**ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY**
Volume **16**, Number 4, 2013, 324–335

# An Improved Authentication Protocol for Session Initiation Protocol Using Smart Card and Elliptic Curve Cryptography

Kan WU, Peng GONG, Jiantao WANG, Xiaopeng YAN, Ping LI

National Key Laboratory of Mechatronic Engineering and Control,
School of Mechatronical Engineering,
Beijing Institute of Technology, Beijing, China

E-mail: wukan12491@gmail.com, wangjiantao_bit@163.com,
{penggong,yanxiaopeng,liping85}@bit.edu.cn

**Abstract.** The authenticated key agreement protocol is an important security protocol for the session initiation protocol, which allows the and the server to authenticate each other and generate a shared session key for privacy, integrity, and non-repudiation in their communications. Recently, Zhang et al. proposed a new authenticated key agreement protocol for the session initiation protocol using smart card and claimed their protocol was secure against various attacks. However, we found that Zhang et al.'s protocol cannot withstand the user impersonation attack, i. e., a malicious user could impersonate any other user to the server. We also propose a new authenticated key agreement protocol using smart card for SIP which is immune to the presented attack. Besides, the proposed protocol also has better performance than Zhang et al.'s protocol.

**Key-words:** elliptic curve; authenticated key agreement; session initiation protocol.

## 1. Introduction

With the quick momentum of multimedia services using the voice over internet protocol, internet telephony over traditional circuit-switched based telephony gain more and more growth. In multimedia services using the voice over internet protocol, the session initiation protocol (SIP) is used to manage sessions between the user and the server. The SIP is a text based protocol, which was introduced by the Internet Engineering Task Force Network Working Group[1].

The original authentication protocol for SIP is not secure for applications since it was based on hyper text transport protocol digest authentication [2]. Yang et al. [3] found that the original SIP authentication protocol was vulnerable to the off-line password guessing attack and the server spoofing attack. Then they proposed an improved SIP authentication protocol based on the difficulty of discrete logarithm problem. Later, Huang et al. [4] found that Yang et al.'s protocol was vulnerable to the off-line password guessing attack and was not suitable for devices with a low computational power. They also proposed an efficient authentication protocol for the SIP. Jo et al. [5] found that Huang et al.'s protocol was still vulnerable to the off-line password guessing attack. In order to get better efficiency,

Durlanik et al. [6] presented an an efficient authentication protocol using the elliptic curve cryptography (ECC) for the SIP. The ECC was first proposed by Koblitz [7] and Miller [8] separately. Compared with the traditional public key cryptography, the ECC is more suitable for devices with a low computational power since it could provide the same security level with much smaller key length. Later, Wu et al. [9] also proposed an efficient authentication protocol using the ECC for the SIP. Wu et al. also demonstrated that their protocol was provably secure in the Canetti-Krawczyk security model [10]. Unfortunately, Yoon et al. [11] found that both of Durlanik et al.'s protocol and Wu et al.'s protocol were vulnerable to the off-line password guessing attack, the Denning-Sacco attack and stolen-verifier attack. In order to enhance security, Yoon et al. also presented an improved authentication protocol using the ECC for the SIP. Later, Pu [12] and Gokhroo et al. [13] found that Yoon et al. protocol was vulnerable to the off-line password guessing attack and the replay attack. In order to improve performance, Tsai et al. [14] also proposed an authentication protocol just using the hash function and the exclusive-or operation for the SIP. However, Yoon et al. [15] found that Tsai et al.'s protocol was vulnerable to the off-line password guessing attack, the Denning-Sacco attack and the stolen-verifier attack. Arshad et al. [16] also found that Tsai et al.'s protocol was vulnerable to the the off-line password-guessing attack. Yoon et al. [15] and Arshad et al. [16] also presented two two improved authentication protocols using the ECC for the SIP respectively. Unfortunately, Xie [17] and He et al. [18] found that Yoon et al.' protocol [15] and Arshad et al.'s protocol [16] were vulnerable to the off-line password guessing attack respectively.

In the above authentication protocols for the SIP, a verification table related all users' passwords is maintained by the SIP server. Thus, those protocols were vulnerable to the stolen-verifier attack. Besides, those protocols were vulnerable to the password management problem since the verification table is usually very large. In order to solve those problems, Zhang et al. [19] proposed the first authentication protocol using smart card and ECC for SIP. However, we find that Zhang et al.'s protocol is vulnerable to the user impersonation attack, i. e., a malicious user could impersonate any other user to the server. In order to enhance security, we also propose an improved authentication protocol using smart card and ECC for SIP.

The remainder of this paper is organized as follows: Section 2 briefly reviews Zhang et al.'s authentication protocol using smart card and ECC for SIP. The user impersonation attack on Zhang et al.'s protocol is proposed in Section 3. Our proposed

protocol is presented in Section 4, while Sections 5 and 6 discuss the security and efficiency of the proposed protocol. Our conclusions are presented in Section 7.

## 2. Review of Zhang et al.'s authentication protocol

In this section, we review Zhang et al.'s authentication protocol for the SIP. Zhang et al.'s protocol consists of four phases: the system setup phase, the registration phase, the authentication phase, and the password changing phase.

### 2.1. System setup phase

In this phase, the SIP server generates the system parameters, that are available for public uses. The below steps will be undertaken by the SIP server.

**Step 1.** The server chooses an elliptic curve equation $E_p(a, b)$ with the order $n$ and a base point $P$ on the elliptic curve $E_p(a, b)$.

**Step 2.** The server chooses a random number $s \in Z_n^*$ and computes the public key $P_{pub} = sP$.

**Step 3.** The server chooses three secure hash functions $h(\cdot) : \{0,1\}^* \to \{0,1\}^k$, $h_1 : G \times \{0,1\}^*\{0,1\}^* \to \{0,1\}^k$ and $h_2 : G \times G \times \{0,1\}^*\{0,1\}^* \to \{0,1\}^k$, where $G$ denote the addition group generated by $P$.

**Step 4.** The server keeps $s$ secretly and publishes the system parameters $\{E_p(a, b), P, P_{pub}, h(\cdot), h_1(\cdot), h_2(\cdot)\}$.

### 2.2. Registration phase

In this phase, the user registers on the SIP server through a secure channel. The below steps will be undertaken by the SIP server and the user.

**Step 1.** The user chooses his password $PW$ and a random number $a \in Z_n^*$. Then, the user sends $\{h(PW||a), username\}$ to the server through a secure channel.

**Step 2.** After that, the server computes $R = h(h(PW||a)||username)s^{-1}P$. Then, the server stores $R$ into the memory of the smart card and delivers it to the user through a secure channel.

**Step 3.** After receiving the smart card, the user stores $a$ into the memory of the smart card.

### 2.3. Authentication phase

Whenever the user wants to log into the server, he inserts his smart card into a smart card reader. He also inputs his username and password $PW$. As shown in Fig. 1, the below steps will be undertaken by the SIP server and the user.

**Step 1.** The user generates a random number $b \in Z_n^*$ and calculates $V = bR + h(username)P$ and $W = bh(h(PW||a||username))P_{pub}$. Then, the users sends the request message $REQUEST(username, V, W)$ to the server.

**Step 2.** The server, after receiving the request message, calculates
$X = h(username)P$ and $W' = s^2(V - X)$.

Then, the server verifies whether $W$ and $W'$ are equal. If they are equal, the server generates two random numbers $c, r \in Z_n^*$, calculates $S = cP$, $K = cs(V - X)$, $SK = h_1(K||r||username)$ and $Auth_s = h_2(K||W'||r||SK)$. Then, the server sends the challenge message $CHALLENGE(realm, Auth_s, S, r)$ to the users.

**Step 3.** The user, after receiving the challenge message, calculates

$K = bh(h(PW||a)||username)S$ and $SK = h_1(K||r||username)$.

Then, the users verifies whether $Auth_s$ and $h_2(K||W||r||SK)$ are equal. If they are equal, the users calculates $Auth_u = h_2(K||W||r + 1||SK)$ and sends the response message $RESPONSE(realm, Auth_u)$ to the server.

**Step 4.** The server, after receiving the response message, verifies whether $Auth_u$ and $h_2(K||W'||r + 1||SK)$ are equal. If they are equal, the server sets $SK$ as the shared session key.

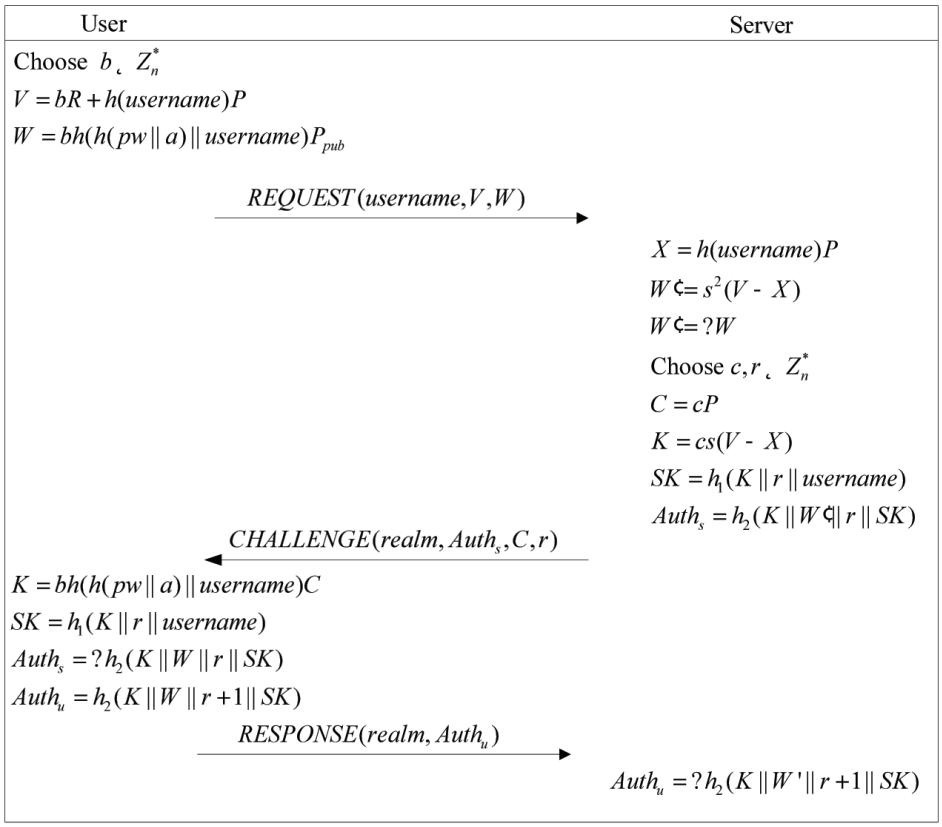


**Fig. 1.** Authentication phase of the proposed protocol.

### 2.4. Password changing phase

After the user and the server authenticate each other, the user can change his password using the current session key $SK$. The below steps will be undertaken by the SIP server and the user.

**Step 1.** The user chooses a new password $PW^{new}$, a random number $a^* \in Z_n^*$ and a nonce $N$. Then, the server sends the request message $E_{SK}(username||N||h(PW^*||a^*)||h(username||N||h(PW^*||a^*))$ and $N$ to the server.

**Step 2.** The server, after receiving the request message, decrypts the message and verifies whether the authentication tag $h(username||N||h(PW^*||a^*))$ is valid. If it is valid, the server computes $R^* = h(h(PW^*||a^*)||username)s^{-1}P$ and sends the response message $E_{SK}(R^*||h(username||N+1||R^*))$ to the user.

**Step 3.** The user, after receiving the response message, decrypts the message and verifies whether the authentication tag $h(username||N+1||R^*)$ is valid. If it is valid, the user stores $R^{ast}, a^*$ into his smart card.

## 3. Attack on Zhang et al.'s authentication protocol

Suppose that $U_A$ is malicious user. Through the registration phase, $U_A$ could get his smart card from the SIP server, where the smart card contains $R_A = h(h(PW||a)||username)s^{-1}P$ and the random number $a$. Then $U_A$ could impersonate another user $U_i$ through the following steps.

1) $U_A$ computes $s^{-1}P = h(h(PW||a)||username)^{-1}R_A$.

2) $U_A$ generates two random numbers $PW_i^*$, $a_i^*$ and computes $R_i^* = h(h(PW_i^*||a_i^*)||username_i)s^{-1}P$, where $username_i$ is the name of $U_i$.

3) $U_A$ generates a random number $b \in Z_n^*$ and calculates $V_i = bR_i^* + h(username)P$ and $W_i = bh(h(PW||a||username))P_{pub}$. Then, the users sends the request message $REQUEST(username_i, V_i, W_i)$ to the server.

4) The server, after receiving the request message, calculates $X = h(username_i)P$ and $W' = s^2(V_i - X)$. Then, the server verifies whether $W_i$ and $W'$ are equal. It is easy to say that $W$ and $W'$ are equal. Then, the server generates two random numbers $c, r \in Z_n^*$, calculates $S = cP$, $K = cs(V_i - X)$, $SK = h_1(K||r||username)$ and $Auth_s = h_2(K||W'||r||SK)$. Then, the server sends the challenge message $CHALLENGE(realm, Auth_s, S, r)$ to the users.

5) $U_A$ calculates $K = bh(h(PW_i^*||a_i^*)||username_i)S$ and $SK = h_1(K||r||username)$. Then, the users verifies whether $Auth_s$ and $h_2(K||W_i||r||SK)$ are equal. If they are equal, $U_A$ calculates $Auth_u = h_2(K||W_i||r+1||SK)$ and sends the response message $RESPONSE(realm, Auth_u)$ to the server.

Step 4. The server, after receiving the response message, verifies whether $Auth_u$ and $h_2(K||W'||r+1||SK)$ are equal. It is easy to say that $Auth_u$ and $h_2(K||W'||r+1||SK)$ are equal. Then, the server sets $SK$ as the shared session key.

Thus, the malicious user $U_A$ can impersonate $U_i$ to the server easily. Therefore, Zhang et al.'s protocol cannot withstand the user impersonation attack.

## 4. Proposed authentication protocol

Based on He et al.'s work [20], we present an improved authentication protocol for the SIP to overcome weakness in Zhang et al.'s protocol. The proposed protocol consists of four phases: the system setup phase, the registration phase, the authentication phase, and the password changing phase.

### 4.1. System setup phase

In this phase, the SIP server generates the system parameters, that are available for public uses. The below steps will be undertaken by the SIP server.

**Step 1.** The server chooses an elliptic curve equation $E_p(a, b)$ with the order $n$ and a base point $P$ on the elliptic curve $E_p(a, b)$.

**Step 2.** The server chooses a random number $s \in Z_n^*$ and computes the public key $P_{pub} = sP$.

**Step 3.** The server chooses three secure hash functions $h(\cdot) : \{0,1\}^* \rightarrow \{0,1\}^k$, $h_1 : G \times \{0,1\}^*\{0,1\}^* \rightarrow \{0,1\}^k$ and $h_2 : G \times G \times \{0,1\}^*\{0,1\}^* \rightarrow \{0,1\}^k$, where $G$ denote the addition group generated by $P$.

**Step 4.** The server keeps $s$ secretly and publishes the system parameters $\{E_p(a, b), P, P_{pub}, h(\cdot), h_1(\cdot), h_2(\cdot)\}$.

### 4.2. Registration phase

In this phase, the user registers on the SIP server through a secure channel. The below steps will be undertaken by the SIP server and the user.

**Step 1.** The user chooses his password $PW$ and a random number $a \in Z_n^*$. Then, the user sends $\{h(PW||a), username\}$ to the server through a secure channel.

**Step 2.** After that, the server computes $R = ((s+h(username))^{-1}+h(PW||a))P$. Then, the server stores $R$ into the memory of the smart card and delivers it to the user through a secure channel.

**Step 3.** After receiving the smart card, the user stores $a$ into the memory of the smart card.

### 4.3. Authentication phase

Whenever the user wants to log into the server, he inserts his smart card into a smart card reader. He also inputs his username and password $PW$. As shown in Fig. 2, the below steps will be undertaken by the SIP server and the user.

**Step 1.** The user generates a random number $b \in Z_n^*$ and calculates $V = b(R - h(PW||a)P) = b(s + h(username)^{-1}P)$ and $W = h_1(username||V||bP)$. Then, the users sends the request message $REQUEST(username, V, W)$ to the server.

**Step 2.** The server, after receiving the request message, calculates $T = (s + h(username))V = bP$ and $W' = h_1(username||V||T)$. Then, the server verifies whether $W$ and $W'$ are equal. If they are equal, the server generates two random numbers $c \in Z_n^*$, calculates $S = cP$, $K = cT$, $SK = h_1(K||username)$

and $Auth_s = h_2(\text{"1"}||K||W'||SK)$. Then, the server sends the challenge message $CHALLENGE(realm, Auth_s, S, r)$ to the users.

**Step 3.** The user, after receiving the challenge message, calculates $K = bh(h(PW||a)||username)S$ and $SK = h_1(K||r||username)$. Then, the users verifies whether $Auth_s$ and $h_2(\text{"1"}||K||W'||SK)$ are equal. If they are equal, the users calculates $Auth_u = h_2(\text{"2"}||K||W||SK)$ and sends the response message $RESPONSE(realm, Auth_u)$ to the server.

**Step 4.** The server, after receiving the response message, verifies whether $Auth_u$ and $h_2(\text{"2"}||K||W'||SK)$ are equal. If they are equal, the server sets $SK$ as the shared session key.

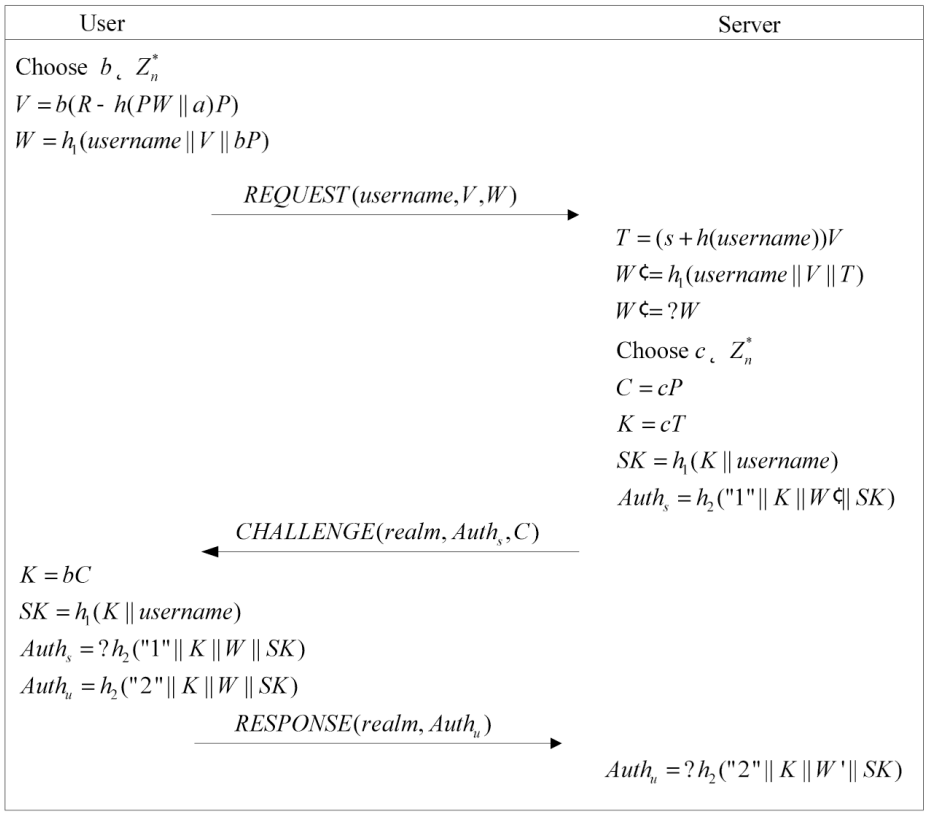


**Fig. 2.** Authentication phase of the proposed protocol.

### 4.4. Password changing phase

After the user and the server authenticate each other, the user can change his password using the current session key $SK$. The below steps will be undertaken by the SIP server and the user.

**Step 1.** The user chooses a new password $PW^{new}$, a random number $a^* \in Z_n^*$ and a nonce $N$. Then, the server sends the request message $E_{SK}(username||N||h(PW^*||a^*)||h(username||N||h(PW^*||a^*))$ and $N$ to the server.

**Step 2.** The server, after receiving the request message, decrypts the message and verifies whether the authentication tag $h(username||N||h(PW^*||a^*))$ is valid. If it is valid, the server computes $R^* = h(h(PW^*||a^*)||username)s^{-1}P$ and sends the response message $E_{SK}(R^*||h(username||N+1||R^*))$ to the user.

**Step 3.** The user, after receiving the response message, decrypts the message and verifies whether the authentication tag $h(username||N+1||R^*)$ is valid. If it is valid, the user stores $R^{ast}, a^*$ into his smart card.

## 5. Security analysis

In this section, we demonstrate that the proposed protocol could withstand the user impersonation attack, the server spoofing attack, the privileged insider attack, the stolen-verifier attack, the replay attack, the modification attack, and the man-in-the- middle attack [19–21]. Beside, we demonstrate that the proposed protocol could provide the mutual authentication and the perfect forward secrecy [19–21].

### 5.1. User impersonation attack

Suppose there is an adversary $A$ wants to impersonate a valid user $U_i$ to the server. $A$ should generate a request message $REQUEST(username, V, W)$ to pass the server's verification, where $V = b(R - h(PW||a)P) = b(s + h(username)^{-1}P)$ and $W = h_1(username||V||bP)$. However, he cannot generate $V$ and $W$ since he does know the secret information $R$ and the password $PW$. As a result, the proposed protocol could resist the user impersonation attack.

### 5.2. Server spoofing attack

Suppose there is an adversary $A$ wants to impersonate the server to a valid user $U_i$. $A$ should generate a challenge message $CHALLENGE(realm, Auth_s, S, r)$ when he receives a request message $REQUEST(username, V, W)$, where $V = b(R - h(PW||a)P) = b(s + h(username)^{-1}P)$, $W = h_1(username||V||bP)$, $S = cP$, $K = cT$, $SK = h_1(K||username)$ and $Auth_s = h_2(\text{"1"}||K||W'||SK)$. However, he cannot generate $Auth_s$ since he does not know the secret key $s$. As a result the proposed protocol could resist the spoofing attack.

### 5.3. Privileged insider attack

Suppose there is a privileged insider $A$ wants to get the user $U_i$'s password $PW$ in the registration phase. However, he cannot get $U_i$'s password $PW$ from the registration request message $\{h(PW||a), username\}$ since it is protected by the hash function $h(\cdot)$ and the random number $a$. As a result, the proposed protocol could resist the privileged insider attack.

### 5.4. Stolen-verifier attack

In the proposed protocol, the server just needs to maintain his secret key $s$ and maintain no verifier table at all. Then, there is no information about user's passwords stored in the server's database. As a result, the proposed protocol could resist the stolen-verifier attack.

### 5.5. Replay attack

Suppose there is an adversary $A$ eavesdrops a request message $REQUEST$ $(username, V, W)$ and replays it to the server, where $V = b(R - h(PW||a)P) = b(s + h(username)^{-1}P)$ and $W = h_1(username||V||bP)$. However, he cannot generate the response message $RESPONSE(realm, Auth_u)$ when he receives the server's challenge message $CHALLENGE(realm, Auth_s, S, r)$ since he does not know the random number $b$. Suppose there is an adversary $A$ eavesdrops the server's challenge message $CHALLENGE(realm, Auth_s, S, r)$ and replays it to the user $U_i$. However, $U_i$ could find the attack by verifying if $Auth_s$ and $h_2(\text{"1"}||K||W'||SK)$ are equal since $U_i$ generates a new random number $b$ in each session. As a result, the proposed protocol could resist the replay attack.

### 5.6. Modification attack

Suppose there is an adversary $A$ eavesdrops a request message $REQUEST$ $(username, V, W)$, modifies it and replays it to the server, where $V = b(R-h(PW||a)$ $P) = b(s+h(username)^{-1}P)$ and $W = h_1(username||V||bP)$. However, the message is protected by the hash value $W$. Any modification of the message could by find by the server in the Step 2) of the authentication phase. We also could show that any modification on the challenge message $CHALLENGE(realm, Auth_s, S, r)$ and the response message $RESPONSE(realm, Auth_u)$ could be found. As a result, the proposed protocol could resist the modification attack.

### 5.7. Man-in-the- middle attack

Suppose there is an adversary $s$ could intercept the messages transmitted between the user and server and replace them with his own messages. We have demonstrated that there is no adversary could generate a request message $REQUEST(username,$ $V, W)$, a challenge message $CHALLENGE(realm, Auth_s, S, r)$ or a response message $RESPONSE(realm, Auth_u)$. Then, the user or the sever could find the attack by checking the validity of the received message. As a result, the proposed protocol could resist the man-in-the- middle attack.

### 5.8. Mutual authentication

Mutual authentication means that the server and user could verify the legality of each other. According to Section 5.1 and Section 5.2, we could conclude that no adversary could generate valid message to impersonate the user or the server. Then,

the user and the server could authenticate each other by verifying the validity of received message. As a result, the proposed scheme could provide mutual authentication between the user and the server.

### 5.9. Perfect forward secrecy

Perfect forward secrecy means the adversary cannot get the previous session key even he gets the the long-term keys of the server and the user. Suppose there is an adversary gets the server's key $s$ and the user's key $(s + h(username)^{-1}P$. We also assume the adversary gets the request message $REQUEST(username, V, W)$, the challenge message $CHALLENGE(realm, Auth_s, S, r)$, and the response message $RESPONSE(realm, Auth_u)$, where $V = b(R - h(PW||a)P)$, $W = h_1(username||V||bP)$, $S = cP$, $K = cT$, $SK = h_1(K||username)$, $Auth_s = h_2(\text{"1"}||K||W'||SK)$, and $Auth_u = h_2(\text{"2"}||K||W||SK)$. The adversary could computes $T = (s+h(username))V = bP$. In order to get the session key $SK = h_1(K||username)$, he has to compute $K = cT = bcP$ from $bP$ and $cP$. Then, the adversary has to solve the computational Diffie-Hellman problem. As a result, the proposed scheme could provide the perfect forward secrecy.

## 6. Comparison and cost analysis

In this section, we compare the efficiency of the scheme with that of Zhang et al.'s scheme. For convenience, we let $M_{ec}$ and $H$ denote the elliptic curve scale multiplication operation and the hash function operation respectively. The comparisons are listed in Table 1.

**Table 1.** Comparison of computation cost

| | Zhang et al.'s scheme | The proposed scheme |
|---|---|---|
| User side | $4M_{ec} + 5H$ | $3M_{ec} + 5H$ |
| Server side | $4M_{ec} + 4H$ | $3M_{ec} + 4H$ |
| Total | $8M_{ec} + 9H$ | $6M_{ec} + 9H$ |

According to Table 1, we can see that the computation cost of the proposed scheme is lower than that of Zhang et al.'s scheme in the user side and the server side. Furthermore, Zhang et al.'s scheme is vulnerable to the user impersonation attack and the proposed scheme could overcome such security vulnerability. As a result, the proposed scheme is superior to Zhang et al.'s scheme in terms of both of security and efficiency.

## 7. Conclusions

In this paper we presented a cryptanalysis of Zhang et al.'s authenticated key agreement for session initiation protocol and demonstrate that their scheme is vulnerable to the user impersonation attack. To address the problem, we proposed an improved authenticated key agreement for session initiation protocol using smart

card. The proposed scheme not only resists the proposed attack, but also has better performance. As a result, the proposed scheme is more

**Acknowledgments.** This research was supported in part by the National Science foundation of China (No. 61201180), the Beijing Natural Science Foundation (No. 4132055), and the Excellent Young Scholars Research Fund of Beijing Institute of Technology.

# References

[1] Rosenberg J., Schulzrinne H., Camarillo G., Johnston A., Peterson J., Sparks R., Handley M., Schooler E., *SIP: session initiation protocol*, RFC 3261, 2002.

[2] Franks J., Hallam-Baker P., Hostetler J., Lawrence S., Leach P., Luotonen A., Stewart L., *HTTP authentication: basic and digest access authentication*, Internet RFC2617, 1999.

[3] Yang C., Wang R., Liu W., *Secure authentication scheme for session initiation protocol*, Computers and security, 2005, Vol. **24**, pp. 381–386.

[4] Huang H., Wei W., Brown G., *A new efficient authentication scheme for session initiation protocol*, Proceedings of JCIS 06, 2006, pp. 180–182.

[5] Jo H., Lee Y., Kim M., Kim S., Won D., *Off-line password-guessing attack to Yang's and Huang's authentication schemes for session initiation protocol*, Proceedings of INC 09, 2009, pp. 618–621.

[6] Durlanik A., Sogukpinar I., *SIP authentication scheme using ECDH*, World Enformatika Society Transaction on Engineering Computing and Technology, 2005, Vol. **8**, pp. 350–353.

[7] Koblitz N., *Elliptic curve cryptosystems*, Mathematics of Computation, 1987, Vol. **48**, pp. 203–209.

[8] Miller V., *Uses of elliptic curves in cryptography*, *Advances in Cryptology CRYPTO'85*, 1985, pp. 417–426.

[9] Wu L., Zhang Y., Wang F., *A new provably secure authentication and key agreement protocol for SIP using ECC*, Computer Standards and Interfaces, 2009, Vol. **31**, pp. 286–291.

[10] Canetti R., Krawczyk H., *Analysis of key-exchange protocols and their use for building secure channels*, *Proceedings of EUROCRYPT 2001*, 2001, pp. 453–474.

[11] Yoon E., Yoo K., *A secure and efficient SIP authentication scheme for converged VoIP networks*, Computer Communications, 2010, Vol. **33**, pp. 1674–1681.

[12] Pu Q., *Weaknesses of SIP authentication scheme for converged VoIP networks*, IACR Cryptology ePrint Archive, 2010, 464.

[13] Gokhroo M., Jaidhar C., Tomar A., *Cryptanalysis of SIP secure and efficient authentication scheme*, Proceedings of ICCSN 2011, 2011, pp. 308–310.

[14] Tsai J., *Efficient nonce-based authentication scheme for session initiation protocol*, International Journal of Network Security, 2009, Vol. **9**, pp. 12–16.

[15] Yoon E., Shin Y., Jeon I., Yoo K., *Robust mutual authentication with a key agreement scheme for the session initiation protocol*, IETE Technical Review, 2010, Vol. **27**, pp. 203–213.

[16] Arshad R., Ikram N., *Elliptic curve cryptography based mutual authentication scheme for session initation protocol*, Multimedia Tools and Applications, 2013, Vol. **66**(2), pp. 165–178.

[17] Xie Q., *A new authenticated key agreement for session initiation protocol*, International Journal of Communication Systems, 2012, Vol. **25**, pp. 47–54.

[18] He D., Chen J., Chen Y., *A secure mutual authentication scheme for session initiation protocol using elliptic curve cryptography*, Security and Communication Networks, 2012, Vol. **5**(12), pp. 1423–1429.

[19] Zhang L., Tang S., Cai Z., *Efficient and flexible password authenticated key agreement for voice over internet protocol session initiation protocol using smart card*, Internal Journal of Communication Systems, 2013, DOI: 10.1002/dac.2499

[20] He D., Chen J., Hu J., *An ID-based client authentication with key agreement protocol for mobile client-server environment on ECC with provable security*, Information Fusion, 2012 Vol. **13**(3), pp. 223–230.

[21] Shi W., Chen Y., *An efficient RSA-based remote user authentication scheme*, Romanian Journal of Information Science and Technology, Vol. **15**(3), pp. 266–276.