An Efficient Group Key Transfer protocol for Secure Encryption

1A.V.V.S. Murthy and 2*P. Vasudeva Reddy

1Department of Mathematics
Dr S.R.K. Govt. Arts College
Yanam (U.T of Puducherry), India
2Department of Engineering Mathematics
College of Engineering, Andhra University,
Visakhapatnam, A.P, India

E-mail: murthyavvs@gmail.com, "vasucrypto@andhrauniversity.edu.in
"corresponding author

Abstract. Cryptography algorithms provide confidentiality, integrity and authentication of communication channels. These algorithms and protocols are based on sophisticated mathematics such as computational and algorithmic algebraic geometry, coding theory, and generic algorithms for finite abelian groups. For secure encryption, establishing or transferring a shared key between the parties is an interesting challenge. Key management is an essential cryptographic primitive upon which other security primitives are built.

In this paper, we present a generic construction of asymmetric key Group key transfer protocol using one way function and coding theory technique. By guaranteeing the freshness of authentication messages, the authenticity of the generator of authentication messages and the completeness of the authenticator, the improved protocol can resist various passive and active attacks. The protocol allows the participants in the group to derive a common encryption key, provide the key security and unknown key share properties. Security analysis of proposed scheme is also discussed.

1. Introduction

Key management [1] is an essential cryptographic primitive upon which other security primitives are built. So, the proposed key management protocol should not only support the establishment of pairwise keys but also other keys such as cluster keys or group key. A Group key [3,4] is a common key shared among all participants in the same group, and is mainly used for securing locally broadcast messages. A group key is a globally shared common key, used by the base station for the encryption [17] of messages that are broadcast to the whole network. The Group Key Management scheme can be used in security group communication of revocation encryption schemes [8], group encryptions and threshold encryption schemes. The group key security protocol [2] should solve such problems as group key produce, group key update regularity. These group key management schemes can be divided into three basic categories: (i) centralized group key agreement protocols (CGKAP) [15], (ii) decentralized group key management protocols with relaying (DeGKMP) [12], and (iii) Distributed group key agreement.

In distributed group key transfer [7,8], all group members are treated equally. Hence, group keys should be negotiated among all group members through Diffie Hellman (DH) key exchange or based on secret sharing theory to ensure fairness. In the CLIQUE scheme [18], group members can deliver their DH seeds orderly through insecure channels and the last member gets all the DH seeds to compute the group key and then multicast the received DH seeds to other members so that all members can get the group key, thus generating $O(N^2)$ key messages and incurring $O(N^2)$ computation cost. Secret sharing theory [6] can enhance the robustness of group key generation in which each group member is issued a seed of the group key securely and any group member must collect $M(M < N)$ secret seeds from a subset of the group members to recover the group key [5,11]. Therefore, an attacker who captures less than $M$ members cannot recover the group key. This approach causes frequent interactions and incurs high computation cost, and the exchange of secret seeds must be protected using shared keys between the peers. In the IoT scenario [12,13], the above group key negotiation schemes are not suitable for WSNs [10,14] since the cost of communication and computation is more than that of group key distribution schemes. Moreover, the reasons for the infeasibility of group key distribution schemes are also exist.

1.1. Motivation and Our Contribution

A group key management is a process or a protocol allows a set of users to establish a common secret via open networks. The common use of this established group key (also termed as conference key) is to permit users to encrypt and decrypt particular broadcast message that is meant for the total user group.

The motivation for our scheme is to come up with a new key transfer protocol in which rekeying operation can be done easily because of less computations. In the previous schemes which are based on polynomial interpolation, it is difficult to perform rekeying operation because for every rekeying operation the KGC has to select new polynomial. Our scheme is based on MDS codes where all the computations are done on the matrix. The computations done on the matrix are less compared to the computations done on polynomial. In [1] scheme every participant has to send random challenge to the KGC. But in our scheme cost of communication has been reduced drastically no need of any random challenge and every participant will recover session key by using their private share and public shares.

2. Preliminaries

2.1. Secret Sharing Scheme

Secret sharing scheme is a protocol, which is used to distribute secret among participants in such a way that an authorized subset of participants can uniquely reconstruct the secret and an unauthorized subset can get no information about the secret. The concept and the first realization of secret sharing were presented independently in [6] and in [7]. In a secret sharing scheme, there exists a dealer, $n$ participants, and possibly a reconstructor. The dealer splits a secret $s \in S$, into $n$ pieces, called shares, and sends one share to each participant over a private point-to-point channel. An access structure is the set of subsets of participants that are qualified to recover the secret. In a $(t, n)$-threshold access structure, where $1 \leq t < n$, any $t$ or more participants can reconstruct the secret, and the knowledge of $t$ or less shares leaves the secret $s$ indeterminate. A $(t, n)$-threshold secret sharing scheme is said to be perfect if no subset of $t$ or less shares can leak any information about the secret $s$.

2.2. Maximum Distance Separable Codes

An $[n, k, d]$ block code over $F_q$ is called Maximum Distance Separable (MDS) code if distance $d = n - k + 1$. Two important properties, namely,
Any k columns of a generator matrix are linearly independent and
Any k symbols of a codeword may be taken as message symbols,
of MDS codes have been exploited in the construction of our schemes. It may be noted that
for any \( k \), \( 1 \leq k \leq q - 1 \), and \( k \leq n \leq q - 1 \) there is an \([n,k,n-k+1]\) MDS code and an
\([q,k,q-k+1]\) extended Reed Solomon code.

Shamir [6] pointed out that a hierarchical variant of threshold secret sharing scheme can be
introduced simply by assigning larger number of shares to higher level participants. However,
such a solution can be easily seen to be not ideal. In [16,17,18] the authors proposed a secret
sharing scheme for hierarchical access structures which are based on MDS codes and the scheme
is ideal, secret can be recovered in polynomial time for authorized set.

3. Group Key Transfer Protocol

Group key transfer protocol is very important for secure group communication systems. The
goal of this protocol is to efficiently generate a session key and securely transfer this key to
the group members. Group key transfer protocols [23,24] can be classified into centralized and
distributed ones. In centralized scheme key generation and management is done by trusted third
party KGC (Key Generation Center). In distributed scheme keys are generated and managed by
uniform distribution of all group members.

Diffie-hellman protocol [3] is a key agreement protocol. It generates session key by publicly
exchanging information between the communicating entities. But this algorithm can only
provide session key for two entities; not for a group more than two members. Most distributed
key agreement protocols [21,22] make use of DH protocol.

Key Management for multicast communication [19,20] applications include generating,
distributing and updating of the group keys. If any new member joins the group or any old
member leaves the group then the new key needs to be generated and distributed to all the valid
group members. This process of updating the key is called as rekeying operation. If this is not
done new users can read old messages and old users can keep on reading new messages. That is
backward and forward secrecy cannot be guaranteed.

Group key transfer protocol based on secret sharing was first proposed by Lein Harn and
Changlu [1]. Their method uses shamir secret sharing scheme, which is based on lagrange
interpolation.

3.1. Lein Harn et.al Group Key Transfer Protocol

Harn et.al [1] developed a key transfer protocol which is used to transfer secret session key among
a group. This scheme is based on secret sharing and uses lagranges polynomial interpolation for
recovery. Harn et.al [1] algorithm consist of four phases.

- Initialization of KGC
- User Registration
- Key Generation and Distribution
- Recovery

**Initialization of KGC:** The KGC randomly chooses two safe primes \( p \) and \( q \) (i.e., primes such
that \( p' = p - 1 \) and \( q' = q - 1 \) are also primes) and compute \( n = pq \). \( n \) is made publicly known.

**User Registration:** Each user is required to register at KGC for subscribing the key distribution
service. The KGC keeps tracking all registered users and removing any un-subscribed users.
During registration, KGC shares a secret, \((x_i, y_i)\), with each user \( U_i \), where \((x_i, y_i) \in \mathbb{Z}_n^*\).

**Key Generation and Distribution:**
• Distribution: The initiator sends a key generation request to KGC with a list of group members as 
  \(<U_1,U_2,\ldots,U_t>\).
• KGC broadcasts the list of all participating members, 
  \(<U_1,U_2,\ldots,U_t>\), as a response.
• Each participating group member sends a random challenge, \(R_i\in\mathbb{Z}_n\), to KGC.
• KGC randomly selects a group key, \(k\), and generates an interpolated polynomial \(f(x)\) with degree \(t\) to pass through \((t+1)\) points, \((0,k)\) and \((x_i,y_i\oplus R_i)\) for \(i = 1\cdots t\). KGC also computes \(t\) additional points, \(P_i\), for \(i = 1\cdots t\), on \(f(x)\) and \(\text{Auth} = h(k,U_1,\ldots,U_t,R_2,\ldots,R_t,P_1,\ldots,P_t)\), where \(h\) is a one-way hash function. All computations on \(f(x)\) are over \(\mathbb{Z}_n\). KGC broadcasts \(<\text{Auth},P_i>\) for \(i = 1,\ldots,t\) to all group members.

Recovery: For each group member, \(U_i\), knowing the shared secret, \((x_i,y_i\oplus R_i)\) and \(t\) additional public points, \(p_i\), for \(i = 1\cdots t\) on \(f(x)\), is able to compute the polynomial \(f(x)\) and recover the group key \(k = f(0)\). Then \(U_i\) computes \(h(k,U_1,\ldots,U_t,R_2,\ldots,R_t,P_1,\ldots,P_t)\) and checks whether his hash value is identical to Auth. If these two values are identical, \(U_i\) authenticates the group key is sent from KGC.

4. Proposed Group Key transfer protocol
In Group Key transfer protocol the Key Generation center (KGC) will broadcast group key information to all group members at once and only authorized group members can recover the group key; whereas unauthorized group members cannot recover the group key.

Our proposed Group key transfer protocol is based on MDS (maximum distance separable codes). The KGC will use \((n,k,n-k+1)\) MDS code over field \(\mathbb{F}_q\) to distribute common session keys to participants, where \(n\) is total number of participants going to communicate each other and \(k\) is the total number of shares required for each participant to recover the session key \(s\). Here we are using Vandermonde matrix as a generator matrix.

This key transfer protocol has five phases:

• User Registration phase
• Initialization phase
• Key Generation and Distribution phase
• Reconstruction phase
• Verification phase.

User Registration Phase:
Each user is required to register at KGC and subscribe to the key distribution service. The KGC keeps track of all registered users and removes the unsubscribed users from the list.

Initialization phase:
Let \(\mathbb{F}_q\) be a finite field with \(q\) elements, where \(q\) is a prime power with \(q > n\). The KGC randomly chooses \(n\) vectors, each of length \((k-1)\). Each vector is used for generating the code word for one participant.

\((R_{11},R_{21},R_{31},\ldots,R_{(k-1)1})\epsilon\mathbb{F}_q^{[1,n]}\).

Group Key Generation and Distribution:
Let the group consist of \(n\) participants \(u_1,u_2,u_3,\ldots,u_n\). The KGC will generate random session
key for every session, it will distribute this common key to communicating parties by generating code word for each participant. The key generation and distribution can be done as follows.

step 1: Initiator sends \( <u_1, u_2, u_3, \cdots, u_n> \) with a key generation request to KGC.

step 2: KGC broadcasts \( <u_1, u_2, u_3, \cdots, u_n> \) as a response.

step 3: Now KGC generates random session key \( s \). KGC calculates code word for every participant in the following procedure.

- Form a vector of length \( k \) where first component is common key \( s \) and remaining components are \( (R_1^i, R_2^i, R_3^i, \cdots, R_{k-1}^i) \forall i \epsilon [1, n] \).
- Reduce the generator matrix in such a way that first column will become identity matrix.
- Multiply this reduced matrix with above vector \( s, R_1^i, R_2^i, R_3^i, \cdots, R_{k-1}^i \). Now the resultant vector is the code word for participant \( i \). That code word is \( C_i = (k, v_{i1}, v_{i2}, v_{i3}, \cdots, v_{i(k-1)}, u_{i1}, u_{i2}, \cdots, u_{in-k}) \).
- Now KGC will privately give \( (v_{i1}, v_{i2}, v_{i3}, \cdots, v_{i(k-1)}) \) to \( i^{th} \) participant and keeps one of the \( u_{ij} \) as the public share of \( i^{th} \) participant.

Reconstruction phase:
Each user can recover common session key \( s \) by using his \( (k-1) \) private shares and any one of his public shares. This can be done by using the following procedure.

- Form a message vector of length \( k \) where the first \( (k-1) \) components are his private shares \( (v_{i1}, v_{i2}, v_{i3}, \cdots, v_{i(k-1)}) \) and remaining one component is his public share.
- If participant public share \( u_{ij} \) then reduce the generator matrix in such a way that second column to \( k^{th} \) column and \( (j+k)^{th} \) column will make identity matrix.
- Multiply the message vector with reduced matrix, in the resultant code word first component is our common session key \( s \).
- The above three steps can be done by all participants to recover the key \( s \).

Verification phase:
In order to verify whether all the communicating parties have received same session key the KGC will calculate \( Auth = g^{id_1+id_2+id_3+\cdots+id_n+s} \) and this value is broadcasted to every participant. After recovering the key \( s \) every participant will calculate this \( Auth \) value and ask remaining people to give their calculated value, if all this values are same that means every participant has received same key \( s \).

Re Keying Operation:
The process of updating the keys, whenever a new member added to the group or an existing member leave the group is called re keying operation. This operation can be done by KGC, whenever it receives a request from initiator. Re keying can be done for two cases.

Member join operation:
When any new member wants to join the group, this can be done as follows.

- First the new member requires to register at KGC.
- Then he has to broadcast the request to all the group members, that he wants to join the group.
- If all the group members are agreed then the initiator of the group will send the group key request to KGC by specifying the names of all the group members.
Now the KGC generates new random session key and distribute it to group members by using above distribution procedure.

**Member leave operation:**
Handling members who leave is more complicated than handling new members. To join a group, a new user needs to broadcast a request. The new user becomes a legitimate group member once its request is approved by any existing group member or by the initiator. However, we cannot assume that a leaving member will send out a leaving notice. A member could leave the group silently. This can be handled by using following procedure.

1. The initiator of the group sends out a member refresh message to all the group members periodically. All group members should send an ack message back to indicate their presence in the group.
2. The initiator will determine the presence of particular group member based on their response within a certain amount of time.
3. If any group member left the group then the initiator will send the new group key request message to the KGC, which contains the names of all the group members except the member who left the group.
4. Now KGC will generate new random session key and distribute it to the group members by using above distribution procedure.

5. **Security Analysis**
In this section we prove that our proposed protocol achieve the security goals of a group key transfer protocol. The main goals of a key transfer protocol include 1) Key freshness 2) Key confidentiality 3) Key authentication 4) forward secrecy 5) backward secrecy. Key freshness is to ensure that the group key never been used. Thus a compromised group key cannot cause any further damage of group communication. Key confidentiality is to ensure that the group key can be recovered by only authorized group members. Key authentication ensures that the group key is generated by KGC and not by an attacker. Forward secrecy ensures that the group member who leaves the group cannot know the further communication details.

There are two types of attacks that exist in a group key transfer protocol; 1) Insider attack, 2) Outsider attack. The outside attacker try to recover the secret group key belonging to group that an outsider is unauthorized to know. The inside attacker is authorized to know the common secret key but he try to recover other persons private keys shared with KGC.

### 5.1. **Outsider attack**
Assume that an outside attacker who impersonates a group member for requesting a group key service, then the attacker neither get the group key, nor share group key with any other members.

**Proof:** Although any attacker can impersonate a group member but they cannot recover the secret key because the secret key can be recovered by using the private shares of a participant these private shares are privately shared between participant and KGC.

### 5.2. **Insider attack**
The inside attacker is one of the participant in the group, who is trying to know about the private shares shared between KGC and other participants. **Proof:** If the protocol run for several times then also the attacker will not predict any thing about the private shares shared between KGC and other participants. Because for every request the KGC will generate the code word for all participants based some random vectors, these random vectors are not shared with anyone. so every time the private shares are generated differently.
6. Conclusions

The group key information is broadcast in an open network environment, and only authorized group members which are part of the particular cluster, can obtain the group key. In this paper, we proposed a group key transfer protocols for group communication. The objective of securing group communication is to provide confidentiality, integrity and authentication security services. Maintenance of this confidentiality requires establishing and maintaining a common key between group members. Proposed group key transfer protocol based on secret sharing scheme using MDS codes.

The major advantage of our scheme is less computational complexity, so rekeying operation can be done easily compared to the schemes based on polynomial interpolation. Our schemes are applicable for any number of participants. They are efficient and require $O(n^3)$, where $n$ is the number of participants, operations.

References

[1.] Harn, Lien, and Changlu Lin. authenticated group key transfer protocol based on secret sharing.” Computers, IEEE Transactions on 59.6 (2010): 842-846.
[2.] McEliece, R. J., Sarwate, D. V., On secret sharing and Reed Solomon codes, Communications of the ACM, 24, pp. 583 - 584, 1981.
[3.] W. Differ and M. E. Hellman, New Directions in Cryptography, IEEE Transactions on Information Theory, vol. IT-22, no. 6, 1976, pp. 64454.
[4.] Ozadam, H., Ozbudak, F., Saygi, Z., Secret sharing schemes and linear codes, ISC, Ankara pp. 101 - 106, 2007.
[5.] Josef Pieprzyk, and Xian-Mo Zhang, Ideal threshold schemes from MDS codes, ICSC, vol. 2587, pp. 253 - 263, 2003.
[6.] Shamir, A. 1979. How to share a secret. Comm. ACM 22, 612613.
[7.] M. Burmester and Y.G. Desmedt, A Secure and Efficient Conference Key Distribution System, Proc. Eurocrypt 94 Workshop Advances in Cryptology, pp. 275-286, 1994
[8.] I. Ingemarsson, D.T. Tang, and C.K. Wong, A Conference Key Distribution System, IEEE Trans. Information Theory, vol. IT-28, no. 5, pp. 714-720, Sept. 1982.
[9.] D.G. Steer, L. Stravinsky, W. Differ, and M.J. Wiener, A Secure Audio Teleconference System, Proc. Eighth Ann. Intl Cryptology Conf. Advances in Cryptology (Crypto 88), pp. 520-528, 1988.
[10.] M. Steiner, G. Tsudik, and M. Waidner, Differ-Hellman man Key Distribution Extended to Group Communication, Proc. Third ACM Conf. Computer and Comm. Security (CCS 96), pp. 31-37, 1996.
[11.] S. Mittra, Iolus: A Framework for Scalable Secure Multi-casting, ACM SIGCOMM, 1997.
[12.] J.J.Simmons., How to (Really) Share a secret, Advances in Cryptology CRYPTO’88, LNCS,403(1990),390-448.
[13.] Nam, Junghyun, et al. "Cryptanalysis of a group key transfer protocol based on secret sharing." Future generation information technology. Springer Berlin Heidelberg, 2011. 309-315.
[14.] H. Bettahar, A. Bouabdallah, and Y. Challa. An adaptive key management protocol for secure multicast. In ICCCN, Florida USA, October 2002.
[15.] M. Bouassida, I. Chrisment, and O. Pestor. Efficient Clustering for Multicast Key Distribution in MANETs. In Networking 2005, Waterlo, CANADA, May 2005.
[16.] A.N. Tentu, P.Paul,V.Ch.Venkiah.,Ideal and Perfect Hierarchical Secret Sharing Schemes based on MDS codes., proceeding of International Conference on Applied and Computational Mathematics, pages 256-272, Ankara, Turkey, 2012.
[17.] A.N. Tentu, P.Paul,V.Ch.Venkiah., Conjunctive Hierarchical Secret Sharing Schemes based on MDS codes, 24th International Workshop on Combinatorial Algorithms(IWOCA),France,Lecture Notes in Computer Science, pages 463-468, Springer Verlag, 2013.
[18.] A.N. Tentu, P.Paul,V.Ch.Venkiah.,Ideal and Perfect Hierarchical Secret Sharing Schemes based on MDS codes., eprint.iacr.org/2013/189.pdf.
[19.] S. Rafaeli and D. Hutchison, “A survey of key management for secure group communication,” ACM Computing Surveys, Volume 35 Issue 3, Sept. 2003.
[20.] Q. Zhang and Y. Wang. A centralized key management scheme for hierarchical access control. In IEEE Global Telecommunications Conference (Globecom’04), volume 4, pages 2067–2071, 2004.
[21.] H. Harney, C. Muckenhirn, and T. Rivers, “Group key management protocol architecture,” IETF, RFC2093, 1997.
[22.] Chingfang Hsu, Bing Zeng, Guohua Cui, Liang Chen, A New Secure Authenticated Group Key Transfer Protocol, Wireless Pers Commun (2014) 74:457-467.

[23.] Ramzi Bellazreg and Noureddine Boudriga, DynTunKey: a dynamic distributed group key tunneling management protocol for heterogeneous wireless sensor networks, EURASIP Journal on Wireless Communications and Networking 2014, 2014:9.

[24.] Ruxandra F. Olimid, On the Security of an Authenticated Group Key Transfer Protocol Based on Secret Sharing, ICT-EurAsia 2013, LNCS 7804, pp. 399-408.