International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Multi-Server Authentication and Key Exchange

Muzammil M. Ahmad

B.N College of Engineering Pusad, India

Abstract: ***Multi-server authentication and key exchange is where the multiple servers contribute in the authentication process of clients, and granting the access to authenticated clients. Present protocols used for authentication purpose keeps the authentication information or secrete codes of all clients on a single server, which is very insecure from the prospect of security. In this paper I have proposed a multi-server authentication and key exchange process where client communicate with the main server by means of intermediate servers, intermediate servers are very useful for checking or verifying authenticity of clients. This protocol is simple and secure against present two server password-only authenticated key exchange protocols.***

Keywords: Authentication, key exchange, personification, identity theft

1. Introduction

We are living in the era of information technology with the computer based society. Near about in each and every movement of our day to day life we need passwords or some private or secrete codes, to prove our authenticity and many times checks the authenticity of others. For example doing transaction on ATM machines [1], accessing emails, handling mobile phones, computers, accounts etc. In all these processes, secrete codes or passwords plays very much important role. We can't imagine all these stuff without having passwords or private secret codes, and this is for the purpose of authentication and security. Servers or service providers by accepting these codes comes to know to whom they providing the services or access, is authentic or not and to resisting the unauthentic penetration into the system or unauthorized access. Previous authentication processes were having one way authentication, where the server authenticating the clients and granting the access to authentic clients, but the problem with these authentication system was one party (client) blindly trusting on the other party (server) there were no way for clients to check whether the server is authentic or not, and hence the attacker many times pretending to be a true server and collecting the critical information of the clients which causes then identity theft. To overcome this issue mutual authentication [5][6] then introduced which allows both parties to mutually authenticate each other i.e. client authenticating the server and server authenticating the clients and both party try to satisfying that they are communicating with the reliable party.

In my system there is a client and three Servers. Server 3rd is the main Server and remaining two Servers are intermediate Servers. Client will split his random codes into two parts, half part sending towards one Server and another half part sending to anther Server and complete hash code sending towards the main Server. Main server receiving the code from client and two servers and verifying the integrity of code, finally the main server will providing the key to client for further communication. Client will check the authenticity of all these Servers and integrity of codes and then using the key. We have used the public key cryptosystem for encrypting the key and messages.

2. Related Work

In 1976 , Diffie and Hellman proposed the shared secret key exchange protocol called as Diffie-Hellman protocol where two parties who knows nothing about each other and wish to communicate over an unprotected communication channel finally they got agree on a particular shared secret key[2] . In 1985 , T.ElGamal [3] proposed his public key protocol called as ElGamal encryption scheme having three main phases namely key generation, encryption and decryption. For distributed system the authentication scheme was invented at MIT known as Kerberos authentication scheme it makes use of trusted third party authentication service that enable clients and servers to establish authenticated communication [4].

Katz et al proposed the two Server password-only authenticated key exchange protocol which was not that much efficient for practical use [1] . Xun Yi, San Ling and Huaxiong Wang , proposed the Efficient Two-Server Password-only Authenticated Key Exchange in 2013 which is good from the prospect of security but it is too much complex as it has lot of calculations and multiple steps for client and two servers during authentication and key exchange phase.[1]

3. Preliminaries

3.1 Kerberos

Kerberos is basically the authentication scheme or protocol where the client gets authenticated by server and it provided some ticket for accessing further servers. The ticket reflecting that the client got authenticated by earlier server or servers and is valid client.

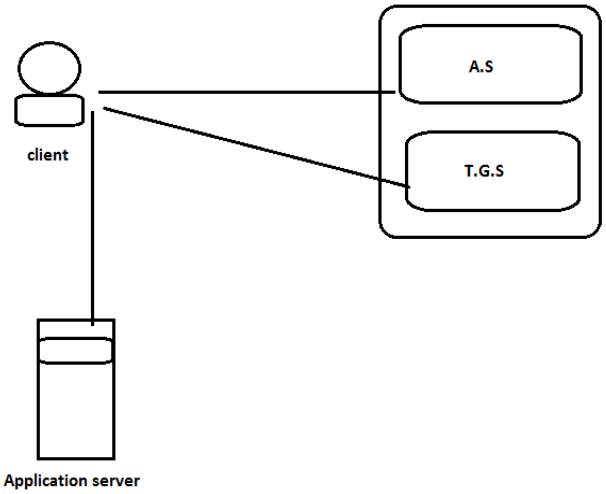

Figure 1: Overview of Kerberos

Above figure shows the generalize overview of Kerberos authentication scheme, client have to register himself to authentication server (A.S), A.S having its database and it checking or verifying the client's identity and providing ticket to go through the ticket granting server (T.G.S), T.G.S checks the authenticity of client and other relevant things like validity of earlier ticket, timestamp etc. and providing the ticket for accessing the application server. The process of encryption and decryption carried out in each phase of Kerberos. Client using the element called the Authenticator which helps the other servers to authenticate the client [4].

3.2 Diffi-Hellman key exchange protocol

Diffie-Hellman protocol was invented in 1976 by Diffie & Hellman, basically use for key exchange purpose. It allows two users to establish secrete shared key.

1) Party A and B agree on cyclic group G with prime order q and generator g.
2) A chooses an integer a randomly from Zq* and calculate$C = g^a$, and B chooses an integer b from Zq* randomly and calculate D= g^b . Now party A & B, exchange the massage C and D
3) Party A computes the secrete session key k1=D^a=g^{ba}, and Party B calculate the secrete session as k2= C^b= g^{ab}

Since k1=k2, hence Party A and Party B agree on the same secret session key

3.3 Elgamal Encryption Scheme

The ElGamal encryption scheme was invented by T.ElGamal in 1985, on the basis of Diffie-Hellman key exchange protocol. It mainly consist of three parts name as Key generation , Encryption and Decryption.

1) On input a security parameter k, it publishes a cyclic group G of large prime order q with a generator g. Then it chooses a decryption key x randomly from Zq* and calculates an encryption key y = g^x
2) On input a message m belongs to G and the encryption key y, it chooses an integer r randomly from Zq* and outputs the ciphertext C = E (m,y) = (A , B) = (g^r, my^r)
3) On inputs a ciphertext (A , B), and the decryption key x , it outputs the plaintext m = D(C , x) = B / A^x

4. Multi-Server Authentication & Key Exchange

C→S1: E(sr1′, PUs1)
C→S2: E(sr1′, PUs2)
S1→S3: E(sr1′, PUs3)
S2→S3: E(sr1′, PUs3)
S3→C: E(K||sr1′||sr2′, PUc)
Where
S1= Server 1
S2= Server 2
C= Client
r = Random number
r′= hash code of random number
h = hash value
sr1′= split hash code of r
sr2′= split hash code of r
PUc= public key of client
PUs1= public key of server1
PUs2= public key of server 2

4.1 Sequential flow of our protocol

1) Client C chooses a password and generate the random number r
2) Perform hash of the random number r which becomes r'
3) Send this hash code (r') to the server 3 &
4) Split the hash code into two equal parts such that sr1′+ sr2′=r′
5) Encrypt the sr1′ using public key of server 1
6) Encrypt the sr2′ using public key of server 2
7) Send these encrypted messages towards server1 and server2 respectively
8) Server1 & server2 decrypt the messages & receives sr1′ & sr2′
9) Server1 & server2 again encrypt the sr1′ & sr2′ using public key of server 3 & send to server 3 respectively
10) Server 3 decrypt the messages and concatenate sr1′ & sr2′ and compare sr1′+ sr2′ & r′
11) After comparison , Server3 establishes a secrete session key K with client along with the sr1′+ sr2′ by using the public key of client
12) Client verify the sr1′+ sr2′ and use the use key.

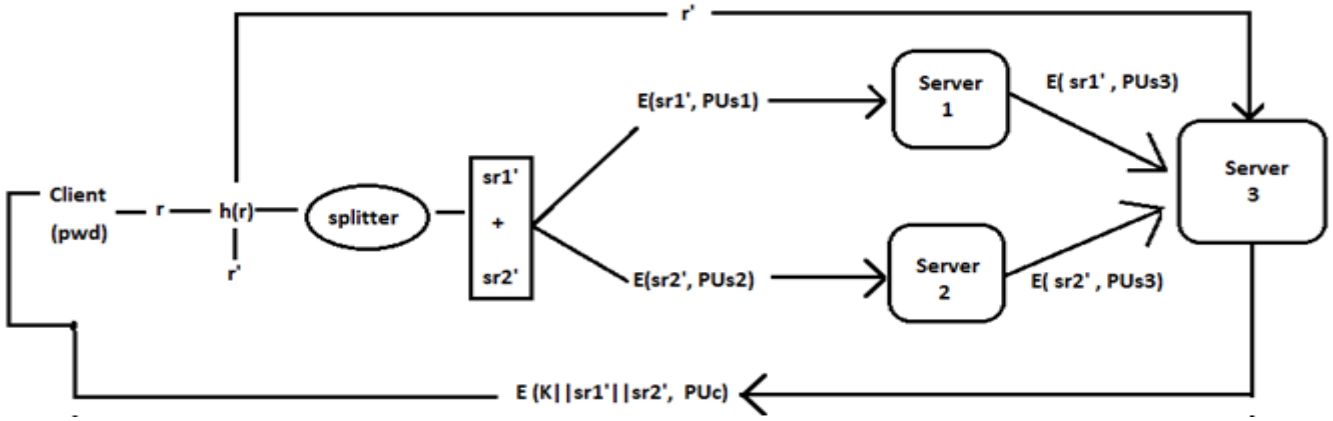

Figure 2: Overview of protocol

With reference to above figure, there are three severs and a client. Client having his password and he wish to communicated with the main server i.e. Server 3, client generate the random number r and prepare the hash value of it and send one copy of that random number to the Server 3, in this setting the client splits the hash code of random number into two equal parts and encrypts these parts with the public key of Server 1 and Server 2 respectively , then send these encrypted hash code send towards the Server1 and Server 2. Server1 and Server2 receives the information, decrypts the encrypted hash code and both the Servers again encrypt it with the public key of Server3 and both send the encrypted massage to the Server 3. Now Server 3 receives the messages, decrypts the massage and performing concatenation operation on the both hash code and compare the concatenated value with r′ which had earlier received from client . After the verification of all these values, the Server 3 send the key along with the hash codes received from the Server 1 & Server 2 by using the public key of client , and hence the client and the main Server (Server3) agree on the particular session key K.

5. Security of Protocol

Theorem 1. *This protocol is secure against any passive and active attack on all four levels i.e. on server1 level, server2 level, server 3 level and client level .*

Proof.
We are using public key cryptography while exchanging the messages or key

5.1 Server 1 and server 2 level security

Client instead of passing his actual password, generate the random number, prepare its hash code and splits the hash code into two parts for S1 & S2 but instead of directly sending the hash code client encrypt the messages with the public keys of S1 & S2 . As the messages are encrypted using the public keys of S1 & S2 so only S1 & S2 can decrypt the massages this is level one and level 2 security.

5.2 Server 3 level security

Server 1 & 2 after decryption of messages again encrypt it by using the public key of S3 and hence only Server 3 can decrypt the message and Server 3 after receiving the data, compare the data of these two Servers with the data of client, hence in case of any insider or outsider attack it can be detected here.

5.3 Client level security

Now client received the session key K from the Server 3, which had encrypted by the public key of client and hence only client can decrypt the data, and along with key it provided the splits hash code of Server 1 and Server2 which earlier had passed by client to both Server1 and Server2 and reflecting or detecting any misbehave with that code in the comparison, hence I proved the multi-level security in this system.

6. Conclusion

In this paper, I proposed a Multi-Server Authentication and Key Exchange protocol. Present protocols for password-only authentication, keeping all the passwords necessary to authenticate the clients on a single server which is not good from the prospect of security. Multi-level security analysis shows that this multi-server protocol is secure, easy and efficient for practical use.

References

[1] Xun Yi, San Ling and Huaxiong Wang , " Efficient Two-Server Password-only Authenticated Key Exchange" IEEE Trans. Parrell and distributed system, VOL 24, NO 9 Sep. 2013.
[2] W. Diffie and M.E. Hellman, "New Directions in Cryptography," IEEE Trans. Information Theory, IT 22, no. 6, pp. 644-654, Nov. 1976.
[3] T. ElGamal, " A public key cryptosytem and a Signature Scheme Based on Discrete Logarithms," IEEE Trans. Information theory, vol. IT-31, no. 4, july 1985
[4] William Stallings. "Cryptography and network security," Pearson 5th edition
[5] M. Abdalla and D. Pointcheval, "Simple Password-Based Encrypted Key Exchange Protocols," Proc. Int'l Conf. Topics in Cryptology (CT-RSA), pp. 191-208, 2005.
[6] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated Key Exchange Secure against Dictionary Attacks," Proc. 19th Int'l Conf. Theory and Application of Cryptographic Techniques (Eurocrypt '00), pp. 139-155, 2000.