Dhaka Univ. J. Eng. & Tech. Vol. 1(1) 41-44, 2010 (July)

Improved Needham-Schroeder Protocol for Secured and Efficient Key Distribution

Shamima Sultana[1], **Md. Ismail Jabiullah**[2] **and M. Lutfar Rahman**[3]

[1]*Department of Computer Science and Engineering, Dhaka City College, Dhaka.*

[2]*Department of Software Engineering, Daffodil International University, Dhaka,*

e-mail: mijjabi@daffodilvarsity.edu.bd.

[3]*Department of Computer Science and Engineering, University of Dhaka, Dhaka.*

Abstract

A key distribution procedure is an essential constituent of secured exchange of information between the participants. In this paper, a fast symmetric key distribution technique with additional security services is presented. The aim of the proposed technique is to improve the conventional Needham-Schroeder five-message protocol in four aspects. The first aspect is to introduce an additional authentication level in originator's identity and the second aspect is to provide the integrity of the originator's message. The third aspect is to reduce the time needed to distribute a session-key between a pair of entities, and the fourth aspect is to develop the key freshness for security.

Keywords: Symmetric key, key distribution center, Needham-Schroeder Protocol and key freshness.

1. Introduction

Key Distribution refers to delivering of a key between two communicating parties who wish to exchange data, without allowing others to see the key[1]. Symmetric cryptographic schemes require both the communicating parties to share a common secret key. But the prime issue of the communication is how to distribute this key securely [2], [3], [4]. Two parties A and B have various key distribution alternatives [2] [3], as listed below.

(a) Party A can select key and physically deliver it to B.
(b) A third party can select and deliver a key to A and B.
(c) If party A and B have communicated previously, they may use previous key to encrypt and distribute a new key.
(d) If party A and B have secure communications with a third party C, C can relay key between A and B.

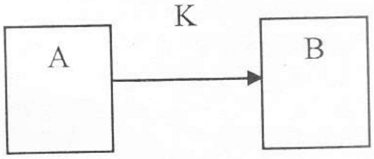

Fig 1: Point-to-Point Key Distribution Model

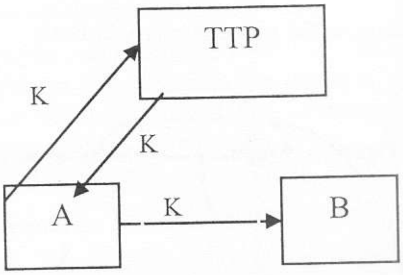

Fig. 2: Centralized Key Distribution Model

Physical delivery (option a and b) is simplest - but only applicable when there is personal contact between recipient and key issuer. A third party is a trusted intermediary, whom all parties trust, to mediate the establishment of secure communications between them [5], [6]. As numbers of parties grow, some variant of the option last is the only practical solution and widely adopted [7]. In this proposed work, option (d) is used with the centralized Key Distribution Center (KDC) as the trusted third party.

2. Key Distribution Models

Two Simple Key Distribution Models are: Point-to-Point Key Distribution and Centralized Key Distribution Model [8]. Point-to-Point Key Distribution model (Figure 1) involves two parties communicating directly and Centralized Key Distribution Model (Figure 2) use a Trusted Third Party (TTP) to distribute a key between the communicating users [9].

3. Conventional Needham-Schroeder Key Distribution Technique

Many existing authentication protocols are derived from the Needham-Schroeder protocol [10]. Here, party A makes contact with the KDC, who provides A with the session key, $K_{ab,}$ and a certificate encrypted with B's key conveying the session key and A's identity to B, (Figure 3). Then B decrypts this certificate and carries out a nonce handshake with A to be assured that A is present currently, since the certificate might have been a replay [11], [12]. Interpretation of messages m1, m2, m3, m4 and m5 are given below.

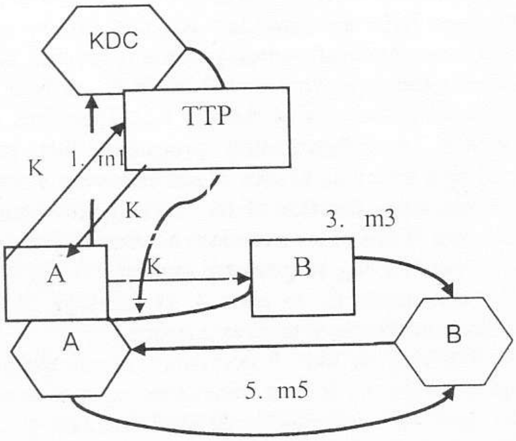

Fig. 2: Centralized Key Distribution Model

Here,
Step 1: $m_1 = (IDA, IDB, n_A)$, where
n_A = code for the request made by user A.
IDA= identifier of user A.
IDB= identifier of user B.
Step 2: $m_2 = E_{Ka}(K_{ab}, n_{A,} IDB, E_{Kb}(K_{ab}, IDA))$, where

K_{ab} = secret key generated by the KDC for secure communication between users A and B.
$C_1 = E_{Kb}(K_{ab}, IDA)$
K_a = private key of user A.
Step 3: $m_3 = C_1$
Step 4: $m_4 = C_2 = E_{kab}(n_B)$, where, n_B = a random number (nonce) generated by user B.
Step 5: $m_5 = C_3 = E_{kab}(n_B\text{-}1)$.

The working process of conventional technique is as follows:

(a) User A sends a request message m1 to the KDC indicating that it wants to establish a secure logical communication with user B. The message contains a code for the request n_A, the identifier of A (ID_a) and the identifier of B (ID_b). This message is transmitted from user A to KDC in plaintext form.

(b) On receiving m_1, the KDC extracts from its table the keys K_a and K_b, which corresponds respectively to the user identifiers ID_a and ID_b in the message. It then creates a secret key K_{ab} for secure communication between user A and B. By using K_b the KDC encrypts the pair (K_{ab} , ID_a) to generate the cipher text $C1 = E((K_{ab}, ID_a), K_b)$. Finally it sends a message m2 to user A that contains n_A,, ID_a , K_{ab} , C_1 . The message m2 is encrypted with the key K_a so that only user A can decrypt it.

(c) On receiving m2 user A decrypts it with its private key K_a and checks whether n_A and ID_a of the message match with the originals to get confirmed that m_2 is the reply for m_1. If so, user A keeps the key K_{ab} with it for future use and sends a message m_3 to user B. This message contains cipher text C1. Note that only user B can decrypt C1 because it was generated using key K_b.

(d) On receiving m3 user B decrypts C1 with its private key K_b and receives both K_{ab} and ID_a. At this stage both the users have the same key K_{ab} that can be used for secure communication between them because no other user has this key. Now user B needs to verify if user A is also in possession of the key K_{ab}. Therefore, user B initiates an authentication procedure that involves sending a nonce n_B to user A and receiving a reply that contains some function of the recently sent nonce. For this, user B generates a random number n_B, encrypts n_B by using key K_{ab} to generate cipher text $C_2 = E\ (n_B, K_{ab})$ and sends C_2 to user A in message m_4 . The random number n_B is used as a nonce.

(e) On receiving m_4 user A decrypts C2 with the key K_{ab} and retrieves n_B. It then transforms n_B to a new value $N_t = n_B\text{-}1$ by a previously defined function *f*. User A encrypts N_t by using K_{ab} to generate the cipher text $C3 = E(N_t, K_{ab})$ and sends C3 to user B in message m5 .

(f) On receiving m5 user B decrypts C3, retrieves N_t, and applies the inverse of function *f* to N_t to check if the value obtained is n_B. If so, user B gets confirmed that a secure channel has been created between user A and user B by using key K_{ab}. This is enough to achieve mutual confidence and from now on the exchange of actual message encrypted with key K_{ab} can take place between users A and B.

4. Proposed Key Distribution Technique

The replay attack of the original protocol is removed in the following proposed method and three additional security services namely: Authentication of Originator's Identity, Originator's Message Integrity and Key-Freshness are introduced. Another major point is parallel transfer of symmetric key by the KDC server to the pair A and B (Figure 4). This parallel transfer of symmetric key saves much time than to distribute symmetric key sequentially from the KDC to the initiator node and then the symmetric key is sent to the responder node (from the initiator) to which the initiator wants to communicate. Interpretation of messages as follows:

$m1 = IDA, IDB, E_{Ka}(IDA, IDB ,n_A)$.
where n_A = code for the request made by user A.
IDA= identifier of user A.
IDB= identifier of user B.
K_a = private key of user A.
$m2 = E_{Ka}(K_{ab}, n_{r,} n_A)$
where, K_{ab} = secret key generated by the KDC for secure communication between user A and B.
n_r= Common nonce
$m3 = E_{Kb}(K_{ab}, n_{r,} IDA)$
where, K_B = private key of user B.
$m4 = E_{Kab}(n_r)$
$m5 = E_{Kab}(n_r)$

Step1. A->KDC:
$m1 = IDA, IDB, E_{Ka}(IDA, IDB ,n_A)$
Step2. KDC->A: KDC->B:
$m2 = E_{Ka}(K_{ab}, n_{r,}\ n_A)$
$m3 = E_{Kb}(K_{ab}, n_{r,} IDA)$
Step3. B->A:
$m4 = E_{Kab}(n_r)$
A->B:
$m5 = E_{Kab}(n_r)$

The summary of the procedure is as follows:

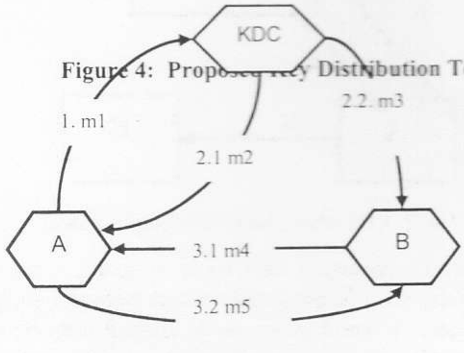

Figure 4: Proposed Key Distribution Technique

Fig. 4: Proposed Key Distribution Technique

The working process of proposed technique is as follows:

(a) User A sends a request message m1 to the KDC indicating that it wants to establish a secure logical communication with user B. The message contains the identifier of A (ID_a), the identifier of B (ID_b) and a ciphertext of (IDA, IDB and a code for the request $n_{A)}$ which is encrypted by the private key K_a.

(b) On receiving m2, the KDC extracts from its table the keys K_a and K_b, which corresponds respectively to the user identifiers ID_a and ID_b in the message. KDC decrypts the ciphertext part of m1 with the private key K_a and checks whether ID_a and ID_b of the message match with the originals to get confirmed that m1 is sent by the valid originator A and also checks the integrity of message m1. It then creates a secret key K_{ab} for secure communication between users A and B. It then generates a random number n_r which will be used by A and B to authenticate each other. Then it creates two messages m2 and m3, for A and B respectively, and sends them simultaneously. The message m2 contains K_{ab}, $n_{r,}$ n_A and is encrypted by the key K_a so that only user A can decrypt it. The message m3 contains K_{ab}, n_r, IDa and is encrypted by the key K_b so that only user B can decrypt it.

(c) On receiving m2 user A decrypts it with its private key K_a and checks whether n_A of the message match with the original to get confirmed that m2 is the reply for m1. If so, user A keeps the key K_{ab} with it for future use and sends a message m5 to user B. This message contains cipher text = E_{Kab} (n_r). User A also saves a copy of n_r. The message m5 indicates the readiness of user A. By this message user A also indicates to user B that it is in possession of the common key K_{ab} and is ready for secure communication with B.

(d) On receiving m3 user B decrypts it with its private key K_b and receives K_{ab}, n_r and ID_a. At this stage both the users have the same key K_{ab} that can be used for secure communication between them because no other user has this key. User B sends a message m4 to user A. This message contains cipher text= E_{Kab} (nr). User B also saves a copy of n_r. The message m4 indicates the readiness of user B. By this message user B also indicates to user A that it is in possession of the common key K_{ab} and is ready for secure communication with A.

(e) On receiving m4, user A decrypts it by K_{ab}, retrieves n_r and compares its value with the stored n_r value. If the values are equal then user A gets confirmed that user B is in possession of the common key K_{ab}. On receiving m5, user B also does the same thing.

5. Improvements on Needham-Schroeder Techniques

Origin Authentication: Existing Needham-Schroeder Key Distribution Technique does not provide Authentication of Originator's Identity. But, Proposed Technique provides Authentication of Originator's Identity by using encryption function on message m1. If IDA is changed by attacker, then KDC could recognize it by matching ID_A with the decrypted ID_A and terminates the communication.

Message Integrity: Existing Needham-Schroeder Key Distribution Technique does not provide Originator's Message Integrity. Since m1 is in plaintext form, any parameter in m1 may be altered by the attacker. But, Proposed Technique provides Originator's Message Integrity by using encryption function on message m1. Any alternation in m1 could be identified by matching IDA, IDB with the decrypted ID_A and decrypted ID_B respectively.

Key Freshness: Proposed technique provides Key Freshness property by creating and distributing Kab and Nr from KDC. But Key Freshness property is absent in the conventional technique.

Speed: In proposed technique, message m2 and message m3 can pass in parallel. Also, message m4 and message m5 can pass in parallel. Hence it is faster than the existing one.

6. Implementation

For implementation, C programming language is used. The three communicating entities in the proposed technique are: initiator (A), trusted server (KDC) and the responder (B). Each entity has the capability of creating messages; and several remote procedure calls are used to establish communication link and also for the purpose of message passing among entities. Running time in different machines is given in the following Table 1:

Table 1: Running Time in Different Machines

Processor's speed	Running time for Conventional technique (milliseconds)	Running time for proposed technique (milliseconds)
1.73 GHz	5055.555556 ms	3000.000000 ms
700 MHz	5111.111111 ms	3011.111111 ms
400 MHz	5611.111111 ms	3555.555556 ms

7. Comparative Analysis

Efficiency and Security Services provided by the proposed and Needham-Schroeder conventional technique is summarized in the Table 2. Three additional security services as originator's identity authentication, originator's message integrity and key-freshness are introduced in the proposed technique.

Table 2: Comparative Analysis between Proposed and Needham-Schroeder Conventional Technique

Security Services	Proposed Technique	Conventional Technique
(1) Authentication of Originator's Identity	√	×
(2) Originator's Message Integrity	√	×
(3) Originator's Message Freshness	√	√
(4) Key Freshness	√	×
(5) Key Authentication	√	√
(6) Key Confirmation	√	√
(7) Entity Authentication	√	√
(8) Efficiency (time)	3000.00000 milliseconds	5055.555556 milliseconds

8. Discussions

A session-key is a key used for encrypting one message or a group of messages in a single communication session. Security solutions require that the secret session-keys to be kept out of reach from the adversaries. When designing or selecting a key establishment technique for use, it is important to consider what assurances and properties an intended application requires. The fundamental security services of key distribution protocol are: authentication of the originator's identity, originator's message-integrity, originator's message-freshness, key authentication, entity authentication, key freshness and key confirmation. In the proposed technique, the replay attack of the original Needham-Schroeder five-message protocol is removed and three additional security services, such as: authentication of originator's identity, originator's message integrity and key-freshness are introduced (these three security services are absent in conventional Needham-Schroeder five- message protocol). Moreover, the time needed for distributing keys between pair of nodes in the proposed technique is reduced.

9. Conclusions

The motivation of the proposed work is to improve the security issues of the conventional Needham-Schroeder five-message protocol and to make the key distribution faster. From the above discussion, the following conclusions can be drawn:

- The proposed technique gives all the benefits (security services) that the conventional Needham-Schroeder five-message protocol can provide.
- It provides an additional authentication level in originator's identity.
- It enhances the security services by providing the integrity of the originator's message.
- It removes the replay attack by establishing key freshness security issue.
- It reduces the time needed to distribute session-key between a pair of entities and it is found that for all cases the proposed technique is faster than the conventional protocol.

It is concluded that the Proposed Key Distribution Technique will perform better than the conventional Needham-Schroeder protocol.

References

[1] Stallings, W. "Cryptography and Network Security Principles and Practice", Third edition, Prentice Hall Upper Saddle River, New Jersey 07458, ISBN: 981-403-589-0.

[2] Menezes, A., , Oorschot. P. van, and Vanstone S., "Handbook of Applied Cryptography", CRC Press, 1996.

[3] Otway D., and O. Rees. "Efficient and Timely Mutual Authentication". Operating Systems Review, 1987.

[4] Paulson, Lawrence C.. "Relations between Secrets: Two Formal Analyses of the Yahalom Protocol". J. Computer Security, 2001.

[5] Satyanarayanan M. "Integrating Security in a Large Distributed System". ACM Transactions on Computer Systems, 1989.

[6] Kaufman C., Perlman R., Speciner M., "Network Security: Private Communication in a Public World", 2nd Edition.

[7] D. Boneh, C. Gentry, B. L. and Shacham, H. "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps". In Eurocrypt '03, LNCS 2656, Springer-Verlag, 2003.

[8] Juels, A. and Brainard, J. G. "Client Puzzles: A cryptographic countermeasure against connection depletion attacks". In NDSS. The Internet Society, 1999. ISBN 1-891562-04-5, 1-891562-05-3.

[9] Kagal, L., Finin, T., Cost, R. S., and Peng, Y. "A Framework for Distributed Trust Management". In Second Workshop on Norms and Institutions in MAS, Autonomous Agents, May 2001.

[10] R. Kemmerer, C. Meadows, and J. Millen. "Three Systems for Cryptographic Protocol Analysis." Journal of Cryptology, 7(2):79–130, 1994.

[11] R. Needham and M. Schroeder. "Authentication Revisited". Operating Systems Review, January 1987.

[12] Goel, S., Robson, M., Polte, M., and Sirer, E. G. "Herbivore: A Scalable and Efficient Protocol for Anonymous Communication". Technical Report TR2003-1890, Cornell University Computing and Information Science, 2003.