An Optimal Strong Password Authentication Protocol with USB Sticks

D.Vikram , M.E, Student
Department of Computer Science & Engineering
Dr.Mahalingam College of Engineering & Technology
Pollachi, India -642003
Email: dvikram91@gmail.com

Abstract

Authentication is the process for identify the correct user or not. The identities enclose mainly the username and passwords for verifying the two entities. The authentication information's are stored in the form of encryption in a device which is properly registered in the server. At the time of authentication process performs between user and server the intruder can eves-dropping the communication channel and login into the system by an authorized user. To overcome this optimal strong password authentication (OSPA) protocol uses the multiple hash operation the time of authentication for the users. The server chooses the hash function only at the time of user requests for the login process. So the intruder cannot know the information which transferred at the time of authentication process.

The OSPA can improve the authentication process for obtaining mutual communication between user and server. The authentication information will not be known to the intruder. So the multiple hash operation obtains the secure authentication information. The OSPA protect information of the user & server and protect from the guessing attack. The guessing attack prevention performs by the server using the multiple hash operation and USB Stick. Since the intruder cannot perform the guessing attack on the authentication information without knowing the proper hash operations.

Keywords : User Authentication, Multiple Hash Function, USB sticks

INTRODUCTION

Authentication is the process of determining whether someone or something is, in fact, who or what it's declared to be. In private and public computer networks (including the Internet), authentication is commonly done through the use of log on passwords. Knowledge of the password is assumed to guarantee that the user is authentic. Each user registers initially (or is registered by someone else), using an assigned or self-declared password. On each subsequent use, the user must know and use the previously declared password. The weakness in this system for transactions those are significant (such as the exchange of money) that passwords can often be stolen, accidentally revealed, or forgotten. For this reason, Internet business and many other transactions require a more stringent authentication process [11]. The use of digital certificates issued and verified by a Certificate Authority (CA) as part of a public key infrastructure is considered likely to become the standard way to perform authentication on the Internet. The password based authentication mechanism is the simple and more convenient schemes, to user and server to verify the identity parties. Lamport [1] presented the Password Authentication PA scheme over an insecure channel in 1981.

The password authentication mechanism has three class classifications are: the password-only PA protocols, the dedicated device-aided protocols and memory device-aided protocols.

The Password-only PA protocol, has no extra devices are used, the user only know the password than can be memorize by human easily and server maintains the password file to verify the user authentication request. While the password file maintained is more difficult.

In order to reduce the risk in dedicated devices-aided PA protocol has been used where the user remember the short password to hold a special devices (e.g. Smart cards) to complete successful authentication. The authentication information which stored in the smart card is only known to the server and user. In smart cards the content can be leaked by monitoring its power consumption.

To overcome the problem in memory devices-aided PA protocols come into existence where user can store the information into devices or driver for authentication purpose issued by server. This scheme called as PA without using smart cards [5].

The rest of the paper is organized as follow. Section 2 Existing System has been discussed. In section 3 Proposed System has been discussed. In section 4 security and efficiency analysis. In section 4 the result can be explained. In section 6 we gave the conclusion.

2. EXISTING SYSTEM

For authentication purpose, an improved authentication protocol which is used by the remote user using USB sticks was implemented. The authentication protocol maintains the connectivity information which is used for authentication between user and server. Remote user being at any location can access the information provided by the admin which is stored in the server with the help of USB sticks. These USB stick have a standard manufacturing id, these id's should be registered prior in the server. Server allows the user to login only when the USB manufacturing id is valid [10]. These USB sticks contain the values of the private and public keys, which improves the efficiency of the authentication process of the protocol.

During login process remote user uses the information's which are stored in the USB sticks, but the server uses only the information which is sent by the user at the time of login process and it does not needs the information which is stored in the USB sticks [3]. Both the user and server process should be held within the given timestamp value with this same timestamp value authentication process is improved. While using Mod operation in the process, there occurs a less chance of getting the original value. The values in the USB sticks are obtained by performing XOR operation, so that the values are not easily identified by the Attacker. This protocol is implemented in the following five phases: Initialization Phase, Registration, Login, Authentication and Password Change.

2.1 Description

The notations used on process are as follows:
ID_i: the user's identity;
PW_i: the user's password;
x,X: the server S's secrete key and public key;
p,q: two large prime numbers where p=2*q+1;
g: a generator with q order in GF(p);
H: a secure one way hash functions;
T: timestamp;
ΔT: maximum transmission delay;
Z_q: ring of integer modulo q;
Z_q^*: multiplicative group of Z_q;
||: the concatenation operation;
n: an integer which indicates times of authentication sessions.

2.2 Initialization Phase

The user request the server for their registration process by using the user identity ID, password PW where the server start the initialization process for user to make the authenticated user.

The server processing on the following steps:

1. Q is the prime number and generates the P=2Q+1 where P is also prime number,
2. Select the generator value $g=Z_q^*$
3. Select the secrete key value $x \in Z_q^*$,
4. Secure one way hash function operation H, Compute public key $X= g^x \bmod p$.

2.3 Registration Phase

The server performs the registration process only after the initialization has been finished the registration process performed by using the values generated at the time of initialization. The values on the initialization like P, Q, X, x are used for registration.

Server performs the registration process for the user:

1. User U selects their user identity ID, password PWD and send to server S.
2. Server receives and performs the compute y_i= H (ID_i ||x)⊕H(PW_i). Then S send the authentication information to user {X,Y_i,H,P,Q} and S store the user identity ID in the ID table.
3. After that user receives the authentication information from server stores it on XML file format in USB sticks.

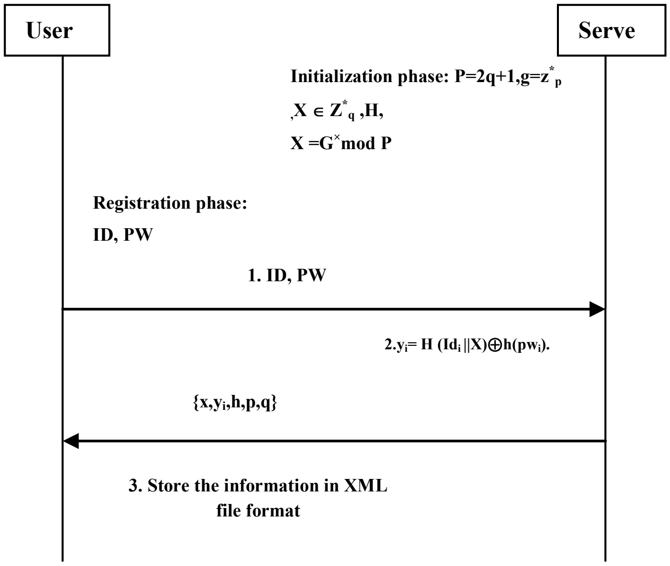
User
Serve
Initialization phase: P=2q+1,g=z^*_p
,X ∈ Z^*_q ,H,
X =G^xmod P
Registration phase:
ID, PW
1. ID, PW
2.y_i= H (Id_i ||X)⊕h(pw_i).
{x,y_i,h,p,q}
3. Store the information in XML file format

Fig. 1: Initialization and Registration Phase

2.4 Login and Authentication Phase

After successful registration the user login into the server by using the user identity and password with help of the XML file. The login and authentication process are as follows:

1. The user first retrieves the xml file from the hard disk drive and login into the server by using UID, PWD.
2. User U choose the random number $\alpha \in Z^*_p$, compute $y_i'' = y_i \oplus H(PW_i)$, $C_i = g^a \bmod p$,$D_i = x^a \bmod p$, $V_i = H(ID||y_i''||C_i||D_i||T_1)$ were T_1, and U send login request are $\{ID, C_i, V_i, T_1\}$ to server S.

3. The serve receive the login request of user and check the valid login ID from the ID table and $(T_2 - T_1) < \Delta T$ where T_2 current time of S , ΔT is the maximum time limit allocated by server . if the both condition satisfied S perform the login operation as follows:

    a. Then compute $Y_i'' = H(ID || x)$, $D_i' = C_i^x \bmod p = g^{x\alpha} \bmod p = X^{\alpha} \bmod p = D_i$ compare Vi with $H(ID_i||Y_i'' || C_i || D_i || T_1)$if the both Vi= Vi' then compute $M_i = H(ID_i ||D_i'||T_3)$ where T_3 is current time of S otherwise drop the U request and send $\{M_i,T_3\}$to U.

    b. User U receive the $\{M_i,T_3\}$ and check the T_3time process is valid or not and$M_i = H(ID||D_i||T_3)$ if it's satisify then S authenticated U otherwise S not authenticated.

    c. After this mutual authentication has finish then U and S compute the symmetric session key sk=$H(D_i)$user-side = $H(D_i')$ server-side for communication channel.

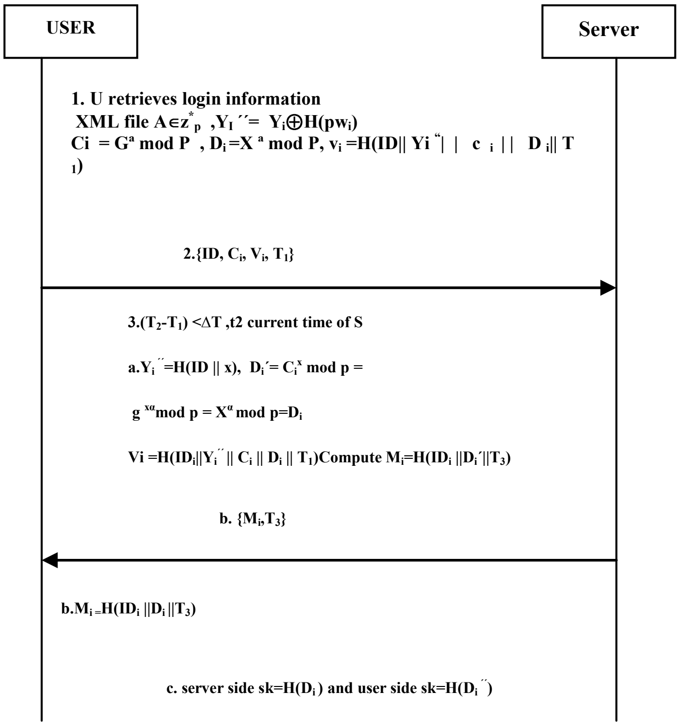
USER
Server
1. U retrieves login information
XML file $A \in z^*_p$,$Y_I'' = Y_i \oplus H(pw_i)$
$Ci = G^a \bmod P$, $D_i = X^a \bmod P$, $v_i = H(ID|| Yi''| | c_i || D_i|| T_1)$
2.$\{ID, C_i, V_i, T_1\}$
3.$(T_2 - T_1) < \Delta T$,t2 current time of S
a.$Y_i'' = H(ID || x)$, $D_i' = C_i^x \bmod p =$
$g^{x\alpha} \bmod p = X^{\alpha} \bmod p = D_i$
$Vi = H(ID_i||Y_i'' || C_i || D_i || T_1)$Compute $M_i = H(ID_i ||D_i'||T_3)$
b. $\{M_i,T_3\}$
b.$M_i = H(ID_i ||D_i||T_3)$
c. server side sk=$H(D_i)$ and user side sk=$H(D_i'')$

Fig. 2: Login and Authentication Phase

2.5 Password Change Activity

The user U can change the password freely in the server. The password change can be process after the authentication process completed. Server updates the password change operation on the user information.

The user can choose his/her password by new password PW_i^* and compute $Y_i^* = Y_i \oplus H(PW_i) \oplus H(PW_i^*)$ and replace the Y_i as Y_i^*.

3. PROPOSED SYSTEM

Optimal Strong Password Authentication (OSPA) protocol is also used for authentication process in Smart Card applications. It stores the userid in the smart card which is issued by the server for user authentication which in turn password given by the user should be validated by the server. This type of Authentication process is vulnerable to guessing attack. To overcome this type of attack and to implement this algorithm without smart card, multiple hash functions are used.

OSPA is implemented using USB sticks which are properly registered by the server. Only the primary key values are stored in the USB sticks, these primary key values are generated at the time of user registration and it is also validated at the time of user login. The values in the USB sticks are the encrypted before storing in it. The server validates the manufacturing id of the USB stick to ensure that the stick is registered one or not and verifies the information present in it which conforms the originality of the user [7] [12].

The multiple hash functions are randomly selected by the server only at the time of user requisition for login [2]. Because of using multiple hash functions different values are generated at each time of login, so that the attacker , user and server is unaware about the hashing function which is involved and at the same time guessing attack can be avoided [6] [8].

The term ‘n’ denotes how many times the user login’s to the system, each time when user login the value of ‘n’ is incremented and it stored in the server, this is used to track the user behavior [5] [9].

3.1 Registration phase

The user U wants to acces the resources then he/she first register into the system by using userid, password P.

The user registration are as follows:

1. The user request for registration by using its username and password by $h^2(P \oplus 1)$.

2. The user send the user identity ID and $h^2(P \oplus 1)$ to the server S for their register process.

3. The server S receive the information and store the values $\{ID, h^2(P \oplus 1), n\}$ into the database.

4. Where initial the n value is 1 and X is the secrete key generated by using prime number and mod operation and store X value and manufacture id’s of the USB in the XML file format which is store in the USB Sticks.

3.2 Authentication phase

After the successful registration the user U want to access the resources allocated by the server S then U make the login process by using the USB Sticks.

1. The user U sends the login request to the server S by retrieving the information from the XML file.
2. Then server S sends the sequence number n and hash operation for login operation.
3. The user U receives it and perform the computation process $c_1= h(P \oplus n) \oplus h^2(P \oplus n)$, $c_2= h^2(P \oplus (n+1)) \oplus h(P \oplus n)$, $c_3= h^3(P \oplus (n+1))$.
4. User U send $\{c_1, c_2, c_3\}$ to the server S.

5. Then the server S receive the informationand chck the time procees. If the time is valid and perform the authentication process $h(P \oplus n) = c_1 \oplus h^2(P+n)$, $h^2(P \oplus (n+1)) = c_2 \oplus h(P \oplus n)$ and check $c_3 = h(h^2(P \oplus (n+1)))$. If the process same the user U is an authenticated user.
6. The server S authenticated the user U.
7. Then server Sstore the $h^2(P \oplus (n))$ and replace $h^2(P \oplus (n+1))$ where n=n+1

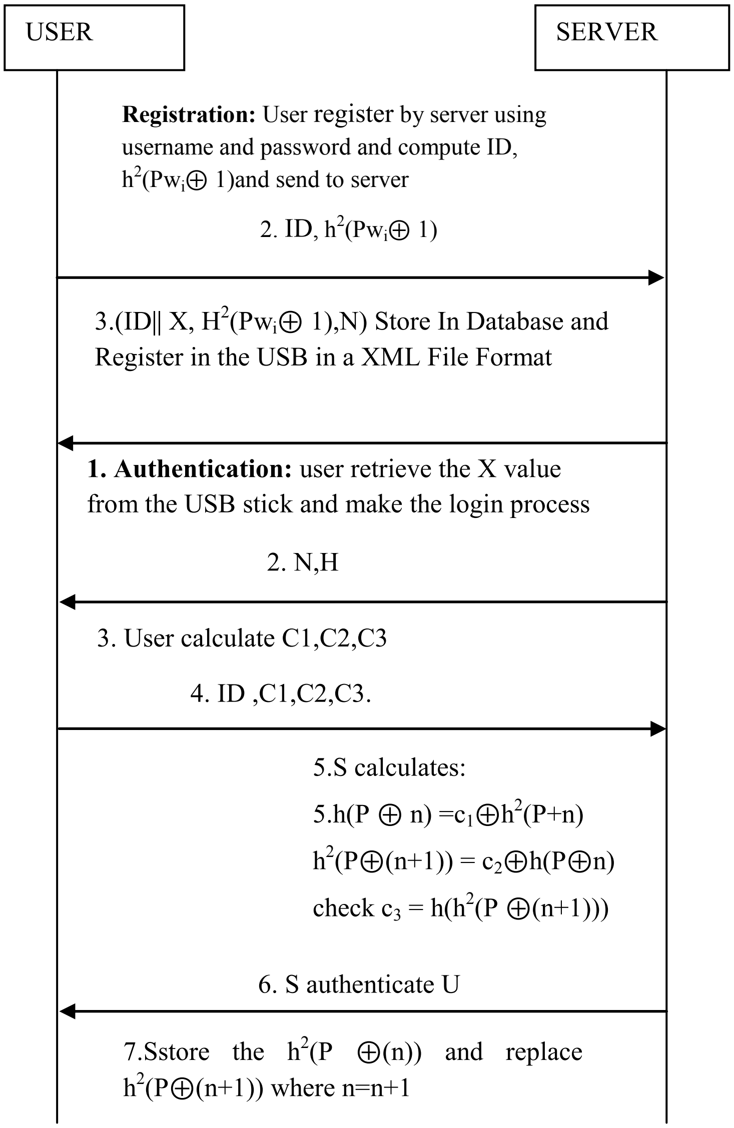
USER
SERVER
Registration: User register by server using username and password and compute ID, $h^2(Pw_i \oplus 1)$and send to server
2. ID, $h^2(Pw_i \oplus 1)$
3.(ID|| X, $H^2(Pw_i \oplus 1)$,N) Store In Database and Register in the USB in a XML File Format
1. Authentication: user retrieve the X value from the USB stick and make the login process
2. N,H
3. User calculate C1,C2,C3
4. ID ,C1,C2,C3.
5.S calculates:
5.$h(P \oplus n) = c_1 \oplus h^2(P+n)$
$h^2(P \oplus (n+1)) = c_2 \oplus h(P \oplus n)$
check $c_3 = h(h^2(P \oplus (n+1)))$
6. S authenticate U
7.Sstore the $h^2(P \oplus (n))$ and replace $h^2(P \oplus (n+1))$ where n=n+1

Fig. 3: Registration and Authentication Phase

4. Guessing attack prevention

The user S authenticate with server S by using the information stored on the XML file.

User make the login request to the server S and S send the n values and the H hash operation where the intruder observer the information and perform the guessing attack on it but the information N, H does not contain any information about the user ID, password Pw.

If the intruder eves dropping the communication channel at the time authentication process between user and server and perform the guessing attack on c1, c2, c3 and make the login into the server. The server can easily identified the intruder by the hash operation where the c_1, c2, c3 is not valid [4].

The server S can choose the hash operation only at the time of user U make the login request. The server S send the N, H and start the time process. If the user send the value c1,c2,c3 the server check the time and perform the compuation othwerwise it discard the user request. If the time process valid the the server comput the compuation process if the intruder does not know the multiple hash operation.

5. RESULT

Authentication is improved because of using multiple hash functions. In Existing system only one hash function is used, so only the same value is passed every time during authentication. In Proposed system multiple hash function is used, so the values keep changing each time of authentication. So, the attacker cannot find the values.

6. CONCLUSION

I have analyzed the authentication process without using smartcards and try to improve the communication process by using the memory sticks (USB) for secure authentication process. The Optimal Strong Password Authentication (OSPA) protocol can be able to perform the secure communication between user and server. This protocol also can be with stand for guessing attack by using the multiple hash operation on it. Compare to existing protocol the OSPA can be able to guarantees for the user to protect from the guessing attack.

References

[1] Leslie Lamport. (1981) ‘Password Authentication with Insecure Communication’. Communications of the ACM, Vol 24, No 11, pp. 770–772.

[2] W. C, Ku. (2004) ‘A hash-based strong-password authentication scheme without using smart cards’. ACM Operating Systems, Review, Vol 38, pp. 29-34.

[3] B.L, Chen. W.C, Kuo. and L.C, Wuu. (2012) ‘A secure password-based remote user authentication scheme without smart cards’, journal of Information Technology and Control, Vol 41, No 1, pp. 53-59.

[4] Cheng-Chi-Lee. Chia-Hsin-Liu. and Min-Shia Hwang, L. (Jan. 2013) ‘Guessing Attacks on Strong-Password Authentication Protocol’, International Journal of Network Security, Vol.15, No.1, pp. 64-67.

[5] Kumar Mangipudi. and Rajendra Katti. (May 2006) ‘A Hash-based Strong Password Authentication Protocol with User Anonymity’, International Journal of Network Security, Vol.2, No.3, pp. 205–209.

[6] C.W, Lin. C.-S, Tsai. and M.S, Hwang. (2006) ‘A New Strong-Password Authentication Scheme Using One-Way Hash Functions’, Journal of Computer and Systems Sciences International, Vol. 45, No. 4, pp. 623–626,.

[7] Hyun Sook Rhee. Jeong Ok Kwon. and Dong Hoon Lee. (2009) ‘A remote user authentication scheme without using smart cards’, Computer Standards & Interfaces Vol 31, pp. 6–13.

[8] Mary Cindy Ah Kioon. Zhao Shun Wang. and Shubra Deb Das. (2013) ‘Security Analysis of MD5 algorithm in Password Storage’, Proceedings of the 2nd International Symposium on Computer, Communication, Control and Automation (ISCCCA-13), pp. 0706-0709.

[9] Chun-Ta Li .(2011) ‘Secure Smart Card Based Password Authentication Scheme With User Anonymity’, Journal Of Information Technology And Control, Vol.40, No.2, pp 157-162l.

[10] Qi Jiang. Jianfeng Ma. Guangsong Li. and Zhuo Ma. (2013) ‘An improved password-based remote user authentication protocol without smart cards’, Journals of Information Control And Technology, Vol.42,No.2, pp150-158.

[11] Margaret Rouse.(2007), ”Authentication”,‘ www://searchsecurity.techtarget.com /definition/Authentication’.

[12] M. S, Hwang. H. C, Wu. and C. H, Liu. (2001) ‘A Secure Strong-Password Authentication Protocol’, Fundamenta Informaticae XXI journal of IOS Press, pp. 1001-1008.