www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 4 April 2017, Page No. 20831-20839
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i4.06

Diffie-Hellman Key Exchange Protocol with Entities Authentication

Om Pal [1*], Bashir Alam [2]
[1]Ministry of Electronics and Information Technology, Government of India
[2]Department of Computer Engineering, Faculty of Engineering & Technology,
Jamia Millia Islamia, New Delhi
[*]ompal.cdac@gmail.com

Abstract: The Diffie-Hellman key exchange protocol provides the opportunity to arrive at a common secret key by exchanging texts over insecure medium without meeting in advance. Diffie-Hellman key exchange protocol is limited to the exchange of key only. Due to lack of authentication of entities, this protocol is vulnerable towards man-in-middle attack and impersonation attack. To eliminate the man-in-middle attack, Nanli[9] presented a research paper on Diffie-Hellma key exchange protocol. It is observed that Nanli's protocol, still suffers with impersonation attack. To deal with this vulnerability, an improved key exchange approach based on third party authentication scheme is proposed in this paper.

1. Introduction

Diffie-Hellman key agreement protocol provides method for exchange of secret key between two parties without meeting in advance over an unprotected channel. Secret key is used for further encryption and decryption of message and cipher text, respectively. There are many challenges in respect to key management. First challenge for secure communication is that every pair of users should have unique key. Therefore, if a communication network has n users then every member of the network has to keep (n-1) keys. It becomes very hard to manage such a huge number of keys when the network has a large number of members. In public key cryptography the participant has to determine two different keys for encryption and decryption. These two keys should be multiplicative inverse of each other [1]. The main problem with public key cryptography is that encryption and decryption process are too slow [2]. Public key cryptography also suffers from man-in-middle attack where trusted third party plays the role of man in the middle. In man-in-middle attack, third party (C) impersonate itself into A to B and B to A . In this way, both parties talks to each other through third party C. The remedy to this problem is to use the method of authentication between communicating parties.

Many schemes [4, 5, 6, 7, 8] have been presented on Group Key Management. In these schemes, researchers discussed the recent trends of security parameters in key management, methods of distribution of session keys in secure manner and they also discussed about refreshment of keys. Multicast or group communication enables the distribution of content at large-scale, by providing an efficient mechanism for many-to-many and one-to-many communications.

One of the major challenges with symmetric key cryptography is to establish the secret key between two participants. Whitfield Diffie and Martin Hellman were the first persons to establish the feasible

approach to construct a shared secret over an insecure medium. This scheme provides the mechanism to key exchange only. This key exchange takes place in a certain mathematical setup where there is no user authentication [3]. Cas Cremers [14] presented the research paper on how to improve the ISO/IEC 11770 standard for key management.

As the Diffie- Hellman protocol suffers from man-in-middle attack, so it is necessary to devise the solution to eliminate the man-in-middle attack for secure transmission of the secret key between two parties. Many schemes have been presented [9, 10, 11,12,13] to deliver the key exchange with user authentication to eliminate the man in middle attack using hashing algorithms. Nan Li [9] proposed an enhanced version of the protocol which is based on the hash algorithm. In this paper, the approach proposed by Nanli[9] for eliminating man-in-middle attack has been discussed, the problem of impersonation by authenticated member is identified and a solution for the identified attack is proposed.

2. Diffie- Hellman key agreement protocol

The objective of Diffie Hellman key exchange [Fig 1] is to provide the opportunity to parties to create a symmetric session key over insecure medium. Further symmetric key is shared using session key for encryption and decryption of data. The strength of secret key generated in Diffie–Hellman protocol depends on discrete logarithm problem. Discrete logarithm problem as defined in reference [2] is the level of security which protect the deducing of the key. Let 'a' is a number and power modulo p of this number generates all integer from 1 to p-1, then 'a' is called a primitive root or generator of prime number 'p'. Generated numbers from 1 to p-1 are :

$a(\text{mod})p, a^2(\text{mod})p, \ldots\ldots\ldots a^{p-1}(\text{mod})p.$

These integers from 1 to p-1, make a form of permutation. For an integer b,{b:b<p}, prime number 'p' , and generator 'a'of prime number 'p' , 'b' is obtained as

$b=a^i(\text{mod})p$; where $0 \leq i \leq (p-1)$.

Here 'i' is called the problem of discrete logarithm of integer 'b' on base 'a' modulo p. This value ('i') can be denoted as $d.\log_{a,p}(b)$. As in reference[2], the key exchange protocol is described by using two public parameters which are known as a prime number given 'q' and an integer given as 'α'. The given integer 'α' is a generator of 'q'.

User A selects one time random number (private key) X_A , such that $X_A < q$ and computes public key parameter $Y_A= [\alpha^{(X_A)}]\text{mod } q$. In similar way, user B also selects a fresh random number as its private key and user B computes its public key as

$Y_B= [\alpha^{(X_B)}]\text{mod } q.$

The X value is kept as private at each side and Y value is made publicly available to other user. The key for user B is calculated as

$k= [(Y_A)^{(X_B)}]\text{mod } q.$

DOI: 10.18535/ijecs/v6i4.06

Above calculations compute the same key.

$K=[(Y_B)^{(X_A)}]\text{mod } q$

But $Y_B= \alpha^{(X_B)} (\text{mod})q$, therefore by putting the value of Y_B in above equation

$= [[(\alpha^{(X_B)})\text{mod } q]^{(X_A)}]\text{mod } q$

$= [\alpha^{(X_B)(X_A)}]\text{mod } q$

Now after applying the commutative property

$= [\alpha^{(X_A)(X_B)}]\text{mod } q$

$=[[[\alpha^{(X_A)}]\text{mod } q]^{(X_B)}]\text{mod } q$

But $Y_A= \alpha^{(X_A)} (\text{mod})q$, therefore

$=[(Y_A)^{(X_B)}]\text{mod } q.$

Hence both sides compute the same secret key. As X_A and X_B are private , an attacker have only the values Y_A ,Y_B ,α, and q to find the key. Therefore, an attacker has to solve the discrete logarithm problem to deduce the key.

An attacker needs to obtain $X_B=d.\log_{\alpha,q}(Y_B)$ to deduce the private key of user B

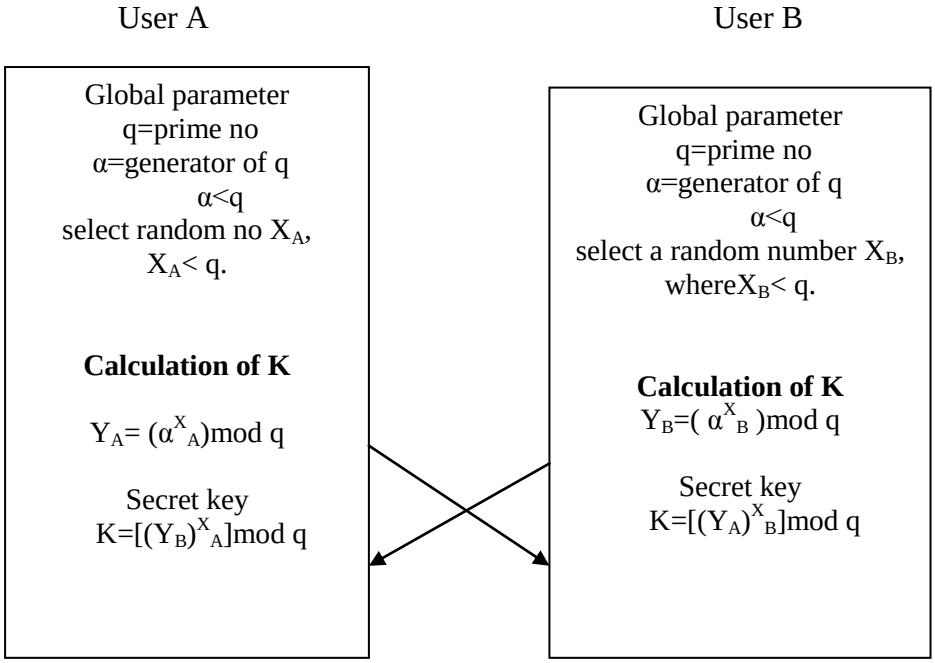

Fig 1: Diffie – Hellman key exchange protocol

3. Man-in-middle attack

Diffie-Hellman algorithm is vulnerable to man-in-middle attack [Fig 2], where third party (an attacker) adds himself between two communicating parties. Both parties think that they are communicating to each other, however, an attacker can listen their communication and can alter or modify the content of communication to his wish. In man-in-middle attack, third party 'C' impersonate himself A -to -B and B-to-A in such a way that both the parties end up in negotiating the key with C.

DOI: 10.18535/ijecs/v6i4.06

As in the Fig [2], user A sends it's public key Y_A to user B . The man-in-middle 'C' intercepts the communication and saves the public key of user 'A', calculates its own public key Yc and send it to the user 'B'. In a similar manner, user B sends it's public key Y_B to user A , the man in middle C intercepts the key ,saves it and sends it's public key Yc to user A. When user A sends message to user B encrypted by using key Yc, C intercepts the message since the message is actually encrypted by his own public key Y_C instead of B's public key, So he can decrypt it with his private key and he can make the modifications to the message at his own will, then C again encrypts the message with public key of B and sends it to B .When user B sends the message to user A, C applies the same approach as he applied in case of party A. In this way Mallory C is able to imitate B when talking to A, imitate A when talking to B. The fundamental reason that Diffie Hellman suffers from the man-in-middle attack is that user have no way to verify that they are talking to each other.

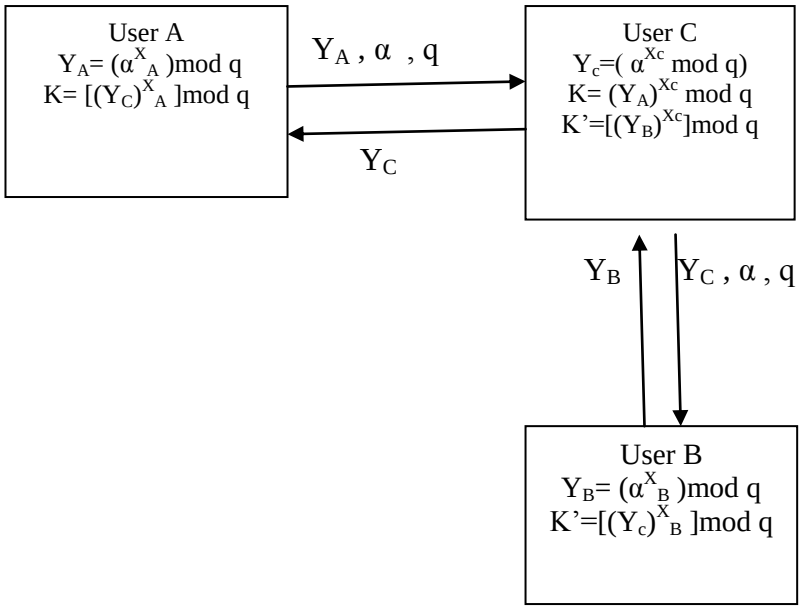

Fig 2: Man-in-middle attack

4. Impersonation Attack

In impersonation attack [Fig 3] the user A sends its public key Y_A to user B but before delivery of message to user B, it is intercepted by a third party C, middle-man C saves the public key of user A. C sends back its public key Yc to user A. In impersonation attack, the user B doesn't have any role in key agreement. In this way user A thinks that it has negotiated the key with user B.

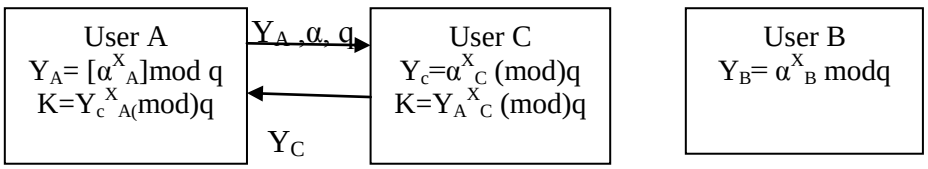

Fig 3: Impersonation Attack

5. Nanli's Key Agreement Protocol

Nanli [9] proposed an enhanced version of Diffie-Hellman key exchange protocol. To authenticate entities, author used hash functions along with key exchange parameters. In Nanli [9] protocol A and B are two parties which want to exchange the messages securely. The Nanli [9]' protocol proceeds in the following way:

Step1: A→AS, $ID_A||ID_B$

Step2: AS→A, $N_1 \oplus P_A$

Step3: AS→B, $N_1 \oplus P_B$

Step4: A→B,$Y_A||H(Y_A||N_1)$

Step5: B→A, $Y_B||\ H(Y_B||f(N_1))$

Step6: A→B, $H(N_1)$

Step7:

A calculates session key (K)= $(Y_B)^{(X_A)}(mod)q$

B calculates the same session key (K)= $(Y_A)^{(X_B)}(mod)q$

Where AS = Authentication server

ID_A=Identifier used for user A

ID_B=Identifier used for user B

P_A=Password of A

P_B=Password of B

N_1= Random number (one time)

$\oplus$=XOR operation

|| = Concatenation which is used for concatenation of two strings

H= A hash function, such as SHA-1 or MD5.

$Y_A= [\alpha^{(X_A)}]mod\ q$

$Y_B= [\ \alpha^{(X_B)}\ ]mod\ q$

f = Transformation function which is subtraction, addition or shift operation

Steps of Nanli's [9] proposed protocol (As shown in fig 4) are as follows:

In initial step user A sends a communication request to Authentication Server (AS) consisting ID(A)and ID(B).

In response to above message, AS sends a message 1 shown in fig 4 ($P_A \oplus N_1$) and ($P_B \oplus N_1$) to A and B, respectively. Now user A computes $N_1 \oplus P_A \oplus P_A$ and user B also computes $N_1 \oplus P_B \oplus P_B$. Hence, N_1 is shared between A and B.

DOI: 10.18535/ijecs/v6i4.06

User A selects the random Value X_A and calculate the public key Y_A and then A sends a message Y_A $||H(Y_A||N_1)$ to B. B calculates $H(Y_A||N_1)$ using received Y_A in step 4 and N_1 in step 3. If computed hash is equal to received hash then B believes that message is sent by A.

Similarly user B selects the random value XB and computes Y_B $||H(Y_B||f(N_1))$ and sends the values through message 5 to A. A calculates $H(Y_B||f(N_1))$ using received Y_B in step 5 and N_1 in step 2. If computed hash is equal to received hash then A believes that message is sent by B. Now A calculates the key $K=(Y_B)^{(X_A)}(\text{mod})q$ and user A sends the message $H(N_1)$ as a confirmation message to user B. B also calculates the key $K=(Y_A)^{(X_B)}(\text{mod})q$.

6. Cryptanalysis on Nanli's Protocol

In the Nanli approach [9] it is seen that only one hash comparison is done to check whether the message is fresh or not. But this does not give any information to destination about the identity of sender, so sender takes this advantage and sends the forge message to B using ID of some other user within the network say C. Here, record of communication is maintained at AS only and there is no provision to check the identity of sender (A) by the receiver (B) at the time of establishment of communication.

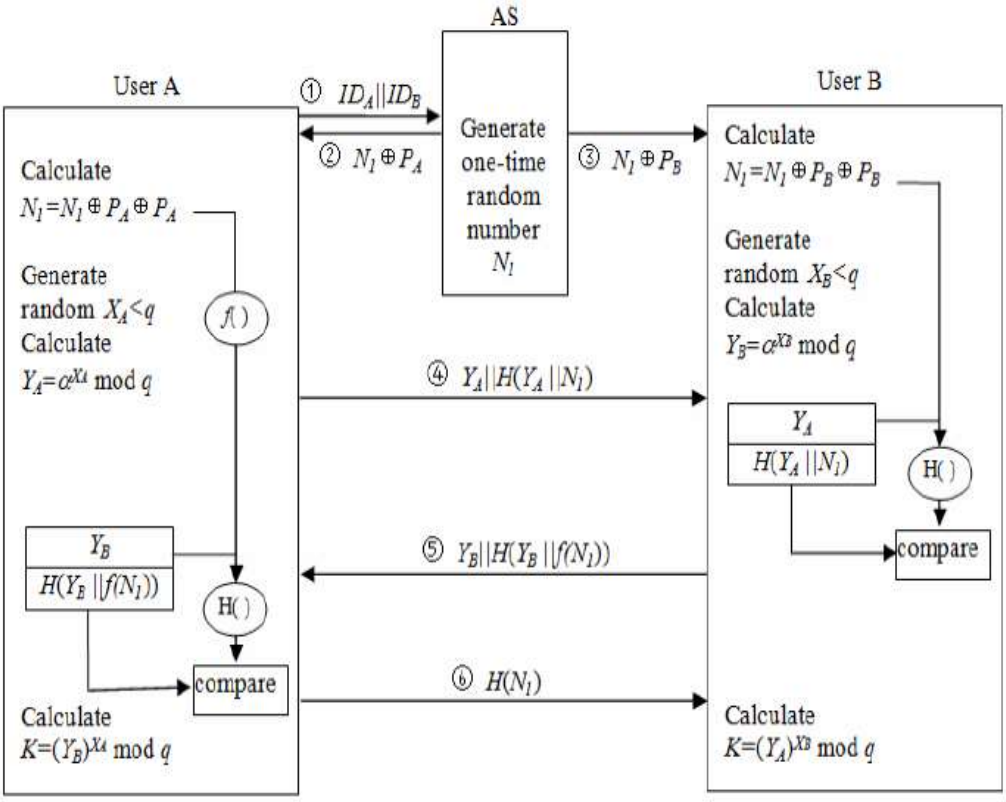

Fig 4: Diffie –Hellman key exchange approach proposed by Nanli [9].

7. The Proposed Key Agreement Protocol

In Nanli's protocol AS does not provide any information about the identity of the sender to the receiver, so an authenticated entity can impersonate another entity within the network. To eliminate the impersonation attack within the network, following protocol [Fig 5] is proposed:

DOI: 10.18535/ijecs/v6i4.06

Proposed Key agreement protocol-

1. User A→AS , $ID_A||ID_B$

2. AS →User A, $N_1 \oplus P_A$

3. AS →User B, $N_1 \oplus P_B$ and $H(N_1 \oplus P_B ||ID_A)$

4. User A →User B, $Y_A||H(Y_A||N_1)||ID_A$

5. User B → User A, $Y_B||H(Y_B||\ f(N_1))||ID_B$

6. User A→User B, $H(N_1)$

User A computes $K = (Y_B)^{(X_A)}(mod)q$

User B computes $K = (Y_B)^{(X_A)}(mod)q$

1. Authentication server sends a message $(N_1 \oplus P_B)$ to B along with the message digest $H(N_1 \oplus P_B ||ID_A)$. Here H(.) is an algorithm like MD5 which is used for the purpose of verification of sender's identity. At the receiver end ,receiver apply the same algorithm to calculate the message digest (hash value) using $(N_1 \oplus P_B)$ and ID_A, if the calculated value is same as received in step 3 then receivers confirmed that user A has not perform any impersonation. Detail of execution steps is given in next paragraph.

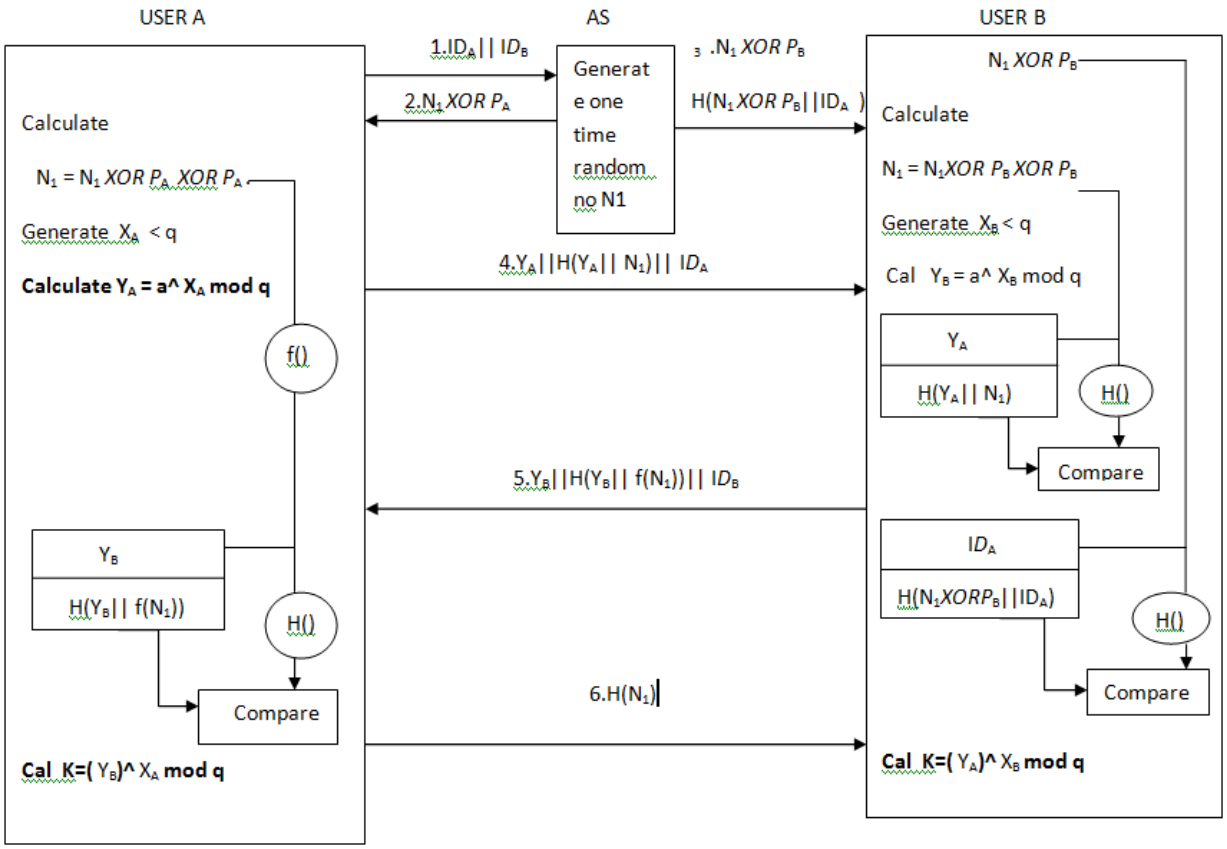

Fig 5: Proposed Key Agreement Protocol

2. Sender A sends its identity along with hash of nonce and public parameter Y_A to the requested intended destination B in step 4. The message is $Y_A||H(Y_A||N_1)||ID_A$. Now the destination user first calculates $H(Y_A||N_1)$ using Y_A received from A and Nonce received from AS.

If the hash value calculated is same as the hash value obtained from the user A then destination user confirms that message is fresh and no replay attack is done. Receiver then calculates $H(N_1 \oplus P_B || ID_A)$ through received $N_1 \oplus P_B$ from AS and ID_A from user A. If calculated value is same as the hash value received through AS then user B gets confirmation that it is communicating with the same user who have requested to AS for initiating the communication with B. User A also checks the freshness of the one time random number N_1, authenticates user B and computes the secrete key using the values received in step 5. Now, user A sends the green signal of successful connection establishment between user A and user B through message sent to B in step number 6. Finally both users compute the common secret key K in last step of the protocol.

8. Security Analysis

Here it can be seen that if user A intends to impersonate the user B then it will be rectified by user B in the following manner. User B computes $H(N_1 \oplus P_B || ID_A)$ and if computed value is different from the value sent by authentication server in step 3 then user B conceives that user is playing impersonation attack. Subsequently user B denies the communication with the sender.

9. Conclusion

As there is no provision of entity authentication in Diffie- Hellman key exchange protocol Diffie-Hellman protocol is vulnerable to man-in-middle attack and impersonation attack. Nanli[9] presented a research paper on Diffie-Hellman key exchange protocol to eliminate the man-in-middle attack. After analysing the approach suggested by Nanli [9] for eliminating man-in-middle attack in Diffie – Hellman Key exchange protocol, it is found that impersonation attack still exists. Therefore an improved approach is proposed in this paper. By doing two comparisons of hash values in proposed approach, impersonation attack which is present in Nanli's approach is successfully eliminated. With inclusion of authentication mechanism along with two hash comparisons at the destination side, it eliminates the replay attack, man-in-middle attack and impersonation attack successfully.

10. References

[1] Spyros S Magliver "Secure group communication over data network" 2005 springer science + Business media Inc ,10-11.

[2] William Stallings "Cryptography and Network Security ,Principles and Practices" (Book) Fourth Edition,Pearson Education .

[3] Mahender Kumar, C.P. Katti, P.C. Saxena, "An ID-based Authenticated Key Exchange Protocol", International Journal of advanced studies in Computer Science and Engineering IJASCSE Volume 4 Issue 5, 2015.

[4] R. Siva Ranjani, D. Lalitha Bhaskari, P.S. Avadhani, "Current Trends in Group Key Management", International Journal of Advanced Computer Science and Applications, Vol. 2, No. 11, Nov 2011.

[5] Saravanan, K., T. Purusothaman, "Efficient Star Topology based Multicast Key Management Algorithm", Journal of Computer Science 8 (6), Year 2012.

[6] Michael Steiner, Gene Tsudik, Michael Waidne, "Diffie-Hellman Key Distribution Extended to Group Communication", IBM Ziirich Research Laboratory CH-8803 Riischlikon, Switzerland.

[7] Antoine Joux, "A one round protocol for tripartite diffie-hellman", In Proceedings of the 4th International Symposium on Algorithmic Number Theory, pages 385–394. Springer-Verlag, 2000.
[8] Iuon-Chang Lin, Shih-Shan Tang, and Chung-Ming Wang, "Multicast Key Management without Rekeying Processes", The Computer Rekeying Processes", The Computer Journal, Vol. 53 No. 7, 2010.
[9] Nan Li, "Research on Diffie – Hellman Key, Exchange Protocol", IEEE 2nd International Conference, on Computer Engineering and Technology, Vol. No 4, pp 634 – 637, 2010.
[10] Ik Rae Jeong, Jeong Ok Kwon, Dong Hoon Lee, "Strong Diffie-Hellman-DSA Key Exchange", IEEE Journals and magazines , pp. 432 – 433, 2007.
[11] Harn, L., and Lin, H.-Y. "An authenticated key agreement without using one-way hash functions". Proc. 8th Nat. Conf. on Information Security, Kaohsiung, Taiwan, pp 155–160, 1998.
[12] D. Liu, P. Ning, Establishing Pairwise Keys in Distributed Sensor Networks, 10th ACM CCS '03, Washington D.C., October, 2003.
[13] Chen Hao and Guo Yajun, "A Key Agreement Scheme Based on Bilinear Pairing for Wireless Sensor Network", Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, pp 384-388, 2009.
[14] Cas Cremers, Marko Horvat, "Improving the ISO/IEC 11770 standard for key management techniques", International Journal of Information. Security, Springer, November 2015.