID'_B, r''_A, s'_B)$ by decrypting π with a guessing password pw'. If the $ID'_A = ID_A$, $ID'_B = ID_B$ and $r''_A = r'_A$, C discovers the correct password of client B. Otherwise, C continuously finds B's password off-line by choosing another password from the password pool and decrypts π to verify content again until C finds out the correct password.

In this case, attacker C poses as the server A and obtains the common password shared between server A and client B off-line. However, attacker C does not finish the protocol, so the client B could possibly figure out something wrong unless C sends $E_\sigma(ID_B)$ to B.

3 The Improved Scheme

In this section, an improved scheme is proposed in order to avoid the man-in-the-middle attack. The proposed protocol is illustrated in Figure 2 and the steps are stated in the following.

Step 0: Both A and B share a common password pw.

Step 1: A sends a public key (n, e) and $r_A \bigoplus pw$ to B where r_A is a random number chosen by A.

Step 2: B uses the interactive protocol to check the validity of the keys of RSA. Next, B selects a number s_B and computes $\pi = ID_A||ID_B||E_{pw}(r_A \bigoplus s_B)$, and then sends $z = \pi^e \bmod n$ to A.

Step 3: First, A obtains the value of $E_{pw}(r_A \bigoplus s_B)$ from π where $\pi = z^d \bmod n$, and then computes the equation $D_{pw}(E_{pw}(r_A \bigoplus s_B)) \bigoplus r_A$ to obtain the value of s_B. Next, A computes its session key contribution $c_B = G_1(s_B)$ and then the session key $\sigma = G_2(r_A, c_B, ID_A, ID_B)$. Then, A sends out the cipher $E_\sigma(ID_B)$ to B.

Step 4: In the mean while, B computes $c'_B = G_1(s_B)$ and the session key $\sigma' = G_2(r_A, c'_B, ID_A, ID_B)$. When receiving the cipher $E_\sigma(ID_B)$, B decrypts the cipher with $D_{\sigma'}(E_\sigma(ID_B))$ and checks if it contains B's own identity ID_B. B rejects the connection if it is false. Otherwise, B sends $h(\sigma')$ to A.

Step 5: A computes the $h(\sigma)$ and accepts the connection only if the result is identical to the incoming data $h(\sigma')$. Otherwise, A terminates the protocol with failure.

In this improvement, attacker C cannot intercept the message and modify it because C cannot figure out the r_A and s_B through the message flow. Therefore, C cannot execute the man-in-the-middle attack.

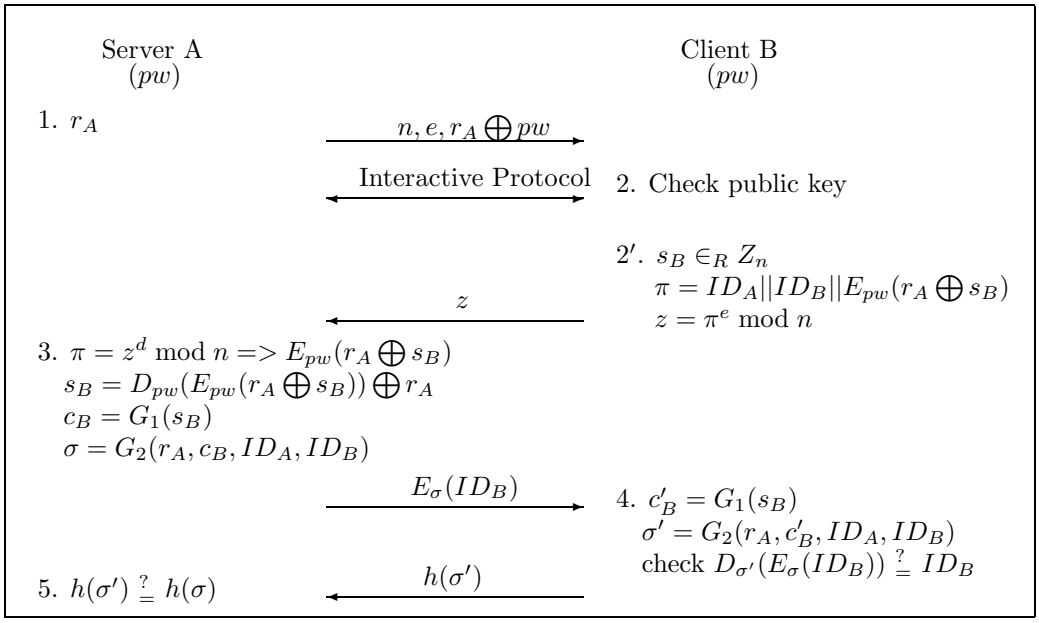

Figure 2: The improved protocol

4 Discussions

In this section, the improvement of the proposed scheme in security and efficiency is discussed.

Man-in-the-middle attack prevention
The flaw of YSYCT scheme in the man-in-the-middle attack is that the random number r_A is not protected well, so an attacker can derive the s_B from the π by this plain r_A. In the improved scheme, when the attacker C intercepts $(n, e, r_A \bigoplus pw)$ that A wants to send to B, C cannot obtain r_A due to lack of real password. Therefore, C only can pass $r_A \bigoplus pw$ and along with the fake public key (n', e') to B. When receiving z from B, C can decrypt z to obtain π but C still cannot have the value of r_A and s_B. Even after receiving the $E_\sigma(ID_B)$ from A, C still cannot obtain the value of r_A and s_B. Because there are two unknown numbers r_A and s_B inside the variable π, the attacker cannot obtain correct s_B for computing key σ.

Impersonation attack prevention
When the attacker C impersonates the server A to send out the message $r_A \bigoplus pw'$, the client B will have different value of r_A which is not the same as A chosen. Therefore, after receiving the message z sent from B, C still have a problem to derive s_B from the data π because C does not know the pw. Therefore, the attack can be avoided.

When the attacker C impersonates the client B, C will have a different value of r_A due to lack of the real password pw. The server A cannot obtain the s_B chosen by C from $E_{pw'}((r_A \bigoplus pw) \bigoplus pw' \bigoplus s_B)$. Therefore, C only receives an unexpect data $E_\sigma(ID_B)$. If C guesses the password by interactive communication, A can detect it easily.

In fact, the impersonation attack could be possibly successful because the password pool is too small to prevent attacker's guessing and the probability of correct choosing is greatly increase. In the PAKE-liked schemes, the common password is the only information shared between the server and client. To authenticate each other within this information is very difficult because they should exchange some information relative to the original sending data which should include the common password for authentication. Therefore, this kind of schemes is insecure against the exhaustive password guessing attack and results in the password exposed. The best idea is to separate the scheme into two parts: authentication protocol and key exchange protocol. If an authentication protocol is executed before processing key exchange protocol, the security of the PAKE-liked scheme will be highly improved. For example, running the PAKE-liked scheme under the public key infrastructure (PKI) system. The trusted third party publishes the public keys of involvers so that both server and client can authenticate each other. Thus, the interactive protocol can be ignored for reducing communication cost and the impersonation attack cannot be happened. Therefore, the proposed scheme can resist against the impersonation attack only if participators already authenticate each other before the scheme is executed.

Length of ID_B
For security reason, the length of ID_B should be greater than the length of password, such as 20 bytes or more because Bao had proposed a paper [1] to analyze the security of Zhu et al.'s scheme. In the paper, attacker C impersonates server A and uses the interactive way to guess the password. It could be successful if the length of password is too short. The YSYCT scheme is also based on Zhu et al.'s scheme, so the attack could also be successful. To

prevent probability attacks, the length of the ID_B should be large enough.

Interactive Protocol Improvement
The Interactive Protocol was proposed by Bellovin and Merritt for preventing e-residue attacks [2]. In 2003, Wong et al. proposed two efficient methods for an imbalanced wireless network [9]. Two methods they used were to reduce the frequency of transmission: only one time transmission from server to client for efficiency reasons and only one time transmission from client to server for battery power saving. Their methods still can apply in the proposed scheme for improving the efficiency.

5 Conclusion

Yeh et al. proposed a new password authenticated key exchange scheme for providing the explicit key authentication which actually guarantees the exchanged key being computed by both parties. Nevertheless, their scheme is still vulnerable. The way of an attacker intercepts the message to play man-in-the-middle attack and discloses the client's password off-line are described in this article. An improved scheme was proposed to resist the man-in-the-middle attack. Besides, the security and efficiency of the improved scheme were discussed. Furthermore, if the participators can make sure who is communication with, the proposed scheme can resist against the password guessing attack.

Acknowledgements

The author would like to thank the anonymous reviewers for their comments that significantly improved this paper.

References

[1] F. Bao, "Security analysis of a password authenticated key exchange protocol," in *Proceeding of ISC 2003*, LNCS 2851, pp. 208–217, Springer-Verlag, 2003.

[2] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," in *Proceedings of 1992 IEEE Computer Society Conference on Research in Security and Privacy*, pp. 72–84, 1992.

[3] Y. F. Chang, C. C. Chang, and J. H. Yang, "An efficient password authenticated key exchange protocol for imbalanced wireless networks," *Computers standards & Interfaces*, vol. 27, pp. 313–322, 2005.

[4] W. Diffie and M. Hellman, "New directions in cryptology," *IEEE Transations on Information Theory*, vol. IT-22, no. 6, pp. 644–654, 1976.

[5] Eric J. L. Lu, C. C. Lee, and M. S. Hwang, "Cryptanalysis of some authenticated key agreement protocols," *International Journal of Computational and Numerical Analysis and Applications*, vol. 3, no. 2, pp. 151–157, 2003.

[6] S. Luck, "Open key exchange: How to defeat dictionary attacks without encrypting public keys," in *Proceedings of Security Protocols Workshop*, LNCS 1361, pp. 79–90. Springer-Verlag, 1997.

[7] P. MacKenzie, S. Patel, and R. Swaminathan, "Password-authenticated key exchange based on RSA," in *ASIACRYPT 2000*, LNCS 1976, pp. 599–613, Springer-Verlag, 2000.

[8] S. Wang, F. Bao, and J. Wang, "Security analysis on an improvement of RSA-based password authenticated key exchange," *IEICE Fundamental Theories for Communications*, vol. E88-B, no. 4, pp. 1641–1646, 2005.

[9] D. S. Wong, A. H. Chan, and F. Zhu, "More efficient password authenticated key exchange based on RSA," in *INDOCRYPT 2003*, LNCS 2904, pp. 375–387, Springer-Verlag, 2003.

[10] H. T. Yeh, H. M. Sun, C. T. Yang, B. C. Chen, and S. M. Tseng, "The improvement of password authenticated key exchange scheme based on RSA for imbalanced wireless network," *IEICE Transactions on Communications*, vol. E86-B, no. 11, pp. 3278–3282, 2003.

[11] F. Zhu, D. S. Wong, A. H. Chan, and R. Ye, "Password authentication key exchange based on RSA for imbalance wireless networks," in *The 5th International Information Security Conference*, LNCS 2433, pp. 150–161, Springer-Verlag, 2002.

Jung-Wen Lo was born on February 21, 1964 in Taiwan. He received the B.S. degree in Information and Computer Engineering in 1987 from Chung Yuan Christian University, Chung-Li, Taiwan and the M.S. degree in Computer Science & Information Systems in 1994 from Texas A&M University at Commerce, Texas, U.S.A. He is working for his Ph.D. program in the Department of Computer Science at National Chung Hsing University, Taichung, Taiwan. He was an Associate Engineer in Product Development Department of Institute for Information Industry from 1994 to 1996. Since August 1998, he has been an Instructor of the Department of Information Management at National Taichung Institute of Technology, Taichung, Taiwan. His research interests include electronic commerce, computer cryptography and computer networks.