Cryptanalysis and Improvement of Password Authenticated Key Agreement for Session Initiation Protocol Using Smart Cards

L. Zhang, Shanyu Tang*, *Senior Member IEEE*, Z. Cai

Secure Communication Institute, China University of Geosciences, Wuhan, 430074, China

**Corresponding author: shanyu.tang@gmail.com, carolyn321@163.com,Tel/Fax:+862767848563*

Short Title: Cryptanalysis and Improvement of Authenticated Key Agreement

ABSTRACT

Session Initiation Protocol (SIP) is one of the most commonly used protocols for handling sessions for Voice over Internet Protocol (VoIP)-based communications, and the security of SIP is becoming increasingly important. Recently, Zhang *et al.* proposed a password authenticated key agreement protocol for SIP by using smart cards to protect the VoIP communications between users. Their protocol provided some unique features, such as mutual authentication, no password table needed, and password updating freely. In this study, we performed cryptanalysis of Zhang *et al.*'s protocol and found that their protocol was vulnerable to the impersonation attack although the protocol could withstand several other attacks. A malicious attacker could compute other users' privacy keys and then impersonated the users to cheat the SIP server. Furthermore, we proposed an improved password authentication key agreement protocol for SIP, which overcame the weakness of Zhang *et al.*'s protocol and was more suitable for VoIP communications.

KEY WORDS: authentication; key agreement; session initiation protocol; elliptic curve

1. INTRODUCTION

Session Initiation Protocol (SIP) as an IP-based telephony protocol was proposed by Rosenberg *et al*. at the Internet Engineering Task Force (IETF), Multi-Party Multimedia Session Control (MMUSIC) Working Group [1]. It is used for creating, modifying and terminating multimedia sessions between one or more participants as an application layer signaling protocol [1]. SIP is a text-based protocol, and has ability to operate on TCP or UDP and to control all the signaling requirements during a VoIP session. Similar to HTTP, SIP is a request-response protocol, and so constructing secure mutual authentication and key agreement mechanism is a critical issue for SIP-based IP telephony services.

The original authentication protocol for SIP was based on HTTP Digest authentication, so it was not strong enough for providing acceptable security level in practice [2-3]. In order to strengthen the security of SIP, several authenticated key agreement protocols have been proposed [4-17] for the prevalent VoIP service. In 2004, Badra and Urien [4] firstly introduced smartcards to remote authenticate passwords using public key encryption. In 2005, Yang *et al.* [5] reported that the original SIP authentication protocol was vulnerable to the off-line password guessing attack and the server-spoofing attack, and they proposed a SIP authentication protocol to overcome these weaknesses. However, Yang *et al*.'s protocol was vulnerable to the off-line password guessing attack and involved expansive exponential computation, so it was not suitable for the devices with a low computational power. Ring et al. [6] proposed an authentication key agreement (AKA) for SIP by using identity-based cryptography (IDC). Wang *et al*. [7] presented an AKA based on certificateless cryptography. Guillet *et al*. [8] suggested mutual authentication for SIP by using a semantic meaning for the SIP opaque values. Although these protocols avoid the requirement of a large Public Key Infrastructure (PKI), the computational costs of these protocols are still very high due to the usage of expansive bilinear pairings.

In order to reduce the computational costs and improve efficiency, several protocols based on elliptic curve cryptosystem (ECC) have been proposed [10-11, 14-17]. In 2009, Wu *et al*. [10] proposed an SIP authentication protocol based on ECC. Since ECC offered a level of security comparable to the classical cryptosystems that use much larger size keys, Wu *et al.*'s protocol would reduce computational costs efficiently. In Wu *et al*.'s protocol, the communicating parties shared a common secret beforehand

between the IM Services Identity Module (ISIM) and the Authentication Center (AC). Although, this pre-shared key protocol was more efficient, the problem of distributing the shared secrets made this solution hard to scale up. In 2010, Yoon *et al*. [11] claimed that Wu *et al.*'s protocol was suffered from Denning-Sacco attacks, off-line password guessing attacks, and Stolen-verifier attacks. To strengthen the security, they also suggested an AKA protocol for SIP based on ECC. However, Gokhroo *et al*. [12] argued that Yoon *et al*.'s protocol still suffered from both off-line password guessing attacks and replay attacks. Recently, a new AKA protocol based on nonce for SIP was proposed by Tsai *et al.* [13]. In their protocol, only exclusive-or operations and one-way hash function was used to realize mutual authentication and key agreement, so it reduced the computational costs efficiently. Unfortunately, Tsai's protocol was vulnerable to off-line password guessing attacks, stolen verifier attacks and could not provide perfect forward secrecy and known-key secrecy. To overcome above weaknesses, Arshad *et al*. [14] proposed an enhanced AKA protocol based on ECC. But, He *et al*. [15] demonstrated that Arshad *et al*.'s protocol still suffered from the off-line password-guessing attack.

Most recently, Zhang *et al.* [17] argued that the existing protocols designed for SIP required the SIP server maintaining a password or verification table, which makes these protocols vulnerable to stolen-verifier attacks, server-spoofing attacks, insider attacks and password guessing attacks. On the other hand, as the password and verification tables are usually very large, the maintenance and updating problems would reduce its applicability for practical use. Motivated by the above analysis, Zhang *et al*. proposed a new AKA protocol for SIP by using smart cards to avoid storing password tables at the SIP server. Unfortunately, their effort failed to address the impersonation attack problem.

In this study, we carried out cryptanalysis of Zhang *et al.*'s protocol and show that Zhang *et al.*'s protocol was vulnerable to the impersonation attacks. To solve this problem, we proposed a much improved protocol based on Zhang *et al.*'s protocol by using smart cards. The proposed protocol not only preserved the merits of Zhang *et al.*'s protocol but also solved the security problem of impersonation attacks.

The rest of this paper is organized as follows. Section 2 describes the SIP authentication procedure in general. In Section 3, a brief review of Zhang *et al.*'s protocol is given. Section 4 presents a

cryptanalysis of Zhang *et al*.'s protocol. Our authenticated key agreement protocol is detailed in Section 5. In Section 6, the security of our proposed protocol is discussed. The performance of the proposed protocol is examined in Section 7, and the paper is concluded in Section 8.

2. SIP AUTHENTICATION PROCEDURE

The security of the original SIP authentication is based on the challenge-response mechanism. In order to verify the identity of the user or the server in the authentication procedure, the original SIP authentication protocol requires the user and the server pre-sharing a password beforehand. The details of the original SIP authentication procedure are illustrated in Fig. 1.

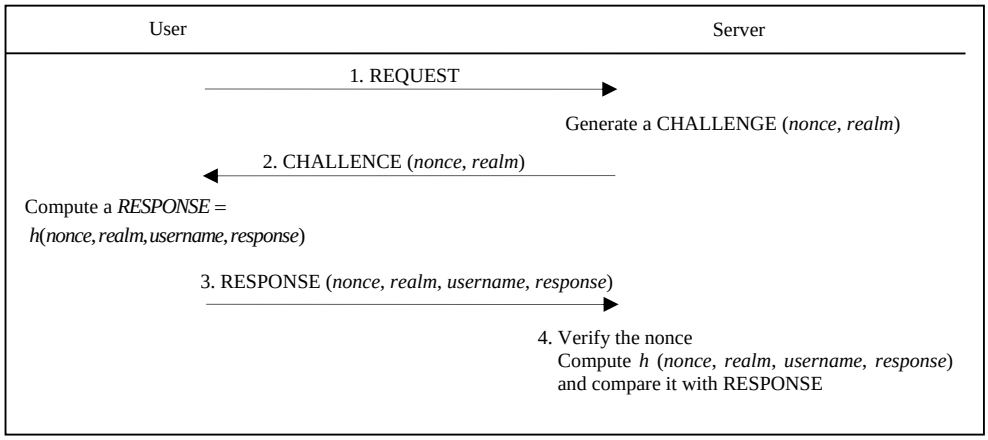

Fig. 1 SIP authentication procedure

$Step1: User \rightarrow Server : REQUEST$

The user submits a REQUEST to the server.

$Step2: Server \rightarrow User : CHALLENGE(nonce, realm)$

The server generates a CHALLENGE as a response message consisting of a nonce and the user's realm. The realm used in the CHALLENGE is the digest algorithm. The server then sends the response message CHALLENGE to the user.

$Step3: User \rightarrow Server : response(nonce, realm, username, response)$

The user computes a *RESPONSE=h*(*nonce*, *realm*, *username*, *response*) by using an nonce value, realm, username and the computed response value, where *h*(.) is a one-way hash function

used for generating a digest authentication message. Then the user sends back the RESPONSE to the server.

*Step*4: After receiving the RESPONSE message, the server extracts the user's password in relation to the username. The server then verifies whether the nonce is correct. If it is correct, the server computes a hash value *h* (*nonce*, *realm*, *username*, *response*), and then checks whether this hash value is equal to the received value of RESPONSE. If they match, the server authenticates the identity of the user.

The limitations of this authentication mechanism in a real application are summarized as follows: 1) lack of media protection mechanism, and the sensitive message can be intercepted; 2) vulnerable to offline dictionary guessing attack, since the security of the authentication mechanism depends on a simple password only; 3) the password table needing configure beforehand at the SIP server side, which makes this authentication mechanism suffer from stolen-verifier attacks, server-spoofing attacks and insider attacks. In addition, the maintenance of the table and the password updating can be intractable problems; 4) not providing mutual authentication between the users and the SIP server, so that the man-in-middle attacks and the server spoofing attacks may occur; 5) the impersonate attacks may be successful by generating a forgery SIP message, since no secure mechanism is provided to protect the important parameters in the header field of the SIP message. According to above analysis, the original SIP authentication cannot provide services of acceptable security, it needs to be improved for SIP-based IP telephony services.

3. REVIEW OF ZHANG ET AL.'S PROTOCOL

In this section, we briefly review Zhang *et al.*'s protocol as follows:

3.1. System setup phase

In the system setup phase, the SIP server generates several security parameters.

Step $S1$: The SIP server chooses an elliptic curve equation $E_p(a,b): y^2 = x^3 + ax + b(\text{mod } p)$ over a prime finite field F_p, where $a,b \in F_p$ and $4a^3 + 27b^2 \neq 0(\text{mod } p)$. Then it selects a base

point P over $E_p(a,b)$, and chooses a random integer $s \in_R Z_p^*$ as a secret key, and then computes the public key $P_{pub} = sP$.

Step $S2$: The SIP server chooses three secure one-way hash functions $h(\cdot): \{0,1\}^* \to \{0,1\}^k$, $h_1(\cdot): G \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^k$, and $h_2(\cdot): G \times G \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^k$, where G is a cyclic addition group that is generated by P over $E_p(a,b)$.

Step $S3$: The SIP server keeps s secret, and publishes the public information $\{E_p(a,b), P, P_{pub}, h(\cdot), h_1(\cdot), h_2(\cdot)\}$.

3.2. Registration phase

When the user U wants to register with the SIP server, it performs the following steps with the server.

Step $R1$: The user U chooses its password PW and a random integer $a \in_R Z_p^*$. Next, it computes $h(PW \| a)$, and sends $\{h(PW \| a), username\}$ to the SIP server over a secure channel.

Step $R2$: The SIP server computes secret information $R = h(h(PW \| a) \| username) s^{-1} P$.

Step $R3$: The SIP server stores R in a smart card, and delivers this smart card to the user U in a secure channel.

Step $R4$: The user U stores a in the smart card, and so the smart card contains (R, a).

3.3. Authentication phase

When the user U wishes to logon to the SIP server, the smart card and the server perform the following steps as shown in Figure 2.

Step $A1$: $U \to S: REQUEST(username, V, W)$

The user U selects a random integer $b \in_R Z_p^*$, and computes $V = bR + h(username)P$ and $W = bh(h(PW \| a) \| username) P_{pub}$. Then it sends $REQUEST(username, V, W)$ to the SIP server.

Step $A2: S \rightarrow U : CHALLENGE(realm, Auth_s, S, r)$

After receiving the request message, the SIP server computes $X = h(username)P$ and $W' = s^2(V - X)$. It then verifies whether the following equation holds $W' \stackrel{?}{=} W$. If the equation holds, it chooses two random integers $r, c \in_R Z_p^*$, and computes $S = cP$, $K = cs(V - X)$, $SK = h_1(K \| r \| username)$ $Auth_s = h_2(K \| W' \| r \| SK)$. Next, it sends $CHALLENGE(realm, Auth_s, S, r)$ to the user U.

Step $A3: U \rightarrow S : RESPONSE(realm, Auth_u)$

Upon receiving the challenge message, the user U computes $K = bh(h(PW \| a) \| username)S$ and $SK = h_1(K \| r \| username)$. Next it verifies whether the following equation holds $Auth_s \stackrel{?}{=} h_2(K \| h(h(PW \| a) \| username)bP_{pub} \| r \| SK)$. If the equation holds, it computes $Auth_u = h_2(K \| h(h(PW \| a) \| username)bP_{pub} \| r+1 \| SK)$, and sends $RESPONSE(realm, Auth_u)$ to the SIP server.

Step $A4$: After receiving the response message, the SIP server verifies if $Auth_u \stackrel{?}{=} h_2(K \| W' \| r+1 \| SK)$ holds. If so, the SIP server sets SK as the shared session key with the user U

User U $(Username, PW, Smartcard(R, a))$	Server (s)
1. $b \in_R Z_p^*$, $V = bR + h(username)P$, $W = bh(h(PW \| a) \| username)P_{pub}$	
$REQUEST(username, V, W)$ →	
	2. $X = h(username)P$ $W' = s^2(V - X)$, $W \stackrel{?}{=} W'$ If the equation holds, $c \in_R Z_p^*$, $r \in_R Z_p^*$ $S = cP$, $K = cs(V - X)$ $SK = h_1(K \| r \| username)$ $Auth_s = h_2(K \| W' \| r \| SK)$
	← $CHALLENGE(realm, Auth_s, S, r)$
3. $K = bh(h(PW \| a) \| username)S$, $SK = h_1(K \| r \| username)$ $Auth_s \stackrel{?}{=} h_2(K \| h(h(PW \| a) \| username)bP_{pub} \| r \| SK)$ If the equation holds, $Auth_u = h_2(K \| h(h(PW \| a) \| username)bP_{pub} \| r+1 \| SK)$	
$RESPONSE(realm, Auth_u)$ →	
	4. Check $Auth_u \stackrel{?}{=} h_2(K \| W' \| r+1 \| SK)$

Fig.2. Authenticated key agreement phase of Zhang *et al.*'s protocol

4. CRYPTANALYSIS OF THE PROTOCOL OF ZHANG ET AL.

In this section, we describe our findings that the protocol of Zhang *et al*. [15] is vulnerable to the impersonation attack [16].

Assuming that an adversary Bob is a legal user; he can impersonate other legal user to cheat the SIP server through forging other user's secret information R . If Bob attempts to impersonate the user U_i to logon to the SIP server, a possible impersonation attack can be performed as described below:

*Step*1: The adversary Bob computes $s^{-1}P = (h(h(PW \| a) \| username))^{-1} R$ by using his secret information R , password PW , secret nonce a and username. Then Bob can construct a valid $R_{U_i} = h(h(PW^* \| a^*) \| username^{'}) s^{-1} P$ of the legal user U_i , where $username^{'}$ is U_i 's username, and PW^*, a^* chosen by Bob. Then Bob can impersonate the legal user U_i to cheat the SIP server. First, Bob chooses a random integer $b^* \in_R Z_p^*$, and computes $V^* = b^* R_{U_i} + h(username^{'}) P$ and $W^* = b^* h(h(PW^* \| a^*) \| username^{'}) P_{pub}$. Next Bob sends $REQUEST(username^{'}, V^*, W^*)$ to the SIP server.

*Step*2: Since $V^* = b^* R_{U_i} + h(username^{'}) P$ is valid, the SIP server computes $W^{'} = s^2 (V^* - h(username) P) = b^* h(h(PW^* \| a^*) \| username^{'}) P_{pub}$. Since the computed result $W^{'}$ equals to the received $W^* = b^* h(h(PW^* \| a^*) \| username^{'}) P_{pub}$, the SIP server proceeds to compute $S = cP$, $K = cs(V^* - h(username^{'}) P)$, $SK = h_1(K \| r \| username^{'})$, $Auth_s = h_2(K \| W^{'} \| r \| SK)$, and sends $CHALLENGE(realm, Auth_s, S, r)$ to the adversary Bob.

*Step*3: The adversary Bob can then compute the session key $SK = h_1(K \| r \| username^{'})$ $Auth_u = h_2(K \| h(h(PW^* \| a^*) \| username^{'}) bP_{pub} \| r+1 \| SK)$, and sends the *RESPONSE* message to the SIP server.

*Step*4: Since the computed result $h_2(K \| W^{'} \| r+1 \| SK)$ equals to the received $Auth_u$, the SIP server accepts the adversary Bob's login request, and believes that the adversary Bob is the user U_i.

According to above analysis, the adversary can easily impersonate any legal user to logon to the SIP server at any time. Therefore, Zhang *et al.*'s protocol is vulnerable to the impersonation attack.

5. OUR PROPOSED PROTOCOL

In this section, we detail our much improved protocol based on Zhang *et al.*'s protocol. The newly designed protocol could overcome the original protocol's security weakness while as kept the merits of the original SIP protocol, such as no password tables stored on the SIP server. There were three phases in our protocol: system setup phase, registration phase, and authentication phase. The procedures of the protocol are described in detail as follows:

5.1. System setup phase

Step *S*1: The SIP server generates several security parameters: an elliptic curve equation $E_p(a,b): y^2 = x^3 + ax + b(\text{mod } p)$ over a prime finite field F_p, where $a, b \in F_p$ and $4a^3 + 27b^2 \neq 0(\text{mod } p)$; a base point P with the order n over $E_p(a,b)$, where n is a large number in terms of security considerations; a cyclic addition group G, which is generated by P over $E_p(a,b)$.

Step *S*2: The SIP server chooses a random integer $s \in_R Z_p^*$ as a secret key, and chooses two secure one-way hash functions $h(\cdot): \{0,1\}^* \rightarrow \{0,1\}^k$ and $h_1(\cdot): G \times G \times G \times \{0,1\}^* \rightarrow \{0,1\}^k$.

Step $S3$: The SIP server keeps s secret, and publishes the public information $\{E_p(a,b), P, h(\cdot), h_1(\cdot)\}$.

5.2. Registration phase

When the user U wants to register with the SIP server, it performs the following steps with the SIP server.

Step $R1: U \rightarrow S: (h(PW \| a), username)$

The user U selects its password PW freely and chooses a random integer $a \in_R Z_p^*$. Then it computes $h(PW \| a)$ and sends $\{h(PW \| a), username\}$ to the SIP server over a secure channel.

Step $R2$: After receiving the information from the user U, the SIP server computes secret information $R = \dfrac{h(PW \| a)}{h(username) + s} P$ for the user U.

Step $R3$: The SIP server stores R in the memory of a smart card, and delivers the smart card to the user U in a secure channel.

Step $R4$: Upon receiving the smart card, the user U stores a in the smart card, as so the memory of the smart card contains (R, a).

5.3. Authentication phase

When the user U wishes to logon to the SIP server, the smart card and the SIP server cooperate to perform the following steps as shown in Figure 3.

User U $(Username, PW, Smartcard(R,a))$		Server (s)
1. $b \in_R Z_p^*$, $V = bR$, $W = bh(PW \| a)P$		
	$REQUEST(username, V, W)$ $\longrightarrow$	
		2. $W' = (h(username) + s)V = h(PW \| a)bP$ $W \stackrel{?}{=} W'$, If the equation holds, $c, r \in_R Z_p^*$ $S = cP$, $SK = ch(username)W$ $Auth_s = h_1(S \| W' \| SK \| r)$
	$CHALLENGE(realm, Auth_s, S, r)$	

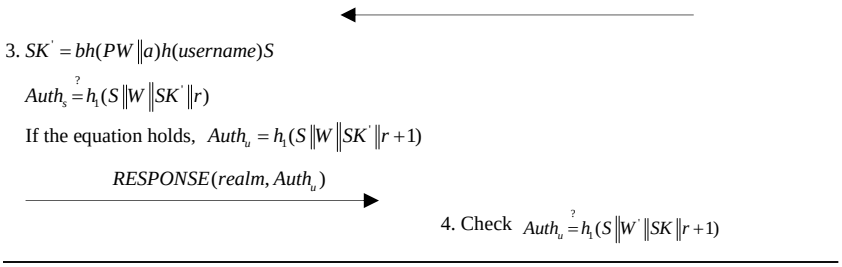

Fig.3. Authenticated key agreement phase of our protocol

Step $A1: U \rightarrow S: REQUEST(username, V, W)$

The user U selects a random integer $b \in_R Z_p^*$, and computes $V = bR$ and $W = bh(PW \| a)P$. It sends $REQUEST(username, V, W)$ to the SIP server.

Step $A2: S \rightarrow U: CHALLENGE(realm, Auth_s, S, r)$

After receiving the request message, the SIP server computes $W' = (h(username) + s)V$ $= (h(username) + s)bR = bh(PW \| a)P$. It then checks whether the following equation $W' \stackrel{?}{=} W$ holds. If so, it generates two random integers $c, r \in_R Z_p^*$, and computes $S = cP$, $SK = ch(username)W' = cbh(PW \| a)h(username)P$ and $Auth_s = h_1(S \| W' \| SK \| r)$. Then it sends $CHALLENGE(realm, Auth_s, S, r)$ to the user U.

Step $A3: U \rightarrow S: RESPONSE(realm, Auth_u)$

Upon receiving the challenge message, the user U inputs its password PW and username to compute the session key $SK' = bh(PW \| a)h(username)S = cbh(PW \| a)h(username)P$. Then it checks whether the equation $Auth_s \stackrel{?}{=} h_1(S \| W \| SK' \| r)$ holds. If the equation holds, it computes $Auth_u = h_1(S \| W \| SK' \| r+1)$, and sends $RESPONSE(realm, Auth_u)$ to the SIP server. Otherwise, it deletes the received information and the protocol stops.

Step $A4$: After receiving the response message, the SIP server verifies whether the equation $Auth_u \stackrel{?}{=} h_1(S \| W' \| SK \| r+1)$ holds. If the message is authenticated, the SIP server sets SK as the shared session key with the user U; otherwise, it deletes the received information and the protocol stops.

6. SECURITY ANALYSIS

The newly designed protocol is a modified form of Zhang *et al.*'s protocol. The security analysis of Zhang *et al.*'s protocol was discussed and demonstrated in the original protocol. Therefore, in this section, only extra security features (e.g. withstanding impersonation attacks) are discussed.

Theorem 1. *Our protocol can resist the impersonation attack.*

Proof. Assuming that an adversary Bob forges a request message $REQUEST(\ username, V^*, W^*)$ by constructing V^* and W^*, and sends it to the SIP server to impersonate a legal user. The SIP server will find the attack by checking whether W' and W^* are equal, because Bob cannot construct a valid $R = (h(PW \| a)/(h(username) + s))P$ without the knowledge of the secret key s. In addition, even if Bob is a legal user, he cannot compute other legal user's privacy key by using its privacy key, since Bob cannot construct a valid R without the knowledge of the secret key s.

On the other hand, an adversary Bob may generate a random number $c^*, r^* \in_R Z_p^*$, compute $S^* = c^*P$ and $Auth_s^*$, and then send a forge challenge message $CHALLENGE(realm, Auth_s^*, c^*P, r^*)$ to the user U to impersonate the SIP server. However, the *CHALLENGE* message cannot go through the verification process of the user U, as Bob does not know the password PW, nonce a and random number b.

Furthermore, an adversary Bob may guess authentication information $Auth_u^*$; send a forged response message $RESPONSE(realm, Auth_u^*)$ to the SIP server to impersonate the user U. However, the SIP server will find the attack by checking whether $Auth_u^* \stackrel{?}{=} h_1(S \| W' \| SK \| r + 1)$ holds. Therefore, the proposed protocol can resist the impersonation attack.

7. COMPLEXITY ANALYSIS

In this section, we compare the performance of our proposed protocol with He *et al.*'s protocol and Zhang *et al.*'s protocol. First, we define some notations as follows:

(1) T_{ecsm} the time for executing a scalar multiplication operation of elliptic curve.

(2) T_{ecpa} the time for executing a point addition operation of elliptic curve.

(3) T_h the time for executing a one-way hash function.

(4) T_{inv} the time for executing a modular inversion operation.

In our proposed protocol, the user registration takes one hash operation $h(PW \| a)$ at the user side and one scalar multiplication of elliptic curve $R = (h(PW \| a)/(h(username) + s))P$ at the server side. In the authentication phase, three scalar multiplication operations are needed to compute $V = bR$, $W = bh(PW \| a)P$ and $SK' = bh(PW \| a)$; four one-way hash function operations are required to compute $h(username), h(PW \| a), Auth_s$ and $Auth_u$ at the user side. The SIP server takes three scalar multiplication operations to obtain $W' = (h(username) + s)V, S$ and$_C$ SK ; and three one-way hash function operations to get $h(username), Auth_s$ and $Auth_u$.

Table 1 shows that He *et al.*'s protocol has slightly better performance than our proposed protocol. However, in He *et al.*'s protocol, the SIP server needs to store a hashed password for verification purposes, which makes the solution hard to scale up. The protocol proposed by Zhang *et al.* can overcome this weakness, but their protocol suffers from impersonation attacks as described in Section 4. Since our protocol can reduce the operations of scalar multiplication operation of elliptic curve, it is more efficient than Zhang *et al.*'s protocol, as shown in Table 1. Compared with Zhang *et al.*'s protocol, our protocol not only can resist impersonation attacks but also reduce the computational costs. So, it is more suitable for SIP.

8. CONCULSION

In this study, we reviewed some authentication key agreement protocols for SIP. Zhang *et al.*'s protocol provided some unique properties such as no password or verification table stored on the SIP server compared with other related protocols. Unfortunately, we found that Zhang *et al.*'s protocol was vulnerable to the impersonation attack. This means an adversary can compute other user's privacy key by using the adversary's privacy key, and then impersonate other legal user to cheat the SIP server. To address the problem, we proposed a much improved protocol based on Zhang *et al.*'s protocol. The proposed protocol not only inherited the merits of Zhang *et al.*'s protocol, *i.e.* no need to maintain any password or verification table on the SIP server, but also solved the security problem of suffering from impersonation attacks with low computational costs.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China [Grant numbers 61303237, 61272469, 61075063], Wuhan Scientific Research Program [Grant number 2013010501010144] and China Postdoctoral Fund [Grant number 2012194091]. The authors would like to thank the anonymous reviewers of the paper for their valuable comments.

REFERENCES

1. J.Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J.Peterson, R.S parks, M.Handley, E.Schooler. SIP: session initiation protocol. *IETF RFC3261*, June 2002.
2. Geneiatakis D, Dagiuklas T, Kambourakis G, Lambrinoudakis C, Gritzalis S. Survey of security vulnerabilities in session initiation protocol. *Communications Surveys & Tutorials IEEE* 2006; 8(3):68-81.
3. Callegari C, Garroppo RG, Giordano S, Pagano M. Security and delay issues in SIP systems. *International Journal of Communication Systems* 2009; 22:1023-1044.

4. M. Badra, P. Urien, "Introducing smartcards to remote authenticate passwords using public key encryption", *Proceedings of 2004 IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication*, 2004: 123-126.

5. Yang C, Wang R, Liu W. Secure authentication scheme for session initiation protocol. *Computers& security* 2005; 24: 381-386.

6. J. Ring, K.-K. R. Choo, E. Foo, and M. Looi. A new authentication mechanism and key agreement protocol for sip using identity-based cryptography. *AusCERT R&D Stream* 2006; 61-72.

7. F. Wang and Y. Zhang. A new provably secure authentication and key agreement mechanism for SIP using certificateless public-key cryptogrphy. *Computer Communication*s 2008; 31:2142-2149.

8. T. Guillet, A. Serhrouchni, M. Badra. Mutual Authentication for SIP: A Semantic Meaning for the SIP Opaque Values. Proceedings of New Technologies, Mobility and Security, Tangier, Marrocco, 2008: 1-6.

9. Yi-Pin Liao, Shuemn-Shyang Wang. A new secure password authenticated key agreement scheme for SIP using self-certified public keys on elliptic curves. *Computer Communications* 2010; 33:372-380.

10. L.Wu, Y. Zhang, F. Wang A new provably secure authentication and key agreement protocol for SIP using ECC. *Computer Standards & Interfaces* 2009; 31:286-291.

11. EJ Yoon, KY Yoo, *et al*..A secure and efficient SIP authentication scheme for converged VoIP networks. *Computer Communications* 2010; 33:1674-1681.

12. Gokhroo M.K., Jaidhar C.D., Tomar A.S. Cryptanalysis of SIP Secure and Efficient Authentication Scheme. *Proceedings of ICCSN 2011* 2011; 308-310.

13. Jia Lun Tsai. Efficient Nonce-based authentication scheme for session initiation protocol. *International Journal of Network Security* 2009; 9:12-16.

14. Arshad R, Ikram N. Elliptic curve cryptography based mutual authentication scheme for session initation protocol. *Multimedia Tools and Applications* 2011; DOI: 10.1007/s11042-011-0787-0.

15. Debiao He, Jianhua Chen and Yitao Chen. A secure mutual authentication scheme for session initiation protocol using elliptic curve cryptography. *Security and Communication Networks* 2012; DOI:10.1002/sec.506.

16. Chia-Hui Wang, Yu-Shun Liu. A dependable privacy protection for end-to-end VoIP via Elliptic-Curve Diffie-Hellman and dynamic key changes. *Journal of Network and Computer Applications* 2011; 34:1545-1556.

17. Liping Zhang, Shanyu Tang, Zhihua Cai. Efficient and Flexible Password Authenticated Key Agreement for VoIP Session Initiation Protocol Using Smart Card. *International Journal of Communication Systems* 2013; DOI: 10.1002/dac.2499.

18. Yoon E, Kim W, Yoo K. Robust and simple authentication protocol for secure communication on the web. In *ICWE 2005, Lecture Notes in Computer Science*, Springer-Verlag: Sydney, Australia, 2005; 352-362.

Table 1. Comparisons with two existing protocols

	He et al. [13]	Zhang et al. [15]	Our protocol
No password or verifier table	No	Yes	Yes
Resist impersonation attacks	Yes	No	Yes
Computational cost (client)	$3T_{ecsm} + 3T_h$	$3T_{ecsm} + 1T_{ecpa} + 6T_h$	$3T_{ecsm} + 4T_h$
Computational cost (server)	$3T_{ecsm} + 3T_h$	$5T_{ecsm} + 1T_{ecpa} + 5T_h + 1T_{inv}$	$4T_{ecsm} + 4T_h$
Computational cost (Total)	$6T_{ecsm} + 6T_h$	$8T_{ecsm} + 2T_{ecpa} + 11T_h + 1T_{inv}$	$7T_{ecsm} + 8T_h$