A New Anonymous Conference Key Distribution System based on the Elliptic Curve Discrete Logarithm Problem*

Chou-Chen Yang[‡] Ting-Yi Chang[‡] Min-Shiang Hwang[†]

Graduate Institute of Networking and Communication Engineerings[†]
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.
Email: mshwang@cyut.edu.tw
Fax: 886-4-23742337

Department of Information and Communication Engineering[‡]
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.
Email: ccyang@cyut.edu.tw

October 11, 2002

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC91-2213-E-324-003.

[†]Responsible for correspondence: Prof. Min-Shiang Hwang.

A New Anonymous Conference Key Distribution System based on the Elliptic Curve Discrete Logarithm Problem

Abstract

In 1999, Tseng and Jan [11] proposed two conference key distribution systems (CKDS) with user anonymity based on the discrete logarithm problem and the interpolating properties of polynomials. Their first CKDS scheme uses a one-way hash function to hide the identities of the participants and to protect each participant's common key that is shared with the chairperson. In this article, we will propose a more efficient CKDS scheme with user anonymity which is based on the elliptic curve discrete logarithm problem and the properties of the line. Our scheme has the advantage of requiring less computing time than the Tseng-Jan CKDS with a one-way hash function.

Keywords: Cryptography, Conference key distribution system, User anonymity, One-way hash function, elliptic curve discrete logarithm.

1 Introduction

Diffie and Hellman [1] proposed a key distribution system (KDS) based on discrete logarithm problem for distributing the common session key between two users. It allows two users to establish a secret communication over an insecure channel. However, the Diffie-Hellman KDS is only suitable for the point-to-point situation. In order to make it suitable for more users when communicating with each other in a conference, Ingemarsson et al. [5] proposed a conference key distribution system (CKDS). In the CKDS, only legal participants attending the conference can recover the common conference key.

Most of the existing CKDS schemes [2, 3, 4, 5, 7] today do not have the privacy for the attending participants in the conference. To protect a participant

from the influence of other participants, the identity of the participant should be kept secret. Lin et al. [8] used sealed locks to achieve the CKDS with user anonymity. However, the computational complexity of their scheme makes it impractical. Later, Wu [13] proposed a CKDS with user anonymity based on the Diffie-Hellman KDS and the algebraic approach. In his scheme, a one-way function is used to hide the identities of the participants and to protect each participant's common key shared with the chairperson. On the other hand, the algebraic approach is used for the chairperson and legal attending participants to distribute and recover the common conference key, respectively.

In 1999, Tseng and Jan proposed two CKDS schemes with user anonymity. One (Tseng-Jan CKDS-1) is a modified version of Wu's scheme by using the interpolating properties of polynomials in place of the algebraic approach to reduce the computational complexity. The other (Tseng-Jan CKDS-2) does not use a one-way function to achieve the same purposes. However, Yang et al. [14] pointed out that the conspiracy attack could break the Tseng-Jan CKDS-2. If there are n participants attending the conference, $n-1$ participants can conspire to reveal the only other participant's common session key shared with the chairperson.

In this article, we will substitute some simple properties of the line for the interpolating properties of polynomials. Our scheme has a lower computational complexity than that of the Tseng-Jan CKDS-1. Moreover, we use the less time-consuming elliptic curve discrete logarithm problem to achieve the same purposes as the ordinary discrete logarithm problem in the Tseng-Jan CKDS-1.

The remainder of our paper is organized as follows. In Section 2, we shall briefly review the Tseng-Jan CKDS-1. In Section 3, we shall propose our CKDS with user anonymity. In Section 4, we shall analyze the security of our scheme. In Section 5, we shall compare the performance of our scheme with the Tseng-Jan CKDS-1. Finally, the conclusion will be in Section 6.

2 Review of the Tseng-Jan Conference Key Distribution System

The scheme includes into three stages: (1) system initiative stage, (2) conference key distribution stage, and (3) conference key recovery stage.

In the system initiative stage, the system chooses two large primes p and q such that $q|p-1$ and a generator g with order q in $GF(p)$. Then system assigns a secret key $x_i \in Z_q^*$ and the corresponding public key $Y_i = g^{x_i} \bmod p$ and the identity ID_i to each participant U_i $(i = 1, 2, \cdots, m)$ in the system. The set of all the participants in the system is denoted as $A = \{U_1, U_2, \cdots, U_m\}$. Then, the system delivers the secret key x_i to $U_i \in A$ over a secret channel.

In the conference key distribution stage, U_c, a chairperson, performs the following steps for distributing a conference key CK to the participants in B (Let $B = \{U_1, U_2, \cdots, U_n, n < m\}$ denotes the set of attending members).

Step 1. Compute the common session key $k_{ci} = Y_i^{x_c} \bmod p$ shared with each U_i.

Step 2. Compute the hash value $h_i = H(k_{ci} \parallel ID_c \parallel ID_i \parallel T) \parallel m$, where $H(\cdot)$ is a secure one-way hash function with fixed-length output, T is a timestamp, and $\parallel$ denotes the concatenation.

Step 3. Randomly choose a conference key $CK \in Z_q^*$ and construct the n-th degree of the polynomial $F(x) = \prod_{i=1}^{n}(x - h_i) + CK = x^n + c_{n-1}x^{n-1} + \cdots + c_1 x + c_0 \bmod q$.

Step 4. Compute the check value for CK on timestamp T as $V = H(CK \parallel ID_c \parallel T)$.

Step 5. Broadcast the message $M = \{ID_c, V, T, c_{n-1}, c_{n-2}, \cdots, c_0\}$.

In the conference key recovery stage, each $U_i \in B$, according to the message $M = \{ID_c, V, T, c_{n-1}, c_{n-2}, \cdots, c_0\}$ broadcasted by U_c, performs the following steps for recovering the conference key CK shared by the participants in B.

Step 1. Check the validity of the timestamp T. If it is invalid, stop the conference key recovery stage.

Step 2. Compute the common session key $k_{ic} = Y_c^{x_i} \bmod p$ shared with U_c.

Step 3. Compute $h_i = H(k_{ic} \parallel ID_c \parallel ID_i \parallel T) \parallel m$ and recover CK as $F(h_i) = (h_i)^n + c_{n-1}(h_i)^{n-1} + \cdots + c_1(h_i) + c_0 = CK \bmod q$.

Step 4. Check the validity of CK by verifying $H(CK \parallel ID_c \parallel T) = V$.

From the above descriptions, it can be seen that only $U_i \in B$ can recover the valid conference key CK from the polynomial by using his/her common session key k_{ic} shared with U_c.

3 The Proposed Scheme

Our CKDS with user anonymity is also composed of three stages, and the notations $\{A, B, U_i, ID_i, CK, H(\cdot), T, \|\}$ are the same as those in the Tseng-Jan scheme. During the system initiative stage, the system publicly chooses an elliptic curve E over a finite field $GF(q)$ and a base point G with order p [9]. Then the system assigns a secret key $x_i \in [1, p-1]$ and the corresponding public key $Q_i = x_i G$ to each participant $U_i \in A$. Then, the system delivers the secret key x_i to $U_i \in A$ over a secret channel.

In the conference key distribution stage, U_c performs the following steps for distributing the conference key CK shared by the participants in B.

Step 1. Compute the common session key $k_{ci} = x_c Q_i$ shared with each U_i.

Step 2. Compute the hash value $h_i = H(k_{ci} \parallel ID_c \parallel ID_i \parallel T) \parallel m$.

Step 3. Randomly choose a line $L(x) = c_1 x + CK \bmod p$ and compute $y_i = L(h_i) \bmod p$.

Step 4. Compute the check value for CK on timestamp T as $V = H(CK \parallel ID_c \parallel T)$.

Step 5. Broadcast the message $M = \{ID_c, V, T, c_1, y_1, y_2, \cdots, y_n\}$.

In the conference key recovery stage, each $U_i \in B$, according to the message $M = \{ID_c, V, T, c_1, y_1, y_2, \cdots, y_n\}$ broadcasted by U_c, performs the following steps for recovering the conference key CK shared by the participants in B.

Step 1. Check the validity of the timestamp T. If it is invalid, stop the conference key recovery stage.

Step 2. Compute the common session key $k_{ic} = x_i G_c$ shared with U_c.

Step 3. Compute $h_i = H(k_{ic} \parallel ID_c \parallel ID_i \parallel T) \parallel m$ and use it to reconstruct the line $L(x)$ to obtain CK.

Step 4. Check the validity of CK by verifying $H(CK \parallel ID_c \parallel T) = V$.

Obviously, we use the elliptic curve discrete logarithm problem to achieve the same purposes as the ordinary discrete logarithm problem in the Diffie-Hellman scheme and substitute some simple properties of the line for the interpolating properties of the polynomials. Only $U_i \in B$ can recover the valid conference key CK from the line by using his/her common session key k_{ic} shared with U_c.

4 Security Analysis

The security level of the proposed CKDS with user anonymity is based on the intractability of the elliptic curve discrete logarithm problem (ECDLP). An adversary who intends to reveal a secret key x_i from its corresponding public key Q_i will have to face ECDLP. In the rest of this section, several attacks will be raised and fought against to demonstrate the security of our scheme.

Attack 1: A non-attending participant $\hat{U}_i \notin B$ of this conference tries to reveal the common conference key CK from the message $M = \{ID_c,\ V,\ T,\ c_1,\ y_1,$

$y_2, \cdots, y_n\}$.

Analysis of Attack 1: The non-attending participant $\hat{U}_i \notin B$ first computes the hash value $\hat{h}_i = H(k_{ic} \parallel ID_c \parallel ID_i \parallel T)||m$ and then tries to reconstruct the line $L(x)$ with the knowledge of the message $M = \{ID_c, V, T, c_1, y_1, y_2, \cdots, y_n\}$. However, the valid value $\hat{y}_i = L(\hat{h}_i)$ is broadcasted by U_c using the attending participants' h_is. Hence, any non-attending participant has no ability to obtain the common conference key CK. For this reason, it is impossible for any adversary to reveal the common conference key CK.

Attack 2: An attending participant $U_i \in B$ of this conference tries to reveal another common session key k_{cj} and to identify another participant U_j, where $1 \leq j \leq n, j \neq i$.

Analysis of Attack 2: The attending participant $U_i \in B$ can easily reconstruct the line $L(x)$ and compute h_j of another participant. However, the common session key k_{cj} shared with the chairperson and the identity ID_j are protected by the one-way function $H(\cdot)$. Hence, the common session key k_{cj} cannot be revealed, and the identities of the participants in the conference are anonymous to each other.

Attack 3: An adversary tries to replay the intercepted message $M = \{ID_c, V, T, c_1, y_1, y_2, \cdots, y_n\}$ for impersonating the chairperson U_c to hold a conference.

Analysis of Attack 3: The adversary should first set a new acceptable timestamp T, so that the attending participant can verify the validity of T at Step 1 of the conference key recovery stage. However, he/she cannot forge the valid h_is without knowing x_c. To obtain x_c from Q_c is equivalent to solving ECDLP.

Attack 4: Some participants $U_i \in B$ try to collaboratively reveal the common session key k_{ic} of another participant $U_j \in B$.

Analysis of Attack 4: As in *Analysis of Attack 2*, the participants can obtain h_j. However, the common session key k_{cj} shared with the chairperson and the identity ID_j are protected by the one-way hash function.

5 Performance Comparison

In this section, we shall compare the computational complexity of our scheme with that of the Tseng-Jan scheme. To analyze the computational complexity, we first define the following notations.

T_H: the time for computing the adopted one-way hash function $H(\cdot)$.

T_{MUL}: the time for computing modular multiplication.

T_{EXP}: the time for computing modular exponentiation.

T_{EC_MUL}: the time for computing the multiplication of a number and a point on the elliptic curve.

n: the number of participants in the conference.

Because the time for computing modular addition is much less than T_H, T_{MUL}, T_{EXP} and T_{EC_MUL}, we ignore it in the comparison. Furthermore, the authors of [6, 10, 12] have pointed out that the elliptic curve discrete logarithm problem with order 160-bit prime offers approximately the same level of security as the discrete logarithm problem with 1024-bit modulus. Computing the multiplication of a number and a point on the elliptic curve and a modular exponentiation require an average of 29 1024-bits and 240 1024-bits modular multiplications, respectively. Thus, T_{EC_MUL} can be expected to be about 8 times faster than T_{EXP}, i.e., $8 \times T_{EC_MUL} = T_{EXP}$.

In the conference key distribution stage of our scheme, the chairperson U_c computes the common session key k_{ci} shared with each U_i (for $i = 1, 2, \cdots, n$). Step 1 requires $n \times T_{EC_MUL}$. Then, in Step 2, a timestamp T is acquired, and the hash value h_i is computed, which requires $n \times T_H$. Next, U_c randomly

chooses the line $L(x)$ and computes y_i (for $i = 1, 2, \cdots, n$) in Step 3, which requires $n \times T_{MUL}$. Finally, in Step 4, the check value V is computed, which requires T_H. Total computational complexity in this stage is required $n \times T_{EC_MUL} + (n+1) \times T_H + n \times T_{MUL}$.

After receiving the message $M = \{ID_c, V, T, c_1, y_1, y_2, \cdots, y_n\}$ broadcasted by U_c, each U_i enters the conference key recovery stage. He/She verifies T and computes the common session key k_{ic} shared with U_c, which requires T_{EC_MUL} (in Steps 1 and 2). Then, in Step 3, each participant computes h_i and reconstructs the line $L(x)$ to obtain CK, which requires $T_H + T_{MUL}$. Finally, in Step 4, each participant checks the validity of CK, which requires T_H. The total computational complexity in this stage is $T_{EC_MUL} + 2 \times T_H + T_{MUL}$.

Table 1: Computational complexities of the Tseng-Jan scheme and our scheme

	Key distribution stage	Key recovery stage
Tseng-Jan scheme	$n \times T_{EXP} + (n+1) \times T_H$ $+(n \times (n-1)/2) \times T_{MUL}$	$T_{EXP} + 2 \times T_H + (n-1)$ $\times T_{MUL}$
Our scheme	$n \times T_{EC_MUL} + (n-1) \times$ $T_H + n \times T_{MUL}$	$T_{EC_MUL} + 2 \times T_H + T_{MUL}$

The computational complexity of the Tseng-Jan scheme has been shown in [11]. According to Table 1, it is obvious that our scheme is more efficient than the Tseng-Jan scheme. We use the elliptic curve discrete logarithm to replace the ordinary discrete logarithm problem in the Tseng-Jan scheme. T_{EC_MUL} can be about 8 times faster than T_{EXP}. To construct the n-th degree polynomial $F(x)$ in the Tseng-Jan scheme requires $(n \times (n-1)/2) \times T_{MUL}$. We use the line $L(x)$ to replace the n-th degree polynomial $F(x)$ because it only requires $n \times T_{MUL}$ to compute y_i (for $i = 1, 2, \cdots, n$). On the other hand, to recover the common conference key CK from $F(x)$ and $L(x)$, the participant separately requires $(n-1) \times T_{MUL}$ and T_{MUL} in the Tseng-Jan scheme and our scheme. The computational complexity of recovering the conference key CK

from $F(x)$ in the Tseng-Jan scheme increases as the number of the participants in the conference increases, but it is only T_{MUL} in our scheme.

On the other hand, the chairperson broadcasts $n+3$ (i.e. $M = \{ID_c, V, T, c_{n-1}, c_{n-2}, \cdots, c_0\}$) and $n+4$ (i.e. $M = \{ID_c, V, T, c_1, y_1, y_2, \cdots, y_n\}$) messages, respectively. The number of messages broadcasted in our scheme is only larger than that in the Tseng-Jan scheme by 1.

6 Conclusion

In this paper, we have employed the elliptic curve discrete logarithm problem and some simple properties of the line to replace the interpolating properties of polynomials in the Tseng-Jan CDKS scheme with user anonymity. Our new scheme is more efficient than the schemes in [11, 13]. Our scheme outperforms the Tseng-Jan CDKS with user anonymity and is secure against impersonation and conspiracy attack.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, pp. 644–654, Nov. 1976.

[2] S. Hirose and K. Ikeda, "A conference distribution system for the start configuration based on the discrete logarithm problem," *Information Processing Letters*, vol. 62, no. 4, pp. 189–192, 1997.

[3] Min-Shiang Hwang and W. P. Yang, "Conference key distribution protocols for digital mobile communication systems," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 416–420, Feb. 1995.

[4] T. Hwang and J. L. Chen, "Identity-based conference key broadcast system," *IEE Proceedings - Computer Digital Technology*, vol. 141, no. 1, pp. 57–60, 1994.

[5] I. Ingemarsson, D. T. Tang, and C. K. Wong, "A conference key distribution system," *IEEE Transactions on Information Theory*, vol. IT-28, pp. 714–720, Sep. 1982.

[6] Neal Koblitz, Alfred Menezes, and Scott A. Vanstone, "The state of elliptic curve cryptography," *Designs, Codes and Cryptography*, vol. 9, no. 2/3, pp. 173–193, 2000.

[7] K. Koyama and K. Ohta, "Identity-based conference key distribution system," in *Advances in Cryptology, CRYPTO'87*, pp. 194–202, Lecture Notes in Computer Science, No. 293, 1987.

[8] C. H. Lin, C. C. Chang, and R. C. T. Lee, "A conference key broadcasting system using sealed lock," *Inf. Syst.*, vol. 17, no. 4, pp. 323–328, 1992.

[9] A. Menezes, "Elliptic curve public key cryptosystem," *Kluwer Academic Publishers*, 1993.

[10] R. Schroeppel, H. Orman, S. O'Malley, and O. Spatscheck, "Fast key exchang with elliptic curve systems," in *Advances in Cryptology, CRYPTO'95*, pp. 43–56, 1995.

[11] Yuh-Min Tseng and Jinn-Ke Jan, "Anonymous conference key distribution systems based on the discrete logarithm problem," *Computer Communications*, vol. 22, no. 8, pp. 749–754, 1999.

[12] E. De Win, A. Bosselaers, S. Vandenberghe, P. De Gersem, and J. Vandewalle, "A fast software implementation for arithmetic operations in $GF(2^n)$," *Asiacrypt'96*, vol. 1163, pp. 65–76, 1996.

[13] T. C. Wu, "Conference key distribution system with uesr anonymity based on algebraic approach," *IEE Proceedings - Computer Digital Technology*, vol. 144, no. 2, pp. 145–148, 1997.

[14] Chou-Chen Yang, Ting-Yi Chang, and Min-Shiang Hwang. "Comment on Tseng-Jan anonymous conference key distribution system without using a one-way hash function,". Technical Report CYUT-IM-TR-2002-016, CYUT, Sep. 2002.