An Advanced Password Authenticated Key Exchange Protocol for Imbalanced Wireless Networks

Jung-Wen Lo[1,2], Ji-Zhe Lee[3], Min-Shiang Hwang[4], Yen-Ping Chu[5]
[1]Department of Computer Science and Engineering, National Chung Hsing University, Taiwan
[2]Department of Information Management, National Taichung Institute of Technology, Taiwan
[3]Customer Service Systems Laboratory, Telecommunication Laboratories, Chunghwa Telecom Co., Ltd., Taiwan
[4]Department of Management Information Systems, National Chung Hsing University, Taiwan
[5]Department of Computer Science & Information Engineering, Tunghai University, Taiwan
asalo@ntit.edu.tw, licc@cht.com.tw, mshwang@nchu.edu.tw, ypchu@thu.edu.tw

Abstract

Less computational load and better efficiency on password authenticated key exchange between a low-power client and a powerful server is very important, especially in a wireless communication environment due to the constraints on the electricity supply. The proposed scheme is based on the RSA cryptography and the concept of key exchange. It reduces the load of computation on clients to fit the low-power property of clients without yield the security. Through the comparisons with other protocols, we show that the proposed scheme is more suitable to apply in imbalanced wireless communication environment.

Keywords: Key exchange, Authentication, Password, RSA, Network security, Wireless communications.

1 Introduction

Owing to the popularity of Internet and the benefit of client-server architecture, lots of servers are established and provide variant services within the network. Due to the convenience of use and the advanced development of wireless technology, more and more clients log in to servers over the wireless communication. The mobile clients usually suffer from the limitation of electronic power, computation ability and memory capacity. This kind of network with highly different computation ability between mobile clients and servers was named imbalanced network. In essence, to design an efficient and secure mechanism for the imbalanced networks would be very important.

To guarantee the security of the communication, a proper cryptosystem should be seriously considered. To prevent illegal clients from using the services of a server, anyone who logs in to the server should be authenticated first [20][23]. The simplest way is to authenticate the logged in user by utilizing a common datum, such as a password [10-11]. In addition, to avoid the information leaking and to achieve the efficient computation, both the server and the client could utilize a common session key to encrypt or decrypt the transmitting data. It is insecure to use the password as a session key directly because the length of a password usually is short so that it is easy to memorize. In addition, using the password as a long-term secret key is easily cracked by a malicious user. Using a one-time session key generated by the key exchange mechanism is the simplest method for a securer communication [24].

In 1976, Diffie-Hellman proposed a concept of the key exchange which allows both participants to construct a common session key without leaking any session key information during the transmission [4]. It is vulnerable for unknown-key attack due to the lack of authentication [5][7][13]. Some scholars proposed authentication mechanism for key agreement protocol [3][9][15][25]. The Bellovin and Merritt proposed an integrated scheme named EKE (Encrypted Key Exchange) which used a common password as a common secret key to reach the goal of the key exchange and the mutual authentication [2]. To improve the efficiency and security, many literatures based on the password authenticated key exchange were proposed [6][27][29]. Jablon proposed a strong password-only authenticated key exchange scheme in 1996 [8]. MacKenzie, Patel and Swaminathan proposed a RSA-based scheme later [19]. To improve the security, Seo and Sweeney proposed a simpler scheme to prevent the man-in-middle attack in 1999 [21]. Later, Lu and Hwang proposed an improved scheme [17]. However, both Seo-Sweeney scheme and Lu-Hwang's scheme are insecure [18]. According to our survey, these schemes are not perfect for a wireless environment until Zhu et al.'s proposed a new scheme in 2002 which was based on the Bellovin-Merritt's scheme and gave a new direction for imbalanced wireless network [29]. In 2003, Yeh et al. pointed out the Zhu's scheme would not be able to resist against undetectable on-line password guessing attacks [27]. However, Yeh et al.'s improved scheme has a serious security problem to resist against the man-in-the-middle attack and off-line password guessing attack [16][26][28]. Hsu, Lin and Chou proposed a performance improved scheme in 2007 [6].

*Part of this article is published in Proceedings of MC2010 (The 15th Mobile Computing Workshop).

*Corresponding author: Min-Shiang Hwang; E-mail: mshwang@nchu.edu.tw

Now, we proposed a more efficient scheme on the securer communication, lower computation load and less memory use for the low-power clients.

The organization of this paper is described as follows. The Yang-Wang's protocol and Hsu-Lin-Chou's protocol are briefly introduced in Section 2. In Section 3, a detailed description of our protocol is presented. The discussions on property of session key and security are stated in the Section 4 and the comparison results are exhibited in Section 5. The last section is our conclusions.

2 Related Works

In this section, we introduce the Yang-Wang's protocol and Hsu-Lin-Cho's efficient protocol. The notations used throughout the article are shown in Table 1.

2.1 Review of Yang-Wang's Protocol

Yang-Wang's protocol is based on the Yeh et al.'s protocol and only modifies the first four steps. The brief steps are listed as follows.

Step 1 Client C ⇒ Server S: {Request} – The Client C requests a service to Server S.

Step 2 Server S ⇒ Client C: {ω} – Server S generates a RSA keys (n, e, d) and a random integer r_S. Then, S sends the number $\omega=(e||n||r_S) \oplus h_1(pw)$ to Client C.

Step 3 Client C ⇒ Server S: {$(m_i||r_S)^e$ mod n} – C obtains the n, e, r_s from the ω and encrypts the ith message slice m_i with S's RSA public key pair where message m consists of the slices $m_1, m_2, \cdots, m_N$. C sends out the ith encrypted message slice with random r_S to S where i=1 to N.

Step 4 Server S ⇒ Client C: {$h_1(m_i)$} – S decrypts the data to obtain the message slice m_i and then transmits the hash result to C.

Step 5 Client C ⇒ Server S: {z} – If the verifications of all $h_1(m_i)$ are correct, C randomly chooses r_C and computes the $\pi=E_{pw}(ID_S, ID_C, r_S, r_C)$. Next, C sends z to S where $z=\pi^e \bmod n$.

Step 6 Server S ⇒ Client C: {σ} – S decrypts the z to obtain $\pi=z^d \bmod n$ and computes the $D_{pw}(\pi)$ to obtain the r_C. Next, S computes the session key $R=h_3(r_S, c_C, ID_S, ID_C)$ where $c_C = h_2(r_C)$ and then sends out the $\sigma=E_R(ID_C)$ to C.

Step 7 Client C ⇒ Server S: {δ} – After computing the $c'_C = h_2(r_C)$ and session key $R'=h_3(r_S, c'_C, ID_S, ID_C)$, C Decrypts the σ to have the $ID'_C = D_{R'}(\sigma)$. If the data ID'_C equalizes the ID_C, C delivers the data $\delta=h_4(R')$ to S or terminates the protocol if the result is not hold. S computes the data $\delta'=h_4(R)$ and verifies the equality of δ and δ'. S terminates the protocol if the equality is not hold.

2.2 Review of Hsu-Lin-Chou's Protocol

The purpose of the Hsu-Lin-Chou's protocol is to improve the efficiency of Yang-Wang's protocol. The steps are listed as follows.

Step 1 Client C ⇒ Server S: {Request} – The Client C requests a service to Server S.

Step 2 Server S ⇒ Client C: {ω} – Server S generates a RSA keys (n, e, d) and a random integer r_S. Then, S sends the number $\omega=E_{pw}(e||n||r_S)$ to Client C.

Step 3 Client C ⇒ Server S: {$(m_i)^e$ mod n} – C decrypts the ω to obtain the (n, e, r_S) for later using. Then, C encrypts the ith message slice m_i with S's RSA public key pair where message m consists of the slices $m_1, m_2, \cdots, m_N$. C sends out the ith encrypted message slice to S where i=1 to N.

Step 4 Server S ⇒ Client C: {$h_1(m_i)$} – S decrypts the data to obtain the message slice m_i and then transmits the hash result to C.

Table 1 Notations of This Article

S	The powerful server
C	The low-power client
ID_S, ID_C	The identities of S and C, respectively
pw	The password shared between S and C
(n, e) / d	The RSA public key/private key of server S
m_i	The i-th message slice of message m for interactive protocol where $m = m_1||m_2||\ldots||m_N$
N	The total number of testing messages for interactive protocol where N is a security parameter
E_K , D_K	The symmetric encryption and decryption algorithms defined by a symmetric key K
R	The secret session key which is exchanged between S and C
C_C	The challenge data generated with C's data
r_S, r_C	The random number chosen by S and C, respectively
h, h_1, h_2, h_3, h_4	The distinct hash functions
X ⇒ Y: {Z}	The host X sends data Z to Y

Step 5 Client C ⇒ Server S: {σ, z} – If the verifications of all $h_1(m_i)$ are correct, C randomly chooses r_C and computes the $z = (r_C \oplus pw \oplus r_S)^e \bmod n$, $R = r_S \oplus r_C \oplus (ID_S||ID_C)$, $\sigma = h(r_S||r_C||ID_S||ID_C||R)$. Next, C sends σ, z to S.

Step 6 Server S ⇒ Client C: {δ} – S decrypts the z to obtain $r_C \oplus pw \oplus r_S$ and XOR the result value with $pw \oplus r_S$ for obtaining the r_C. Next, S computes the $R' = r_S \oplus r_C \oplus (ID_S||ID_C)$ and checks if $h(r_S||r_C||ID_S||ID_C||R')$ equals the received σ. If the values are not the same, S terminates the protocol. Otherwise, S sends the δ to C where the $\delta = h(R')$. C validates the receiving value with the h(R) and only accepts when they are equal.

3 The Proposed Protocol

Our protocol offers robustness and better performance. The protocol is shown in Figure 1 and the detail steps are stated as follows.

Step 1 Client C ⇒ Server S: {Request} – The Client C sends a service request message to Server S.

Step 2 Server S ⇒ Client C: {ID_S, n, e, ω} – The Server S generate a RSA key pair by using a public key generator, where the public key is (n, e) and private key is d, n is the product of two large primes and $ed = 1 \bmod \varphi(n)$. Next, S chooses a random integer $r_S \in_R \{0, 1\}^l$ where l is a set of all length l bits binary string. After S computes the $\omega = r_S \oplus h(pw)$, S sends the $\{ID_S, n, e, \omega\}$ to C.

Step 3 Client C ⇒ Server S: {$(m_i)^e$ mod n} – C encrypts a message slice m_i with S's RSA public key pair where message m consists of the slices $m_1, m_2, \cdots, m_N$ and m_i is the ith message slice of m. Then, C sends out the encrypted message slice to S.

Step 4 Server S ⇒ Client C: {$h(m_i)$} – S decrypts the ciphertext $\{m^e \bmod n\}$ with S's private key to obtain m_i. In order to avoid the e-residue attack described in Section 4, S transmits the hash result $h(m_i)$ to C.

Step 5 Client C ⇒ Server S: {z} – C randomly chooses an integer $r_C \in Z_n$ and computes $r'_S = \omega \oplus h(pw)$. After computing $z = (r'_S||r_C)^e \bmod n$, C sends z to S.

Step 6 Server S ⇒ Client C: {δ} – S decrypts the received z with its private key d to obtain $(r''_S||r'_C)$. S authenticates C if the r''_S equals r_S. Otherwise, S terminates the protocol. Next, S computes the session key $R = h(r_S, r'_C, ID_S, ID_C)$ and sends the δ to C where $\delta = h(R \oplus r'_C)$. C checks the equality of the received δ and $h(R' \oplus r_C)$ where $R' = h(r'_S, r_C, ID_S, ID_C)$. If it holds, C validates S and both S and C agrees the session key. Otherwise, C terminates the protocol.

Because there is only one hashing function employed, the mobile client can have less memory usage than other schemes. Besides, the XOR function not only has a better efficiency but also can be implemented by hardware. Therefore, our protocol definitely has a better performance for low-power client.

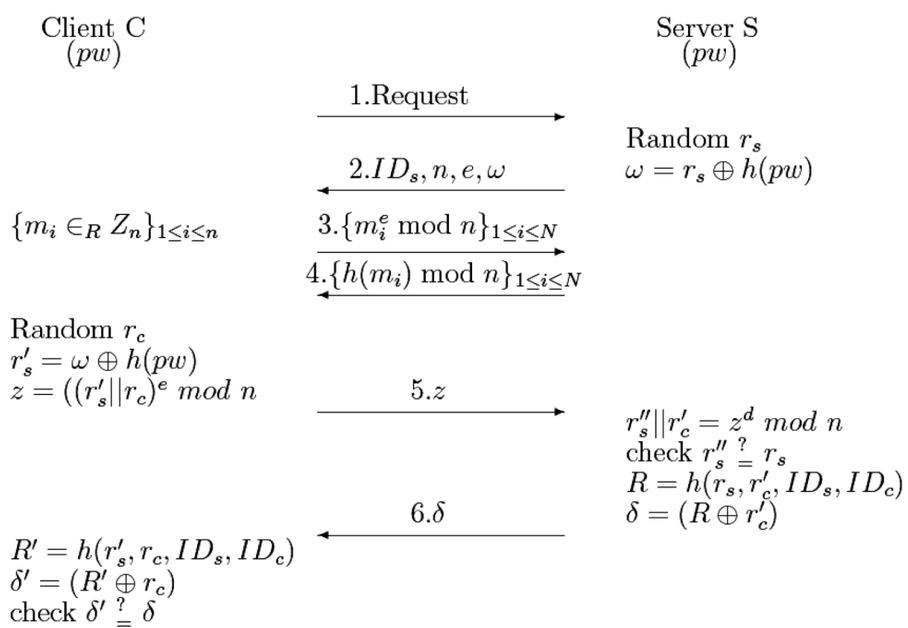

Figure 1 RSA-Based Key Agreement Protocol for Imbalanced Wireless Network

4 Discussions

In this section, we state the properties of our session key and then show out the security discussions.

4.1 Discussions on Property

A session key should satisfy the following properties: mutual authentication, perfect forward/backward secrecy, known-key security, key control security, key verification and session key security. We show that our protocol satisfies these properties.

- **Mutual Authentication**

The two entities can authenticate each other to make sure that they communicate with the correct entity. In Step 6 of our protocol, the server S authenticates client C if C sends back the correct random number r_S which is chosen by S in Step 2. Client C authenticates Server S in Step 6 when S uses the correct r_C to generate the δ. Therefore, the protocol satisfies the property of mutual authentication.

- **Perfect Forward/Backward Secrecy**

When one session key is compromised, any attacker cannot have or cannot derive any following/previous session key. Because the random number r_S and r_C are different in every session, it is impossible to derive a following/previous session by using the current compromised session key.

- **Known-Key Security**

The known-key security is that the session key generated in every session is unique. The reason is the same as the above property. Each session has different random numbers, r_S and r_C, for generating the session key. In other words, every session key is different from any other previous session key. Therefore, we guarantee the property.

- **Key Control Security**

The property of key control is to prevent either entity from deciding the key value. The session key is generated by two random numbers r_S and r_C which is chosen by each entity individually. No one can directly decide the session key.

- **Key Verification**

When a session key is set, it should be verified by both participants. In our protocol, client can verify the validity of session key R from the message δ in Step 6.

- **Session Key Security**

The session key should be known by the communicated entities only. The session key is generated by the identity of the entities and the random numbers r_S and r_C. The attacker can only obtain the information from the transmission flows, ω, z and δ. It is hard to have the random numbers because z is protected by the problem of computing discrete logarithms (DLP) and ω, δ are protected by the hash function. Without the correct random number, the attacker cannot have the session key. Therefore, the session key security is satisfied.

4.2 Discussions on Security

In this subsection, we discuss the security of our protocol. In 2003, Bao indicated that Zhu et al.'s protocol might be exist two attacks when the bit length of C's identity was not too large or less than the length of the password [1]. Because both attacks may need relatively large amounts of computation and therefore it can be avoided. We assume the situations will not happen in our scheme, such as define the length of identity to ensure it is long enough and also longer than the length of password.

- **E-residue Attack**

The e-residue attack is an active attacker who impersonates Server S to choose a RSA public key pair (n, e) where e is not relatively prime to φ(n), and then, the adversary uses e-residue check to remove impossible passwords from the password space. However, it could be avoided through the verification by checking if e is relatively prime to φ(n). There are two methods to enforce the GCD(φ(n),e) = 1 [19]. One is to set e a prime which is greater than n. The other is to set e a prime which is greater than √n and (n mod e) cannot divide n. However, both methods should have a large prime e and they are not suitable in wireless environment. Therefore, we still adopt the interactive method to figure out the result of GCD(φ(n),e) = 1. The probability that the attacker guesses all the e-th roots correctly is at most 3^{-N} [29]. Therefore, even though client C has no idea about the φ(n), client C still can verify e by using the interactive way in sufficient rounds. The purpose of Step 2 and Step 3 in the protocol is to prevent this attack.

- **Replay Attack**

An attacker replays the message which he intercepts before. The common password is shared by server and client only, and others have no knowledge about it. The purpose of the password we employed is not only to authenticate entity but also to protect the transmitting message. If an attacker replays the message ω, the attacker will receive a new z back which is different from the previous old z. In this case, the old session key is not working in this session and will be detected in Step 6 by the client. When an attacker replays the message z, the attacker will be detected by the server in Step 6 because the new ω is different from the previous one. Therefore, this attack can not succeed.

- **Impersonation Attack**

The mutual authentication is processed in the three way communications at Steps 2, 5 and 6. When an attacker tries to impersonate a legal server, he can not generate a proper ω in Step 2 because he lacks the correct password.

Therefore, he can not obtain the correct random number r_C from the message z and will be detected by client in Step 6. In the other case, when an attacker tries to impersonate a legal client, he only can send out an improper z in Step 5 and the server cannot derive the correct r_S for verification. Therefore, this attack can be avoided within our protocol.

- **Man-in-the-Middle Attack**

An adversary intercepts the message flow and tries to impersonate the server and client. However, the adversary does not have the correct common password, so he will fail in the interactive communication. Because every session has a different session key which is well protected by the DLP and hash function, the adversary has no chance to fake any legal client or server.

- **Password Guessing Attack**

A password guessing attack occurs when an unauthorized user tries to guess the password. The unauthorized user can on-line login to a computer repeatedly or off-line to guess from the messages he eavesdrops from the network. We can limit the number of incorrect login to avoid the on-line password guessing attack. In our scheme, the attacker only could be eavesdrop ω, z and δ. The password is hidden in ω, and it XOR with a random number. When an attacker guesses a password to obtain a new r'_s, it still cannot to make sure the value because the correct r_s is well protected by RSA system. Therefore, both the on-line and off-line password guessing attacks fail.

- **Unknown-Key Share Attack**

An entity finishes the key agreement protocol with an entity B together and believes that the session key is shared with the entity B, but B believes that the key is shared with another entity who is not A [5]. This attack happens when entity A and entity B do not mutually authenticate. In our protocol, server authenticates client with the z and client authenticates server with δ, therefore the attack is withstood.

- **Parallel Guessing Attack**

When a server fails to deal with the requests carefully, an attacker can guess some useful data from the messages transmitted from server. It is happened in three-pass/four-pass protocols [12]. All transmitted messages are well designed and there are no relationships between the messages in our protocol so the attack will fail.

- **Reflection Attack**

The reflection attack is an attack happened in a challenge-response authentication system. When an attacker immediately sends back the message which he received from the same sender in the other session, he may obtain the correct session key for this session [22]. This attack is not the same as the replay attack because it uses more then one connections to connect to the target and it does not record the message information for later use. In our protocol, the random number r_s mixed with the h(pw). Without the correct password pw, the attacker cannot have the correct r_s for the other connection to reflect the message. Also, the r_c is protected by the DLP. Therefore, the reflection attack cannot success.

5 Performance Comparisons

The comparisons among our protocol and others are stated in this section.

5.1 Computational Complexity

Some computational time units and bit-length notation of the data for the exhibition of comparisons are defined as follows.

Time relative

- T_e is the time for a RSA cryptosystem to encrypt or decrypt data.
- T_s is the time for a symmetric cryptosystem to encrypt or decrypt data.
- T_h is the time for a one-way hash function process.
- T_a is the time for a modular addition process.
- T_x is the time for an eXclusive OR operation process.

Length relative

- |n| is the bit-length of the modular n.
- |ε| is the bit-length of one block of the ciphertext.
- |h| is the bit-length of the output of a one-way hash function.

Memory relative

- |H| is the size of the hash function in memory.
- |E| is the size of the symmetric encryption function in memory.

It is obvious that our protocol has a less computational complexity in Table 2. Others protocols employee the symmetric cryptography to protect data, but they results in the computational consumption, especially for the low-powered mobile client. Our protocol uses the hash function computation and XOR execution instead of the power-consuming computations, such as symmetric encryption/decryption. Moreover, a symmetric encryption/decryption operation is at least 100 times faster than an asymmetric encryption/decryption operation in software and an exponential operation is approximately equal to 60 symmetric encryptions/decryption operations [14]. Without considering the interactive communication, the Zhu et al.'s protocol almost needs 1042 hash operations, Yeh et al.'s protocol needs 1057 hash operations, Yang-Wang's protocol needs 1058 hash operations, Hsu-Lin-Chou's protocol needs 1039 hash operations and our protocol only needs 1023 hash operations for a session key generation in the client side.

Table 2 Comparisons of Computational Complexities

Protocols	Client	Server
Zhu et al.'s	(N + 1)Te + (N + 5)Th + Ts + Ta	(N + 1)Te + (N + 5)Th + Ts + Ta
Yeh et al.'s	(N + 1)Te + (N + 3)Th + 2Ts	(N + 1)Te + (N + 3)Th + 2Ts
Y-W's	(N + 1)Te + (N + 4)Th + 2Ts + Tx	(N + 1)Te + (N + 4)Th + 2Ts + Tx
Hsu et al.'s	(N + 1)Te + (N + 2)Th + Ts + 4Tx	(N + 1)Te + (N + 2)Th + Ts + 4Tx
Ours	(N + 1)Te + (N + 3)Th + 2Tx	(N + 1)Te + (N + 3)Th + 2Tx

The communication cost depends on the total size of transmission data. The major difference among others protocols and ours is the size of ciphertext which is relative to the size of the input parameters. The parameters for the symmetric encryption in the protocols are $\{r_S, ID_C\}$ in Zhu et al.'s protocol, $\{ID_C\}$ in Yeh et al.'s protocol, $\{ID_S, ID_C, r_S, r_C\}$ in Yang-Wnag's protocol and $\{r_S, n, e\}$ in Hsu et al.'s protocol. From the Table 3, it is obvious that our protocol still has the less communication cost.

Table 3 Comparisons of Communication Costs

Protocols	Communication Costs
Zhu et al.'s	$2\|\varepsilon\| + (N + 5)\|n\| + (N + 1)\|h\|$
Yeh et al.'s	$\|\varepsilon\| + (N + 4)\|n\| + (N + 1)\|h\|$
Y-W's	$4\|\varepsilon\| + (N + 5)\|n\| + (N + 4)\|h\|$
Hsu et al.'s	$3\|\varepsilon\| + (N + 1)\|n\| + (N + 2)\|h\|$
Ours	$(N + 4)\|n\| + (N + 1)\|h\|$

In Table 4, it is obviously that our protocol has less number of transmissions than others, especially in client side. Most protocols need three ways to authenticate their communication party. Our special design decreases the number without losing the property of authentication.

Table 4 Comparisons of the Number of Transmissions without the Interactive Transmission

Protocols	Client	Server	Total
Zhu et al.'s	2	3	5
Yeh et al.'s	2	2	4
Y-W's	2	2	4
Hsu et al.'s	1	2	3
Ours	1	2	3

5.2 Memory Usage

The memory limitation is also a problem of wireless devices. The data stored in memory are almost the same among in all protocols. The major difference on memory usage is the storage size of functions. There are five hash functions and one symmetric cryptosystem in Zhu et al.'s protocol, four hash functions and one symmetric cryptosystem in Yeh et al.'s protocol, one hash function and one symmetric cryptosystem in Yang-Wang's protocol, and one hash function and one symmetric cryptosystem in Hsu et al.'s protocol, but only one hash function was used in our scheme. In Table 5, our protocol has the least memory usage in client side than others.

Table 5 Comparisons of the Memory Usage of Functions in the Client Side

Protocols	Memory Usage
Zhu et al.'s	$5 \cdot \|H\| + \|E\|$
Yeh et al.'s	$4 \cdot \|H\| + \|E\|$
Y-W's	$\|H\| + \|E\|$
Hsu et al.'s	$\|H\| + \|E\|$
Ours	$\|H\|$

6 Conclusions

In the client-server architecture, using a common password to authenticate each other is simple and costless. After Bellovin-Merritt proposed their interactive RSA-EKE protocol, Zhu et al.'s protocol is one of the secure protocols. With an elaborative modification, our scheme is more suitable for low-power clients in a wireless environment. The proposed scheme utilizes less memory and shows a better performance from the comparisons in Section 5. In the discussion of the amount of computations, whether the amount is of the whole system or only focuses on the low-power client, our protocol always surpasses others. Furthermore, the proposed protocol can resist the attacks mentioned in the discussions of Section 4. Therefore, our protocol is more practical to be applied to the wireless communication environments.

Acknowledgements

This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the grants NSC 98-2221-E-005-050-MY3.

References

[1] Feng Bao, *Security Analysis of a Password Authenticated Key Exchange Protocol*, *Proceedings of ISC 2003*, *LNCS*, Vol.2851, 2003, pp.208-217.

[2] Steven M. Bellovin and Michael Merritt, *Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks*, *Proceedings of 1992 IEEE Computer Society Conference on Research in Security and Privacy*, Oakland, CA, May, 1992, pp.72-84.

[3] Ji Young Chun, Jung Yeon Hwang and Dong Hoon Lee, *A Note on Leakage-Resilient Authenticated Key Exchange*, *IEEE Transactions on Wireless Communications*, Vol.8, No.5, 2009, pp.2274-2279.

[4] Whitfield Diffie and Martin E. Hellman, *New Directions in Cryptology*, *IEEE Transactions on Information Theory*, Vol.IT-22, No.6, 1976, pp.644-654.

[5] Whitfield Diffie, Paul C. van Oorschot and Michael J. Wiener, *Authentication and Authenticated Key Exchanges*, *Designs, Codes and Cryptography*, Vol.2, No.2, 1992, pp.107-125.

[6] Chien-Lung Hsu, Wen-Te Lin and Yen-Chun Chou, *New Efficient Password Authenticated Key Exchange Protocol for Imbalanced Wireless Networks*, *Journal of Computers*, Vol.18, No.2, 2007, pp.25-32.

[7] Min-Shiang Hwang, Jung-Wen Lo and Chia-Hsin Liu, *Enhanced of Key Agreement Protocols Resistant to a Denial-of-Service Attack*, *Fundamenta Informaticae*, Vol.61, No.4, 2004, pp.389-398.

[8] David P. Jablon, *Strong Password-Only Authenticated Key Exchange*, *ACM SIGCOMM Computer Communication Review*, Vol.26, No.5, 1996, pp.5-26.

[9] Wen-Shenq Juang and Jing-Lin Wu, *Efficient User Authentication and Key Agreement with User Privacy Protection*, *International Journal of Network Security*, Vol.7, No.1, 2008, pp.120-129.

[10] Manoj Kumar, *An Enhanced Remote User Authentication Scheme with Smart Card*, *International Journal of Network Security*, Vol.10, No.3, 2010, pp.175-184.

[11] Manoj Kumar, *A New Secure Remote User Authentication Scheme with Smart Cards*, *International Journal of Network Security*, Vol.11, No.2, 2010, pp.88-93.

[12] Taekyoung Kwon, *Practical Authenticated Key Agreement Using Passwords*, *LNCS*, Vol.3225, No.5, 2004, pp.1-12.

[13] Cheng-Chi Lee, Min-Shiang Hwang and Li-Hua Li, *A New Key Authentication Scheme Based on Discrete Logarithms*, *Applied Mathematics and Computation Archive*, Vol.139, No.2-3, 2003, pp.343-349.

[14] Jung-San Lee and Chin-Chen Chang, *Secure Communications for Cluster-Based Ad Hoc Networks Using Node Identities*, *Journal of Network and Computer Applications*, Vol.30, No.4, 2007, pp.1377-1396.

[15] Jie Liu and Jianhua Li, *A Better Improvement on the Integrated Diffie-Hellman-DSA Key Agreement Protocol*, *International Journal of Network Security*, Vol.11, No.2, 2010, pp.114-117.

[16] Jung-Wen Lo, *The Improvement of YSYCT Scheme for Imbalanced Wireless Network*, *International Journal of Network Security*, Vol.3, No.1, 2006, pp.39-43.

[17] Eric Jui-Lin Lu and Min-Shiang Hwang, *An Improvement of a Simple Authenticated Key Agreement Algorithm*, *Pakistan Journal of Applied Sciences*, Vol.2, No.1, 2002, pp.64-65.

[18] Eric Jui-Lin Lu, Cheng-Chi Lee and Min-Shiang Hwang, *Cryptanalysis of Some Authenticated Key Agreement Protocols*, *International Journal of Computational and Numerical Analysis and Applications*, Vol.3, No.2, 2003, pp.151-157.

[19] Philip MacKenzie, Sarvar Patel and Ram Swaminathan, *Password-Authenticated Key Exchange Based on RSA*, *Proceedings of ASIACRYPT 2000*, *LNCS*, Vol.1976, 2000, pp.599-613.

[20] Babu B. Sathish and Pallapa Venkataram, *A Dynamic Authentication Scheme for Mobile Transactions*, *International Journal of Network Security*, Vol.8, No.1, 2009, pp.59-74.

[21] Dong Hwi Seo and Peter Sweeney, *Simple Authenticated Key Agreement Algorithm*, *IEE Electronics Letters*, Vol.35, No.13, 1999, pp.1073-1074.

[22] Andrew S. Tanenbaum, *Computer Networks*, Prentice Hall, Upper Saddle River, NJ, 2003.

[23] Jia Lun Tsai, *Efficient Nonce-Based Authentication Scheme for Session Initiation Protocol*, *International Journal of Network Security*, Vol.9, No.1, 2009, pp.12-16.

[24] Shengbao Wang, Zhenfu Cao and Haiyong Bao, *Efficient Certificateless Authentication and Key Agreement (CL-AK) for Grid Computing*, *International Journal of Network Security*, Vol.7, No.3, 2008, pp.342-347.

[25] Shengbao Wang, Zhenfu Cao and Feng Cao, *Efficient Identity-Based Authenticated Key Agreement Protocol with PKG Forward Secrecy*, *International Journal of Network Security*, Vol.7, No.2, 2008, pp.181-186.

[26] Chou-Chen Yang and Ren-Chiun Wang, *Cryptanalysis of Improvement of Password Authenticated Key Exchange Based on RSA for Imbalanced Wireless Networks*, *IEICE Transactions on Communications*, Vol.E88-B, No.11, 2005, pp.4370-4372.

[27] Her-Tyan Yeh, Hung-Min Sun, Cheng-Ta Yang, Bing-Cheng Chen and Shin-Mu Tseng, *The Improvement of Password Authenticated Key Exchange Based on RSA for Imbalanced Wireless Networks*, *IEICE Transactions on Communications*, Vol.E86-B, No.11, 2003, pp.3278-3282.

[28] Taek-Young Youn, Young-Ho Park, Changhan Kim and Jongin Lim, *Weakness in a RSA-Based Password Authenticated Key Exchange Protocol*, *Information Processing Letters*, Vol.108, No.6, 2008, pp.339-342.

[29] Feng Zhu, Duncan S. Wong, Agnes H. Chan and Robbie Ye, *Password Authenticated Key Exchange Based on RSA for Imbalance Wireless Networks*, *Proceedings of The 5th International Information Security Conference*, *LNCS*, Vol.2433, 2002, pp.150-161.

Biographies

Jung-Wen Lo received his PhD in the department of Computer Science at National Chung Hsing University, Taichung, Taiwan. He is an Associate Professor in the Department of Information Management at National Taichung Institute of Technology, Taichung, Taiwan. His research interests include electronic commerce, network security and computer networks.

Ji-Zhe Lee received the MS degree from the Information Management at Chaoyang University of Technology, Taichung, Taiwan. His current research interests include wireless security and information security.

Min-Shiang Hwang received the BS in Electronic Engineering from National Taipei Institute of Technology, Taipei, Taiwan, in 1980; the MS in Industrial Engineering from National Tsing Hua University, Taiwan, in 1988; and the PhD in Computer and Information Science from National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at National Cheng Kung University, Taiwan, from 1984-1986. He was a Chairman and Professor of the Department of Information Management, Chaoyang University of Technology (CYUT), Taiwan, during 1999-2003. He obtained the 1997, 1998, 1999, 2000, and 2001 outstanding research awards of the National Science Council of the Republic of China. He is currently a Professor of the Department of Management Information Systems, National Chung Hsing University, Taiwan. His current research interests include electronic commerce, database and data security, cryptography, image compression, and mobile computing.

Yen-Ping Chu is a Professor in the Department of Computer Science, Tunghai University, Taiwan and the Director of Tunghai University Library. His research interests include distributed systems, computer networks and e-learning. His research interests include high-speed networks, operating system, neural network, and computer assistant learning.