Peer-to-Peer Netw. Appl. (2017) 10:857–864
DOI 10.1007/s12083-016-0434-7

An authenticated group key transfer protocol using elliptic curve cryptography

Priyanka Jaiswal[1] · **Sachin Tripathi**[1]

Received: 6 January 2016 / Accepted: 12 January 2016 / Published online: 10 February 2016

Abstract Several groupware applications like e-conferences, pay-per view, online games, etc. require a common session key to establish a secure communication among the group participants. For secure communication, such applications often need an efficient group key establishment protocol to construct a common session key for group communications. Conventional group key transfer protocols depends on mutually trusted key generation center (KGC) to generate and distribute the group key to each participant in each session. However, those approaches require extra communication overheads in the server setup. This paper presents an efficient and secure group key transfer protocol using elliptic curve cryptography (ECC). The proposed protocol demonstrates a novel group key transfer protocol, in which one of the group member plays the role of KGC (the protocol without an online KGC, which is based on elliptic curve discrete logarithm problem (ECDLP) and Shamir's secret sharing scheme. The confidentiality of the proposed protocol is ensured by Shamir's secret sharing, i.e., information theoretically secure and provides authentication using ECDLP. Furthermore, the proposed protocol resists against potential attacks (insider and outsider) and also significantly reduces the overheads of the system. The security analysis section of the present work also justifies the security attributes of the proposed protocol under various security assumptions.

✉ Priyanka Jaiswal
jaiswal.priyanka1985@gmail.com

Sachin Tripathi
var_1285@yahoo.com

[1] Department of Computer Science and Engineering, Indian school of Mines, Dhanbad, Jharkhand, India

Keywords Group key transfer protocol · Secret sharing · Confidentiality · Authentication

1 Introduction

Secure group communication is a primary need for various groupware applications like video conferencing, e-voting, online chatting, online gaming, etc. The fundamental criteria of the secure group communication are confidentiality and authentication. Confidentiality ensures the privacy of the message (secret) within the group means the message can be read only by an intended receiver. Message authentication ensures the receiver that the messages are sent by the particular sender and are not altered in route. To provide these security features in a group, a common session key is required to be shared among communication entities for encryption/decryption or other cryptographic operation. Therefore, a key establishment protocol is needed to construct a common session key among all legitimate members of the group. The key establishment protocols are broadly categories into [1]: key transfer protocols [2–4] and key agreement protocols [5, 6]. A Key transfer protocol can be subdivided into key transfer protocols with the KGC and key transfer protocol without KGC. In the first type, key transfer protocol depends on a mutually trusted third party called as KGC to select a session key and then distribute the session keys to all the group members secretly. In the second type of key transfer protocol, session keys are generated with the help of group members.

Traditional group key management protocols can be grouped into two categories: Centralized group key management protocols and distributed group key management protocols. The centralized approach is simple as it involves a single entity (or a small set of entity) to generate and distribute key to all the group members [7, 8] but there is a drawback of

centralized approach that there should be continuous availability of the central server for supporting the group communication. To overcome this type of problem, distributed key management approach is introduced [9–11]. Distributed key management involves dynamically selecting a group member that acts as a key distribution server. Distributed key management protocols are generally based on either Diffie–Hellman (DH) key exchange approach [10, 12–14] or non DH key exchange approach [15, 16]. However, these types of key management protocols use encryption/decryption techniques in the generation of the secret (session) key. In DH based key agreement protocol, the session key is determined by exchanging public key of two communication entities. Since the public key does not provide the property of authentication, a signature/certificate can be attached in public key to provide authentication. However, the DH key agreement protocol is not suitable for group communication which has more than two parties. So the two party DH key exchange protocol is generalized for group communication [10]. The non DH protocols generally provide a key agreement with fault tolerance [15, 16]. In general terms, fault tolerance means the system continuing its operation even if in the event of a power failure. Tzeng et al. [6] proposed a group key agreement protocol based on the discrete logarithm problem and pointed out that klein et al. [14] protocol is quite inefficient in terms of security. Chang and Laih [15] modified the Tzeng conference key agreement protocol based on bilinear pairing. In 2009, Huang et al.[16] proposed an enhanced non interactive group key agreement protocol based on the discrete logarithm problem (DLP) and their protocol was more efficient than Tzeng protocol, in terms of computation and communication cost. In 2010, Zhao et al. [17] proposed a group key agreement protocol based on the RSA cryptosystem to improve the performance of Huang et al.'s protocol.

On the other hand, to avoid the use of encryption secret sharing has been used to design group communication protocol. The concept of secret sharing was first introduced by Blakley [18] and Shamir [19] separately in 1979. There are two different methods to implement a secret sharing scheme: One assumes that a trusted offline server is involved only in the initialization process [20–22] and the other one assume that an online trusted server called KGC involved in all the processes [23]. The first scheme of secret sharing is called key pre-distribution scheme. In key pre-distribution scheme, a trusted authority generates and distributes pieces of information to all users offline. The main drawback of the pre-distribution scheme is that every user requires storing a large piece of information. The second one requires an online server to be active so that the trusted KGC generates and broadcasts group key information to all group members at once [23]. This approach uses the similar model like IEEE 802.11i standard [24]. In 1989, Laih et al. [23] proposed the first algorithm based on this approach using threshold secret sharing scheme. Later, there are some papers [2, 25, 26] following the same approach. Harn et al. [27] proposed a key transfer protocol uses secret sharing that provide confidentiality and authentication, in which KGC and each group member computed t degree interpolating polynomial. However, [28, 29] pointed out that it doesn't protect from malicious user and gave an improvement.

This paper represents a group key transfer protocol based on the concept of Shamir's secret sharing and ECC. The involvement of ECC reduces the size of the key as well as it takes less time for key computation. Furthermore, the proposed protocol replaces the role of KGC by a member called initiator that reduces the extra overheads of the online KGC.

The rest of this paper is organized as follows. In Sect. 2, we provide some preliminaries. The design principle of the proposed protocol is given in Sect. 3. The proposed group key transfer protocol is given in Sect. 4. In Sect. 5 security analysis and performance comparison is given and finally, we conclude in Sect. 6.

2 Preliminaries

In this section we introduce some fundamental backgrounds. The notations and the meaning of the notations are shown in Table 1.

2.1 Background of elliptic curve group

Let the symbol $E(F_p)$ denote an elliptic curve E over a prime finite field F_p, defined by an equation. $Y^2 \bmod p = (x^3 + ax + b) \bmod p$, where $a, b \in F_p$ with the discriminant $\Delta = (4a^3 + 27b^2) \bmod p \neq 0$. The point on $E(F_p)$ together with an extra point O called the point at infinity that forms a group G.: $G = \{f(x,y) : x, y \in F_p and (x,y) \in E(F_p)\} \cup \{o\}$

2.1.1 Point addition

Let the order of G is n. G is a cyclic additive group under the point addition operation '+' defined as follows:

Let $A, B \in G$, l be the line connecting A and B, and C' be the point of intersection of line l with $E(F_p)$. C' reflects on x axis and defines C.

$$C = A + B$$

2.1.2 Point multiplication

For any scalar k scalar point multiplication over $E(F_p)$ can be computed as follows:

$$kA = k.A = A + A + \ldots + A(kTimes)$$

Table 1 Notation table

Notations	Meaning
$E(F_p)$	Elliptic Curve over F_p
G	Additive group formed by the points on $E(F_p)$
Q	A generator of G (point on ECC)
h	A cryptographic secure one way hash function
$Puk_{(i/j)}$	Public key for entity i/j
$Prk_{(i/j)}$	Private key for entity i/j, $(i \neq j)$

The detail description of ECC can be found in [30, 31].

2.1.3 Discrete logarithm problem (DLP) on elliptic curve group

For a Given generator Q of G and an element $A \in Z_p^*$, to find an integer $a \in Z_p^*$ such that $A = a.Q$.

2.2 Background of secret sharing

Secret sharing scheme was first introduced by Blakley [18] and Shamir [19] for safeguarding the cryptographic key. In secret sharing a secret is divided into n shares and shares among n shareholders in such a way that with any t or more than t shares it is able to reconstruct the secret but with less than t shares, it cannot reconstruct the secret. This scheme is called as (t, n) Secret Sharing Scheme. It is denoted as $(t, n) - SS$.

Shamir's $(t, n) - SS$ Secret sharing is an algorithm in cryptography. It is a form of secret sharing where the secret is divided into n parts, giving each participant its own unique parts, where some of the parts or all of them are needed in order to construct the secret. There are n shareholders $U = \{U_1, U_2, \ldots, U_n\}$ and a mutual trusted participant $U_i \in U$ called the initiator of the group. This scheme consists of two algorithms.

1) Share Generation Algorithm

Share generation and secret reconstruction are related to each other. In share generation algorithm the initiator does the following.

- The initiator first picks a polynomial of degree $(t-1)$ arbitrarily: $f(x) = a_0 + a_1 x + \ldots + a_{t-1} x^{t-1}$, in which the secret $s = a_0 = f(0)$ and all coefficients $a_0, a_1, \ldots, a_{(t-1)}$ are in finite field $F_p = GF(p)$ with p elements.
- U_i computes all secret $s_i = f(i) (\mathrm{mod}\, p)$ for $i = 1, \ldots n$.
- Then the initiator gives outputs a list of n secret shares s_i to corresponding shareholders privately.

2) Secret Reconstruction Algorithm

In this phase any t shares $(s_{i1}, \ldots, s_{it})$ are used to reconstruct the secret s.

$$s = f(0) = \sum_{i \in A} s_i \beta_i = \sum_{i \in A} s_i \left(\prod_{j \in A - \{i\}} \frac{x_j}{x_j - x_i} \right) (\mathrm{mod} p)$$

where $A = \{i_1, \ldots, i_t\} \subseteq \{1, 2, \ldots, n\}$, β_i for $i \in A$ is Lagrange coefficients.

The above scheme satisfies the fundamental security criteria of secret sharing schemes as follows: 1) Secret s can be constructed with the knowledge of any t or more than t shares. 2) Secret s cannot be reconstructed with the knowledge, the knowledge of less than it shares. Shamir's scheme is information theoretically secure since it satisfies these two basic requirements to share a secret without any computational assumption.

3 Design principle

This section consists of two subsections. The first section (Sect. 4.1) describes the designing concept, and the security goals for our group key transfer protocol is given in Sect. 4.2.

3.1 The concept of our design

To maintain the confidentiality of the group key a onetime session key is required to share among the group members. Shamir's *(t, n)-SS* can be used to establish the common session key for all the group members. However, the conventional group key transport schemes based on secret sharing require an online trusted KGC to share secrets among group members. In addition, KGC must generate a group key and then uses secret sharing scheme to transmit group key to all members. This type of result causes extra overheads in system implementation. To overcome these drawbacks an alternative approach, in which one of the group members is chosen as initiator and has endowed with the authority to select the secret key as group key and to originate the group communication.

In our design, the concept of ECC is used to share a secret between initiator and other group member. Further, the initiator constructs an interpolated polynomial *f(x)* of degree less than one from the group members. The polynomial *f(x)* passes through these shares and the selected session key by using Lagrangian interpolation, where *f*(0) represents the session key. Furthermore, the initiator publishes some additional point on *f*(x), where the number of those public points is equal to the number of group members minus one. On the other hand, each group member except the initiator, who know the public points and the shared secret is able to reconstruct the interpolated polynomial *f*(x) and derive the session key as *f*(0) by

using Lagrangian interpolation. Finally, all group members share a common group session key.

3.2 Security goals

The main security goals for our group key transfer protocol are: *key freshness, key confidentiality and key authentication.*

Key freshness ensures that a group key never been used before. Thus, a compromised group key cannot cause any further damage in group communication.

Key confidentiality protects the group key such that a session key can only be recovered by authorized member i.e., a session is available to only authorized group members. It protects the group key from unauthorized access.

Key authentication confirms the identity of the users, it provides assurance to authorized group members that the group key is distributed by the initiator, but not an attacker.

4 The proposed protocol

Suppose that a set of t participants, $U = \{U_1, U_2, ..., U_t\}$ wants to establish a secure communication and each participant, including initiator must maintain a public/private key pair*(puk, prk)* such that $puk = prk \,.\, Q$, where Q is a generating point in the elliptic curve group. Note that the long term pair (*puk, prk*) is authenticated by a trusted authority with the corresponding certificate. An initiator as $(U_i \in U)$, one of the group members, has an authority to select the secret key as the group key and to originate the group communication. Figure 1 represents the proposed protocol structure. The proposed protocol consists of two phases, i.e., 1) secret establishment phase and 2) session key transfer phase.

The secret establishment phase consists of the following operations.

1. The initiator randomly selects a number $r_i \in Z_p^*$ broadcast the following information to all other members to announce the group communication:

   $$(r_i, puk_i, U_1, U_2, \ldots U_t).$$

2. Upon receiving the announcement from the initiator, each participating group member $U_j (j \neq i)$ selects a random number $r_j \in Z_p^*$ and compute the following:

   - $R_j = r_j.puk_j$
   - $\overline{R}_j = r_j.puk_i$
   - $s_j = \overline{R_j}.prk_j$
   - $Auth_j = h\left(s_j \middle|\middle| r_i\right)$

3. U_j send the following information to the initiator as the response: $\{R_j, Auth_j\}$
4. After receiving a message from U_j, the initiator computes $s_j^{'} = R_j.prk_i$ and then checks $Auth_j \underline{?} h\left(s_j^{'} || r_i\right)$, if the result is valid, the initiator believes that the secret $s_j = r_j.prk_i.\text{-}prkj.Q$ is shared with corresponding U_j otherwise clams that U_j is fraudulent and then restart the protocol.

In the session key transfer phase, the initiator and the other participating members U_j execute the following operation.

1. The initiator has a secret s_j shared with each member. These are basically elliptic curve points having x and y co–ordinates $s_j = (x_j, y_j)$. The initiator randomly selects a group session key k and constructs an interpolated polynomial $f(x)$ of degree $(t-1)$ passes through t points $(0, k)$ and (x_j, y_j) for $j = 1$ to $(t-1)$, by using Lagrange interpolation. Further the initiator also computes $(t-1)$ additional points P_i on $f(x)$, where $P_i = (x_i, y_i)$ for $i = 1$ to (t-1) to U_j. Finally, the initiator computes $Auth = h(k||r_i||U_1||U_2|| \ldots ||U_t||P_1||P_2|| \ldots ||P_{(t-1)})$ and broadcast the message $\{Auth, P_i\}$ to all other members.
2. On receiving the above message from initiator each participating member U_j knowing s_j, and $(t-1)$ additional points P_i, is able to reconstruct the polynomial $f(x)$ and derive the group key $k = f(0)$ by using Lagrange interpolation. Afterward, U_j computes $Auth^* = h(k||r_i||U_1||U_2|| \ldots ||U_t||P_1||P_2|| \ldots ||P_{(t-1)})$ and then check the hash value $Auth^* \underline{?} Auth$. If the result is correct the result is authenticated.

   After the successful execution of the above process, the session key k is established among all group members. Later the key (k) can be used for secure group communication.

Remark Most of the key transfer protocols based on Shamir's (*t, n*)-*SS* are claimed information theoretically secure. However, these schemes must pre shared secret between the dealer and each participant. In other words, the secret must be shared via secure channel. Actually, it is strong assumption to suppose that a secure channel is existed in public networks. That is most existing scheme does not propose any practical method to share secrets in the public network. This work presents the first ECDLD assumption based group key transfer protocol to share the secret between initiator and other group participants. Next, we propose a group key transfer protocol based on Shamir's (*t, n*)-*SS*. Since the concept of Shamir's (*t, n*) -*SS* is information theoretically secure and adapted to transfer the group key, So we can say that group key procedure of the proposed scheme is information theoretically secure.

Steps	Initiator (U_i)	Other Group Members(U_j)
1.	Select $r_i \in Z_p^*$ $(r_i, puk_i, (U_1, U_2, ..., U_t)) \xrightarrow{*}$	
2.		Select $r_j \in Z_p^*$ and Compute $R_j = r_j.puk_j$ $\overline{R}_j = r_j.puk_i$ secret $s_j = prk_j.\overline{R}_j$ $Auth_j = h(s_j \| r_i)$ $\leftarrow (R_j, Auth_j)$
3.	Compute $s_j^{'} = prk_i.R_j$ Check $Auth_j \stackrel{?}{=} (s_j^{''} \| r_i)$ If found valid result The U_i belives on U_j,	
4.	Shared secret s_j with U_j U_i sepret each secret, Derive the point $s_j = (x_j, y_j)$ Compute $f(x)$ of degree (t-1) passing through $(0,k)$ to (x_j, y_j) Construct (t-1) additional point P_i on $f(x)$ and $Auth = h(k \| r_i \| U_1, U_2 \| ... \| U_t \| P_1 \| P_2 \| ...P_{(t-1)})$ $\{Auth, P_i\} \xrightarrow{*}$	
5.		U_i computes interpolating polynomial $f(x)$ and (t-1) additional points P_i and then check weather $Auth \stackrel{?}{=} h(k \| r_i \| U_1, U_2 \| ... \| U_t \| P_1 \| P_2 \| ...P_{(t-1)})$

Fig. 1 The proposed group key transfer protocol

5 Security analysis

This section justifies the proposed protocol against different types of adversary attacks. Adversaries can be categorized as two types: 1) Outsider adversaries and insider adversaries. The first types of adversaries want to crack the confidentiality. The Second type of adversaries authorized to know the group session key and attempting to recover the individual member secret. In the following security discussion, we will show that our group key transfer protocol is secure against outsiders and insiders adversaries and achieves the following security goals: *1) key freshness 2) key confidentiality 3) key authentication.*

Theorem 1 The key transfer protocol achieves key freshness.

Proof Key freshness is ensured by the initiator, since a random group key is selected by the initiator for each service request. In addition, the group key is a function of random number selected by each participating group member and one time secret shared between corresponding group member and initiator.

Theorem 2 The key transfer protocol achieves key confidentiality.

Proof key Confidentiality is ensured due to the security feature and ECDLP assumption and Shamir's (*t,n*)-*SS*. Let us focus on the stage of group key generation and distribution. The initiator computes secrets $s_j^{'} = prk_i.R_j$ as a point and construct an interpolated polynomial of degree $(t-1)$ to pass through the t points, $(0,k)$ and (x_j, y_j) for $i = 1 to (t-1)$ and makes $(t-1)$ additional points P_i publicly known, so that the authorized member gets the information to construct the secret key. However, the unauthorized member (outsider) has only $(t-1)$ public points on *f(x)* are available. They know only public information which is broadcasted by the initiator. Thus, the unauthorized members know nothing about the group key. In other words, the proposed protocol is secure since the Shamir (*t,n*)-SS is information theoretically secure (i.e., it does not involve any computational assumption) and elliptic curve cryptosystem is based on the difficulty of ECDLP.

Theorem 3 The proposed protocol achieves key authentication.

Proof Group key authentication is provided through the value of $Auth = h(k||r_i||U_1||U_2||\ldots||U_i||P_1||P_2||\ldots||P_{(t-1)})$ where *Auth* is a one way hash function with the secret group key and random challenges generated by the initiator. It follows Theorem 2 that the unauthorized member knows nothing about the group key and so, they cannot forge *Auth* value. Any insider also cannot forge the group key since the group key is the function of all secrets shared between the group members and the initiator. Furthermore, any replay of *Auth* and P_i can be detected since the group key is constructed with the help of shared secrets between initiator and each group member's which is a function of the initiator and each group member's random number.

Theorem 4 (Outsider Attack) Assume that an adversary who impersonates a group member requesting for group communication, then the attacker can neither obtain the group key nor share the group key with any other member.

Proof Although the adversaries can intercept the messages between the initiator and the participating members, the adversaries can't find shared secret s_j, i.e., $s_j = prk_j.\overline{R}_j$ due to the long term private key prk_j of any members U_j are unknown. Furthermore, the group key k is can only be recovered by the honest member who has the correct private key corresponding to the shared secret s_j. Therefore, the adversaries can't impersonate as any group member to obtain the group key. On the other hand, since the adversaries do not have the information about the private key prk_i of the initiator, thus the adversary cannot impersonate, as the initiator securely share the secret with the other member, in other words the adversary can't share the group key with any group member by masquerading as an initiator.

Theorem 5 (Insider Attack) Assume that the protocols run successfully many times; then, the onetime secret (x_j, y_j) of each U_j shared with the initiator still can't be traced by the other group member.

Proof In order to provide the group service k on receiving the group key request, the initiator generates a polynomial $f(x)$ of degree $(t-1)$ to pass through t points, $(0, k)$ and (x_j, y_j) for $j = 1 to (t-1)$. Each appropriate group member can obtain the one time secret (x_j, y_j) shared with the initiator by using the interactive key agreement protocol. Furthermore, with the knowledge of shared secret and $(t-1)$ public information, only the honest group member is able to reconstruct the polynomial $f(x)$. However, the secret of each group member shared by the initiator is still untraceable by insiders, due to the fact that the onetime secret (x_j, y_j) depends on the random number's and long term private keys (prk_i, prk_j)

5.1 Functionality and security comparison

This section compares the major functionalities and security aspects of the proposed protocol with some other existing protocols in Table 2. The result of the table under the proposed protocol is obtained from the security analysis section (section 5) while the same for existing protocols are taken from the referenced papers. The results show that the proposed protocol achieving all desired functionalities, however, others are not achieving all functionalities.

F1 (Without an online KGC)
The proposed scheme supports communication among group members without initialization of an online KGC. In the proposed protocol one of the group member play the role of the KGC and initiate communication. However, in other key transfer protocol, KGC is required for the generation and distribution of group key.

F2 (Group key generated by user)
The other functionality that supports the proposed key transfer protocol is that the group session key is generated with the help of members of the group, is no need of a trusted server.

F3 (No need for additional synchronized time)
Most of the group key transfer schemes required additional synchronization time at the starting phase of the protocol. However, the proposed scheme does not require additional synchronization time.

F4 (Mutual Authentication)
The proposed scheme provides mutual authentication between the initiator and other participating member, the mutual authentication is supported by ECDLP assumptions.

F5 (Session key agreement)
The proposed schemes also support the session key agreement technique, which helps to establish a common and secure session key among the group members in each session. With the session key agreement, the member of the group exchange high confidential data among them.

Table 2 Functionality comparison of different schemes with the proposed scheme

Functionality /schemes	F1	F2	F3	F4	F5
Proposed	Y	Y	Y	Y	Y
Huang [16]	Y	Y	N	N	Y
Zhao[17]	Y	Y	N	N	Y
Harn and Lin [27]	N	N	Y	Y	Y
Sun [4]	N	N	Y	N	Y
Liu [29]	N	N	Y	Y	Y

Y Yes (Supported), *N* No (Not Supported)

Table 3 List of different security principles, operations used and overall computation cost of different schemes

Schemes	Security principles	Operations used	Over-all computation cost	ECC is used
Proposed	ECDLP + Shamir Secret Sharing	Hash, EPM	Low	Yes
Huang [16]	DLP	Hash, XOR, MEXP	High	No
Zhao [17]	Factorization Problem	Hash, MEXP	High	No
Harn and Lin [27]	Shamir Secret Sharing + Factorization Problem	Hash, XOR	High	No
Sun [4]	Derivative secret sharing + DLP	Hash, XOR, MEXP	High	No
Liu [29]	Shamir Secret Sharing + Factorization Problem	Hash, XOR	High	No

5.2 Performance analysis

A comparative study in terms of security principles, operation used, ECC is used, and overall computation cost in different schemes such as [4, 16, 17, 27, 29] and the proposed scheme is shown in Table 3. The key distribution protocol of Huang scheme uses DH key exchange algorithm [32] and since random challenges required to execute modular exponentiation (MEXP), which is an expensive operation. The computation cost of elliptic curve point multiplication is much less than that modular exponentiation [33]. This is because 160 bit ECDLP and 1024 bit discrete logarithm problem (DLP) have the same security level [30]. Therefore, the Huang scheme has a high computational cost. The proposed protocol reduces the computation, communication and storage space cost, as the ECC and Shamir's secret (t, n) -*SS* is used. It is to be noted that the proposed scheme uses the general cryptographic hash function instead of XOR operation. Elliptic curve multiplication/addition (EPM/EAD) is used instead of modular exponentiation. EPM/EAD is quite slower than XOR operation, but instead, encryption/decryption technique (having slower processing speed) secret sharing (faster) is used. Therefore, overall computation cost of the proposed protocol is less than others [4, 16, 17, 27]. From the security analysis and efficiency discussion, it is obvious that the proposed scheme is efficient, secure and user friendly.

6 Conclusion

This paper presents an efficient group key transfer protocol using ECC. In the proposed protocol, the role of KGC is replaced by a group member called as initiator. Initiator distributes information related to the group key to all other members of the group. In this paper, we remove the extra overheads of KGC in system preparation as well as save the computation and communication cost with minimal storage overheads. The confidentiality of the group key distribution is information theoretically secure. This protocol also provide group key authentication. Security analysis shows that the protocol is also safe from possible attack. This protocol is fairly interesting for group oriented application.

Acknowledgments The work presented in this paper is supported by UGC (University Grant Commission), Govt. of India under grant no. /file no. 41/632/2012(SR) and project no.-UGC(77)/2012-13/316/CSE.

References

1. Boyd C (1997) On key agreement and conference key agreement. In: Proceeding of Second Australasian Conference on Information Security and Privacy (ACISP'97) 294–302
2. Sáez G (2003) Generation of key predistribution schemes using secret sharing schemes. Discret Appl Math 128(1):239–249
3. Olimid RF (2013) On the security of an authenticated group key transfer protocol based on secret sharing. In: Mustofa K, Neuhold EJ, Tjoa AM, Weippl E, You I (eds) ICT-EurAsia 2013. LNCS, vol 7804. Springer, Heidelberg, pp 399–408
4. Sun Y, Wen Q, Sun H, Li W, Jin Z, Zhang H (2012) An authenticated group key transfer protocol based on secret sharing. Procedia Eng 9:403–408
5. Katz J, Yung M (2003) Scalable protocols for authenticated group key exchange. In: Boneh D (ed) CRYPTO 2003. LNCS, vol 2729. Springer, Heidelberg, pp 110–125
6. Tzeng WG (2002) A secure fault-tolerant conference key agreement protocol. IEEE Trans Comput 51(4):373–379
7. Fiat A, Naor M (1994) Broadcast Encryption. In: Stinson DR (ed) CRYPTO 1993. LNCS, vol 773. Springer, Heidelberg, pp 480–491
8. Canetti R, Garay J, Itkis G, Micciancio D, Naor M, Pinkas B (1999) Multicast security: a taxonomy and some efficient constructions. In: Proceedings of Eighteenth annual joint conference of the IEEE Computer and communication societies (INFOCOM'99), IEEE. 2:708–716
9. Ingemarsson I, Tang DT, Wong CK (1982) A conference key distribution system. IEEE Trans Inf Theory 28(5):714–719
10. Steiner M, Tsudik G, Waidner M (1996) Diffie-Hellman key distribution extended to group communication. In: Proceedings of Third ACM Conference Computer and Communication Security (CCS'96), ACM Press. 31–37
11. Steer DG, Strawczynski L, Diffie W, Wiener MJ (2000) A secure audio teleconference system. In: Proceedings of Eighth Annual International Cryptology Conference: Advances in Cryptology (Crypto'88), LNCS. Springer New York. 403:520–528
12. Bohli JM (2006) A framework for robust group key agreement. In: Proceedings of International Conference on Computational Science and Applications (ICCSA'06), LNCS, Springer, Heidelberg. 3982: 355–364

13. Hsu C, Zeng B, Cheng Q, Cui G (2012) A novel group key transfer protocol. Cryptology ePrint Archive, Report 2012/043
14. Klein B, Otten M, Beth T (1995) Conference key distribution protocols in distributed systems. In: Proceedings of Codes and Ciphers: Cryptography and coding IV. pp. 225–241
15. Cheng JC, Laih CS (2009) Conference key agreement protocol with non-interactive fault-tolerance over broadcast network. Int J Inf Secur 8(1):37–48
16. Huang KH, Chung YF, Lee HH, Lai F, Chen TS (2009) A conference key agreement protocol with fault-tolerant capability. Comput Stand Interfaces 31(2):401–405
17. Zhao J, Gu D, Li Y (2010) An efficient fault tolerant group key agreement protocol. Comput Commun 33(7):890–895
18. Blakley G (1979) Safeguarding cryptographic keys. In: Proceedings of the 1979 AFIPS National Computer Conference, AFIPS Press. pp. 313–317
19. Shamir A (1979) How to share a secret. Commun ACM 22(11): 612–613
20. Blom R (1985) An optimal class of symmetric key generation systems. In: Beth T, Cot N, Ingemarsson I (eds) EUROCRYPT 1984. LNCS, vol 209. Springer, Heidelberg, pp 335–338
21. Blundo C, De Santis A, Herzberg A, Kutten S, Vaccaro U, Yung M (1993) Perfectly-secure key distribution for dynamic conferences. In: Brickell EF (ed) CRYPTO 1992. LNCS, vol 740. Springer, Heidelberg, pp 471–486
22. Hsu CF, Cui GH, Cheng Q, Chen J (2011) A novel multi-linear secret sharing scheme for group communication in wireless mesh networks. Netw Comput Appl 34(2):464–468
23. Laih C, Lee J, Harn L (1989) A new threshold scheme and its application in designing the conference key distribution cryptosystem. Inf Process Lett 32(2):95–99
24. IEEE 802.11i-2004 (2004) Amendment 6: Medium access control (MAC) Security Enhancements
25. Berkovits S (2001) How to broadcast a secret. Workshop on Theory and Application of Cryptographic Technique (Eurocrypt'91), LNCS, Springer, Heidelberg. 547:535–541
26. Li CH, Pieprzyk J (1999) Conference key agreement from secret sharing. In: Pieprzyk JP, Safavi-Naini R, Seberry J (eds) ACISP 1999. LNCS, vol 1587. Springer, Heidelberg, pp 64–76
27. Harn L, Lin C (2010) Authenticated group key transfer protocol based on secret sharing. IEEE Trans Comput 59(6):842–846
28. Nam J, Kim M, Paik J, Won D (2012) Security weaknesses in Harn-Lin and Dutta-Barua protocols for group key establishment. KSII Trans Internet Inf Syst 6(2):751–765
29. Liu Y, Cheng C, Cao J, Jiang T (2013) An improved authenticated group key transfer protocol based on secret sharing. IEEE Trans Comput 62(11):2335–2336
30. Hankerson D, Menezes A, Vanstone S (2004) Guide to elliptic curve cryptography. Springer, New York
31. Stinson DR (2002) Cryptography theory and practice, 2nd ed., CRC Press
32. Diffie W, Hellman ME (1976) New directions in cryptography. IEEE Trans Inf Theory 22(6):644–654
33. Chung YF, Huang KH, Lai F, Chen TS (2007) ID-based digital signature scheme on the elliptic curve cryptosystem. Comput Stand Interface 29(6):601–604

Priyanka Jaiswal received her B. Tech degree in Computer Science and Engineering from Institute of Technology & Management, GIDA, Gkp INDIA. Currently, She is pursuing Ph.D in the department of Computer Science and Engineering, Indian School of Mines, Dhanbad. She has done several IEEE international conferences and published papers in conference proceedings and journals. Her current research interests include cryptographic authentication protocols, and network security in a wireless environment. Email: Jaiswal.priyanka1985@gmail.com

Sachin Tripathi is an Assistant Professor in Computer Science & Engineering Department at Indian School of Mines, Dhanbad, Jharkhand,India. He received his Ph.D. in Computer Science and Engineering from the Indian School of Mines and has been teaching computer science subjects for over more than 10 years. His research interest is in group security. Email:var_1285@yahoo.com