An Authenticated Constant Round Group Key Agreement Protocol Based on Elliptic Curve Cryptography

Zhu Liehuang[†], *Liao Lejian*[†], *Li Wenzhuo*[††], *Zhang Zijian*[†],

[†]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
[††]School of Software, Beijing Institute of Technology, Beijing, China

Summary
Group key agreement protocol allows all the members to agree upon a common session key, which may be used for later secure communication among all the participants. In this paper, an authenticated constant round group key agreement protocol is proposed based on authenticated two-party elliptic curve Diffie-Hellman key agreement protocol and the constant round group key agreement protocol recommended by Burmester and Desmedt (BD). Our protocol is more efficient than BD in terms of both communication and computation power and can prevent the man-in-the-middle attack as opposed to BD.

Key words:
Group Communication, Secure Communication, Key Agreement, Elliptic Curve.

1 Introduction

Group communication is the main communication form in distributed systems, such as Grid computing, multi-agent communication, live multiparty conferencing, and network games. How to make group communication secure becomes very important. However, as compared with the mature security mechanism of point to point communication, there are many security problems in group communication to be solved [1][2][3][4]. The foremost problem is how to make all members of a group have the same group key in security[5][6][7][8].

A group key agreement protocol allows a group of users to exchange information over an insecure network and agree upon a common session key, which can be used to encrypt and authenticate the group communication message, and ensure the confidentiality, integrity and authenticity of group communication messages.

Steiner et al[9]. extendeded the Diffie-Hellman protocol, which is the first pioneering two-party key agreement protocol, to multi party setting. The group Diffie-Hellman key agreement protocol (GDH) needs n rounds to agree upon a common session key for a group with n members. In particular, the number of rounds may be crucial in an environment where the number of group members is quite large, because members can't communicate until the other members finish the foregoing round protocol.

Burmester and Desmedt[10] proposed a constant round group key agreement protocol, which need just 2 rounds and just 3 times modular exponentiations computed per member. This paper replaces the traditional Diffie-Hellman protocol with elliptic curve Diffie-Hellman key exchange protocol to improve the performance in terms of both communication and computation power.

BD group key agreement protocol does not authenticate the members in the sense that an active adversary who has control over the channel can mount a man-in-the-middle attack to agree upon n separate keys with the members, without the members being aware of this. This paper analyzes how to break BD using man-in-the-middle attack and proposes an authenticated group key agreement protocol based on authenticated two party elliptic curve Diffie-Hellman key agreement protocol.

2 Constant Round Group Key Agreement protocol Based on Elliptic Curve Diffie-Hellman

As compared to currently prevalent cryptosystems such as RSA, DSA, and DH, ECC offers equivalent security with smaller key sizes, which is illustrated in the following table. Smaller key sizes result in power, bandwidth, and computational savings that make ECC especially attractive for constrained environments, such as AdHoc networks.

Table 1. Key size comparition between ECC and DH/DSA/RSA

Security Level in bits	ECC	DH/DSA/RSA	Key Size Ratio
80	192	1024	1:5
112	224	2048	1:9
128	256	3072	1:12
192	384	7680	1:20
256	521	15360	1:29

Elliptic curve Diffie-Hellman (ECDH) is more efficient than DH in terms of both communication and

computation power. This paper replaces DH as ECDH to improve the performance of BD.

2.1 Protocol Description

Elliptic curve domain parameters over F_p are a sextuple: $T=(p,a,b,P,t,h)$, where p is an integer specifying the finite field, $a,b\in F_p$ specify an elliptic curve $E(F_p)$, G is a base point on $E(F_p)$, prime t is the order of P, integer h is the cofactor $h=\#E(F_p)/t$. A set of members $m_1,m_2,...,m_n$ who want to agree upon a common session key should firstly agree on the elliptic curve domain $T=(p,a,b,P,t,h)$, and execute the group key agreement protocol as figure 1 **(Protocol 1)**.

1. Each m_i, $i=1,2,...n$, chooses $r_i<t$, and then computes and broadcasts $Z_i=r_iP$. The indices of r_i are taken in a cycle: so $r_{n+1}=r_1$ and $r_n=r_0$.

2. After receiving Z_{i-1} and Z_{i+1}, each m_i, $i=1,2,...n$, computes and broadcasts $X_i=r_i(Z_{i+1}-Z_{i-1})P=r_i(r_{i+1}-r_{i-1})P$.

3. After receiving all required X_i, each m_i, $i=1,2,...n$, compute the session key,

$$\begin{aligned}K_i &= nr_iz_{i-1}+(n-1)X_i+(n-2)X_{i+1}+...+X\\ &=(r_ir_{i+1}+r_{i+1}r_{i+2}+...+r_{i-2}r_{i-1})P\\ &=(r_1r_2+r_2r_3+...+r_nr_1)P\end{aligned}$$

Fig.1 Constant Round Group Key Agreement protocol Based on Elliptic Curve Diffie-Hellman (protocol 1).

2.2 Performance Analysis

Our protocol 1 needs 2 rounds 3 times elliptic curve point multiplications per member to agree upon a common session key. Because the time to execute one time elliptic curve point multiplication is much less than to execute one time modular exponentiation, our protocol is more efficient in term of computation power. Figure 2 shows the time to execute one time group key agreement protocol when using different ECDH schemes and DH schemes.

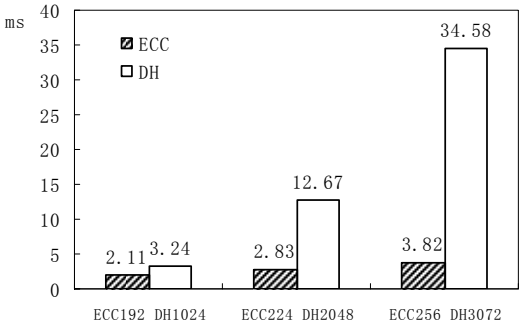

Fig.2 The time to execute one time group key agreement protocol when using different ECDH schemes (ECDH192, ECDH224, and ECDH256) and DH schemes (DH1024, DH2048, and DH3072).

Both BD and our protocol 1 need each member to broadcast simultaneity one message per round. The total communication quantities respectively are $2nL_{dh}$ and $2nL_{ecdh}$, where L_{dh} and L_{ecdh} respectively denote the length of one exchanged message in BD and our protocol. According to table 1, our protocol is more efficient in term of communication.

2.3 Security Analysis

Both BD and our protocol 1 are secure against passive adversaries, if the discrete logarithm problem is hard. However an active adversary can break the protocols with a man-in-the-middle attack as shown in figure 3.

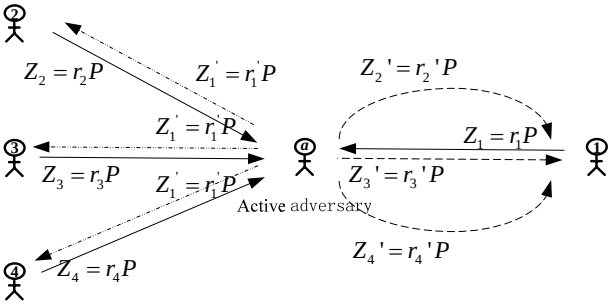

Fig.3 In the first round, active adversary intercepts $Z_i=r_iP$, replaces Z_i as $Z_i'=r_i'P$, $i=1,2,...n$, and sends $Z_i'=r_i'P$ to the other members m_j, $j=1,2,...,i-1,i+1,...,n$.

After intercepting and modifying the exchange information among members, the adversary can compute the pseudo session key K_i between $m_i\,(i=1,2,...n)$

$$\begin{aligned}K_i &= nr_{i-1}'Z_i+(n-1)X_i'+(n-2)X_{i+1}'+...+X_{i-2}'\\ &=nr_ir_{i-1}'P+(n-1)X_i'+(n-2)X_{i+1}'+...+X_{i-2}'\\ &=nr_iZ_{i-1}'+(n-1)X_i'+(n-2)X_{i+1}'+...+X_{i-2}'\end{aligned}$$

Then the adversary hosts the same group key as any

member m_i ($i = 1,2,...n$), and can modify and forge any packets.

3 Authenticated Constant Round Group Key Agreement protocol

Our protocol 1 proposed in section 2 is not authenticated protocol; because the active adversary can compute the same session key as each member. Several two-party authenticated key agreement protocols [7][8][9] have been proposed, such as MQV ECDH and MIT ECDH.

3.1 Protocol Description

We revise our protocol 1 to an authenticated group key agreement protocol using certificate as MIT ECDH. In the revised protocol, the static public key W_i of m_i is authenticated via certificate issued by a certifying authority (CA), where $W_i = w_i P$ and w_i is the private key of m_i. Each member possesses all the certificates of the other members. We illustrate the authenticated group key agreement protocol described in the figure 2 (Protocol 2).

1. Each $m_i, i = 1,2,...n$,chooses $r_i < t$, and then computes $T_i^l = r_i W_{i-1}$ and $T_i^r = r_i W_{i+1}$
2. Each m_i sends T_i^l to m_{i-1} and sends T_i^r to m_{i+1}.
3. Each m_i computes Z_{i+1} and Z_{i-1}, where $Z_{i+1} = w_i^{-1} T_{i+1}^{\ l} = w_i^{-1} r_{i+1} w_i P = r_{i+1} P$ and $Z_{i-1} = w_i^{-1} T_{i-1}^{\ r} = w_i^{-1} r_{i-1} w_i P = r_{i-1} P$.
4. Each m_i computes and broadcasts $X_i = r_i (Z_{i+1} - Z_{i-1}) P = r_i (r_{i+1} - r_{i-1}) P$.
5. After receiving all required X_i, each m_i computes the session key,

$$\begin{aligned} K_i &= n r_i z_{i-1} + (n-1) X_i + (n-2) X_{i+1} + ... + \\ &= (r_i r_{i+1} + r_{i+1} r_{i+2} + ... + r_{i-2} r_{i-1}) P \\ &= (r_1 r_2 + r_2 r_3 + ... + r_n r_1) P \end{aligned}$$

Fig.4 Authenticated Group Key Agreement protocol (protocol 2).

3.2 Security Analysis

The security of our protocol 2 depends on the hard of elliptic curve discrete logarithm problem.

Definition 1 (Elliptic Curve Discrete Logarithm Problem). If $E(\mathsf{F}_p)$ is an elliptic curve over F_p and P is a point on $E(\mathsf{F}_p)$, then the elliptic curve *discrete* logarithm *problem* on $E(\mathsf{F}_p)$ to the base P is the following problem: given a point $Q \in E(\mathsf{F}_p)$, find an integer r such that $Q = rP$, if such an integer exists.

Definition 2 (Authenticated Group Key Agreement Protocol). Let $\{M_1, M_2, ..., M_u\}$ is all the messages exchanged between members where u is the number of messages, $\{\overline{M_1}, \overline{M_2}, ..., \overline{M_u}\}$ is all the messages modified by an active adversary, $\{K_1, K_2, ..., K_n\}$ is the session key list where K_i is computed by m_i member computes, if adversary cannot compute a session $K \in \{K_1, K_2, ..., K_n\}$, the group key agreement protocol is authenticated.

Theorem 1: If elliptic curve discrete logarithm problem is hard, our protocol 2 is authenticated.

Proof: Suppose the adversary modified all T_i^l and T_i^r to $\overline{T}_i^{\ l} = r_i' W_{i-1}$ and $\overline{T}_i^{\ r} = r_i' W_{i+1}$, m_i compute the session key K_i as following.

$$K_i = n r_i \overline{Z}_{i-1} + (n-1)\overline{X}_i + (n-2)\overline{X}_{i+1} + ... + \overline{X}_{i-2}$$

If calculating $n r_i \overline{Z}_{i-1}$, the adversary can compute the same K_i as m_i because of possessing $\overline{X}_i, \overline{X}_{i+1}, ..., \overline{X}_{i-2}$.

In order to calculate $n r_i \overline{Z}_{i-1} = n r_i r_{i-1}' P$, the concerned messages possessed by the adversary are $T_i^l = r_i W_{i-1} = r_i w_{i-1} P$, $T_i^r = r_i W_{i+1} = r_i w_{i+1} P$, and r_{i-1}'.

Elliptic curve discrete logarithm problem is hard, the adversary cannot calculate r_i from $r_i W_{i-1}$ and $r_i W_{i+1}$, and cannot calculate $r_i P$ from $r_i w_{i-1} P$ and $r_i w_{i+1} P$ because of lacking w_{i-1} and w_{i+1}. So the adversary cannot calculate $n r_i \overline{Z}_{i-1}$. That is our protocol 2 is authenticated.

4 Conclusions

Because ECDH is more efficient than DH in terms of communication and computation power, our protocol 1 is more efficient than BD through replaces DH as ECDH. The times to execute one time BD is 9 times as to execute one time our protocol 1, and the messages length of BD is 12 times as of our protocol 1, when using DH3072 and ECDH256 respectively. Through using static public key issued by a certifying authority, our protocol 2 is authenticated, if the elliptic curve discrete

logarithm problem is hard.

5 References

[1] Kim H, Hong S M, Yoon H, Cho J W. Secure Group Communication with Multiplicative One-way Functions. In: Proc of the International Conference on Information Technology: Coding and Computing. Las Vegas: IEEE Press, 2005. 685－690.
[2] Agarwal D A, Chevassut O, RThompson M, etal. An Integrated Solution for Secure Group Communication in Wide-Area Networks. In: Proc of the 6th IEEE Symposium on Computers and Communications. Hammamet : IEEE Press, 2001. 22－28.
[3] Chu H H, Qiao L, Nahrstedt K. A Secure Multicast Protocol with Copyright Protection. ACM SIGCOMM Computer Communications Review, 2002, 32(2): 42－60.
[4] Amir Yair, Cristina N R, Stanton J, Tsudik G. Secure Spread: An Integrated Architecture for Secure Group Communication. IEEE Transactions on Dependable and Secure Computing, 2005, 2(3): 248－261.
[5] Abdel-Hafez A, Miri A, Orozco-Barbosa L. Scalable and fault-tolerant key agreement protocol for dynamic groups. International Journal of Network Management, 2006, 16(3): 185－201.
[6] Zhu L H, Cao Y D, Wang D. Digital signature of multicast streams secure against adaptive chosen message attack. Computers & Security, 2004, 23(3): 229－240.
[7] Zhu L H, Cao Y D, Liao L J, Tan Y A. Secure Group Key Distribution Protocol Based on Huffman One-way Key Chain Tree. WSEAS Transactions on Computers, 2006, 6(5): 1208－1213.
[8] Huang D J, Medhi D. A key-chain-based keying scheme for many-to-many secure group communication. ACM Transactions on Information and System Security, 2004, 7(4):523－552.
[9] Steiner M, Tsudik G, Waidner M. Diffie-Hellman key distribution extended to group communication. In: Proc of the 3rd ACM conference on Computer and communications security. Florida: ACM Press, 1996. 31－37.
[10] Burmester M, Desmedt Y. A secure and efficient conference key distribution system. Advances in Cryptology- EUROCRYPT'94, LNCS, Springer, Berlin, 950:275–286,1995.

Zhu Liehuang received the M.S. degree in Computer Science from Wuhan University in 2001 and Ph.D degree in Computer Science from Beijing Institute of Technology in 2004. He has also worked as a lecturer of Computer Science at Beijing Institute of Technology from 2004. His research interests are in cryptographic algorithms and protocols for cryptography, computer arithmetic, computer and network security.