IJCSN - International Journal of Computer Science and Network, Volume 6, Issue 6, December 2017
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

An Improved Certificate - less Cryptography Scheme based on Time Stamping

[1] **Arun Kumar Singh;** [2] **Arun K Misra**

[1] Ewing Christian Institute of Management and Technology,
Allahabad, Uttar Pradesh, India

[2] SP Memorial Institute of Technology,
Kaushambi, Allahabad, Uttar Pradesh, India

Abstract - Replay attack is a typical breach of communication between two parties that threatens the very design of authentication and key distribution protocols. In this paper, an authentication protocol has been proposed that provides a strong authentication mechanism which is based on time stamping. The authentication protocol enables the sender to encrypt a message with recipient's identity only and users do not need certificates to bind identity with specific public key. High Level Protocol Specific Language (HLPSL), which is based on temporal logic has been used for formal verification.

Keywords - *Authentication Mechanism, Time Stamping, Public key cryptosystems, HLPSL*

1. Introduction

A characteristic feature of certain standardized communication protocols are the various quality issues including security. The aim of the security is that of authenticated message exchange. The server needs to be ensured that the request has been originated from the client and is not from a duplicate/unauthorized one, and the client to be convinced that the response to its request is from the designated server only. The goal of authenticated message exchange can be achieved using digital signatures. Another security goal is that of confidentiality. Both parties want to be sure that they are the only one using the information exchanged.

Encryption system largely depends on a secure key distribution used by the security protocols. Needham and Schroeder (NS) [1] proposed the first important authentication and key distribution protocol in 1978, which is now the basis of many authentication protocols [2]. Denning pointed out a flaw in NS protocol in 1981, which drew attention of the researchers [2. Replaying attack is a typical breach of secured communication between peers that threatens the very design of authentication and key distribution protocols.

Authentication confirms the identity of both communicating parties and distributes session key and guarantees the confidentiality of information transmitted. In 1976, Diffie and Hellman proposed the idea of public key [3] and Rivest, Shamir and Adleman proposed the famous RSA public key algorithm in 1978 [4]. Public key cryptosystems do not need the distribution of the private key system for establishing secure communication. But the efficiency of the algorithm is slower than the one using private key and is not suitable for large data encryption [4].

Authentication protocols focus on the public key cryptography, which contain a key distribution server to distribute a key pair (public and private key) to user. It is always easy to get access to public key of other, because that is publically available and used for encryption, which is not the case with the private key. As such, for secure decryption, private key must be used for authentication, e.g. Kerberos protocol and others are based on public key encryption which uses CA certificate to distribute public key, such as X.509 protocol. In 1983, Dolev and Yao pointed out that a protocol can be designed on the assumption that cryptosystem and technology are good in the sense of reliability [5, 6, 7, 8]. A replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. "This is carried out either by the originator or by an adversary who intercepts the data and retransmits it, possibly as part

of a masquerade attack by IP packet substitution"[9, 10, 13, 14, 15].

Man-in-the-Middle Attack is a type of Replay Attack in cryptographic schemes. The MITM is a form of active eavesdropping in which the attacker makes independent connections with the victims (typically end users and banks) and relays messages between them, making them believe that they are talking directly to each other over a private connection where as in fact the entire conversation is controlled by the attacker. The countermeasure taken to prevent MITM attacks is to authenticate both the client and server. The technology to implement a safe transaction utilizes x.509 certificates in a public key infrastructure deployment, but attackers can create a fake certificate.

MITM attack has been successful established on Diffe-Hellman (DH) authentication protocol [MITM-DH] and Needham Scroceder protocol. G. J. Lowe found the NS protocol vulnerable to MITM attack and proposed a new protocol but this was also vulnerable to type flow attack [6]. In the next section two authentication protocols i.e. Li [2] and Raman et al. [11] have been analysed for vulnerabilities and solutions have been proposed.

1.1 Attack on authentication protocol by Li

Li [2] has proposed an authentication protocol in which communication is always through server and symmetric key cryptography has been used for encryption and decryption. The following symbols have been used in the description of this protocol:

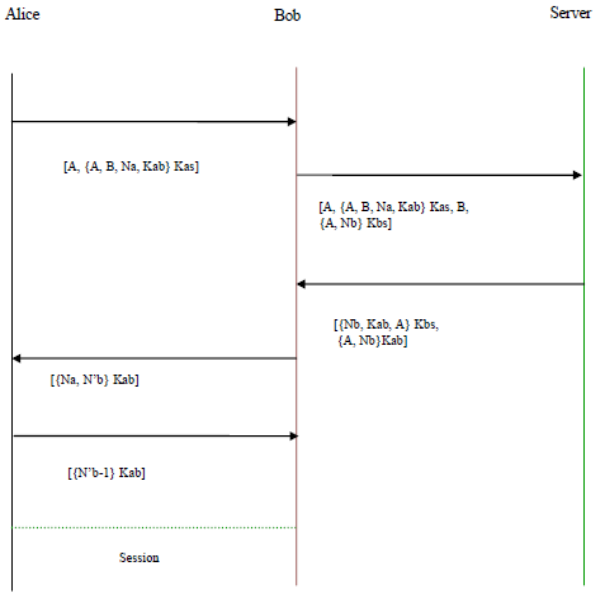

Figure 1 Key Distribution Protocol [Li]

A: User A (Sender)
B: User B (Receiver)
S: Server
N_A: Nonce generated by user A
N_B: Nonce generated by user B
K_{ab}: Key i.e. symmetric key between user A and user B
K_{as}: Key i.e. symmetric key between user A and server S
K_{bs}: Key i.e. symmetric key between user B and server S
{ }: The curly bracket contains the message
If user A wants to communicate with B then following steps are followed:

Step 1: In the first step A sends his message to B using K_{as}, which consists of the names of A and B, a nonce N_A and a key K_{ab} generated between the sender and receiver by A. The whole process is shown in figure 1.

$A \rightarrow B : \{A, \{A, B, N_a, K_{ab}\}_{Kas}\}$

K_{as}: is symmetric key between A and server S.

K_{ab}: is session key generated by A

1.2 Proposed Attacks

The protocol has been analyzed for the following attacks:

1. If the attacker is insider and intercepts the message being send by A and sends it to server with his own nonce and his symmetric key, the protocol become vulnerable to Man in the Middle (MITM) by an insider.
2. If the server compromised then attacker can access to all communication between user and server.

The above two attacks are as follows:

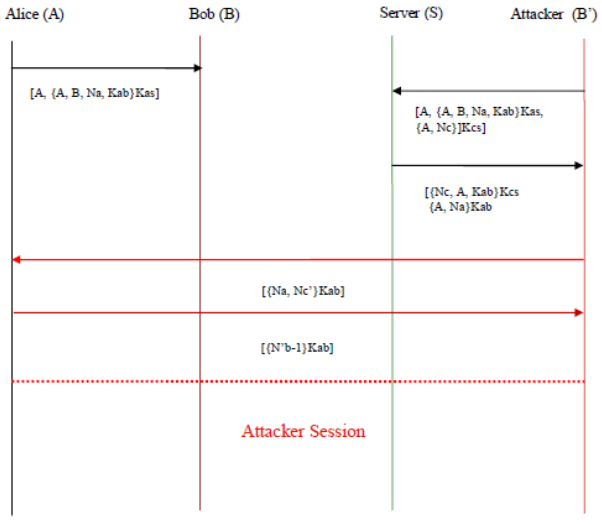

Figure 2 Proposed MITM Attak on Li Key Distribution Protocol

Man in the Middle attack: The MITM attack has been shown in figure 2. In this proposed attack, attacker listens to each and every communication and intercepts the message. The attacker B' intercepts the message sent by A and modifies this message by replacing $\{A, Nb\}_{Kbs}$ by $\{A, Nb'\}K_{b's}$ and sends to server the following:
B'→S: $\{A,\{A, B, Na,K_{ab}\}K_{as}, B' \{A,Nb'\}K_{b's}\}$
Instead of replying to B, server will now reply to attacker B' with key K_{ab} , which will be encrypted by $K_{b's}$. This will facilitate the attacker to decrypt the message and B' will also send $\{Na, Nb'\}K_{ab}$ to A. As such the sender never knows that his message has been intercepted.

In the other attack, if the server is compromised then attacker will know K_{as} and will be able to access every information.

1.1.1 Attack on Authentication Protocol by Raman et al.

Raman et. al. [11] has proposed an authentication protocol in which communication is through server and Symmetric and Asymmetric key cryptography has been used for encryption and decryption.

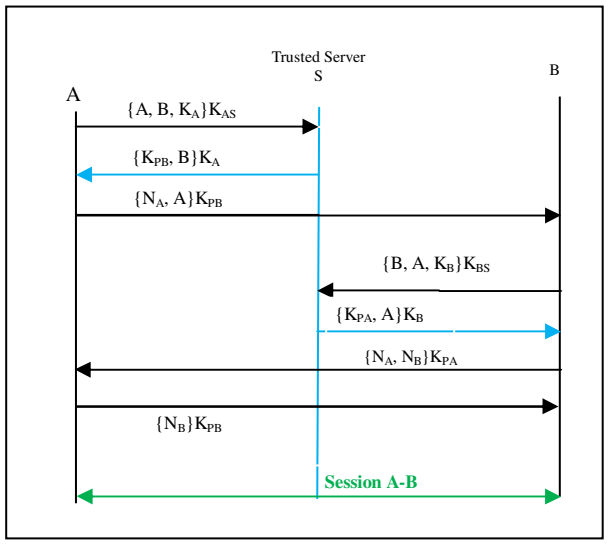

Figure 3 Authentication Protocol by Raman et al.

The following symbols have been used in the description of this protocol:
A: User A (Sender)
B: User B (Receiver)
S: Server
N_A: Nonce generated by user A
N_B: Nonce generated by user B
K_{as}: Key i.e. symmetric key between user A and server S
K_{bs}: Key i.e. symmetric key between user B and server S
K_{PA}: Public key of A
K_{PB}: Public key of B
K_A: Symmetric key generated between A and S
K_B: Symmetric key generated between B and S
{ }: The curly bracket contains the message

If user A wants to communicate with B then following steps are followed. The whole process ia shown in figure 3.

1. In the first step A request to S for B's public key using Kas , which consists of the names of A and B, and a key K_{ab} generated between the sender and receiver by A. The whole process is shown in figure 3.
   $A \rightarrow S: \{A, B, K_A\}K_{AS}$
2. Upon receipt of K_A from A, S will decrypt the message and will send the following to A.
   $S \rightarrow A: \{K_{PB}, B\}K_A$
3. A generate a random nonce N_A and send to B with identity of A and the message encrypted by public key of B
   $A \rightarrow B: \{N_A, A\}K_{PB}$
4. B receive the message and request to S for A's public key using Kbs , which consists of the names of B and A, and a key K_A generated between the server and B
   $B \rightarrow S: \{B, A, K_B\}K_{BS}$
5. Upon receipt of K_B , S will decrypt the message and send the following
   $S \rightarrow B: \{K_{PA}, A\}K_B$
6. Upon receipt of K_{PA} from server, B will decrypt the message and will send the following to A
   $B \rightarrow A: \{N_A, N_B\}K_{PA}$
7. Upon receipt of N_b, A will send the acknowledgement as follows
   $A \rightarrow B: \{N_B\}K_{PB}$

1.1.2 Proposed Attack on Raman Protocol

The protocol has been analysed for the following attacks:
1. If the attacker is insider then Man in the Middle (MITM) is possible.
2. If the server is compromised, then attacker can access to all communication between user and server.

The above two attacks are as follows and shown in figure 4. In this attack, the attacker can get an access to public key of user A and B as discuss below. Because this is an insider attack the attacker is also a registered user with the server.

Step 1: Now, let us assume that A is sending a message to I (attacker), encrypted by public key of attacker i.e. K_I, the message contain nonce N_A generated by A and user name A.
A→I: $\{N_A, A\}K_{PI}$

Step 2: When, I received the message from a, then decrypt the message, after that encrypt the whole message by public key of B i.e. K_{PB} and send to B.
I→B: $\{N_A, A\}K_{PB}$

Step 3: Now, B reply to I, because B thought the message is send by A. B generated a nonce N_B along with N_A, encrypted by K_{PA} public key of A.
B→I: $\{N_A, N_B\}K_{PA}$

Step 4: Attacker I forward this message as it is to A.
I→A: $\{N_A, N_B\}K_{PA}$

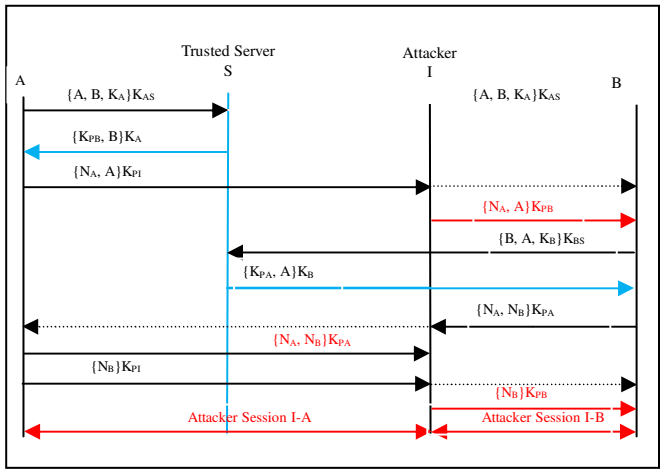

Figure 4 Proposed Attack on Raman Protocol

Step 5: A will send the acknowledgement to I as follows.
A→I: $\{N_B\}K_{PI}$

Step 6: Now, attacker decrypt the message received buy A, and forwarded to B with encrypted by K_{PB}
I→B: $\{N_B\}K_{PB}$

Now, attacker created two sessions, one is with A i.e. A-I session and other is with user B i.e. I-B session, so this protocol is vulnerable w.r.to the MITM attack. In the other attack, if the server is compromised then attacker will know public and private key pair and will be able to access every information.

In the protocol suggested by Raman [Raman], it is possible for any registered user to get an access to the public key, which he/she can send to anyone. As such there is no meaning of "requesting for public key from server", and no use of K_A and K_B. As far as the private key is concerned, it will also have no meaning if server or any registered user is compromised.

Visualizing the weakness in the protocols suggested by Li [2] and Raman [11], a scheme has been proposed, which is described in the next section.

2. Proposed Scheme for Li and Raman Authentication

The proposed scheme aims to remove the weaknesses identified in previous section and enhance the security through client side, and reduce the security trust on server side. Further, use of certificates issued by CA has also been avoided, because attackers are also able to generate fake certificate and succeed in breaching security.

In the proposed scheme, Certificateless cryptography and time stamping has been used. The later one ensures that, after a particular time period or when the session's time out, previous authentication data has no more validity. In the proposed Certificateless cryptography, the server creates public keys with respect to the identity of the user and half of the private key i.e. partial private key. Other half of the private key is generated by the user. As such in case of server compromise, the attacker will not be able to obtain the sensitive information because as the full private key for decryption is not available with the server.

The proposed scheme has following advantages:

1. No certificate is required, so attacker cannot intercept the communication via fake certificate.
2. In the IBE encryption technique, key escrow issues have been found, but in the proposed scheme key escrow issues have been removed, because here trusted server is generating only half of the private key and rest of the key is generated by the user himself.

3. Further, if the attacker attempts to predict the private key with other parameters, because of time stamping his intentions will fail.

The proposed protocol model has been verified in AVISPA [12] through HLPSL language through simulation, control flow pattern of the protocol have been obtained and security verified.

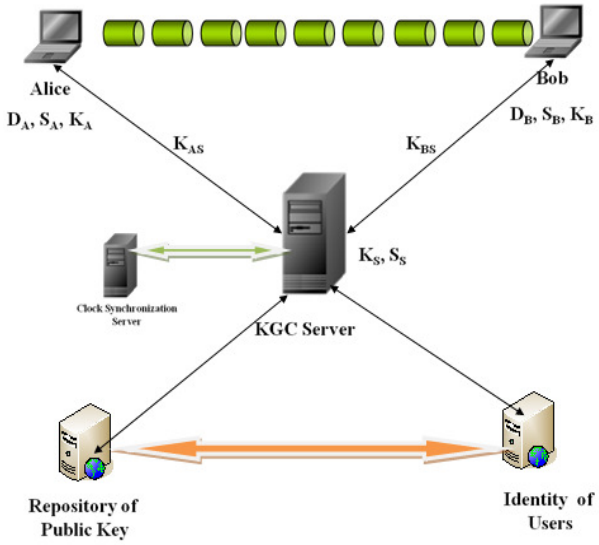

Figure 5 Secure Communications by Authentication Server

In this proposed scheme ID based nonce sequences are used for secure communication. A is sending a request to server (S) for registration and gets the shared symmetric key Kas between A and S. A requests to key generated server (KGS) for public key and partial private key (PPK). The server, after getting PPK, generates the private key using random number x_A. Same procedure for B is followed.

All the communication from A to B is totally secure and each stage is authenticated as well as verification has been carried out. No chance for replay attack has been left and the design is based on Certificateless PKI concept, so key escrow problem has been totally resoled. Partial dependence on KGC for exchanging the message for authentication happens to be the key of security.

The following symbols have been used in the description of this protocol:
A: User A (Sender) or Agent A
B: User B (Receiver) or Agent B
PK_A: Public key of A
PK_B: Public key of B
inv(PK_A) : Private key of A
inv(PK_B) : Private key of B
N_A: Nonce generated by user A
N_B: Nonce generated by user B

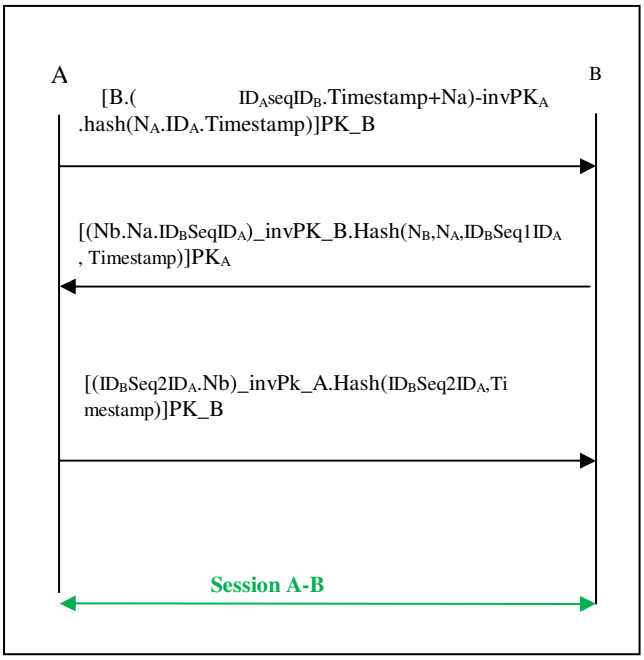

Figure 6 Proposed Protocol

Timestamp: System time is recorded at the time of sending a message
Hash ( ): Hash value of the message within the bracket
ID_Aseq1D_B : A random sequence number sent by A to B, with identity of users
ID_Bseq2ID_A: A random sequence number sent by B to A, with identity of users
{ }: The curly bracket contains the message

For communication between A nad B, following steps are followed and shown in figure 6.

Step 1: In the first stage A is sending a message to B, which consists of the identity of user A and B with the random sequence number seq, a nonce N_A generated by A along with timesatmp and this message is encrypted by the private key of A, and hash of N_A, ID_A, and timestamp and the whole message is encrpted by public key of B.

A→B: [B.(ID_AseqID_B.Timestamp+N_A)-invPK$_A$ .Hash(N_A.ID_A.Timestamp)]PK_B

Step 2: Upon receipt of the message from A, B sends a message to A, which consists of N_A, and a nonce generated by N_B along with identities of A and B with random sequence and encrypted by the private key of B.

It also contains Hash of N_B,N_A , identity with random sequence number with time stamp and this whole message is encrypted by the public key of A i.e. PK_A .

B→A :
[(Nb.Na.ID_BSeq2ID_A)_invPK_B.Hash(N_B,N_A,ID_BSeq1ID_A, Timestamp)]PK_A

Step 3: Now, A sends the acknowledgement which contains the identities of the A, B along with the random sequence number, with nonce N_B, encrpted by private key of A and a Hash of the identities of the A, B along with the random sequence number with timestamp and the whole message encrpted by the public key of B.

A→B:
[(ID_BSeq2ID_A.Nb)_invPk_A.Hash(ID_BSeq2ID_A,Timestamp)]PK_B

Now, a session key K has been created which is the hash of the nonce N_A,N_B , identity of the user i.e. ID_A, ID_B, random sequence numbers ID_ASeq1ID_B, ID_BSeq2ID_A , public key of A and B, and timestamp.

K= K_{AB}=H{{N_A,N_B}||{ ID_ASeq1ID_B, ID_BSeq2ID_A }||{ID_A}||{ID_B}||PK_A||PK_B||{Timestamp}}
Hash is the one way function, so attacker can not the get this session key, and for every session the key will be changed.

Time stamping is used, by user A, when A sends a message to B, then user B matches this time with the threshold value of time as well as the value of delay.
Let, Time stamp in the message=T_{stamp}
Threshold value =T_h
Time delay to reach the message =T_D
Current time =T_C at which user services the message then
$T_C <= T_{stamp}+T_h+T_D$
$T_C-T_h-T_D \leq T_{stamp}$
$T_C-T_h-T_D \leq T_{stamp} \leq T_C+T_h$ **for its logic.**

The proposed scheme has been formally verified/as explained below:

3. Formal Verification

It is very hard problem to design a secure protocol and replay attack is simply overlooked by human. High Level Protoco Specific Language (HLPSL), which is based on temporal logic has been used for formal verification.

HLPSL allows for specification of different cryptographic operatios and their algebric properties, adversary model/ intruder model. AVISPA integrate four backend On The Fly Model Checker (OFMC) [12].

HLPSL supports symmetric and asymmetric key, hash function, algebric function etc. And uses .hlpsl extension for saving the file.
HLPSL contains basic roles, parallel and sequential composition, session, environment etc. the keywords and variables in HLPSL are as follows:

Agent: Principle names are represented by agent, and intruder is always assumed to have the special identifier (I or i).

Asymmetric Key: The public key is represented by P_K and private key is represented by inv(P_K).

Symmetric Key: K_a and K_b the symmetric key for encryption.

Text: Text value is used as a nonce.

Nat: natural number represented in non message context.

Function: Hash function is a one way function and is used for modeling.

Concatenation of message is via associative operation "**.**". Encrypted message is represented as $\{message\}_{Key}$. Intruder has been introduced through Dolev-Yao model (dy).

^- : Operator express conjunction, and /\ operator is for parallel composition.
=|> : Indicate immediate reaction, ex. X=|>Y represent event X and action Y.
SND: is used for sending channel
RCV: is used for receiving channel

The simulation results is given in figure 1.8 and the code for simulation is as follows:

```
################ A and B is communicating ##########
role arun (A,B       : agent,
           SND,RCV   : channel(dy),
           Hash      : hash_func,
           PK_A , PK_B    : public_key,
           IDAseqIDB, IDBseqIDA : text)
played_by A
def=
  local
    State     : nat,
```

```
   Timestamp+Nonce       : text,
Timestamp     : text,
   Nonce      : text
 init  State := 0
 transition
1. State  = 0 /\ RCV(start) =|>
   State':= 2 /\ Timestamp+Nonce' := new()/\ SND({
B.{IDAseqIDB.Timestamp+Nonce'}_inv(PK_A).Hash(PK
_A.Timestamp')}_PK_B)
            /\
witness(A,A,Timestamp+Nonce,Timestamp+Nonce')

3. State  = 2 /\ RCV({{Nonce'}_inv(PK_B).Hash(Nonce'.
Timestamp')}_PK_A) =|>
   State':=                    4                    /\
SND({{IDBseqIDA.Nonce'}_inv(PK_A).Hash(IDBseqID
A. Timestamp')}_PK_B)
end role


role brijesh (B,A       : agent,
        SND,RCV    : channel(dy),
        Hash       : hash_func,
        PK_A , PK_B      : public_key,
        IDAseqIDB,IDBseqIDA  : text)
played_by B
def=
 local
   State      : nat,
   Nonce      : text,
Timestamp      : text,
   Timestamp+Nonce       : text
 init State := 1
 transition
1.        State              =        1        /\        RCV(
{B.{IDAseqIDB.Timestamp+Nonce'}_inv(PK_A).Hash(PK_
A.Timestamp')}_PK_B)  =|>
   State':= 5 /\ Nonce' := new()/\
SND({{Nonce'}_inv(PK_B).Hash(Nonce'.
Timestamp')}_PK_A)

3. State  = 5 /\ RCV({IDBseqIDA.Nonce}_inv(PK_A))
=|>
   State':=                    7                    /\
request(A,A,Timestamp+Nonce,Timestamp+Nonce)
end role

role session(A,B       : agent,
        Hash       : hash_func,
        PK_A, PK_B : public_key,
        IDAseqIDB,IDBseqIDA  : text)
def=
 local SNDA,RCVA,SNDB,RCVB  : channel (dy)
 composition

arun(A,B,SNDA,RCVA,Hash,PK_A,PK_B,IDAseqIDB,IDBse
qIDA)
 /\
brijesh(B,A,SNDB,RCVB,Hash,PK_A,PK_B,IDAseqIDB,IDB
seqIDA)
end role
```

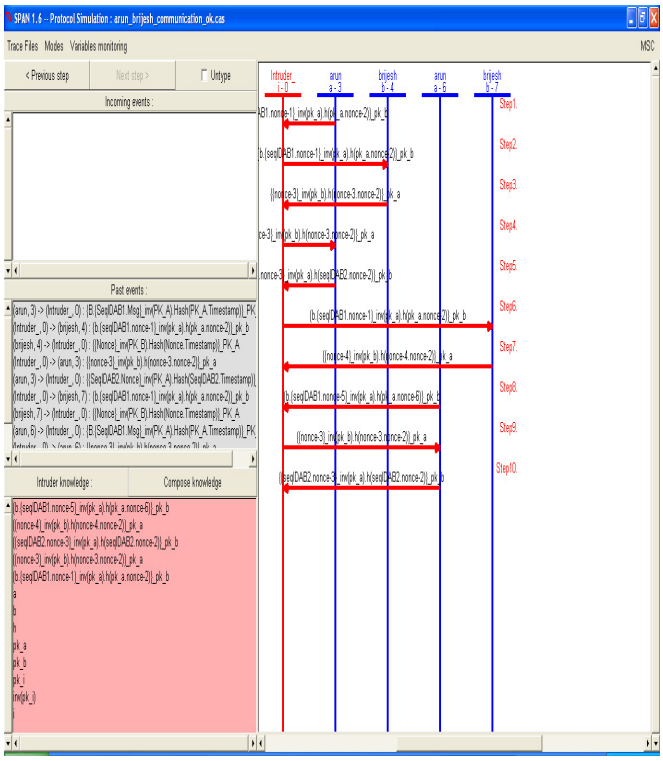

Figure 7 Proposed protocol simulation by HPLSL

```
role environment() def=
 const
   a,b          : agent,
   h            : hash_func,
   Timestamp+Nonce          : protocol_id,
   pk_a,pk_b,pk_i : public_key,
     IDAseqIDB,IDBseqIDA     : text

 intruder_knowledge = {a,b,h,pk_a,pk_b,pk_i,inv(pk_i)}

 composition
   session(a,b,h,pk_a,pk_b,IDAseqIDB,IDBseqIDA)
/\ session(a,b,h,pk_a,pk_b,IDAseqIDB,IDBseqIDA)

end role
```

```
goal
  % Sender Invariance (G16)
  authentication_on Timestamp+Nonce
end goal
environment()
```

4. Conclusion

Authentication protocols are generally vulnerable due to replay attacks.. The two protocols suggested by Li and Raman have been analysed and both of them were found to be vulnerable to replay attacks.

Based on the weaknesses of protocols of Li and Raman the proposed scheme has been designed using Certificateless cryptography and time stamping and has been established to have the following advantages.

1. No certificate is required, so attacker cannot intercept the communication via fake certificate.
2. In the IBE encryption technique, key escrow issues have been found, but in the proposed scheme key escrow issues have been removed, because here trusted server is generating only half of the private key and rest of the key is generated by the user himself.

Further, in the proposed Certificateless cryptography, the server creates public keys with respect to the identity of the user and half of the private key i.e. partial private key. Other half of the private key is generated by the user. As such in case of server compromise, the attacker will not be able to obtain the sensitive information because the full private key for decryption is not available with the server.

References

1. R. Needham and M. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," Communications of the ACM, Vol. 21, No. 12, December 1978.
2. Junhong Li , "Design of Authentication Protocols Preventing Replay Attacks" , College of Mathematics and Information Science Hebei Normal University, 2009.
3. W. Diffie, M. E. Hellman, "New Directions in Cryptography," IEEE transactions on Information Theory, vol. IT-11, pp. 644-654, November 1976.
4. Jeremy Brun-Nouvion Hicham Hossayni, Logical Attacks Using RSA, Security Models Lecturer 2010.
5. Danny Dolev and Andrew C. Yao. On the security of public-key protocols. IEEE Transactions on Information Theory, 2(29):198–208, 1983.
6. Gavin Lowe, "An attack on the Needham-Schroeder public key authentication protocol", Information Processing Letters, 56(3):131—136, November 1995.
7. L. Gong, "Verifiable-text Attacks in Cryptographic Protocols," Proceedings of IEEE
8. Li Gong, Variations on the Themes of Message Freshness and Replay -or the Difficulty in Devising Formal Methods to Analyze Cryptographic Protocols, SRI International Computer Science Laboratory 333 Ravenswood Avenue Menlo Park, California 94025 U.S.A.
9. Amir Herzberg, "Internet Cryptography Tools", Computer Science Department, Bar Ilan University, 2003.
10. Arun K Singh and Arun K. Misra, Analysis of Cryptographically Replay Attacks and Its Mitigation Mechanism, International Conference on Information Systems Design and Intelligent Applications-2012 (INDIA-2012)
11. Raman Kumar et. al. "An Image Based Authentication System- Using Needham Schroeder Protocol, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.11, November 2010.
12. AVISPA, Automated Validation of Internet Security Protocols and Applications, http://www.avispa-project.org/.
13. Courtois, N. & Meier, W., Algebraic attacks on stream cipher with linear feedback, in E. Biham, ed., 'Advances in Cryptology - Eurocrypt 2003', Vol. 2656 of Lecture Notes in Computer Science, Springer.
14. David Ahmad "Attack Trends :Two Years of Broken Crypto", Published by the IEEE Computer Society, IEEE Security & Privacy, 2008.
15. Fahime Javdan Kherad, Hamid R. Naji, Mohammad V. Malakooti and Payman Haghighat, A New Symmetric Cryptography Algorithm to Secure E-Commerce Transactions, 2010 International Conference on Financial Theory and Engineering, Department of Computer Engineering, IAU Dubai, U.A.E.
16. A. Fiat, "Batch RSA", Journal of Cryptology, (1997)10: page 75-88.
17. Dan Boneh, "Fast Variants of RSA", CryptoBytes, Vol. 5, No. 1, pp. 1-9, 2002.
18. Danny Dolev and Andrew C. Yao. On the security of public-key protocols. IEEE Transactions on Information Theory, 2(29):198–208, 1983.
19. Eli Biham Orr, Dunkelman, National, Differential Cryptanalysis in Stream Ciphers, eprint-2007.
20. Joris Claessens, Valentin Dem, Danny De Cock, Bart Preneel and Joos Vandewalle" On the Security of Today's Online Electronic Banking Systems" Elsevier, Computers & Security, Vol 21, No 3, pp 257-269, 2002
21. L. Gong, "Verifiable-text Attacks in Cryptographic Protocols," Proceedings of IEEE
22. Li Gong, Variations on the Themes of Message Freshness and Replay -or the Difficulty in Devising Formal Methods to Analyze Cryptographic Protocols, SRI International Computer Science Laboratory 333

Ravenswood Avenue Menlo Park, California 94025 U.S.A.

23. M.H. Sherif, A. Serhrouchni, A. Y. Gaid and F. Farazmandnia, "SET and SSL: Electronic payments on the Internet", IEEE, 1998.
24. Matt Blumenthal, "Encryption: Strengths and Weaknesses of Public-key Cryptography", Department of Computing Sciences Villanova University, Villanova, PA 19085 CSC 3990 – Computing Research Topics, 1999.
25. Stefek Zaba, "Cryptographic Security in the Internet Protocol Suite: Practice and Proposals", Elsevier Information Security Technical Report, Vol. 2, No. 2 (1997) 54-73.
26. W. Diffie, M. E. Hellman, "New Directions in Cryptography," IEEE transactions on Information Theory, vol. IT-11, pp. 644-654, November 1976.
27. W. Küchlin, "Public key encryption", ACM SIGSAM Bulletin Volume 21 Issue 3, Aug. 1987 Pages 69-73.
28. Wang Yanhua Yang Kuihe Zhang Yun, "Research and Realization of Security Proxy Based on SSL Protocol" The Eighth International Conference on Electronic Measurement and Instruments ICEMI'2007.
29. Xianxian Li, Jun Han, Zhaohao Sun, " Design Principles and Security of Authentication Protocols with Trusted Third Party" AUG 2004 - Who Are You?
30. Yuping Deng, Xiaowei Guo, and Xiamu Niu , "A New Design Scheme of Role-Based Access Control Based on PKI Yuping Deng, Xiaowei Guo, and Xiamu Niu" , Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06), 2006 IEEE.
31. Zhikao Ren, Minghua Liu, Chen Ye and Chuansheng Wang, "A Scheme of E-Commerce Security based on ECC & SSL Protocol", 2009 IEEE.

Dr. Arun Kumar Misra:

Arun Kumar Singh: Corresponding Author: Arun Kumar Singh received his B.Tech in Electronics and Communication from SRMCEM College, Lucknow, Uttar Pradesh, India in 2005. He received his Master degree in Information Security from Indian Institutes of Information Technology, Allahabad, Uttar Pradesh, India in 2008. Currently, he is completed the Ph.D. degree in Computer Sciences and Engineering at the Motilal Nehru National Institute of Technology (MNNIT), Uttar Pradesh, India. His research interests include network security, network protocol design and verification, in network security, Cryptography and Computer Forensic fields.

Dr. Arun Kumar Misra has forty years of teaching experience at Motilal Nehru National Institute of Technology, Allahabad, India and is presently working at S.P. Memorial Institute of Technology, Allahabad, India. His special field of interest include Software Engineering, Information Security, Soft Computing and Optimization Techniques.