ISSN 0974 - 9373

Vol. 16 No.1 (2012) **Journal of International Academy of Physical Sciences** pp. 75-80

A Key Confirmation Based Key Agreement Protocol over the Diffie-Hellman Key Exchange Protocol

K. K. Dubey and Gaurav Agarwal
Invertis University, Bareilly, India
Email: kamlesh778@gmail.com; gauravagarwal95@gmail.com

(Received December 23, 2011)

Abstract. Diffie Hellman key exchange algorithm is the most famous algorithm to exchange keys over a network but it has some false and drawbacks. So in our work we have proposed a new agreement protocol based on key confirmation as well as Diffie Hellman algorithm. This protocol also works on the elliptic curve cryptography in asymmetric encryption.
Keywords: Key agreement, Diffie-Hellman protocol, Public key cryptography.

1. Introduction

For the establishment of the communication the parties requires the session key which can be generated by the key establishment protocol and it can also be refers as the key agreement protocol. For the key agreement the first famous protocol developed by Diffie and Hellman which was based on asymmetric encryption or public key cryptography[1]. They proposed the two versions of protocol. In the first protocol all the entities in the communication network exchange the static public keys in this case there is a drawback that the entities A and B computes the same session key for each run of protocol, and in the second case they exchange the ephemeral public keys which is vulnerable to the man-in-middle attack. For overcoming of these situation a authenticated key agreement is proposed, which is the combination of both static and ephemeral versions of two entities A and B which meets all security attributes.

1.1 Introduction to Diffie-Hellman Key Agreement Protocol. The Diffie-Hellman[2] proposed a cryptosystem which is based upon the difficulty of finding discrete logarithm in field. For this protocol we need to know that two publicly known numbers i.e. p and g which will be the primitive root of p. Suppose there are two users A and B which want to exchange a key and

unknown to each other. First time user A select a random Integer X_A <p and compute $Y_A = g^{XA} \bmod p$. Similarly, user B selects a random number X_B <p and computes $Y_B = g^{XB} \bmod p$. Each side keeps the value of X privately and make the Y value available publicly to the other side so this is called public key for both of users A and B and the process is known as the public key generation for A and B. Now user A computes the key as $K= (Y_B)^{XA} \bmod p$ and user B computes the key $K= (Y_A)^{XB} \bmod p$. These two values should produce similar result.

$$\begin{aligned} K &= (Y_B)^{X_A} \bmod p \\ &= (g^{X_B} \bmod p)^{X_A} \bmod p \\ &= (g^{X_B})^{X_A} \bmod p \end{aligned}$$

by the rule of modular arithmetic.

$$\begin{aligned} K &= g^{X_B X_A} \bmod p \\ &= (g^{X_A})^{X_B} \bmod p \\ &= (g^{X_A} \bmod p)^{X_B} \bmod p \\ &= (Y_A)^{X_B} \bmod p. \end{aligned}$$

The result shows that the two sides can change the secret value and both values are identical with each other.

2. Proposed System

The proposed system works as follows there are the following notations which we have used in the protocol.

A,B	Entities
ID_A, ID_B	Identity parameter of A,B
G	Generator Point
$E_k(x)$	Encryption of x using the key k
$D_k(x)$	Decryption of x using the key k
KR_A	Static private key of A
KU_A	Static public key of A which is

	elliptic curve point i.e. $KR_A.G$
r_A	A's ephemeral key (Random no.)
K	Session key between entities
A→B: M	Sending of message M From A to B
Sgn_A	Signature using private key of A
SK	Session key between A and B

For B we have similarly KR_B, KU_B, and r_B
The proposed protocol works on the domain parameter of elliptic curve that are common to both entities and cover an elliptic curve E defined over a field Fq which is generating a point G of elliptic curve cryptography so that G belongs to E (Fq), n is order of G in E (Fq), and h is cofactor of n.

2.1. The Protocol for the system. For the establishment of any session the entities need a session key and for sharing of a session key they should know the public key of each other this process can be done by the certificate authority[3] which provides CA_A i.e. A's certificate congaing the public key and the signature of A. the proposed protocol will work as follows.

2.1.1. The communicating entities will take the public keys of each other with the help of certificate authority. Now A will have KU_B and B will have KU_A.

2.1.2. The session key K will be generated by using the KR_A and KU_B as $K = KR_A.KU_B = KR_A.KR_B.G$

2.1.3. In the next step a select a random number r_A as its ephemeral key and computes a point on elliptic curve $M_A = r_A\, KU_B$. After encryption of signed message with K the result is like.

$$A \rightarrow B: ID_B, E_K(M_A, Sgn_A\,(ID_B, KU_A, KU_B))$$

2.1.4. With the same process like A, B will also find the value of K, and decrypts the message received from A, recovers M_A and verify the signature sent by A.B will select again a random number r_B as its ephemeral key and calculate the session key SK = h(r_B KU_A + MA).If SK = 0 then B can terminate the protocol. Otherwise B will send a message to A as A did in previous step. i.e.

$$B \rightarrow A: E_K (M_B), E_{SK} (Sgn_B (ID_A, M_A, M_B))$$

2.1.5. After receiving the message from B, A decrypts with K to recover M_B. The session key will be computed again with the help of KU_B and M_B if SK = 0 then A will terminate the protocol otherwise a message will be sent to B.

$$A \rightarrow B: E_{SK} (Sgn_A (ID_A, M_A, M_B))$$

2.1.6 In the last step B will decrypt the received message using SK and verify the signature created by A. if the signature verified then B will store the session key SK. The multiplication by h in SK will ensure that the session key SK is a point in the subgroup of order n in E (Fq) to protect against small subgroup attack[4].

3. Result

3.1. Proof of correctness. The protocol works correctly for the session can be shown as follows

For A

$$SK = h (r_A.KU_B + M_B)$$

$$= h (r_A.KU_B + r_B.KU_A)$$

$$= h (r_A.KR_B.G + r_B.KU_A.G)$$

$$= h (r_A.KR_B + r_B .KR_A).G$$

$$= h (r_B.KR_A + R_A.KR_B).G$$

= Which is the SK of B.

For B

$$SK = h\,(r_B.KU_A + M_A)$$

$$= h\,(r_B.KU_A + r_A.KU_B)$$

$$= h\,(r_B.KR_A.G + r_A.KU_B.G)$$

$$= h\,(r_B.KR_A + r_A.KR_B).G$$

$$= h\,(r_A.KR_B + r.KRA).G$$

= Which is the SK of A.

3.2. Security Analysis. Security issue of proposed protocol works on the Diffie-Hellman problem in ECC. The proposed protocol provides known-key security because each run of the protocol between A and B should produce a unique session key which depends on r_A and r_B. The proposed protocol also provide the prevention against the meet- in- middle attack[5] in which an attacker fools both the communication parties in a legitimate conversation by creating two private, public key pairs. In the proposed protocol an attacker cannot forget the private keys of entities to create the signature. If it is possible then the signature will not be verified because of certificates provided by certificate authority.

4. Conclusion

In this paper we have presented a new key agreement protocol based on the key confirmation. The protocol is designed to provide the desirable security attributes which are not provided by other key agreement protocol like Diffie-Hellman protocol and Unified Model etc. The security analysis of protocol has been proposed against the different types of attack and the correctness is also provided for the proposed protocol.

References

1. W. Diffie and M. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, **IT-l 22(6)** (1976) 644-654.

2. E. Rescorla, Diffie-Hellman Key Agreement Method, RFC 2631, IETF Network Working Group, http://www.ietf.org/rfc/rfc2631.txt.

3. Curry, Ian, Entrust Technologies, Getting Acquainted With Entrust/Solo and Public-key Cryptography, version 1.0, July 1997

4. N. Howgrave-Graham and N. Smart, Lattice attacks on digital signa- ture schemes, Designs, *Codes and Cryptography*, **23** (2001) 283-290.

5. Bon Wook Koo, Hwan Seok Jang and Jung Hwan Song, *Constructing and Crypt-analysis of a 16 £ 16 Binary Matrix as a Di®usion Layer*. In K. Chae and M. Yung (Eds.): WISA 2003, LNCS 2908, pp.489-503, Springer-Verlag 2004.

6. A. Lenstra and E. Verheul, Selecting Cryptographic Key Sizes, *Journal to Cryptology*, **14** (2001) 255 – 293, http:/www.cryptosavvy.com/.

7. A. Chandrasekar and V. R. Rajasekar, *Improved Authentication and Key Agreement Protocol Using Elliptic Curve Cryptography*.

8. NIST, Special Publication 800-57: Recommendation for Key Management. Part 1: General Guideline, Draft Jan. 2003.