Weakness in Zhang et al.'s Authentication Protocol for Session Initiation Protocol

Mourade Azrour(✉), Yousef Farhaoui, and Mohammed Ouanan

M2I Laboratory, ASIA Team, Department of Computer Science,
Faculty of Sciences and Techniques, Moulay Ismail University,
Errachidia, Morocco
azrour.mourade@gmail.com, youseffarhaoui@gmail.com,
ouanan_mohammed@yahoo.fr

Abstract. Authentication is the most security service required by Session Initiation Protocol (SIP). In recently years, Zhang et al. proposed for the first time an efficient and flexible authentication protocol for SIP using smart card and Elliptic Curve Cryptography. But, in 2014, Zhang et al. showed that their latest proposed protocol is vulnerable to impersonation attack. In order to improve their protocol, Zhang et al. proposed a second protocol. However, in this work we demonstrate that Zhang et al.'s protocol is vulnerable to server spoofing attack. Furthermore to overcome the weakness of Zhang et al.'s protocol we propose an improved and secured SIP authentication and key exchange protocol. The security analysis shows that our proposed protocol can resist to various attack including server spoofing attack.

Keywords: Session Initiation Protocol · Security · Authentication protocol · Elliptic Curve Cryptography · Smart card · Server spoofing attack

1 Introduction

The Telephony over IP (ToIP) is a service that allows to exchange multimedia flows (voice, text, video.) trough internet; ToIP is based on two types of protocols: signaling protocols and transport protocols. In recently decade, Session Initiation Protocol (SIP) [1] is the most signaling protocol used for establishing, altering and terminating session multimedia between different users. The architecture of SIP consists of a proxy server, redirect server, register server, location server, and User agents.

Authentication is the most security service required for SIP. Since, the original SIP authentication protocol (HTTP Digest Authentication [2]) was found vulnerable to deferent attacks; a large community has been participated by proposing the different protocols based on various mechanisms.

SIP authentication protocols proposed before 2013 [3–7] are based on the password verification using several mechanisms. Then, the password must be shared between the user and the server. The shared password is stored in the server database. Therefore, these protocols are vulnerable to stolen verifier attack. In addition to this attack these protocols suffer from the problem of managing of password's database. In 2013, Zhang et al. [8] proposed a first SIP authentication protocol using the Smart Card. Zhang et al.

M. Ezziyyani et al. (eds.), *Advanced Information Technology, Services and Systems*,
Lecture Notes in Networks and Systems 25, https://doi.org/10.1007/978-3-319-69137-4_22

have demonstrated that their protocol offers several advantages such as mutual authentication secrecy and password updating, and it is secure against replay attacks, server spoofing attack, stolen verifier attack, man in the middle attack, and offline password guessing attack. However, Wu et al. [9], Tu et al. [10], and Jiang et al. [11] showed that the protocol of Zhang et al. [8] is vulnerable to user impersonation attack. In order to solve this attack, Zhang et al. proposed a year after their second protocol [12]. Tu et al. [10] proposed a new secured SIP authentication protocol. Then, Tu et al. prove that authentication phase of their protocol reduce the computing time cost to 75% compared to the same Zhang et al.'s phase. Despite these advantages, Tu et al.'s protocol is demonstrated vulnerable to many attacks by Farash et al. [13], Mishra et al. [14] and Zhu et al. [15]. Then, Farah et al. [13] proposed their SIP authentication protocol. However, in 2015, Chaudhry et al. [16] demonstrated that the Farash et al.'s protocol is defenseless to replay attack and Denial of Service attack. As result, they proposed a new protocol and they have proven that is secure against known attacks. Also, Kumaris et al. [17] noticed that Farah et al.'s protocol is not secured against different attack. So, Kumari et al. proposed a new protocol that can resist user impersonation attack, Off-line password guessing attack, replay attack and man-in-the-middle.

In 2014, Arshad et al. [18] proposed a new SIP authentication protocol. However, in 2016, Lin et al. [19] have discovered that Arshad et al.'s protocol is vulnerable. To overcome this problem Lin et al. proposed a new protocol that is more secure and allows to users to update their password using a new method.

Recently, M. Azrour et al. [20] proved that Jiang et al.'s protocol suffer from server spoofing attack, in order to enhance the security of SIP, M. Azrour et al. proposed their protocol which is secured against various attack. For more information about related protocol please refer to [21–23].

In this paper, we will analysis the security performance of Zhang et al.'s [12] SIP authentication protocol. We will show that is vulnerable to server spoofing attack. Then, we propose our solution to overcome the weakness in Zhang et al.'s protocol. Performance analysis shows that our protocol is more secured if it is compared with Zhang et al.'s protocol.

The remainder of this paper is organized as follows. Section 2 delivers general information on the original SIP authentication protocol. In the Sect. 3, we briefly reviewed the Zhang et al.'s authentication protocol. Then, this protocol is analyzed in Sect. 4. The Sect. 5 presents our proposed protocol. The security and performance of our proposed protocol are analyzed respectively in Sects. 6 and 7. Finally the Sect. 8 concludes the paper.

2 Original SIP Authentication Protocol

HTTP Digest Authentication for SIP is based on the mechanism challenge/response. Before the protocol execution, the client and the server share the password, which is used to verify the client's identity. The messages exchanged between the server and the

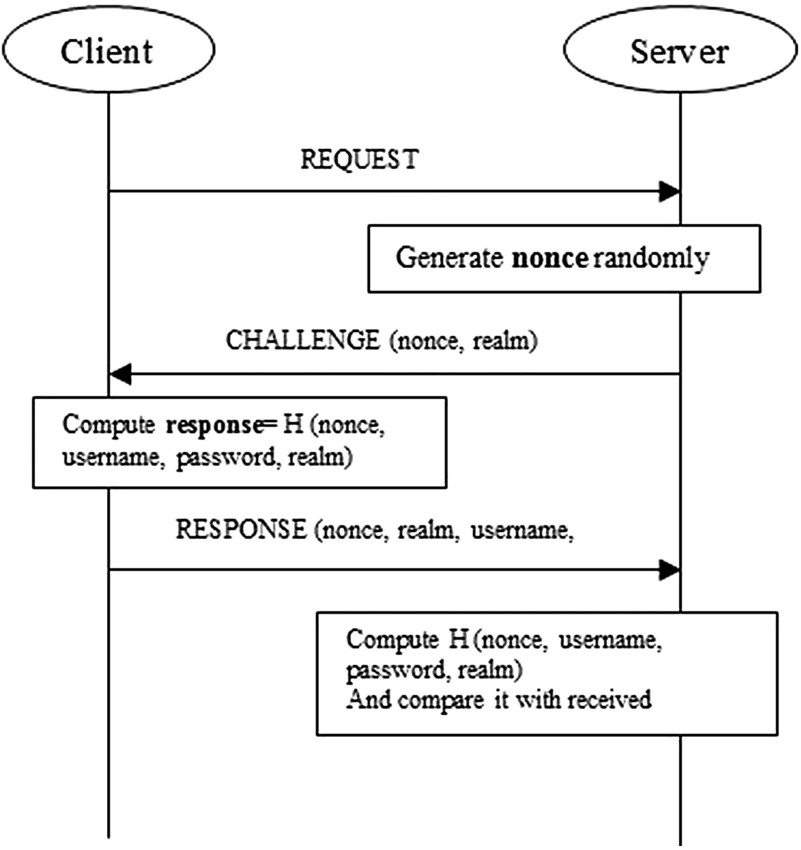

Fig. 1. HTTP digest authentication

client during authentication procedure are illustrated in Fig. 1 and they are described as following.

- **Step 1**. Client → Server: REQUEST
  The client sends a REQUEST to the server.
- **Step 2**. Server → Client: CHALLENGE (nonce, realm)
  After receiving REQUEST; the server generates CHALLENGE that includes a nonce and the client's realm. Note that realm is used to verify username and password. Then, the server sends back CHALLENGE to the client.
- **Step 3**. Client → Server: RESPONSE (nonce, realm, username, response)
  After receiving CHALLENGE from the server, the client computes the response by using received nonce, username, secret password, and realm. response = F(nonce, username, password, realm). Note that F(.) is a one-way hash function. Next, the client sends back the original REQUEST with the computed response, username, nonce and realm.

- **Step 4.** According to username the server extracts the client's password. Then, the server verifies wither nonce is correct or not. If it is correct, the server computes F (nonce, username, password, realm) and uses it to compare it with the response. If they match, the server authenticates the identity of the client.

3 Review of Zhang et al.'s Scheme

In this section, we briefly review Zhang et al.'s protocol as follows. The notations used in this paper are shown in Table 1.

Table 1. Notations and their explanations

Notations	Explanations
U	**The remote user**
S	**The remote server**
X → Y:M	**X sends a message M to Y**
username	The identity of user U
PW	**The password of user U**
$\boldsymbol{E_p(a,b)}$	An elliptic curve equation with order n
s	The long-live secret key of server S
$\boldsymbol{P_{pub} = sP}$	The long-live public key of server S
SK	A session key
$\boldsymbol{h(.), h_1(.), h_2(.)}$	Three secure one-way hash functions
$\boldsymbol{Z_q^*}$	Multiplication group of Zq
‖	The string concatenation operator
$\boldsymbol{E_s}$ (·)	Symmetric key encryption under the key s

3.1 System Setup Phase

The server selects $E_p(a,b)$ with the order n, P $\in E_p(a,b)$, it chooses a random number $s \in_R Z_p^*$ as the secret key. Then, it selects two one-way hash functions, $h(.), h_1(.)$. Finally, the server publishes $\{E_p(a,b), P, h(.), h_1(.)\}$ and keeps s in secret.

3.2 Registration Phase

In this phase, the user registers on the SIP server through a secure channel. The details of this phase are as follows.

R1: The user U selects his/her *username*, password *P*W and a random number $a \in_R Z_p^*$. After that, U computes $h(PW||a)$ and sends $\{h(PW||a), username\}$ to the server through a secure channel.

R2: after receiving the registration information, the server computes $R = \frac{h(PW||a)}{h(username)+s}P$.
R3: The server stores R into the smart card and issues it to U.
R4: Upon receiving the card, U stores a in the card. Then, the card contains (R, a).

3.3 Authentication Phase

Whenever the user wants to login into the remote server, he/she performs the following steps.

A1: $U \rightarrow S : REQUEST(username, V, W)$
U selects a random number $b \in_R Z_p^*$, and computes $V = bR$ and $W = bh(PW||a)P$. Next, the card sends a request message REQUEST(username, V, W) to the server.
A2: $S \rightarrow U : CHALLENGE(realm, Auth_s, S, r)$
After receiving the request message, the server S computes $W' = h(username) + s)V = (h(username) + s) = bh((PW||a)P$. Then, it checks $W \stackrel{?}{=} W'$. If true, the server chooses two random integers c, $r \in_R Z_p^*$, and computes $S = cP, SK = ch(username)W' = cbh(PW||a)h(username)P$, and $Auth_s = h_1(S||W'||SK||r)$. Next, it sends message CHALLENGE(realm, $Auth_s, S, r$) to U over a public channel.
A3: $U \rightarrow S : RESPONSE(realm, Auth_u)$
Upon receiving message REQUEST, U computes $SK' = bh(PW||a)h(username)S = cbh(PW||a)h(username)P$. Then, it cheks whether the equation $Auth_s \stackrel{?}{=} h_1(S||W||SK'||r)$ holds. If so, U computes $Auth_u = h_1(S||W||SK'||r+1)$ and sends RESPONSE(realm, Authu) back to the server. Otherwise, it deletes received information and the protocol stops.
A4: After receiving the RESPONSE message, the server verifies $Auth_u \stackrel{?}{=} h_1(S||W'||SK||r+1)$. If the message is authenticated, the server sets SK a shared session key with user U. Otherwise, it deletes received information and the protocol stops.

4 Cryptanalysis of Zhang et al.'s Scheme

In this section, we prove that the Server spoofing attack is still effective in Zhang et al.'s protocol. Suppose that $\mathcal{A}$ is an attacker. $\mathcal{A}$ can eavesdrops the message

REQUEST{username, V, W} transmitted between server S and user U. Then, he/she can execute server spoofing attack. The details of attack are presented as follows.

Step1. U inputs his/her username and password PW, generates randomly a number $b \in_R Z_p^*$, and computes $V = bR$ and $W = bh(PW||a)P$. Then it sends a request message REQUEST(username, V, W) to S.

Step2. $\mathcal{A}$ eavesdrops message REQUEST(username, V, W) and get username, V, W. He/she generates a random number $r \in_R Z_p^*$. Next, he/she get a value of base point P, and puts its value in $S'(S' \leftarrow P)$. Then, he/she computes $SK = h(username)W$ and $Auth_s^{'} = h_1(S'||W||SK||r)$. Next, $\mathcal{A}$ sends message CHALLENGE(realm, $Auth_s^{'}, S^{'}, r$) to U.

Step3. Upon receiving message CHALLENGE (realm, $Auth_s^{'}, S^{'}, r$), U computes $SK' = bh(PW||a)h(username)S'$ and verifies if $Auth_s' \stackrel{?}{=} h_1(S'||W||SK'||r)$. The user will find true because:
$W \leftarrow bh(PW||a)P$ and $S' \leftarrow P$

So

$$\begin{aligned} SK =& h(username)W \\ =& h(username)bh(PW||a)P \\ =& bh(PW||a)h(username)P \\ =& bh(PW||a)h(username)S' \\ =& SK' \end{aligned}$$

As result, user U authenticates attacker $\mathcal{A}$ and sends to him RESPONSE thinking that he/she communicate with a legal server S.

According to previous analysis, the adversary can easily impersonate identity of server at any time. The user U does not know whether the one he contacts is that the valid server or not. So the adversary can impersonate the server successfully. Therefore, Zhang et al.'s protocol is vulnerable to the server spoofing attack.

5 Our Proposed Protocol

In order to overcome weakness in Zhang et al.'s protocol, we propose an improved and secured authentication and key agreement protocol for SIP. Our protocol consists of four phases, which are system setup phase, registration phase, authentication and key agreement phase, and password changing phase.

5.1 System Setup Phase

In this phase, the server selects an elliptic curve equation $E_p(a, b)$, over a finite field F_q, an additive group G of order p and P a base point generator with order n over equation

$E_p(a, b)$, n is a large prime of height entropy. Then, the server picks a random integer $s \in_R Z_p^*$ as its secrete key. Next, the server chooses three one-way hash functions $h(\cdot)$, $h_1(\cdot)$ *and* $h_2(\cdot)$. Finally, the server publishes all parameters except its private key s, which it is saved secretly.

5.2 Registration Phase

In this phase the user and server S perform the following steps over a secured channel.

R1: The user U chooses freely his/her *username*, *password* W and a random number $a \in_R Z_p^*$. After that, U computes $h(PW||a)$ and sends $\{h(PW||a), username\}$ to the S.

R2: After receiving the registration information, the server computes $R = h(PW\|a) \oplus h(username\|s)P$ Then, the server stores R into the smart card and delivers it to U.

R3: Upon receiving the card, U stores a in the card. Therefore, user card contains (R, a)

5.3 Authentication and Key Agreement Phase

As illustrated in Fig. 2, whenever U wishes to log into S, he/she have to inserts his/her smart card in card reader and inputs his/her *username* and password PW. Next, the following steps will be executed between server S and user U.

- ***Auth1***: $U \rightarrow S$: REQUEST(username, V, W,T_1)
  After inserting the smart card in card reader and inputting the username and password. The user's smart card chooses a random $b \in_R Z_p^*$, and computes $V = bR$, $= h(PW||a)$, and $W = bXP$. Then, he/she sends a message REQUEST(*username*, V, W,T_1) to the server over a public channel. T_1 denotes the current timestamp here.
- ***Auth2:*** $S \rightarrow U$: CHALLENGE(realm, $Auth_s$,S, r,T_3)
  Upon receiving the request message form U at time T_2, S verifies validity of $T_2 - T_1 \leq \Delta T$. If it is OK, S computes $Y = h(username||s)$ and $W' = V \oplus Y$. Then, it verifies $W \stackrel{?}{=} W'$. If it holds, the server S picks randomly two integers $c, r \in_R Z_p^*$. Then, it computes $S = cP, K = cW', SK = h_1(W'||r||YP)$ and $Auth_s = h_2(W'||SK||r||K||S)$. Next, it sends message CHALLENGE(realm, $Auth_s, S, r, T_3$) to U over a public channel.
- ***Auth3***: $U \rightarrow S$: RESPONSE(realm, $Auth_u$)
  Once the user U receives the CHALLENGE message form S at time T_4, U verifies validity of $T_4 - T_3 \leq \Delta T$. If is not fresh, U stops the process. Otherwise, U calculates $K' = bXS$ and $SK' = h_1(W\|r\|(R \oplus X)P)$, and checks $Auth_s \stackrel{?}{=} h_2(W\|SK'\|r\|K'\|S)$. If it is true, the server is authenticated. Then, user U computes $Auth_u = h_2(W\|SK'\|r+1\|K'\|S)$ and sends RESPONSE(realm, $Auth_u$) back to server S. Otherwise, it stops the protocol and deletes received and calculated parameters.

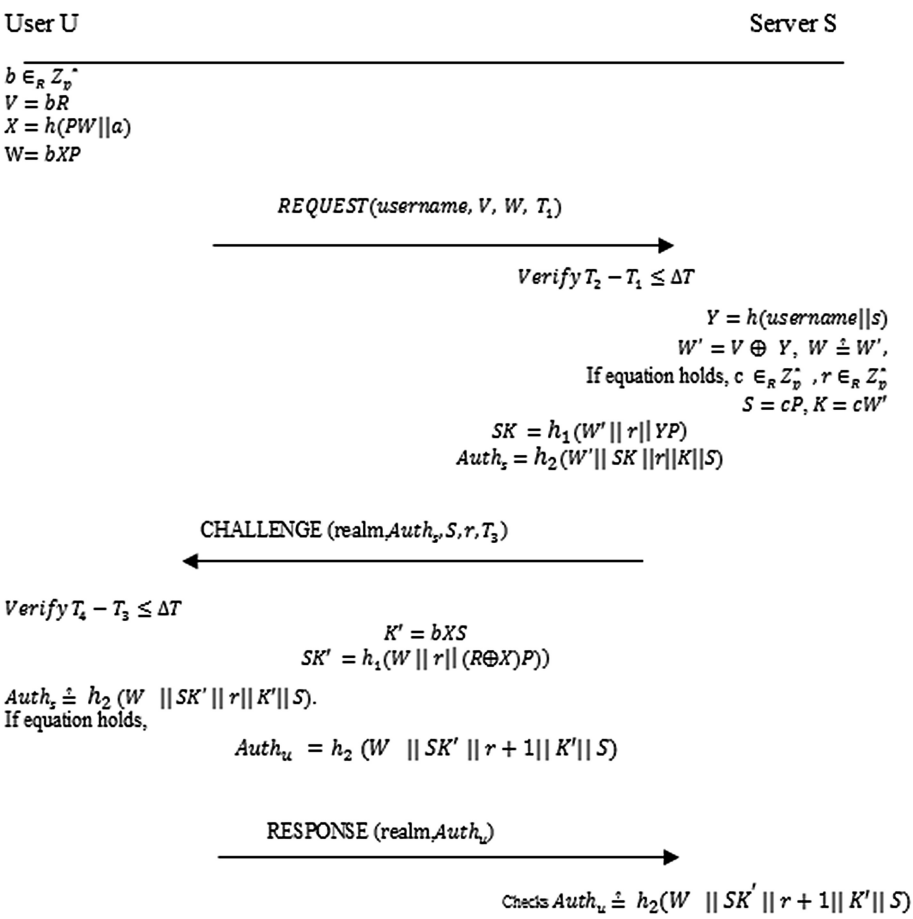

Fig. 2. Authentication phase of our proposed scheme

– **Auth4**:

After receiving the RESPONSE message, the server verifies $Auth_u \stackrel{?}{=} h_2(W||SK||r+1||K'||S)$. If it holds, the user U is authenticated and server S sets SK a shared session key with U. Otherwise, it stops the protocol and deletes received and calculated parameters.

5.4 Password Changing Phase

When the user U wants to update its password, it needs to agree on a session key with the server via the authentication phase in advance. The details of this phase are described as following.

Pass1. $U \rightarrow S$: $(username, e, New_u)$
The user U chooses its new password PW^* and two random integers $a^*, e \in_R Z_p^*$ and computes $h(PW^*||a^*)$ and $tag_u = h(username||e||h(PW^*||a^*))$, it then uses SK to encrypt the new parameters, $New_u = E_{KS}(username||e||h(PW^*\|a^*)\|tag_u)$. Next, it sends message $(username, e, New_u)$ to server.

Pass2. $S \rightarrow U$: (New_s)
Upon receiving the information, the server decrypts the message and then checks the validity of the authentication $tag_u \stackrel{?}{=} h(username||e||h(PW^*\|a^*))$. If it is valid, the server computes the new secret information $R^* = h(PW^*\|a^*) \oplus h(username\|s)P$ and $tag_s = h(username||e+1||R^*)$. Then, it sends encryption information $New_s = E_{KS}(R^*||tag_s)$ back to user.

Pass3. The user U decrypts received message and verifies the validity of $tag_s \stackrel{?}{=} h(username||e+1||R^*)$. If it is valid, U stores R^* *and* a^* in its smart card.

6 Security Analysis

6.1 Mutual Authentication

Mutual authentication means that both the user and server are authenticated to each other within the same protocol. In the proposed scheme the server can authenticate user after receiving REQUEST by checking W, and after receiving RESPONSE by checking $Auth_u$. Upon receiving message CHALLENGE user can authenticate the server by testing validity of $Auth_s$. As result, our protocol provides mutual authentication.

6.2 Session Key Secrecy

In our protocol the session key is computed in this way $SK = h_1(W'\|r\|YP)) = h_1(W\|r\|(R \oplus X)P))$. Since, PW, a, and s are secret, the session key cannot be calculated by anyone except the server and the client. Therefore, our proposed protocol provides session key secrecy.

6.3 Server Spoofing Attack

Our scheme can resist against server spoofing attack. Suppose that an attacker $\mathcal{A}$ wants to impersonate the server and spoof user U, $\mathcal{A}$ has to computes $Auth_s$. However, $\mathcal{A}$ doesn't have any information about a server secret key s. Then, $\mathcal{A}$ can't compute K *and* SK. Therefore, he cannot forge a valid CHALLENGE message.

6.4 User Impersonation Attack

Assume that attacker $\mathcal{A}$ wishes to connect to the server as legitimate user U. $\mathcal{A}$ has to prove its validity by forging two messages REQUEST(username, V, W,T_1) and RESPONSE(realm, $Auth_u$,T_3). While $\mathcal{A}$ need to know some secret information *PW and a*. Therefore, he/she is not capable to send the two validate messages. As result, our scheme can resist user impersonation attack.

6.5 Denning-Sacco Attack

In our scheme, the session key is calculated in this way $SK = h_1(W'\|r\|YP)) = h_1(W\|r\|(R \oplus X)P))$. If an attacker has obtained it, he will have to break the one-way hash function to get YP *or* $(R \oplus X)P$. Then, he has to face Elliptic Curve Cryptography if he wants to guess the user password. So, the proposed scheme is secure against Denning Sacco attack.

6.6 Replay Attack

Suppose that the adversary Alice intercepts the messages REQUEST(username, V, W, T1) and RESPONSE(realm, $Auth_u$) and try to impersonate a legitimate user U. However, she cannot calculate V, W, and $Auth_u$. Since she don't know server secret key. Alice has to face the ECDLP, if she wants to get the correct one by guessing the secret key s from V or W. after replaying REQUEST or RESPONSE the server will detect the attack via comparing if $W \stackrel{?}{=} V \otimes Y$ or $Auth_u \stackrel{?}{=} (W||SK'||r+1||K'||S)$.

Now, Suppose that Alice intercepts the message CHALLENGE(realm, $Auth_s, S, r$) and try to replay it to impersonate the legal server. In order, to be authenticated by the user, Alice have to compute the value of $Auth_s = h_2(W'||SK||r||K||S)$ using secret PW, a, K, and SK. Since Alice don't have any information about secret parameters she cannot computes a valid $Auth_s$. As result, the proposed protocol withstands replay attack.

Table 2. Security performances

Attacks	Zhang et al. [8]	Zhang et al. [12]	Tu et al. [10]	Jiang et al. [11]	Ours
Stolen verifier	Yes	Yes	Yes	Yes	Yes
Denning-sacco	Yes	Yes	–	–	Yes
Password guessing	Yes	Yes	Yes	Yes	Yes
Replay	Yes	Yes	Yes	Yes	Yes
Man in the middle	Yes	Yes	No	–	Yes
Server spoofing	No	No	No	No	Yes
Impersonation	No	Yes	No	No	Yes
Mutual authentication	Yes	Yes	Yes	Yes	Yes
Session key secrecy	Yes	Yes	–	Yes	Yes

Table 3. Computational comparisons between our protocol and related protocols

Phase		Zhang et al. [12]	Ours
Registration	User	$1T_h$	$1T_h$
	Server	$1T_h + 1T_{pm} + 1T_{inv}$	$1T_h + 1T_{pm}$
Authentication	User	$4T_h + 3T_{pm}$	$4T_h + 3T_{pm}$
	Server	$3T_h + 3T_{pm}$	$4T_h + 3T_{pm}$
Total		$9T_h + 7T_{pm} + 1T_{inv}$	$10T_h + 7T_{pm}$

6.7 Stolen Verifier Attack

In the proposed scheme, any user's secret is stored in server database. So, the attacker can't obtain the user's secret information from server. Therefore, our proposed protocol is secure against stolen verifier attack.

6.8 Offline Password Guessing Attack

Suppose that an attacker records all messages (REQUEST, CHALLENGE and RESPONSE) transmitted between user and server, then extract $username, V, W, realm, Auth_s, S, r\ and\ Auth_u$, and tries to guess the password PW^* and verifies its correctness. Since, the attacker does not know any information about values of s, a, b, and c. He/she can't compute K, SK. Then, he can't verify the calculated $V, W, Auth_s\ or\ Auth_u$.

If attacker steals user card he can get $R\ and\ a$, However, he must to know s to check $h(PW||a) \oplus h(username||s)P$. Therefore, our proposed scheme is safe against password guessing attack.

6.9 Man-in-the-Middle Attack

In our protocol all messages are authenticated by server or user, to know their origin. In addition, at the end of authentication, the session key is shared between user and server, so the following messages will be encrypt using session key. To replay these messages, an attacker needs to know a session key. But, he cannot calculate it since he/she does not know $s, a, X, PW\ and\ b$. As result, our protocol is secure against Man-in-the-middle attack.

7 Performance Comparison

In this section, we will compare the performance and computation cost of our proposed protocol with Zhang et al.'s protocol. In this comparison a very lightweight operations like string concatenation operation, Exclusive-OR operation are not examined, because there computation cost is negligible. The notations used are illustrated as follows.

T_h The computational cost of one-way hash operation
T_{pm} The computational cost of elliptic curve point multiplication

T_{inv} The computational cost of modular inversion
T_{Eks}/T_{Dks} The computational cost of Encryption/Decryption algorithm

In the registration phase of our protocol the user uses one hash function and the server computes $1T_h + 1T_{pm}$. The computational costs of the user side and server side in our protocol's authentication phase are $4T_h + 3T_{pm}$ and $4T_h + 3T_{pm}$. In the password changing phase the user computes $3T_h + 1T_{EKs} + 1T_{DKs}$ and the server computes $3T_h + 1T_{pm} + 1T_{EKs} + 1T_{DKs}$.

According to the Table 3 we can observe that modular inversion operation is not used in the registration phase of our protocol. So, this phase is faster than the same phase of Zhang et al.'s protocol. In the Table 2, we can see that our protocol is secured against different attacks especially server spoofing attack, which is effective in the protocol of Zhang et al. Moreover, Zhang et al.'s protocol consist on three phases: System setup phase, Registration phase, and Authentication phase; so it is impossible to change the password or it's not clear how it can be changed. Contrary, our protocol defined Password changing phase in addition to the last three cited phases. Therefore, we can say that our protocol is suitable for applications developed on the base of SIP.

8 Conclusion

In this paper, we have showed that Zhang et al.'s protocol is vulnerable to server spoofing attacks. In order to overcome this weakness we proposed an efficient and secure SIP authentication scheme. According to our analysis, our proposed protocol is secure against various attacks and can provide many security services.

References

1. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E.: SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard). Updated by RFCs 3265, 3853, 4320, 4916, 5393, June 2002
2. Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., Stewart, L.: HTTP Authentication: Basic and Digest Access Authentication, June 1999
3. Durlanik, A., Sogukpinar, I.: SIP authentication scheme using ECDH. World Enformatika Soc. Trans. Eng. Comput. Technol. **8**, 350–353 (2005)
4. Huang, H., Wei, W., Brown, G.E.: A new efficient authentication scheme for session initiation protocol. In: Proceedings of the 9th Joint Conference on Information Sciences (2006)
5. Yoon, E.J., Yoo, K.Y., Kim, C., Hong, Y.S., Jo, M., Chen, H.H.: A secure and efficient SIP authentication scheme for converged VoIP networks. Comput. Commun. **33**(14), 1674–1681 (2010)
6. Liu, W., Koenig, H.: Cryptanalysis of a SIP authentication scheme. In: 12th IFIP TC6/TC11 International Conference, CMS 2011. Lecture Notes in Computer Science, vol. 7025, pp. 134–143 (2011)
7. Xie, Q.: A new authenticated key agreement for session initiation protocol. Int. J. Commun. Syst. **25**(1), 47–54 (2012)

8. Zhang, L., Tang, S., Cai, Z.: Efficient and flexible password authenticated key agreement for voice over internet protocol session initiation protocol using smart card. Int. J. Commun. Syst. **27**(11), 2691–2702 (2013)
9. Wu, K., Gong, P., Wang, J., Yan, X., Li, P.: An improved authentication protocol for session initiation protocol using smart card and elliptic curve cryptography. Rom. J. Inf. Sci. Technol. **16**(4), 324–335 (2013)
10. Tu, H., Kumar, N., Chilamkurti, N., Rho, S.: An improved authentication protocol for session initiation protocol using smart card. Peer-to-Peer Netw. Appl., 1936–6442 (2013). doi:10.1007/s12083-014-0248-4
11. Jiang, Q., Ma, J., Tian, Y.: Cryptanalysis of smart-card-based password authenticated key agreement protocol for session initiation protocol of zhang et al. Int. J. Commun. Syst. **28**(7), 1340–1351 (2014)
12. Zhang, L., Tang, S., Cai, Z.: Cryptanalysis and improvement of password-authenticated key agreement for session initiation protocol using smart cards. Secur. Commun. Netw. **7**, 2405–2411 (2014)
13. Farash, M.S.: Security analysis and enhancements of an improved authentication for session initiation protocol with provable security. Peer-to-Peer Netw. Appl. doi:10.1007/s12083-014-0315-x
14. Mishra, D., Das, A.K., Mukhopadhyay, S.: A secure and efficient ECC-based user anonymity-preserving session initiation authentication protocol using smart card. Peer-to-Peer Netw. Appl. doi:10.1007/s12083-014-0321-z
15. Zhu, W., Chen, J., He, D.: Enhanced authentication protocol for session initiation protocol using smart card. Int. J. Electr. Secur. Digital Forensics **7**(40), 330–342 (2015)
16. Chaudhry, S.A., Mahmood, K., Naqvi, H., Sher, M.: A secure authentication scheme for session initiation protocol based on elliptic curve cryptography. In: 2015 IEEE International Conference on Computer and Information Technology, Ubiquitous Computing and Communications. Dependable, Autonomic and Secure Computing, Pervasive Intelligence and Computing (2015)
17. Kumari, S., Chaudhry, S.A., Wu, F., Li, X., Farash, M.S., Khan, M.K.: An improved smart card based authentication scheme for session initiation protocol. Peer-to-Peer Netw. Appl. **10**, 92–105 (2015)
18. Arshad, H., Nikooghadam, M.: An efficient and secure authentication and key agreement scheme for session initiation protocol using ECC. Multimed Tools Appl. **75**, 181–197 (2014)
19. Lin, H., Wen, F., Du, C.: An anonymous and secure authentication and key agreement scheme for session initiation protocol. Multimed Tools Appl. **76**, 2315–2329 (2016)
20. Azrour, M., Farhaoui, Y., Ouanan, M.: A new secure authentication and key exchange protocol for session initiation protocol using smart card. Int. J. Netw. Secur. **19**(6), 866–875 (2017). doi:10.6633/IJNS.201711.19(6).2)
21. Azrour, M., Ouanan, M., Farhaoui, Y.: SIP authentication protocols based on elliptic curve cryptography: survey and comparison. Indones. J. Electr. Eng. Comput. Sci. **4**(1), 231–239 (2016)
22. Azrour, M., Farhaoui, Y., Ouanan, M.: Cryptanalysis of Farash et al.'s SIP authentication protocol. Int. J. Dyn. Syst. Differ. Equ. (in press)
23. Azrour, M., Farhaoui, Y., Ouanan, M., et al.: A server spoofing attack on Zhang et al. SIP authentication protocol. Int. J. Tomogr. Simul. **30**(3), 47–58 (2017)