Volume 1, No. 2, July-August 2010

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

New Authenticated Key Agreement Protocol with Key Confirmation

Pritiranjan Bijayasingh*
Department of Computer Science and Engineering
Balasore College of Engineering and Technology
Balasore, Orissa, India – 756 060
pritiprbs@yahoo.com

Asis Kumar Tripathy
Department of Information Technology
NM Institute of Engineering and Technology
Bhubaneswar, Orissa, India – 751 019
asistripathy@gmail.com

Biswanath Sethi
Department of Computer Science and Engineering
Balasore College of Engineering and Technology
Balasore, Orissa, India – 756 060
biswanath_bcet@yahoo.co.in

Abstract: Diffie-Hellman key agreement protocol is the first and most famous protocol, but it has many flaws and drawbacks. Therefore, in this paper a new authenticated key agreement protocol (AKAP) with key confirmation is proposed. The proposed protocol is based on Diffie-Hellman problem and it is working over elliptic curve group in the setting of asymmetric techniques.

Keywords: Diffie-Hellman protocol; ephemeral key; man-in-the-middle attack; public key certificates; static public key.

I. INTRODUCTION

Key agreement refers to one form of key establishment protocols in which two or more users execute a protocol to securely share a session key. The most famous protocol for key agreement was proposed by Diffie and Hellman which is based on concept of public-key cryptography [1]. There are two versions of the Diffie-Hellman protocol namely static and ephemeral. In the first one, the entities exchange static public keys, and in the second, the entities exchange ephemeral public keys. Therefore, the static protocol has a major drawback, is that the entities A and B compute the same session key for each run of the protocol. Also the ephemeral Diffie-Hellman protocol is vulnerable to a man-in-the-middle attack. To overcome these security flaws, a new authenticated key agreement protocol is proposed, which is a hybrid approach of both static and ephemeral version. The established session key is formed as combination of static and ephemeral private keys of two entities A and B. The discussion shows the present protocol meets all security and efficiency attributes.

II. PROPOSED PROTOCOL

A. *Domain Parameters*

The domain parameters for the protocol proposed are the elliptic curve parameters that are common to both entities and consist of an elliptic curve E defined over a finite field F_q, generating element G (point) of ECC $G \in E(F_q)$, n is order of G in $E(F_q)$, and h is cofactor of n, i.e., $h = \#E(F_q).n$.

B. *Notations*

The following notations are used in the paper.

Table I: Notations used in this Paper

A, B	Honest Hosts
ID_A, ID_B	Identities of A and B
$E_K(X)$	Encryption of plaintext X using key K
$D_K(X)$	Decryption of plaintext X using key K
G	Generator Point
x_A	*A*'s static private key, is an integer $\in R[1, q-1]$
Y_A	*A*'s static public key, is the elliptic curve point $= x_A.G$
r_A	*A*'s ephemeral key (random number in Z_n)
SK	Session Key between A and B
K	Static Session key
A→B: M	A sends message "M" to B
$sgn_A(.)$	Signature generated using the private key of A

Here, A has x_A, Y_A and r_A. Similarly, *B* has x_B, Y_B and r_B.

Both the entities want to share the session key should have the authentic copies of the static public keys of each other. This can be done using the public certificates issued by the certification authority (CA) as in the case of public-key cryptography [12]. $Cert_A$ denotes A's public-key certificate, containing her static public key Y_A, and a certifying authority CA's signature.

C. *The Protocol*

In this section, the detail flow of the proposed protocol is discussed.

- Both the communicating parties *A* and *B* recover the authentic copies of each other's static public keys from the publicly available certificates. As a result, *A* now has Y_B and B has Y_A.
- *A* computes the static session key *K* using x_A and Y_B as $K = x_A.Y_B = x_A.x_B.G$. Next, *A* chooses a random integer $r_A \in R[1, n-1]$ as her ephemeral key and computes the point $M_A = r_A.Y_B$. *A* then concatenates ID_B, Y_A and Y_B, signs the concatenated result with her static private key x_A, encrypts the M_A along with the signed message using *K* and sends it to *B*.

$$A \rightarrow B : ID_B, E_K(M_A, \text{sgn}_A(ID_B, Y_A, Y_B))$$

- *B* also finds the static session key *K* using x_B and Y_A as $K = x_B.Y_A = x_B.x_A.G$. Upon receiving the message from *A*, it decrypts using *K*, recovers M_A and verifies the signature sent by *A*. Next, *B* selects a random number $r_B \in R[1, n-1]$ as his ephemeral key and calculates the intended session key $SK = h(r_B.Y_A + M_A)$. If the calculated $SK = 0$, then *B* terminates the protocol. Otherwise, *B* computes $M_B = r_B.Y_A$. Then, it concatenates ID_A, M_A and M_B. *B* then signs the result with x_B and encrypts the signed result using the session key *SK*. *B* sends this encrypted value along with the encrypted value of M_B using *K* to *A*.

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

- After receiving the message from *B*, *A* decrypts with *K* to recover M_B. Next, she computes the intended session key $SK = h(r_A.Y_B + M_B)$. If this computed $SK = 0$, then *A* terminates the protocol. Otherwise, A concatenates ID_B, M_A and M_B, signs the result using x_B. Then she encrypts the signed result using the session key *SK* and sends to *B*.

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

- Finally, *B* decrypts the received message using *SK* and verifies the signature created by *A*. If the signature is verified by *B* correctly, he keeps the session key *SK*.

Multiplication by *h* ensures that the session key *SK* is a point in the subgroup of order *n* in $E(F_q)$ to protect against small subgroup attack as described in [3].

The small subgroup attack can be launched if the order *n* of the base point *G* is not prime; say, $n = m.t$ where $t > 1$ is small. The attacker forces the shared secret key to be one of small and known subset of points. If *SK* lies in the subgroup of order *t* of the group generated by *G*, then the attacker tries only *t* possible to find the key *SK*. The check $SK = 0$ ensures that *SK* is a finite point.

D. Correctness

The protocol correctly establishes an intended session key and it can be shown as follows:

Alice

$$\begin{aligned} SK &= h(r_A.Y_B + M_B) \\ &= h(r_A.Y_B + r_B.Y_A) \\ &= h(r_A.x_B.G + r_B.x_A.G) \\ &= h(r_A.x_B + r_B.x_A).G \\ &= h(r_B.x_A + r_A.x_B).G \\ &= SK \text{ of Bob} \end{aligned}$$

Bob

$$\begin{aligned} SK &= h(r_B.Y_A + M_A) \\ &= h(r_B.Y_A + r_A.Y_B) \\ &= h(r_B.x_A.G + r_A.x_B.G) \\ &= h(r_B.x_A + r_A.x_B).G \\ &= h(r_A.x_B + r_B.x_A).G \\ &= SK \text{ of Alice} \end{aligned}$$

III. SECURITY ANALYSIS

The security of proposed protocol is based on the Diffie-Hellman problem in elliptic curve group (ECDHP): given an elliptic curve *E* defined over a finite field F_q, a base point $G \in E(F_q)$ of order *n* and two points generated by *G*, $x.G$ and $y.G$ (where x and y are integer), find $x.y.G$. This problem is closely related to the well-known elliptic curve discrete logarithm problem (ECDLP) (given $E(F_q)$, *G*, *n* and $x.G$, find *x*).

The proposed protocol meets the following desirable security attributes.

E. Known-Key Security

A protocol is said to be vulnerable to a known-key attack if compromise of past session keys allows either a passive adversary to compromise future session keys, or impersonation by an active adversary in the future. The protocol should still achieve its goal in the face of an adversary who has learned some other session keys.

The proposed protocol provides known-key security. Each run of the protocol between two entities *A* and *B* should produce a unique session key which depends on r_A and r_B. Although an adversary has learned some other session keys, he can't compute $r_A.x_B.G$, and $r_B.x_A.G$ from them, because he doesn't know ephemeral private keys r_A and r_B. Therefore the protocol still achieves its goal in the face of the adversary.

F. Perfect Forward Secrecy

A protocol is said to have perfect forward secrecy if compromise of long-term keys does not compromise past session keys.

The proposed protocol also possesses forward secrecy. Suppose that static private keys x_A and x_B of two entities are compromised. However, the secrecy of previous session keys established by honest entities is not affected, because an adversary who captured their private keys x_A or x_B should extract the ephemeral keys r_A or r_B from the information M_A and M_B to know the previous or next session keys between them. However, this is the Elliptic Curve Discrete Logarithm Problem (ECDLP).

G. Key-Compromise Imporsanation

When *A*'s static private key is compromised, it may be desirable that this event does not enable an adversary to impersonate other entities to *A*.

Suppose A's long-term private key x_A, is disclosed. Now an adversary who knows this value can clearly impersonate A. But he can not impersonate B to A without knowing B's long-term private key x_B. For the success of the impersonation, the adversary must know A's ephemeral key r_A at least. So, also in this case, the adversary should extract the value r_A from $M_A = r_A.Y_B$, to generate the same key SK, with A. This also comes to ECDLP.

H. Unknown Key-Share

Entity *B* cannot be coerced into sharing a key with entity *A* without *B*'s knowledge, i.e., when *B* believes the key is shared with some entity $C \neq A$, and *A* correctly believes the key is shared with *B*.

The proposed protocol also prevents unknown key-share. According to the assumption of this protocol that the Certification Authority (CA) has verified that *A* possesses the static public key Y_A corresponding to her static private key x_A. An adversary cannot register *A*'s public key Y_A as its own and subsequently deceive *B* into believing that *A*'s messages are originated from the adversary. Therefore, *B* cannot be coerced into sharing a key with entity *A* without *B*'s knowledge.

I. Man-in-the-Middle Attack

In this attack, an attacker fools both the communicating parties in a legitimate conversation by creating two private, public key pairs: One between the first party and attacker, the other between the attacker and the second party.

The proposed protocol also prevents man-in-the-middle attacks. An attacker cannot forge the static private keys of either *A* or *B* to create the signatures. If the attacker successfully does so, then the signatures cannot be verified using the static public keys of the entities defined in the certificates. It is because the certificates are issued by the Certification Authority (CA).

IV. COMPARISON

Some modern key agreement protocols such as MTA/A0, MQV, LLK, Unified Model and Song-Kim are compared with proposed protocols from the security and efficiency point of view [3]-[9]-[10]-[11].

A. Security

From the security point of view, the proposed protocol provides more desirable security attributes than other AK protocols. For example, the MTI/A0 does not provide implicit key authentication (IKA) and FS and the AKC Unified Model does not support K-CI, while the AKC MQV provides UK-S which the AK MQV doesn't exhibit. So, Proposed AKAP Protocol provides all security attributes as well as the AKC MQV, LLK, and Song-Kim [3]-[10]-[11].

B. Efficiency

In Table I, the number of scalar multiplications required in each protocol is compared. Protocols MTI/A0, Unified Model, Song-Kim and Proposed protocol commonly require 2 scalar multiplications. The MQV Protocol requires 2.5, and the LLK protocol requires two scalar multiplications only.

Table II: Scalar Multiplications Required per Entity

Protocols	*Scalar Multiplications*
MTI/A0	3
LLK	2
Unified Model	3
MQV	2.5
Song-Kim	3
Proposed protocol	2

V. CONCLUSION

In this paper a new authenticated key agreement with key confirmation protocol (AKAP) is proposed. The protocol have been designed to provide the desirable security attributes which are not provided by the other security protocols such as MTI/A0, two-pass Unified Model, and Diffie-Hellman protocol. The proposed protocol is discussed and compared with other reported modern key agreement protocols. However, the results have been shown better security attributes than the currently reported protocols.

VI. REFERENCES

[1] W. Diffie and M. Hellman, "New directions in **cryptography**", IEEE Transactions on Information Theory, Vol. IT-l 22, No.6, November,1976, PP.644-654.

[2] A. Menezes, P. van Oorschot and S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997.

[3] L. Law, A. Menezes, M. Qu, J. Solinas and S. Vanstone, "An Efficient Protocol for Authenticated Key Agreement Protocol", Technical report CORR 98-5, University of Waterloo, Canada, March 1998.

[4] S. Blake-Wilson and A. Menezes, "Unknown Key-Share Attacks on the Station-To-Station (STS) Protocol", Technical report CORR 98- 42, University of Waterloo, 1998.

[5] S. Blake-Wilson and A. Menezes, "Authenticated Diffie-Hellman Key Agreement Protocols", Proceedings of the 5th Annual Workshop on Selected Areas in Cryptography (SAC '98), LNCS 1556, 1999, Springer- Verlag, pp.339-361.

[6] S. Blake-Wilson, C. Johnson and A. Menezes, "Key Agreement Protocols and their Security Analysis", Proceedings of the sixth IMA International Conference on Cryptography and Coding, LNCS 1355, Springer-Verlag, 1997, pp.30-45.

[7] W. Diffie, P. van Oorschot and M. Wiener, "Authentication and authenticated key exchanges", Designs, Codes and Cryptography, 2 (1992), pp.107-125.

[8] M. Bellare and P. Rogaway, “Entity Authentication and Key Distributions", Advances in Cryptology - Crypto '93, LNCS 773, Springer-Verlag, 1994, pp.232-249.

[9] M. Just and S. Vaudenay, “Authenticated Multi-Party Key Agreement", Advances in Cryptology, Asiacrypt '96, LNCS 1163, Springer-Verlag, 1996, pp.36-49.

[10] C. Lee, J. Lim, and J. Kim, “An Efficient and Secure Key Agreement", IEEE p1363a draft, 1998.

[11] B. Song and K. Kim, “Two-Pass Authenticated Key Agreement Protocol with Key Confirmation", Progress in Cryptology – Indocrypt 2000, LNCS 1977, Springer-Verlag, Dec. 2000, pp.237-249.

[12] Kohnfelder L. "Towards a Practical Public-Key Cryptosystem". Bachelor's Thesis, M.I.T, May 1987.