An Efficient Three Factor Based Remote User Authentication Protocol for Distributed Networks

Ashish Singh, Kakali Chatterjee

▶ **To cite this version:**

Ashish Singh, Kakali Chatterjee. An Efficient Three Factor Based Remote User Authentication Protocol for Distributed Networks. 15th IFIP International Conference on Computer Information Systems and Industrial Management (CISIM), Sep 2016, Vilnius, Lithuania. pp.682-693, 10.1007/978-3-319-45378-1_59 . hal-01637488

HAL Id: hal-01637488

https://inria.hal.science/hal-01637488

Submitted on 17 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

An Efficient Three Factor Based Remote User Authentication Protocol for Distributed Networks

Ashish Singh[1] and Kakali Chatterjee[1]

[1]Department of Computer Science & Engineering
[1]National Institute of Technology Patna-800005 Bihar (India)

Abstract In distributed networks, one major security drawback is to identify the legitimate remote users of a web service on the Internet. To eliminate this security problem, many researchers have been proposed smart card based remote user authentication for secure communication in wireless networks. The wireless networks mostly use password based protocols that are based on two factors- smart card and PIN. But, this type of authentication protocols are susceptible to password guessing attack, stolen verifier attack, replay attack etc. In this paper, we propose a three factor based mutual authentication protocol using smart card in distributed networks, which resists all possible attacks. This protocol is suitable for hand held devices due to the low computational and communicational cost.

Keywords Mutual Authentication, Smart Card, Diffie-Hellman Key Exchange

1 Introduction

Modern day's Internet is connecting different types of devices which are communicating with each other in different types of distributed networks. In distributed networks secure communication is challenging as the network is based on client server model where the server may possibly be distributed and replicated. Thus, if a remote user wants to get services, he must authenticate himself in the network.

On the basis of the Lamport's [1] authentication scheme, many single servers authentication protocols are found in the literature. However, his scheme required verification tables, which can be hacked by hackers. When the user access services from more than one server, the single-server authentication schemes become highly inconvenient in a distributed environment. Hence, many multi-server user authentication schemes have been proposed by the researchers [2-12]. Among these protocols, some suffers from the parallel session attack [10,12] and the server spoofing attack and some does not resist replay attack, impersonation attack and fails to proof perfect forward security [9,11]. The concept of dynamic ID-based authentication scheme are found in literature [13-17]. These schemes uses smart cards for distributed systems. Sood et al. scheme [15] is based on elliptic curve cryptography which protects all such attacks. Such authentication schemes based on public key cryptography are very difficult to comprise because of the inherent strength of public key systems, but these schemes are very expensive as the use of public key cryptography involves calculation of exponential operations, which needs a lot of processing time. So, the computational cost and efficiency will increases in such cases.

From the literature, it can be summaries that a multi-server authentication scheme must have mutual authentication with no verification table and low computation and communication cost. Also, the remote user authentication will able to resist following security attack such as insider attack, impersonation attack, replay attack, password guessing attack, stolen-verifier attack and server spoofing attack.

To support these features, the paper proposes a multi-server authentication scheme for remote user, which utilizes three factor- a smart card, a password and a token for authenticating user. It also provide easy password change phase to user without replacing card. This scheme can resist many attacks such as insider attack, impersonation attack, replay

attack, password guessing attack, stolen-verifier attack and server spoofing attack.
The rest of the paper is organized as follows: Section 2 discusses cryptanalysis of Chen et al. scheme, Section 3 discusses proposed mutual authentication protocol based on Diffie-hellman key agreement, Section 4 discusses the security and performance of the recommended system. Finally, the paper concludes in Section 5.

2 Cryptanalysis of Chen et al. scheme

The section of this paper discusses cryptanalysis of Chen et al. scheme [9], that is shown in Figure 1 and 2. Researchers assumed that two main capabilities must be considered while check the security strength of the smart-card based authentication. First, the communication link is under the control of the adversary so that he can insert, delete, and modify messages and second, the attacker will able to extracts the secrets of the smart card or both of them.

Impersonation attack This protocol fails to protect from impersonation attack. During registration process the attacker extracts the user identity ID_u and password PW_u. Now from the next communication, he extracts C_1=h($SID_j||R_u$)$\oplus N_c$=h(SID_j||h(ID_u||X))$\oplus N_c$. He keep this values. Now, the attacker sends ID_u, SID_j, C_1 to AC and ID_u, C_1 to the server S_j. After that the attacker receives the token from AC and from C_3= $N_{rc1}\oplus$h(SID||h(ID||X) he get N_{rc1}. Now he generates C_7= h($N_{rc1}||N_c||$ID) and verify himself to AC. After that he receives C_9= h(ID||h(SID||Y)||N_s+1||N_{rc2}+2)$\oplus$h(SID||h(ID ||X)||N_c+1||N_{rc1}+2). After completing the mutual authentication he can generate the session key which is equal to h(ID||h(SID||Y)||N_s+1||N_{rc2}+2)||h(SID||h(ID||X)||N_c+1||N_{rc1} +2||N_{s2}+1||N_{c2}+2). After that the attacker can impersonate as a valid user and exchange messages with server.

Replay attack This protocol fails to resist replay attack. The protocol does not check the validity of the nonce C_1, C_2 which are coming from the user and target server respectively. Also, AC uses these nonce C_1, C_2 for the generation of two other nonce N_c, N_s. If the attacker manages to know the value of R_u, then he can able to know the value of N_c. All other nonce values can be retrieved from this value. Later, by the attacker can replay the message C_1 for further authentication. There is no validity checking of each message packet. So, he can easily perform replay attack to gather the knowledge of authentication exchange.

Man-in-middle attack This attack can easily perform in this scheme. Suppose, an attacker listen all the communication between the user and AC. Also, he has capture the message carrying the token C_9= h(ID||h(SID||Y)||N_s+1||N_{rc2} +2)$\oplus$h(SID ||h(ID||X)||N_c+1|| N_{rc1}+2). Now, he wants to set the session key for communication. The attacker also captures the message C_{10}=$N_{s2}\oplus$h(SID||h(ID||X)||N_c +1||N_{rc1}+2) and C_{11}=$N_{c2}\oplus$h(ID||h(SID||Y)||N_s+1|| N_{rc2}+2) which user and server exchanged during mutual authentication phase. Now following operations, he performed for getting the value of N_{c2}, N_{s2}. He can perform $C_9\oplus C_{10}\oplus C_{11}$ to get the value of $N_{s2}\oplus N_{c2}$. Now using differential cryptanalysis he can able to find the value of N_{s2}, N_{c2}. After that he can easily get: $C_{10}\oplus N_{s2}$ = h(SID ||h(ID || X) || N_c +1 || N_{rc1} +2), $C_{11}\oplus N_{c2}$ = h(ID ||h(SID || Y) || N_s +1 || N_{rc2}+2) Hence, can calculate session key: K_s= h(ID||h(SID||Y)||N_s+1||N_{rc2}+2)||h(SID||h(ID||X)||N_c+1||N_{rc1}+2||N_{s2}+1||N_{c2}+2) Now, all the encrypted message will come to the attacker, he can modify and send to the server.

Dictionary attack This protocol cannot resist the off line dictionary attack. Suppose, an attacker capture the smart card of the user, now the attacker is interested in finding the password. For example, he gathers the information, the password is 6 digits. He can list of numbers and then apply hash function to every number. A rainbow table is used to attack a hashed password in reverse. That means the attacker has a table with possible hashes and look up a matching password. After a match, the attacker goes for online

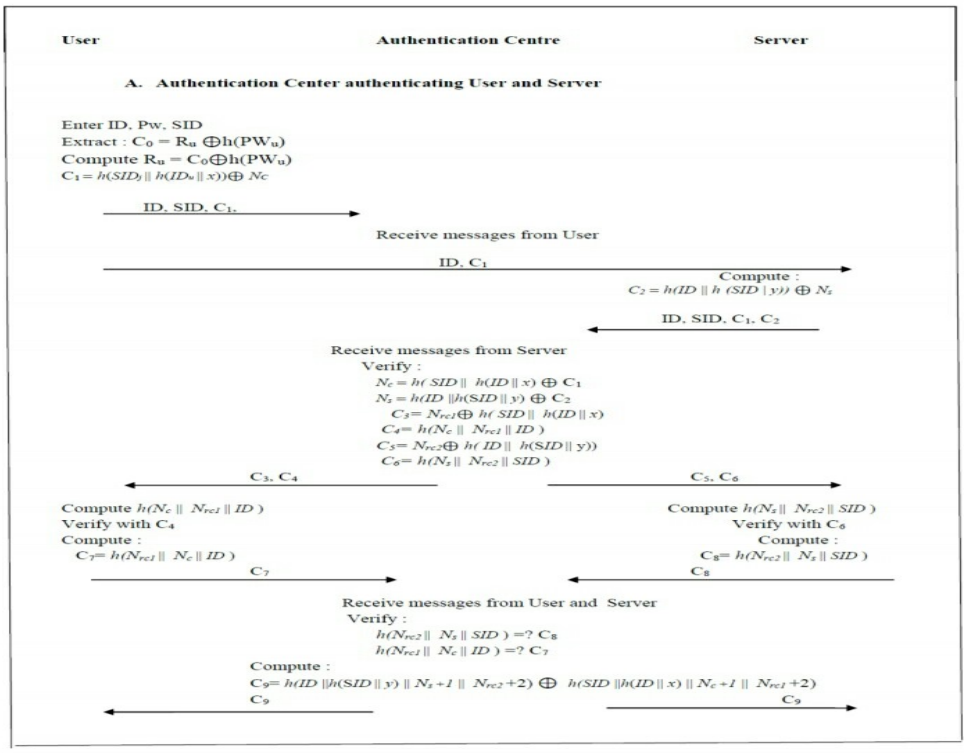

Figure 1: Message transfer in authentication process in Xie et al. Scheme

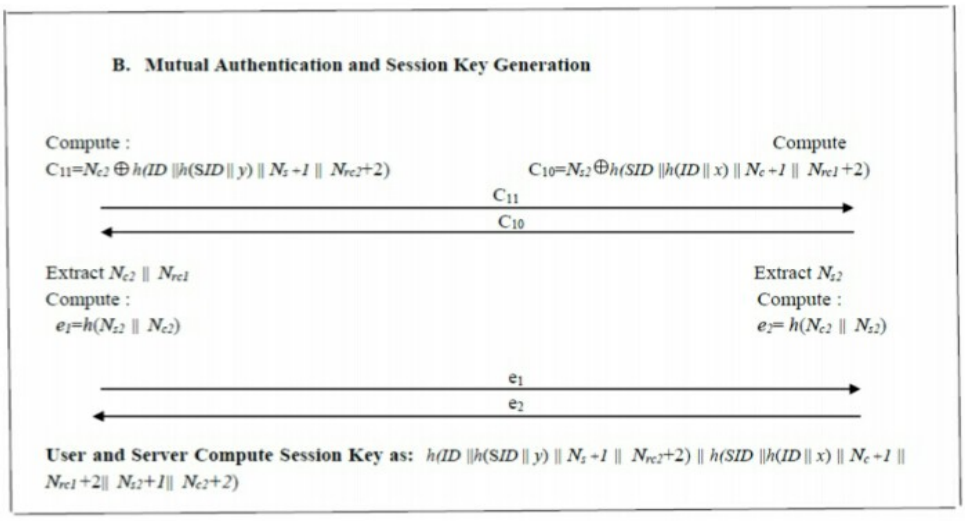

Figure 2: Message transfer in Mutual authentication in Xie et al. Scheme

and use the password to access the system. So, require a password change phase for changing the password.

Perfect forward secrecy In this scheme, we have seen that if the attacker knows the password and stole the smart card, he can retrieve the value R_u = h(ID_u||X) from C_0= $R_u \oplus$h(PW_u). Now from previous session, he can get the nonce value N_c from C_1=h(SID_j||h(ID_u||X))$\oplus N_c$. Similarly, he can get N_s from C_2=h(ID_u||h(SID_j||Y))$\oplus$ N_s and from C_3=$N_{rc1} \oplus$h(SID||h(ID||X) he get N_{rc1}. In such a way the attacker extract all the nonce and calculate the session key. Hence, no perfect forward secrecy is maintained in this protocol.

3 Proposed Authentication Protocol

Our proposed scheme is applied in distributed networks where N number of clients with M number of servers. Initially, all servers and users are registered on the authentication server. After successfully login and authentication, the user and target server directly communicate with each other without interference of authentication server. The user and server authenticate each other and generate the session key for secure communication. Lastly, a password change phase is added. The whole scheme is shown in Fig. 3.

Table 1: Description of notation used in proposed scheme

Symbol	**Definition**
U, S	The user and target server, respectively
AS	The authentication server
UID, PW	User credentials means user ID and user password
SID	Remote server ID
X	AS generated random secret number for user
Y	AS generated random secret number for server
R	User salt
h(.)	Non-invertable one-way hash function
$\oplus$	Bit-wise XOR operation
P	Large positive natural number called prime number
G	Generator of order p-1 in the field Z_p*
T	Timestamp
ΔT	Network delay
a , b , c , d	Random integers generated by user, server and authentication server in {1,. . . . . ,p-1}.

1. **User and Server Registration Phase**
   During this phase all legal users and all servers get registered through the AS. At the time of server registration phase, all steps are given in Table 2.
   (a) In this steps, the server sends a request message containing her/his identity 'SID' to the AS by a communication channel.
   (b) AS selects a secret number Y to calculate h(SID||Y) and send it to S.
   (c) During the user registration phase, the user fills all personal information with UID and PW to the application page of AS. The AS will produce hashed salted password, but never store it. Now AS performs following computation on them.
   (d) On receiving the UID and hashed salted password AS computes: R_u = h(UID||X) and C_0 = $R_u \oplus$h(PW$\oplus$R). Now, AS stores R_u in a smart card and issue it to user. Also AS sends a reply message contains token (C_0) to the user through e-mail. AS also preserves the values of R_u of all registered users. Hence, the user authentication depends upon three factor like, R_u, C_0 and password.

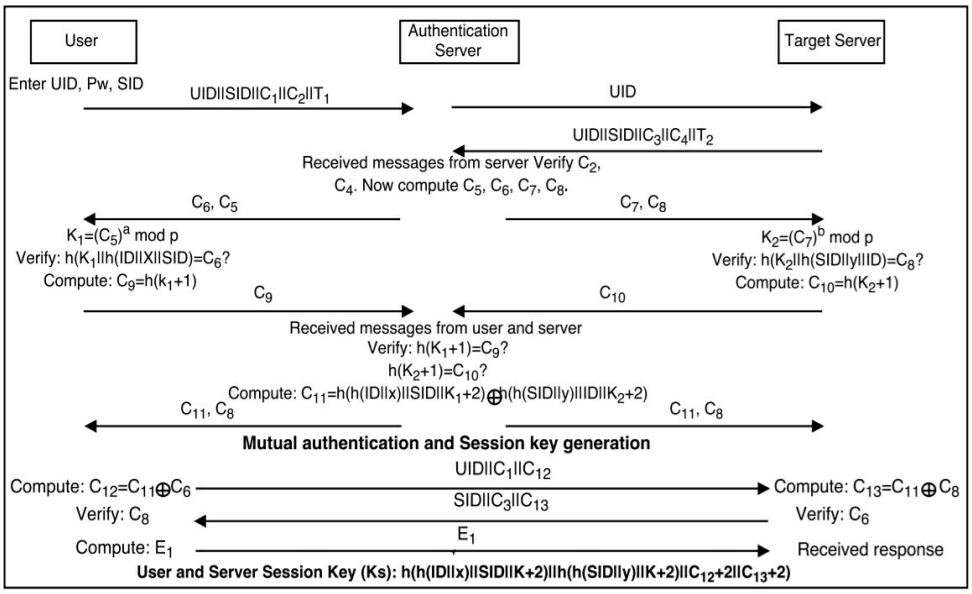

Figure 3: Proposed mutual authentication protocol

Table 2: Flow of proposed mutual authentication scheme

Steps	Message flow	Message format	Message contains
Registration Phase			
1.	S $\rightarrow$ AS	Server_Req_Msg	SID
2.	AS $\rightarrow$ S	Server_Reply_Msg	h(SID$||$Y)
3.	U $\rightarrow$ AS	User_Req_Msg	UID$||$h(PW $\oplus$ R)
4.	AS $\rightarrow$ U	User_Reply_Msg	UID$||C_0$
Login and authentication phase			
1.	U $\rightarrow$ AS	User_Req_Login	UID$||$SID$||C_1||C_2||T_1$
2.	S $\rightarrow$ AS	Server_Req_Login	UID$||$SID$||C_3||C_4||T_2$
3.	AS $\rightarrow$ U	User_Grant_Login	$C_5||C_6$
4.	AS $\rightarrow$ S	Server_Grant_Login	$C_7||C_8$
5.	U $\rightarrow$ AS	User_Challenge	UID$||C_9$
6.	S $\rightarrow$ AS	Server_Challenge	SID$||C_{10}$
7.	AS $\rightarrow$ S	Server_Response	UID$||C_{11}||C_6$
8.	AS $\rightarrow$ U	User_Response	SID$||C_{11}||C_8$
Mutual authentication phase			
1.	U $\rightarrow$ S	User_Req_Msg	UID$||C_1||C_{12}$
2.	S $\rightarrow$ U	Server_Reply_Msg	SID$||C_3||C_{13}$
3.	U $\rightarrow$ S	User_Response	e_1

2. **Login and Authentication Phase**
This phase discusses process of login and authentication of a user. The registered user login to the AS and AS checks that the user is a valid user or not. The steps involve in this process is explained below:
   (a) The user enters his/her smart card to the system and the card reader extracts the value of R_u, UID. Now, he enter the password, C_0 and the target server ID SID with which user desires to communicate. The card reader computes R_u=$C_0 \oplus$h(PW$\oplus$R) and check the two values of R_u. If it is valid then user is connected to the AS through the system. Now, the user randomly chooses a random variable 'a'$\in Z_p^*$ and computes: C_1=(g^a) mod p and C_2=h(R_u||SID||C_1). After computing C_1 and C_2, user sends UID, SID, C_1, C_2 and timestamp to AS.
   (b) The AS sends UID to target server S. On receiving user request, in the form of UID, the server 'S' randomly selects 'b'$\in Z_p^*$ and compute: C_3=(g^b) mod p and C_4=h(h(SID||Y)||UID||C_3). After that target server S sends UID, SID, C_3, C_4 to the AS.
   (c) On receiving messages from user and the server, AS first calculate the timestamp values. If T_2-$T_1 \leq \triangle$T, then the AS checks whether h (h(UID||X)||SID||C_1) is equal to C_2 and h(h(SID||Y)||UID||C_3) is equal to C_4 or not. If two values are equal, then AS authenticates the user and the server, otherwise AS terminates the session. After authenticating, AS chooses randomly 'c'$\in Z_p^*$ and'd'$\in Z_p^*$ computes:C_5= (g^c)(mod p), K_1= (C_1)^c(mod p) = (g^{ac})(mod p), C_6= h(K_1||h(UID||X)||SID), C_7= (g^d)(mod p), K_2= (C_3)^d(mod p) = (g^{bd})(mod p), C_8= h(K_2||h(SID||Y)||UID). Then, AS transfers C_5, C_6 to the user and C_7, C_8 to the target server 'S'. Each message contains present timestamps value.
   (d) After receiving the messages from AS, the user checks timestamp and calculates K_1 as, K_1= (C_5)^a(mod p) = (g^{ac})(mod p). Now user verifies received C_6 as follows, h(K_1||h(UID||X)||SID)=C_6?. If the two values are equal, the user authenticates AS, the user computes C_9 and send it to AS. C_9=h(K_1+1). Similarly, the target server 'S' checks timestamp and computes K_2: K_2= (C_7)^b(mod p) = (g^{bd})(mod p). Now, server verifies received C_8 as follows: h(K_2||h(SID||Y)||UID)=C_8?. If the two values are equal, the server authenticates the AS and computes C_{10}: C_{10}= h(K_2+1). After completion of above operation, server sends C_{10}.
   (e) When AS receives C_9 and C_{10}, it verifies and calculates: h(K_1+1)=C_9? h(K_2+1)= C_{10}?. If the two values are equal, AS ensures authenticity and calculates: C_{11}=h(h(UID||X)||SID||K_1+2) $\oplus$ h(h(SID||Y)||UID||K_2+2). Once the C_{11} is computed, AS sends it to user and server with timestamp. This step marks the end of AS involvement.

3. **Mutual Authentication and Session Key Generation Phase**
In the session key generation phase, the authenticate user and target server communicate directly and generate secure session key given in Table 2. Details of each step are as follows:
   (a) On receiving C_{11} from AS, the user checks timestamp and computes C_{12}=$C_{11} \oplus C_6$. The user transmits (UID||C_1||C_{12}) to server S through the public network.
   (b) Receiving C_{12} target server computes C_6 from C_{12}=$C_{11} \oplus C_6$ and compare the received value of C_6' with the stored value C_6 which AS has sent previously. If it matches than the target server 'S' computes, C_{13}=$C_{11} \oplus C_8$. The server transmits (SID||C_3||C_{13}) to user through the public network. It also calculates secret key K= (C_1)^b mod p.
   (c) When user receives C_{13}, it computes C_8 from C_{13}=$C_{11} \oplus C_8$ and compare the received value of C_8' with the stored value C8 which AS has sent previously. If it matches than the user computes: E = h ($C_{12} \oplus C_{13}$) and send the response

message to target server S. It also calculate secret key K= $(C_3)^a$ mod p. After mutual authentication both of them generate common session key:
K_s=h(h(ID||X)||SID||K+2)||h(h(SID||Y)||K+2)||C_{13}+2||C_{12}+2)
Now user and target server will exchange messages using symmetric encryption (AES) where they use session key K_s for encryption for session time T_s.

4. **Password Change Phase**
The user insert the card, the values UID, R_u is retrieved from the card and C_0, PW is taken from user. The card reader computes R_u, and also compute R'_u=C_0⊕h(PW⊕R) from stored value. It compares the stored value with the computed value and if R_u=R'_u, then the system will accept the user as a valid user. Now, after entering the new password the system will generate the new X_{new} =h(PW_{new}⊕R) and sends it to AS. As the value of R_u remain unchanged, so no new card will be issued to the user. Only AS compute new C_0= R_u⊕ h(PW_{new} ⊕ R), send it to user's e-mail. Next time the user will authenticate himself by using the new C_0. In this way the user can change the password without involving AS.

4 Security and performance of the proposed authentication scheme

This section discusses the security of the proposed authentication protocol, which resists the following attacks:

Dictionary Attack The system will give limited chances for login. After that it will lock the system for security. Moreover, it is not possible to guess token and password correctly at the same time. During login, the user enters UID, her/his password (PW) and the target server ID (SID). The user terminal computes R_u, where R_u= C_0 ⊕ h(PW ⊕ R). Even if the attacker knows C_0, then also very difficult to calculate password and salt. Hence, this protocol will resist offline and online password guessing attack.

Replay Attack The valid and fresh messages completely resists the replay attack. The freshness in messages is because of the use of randomly chosen a, b, c, d, from Z^*_p. Also each message carrying a fresh nonce which is checked the validity of the message. For example, the AS perform following calculation for user and server authentication. h(h(UID||X)||SID||C_1) =? C_2 and h(h(SID||Y)||UID||C_3) =? C_4. The user verifies received C_6 as follows, h(K_1||h(UID||X)||SID)=C_6?. Also the server verifies received C_8 as follows, h(K_2||h(SID||Y)||UID)=C_8?. So, the proposed remote user authentication scheme can withstand replay attack till the adversary doesn't know the value of h(SID||Y) or h(ID||X).

Impersonation Attack Suppose, the attacker track the message (UID||C_0), which the AS sends the user where C_0=R_u⊕h(PW⊕R). Now, the attacker has to insert the correct password to the card reader. The terminal calculate the value of R_u. It is impossible for the attacker to show as a valid user. The attacker can done the mutual authentication, but not able to set the key K_1, K_2. Hence, if the attacker imitates as the valid user, he cannot get the session key without knowing the h(SID||Y) or h(ID||X) values. Thus, the attacker will not get a correct authentication key. So, the proposed protocol resists impersonation attack.

Insider Attack During user registration with the authentication server the user provide her/his password in the form of (PW⊕R) instead of simply providing as (PW). The value of 'R', generated randomly by the application site. The AS never stores the token C_0 = R_u⊕h(PW⊕R) or the salted password. So, any insider in the AS won't be able to know the actual password as well as salted password. Hence, the proposed scheme can successfully resists the insider attack.

Man-in Middle Attack The attacker intercepts the messages through which AS communicates C_5, C_6 to the user and C_7, C_8 to the target server 'S' on the public network. Now the attacker replace the value of C_5, C_6 by C'_5, C'_6 and C_7, C_8 by C'_7, C'_8. Now, the

attacker computes the value $K_1^{'}$, $K_2^{'}$ for the intension to listen all messages. But, the session key actually depends on the following factors: K_s=h(h(ID||X)||SID||K+2)||h(h(SID ||Y)||K+2)||C_{13}+2||C_{12}+2). The adversary does not have the values of 'X' and 'Y'. Hence, he will not be able to set a common session key. Thus, this attack is not possible in this scheme.

Table 3: The functionality comparison of our proposed protocol with previous existing protocols

Functionalities	**Banerjee et al. [2]**	**Chuang et al. [6]**	**Xue et al. [17]**	**Li et al. [5]**	**Our Scheme**
Man-in-the-middle attack	Yes	No	No	No	Yes
Impersonation attack	No	No	No	Yes	Yes
Mutual authentication	Yes	No	No	Yes	Yes
Replay attack	No	Yes	Yes	Yes	Yes
Perfect forward secrecy	yes	No	Yes	No	Yes
Insider attack	Yes	Yes	No	No	Yes
Password guessing attack	No	Yes	No	No	Yes
Computational cost for login and authentication	$17T_h$+$17T_X$	$16T_h$	$27T_h$	$27T_h$	$16T_h$+ $10T_{exp}$

Mutual Authentication This scheme provides mutual authentication to the user and the target server. The user transmits (UID||C_1||C_{12}) to server S through the public network.The target server computes C_6 from C_{12}=$C_{11}\oplus C_6$ and compare the received value of $C_6^{'}$ with the stored value C_6 which AS has sent previously. If it matches than the target server transmits (SID||C_3||C_{13}) to user through the public network. It also calculates secret key K= $(C_1)^b$ mod p. When a user receives C_{13}, it computes C_8 from C_{13}=$C_{11}\oplus C_8$ and compare the received value of $C_8^{'}$ with the stored value C_8 which AS has sent previously. If it matches, then the user computes: e_1 = h($C_{12}\oplus C_{13}$) and send the response message to target server S. In this way the scheme provides mutual authentication between the user and target server.

Server Spoofing Attack In proposed protocol, the attacker not be able to provide authenticity of any user cause of servers do not keep any password table, To authenticate the user, server first needs to get authentication from authentication server and can then communicate with the user. The attacker can get the SID of the target server, but it is impossible to know the value of Y because it is randomly generated by the AS and kept secret. Therefore, this scheme resists server spoofing attack.

Perfect Forward Secrecy The scheme is maintaining perfect forward secrecy. Even if the password of the past session is disclosed, then also the attacker cannot able to calculate the past session key. Assume that the attacker knows C_{12}, C_{13}. Now to generate the past session key, the attacker must know the values of h(SID||Y) and h(ID||X), which depend on two secret values X, Y. Now consider anyway the attacker knows these values. Then he has to calculate the value of K where K = (g^{ab}) mod p. Here, the attacker must know a and b to get the key. The attacker will not be able to guess them accurately and get the session key. Hence, it is proved that the scheme is maintaining perfect forward secrecy.

Performance Analysis

The functionality comparison in Table 3 shows that our scheme is more secure, robust, take less amount of time for authentication operations. The notation T_h, T_X and T_{exp} are

shows as the time complexity for hashing function, time complexity for Ex-or operation and time complexity for exponential operation respectively.

5 Conclusions

This paper proposes an advanced and secure technique for remote users in distributed networks over an insecure channel. It resist all possible attacks in distributed networks. The scheme provides mutual authentication between target server and user and also generate different session key for different server. The weakness in Xie and Chen [9] scheme have been successfully removed by our scheme. The security analysis and comparison of the proposal proven that the proposed remote user authentication scheme is efficient, secure and takes less amount of time for essential authentication operation and can resist the major attacks.

References

1. Lamport L. Password authentication with insecure communication. Communications of the ACM. 1981 Nov 1;24(11):770-772.
2. Banerjee S, Dutta MP, Bhunia CT. A perfect dynamic-id and biometric based remote user authentication scheme under multi-server environments using smart cards. InProceedings of the 8th International Conference on Security of Information and Networks 2015 Sep 8 (pp. 58-64). ACM.
3. Li LH, Lin IC, Hwang MS. A remote password authentication scheme for multiserver architecture using neural networks. Neural Networks, IEEE Transactions on. 2001 Nov;12(6):1498-1504.
4. Yoon EJ, Yoo KY. Robust multi-server authentication scheme. In Network and Parallel Computing, 2009. NPC'09. Sixth IFIP International Conference on 2009 Oct 19 (pp. 197-203). IEEE.
5. Li X, Xiong Y, Ma J, Wang W. An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. Journal of Network and Computer Applications. 2012 Mar 31;35(2):763-769.
6. Chuang MC, Chen MC. An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics. Expert Systems with Applications. 2014 Mar 31;41(4):1411-1418.
7. Lee JH, Lee DH. Efficient and secure remote authenticated key agreement scheme for multi-server using mobile equipment. In Consumer Electronics, 2008. ICCE 2008. Digest of Technical Papers. International Conference on 2008 Jan 9 (pp. 1-2). IEEE.
8. Tsai JL. Efficient multi-server authentication scheme based on one-way hash function without verification table. Computers & Security. 2008 Jun 30;27(3):115-121.
9. Xie Q, Chen D. Hash function and smart card based multi-server authentication protocol. In2010 WASE International Conference on Information Engineering 2010 (Vol. 4, pp. 17-19).
10. Zhu H, Liu T, Liu J. Robust and Simple multi-server authentication protocol without verification table. In2009 Ninth International Conference on Hybrid Intelligent Systems 2009 Aug 12 (pp. 51-56). IEEE.
11. Chen TY, Hwang MS, Lee CC, Jan JK. Cryptanalysis of a secure dynamic ID based remote user authentication scheme for multi-server environment. In Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on 2009 Dec 7 (pp. 725-728). IEEE.
12. Shao MH, Chin YC. A novel dynamic ID-based remote user authentication and access control scheme for multi-server environment. In Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on 2010 Jun 29 (pp. 1102-1107). IEEE.
13. Chen Y, Huang CH, Chou JS. A novel multi-server authentication protocol. IACR Cryptology ePrint Archive. 2009;2009:176.

14. Das ML, Saxena A, Gulati VP. A dynamic ID-based remote user authentication scheme. Consumer Electronics, IEEE Transactions on. 2004 May;50(2):629-631.
15. Kalra S, Sood S. Advanced remote user authentication protocol for multi-server architecture based on ECC. Journal of Information Security and Applications. 2013 Sep 30;18(2):98-107.
16. Hsiang HC, Shih WK. Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standards & Interfaces. 2009 Nov 30;31(6):1118-1123.
17. Xue K, Hong P, Ma C. A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. Journal of Computer and System Sciences. 2014 Feb 28;80(1):195-206.