Secure Group Key Agreement with Node Authentication

Anurag Singh Tomar, Gaurav Kumar Tak, Manmohan Sharma

Abstract— **Group key is used to provide the confidentiality among the group members. Flexible Robust Group Key Agreement scheme has been proposed, in which users contribute to generate the group key. However, it doesn't authenticate the members of the group when data arrives. Previously Efficient Authentication Protocol for Virtual Subnet has been proposed. However, there is no group key agreement phase in it. In this paper, group key agreement with node authentication scheme has been proposed. It's a modified version which combines the features and merits of both Flexible Robust Group Key Agreement as well as Efficient Authentication Protocol for Virtual Subnet protocol. The main advantage of proposed scheme is that it eliminates the need to send the separate parameters for authentication as well as group key contribution.**

Index Terms— **Authentication, Group Controller,Multicast.**

I. INTRODUCTION

IP multicast is used in group based application to use the bandwidth in efficient manner. In unicast communication sender transmits separate packet to each of receiver. In multicast, sender transmits only single packet to network element such as multicast router or switch. Upon receiving multicast packet, network element replicates and forwards the packet to each of the group members. Group based applications like video conferencing, stock data distribution, online chat rooms, online gaming etc. use IP multicast.

There are number of security issues related to multicast data distribution. Most prominent security risks from a user point of view are related to confidentiality, authenticity and integrity. Common group key is needed to provide confidentiality and authentication among the group members in efficient way. It must be updated (rekeying) whenever change in membership occurs i.e, either a new member joins the group or the member leaves the group. At the same time both forward and backward security must be guaranteed. Forward security means that evicted members can't determine any future group key; similarly, backward security means that newly added members can't determine any previous group key. Real-time applications such as distant learning secure audio and visual broadcasts, video streaming, pay TV, secure conferencing, controlling access to broadcast satellite services, collaborative work, online gaming and so on needs very fast rekeying so that changes in group membership are not disruptive.

Manuscript received April, 2014.
***Anurag Singh Tomar**, Lovely Professional University,Phagwara,Punjab,India,*
***Gaurav Kumar Tak**, Lovely Professional University,Phagwara,Punjab,India, ,*
***Manmohan Sharma**, Lovely Professional University, Phagwara,Punjab,India,*

The secure group key distribution over insecure channels has been an active research topic and it is critical issue for the purpose of security as the key should distribute only to legitimate members. Several group key management protocols have been proposed [8, 5] and they can be classified into three categories; centralized, decentralized, and distributed. In centralized group key protocols, single entity called centralized entity/group controller which generates the group key. Group controller is employed to control the whole group and responsible for group rekeying. In the decentralized approach, multiple entities are responsible for managing the group as opposed to a single entity. In the distributed method, each group member contribution is considered to generate the group key and each of them is equally responsible for the rekeying as well as distribution. Rekeying can be done either by periodic or whenever membership change occurs. To accomplish rekeying quickly for every member in the group, it is crucial to avoid the inefficient strategy to send a rekeying message to individual member.

Rest of the paper is organized as follows. The related work described in section II. Proposed scheme explained in section III. Security analysis of the proposed scheme has been discussed is in section IV. Section V concludes the paper.

II. RELATED WORK

Various group key management schemes have been proposed that fall under the different categories of architectures like centralized, decentralized, subgroup and hierarchical [8].

A. Centralized Architecture

A central controller or group controller manages the whole group. Its functions are access control, membership control and key distribution to entire group. Major drawbacks of centralized architecture are a) whole group affected when the centralized controller goes down. b) Single entity cannot control the larger group. Best example for centralized architecture is Group Key Management Protocol (GKMP) [3, 4]. In this architecture, each user first registers at the Group Controller (GC) by sharing its own Key Encryption Key (KEK). GC generates the Group Key Packet (GKP) that contains the Group Key Encryption Key (GKEK) as well as Group Traffic Encryption Key (GTEK). Rekey is needed whenever group membership change occurs. GC generates new GKP and encrypts it by GKEK to distribute to all the group members. When any new user wants to join the group, they have to send a join request to GC and register its own KEK at GC. Upon receiving a request, GC generates a GKP and encrypts it by new member KEK that is shared between GC and joining member and sends to new member. Furthermore, it encrypts the GKP with the old GTEK and multicast the message to all the group members.

ISSN: 2278 – 1323

When a member leaves the group, they will send the leaving request to GC. After receiving the exit request, GC sends the new GKP encrypted by each member KEK. Hence, encryption message overhead is O(n). Moreover, GC stores each member KEK which increases storage overhead. Communication overhead and storage overhead is more in centralized architecture.

B. Hierarchical Architecture

In this architecture group members are arranged at the leaves of tree and internal nodes of key tree represent the KEK that is used to securely deliver the group key to members of group when membership change occurs in the group. It reduces the communication overhead. Ex Logical Key Hierarchy (LKH), One Way Function Tree (OFT).

In Logical Key Hierarchy (LKH) architecture [6, 8], GC maintains a tree of keys, these keys are Group key, KEK and Individual member key. Leaf node represents the individual member key, internal node signifies the KEK and root node of tree is Group Key. Each group member knows the keys from leaf node to root of tree. The GC changes the group key when there is change in the membership in order to provide forward and backward secrecy. These KEK used to deliver the group key to legitimate members of group. Thus, storage overhead at each member is O(log n) and communication overhead at most 2(log n).

One Way Function Tree (OFT) has also been proposed [7], which reduces the communication overhead from 2(log n) to (log n). In case of one way function tree, GC provides the information to group of members, with the help of that information; members will compute the Key Encryption Key independently. Before joining the group, first each group member registers its own secret key at the GC then Group Controller computes the hash of the key and sends the computed hash digest to its sibling node. Upon receiving the digest to its sibling node, each member also computes the hash of its own key and performs the XOR operation to compute KEK. Similarly, each member and controller computes the KEK. Like that same procedure is repeated until they compute the group key. Member computes the key from leaf node to root of tree by the following equation $K_i=f(g(k_{left(i)}), g(k_{right(i)}))$ where left(i) and right(i) denotes left and right children of node i respectively, f is the XOR function and g is the one way hash function and K_i is the KEK.

C. Subgroup Architecture

Large group can't be managed by single entity so large group is divided into small subgroups and each subgroup is managed by different subgroup manager. Data translation from one subgroup to another subgroup becomes a problem in subgroup architecture. Ex Iolus.

Iolus Architecture proposed [2, 8] for group key management and the same is as shown in figure 1. It divides the large group into smaller subgroups and each subgroup is managed by another subgroup manager called Group Security Agent (GSA). These GSA of all subgroups form another group in which members are only these GSA that is controlled by another group GSA. Root of the tree called Group Security Controller (GSC) and others called as Group Security Intermediaries (GSI). Each subgroup has its own independent subgroup key which is different from other subgroup key. Members of the top level group are the subgroup managers corresponding to each subgroup (called as GSI) share a key for that top level group. Whenever any member of subgroup wants to communicate among all the members of different subgroups, then firstly, member will encrypt the data by its own subgroup key and multicast the same encrypted data to the subgroup. After receiving the encrypted data, GSA of that subgroup decrypts the received data and again encrypts it by top level group key that is shared among the subgroup manager of each subgroup. Then it multicasts that next encrypted data to top level group so each of GSA receive the encrypted data and decrypt it and further encrypt it with its own subgroup key and send to all the members. When there is change in membership that does not affect the entire group, only that subgroup will be affected where changes occurs. However, main disadvantage is more computation and that data translation.

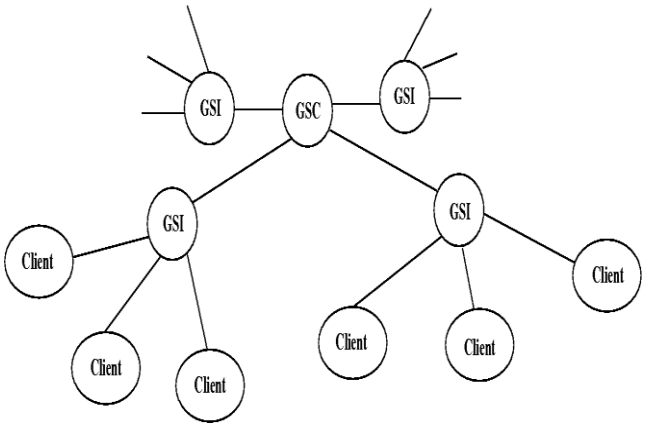

Figure 1. Framework of Iolus

Efficient Authentication protocol for virtual subnet has also been proposed [1]. In this authentication protocol, members of the group can authenticate each other. However, there is no group key agreement. In flexible robust group key agreement [9], user generates group key without authenticating each other. In this work, modified scheme has been proposed with combining the merits and features of [1, 9]. Firstly, member authenticates other members prior to generate the group key. If member passes the authentication phase, then only member contributor is consider to compute the group key.

III. PROPOSED SCHEME

In this paper, group key agreement with node authentication has been proposed. Any group member wants to contribute in group key agreement, firstly, the member will multicast the certificate parameters to his group so that any group member can authenticate any that group members. After successful authentication, member computes the contribution of the sender with these certificate parameters to compute the group key. Proposed scheme works in five phases- group initialization phase, joining to group, certificate generation, certificate verification, group key agreement.

A. Group Inilization Phase

In this phase, GC first choose some parameters for the group such as two large prime number p, q and calculate the value of N =p*q and randomly select $g \in Z_N^*$ where g is the generator of the group such that gcd(g, N) = 1. GC also

calculate the public and private key using RSA algorithm $e \times d \equiv 1 \bmod Ø(N)$. Public and private parameters are (N, g, e) and 'd' respectively.

B. Joining to Group

New user sends a join request to GC whenever they wish to join the group. GC choose the random values like a_i, a_j corresponding to each user such that $\gcd(a_i, a_j) = 1$ and $\gcd(a_i, N) = 1$. In addition, it also calculates the private secret for each member $PS_i = g^{\prod aj} \bmod N$, $j \neq i$ and one public value for the group $PV = 1/ g^{\prod aj} \bmod N$. GC sends the (PS_i, a_i, k) to each member of the group where k is number of members in the group. Group members are arranged in the form of ring such that $M_1 = M_{n+1}$ and each member computes $z_i = g^{ti} \bmod N$ where t_i is the random value chosen by each of the group member. The computed z_i is multicast to the entire group.

C. Certificate Generation

Any member can compute the certificate parameter as per wish. Certificate parameter computation is as follows

$X_i = (Z_{(i+1)} / Z_{(i-1)})^{ti} \bmod N$

$Y_i = (Z_{(i+1)} / Z_{(i-1)})^{ti*(a_i)^{-1}} * PS_i \bmod N$, where $(a_i)^{-1}$ is the multiplicative inverse of (a_i)

$C_i = h(a_i, X_i)$

After computing these parameters, member multicasts the certificate parameters (Y_i, C_i, a_i) to the entire group.

D. Certificate Verification

After receiving the certificate parameters, member can authenticate other member. Each member performs the following steps to authenticate the other member

$X_i^1 = Y_i^{ai} * PV \bmod N$

$C_i^1 = h(a_i, X_i^1)$

If $C_i^1 = C_i$ then member is authentic.

E. Group Key Agreement

Every member computes the Group key K_G as $= (Z_{i-1})^{nti} * (X_i)^{n-1} * (X_{i+1})^{n-2} * \text{----------} * (X_{i-2})$

$K_G = g^{t1t2+t2t3+t3t4+\text{-------}tkt1}$

IV. SECURITY ANALYSIS

In this section, security of proposed scheme is analyzed and it resists the following attacks.

A. Perfect Forward and Backward Secrecy

As the proposed scheme is group key agreement scheme so to compute the group key, every member contribution is required. Whenever group member sends the joining or leaving request to GC, the GC again computes new secrets for every members of the group and each member once again calculates the certificate parameters for authentication. First member authenticates other member in the group whenever the data arrive in it. The received data is processed only when the node is authentic else it is dropped out. Each member is able to calculate other member X_i and the same is used to calculate the group key. Member who left the group can't compute the new group key similarly new member can't compute the old group key. In addition, member can't compute the old group key by using the present group key. Hence it maintains perfect forward and backward secrecy.

B. Replay Attack

The GC computes new secrets for every member whenever change in membership (joining and leaving request) occurs. So attacker can't replay the intercepted data or data used in previous session to next session as the every time parameters are calculated by new secrets provided by GC.

C. Denial of Service Attack

If an attacker modifies the certificate parameters and multicast to the group, it is detected during certificate verification phase. As before considering the member contribution to compute the group key, then authenticity of the member has been checked. Hence Denial of service attack is not possible.

D. Impersonation Attack

During the calculation of certificate parameter $Y_i = (Z_{(i+1)} / Z_{(i-1)})^{ti*(ai)-1} * PS_i \bmod N$ members should know about the PS_i and a_i to calculate the certificate parameters that is known only to authentic group member. So attacker can't compute the correct Y_i. Attacker will calculate these certificate parameter without knowing the value of PS_i, a_i and multicast to members then in certificate verification phase member will able to identify that these parameters have not been sent by authentic member. Hence impersonation attack is not possible.

E. Outsider Attack

Outsider(other than the group member) calculates its own fake certificate parameters X_i, Y_i, C_i without knowing the value of private secret PS_i so in the certificate verification phase the value of $X_i^1 = Y_i^{ai} * PV \bmod N$ can never be equal to X_i because there is relation between private secret PS_i and public value PV. In PS_i we are not considering its own a_i while in PV we are considering all a_i. As $X_i^1 \neq X_i$ so $C_i^1 = h(a_i, X_i^1)$ will not be equal to $C_i = h(a_i, X_i)$ so outsider can't take part or can't give your contribution to calculate the group key hence outsider attack is not possible.

V. CONCLUSION

In this paper, Secure Group Key Agreement scheme with Node authentication has been proposed. Every member in the group contributes to generate the group key. Firstly, Node verifies the authenticity of other node before considering the received data. It considers the received data to compute the group key only when node passes the authentication else it rejects the data. Proposed scheme is secure against Outsider Attack, Impersonation Attack, Denial of Service Attack, Reply Attack, Perfect Forward and Backward Secrecy.

REFERENCES

[1] Chuan-Wang Chang, Ching-Hung Yeh and Chen-Da Tsai, "An Efficient Authentication Protocol for Virtual Subnets on Mobile Ad Hoc Networks" International Symposium on Computer, Communication, Control and Automation 2010.

[2] Suvo Mittra, "Iolus : A Framework for Scalable Secure Multicasting" ACM SIGCOMM 1997.

ISSN: 2278 – 1323

[3] H. Harney and C. Muckenhirn. "Group Key Management Protocol (GKMP) Architecture". July 1997, RFC 2093.

[4] H. Harney and C. Muckenhirn, "Group Key Management Protocol (GKMP) Specification" July 1997, RFC 2094.

[5] S.Jabeenbegum, Dr. T.Purusothaman, Karthi.M, Balachandar.N, Arunkumar.N. "A Cluster Based Cost Effective Contributory Key Agreement Protocol For Secure Group Communication" Second International conference on Computing, Communication and Networking Technologies 2010.

[6] Harnay, Hugh and Eric Harder, "Logical Key Hierarchy Protocol", Internet Draft, Draft-harney-sperta-lkhp-sec-oo.txt, Internet Engineering Task Force(March 1999),22 pages.

[7] Alan T. Sherman, David A. Mcgrew, "Key Establishment In Large Dynamic Groups Using One-Way Function Trees" IEEE Transactions On Software Engineering, VOL. 29, NO. 5, MAY 2003.

[8] Shu-Quan Li, Yue Wu, "A Survey on Key Management for Multicast" Second International Conference on Information Technology and Computer Science 2010.

[9] Stanislaw Jarecki, Jihye Kim and Gene Tsudik, "Flexible Robust Group Key Agreement" IEEE Transcations on Parallel and Distributed Systems 2011.

Anurag Singh Tomar is currently working as Assistant Professor at Lovely Professional University,Phagwara,India.He received Master of Technology from ABV-Indian Institute of Information Technology, Gwalior.His area of research are Network Security,Cloud Computing,,Mobile Ad Hoc Networks.
.

Gaurav Kumar Tak is currently Assistant Professor in Lovely Professional University,Phagwara,India. He received Master of Technology Degree and Bachelor of Technology Degree from ABV-Indian Institute of Information Technology,Gwalior He had published 20+ international publications in reputed international journals and conferences including IEEE,springer, ACM, Science-Direct. He also guided several master thesis.His area of research are Network Security,Cyber Crime and Security,Mobile Ad Hoc Networks.

Manmohan Sharma is currently Asst. Professor at Lovely Professional University, Jalandhar, India. He received Master of Technology From Guru Nanak Dev University,Amritsar,India. His research area is Information Security, Wireless adhoc network.

