A REMOTE AUTHENTICATION METHODOLOGY FOR SECURE COMMUNICATION IN DISTRIBUTED NETWORK

By

Himanshu Mittal | Jaypee Institute of Information Technology
And
Shardha Porwal | Jaypee Institute of Information Technology

ABSTRACT
To make a secure transmission of information in the distributed environment, a methodology based on Diffie-Hellman and hash function is proposed. The new methodology uses no verification table, no encryption techniques, no timestamps but still authenticate and generate session key securely.

KEYWORDS: MULTI-SERVER, AUTHENTICATION SCHEME, DIFFIE-HELLMAN, HASH-FUNCTION

I. INTRODUCTION
With the rush of services and information provided on the Internet in present times, it has become extremely essential to authenticate the flow of information to appropriate recipients. A multi–server user authentication scheme is mechanism which is used by a set of multiple servers to authenticate the user before he/she is allowed to access the services of the respective set of multiple servers. Generally there are three kinds of participants in a multi-server user authentication system: users, a group of servers, and the authentication center. In these multi-
server schemes, the remote user registers with the authentication center only once and can obtain services from multiple servers without repeating registration to every single server [11]. A considerable number of authentication schemes have been proposed till now.

The multi-server authentication methodology must satisfy following security features: (1) No Verification Table (2) Freely Chosen Password (3) Mutual Authentication (4) Low computation and communication cost (5) Single registration (6) Session key agreement (7) User anonymity (8) Access Control (9) Security (10) Session Key Security (11) Known Key Security (12) Forward Secrecy[14]. Also it should be successful to negate the following security attack on a given multi–server environment such as insider attack, impersonation attack, replay attack, password guessing attack, stolen–verifier attack and server spoofing attack[13]. To support these features, we propose “A Remote Authentication Methodology for Secure Communication in Distributed Network”

The structure of the paper is as follows: section 2 explains the proposal; the security analysis is discussed in section 3; next section gives the conclusion and last section includes the future work.

II. METHODOLOGY

System Framework: An Overview of the entire system framework is presented in Figure 2.1. The whole methodology is divided into four phases: the first phase is server registration, then User Registration phase, after which comes the Authentication of Remote User and Server phase and final phase is Mutual Authentication and Session Key Generation. In the first phase, each server sends its ID and registers itself with the authentication center. The user who wants to obtain services from a registered server, register himself/herself with an Id and password at authentication center in the user registration phase. In authentication of remote user and server phase, the user logs in at authentication center. The authentication center verifies the authenticity of the user and the remote server and visa-versa. If the authentication is successful,
then last phase execute where the authentication center allows the user to communicate with the remote server directly and authenticate each other and generate session key. The computed session key is used for data communication over an insecure channel. The completed task flow is elaborated in the following sections.

**Description of Proposed Methodology**

The proposed methodology needs „s” servers, „n” remote users and an authentication centre(AC). At the beginning, AC randomly chooses two secret numbers „x” and „y”. When the user decides to login a server, he/she first registers at the authentication center. The proposed methodology consists of 4 phases:

1. Server Registration,
2. User Registration,
3. Authentication of Remote User and Server,
4. Mutual Authentication and Session Key Generation.

As the proposed methodology is using Diffie-Hellman, „p” and „g” are public, where „p” is a large prime number and „g” is the generator of order p-1 in the group <Zp*,x>. The details for each phase are as follow:

**I. Server Registration Phase:**

In this phase, the server Sj registers at AC by sending its ID, as shown in the figure 3.2.

- □ Sj □ AC: SID

In this, the server sends its identity „SID” to the AC by a secure channel.

- □ AC □ S: h(SID||y)

AC sends back this to server Sj without storing it through a secure channel.
II. User Registration Phase:
The user sends his „ID” and XORed password, Pw K” to AC through a secure channel to register at AC, as shown in the figure 3.3. AC performs following computation on them.

$\text{AC: ID, (Pw K)}$

On receiving the „ID” and XORed password „Pw K”, AC computes:

$\text{Ru} = h(ID||x)$ and

$\text{C0} = Ru \cdot h(Pw K)$.

$\text{AC U: C0}$

AC sends C0 back to user through a secure channel which user stores on its user terminal.

III. Authentication of Remote User and Server:
In this phase, AC allows only the registered user to login and access a registered server. AC generates a mutual session key that helps in the generation of session key. The steps are executed as follows,

A.1 $\text{U AC: ID*, C1; U Sj: ID*}$

Whenever the user wants to access a server, registered on AC, the user enters his/her registered ID and password and the target server Id „SID*” with which user desires to communicate. The user terminal computes Ru
After retrieving Ru, the user randomly computes:
\[ C_1 = (g^a) \pmod{p} \]
On computation of \( C_1 \), user sends \( ID^* \), \( C_1 \), to AC and \( ID^* \) to target server \( S_j \).

**A.2 S_j \rightarrow AC: SID^*, C_3**
On receiving user request, in the form of \( ID^* \), the server „Sj“ randomly compute:
\[ C_3 = (g^b) \pmod{p} \]
After these computation, \( S_j \) sends \( SID^* \), \( C_3 \), to the AC over the public network.

**A.3 AC \rightarrow U: C_5, C_6; AC \rightarrow S_j: C_7, C_8**
On receiving messages in step A.1 and A.2, the AC perform following computation to authenticate user and server as follows:
\[ h(h(ID^*||x)||SID^*||C_1) = C_2? \]
If equal, then AC authenticates the user and the server and proceed with further communication. If either of them is not equal, AC terminates the session.
After authenticating, AC chooses randomly \( c \) \( \in \mathbb{Z}_p^* \) and \( d \) \( \in \mathbb{Z}_p^* \) and computes:
\[ C_5 = (g^c) \pmod{p} \]
\[ C_7 = (g^d) \pmod{p} \]
Then, AC communicates \( C_5 \) to the user and \( C_7 \) to the server „Sj“ on the public network.

**A.4 U \rightarrow AC: C_9; S_j \rightarrow AC: C_{10}**
On receiving the messages from AC, the user computes \( K_1 \) as,
After computing \( K_1 \), user verifies received \( C_6 \) as follows,
\[ h(K_1||SID^*)=C_6? \]
If it equals, the user authenticates AC and if not, the user ends communication with AC.
After the authentication of the AC, the user computes \( C_9 \) and send it to AC.
\[ C_9 = h(K_1+1) \]
Similarly the target server „Sj“ computes \( K_2 \):
After computing \( K_2 \), server verifies received \( C_8 \) as follows,
If it equals, the server authenticates the AC and if not, then server ends communication with AC.
On the authentication of AC, the server also computes \( C_{10} \):
C10 = h(K2+1)

After completion of above operation, server sends C10 and asks for C11 from AC.

A.5 AC □ U: C11; AC □ Sj: C11

When AC receives C9 and C10, it verifies and ensures that the user and server is secure for transmitting mutual session key and computes:

h(K1+1)=C9?
h(K2+1)=C10?

If they both are equal, AC ensures authenticity.

III. Security Analysis

The analysis of the scheme shows that the scheme can easily resist Password Guessing, Replay Attack, Impersonation, Insider Attack, Stolen-Verifier, Man-in-the-Middle attack, Server Spoofing, Authentication Spoofing, Forward Secrecy. Security of session key is also maintained.

IV. CONCLUSION

In this, we proposed a new efficient and secure multi-server authentication using one-way hash function, XOR function and Diffie-Hellman. The proposal does not maintain any verification table, allow freely chosen password, low computation and communication cost, no Encryption/decryption operation, mutual authentication and session key agreement, access control and provide security against every possible attack. It does not authenticate invalid user or invalid server before computing the session key.

V. FUTURE WORK

Since the methodology is one mainly designed for smart-card, we can even allow mobile phones, smartcards, PDA, etc. to act as a client. The future work also includes enhancing the efficiency of the methodology. In addition, we can also look for the proposal of a multi-server multi-authentication center authentication schema.
REFERENCES