A Robust Authentication Protocol for Multi-Server Architecture without Smart Cards

Han-Cheng Hsiang

Department of Information Management Vanung University, Taoyuan, Taiwan

Abstract

With rapid growth of Internet technologies, more and more servers provide different resources to be accessed over the open network. For most of the resources provided by remote servers, users must pass an authentication procedure in order to access data. But most conventional password authentication schemes are designed for the single-server environments and are not satisfied for users' requests. This paper proposes an efficient protocol for multi-server architecture with more security procedure. The computation cost, security, and efficiency of our scheme are well suited to the practical applications.

Keywords: Authentication, Multi-server, Key agreement, System security

1. Intrdouction

In recent years, since the rapid growth of Internet technologies, more and more servers provide different resources to be accessed over the open network. For most of the resources provided by remote servers, users must pass an authentication procedure in order to access data and receive authorization. Remote user authentication scheme allows a server to check the legitimacy of a remote user through insecure communication channel.

In 1981, Lamport proposed a remote user authentication scheme [3] based on verifier table, but this scheme is vulnerable to stolen verifier attack. Since then many password authentication schemes using smart card have been proposed to improve the cost, efficiency, and security of the authentication mechanism [1]. However, these schemes are designed for the single-server architecture. They may not satisfy the users' requests if conventional password authentication methods are applied to multi-server environments. In practice, each user needs to login various remote servers repetitively and also need to remember different identifications and passwords for accessing different servers.

Later, several papers have been devoted to the study of accessing the resources of multi-server environments [2, 5, 6].

Recently, Lee et al. have proposed a novel authentication protocol for multi-server architecture without smart cards [4]. Their protocol is novel and attempts to provide an efficient and secure password authentication protocol without smart cards that can resist all kinds of malicious attacks. Unfortunately, we find that Lee et al.'s scheme is vulnerable to an insider's attack and a stolen verifier attack. To remedy these flaws, this paper proposes an efficient improvement over Lee et al.'s scheme that inherits their merits and with more security.

The rest of this paper is organized as follows. In section 2 shows the details of the proposed scheme. Section 3 makes the security analysis of the proposed scheme. Finally, some concluding remarks are made in the last section.

2. The Proposed scheme

In this section, we propose a robust and secure authentication scheme for multi-server environment. The notations used in our scheme are summarized in Table 1.

Table 1 notations

Notation	Meaning
V_j	the secret key shared between S_j and *RC*
$h(\ )$	the collision-resistant one-way hash function
x	the secret key maintained of registration center
p	a large and published prime
g	the public system parameter, which is the primitive element in $GF(p)$
$E_k(m)$	the encryption function of the message m with the encryption key k
$D_k(m)$	the decryption function of the message m with the decryption key k
SK	the session key shared between U_i and S_j for this protocol run
||	string concatenation operation
$\Rightarrow$	a secure channel.
$\rightarrow$	a common channel.

The multi-server environment contains three participants, the user (U_i), the server (S_j) and the registration center (*RC*). First of all, it is assumed that *RC* is trustworthy. After each server is authorized, *RC* sends each of them a shared secret key $V_j = h(x, SID_j)$.

When the registration center RC permits the entry of a remote server Sj, RC uses SIDj to compute the shared secret key Vj= h(x, SIDj), and sends Vj to Sj via the secure channel. This shared key is used to confirm the legitimacy of the remote server and the registration center. Our scheme consists of four phases: Registration Phase, Login Phase and Authentication Phase. Different phases of work are described as follows:

Registration Phase:

This phase is invoked whenever a new user U_i wants to access the resources of the remote servers, he has to submit his identity ID_i and password PW_i to the registration center *RC* through a secure channel for registration. The details are shown as the following steps.

Step R1: U_i freely selects a password PW_i and a random number r.

Step R2: $U_i \Rightarrow RC$: ID_i, $h(r \oplus PW_i)$.

Step R3: *RC* computes $TPW_i = h(ID_i || x) \oplus h(r \oplus PW_i)$, then stores it in its database.

Step R4: $RC \Rightarrow U_i$: an accepted message.

Login Phase:

When U_i wants to login the remote server S_j, he keys his identity ID_i, password PW_i, r and then performs the following steps:

Step L1: U_i computes $E_{PW_i}(g^a \bmod p)$, where a is a random number.

Step L2: Generate nonce N_i, where the nonce N_i used only once.

Step L3: $U_i \rightarrow S_j$: ID_i, N_i, $E_{h(r \oplus PW_i)}(g^a \bmod p)$.

Authentication Phase:

Upon receiving the login request message $\{ID_i, N_i, E_{h(r \oplus PW_i)}(g^a \bmod p)\}$, S_j and *RC* perform the following procedure to make them authorized.

Step V1: $S_j \rightarrow RC$: ID_i, SID_j , $E_{h(r \oplus PW_i)}(g^a \bmod p)$, $E_{V_j}(N_j, g^b \bmod p, h(E_{h(r \oplus PW_i)}(g^a \bmod p)))$.

S_j generates a nonce N_j and computes $h(E_{h(r \oplus PW_i)}(g^a \bmod p))$ and $E_{V_j}(N_j, g^b \bmod p, h(E_{h(r \oplus PW_i)}(g^a \bmod p)))$, where b is a random number. Next, send ID_i, SID_j, $E_{h(r \oplus PW_i)}(g^a \bmod p)$, $E_{V_j}(N_j, g^b \bmod p, h(E_{h(r \oplus PW_i)}(g^a \bmod p)))$ to *RC*.

Step V2: $RC \rightarrow S_j$: ID_i, SID_j , $E_{V_j}(N_j, g^{as} \bmod p, E_{h(r \oplus PW_i)}(g^{bs} \bmod p))$.

Upon receiving the message sent by S_j, RC computes $D_{V_j}(E_{V_j}(N_j, g^b \bmod p, h(E_{h(r \oplus PW_i)}(g^a \bmod p))))$ to decrypt $Q = h(E_{h(r \oplus PW_i)}(g^a \bmod p))$ by using the secret key V_j shared with S_j. RC computes $Q' = h(E_{h(r \oplus PW_i)}(g^a \bmod p))$ with the received message $E_{h(r \oplus PW_i)}(g^a \bmod p)$ and then compares it with Q. If they are not equal, the connection is terminated; otherwise, RC retrieves $h(r \oplus PW_i) = TPW_i \oplus h(ID_i \| x)$ to compute $D_{h(r \oplus PW_i)}(E_{h(r \oplus PW_i)}(g^a \bmod p))$, and computes $g^{as} \bmod p$ and $g^{bs} \bmod p$, where s is a random number. Finally, compute $E_{V_j}(N_j, g^{as} \bmod p, E_{h(r \oplus PW_i)}(g^{bs} \bmod p))$ and then send the computational result to S_j along with ID_i and SID_j.

Step V3: $S_j \rightarrow U_i$: ID_i, SID_j, $E_{SK}(N_{ij})$, $E_{h(r \oplus PW_i)}(g^{bs} \bmod p)$.

Upon receiving the message, S_j computes $D_{V_j}(E_{V_j}(N_j, g^{as} \bmod p, E_{h(r \oplus PW_i)}(g^{bs} \bmod p)))$ to retrieve N_j. Then check if N_j is in the decryption result for freshness checking. If it holds, S_j computes the session key $SK = (g^{as})^b \bmod p$ for this scheme run. It then generates the nonce N_{ij} and computes $E_{SK}(N_{ij})$. Next, send the message including ID_i, SID_j, $E_{SK}(N_{ij})$, $E_{h(r \oplus PW_i)}(g^{bs} \bmod p)$ to U_i. Otherwise, the connection is interrupted.

Step V4: $U_i \rightarrow S_j$: $E_{SK}(h(N_{ij}))$.

After U_i receives the message sent by S_j, it computes $D_{h(r \oplus PW_i)}(E_{h(r \oplus PW_i)}(g^{bs} \bmod p))$ to retrieve $g^{bs} \bmod p$. Then it computes the session $SK = (g^{bs})^a \bmod p$ and uses SK to decrypt N_{ij}. Next, U_i computes $E_{SK}(h(N_{ij}))$ and sends it to S_j.

Step V5: $S_j \rightarrow U_i$: $E_{SK}(N_i)$.

Upon receiving the message from U_i, S_j computes $D_{SK}(E_{SK}(h(N_{ij})))$ to retrieve $Q = h(N_{ij})$. Then, S_j computes $Q' = h(N_{ij})$ by using N_{ij} generated in Step V3 and compares it with Q. If they are not equal, S_j terminates this session; otherwise, S_j computes $E_{SK}(N_i)$ and sends the computation result to U_i.

Step V6: After getting the transmitted message, U_i computes $D_{SK}(E_{SK}(N_i))$ and checks if N_i is in the decryption result for freshness checking. If it holds, the authentication is successful; otherwise, the connection is interrupted. After finishing mutual authentication, the user U_i and the remote server S_j can use the session key SK to encrypt/decrypt the secret information for the following communication.

3. Security analysis

In this section, we will discuss the security of our proposed scheme. Other parts of our work are same as the original Lee et al.'s scheme [5].

Claim 1. The proposed scheme can resist the insider's attack.

Proof: In the registration phase of Lee et al.'s scheme, a user U_i selects a random number r, password PW_i and computes $h(r \oplus PW_i)$. He submits ID_i and $h(r \oplus PW_i)$ to the registration center RC. If the insider of RC may try to use PW_i to impersonate U_i to login other servers outside of the system, he will fail. Since U_i registers to RC by presenting $h(r \oplus PW_i)$ instead of PW_i, the insider of RC cannot directly obtain PW_i. Moreover, as r is not revealed to RC, the insider of RC cannot obtain PW_i by performing an off-line guessing attack on $h(r \oplus PW_i)$. Hence, the improved scheme can resist the insider attack.

Claim 2. The proposed scheme can resist the stolen-verifier attack.

Proof: In the proposed scheme, the user U_i's authentication data stored in RC is $TPW_i = h(ID_i || x) \oplus h(r \oplus PW_i)$. Suppose that an adversary Eve has stolen the TPW_i, she can obtain $h(r \oplus PW_i)$ only if Eve has the information of $h(ID_i || x)$, which implies she knows RC's long-term secret key x. Since $h(r \oplus PW_i)$ is hidden in TPW_i, and the secret key x is under strict protection as assumed, it is infeasible for Eve to obtain $h(r \oplus PW_i)$ in this way. In addition, if Eve is a legal user and has stolen her TPW_e, it is still computational infeasible for Eve to retrieve x since $h(.)$ is a collision-resistant one-way hash function. That is, our proposed scheme can resist the stolen-verifier attack.

Claim 3. The proposed scheme can resist the server spoofing attack and registration center spoofing.

Proof: If Eve is a legal user, she cannot impersonate as any remote server S_j to cheat U_i, since she cannot construct the session key SK without the knowledge of PW_i, r. Even if Eve has stolen the TPW_i, she cannot obtain $h(r \oplus PW_i)$ as mentioned above. Thus, Eve cannot decrypt the transmitted messages from some legal user. After communicating with the masqueraded remote server, the legal user can detect immediately and terminates the session. Hence, our improved scheme can protect the user from being cheated by the masqueraded remote server.

Similarly, if Eve wishes to masquerade as the registration center to cheat the server, it is infeasible because each server S_j has a $V_j=h(SID_j, x)$. The server can use V_j and a nonce N_j to verify the registration center in Step V3 of the Authentication Phase.

Claim 4. The proposed scheme can resist the off-line password guessing attack.

Proof: Suppose an attacker Eve has the information of $\{h(r\oplus PW_i)\oplus rc, h(h(r \oplus PW_i))\oplus rs$ and $h(SK, rc)\}$. Eve first can guess a password PW_E to compute the corresponding $h(PW_E)$ and then finds ru = $ru\oplus h(r\oplus PW_i)\oplus h(PW_E)$ and $rs =rs\oplus h(r \oplus PW_i)\oplus h(PW_E)$. However, it is computationally infeasible, since Eve does not know ru, rs and SK. In addition, $h(.)$ is a collision-resistant one-way hash function. Hence, even if Eve has guessed the correct password, she cannot verify her guess by analyzing the scheme messages over the network. Obviously, off-line guessing attacks cannot be performed on our proposed scheme.

4. Conclusions

We have shown that our proposed scheme can withstand various attacks. As compared to the Lee et al.'s scheme, the proposed scheme inherits their merits, enhances their security. Therefore, the proposed scheme is well suited to the practical applications environment.

5. References

[1] W.S. Juang, Efficient password authenticated key agreement using smart cards, Computers and Security, vol. 23, pp. 167-173, 2004.

[2] W.S. Juang, Efficient multi-server password authenticated key agreement using smart cards, IEEE Transactions on Consumer Electronics, vol. 50, no. 1, pp. 251-255, 2004.

[3] L. Lamport, Password authentication with insecure communication, Communications of ACM, vol. 24, pp. 770-772, 1981.

[4] J.S. Lee, Y.F. Chang and C.C. Chang, A novel authentication protocol for multi-server architecture without smart cards, International Journal of Innovative Computing Information and Control, vol. 4, no. 6, pp. 1357-1364, 2008.

[5] I.C. Lin, M.S. Hwang and L.H. Li, A new remote user authentication scheme for multi-server architecture, Future Generation Computer Systems, vol. 19, pp. 13-22, 2003.

[6] W.J. Tsuar, An enhanced user authentication scheme for multi-server internet services, Applied Mathematics and Computation, vol. 170, pp. 258–266, 2005.