Simply Security Login Protocol based on Client's Secure Random Texts

Somchai Prakancharoen
Department of Computer and Information Science Faculty of Applied Science
King Mongkut's University of Technology North Bangkok,
Bangkok, Thailand
spk@kmutnb.ac.th

***Abstract*---Nowadays, most applications are transaction processing which are installed on enterprise server. Server has to examine if client's requests are authentic users or not. Most applications ask for login name and password of each user. After user's sending his login name and password, there were used to seek for correction whether it is appeared in client login name and password table or not. If there is found then server permit client to access application program. There are many problems happened in this situation such as password copying by intruders and server responsibility in password table serious keeping. This paper present a simplify login protocol which could be implemented easily. Intruder could not copy user's password which was sending through networking. Server still has to keep user's properties in secret but user's password may be arbitrarily and changed frequently subject to server preference. Performance of this protocol consume a little bit of computational time so that it should not bother user too much.**

Keywords-Secure login protocol, randon identify

I. INTRODUCTION

There are many security protocol were designed to support client's secure login to specific server. Normally, login name and password were sent to server. Computer server search user's login name and password table on received user's sending. If it is found in table then server permit client to access computer resource, otherwise server should ask for the correct new ones. This traditional action has many critical problems. First, server has to keep user's password in top security level seriously. Second, user has to manage his password freshness with password changing frequently to prevent password attack. The other problem is user's login name and password might be wiretapped during login message is traveling on network. This paper suggests a simplify security login protocol which could overcome all the above problems.

II. RELATED SECURITY PROTOCOL

There are many well known authentication protocols which were designed for using in secure login such as Kerberos and Wide mouthed frog.

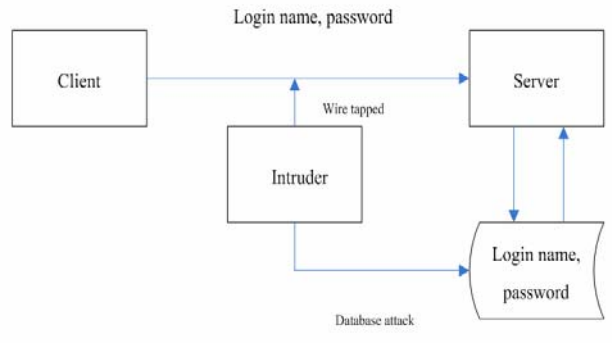

Figure 1. Insecure login protocol

A. Kerberos [1]

This protocol, client has to ask for ticket from authentication and ticket granting server for destination server connection. After that, client should send this ticket to target server. Destination server verify client from received ticket whether the client is authentic or not. If the user is authentic then destination server should either client to access computer resource or reject elsewhere. The problem of Kerberos protocol is that there has to defined authentication and ticket granting server. If there are many client's concurrency occurred requests then there may be a bottleneck affect happened. The other problems are timing synchronous between client and server which should not together matched if they were located in different time zone area.

B. Wide mouthed frog [2]

Wide mouthed frog is authentication security protocol which certify client to destination server. This protocol require authentication server to detect whether it is real client or not. If it is real client then authentication should create a certification message then client send this message to destination server later. Brief explanation of wide mouthed frog protocol is presented as this.

A- > S: A, [Ta, B, Kab]Kas
S - > B: [Ts, A, Kab]Kbs

While

- A, B, S are identities of client, destination server and

authentication server respectively.

- Ta, Ts are time stamps which are generated by client: A and Server: S.
- Kab is a conventional-session key between A and B.
- Kas is a conventional key between A and S.
- Kbs is a conventional key between B and S.

The problems of wide mouthed frog are global clock between all nodes requiring and sending message may be replayed by intruder.

III. THE DESIGN OF PURPOSED PROTOCOL

A. Requirement

The purpose designed secure login protocol has to solve some importance vulnerability points such as

- Server's responsibility in client's login name and password table keeping
- Bottleneck effect of ticket on ticket granting or certify server
- Pass word inspection by dictionary attacks
- Difference time zone of two parties
- Ease of use and consume a little bit of computation complexity

To accomplish all above requirements, public key system technique and simple exclusive or (.XOR.) arithmetic were used.

While

- A, B are identities of Client and Server
- Pra, Prb are Private key of A and B
- Pua, Pub are Public key of A and B
- R1,R2,…Ri,...,Rn are A's defined random number with the same amount of bit length.
- Rm is B's defined random number with the same length to Ri
- EPra, DPua are encrypt and decrypt with private and public key of A
- Eprb, Dpub are encrypt and decrypt with private and public key of B
- .XOR. is an exclusive .or. arithmetic operation
- [Identity of client, Rm] are attributes of each client property in client's login name and

password table which was kept in server site securely. Note that Rm should be difference from another client

- Kas is a session key which created by A. Kas is used in encryption and decryption of sending message between A and B.

B. Protocol designing

Scenario of designed protocol was illustrated in figure 2.

Step 0. – Preparation

In this step each one should together prepare their secure essential data before start login action.

- Sever: B

  - create client's login name and password table then append detail of each client with client's identity (login name) and Rm for each client such as "Alice,123", "Dave,512"

- Client: A

-create private and secure random number R1,R2,…, Ri…; i=1,n and all these random number should not equal to Rm if possible for more security of this protocol. The amount of user's login time number is n-1.

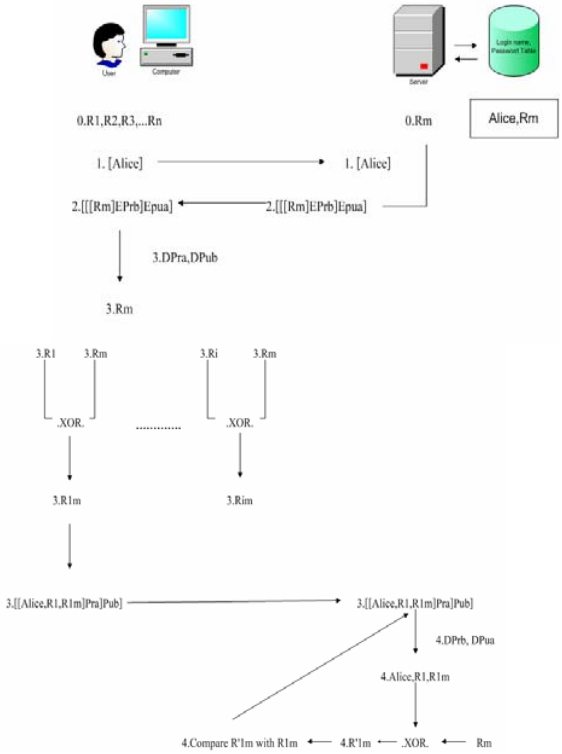

Figure 2. Purposed simplify secure login protocol

Step 1st – Client ask for connection

A connect to B.

A --- [Alice] --- > B

Step 2nd – Server response to Client's request

B acknowledge message from A by sending A's random number : Rm which search from login name and pass word table back to A. This content was encrypted with private key of B then encrypt again with public key of A.

B --- [[Rm]Prb]Pua]--- > A

Step 3rd –Client prepares required data and sent it to Server

A decrypt received message with A's private key then decrypt again with B public key. In this stage, A receive Rm from server. A pick up first his prior defined random number R1. R1 was exclusive .or. with Rm. The new random number R1m was encrypted with Pr of A then encrypt again with Pu of B. This message was sent to B.

A --- [Alice,[R1,R1m]Pra]Pub] --- > B

In addition, A could assign conventional session key (Kas) for encryption and decryption actions between A and B by add it together in this message.

A --- [Alice,[R1,R1m ,Kas]Pra]Pub] --- > B

Step 4th – Server check for validity of client's connection

B decrypt received message with server Pr of B. In this step server known that Alice want to connect with server then decrypt the unreadable part again with Pu of A. B search for random number of A (Rm) from login name and password table. If Rm was found it mean that Alice was authorized in server connection. Rm was exclusive .or. with R1 to produce R' 1m. B compared it with R1m. If there were the same value then server accept that A was righted to connect to B or reject connection else where.

Step 5th – Next login connection activities

After A finish session connection to B, if A want to login new session then A have to perform the same actions on step 2nd through step 4th again. Ri should be used in the ith login activity.

IV. Conclusion

This protocol was operated with simple mathematical operation (.XOR.) which suitable for computer arithmetic logical operation. Public key encryption which was used in transferring message between two nodes may a little consume computational time in encryption and decryption but improve privacy of message and authentic checking of each other.

Server B: unnecessary to keep password of each user in his own responsibility. Random numbers: Rm was only kept. Incase of Rm was stolen, intruder cannot use it as unique password for login since it has to operate with Ri of A before using under public key system.

The password or random number in this protocol was similar to one-time password which used only one time then discard it. This mean that intruder can not inspect A's password easily or even compile password replaying.

There no need to request ticket or certify message from any trusted server or ticket granting server then there are not bottleneck effect happening in this protocol.

V. Further Research

There are some vulnerability issues that have to be improved in next designs. A's defined random number might be stolen. Random number sequence Ri ought to be scrambled and pick it up according to new sequence. The others suggestion is to increase chain of random number (Rim) by prior random number (Ri-1) and some initial random number. Product of this random number will be integrated from many particles so that intruder could not easily use only stolen individual random number for illegal server login.

References

[1] Fulvio Ricciardi, The Kerberos Protocol and its Implementations, the National Institute of Nuclear Physics Computing and Network Services, Italy, 2006.

[2] Peter Ryan and Steve Schneider, The Modelling and Analysis of Security Protocols, Safari Press, Great Britain, 2001.