A NEW TWO-PASS KEY AGREEMENT PROTOCOL

Khaled Al_Sultan[1], Magdy Saeb[2], Medhat Elmessiery[3], and Usama Abd El-Raouf Badawi[1]

Abstract - *Diffie-Hellman key agreement protocol is the first and most famous protocol, but it has many flaws and drawbacks. Therefore, this paper proposes a new two-pass authenticated key agreement protocol (AK) and extent its capabilities to support key confirmation as a three-pass authenticated key agreement with key confirmation protocol (AKC). The present protocols are based on Diffie-Hellman problem and it is working over elliptic curve group in the setting of asymmetric techniques.*

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.

In order to counter these weaknesses, a new authenticated key agreement protocol is introduced in this paper. The important feature of the proposed protocol is 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.

DESIRABLE SECURITY AND EFFICIENCY ATTRIBUTES OF AK AND AKC PROTOCOLS

Desirable security attributes of AK and AKC protocols are as follows [2]-[3]-[5].

- **Known-Key Security (K-KS)***:* A protocol should still achieve its goal in the face of an adversary who has learned some other session keys. The session key is a unique secret key which in each run of a key agreement protocol between A and B is produced.
- **Forward Secrecy (FS)***:* If static private keys of one or more entities are compromised, the secrecy of previous session keys is not affected.
- **Key-Compromise Impersonation (K-CI)***:* 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.
- **Unknown Key-Share (UK-S)***:* 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 $C \neq \mathrm{A}$, and A correctly believes the key is shared with B.

In addition to these security attributes, it would be desirable for a protocol to have low computational cost and low communication overhead for its practical use.

P RPPOSED KEY AGREEMENT PROTOCOLS

The domain parameters for the protocols described in this section 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., $h = \#E(F_q)/n$.

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 $Y_a = Gx_a$

r_a : A's ephemeral key (random number in Z_n)

Similarly, B has x_b, Y_b and r_b.

We will assume that static public keys are exchanged via certificates. $Cert_A$ denotes A's public-key certificate, containing her static public key Y_a, and a certifying authority CA's signature.

The security of proposed protocol in this paper 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, $x\,\mathrm{G}$ and $y\,\mathrm{G}$ (where x and y are integer), find $xy\,G$. 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).

[1] Dept. of Mathematics, Faculty of Science, Cairo University, Cairo, Egypt. kh_sultan2001@hotmail.com
[2] Dept. of Computer Engineering, Arab Academy for Science, Technology and Maritime Transport. Alexandria, Egypt
[3] Dept. of Math & physics, Faculty of Engineering, Cairo University, Cairo, Egypt

Authenticated Key Agreement Protocol (AK)

In this section we describe a proposed two-pass key agreement protocol between two entities A and B. The protocol works in the following steps:

1) A and B obtain authentic copies of each other's static public key, Y_a and Y_b. If A and B do not a priori possess authentic copies of each other's static public keys, then certificates should be included in the flows.

2) A generates a random integer $r_a \in R[1, n-1]$, computes the point $M_A = r_a \mathrm{Y_b}$, and sends it to B.

3) B receives M_A from A, generates a random integer $r_b \in R[1, n-1]$, computes the session key $K = h(\frac{r_b}{x_b} M_A + x_b Y_a) = h(r_a r_b + x_a x_b)G$. If the $K = O$, then B terminates the protocol run with failure. Otherwise he computes $M_B = r_b Y_a$ and sends it to A.

4) A receives M_B from B and computes the session key $K = h(\frac{r_a}{x_a} M_B + x_a Y_b) = h(r_a r_b + x_a x_b)G$. If the $K = O$, then A terminates the protocol run with failure, thus the session key is the point $K = h(r_a r_b + x_a x_b)G$.

Multiplication by h ensures that the session key K 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, $n = mt$ where $t > 1$ is small. The attacker forces the shared secret key to be one of small and known subset of points. If K lies in the subgroup of order t of the group generated by G, then the attacker tries only t possible to find the key K.

The check $K=O$ ensures that K is a finite point

Security Consideration

Here we prove our protocol meets the following desirable security attributes.

Known-Key Security: The 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 $r_a r_b G$, and $x_a x_b G$ from them, because he doesn't know ephemeral private keys r_a and r_b. Therefore the protocol still achieve its goal in the face of the adversary.

Perfect Forward Secrecy: It 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)

Key-compromise Impersonation: 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 impersonates B to A without knowing the 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, K, with A. This also comes to ECDLP.

Unknown Key-Share: It also prevents unknown key-share. According to the assumption of this protocol that CA has verified that A possesses the private key x_a corresponding to her static public key Y_a, an adversary can't 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.

Authenticated Key Agreement with Key Confirmation (AKC)

In this section we extended the proposed protocol as a three-pass key agreement which support key confirmation. AKC protocol is derived from AK protocol by adding the $\mathrm{MAC_s}$ of the flow number, identities, and the M_A, and M_B.

Here, MAC is a message authentication code algorithm such as HMAC and it is used to provide key confirmation. H_1 and H_2 are key derivation functions. These steps describe the protocol as follows:

1) A generates a random integer $r_a \in R[1, n-1]$, computes the point $M_A = r_a Y_b$, and sends it to B.

2)

(a) B receives M_A from A, generates a random integer $r_b \in R[1, n-1]$,computes

$K = h(\frac{r_b}{x_b} M_A + x_b Y_a) = h(r_a r_b + x_a x_b)G$

If the $K = O$, then B terminates the protocol run with failure. Otherwise, the shared secret is the point K.

(b) B computes the point $M_B = r_b Y_a$

(c) B uses the x-coordinate x of the point K to compute two shared keys $k = H_1(x)$ and $k' = H_2(x)$.

(d) B computes $MAC_{k'}(2, B, A, M_B, M_A)$ and sends this together with M_B to A.

3)

(a) A receives M_B and $MAC_{k'}(2, B, A, M_B, M_A)$ from B and computes $K = h(\frac{r_a}{x_a} M_B + x_a Y_b) = h(r_a r_b + x_a x_b)G$ If the $K = O$, then A terminates the protocol run with failure.

(b) Otherwise, A uses the x-coordinate x of the point K to compute two shared keys $k = H_1(x)$ and $k' = H_2(x)$.

(c) A computes $MAC_{k'}(2, B, A, M_B, M_A)$ and verifies that this equals what was sent by B.

(d) A computes $MAC_{k'}(3, A, B, M_A, M_B)$ and sends this to B.

4) B computes $MAC_{k'}(3, A, B, M_A, M_B)$ and verifies that this equals what was sent by A.

5) The session key is k.

AKC Protocol is derived from AK Protocol by adding key confirmation to the latter. This is done in exactly the same way in [6]. In the same fashion, we can prove that the AKC protocol supports the desirable security attributes as disused above. Thus the AKC protocol is secure. From A's perspective, the only person (other than A) who can correctly compute $MAC_{k'}(2, B, A, M_B, M_A)$, and therefore k', is an entity who can compute K. Since AK Protocol provides implicit key authentication, this entity must be B. Thus A has the assurance that B actually has computed k' and K, and therefore is also capable of computing the session key k. Hence AKC Protocol provides explicit key authentication.

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].

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 AKC Proposed Protocol provides all security attributes as well as the AKC MQV, LLK, and Song-Kim [3]-[10]-[11].

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 3 scalar multiplications. The MQV Protocol requires 2.5, and the LLK protocol requires two scalar multiplications only. Also, MAC_s can be computed efficiently.

TABLE I

THE NUMBER OF SCALAR MULTIPLICATIONS REQUIRED PER ENTITY

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

CONCLUSION

In this paper a new key agreement protocol (AK), and authenticated key agreement with key confirmation protocol (AKC) are proposed. These protocols 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. 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.

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, Springer-Verlag, pp.339-361, 1999.

[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, pp.30-45, 1997.

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

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

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

[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, pp.237-249, December 2000.