Security In Group Conference Protocol (SGCP)

K. Bhuuvaneswari[1], T. Arunkumar[2]

ABSTRACT

The technology of digital conference has opened up a new area of research and application to computer networks in industry. It can be used in a board meeting, scientific discussion or in virtual classrooms, through the computers connected by IP networks. To protect conversations from eaves dropping, a common conference key agreement protocol is required. Conference key protocol secures the discussion session and data among multiple conferees engaged in common goal of communication. Numerous works have been carried out in providing secured conference, but most of the works concentrate on an efficient key exchange protocol to prevent malicious users to attempt to play the proxy role or delay or destruct the conference environment. This paper proposes a novel approach of unique dynamic ID based key exchange protocol using Diffie-Hellman algorithm, which possesses the property of fault-tolerance secured session, dynamic ID key generation and key exchange methods.

INDEX TERMS: *Conference Key , Security Key Generation, Fault tolerance Secured Session.*

1. INTRODUCTION

Computer network group communication is a group of people who communicate (or) make a conference in an interactive procedure through the computers connected by networks at distance or discrete location [1]. In order to establish a secured, fault-tolerant communication among groups in an open network, the current internet protocols do not come into needy. Secure group communication is an increasingly popular research area, which has been receiving much attention in recent years. As a rapid growth of the internet, the group communication has become an important feature of the internet technology.

Conferencing in IP network, through group communication, is actually transmitting data as broadcast through multiple channels. Group communication is more complicated with regard to the concept of security .As the group starts to mutate (members leave and join at any interval of time), the members of group are not a well defined entity. Hence, security services in group communication or multicast groups are complicated issues to be dealt.

This paper proposes a session based security model SGCP (Secured Group Communication Protocol), which provides security from establishing a session to the closure (end) of session. Secured session is established for variable time slot dynamically among all conference members, implemented by Common Conference Agreement (CCA) methods. The conference member here after called as "Conferee" in this paper. Secured group communication among multiple conferees can be established only if a conferee handles the secured key for each session.

[1] Assistant Professor and Head, Department of Master of Computer Applications, Karpagam College of Engineering, Coimbatore - 32, Tamilnadu, India.

[2] Professor, Department of CSE, School of Computer Science, Vellore Institute of Technology (Deemed University), Vellore, Tamilnadu, India.

Security communication needs to have a conference key agreement [3] for the group communication. Conference key arrangement is a mechanism in which a shared conference key is derived by "conference key engine" which is exchanged among conferee group members who participate in conference. Each conferee member will acknowledge all other conferees, while the conferee "registry" in member system will either accept the key or reject the key. SGCP protocol follows distributed service architecture, which is designed to be one of the components for secured publish / subscribe and exchange communication infrastructure.

This work is quite different from the work carried out by other researchers. Mostly the work carried out by Tzeng [7] would be suitable only for honest group of users but not for malicious users or who intentionally attempt to delay or destruct the conference. Similarly Yongdae. Kim's [8] work adds communication overhead for peer group overhead on a global network setup like internet. An active attack (malicious participant) tries to disturb establishment of a common conference key among the group of honest participants. Passive attacks are carried out on gathering the conference key by listening to the communication of participants.

This protocol works on the basis of only proper secret keys agreed between honest conferees. If any malicious user, who may destruct the conference or delay the operation, is found or suspected, then the malicious user's port is blocked for further operations. Hence, this work is efficient in secret key generation and operations among honest conferees.

The paper is organized as follows: Section 2 discusses on identifying an efficient security key by group key agreement method. Existing research works carried out in this area are also discussed. Section 3 focuses on the model and design, that have been developed using simple key exchange "Diffe-Hellman" algorithm. This work suggests a simple scheme of 2-way Diffe-Hellman (2DH) method and against n-party Decisional Diffe-Hellman (nDH) problem. The efficiency of group communication using Common Agreement based on Diffe-Hellman secured key algorithm is proved. Section 4 discuss on the architecture and implementation of SGCP on IP based network. Section 5 analyses the performance of SGCP and conclude with the need for future work.

2. Related Work

The literature survey finds good number of recent interesting research works that have been carried out on group key agreement by XunYi[9], M.Steiner and M. Waidner [4]. A most all group key agreement protocol can be directly adapted to conference key agreement. However, most of them operate only when all conferees are honest but do not work when some conferee is malicious and attempts to delay (or) destructs the conference.

Most of entity conference key agreement protocols operate only when all conferees are honest, but do not work when some conferees are malicious and attempt to delay (or) destruct the conference. Sometimes the conferees may cause severe damage to the conference setup or break the session in use.

The problem of common key agreement [5] schemes, in dynamic group key agreement, especially in creating a group, has been the steppingstone for all the other securely service schemes. Several schemes on group key agreement have been done in centralized manner[8], where one dedicated party (typically leader of peer group) has to select the group key which will be distributed among other peer groups in a distributed fashion. This method is actually a key distribution or key transfer among the groups and not Key Agreement. This method can be suitable only for static groups not for dynamic groups. Dynamic peer groups require not only the initial key agreement (IKA) but also auxiliary update key agreement (AKA) operations such as member addition, member deletion and other internal group functions.

Xun Yi [9] explained his research work in novel fault tolerant conference key agreement protocol. in which each conferee only needs to send one message to a semi trusted conference bridge and receives one broadcast message . The identity based key agreement is based on elliptic curve cryptography (ECC). It is resistant to the different key attack from malicious conferee and needs less communication cost than Tzeng protocol [7].Comparatively, the conference key agreements are having disadvantages and our new protocol give an efficient way of conference session key agreement.

3. Design and Model

This section provides a detailed overview of notable features of reliable group communication and session key agreement methods. This work adopts the following assumptions and notations.

G = Peer Group { G1, G2, . Gi,…Gn},

Gi → ith group where i ∈ [1,..n]

M = Conferee members in group,

Mi → ith member in a group
where i ∈ [1,… n]

M* = All group members

K = Group key generated

Ki → ith group key where i ∈ [1,..n]

Kn = Group key shared among all 'n' members

α = exponentiation base; generator in algebraic group G delimited by 'q'

q = prime number, order of algebraic group.

W = secret exponent of key agreed by Mi and generated by Conference Manager

H = sub-set of Wi { W1, … Wn}

S = Conferee Session to hold secret key
Si { S1,…Sn}

S*= Multiple sessions with {M1,..Mi}, {Mi+1,…Mk}, {Mk+1,..Mn} conferees.

Session is created when at least two conferees accept a secured key to establish a communication path. Session can be established among multiple conferees (who can be distributed locations) engaged in communication. Session is a virtual communication path established among one or more conferees with a secured key as an entry point.

The peer group communication semantics is defined as follows:

1) A group Gi is an entity, which depicts various conferees engaged in communication using a single communication session / channel.

2) All conferees engaged in conference / communication require independent security key.

3) A group may consist of minimum two conferees at least.

4) Any conferee in a group may leave the group or join the group or rejoin the group at any time of communication process for a session.

Group Key Agreement – It is defined as a comprehensive group key solution which should handle the adjustments to group key secrets. Subsequent to all membership change operations in the underlying group communication system, the following conferee memberships are considered:

i) The system distinguishes among single and multiple group conferee operations.

ii) Keys are defined and invoked independently for single (one-to-one) conferee sessions and multiple group (many-to-many) conferee operations.

iii) The keys assigned to session are unique and dynamic in nature, where type of key and its hierarchy carry importance.

iv) Key generated for each session is entirely new and remains out of reach of former group members.

Secure Key Ki (such that K is the unique key id and 'i' is the order of group) is generated by "conference server engine" which is basically random in generation. The generated key Ki is agreed by both the conferees who negotiated to communicate. Ki is exchanged by conferee members {M1,...Mn} to establish the conference.

3.1 Proofs on Security Model

The key agreement protocol belongs to family of n-party DH scheme, which is an extension of 2-party DH key exchange [9]. The work adopts n-party DDH method among multiple groups Mn who agree a priori on a cyclic group G. Each key Ki, generated randomly from the generator engine 'α' can belong to Wi $\in$ q. The group key K = $\alpha^{W_1,\ldots W_n}$. This protocol works only when conferees are honest, and will break the operation (by blocking user's port) if user is found to be malicious.

A. Single Session Communication

In 2-party DH scheme, Ki is computed by exchanging α^{W1} for conferee M1 and α^{W2} for conferee M_2. Key Ki can be computed as $(\alpha^{W1})^{W2} = (\alpha^{W2})^{W1}$. W1 represents the secure key selected with r1 being the key of conferee M1 and cr1 being the agreed group key assigned for a group. W1 = $R_1 \oplus CR_1$ and W2 = R_2CR_2. The generated key Ki is assigned to session Si.

B. Group Session Communication

In case of multiple group communication where conferee members {M1,...Mn } of Gi, 2-party DH cannot be implemented, hence n-party DH scheme is used. In n-party DH key exchange scheme, a subset of Si = $\{\alpha^{?(H)} \mid H \subset \{W_1, \ldots. Wn\}\}$ is exchanged among conferees. This set includes random number values $\alpha^{W1 \ldots Wn}$, which can compute Ki. Hence Ki is the secret key agreed by conferee members M1,...Mn belonging to Gi. Wi = Ri $\oplus$ CRi, also Wn = Rn CRn . Session based security key Ki is an extension of n-party DH. The key Ki is assigned for session Si.

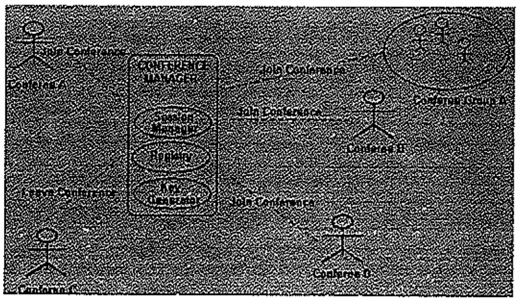

Fig.1. SSGCP Distributed model setup

The key assigned for a session is dynamic. Hence for any random small interval of time the key may be used. Any update of session for change in intervals new key is generated and assigned to session Si.

4. Session Based Secured Conference Key Scheme

4.1 Model and Architecture

SGCP model setup is as shown in Fig-1. Architecture basically consists of three main procedures "Conference Key Generation Engine", "Conference Session Manager" and "Conference Registry" procedures which reside on Conference Server.

1) Key Generation Engine is a trusted key generator which generates the "secured common session key" (CRi) for each session based on DES Scheme [2] and generates individual conferee key (conf_keyi) or multiple group key (gp keyi) based on Diffie-Hellman Scheme [4].

2) Session Manager manages various sessions among conferees in group session or single session as well handles Key Agreement.

3) Registry procedure maintains the complete information of each conferee, that is, their session in use, network port in use and key generated.

Conference key agreement (CKA) is the method of certification of acceptance made by each conferee engaged in conference for a session. The certificate of agreement will be registered in "Conference Server" and in "Registry". CCA method provides the simplicity and flexibility of assigning a common key among conferees which is generated by Conference Key Generation engine and negotiated among all conferees. Conference key agreement method is more secure than Conference Key Distribution (CKD), since the possibility of a malicious user obtaining the secured key is possible. The possibility of deriving the secured key is also possible since, the key generated can be a combination of all conferees engaged in conference.

Fig-2 shows Host architecture of SGCP. Session manager implemented at Conference Server in Conference network enables a secured-session management, start a session and end a session. Registering the conferees for each session as well maintaining a proper synchronization among conferees through session serialization is handled by Registry. Initially each user or a group, is authenticated by logging into conference room. A User can be a conferee only if secured key is allotted by Key-Generator based on key generation and CKA procedures. Conferees can establish communication path with another single conferee or group conferee to form a session. The session established at multiple conferee's end or host of the network manages the secured-session key.

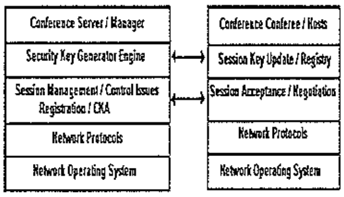
Conference Server / Manager
Security Key Generator Engine
Session Management / Control Issues Registration / CKA
Network Protocols
Network Operating System
Conference Conferee / Hosts
Session Key Update / Registry
Session Acceptance / Negotiation
Network Protocols
Network Operating System

Fig.2. Conference Server / Host Architecture

On mutual agreement session is established with other conferee members or groups. SGCP protocol assumes that server is a computer network server or server module executing in any network nodes, while the conferees or groups reside in multiple end-nodes. Multiple requests and acknowledgments on the network backbone may be altered, blocked, delayed due to various latency effects.

4.2 Security Key Generation

Various types of security keys used in SGC Protocol are discussed in Table-1. Usage of keys is based on type of session requested and number of conferees requested to communicate. The secured keys of conference key protocol generated possess the following properties.

i) Key is unique for a particular group of participants in a pre-distributed conference key protocol.

ii) Secured key will be changed dynamically at unequal time intervals for sessions.

iii) Secured Key generated will be changed for each session update or new session.

iv) The pre-distributed conference key protocol provides lack of flexibility. SGCP uses various keys at each session to maintain uniqueness and independency. In order to increase security for conferees and session, two keys are assigned to conferee for each session or any session update.

1. A common key is allocated (CRi) by Key Generator, which is generated by DES scheme, which will be used by all conferees in a session, but unique to a session. Each session in Registry is identified by CRi.

2. Each individual conferee is assigned a key (Ri) by Key Generator. This key is generated by Diffe-Hellman Scheme. Hence each conferee is assigned a session key Si (discussed in Section-3) to be agreed by another conferee with session key Sk to establish a communication path in a session.

4.3 Secure Keys Used In SGCP

Table-2 shows the list of various security keys used in SGCP scheme. Public Key **pb_key** is assigned to all users who have joined the conference room. User holding this key can view the list of conferees who are actively in live communication but they cannot participate nor view the contents. Both "conferees", "check" required a key to "establish" a connection. Each move at random time interval, "session-negotiation" and "session–key–update" is carried out. If the expected key state is not identified, then "session-expires" and a "session-reconnect" has to be requested.

4.4 Session Protocol Operations

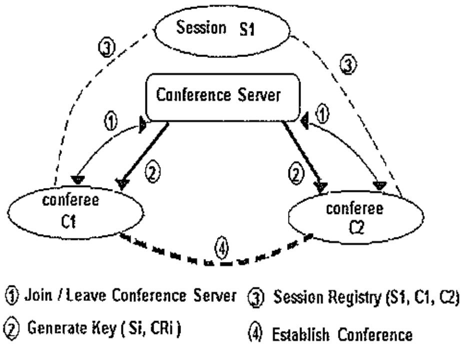
Session S1
③
③
Conference Server
①
①
②
②
conferee C1
conferee C2
④
① Join / Leave Conference Server
③ Session Registry (S1, C1, C2)
② Generate Key (Si, CRi)
④ Establish Conference

Fig 3 SSGCP Phases

4.5 SGCP Protocol

The protocol runs in four different phases ie., Join / Leave Conference Server, Key Generation, Session Registry in Server and Create – Establish - Manage session as shown in Fig-3.

In the first phase, user can join the conference server through a simple user authentication method. During login process, user generates a random value, which may be obtained from keyboard buffer. The generated value with login time, port selected for communication from random available ports together is considered as combination of secured conferee key CK1, CK2 (conf_key) as shown in Fig-4

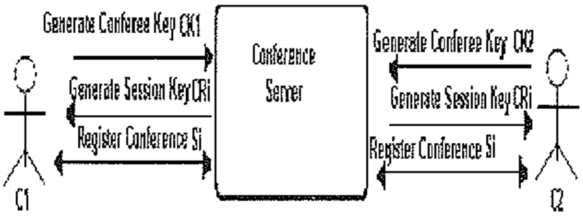

Fig 4. Key Generation Phase

In second phase, conference server generates a random key (ss_key) CRi. CKA method negotiates with Registry module in Conference Server and registers the session conference key. Third Phase focuses on creating the session Si, and managing the session among various conferees (C1,C2,..Cn). Session update is performed at frequent time intervals of few milliseconds, invariably time interval is not consistent. Fourth phase deals with virtual connection establishment among multiple conferees or multiple groups.

5. Implementation

Performance measures of SGCP were carried out by testing on four different 100Mpbs LAN based Ethernet network. The deployment comprises of four servers running on separate domains, so the system was capable of tolerating a single compromised server. The measurements were gathered separately on client systems and server of LAN network. Round-trip times for ICMP echo packets typically measured within 100ms, such that network delays are observable. The hosts and network are relatively quiescent during the experiment; client was executed on separate machine and its signal processing and latency times are also observed.

Experimental test-bed helps to identify the mean execution times between various three conference groups created for test purposes. TCP based signal methodology for Conference_Request and Conference_Reply procedures have been implemented in Java based EJB development. The setup has helped to identify the fatigue property of conference session manager due to malicious user or intruder trying to disturb the conference setup. Malicious user calls from five network stations have been tested to attempt access rights for conference server and disturb conferencing.

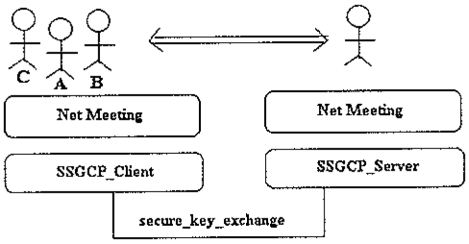

Fig-5 SGCP Session Implementation

Two tests were conducted to determine the overhead of using secure session key generation / control and dynamic certificate agreement in SGCP. These test procedures include raw processing time per routing packet for varying key sessions established and measurements of average route acquisition latency. The RTT time between establishments of conference between multiple conferees and group conferees have discussed. Table-3 shows our results. Results were conducted over two Ethernet of 10Mbps LAN networks.

Multiple call sessions were handled at an instant over three different groups. Test was carried out over Microsoft's NetMeeting installed over Java modules SSGCP_Client and SSGCP_Server. Java modules reside at each client (SSGCP_Client) and server machines (SSGCP_Server) in Fig.5. Java module creates secured key, assigns to user session and maintains the secured session.

Table-3 shows call established between two different LAN networks. Calls were established over SGCP scheme and Microsoft's Net Meeting Scheme. SGCP handles better secured session management, number of sessions established at a time is higher compared to Net-Meeting where only maximum of five sessions were allowed. Security method is common in Net Meeting.

6. Conclusion

SGCP scheme, implemented based on Diffie-Hellman key exchange algorithm, achieves secure and efficient key agreement in the context of one-to-one conference scheme and group conference communication. The scheme works well for small groups as well for number of groups not greater than 100. For very large groups of intentionally large the scheme may not appropriate in selecting unique key distributions. SGCP proves confidentiality of conference shared and authentication. In general this architecture elucidates on secure key generation, key management, control, session control and fault tolerant aspects. Hence data is secured with transaction and processing parameters. The future work stresses on the need for a secured session system for large conference networks, which should be extended to scalability and load balancing.

Table1- Session Protocols

1. Session_Request	To request for a conference to conferee initially.
2. Session_Ack	To acknowledge request (True / False), with request for conference type.
3. Session_Key_Register	To acknowledge type of conference and session key to Conference Manager.
4. Session_Establish/ Update	To establish session for conference with conferees.
5. Session_Negotiate	To check and negotiate on keys for existing conference.
6. Session_Expiry	To indicate that conference session is broken and key has to be generated.
7. Session_Alert	To indicate that session can be broken, due to an intruder malicious use.
8. Session_Close	Bye. {To end conference}
9. Session_Key _Update	Conferee check on Server as request to update session.
10. Issue_Session_State	To identify the state of a conferee session at a time interval and inform as well update on conference.

Table 2 - List of Security Keys

Conf_key	Conference key for individual conferee engaged in conference.
Ss_key	Session conference key for each group or individual conferee engaged in conference. Involves multicasting / broadcast
Gd_key	Conference key generated for multiple peer groups
Dgp_key	Distributed group key on conference. Engages various groups of members in multiple groups on conference.
Pb_key	Public key assigned for all members not engaged in conference.

Table-3-Call conference setup

SI No.	**Sender IP Address**	**Receiver IP Address**	**Type Of Conference Call**	**No of Sessions in Use**	**SGCP (with NetMeeting)**		**NetMeeting**	
					Time taken to establish / call (ms)	**Session Interrupted On Control**	**Time taken to establish / call (ms)**	**Session Interrupted On Control**
1	192.168.67.34	192.168.65.30	Individual Audio	22	190	20	320	10
2	192.168.67.30	192.168.65.31	Group Audio	20	178	20	486	12
3	192.168.67.31	192.168.65.32	Group Video	18	186	17	310	9
4	192.168.67.32	192.168.65.33	Group Video	22	223	21	354	11
5	192.168.67.33	192.168.65.34	Group Video	24	187	24	323	9
6	192.168.67.36	192.168.65.35	Group Video	23	203	22	329	12

7. References

[1] D.Reed. " A Discussion on Computer Network Conferencing" RFC:1324.Network working group May 1992.

[2] FIPS PUB 197, "Advanced Encryption Standard", Federal Information Processing Standard Publications" US Dept. of commerce/N.I.S.T, Nov 2001.

[3] C.E Shannon "Communication Theory of Secret systems "Bell systems Technical.J., Vol 28 No:4 PP: 656-715. 1949.

[4] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman Key Distribution Extended to Groups" Third ACM Conf. Computer and Comm. Security , PP 31-37 , March - 1996

[5] Michael steiner, Gene Tsudik "Key agreement in Dynamic peer groups" IEEE, Transactions on parallel and distributed systems vol 11 No :8 August 2000.

[6] G. Ateniese , M. Steiner and G.Tsudik, " New Multiparty Authentication Services and Key Agreement Protocol," IEEE. J, Selected Area in Comm. Vol.18, no.4, pp:628-629 Apr 2000.

[7] W.G.Tzeng, "A Secure Fault Tolerant Conference Key agreement Protocol," IEEE Trans. Computers, Vol. 51, No 4. pp. 373-379 Apr 2002.

[8] Y. Kim, A. Perig, and G. Tsudik, "Group Key Agreement Efficient in Communication," IEEE Trans. On Computers, vol 53, No.7,pp.905-921, July 2004.

[9] Xun Yi "Identity Based Faulty-Tolerant Conference Key Agreement", IEEE Trans. on Dependable and Secure Computing, Vol.1, No 3 , July-Sep 2004.