Diffie-Hellman Based Smart-Card Multi-server Authentication Scheme

Article · March 2015

DOI: 10.1109/CICN.2014.173

1 author:

Himanshu Mittal
Jaypee Institute of Information Technology
48 PUBLICATIONS **681** CITATIONS

SEE PROFILE

2014 Sixth International Conference on Computational Intelligence and Communication Networks

Diffie-Hellman based Smart-card Multi-Server Authentication Scheme

Himanshu Mittal
Department of Computer Science
Jaypee Institute of Information Technology
Noida, India
himanshu.mittal@jiit.ac.in

***Abstract*— A secure smart-card multi-server authentication scheme has been proposed using Diffie-Hellman, Hash-Function and XOR. The scheme made no use of verification table, or encryption techniques, or timestamps to generate a session key to provide secure communication between user and server and resists all possible security attacks, such as Man-in-the-Middle attack, Impersonation attack, Insider attack and many more. The scheme proved to be better when compared to Xie and Chen scheme in terms of security and performance.**

Keywords- Multi-server, authentication scheme, Diffie-Hellman, Hash-function.

I. INTRODUCTION

With variety of services present in the interest today, it has become extremely essential to authenticate and secure the flow of information to appropriate recipients. A multi–server authentication scheme is a mechanism wherein a set of multiple servers authenticate a user before allowing access to the services of any server. Generally, there are three participants in a multi-server authentication scheme: users, a group of servers, and the authentication center [17]. In this scheme, the remote user registers only once on the authentication center and is allowed to access services from any of the multiple servers without repeating the registration process.

When the single-server authentication schemes were used on multi-server architecture, they became highly inconvenient and impractical. In 2001, Li et al [6] came up with a simple password authentication scheme for multi-server architecture that used a pattern classification system based on neural networks without any verification table. But, it required time to train the neural networks. The scheme cannot resist password guessing attacks and insider attack [13], did not provide mutual authentication and session key agreement [12] and computation and communication costs were extremely high. In 2003, Lin et al [3] gave a multi-server scheme using ElGamal digital signature and geometric transformations on an Euclidean plane. In 2008, Lee et al. [4] proposed an authenticated key agreement scheme using mobile equipment. However, their scheme cannot add server freely that increases the registration center's card-issue cost. Juang [14] also proposed a multi-server authentication scheme that used symmetric encryption techniques without verification table. But, that scheme suffered from insider attack [8]. In 2008, Tsai [5] proposed an efficient multi-server authentication scheme based on one-way hash function without a verification table. Yoon and Yoo [1] claimed that the Jung and Tsai scheme are vulnerable to privileged insider attacks. Zhu et al [2] claimed that both the scheme Liao and Wang [15] and Tsai [5] suffers from the server spoofing attack and the parallel session attack and proposed a new scheme. Chen et al. [16] showed that Tsai's scheme fails against server spoofing attack, and proposed a novel protocol. Xie and Chen [10] claimed that Chen et al scheme [16] cannot resist off-line password guessing attack and proposed a new scheme. However, Xie and Chen scheme [10] cannot handle attacks like insider attack, impersonation attack and forward secrecy attack.

This paper proposes "Diffie-Hellman based Smart-card Multi-Server Authentication Scheme" which thwarts possible security attacks without using any verification table, or encryption techniques to produce a session key and performs better than Xie and Chen scheme [10].

The structure of the paper is as follows: section 2 reviews the Xie and Chen scheme [10] and its weaknesses; the proposed scheme is presented in section 3; the security and performance analysis comparison with Xie and Chen scheme [10] is done in section 4 and 5 respectively and finally paper is concluded along future work in the last section.

II. REVIEW OF XIE AND CHEN SCHEME

Table I. lists all the relevant notations used in the paper. In 2010, Xie and Chen proposed an authentication scheme using hash-function for multi-server environment [10]. This proposed scheme takes *s* servers, *n* remote users and an authentication centre AC. At the beginning, AC randomly chooses two secret numbers x and y, and generates $h(SID_j \| y)$ for each server S_j with identity SID_j. There are four phases: User Registration, Login, Authentication Centre Authenticates Remote User and Server and Mutual Authentication and Session Key Generation.

TABLE I. NOTATIONS

Symbol	*Definition*
U, S_j	User and jth server, respectively
AC	Authentication Center
ID, Pw	U's identity and password, respectively
SID	Server S_j identity
x	AC's random number for U
y	AC's random number for S_j
K	User random nonce
h(.)	Secure one-way hash function

978-1-4799-6929-6/14 $31.00 © 2014 IEEE
DOI 10.1109/CICN.2014.173

Symbol	*Definition*
⊕	Bit-wise Exclusive-OR(XOR) operation
||	Concatenation operation
A → B: M	A sends message M to B
p	Public large prime number
g	Public primitive element in the GF(p)
a, b, c, d, a', b'	Ephemeral random numbers in {1,……,p-1} generated by user, server and authentication center
x, y	Information held by AC

A. User Registration Phase:

The user chooses identity 'ID' and password 'Pw' and sends them to AC through a secure channel. AC performs following computation:

- *U → AC: ID, Pw*

AC computes

$R_u = h(ID||x)$,
$C_0 = Ru \oplus h(Pw)$.

- *AC → U: C_0*

AC sends C_0 to U to store it on smart card.

B. Login

The following computation is performed when user 'U' wants to generate session key with server S_j with identity SID. U inserts smart card into terminal device and enters password. The smart card computes and sends ID, SID, C_1 to AC and ID, C_1 to server S_j.

$R_u = C_0 \oplus h(Pw)$,
$C_1 = h(SID||R_u) \oplus N_c$.

C. Authentication Centre Authenticates Remote User

- *S_j → AC : ID , SID , C_1 , C_2*

On receiving ID and C_1, S_j generates random nonce N_s and computes

$C_2 = h(ID||h(SID||y)) \oplus N_s$.

S_j sends ID, SID, C_1, C_2 to AC.

- *AC →U: C_3 , C_4; AC → S_j: C_5 , C_6*

On receiving messages, AC extracts N_c and N_s and generate two random nonce N_{rc1} and N_{rc2} for computing C_3, C_4, C_5, C_6

$N_c = h(SID||h(ID||x)) \oplus C_1$,
$N_s = h(ID||h(SID||y)) \oplus C_2$,
$C_3 = N_{rc1} \oplus h(SID||h(ID||x))$,
$C_4 = h(N_c||N_{rc1}||ID)$,
$C_5 = N_{rc2} \oplus h(ID||h(SID||y))$,
$C_6 = h(N_s||N_{rc2}||SID)$.

AC sends C_3, C_4 to U and C_5, C_6 to S_j.

- *U → AC : C_7 ; S_j → AC : C_8*

On receiving C_3 and C_4, U retrieves $N_{rc1} = C_3 \oplus h(SID||h(ID||x))$, computes $C_{4'} = h(N_c||N_{rc1}||ID)$ and verifies with C_4. If it equals then communication is accepted and U computes $C_7 = h(N_{rc1}||N_c||ID)$ and sends it to AC, else the communication is rejected.

Similarly, S_j retrieves $N_{rc2} = C_5 \oplus h(ID||h(SID||y))$ and computes $C_{6'} = h(N_s||N_{rc2}||SID)$ and verify its equality with C_6. If it equals then communication is accepted, and S_j computes $C_8 = h(N_{rc2}||N_s||SID)$ and sends it to AC, else communication is rejected.

- *AC → U: C_9 ; AC → S_j , C_9*

On receiving C_7 and C_8, AC computes

$C_{7'} = h(N_{rc1}||N_c||ID)$,
$C_{8'} = h(N_{rc2}||N_s||SID)$,

AC verifies them with received C_7 and C_8, respectively. If both are same, AC computes and sends C_9 to U and S_j.

$C_9 = h(ID||h(SID||y)||N_s+1||N_{rc2}+2) \oplus h(SID||h(ID||x)||N_c+1||N_{rc1}+2)$.

D. *Mutual Authentication and Session Key Generation Phase:*

- *S_j → U : C_{10}*

On receiving C_9, S_j extracts $C_{10'}$ and generates random nonce N_{s2} to send C_{10} to U.

$C_{10'} = C_9 \oplus h(ID||h(SID||y)||N_s+1||N_{rc2}+2)$,
$C_{10} = N_{s2} \oplus C_{10'}$.

- *U → S_j : C_{11}*

On receiving C_9, U extracts $C_{11'}$ and generates random nonce N_{c2} to compute C_{11} to S_j.

$C_{11'} = C_9 \oplus h(SID||h(ID||x)||N_c+1||N_{rc1}+2)$,
$C_{11} = N_{s2} \oplus C_{11'}$.

- *S_j → U : e_3*

On receiving C_{11}, S_j retrieves N_{c2} to compute e_3 and send it to U

$N_{c2} = C_{11} \oplus h(ID||h(SID||y)||N_s+1||N_{rc2}+2)$,
$e_3 = h(N_{c2}||N_{s2})$.

- *U → Sj : e_4*

On receiving, U retrieves N_{s2} to compute e_4 and send it to S_j.

$N_{s2} = C_{10} \oplus (SID||h(ID||x)||N_c+1||N_{rc1}+2)$,
$e_4 = h(N_{s2}||N_{c2})$.

- *Mutual Authentication and Session Key*

U and S_j compute $e_{3'}$ and $e_{4'}$ and verify them with received values respectively, i.e. e_3 and e_4.as:

$e_{3'} = h(N_{c2}||N_{s2})$,
$e_{4'} = h(N_{s2}||N_{c2})$,

If both values are equal, user 'U' and server 'S_j' define a Session Key as:

$(h(ID||h(SID||y)||N_s+1||N_{rc2}+2)||h(SID||h(ID||x)||N_c+1||N_{rc1}+2)||N_{s2}+1||N_{c2}+2)$.

WEAKNESSES OF XIE AND CHEN SCHEME

- *Insider Attack:* The user registers at AC with password (Pw). Any insider at AC would easily know user password.
- *Forward Secrecy Attack:* As in said in [10], session key is not forward secret in Xie and Chen scheme.

III. THE PROPOSED SCHEME

With In this section, the proposed authentication scheme using Diffie-Hellman is described. There are 's' servers, 'n' users and an authentication centre(AC). At the beginning, AC randomly selects secret values for 'x' and 'y'. 'p' and 'g' are two publicly known variables, where 'p' is a large prime

number and 'g' is the generator of order p-1 in the group $<Z_p^*,x>$. The proposal consists of four phases:

A. Server registration phase,
B. User Registration phase,
C. Authentication of Remote User and Server phase and,
D. Mutual Authentication and Session Key Generation.

A. Server Registration Phase:

Each Server S of identity SID registers at AC.

- *S → AC: SID*

In this, S sends identity 'SID' to the AC by a secure channel.

- *AC →S: h(SID||y)*

  h(SID||y)

AC sends computed value to server S by a secure channel.

B. User Registration Phase:

User 'U' registers at AC by entering his/her ID and password.

- *U → AC: ID, (Pw⊕ K)*

On receiving the 'ID', AC computes:

$R_u = h(ID||x)$,
$C_0 = R_u \oplus h(Pw \oplus K)$.

- *AC → U: C_0*

AC sends C_0 to user over a secure channel and stores it on smart card.

C. Authentication of Remote User and Server:

In this phase, AC allows any registered user to login and access registered servers as follows:

- *U → AC: ID*, SID*, C_1, C_2; U → S_j: ID**

At terminal device, User 'U' inserts his smart card, password and target server 'S_j' identity 'SID*' with which user wants to communicate. The terminal computes

$R_u = C_0 \oplus h(Pw \oplus K)$,
$C_1 = (g^a) \pmod p$; where 'a' ϵ Z^*_p,
$C_2 = h(R_u||SID^*||C_1)$.

Terminal transmit ID*, SID*, C_1, C_2 to AC and ID* to S_j over public network.

- *S_j → AC: ID*, SID*, C_3, C_4*

On receiving U message, S_j computes:

$C_3 = (g^b) \pmod p$; where 'b' ϵ Z^*_p,
$C_4 = h(h(SID||y)||ID^*||C_3)$.

S_j sends ID*, SID*, C_3, C_4 to AC over the public network.

- *AC →U: C_5, C_6; AC → S_j: C_7, C_8*

On receiving messages, AC computes and verifies:

$h(h(ID^*||x)||SID^*||C_1) = C_2$?,
$h(h(SID^*||y)||ID^*||C_3) = C_4$?,

If both equals, AC authenticates both user 'U' and target server 'S_j'. If either is not equal, AC terminates the session.

On authentication, AC chooses randomly 'c' ϵ Z^*_p and 'd' ϵ Z^*_p and computes:

$C_5 = (g^c) \pmod p$,
$K_1 = (C_1)^c \pmod p$,
$C_6 = h(K_1||h(ID^*||x)||SID^*)$,
$C_7 = (g^d) \pmod p$,
$K_2 = (C_3)^d \pmod p$,
$C_8 = h(K_2||h(SID^*||y)||ID^*)$.

Then, AC communicates C_5, C_6 to U and C_7, C_8 to S_j over public network.

- *U → AC: C_9; S_j → AC: C_{10}*

U verifies C_6 as follows,

$K_1 = (C_5)^a \pmod p$,
$h(K_1||h(ID||x)||SID^*) = C_6$?,

If verified, U authenticates AC and sends C_9 to AC.

$C_9 = h(K_1+1)$.

If not, U terminates the session.

Similarly, S_j verifies C_8:

$K_2 = (C_7)^b \pmod p$,
$h(K_2||h(SID||y)||ID^*) = C_8$?,

If verified, S_j authenticates AC and send C_{10} to AC.

$C_{10} = h(K_2+1)$.

If not, S_j terminates the session.

- *AC → U: C_{11}; AC → S_j: C_{11}*

On receiving C_9 and C_{10}, AC verifies:

$h(K_1+1) = C_9$?,
$h(K_2+1) = C_{10}$?,

If both verifies, AC computes C_{11} to send to U and S_j.

$C_{11} = h(h(ID^*||x)||SID^*||K_1+2) \oplus h(h(SID^*||y)||ID^*||K_2+2)$.

D. Mutual Authentication and Session Key Generation Phase

This is the last phase where user 'U' and target server 'S_j' generate session key as:

- *U → S_j: C_{14}*

On receiving 'C_{11}', U chooses a'ϵ Z^*_p and perform following computation:

$C_{12} = C_{11} \oplus (h(h(ID||x)||SID^*||K_1+2)$,
$C_{13} = (g^a) \pmod p$,
$C_{14} = C_{13} \oplus C_{12}$,

U transmits C_{14} to S_j over public network.

- *S_j → U: C_{17}*

Similarly, S_j computes:

$C_{15} = C_{11} \oplus h(h(SID||y)||ID^*||K_2+2)$,
$C_{16} = (g^b) \pmod p$,
$C_{17} = C_{16} \oplus C_{15}$.

S_j sends C_{17} to the U.

- *U → S_j: e_1*

On receiving C_{17}, U performs:

$C_{16'} = C_{17} \oplus h(h(ID||x)||SID^*||K_1+2)$,
$e_1 = h(C_{16'}||C_{13})$.

U sends e_1 to S_j over public network.

- *S_j → U: e_2*

Similarly, S_j computes e_2 and sends to U over network.

$C_{13'} = C_{14} \oplus h(h(SID||y)||ID^*||K_2+2)$,
$e_2 = h(C_{13'}||C_{16})$.

- *Mutual Authentication and Session Key*

This is session key generation step. At the recipient of e_1 and e_2, user U and target server S_j compute e_3 and e_4. If both are equal to e_2 and e_1 respectively, U and S_j authenticate each other and produce session key else terminate the session without session key generation.

$e_3 = h(C_{13}||C_{16'})$ and

$e_4 = h(C_{16}||C_{13'})$,

Session Key defined at user 'U' and target server 'S_j':

$h(h(h(ID||x)||SID||K_1+2)||h(h(SID||y)||ID||K_2+2)||C_{13}+2||C_{16}+2)$.

IV. Security Analysis

- *Password Guessing Attack:* The proposal resists on-line password guessing as AC verifies user before generating session key with server, i.e. in step C, password guessing attack will fail. The offline password guessing attack will also fail as password is only used by user and no important information is generated using the password.
- *Replay Attack:* The replay attack cannot occur as messages communicated among server, user and AC contain elements of freshness such as a, b, c, d, a', b'.
- *Impersonation Attack:* If attacker impersonates as valid user or valid server, he must know user's h(ID||x), C1 and C2 and server's h(SID||y). The server is provided only with user ID for computation. Thus, the attacker will not get the correct authentication key.
- *Insider Attack:* User registers at AC with password of the form (Pw $\oplus$ K) instead of (Pw). So, any insider at AC won't know the user password till value of K is not known, which is a randomly generated value.
- *Stolen-Verifier Attack:* There is no verification table at AC and server because of which stolen-verifier attack is impossible. The AC and server authenticates user on the values provided to them.
- *Man-In-The-Middle Attack:* Since the message generated by server and user for AC contains secret identity of user and server, the messages generated by adversary would fail to get authentication by AC. The adversary would not know the values of 'x' and 'y'.
- *Server Spoofing Attack:* Attacker cannot masquerade as Sj. Server does not contain any verification table, so they cannot authenticate any user directly. The attacker must know h(SID||y) to cheat AC.
- *Authentication Center Spoofing Attack:* Attacker cannot masquerade as AC. User and Server use (ID||x) and (SID||y) in C6 and C8 to authenticate AC.
- *Forward Secrecy Attack:* If x and y discloses, attacker has to know four messages C_1, C_2, C_3 and C_4 to authentication from AC. Since C_1 and C_3 are computed on random values, the attacker will not guess them accurately and generate session key.
- *Denning-Sacco Attack:* The Denning-Sacco attack is where an attacker compromises an old session key and tries to find a long-term private key (e.g. user password or server private key) or other session keys[2]. The proposal resists this type of attack as the attacker cannot get to know K_1 and K_2 and spoof other entities which are in communication with it.
- *Mutual Authentication Key Security:* The mutual authentication key cannot be calculated by outsider since it contains random values K_1, K_2, user secret key and server secret key.
- *Security of Session Key:* The session key cannot be calculated by anyone except server and user as it contains values C_{13}, C_{16}, user authentication key, server authentication key. Only, the authenticated server and authenticated user can generate these values and so the session key. Thus, the session key is secure.

TABLE II. Security Properties

Security Properties	*Our Proposed Scheme*	*Xie and Chen Scheme*
Insider Attack	Yes	No
Forward Secrecy Attack	Yes	No
Impersonation Attack	Yes	Yes
Password Guessing Attack	Yes	Yes
Replay Attack	Yes	Yes
Stolen-Verifier Attack	Yes	Yes
Man-In-The-Middle Attack	Yes	Yes
Server Spoofing Attack	Yes	Yes
Authentication Center spoofing Attack	Yes	Yes
Denning-Sacco Attack	Yes	Yes
Authentication Key Security	Yes	Yes
Session Key Security	Yes	Yes

V. Performance Analysis

The performance analysis for proposed scheme is done on the bases of number of hash-functions used at each end, i.e., User, Authentication Centre (AC) and Remote Server. Table III shows that our proposed scheme uses less hash functions than Xie and Chen scheme [10].

TABLE III. Number of Hash Functions Used at Each End

Different Ends	*Our Proposed Scheme*	*Xie and Chen Scheme*
User	8	9
Server	5	8
AC	8	10

VI. Conclusion And Future Work

The paper proves that the proposed scheme is better than Xie and Chen scheme in terms of security and performance - both the schemes do not use encryption, verification tables and timestamp for authentication. However, Xie and Chen scheme is not secure against Insider Attack and Forward Secrecy Attack, whereas the proposed scheme improves upon the Xie and Chen scheme exactly in these terms i.e. is secure against Insider and Forward Secrecy Attack. In addition, the proposed scheme uses lesser no of hash function during authentication as compared to Xie and Chen

scheme. We may further enhance efficiency by reducing hash-functions.

REFERENCES

[1] Eun-Jun Yoon, Kee-Young Yoo*, "Robust Multi-Server Authentication Scheme", IEEE sixth IFIP International Conference on Network and Parallel Computing(2009).

[2] H. Zhu, T. Liu, J. Liu, "Robust and Simple multi-server authentication protocol without verification table", Ninth International Conference on Hybrid Intelligent Systems, 2009.

[3] I. C. Lin, M. S. Hwang, and L. H. Li, "A new remote user authentication scheme for multi-server architecture", Future Generation Computer System January, 19:13–22, 2003.

[4] J. H. Lee, D. H. Lee, "Efficient and Secure Remote Authenticated Key Agreement Scheme for Multi-server Using Mobile Equipment", Proceedings of International Conference on Consumer Electronics, pp. 1-2, January 2008.

[5] J. L. Tsai, "Efficient multi-server authentication scheme based on one-way hash function without verification table", Computers & Security, 27(2008):115-121, 2008.

[6] Li LH, Lin IC, Hwang MS, "A remote password authentication scheme for multi –server architecture using neural network", IEEE Transactions on Neural Network 12(06)(2001)1498-1504.

[7] Lamport L., "Password Authentication with insecure communication". Communication of the ACM November 24 (11) (1981)770-772.

[8] M.L. Das, A. Saxena, and V.P. Gulati, "A dynamic ID-based remote user authentication scheme", IEEE Transactions on Consumer Electronics, vol. 50, pp. 629-631, 2004.

[9] MH Shao, YC Chin, "A Novel Dynamic ID-based Remote User Authentication and Access Control Scheme for Multi-Server Environment", 10th IEEE International Conference on Computer and information Technology (CIT 2010), 2010.

[10] Qi Xie and Deren Chen, "Hash function and smart card based multi-server authentication protocol", IEEE WASE International Conference on Information Engineering, 2010.

[11] Sun HM, "An efficient remote use authentication scheme using smart cards", IEEE Transactions on Consumer Electronics 46(4) (2000)958-961.

[12] TY Chen, MS Hwang*, CC Lee, JK Jan, "Cryptanalysis of a secure dynamic ID based remote user authentication scheme for multi-server environment", Fourth International Conference on Innovative Computing, Information and Control, 2009.

[13] W. C. Ku, "Weaknesses and drawbacks of a password authentication scheme using neural networks for multi-server architecture", IEEE Transactions on Neural Networks, 2005.

[14] WS Juang, "Efficient multi-server password authenticated key agreement using smart cards", IEEE Transactions on Consumer Electronics 50(1)(2004)251-255.

[15] Y. P. Liao, S. S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment", Computer Standards & Interfaces, 31(1): 24-29, 2009.

[16] Y. L. Chen, C. H. Huang, J. S. Chou, "A novel multi-server authentication protocol", http://eprint.iacr.org/2009/176, 2009.

[17] H. Mittal, S. Porwal, "A Remote Authentication Methodology For Secure Communication In Distributed Network", International Journal of Research & Development in Technology and Management Science –Kailash, Volume 21, Issue 1, March 2014 ISBN: 978-1-63102-445-0.