International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 7, July - 2013

Efficient Algorithm For Secure Group Communication

Mayuri Lingayat
ME[CNE]

Prof V. R Chirchi
Asst Prof.,COE,Ambajogai

Abstract— ***With the development of the Internet, Multicast applications are deployed for mainstream use, IP multicasting is critical technology in those applications. The absence of security mechanism has limited the use of multicast. In order to protect communication confidentiality, Traffic in secure multicast is encrypted with a Session Encryption Key which is only to the certificated group members. Key management becomes essential issue for IP multicast. The aim of key management for multicast is for group members in one multicast session to generate, refresh and transfer keys which are used for encryption and authentication. Issues about scalability, reliability and robustness are important criteria's for evaluating key management scheme.***

1. Introduction

MULTICAST is a preferred communication model when an identical message has to be delivered to multiple intended receivers. Multicast communication reduces overheads of the sender as well as the network medium.

IP multicast is used to distribute data to a group of receivers efficiently. A Datagram addressed to the multicast group, identified by a Class D IP address, will be delivered to all group members. Efficiency can be achieved because datagrams need to be transmitted once and they traverse any link between two nodes only once, saving the cost of sender as well as network provider.

Many Internet applications, such as stock quote updates, newscast, multiparty conferences, and military communications, and distributed gaming, can all benefit from multicast communication. Most of the commercial models have a single sender and multiple receivers.

In most of these applications, users typically receive identical information from a single or multiple senders. Hence, grouping these users into a single multicast group and providing a common session encryption key to all of them will reduce the number of message units to be encrypted by the senders.

Securing group communications or computations leads to challenging problems such as maintaining communication integrity in the presence of group membership changes, establishing source authentication, and minimizing key storage size and number of update messages at the senders as well as the receivers.[3]

2. Group Key Management Scheme

Group communication can benefit from IP multicast to achieve scalable exchange of messages. However, there is a challenge of effectively controlling access to the transmitted data. IP multicast by itself does not provide any mechanisms for preventing non group members to have access to the group communication. Although encryption can be used to protect messages exchanged among group members, distributing the cryptographic keys becomes an issue.

2.1. Requirements of Key Management in Multicast:

The main job of key management in multicast is generating, distributing and updating group key for its member. Group key is held by all members and used for encryption and decryption datagram. The messages are protected by encryption using the chosen the group key. Only those who know the group key are able to recover the original message. The group key need to be refreshed when the groups have membership change. However, distributing the group key to valid members is a complex problem.

The basic problems which key management should solve as following [1,2]:

(1) Forward secrecy: that is used to prevent a leaving or expelled group member to continue accessing the group's communication. If the key is changed as soon as a member leaves, that member will not be able to decipher group messages encrypted with the new key.

(2) Backward secrecy: that is used to prevent a new member from decoding messages exchanged before it joined the group. If a new key is distributed for the group when a new member joins, it is not able to decipher previous messages even if it has recorded earlier messages encrypted with the old key.

(3) Collusion: Evicted members must not be able to work together and share their individual piece of information to regain access to the group key.

(4) Computation efficiency: Good key management scheme should keep balance between cryptographic processing performance and key management structure.

(5) Storage efficiency: The key management should be very efficiency on storing key material for group members and key server.

3. Security Issues In Multicast:

The objectives of a multicast security infrastructure are simple, preserve authentication and secrecy for all group communication so that only registered senders can send packets to the group and only registered receivers can read packets sent to the group.

Due to the lack of network-level access control in the Internet, enforcing message secrecy for a multicast group requires data encryption. This requires a group key management solution to distribute and maintain cryptographic keys with registered group members. Similarly, cryptographic authentication schemes are necessary to ensure that registered receivers can verify that received packets come from registered senders. Most research on secure group communication has focused on the architecture of secure groups and the problem of group key management, however, recent research has also focused on efficient packet authentication

An Efficient Group keying (EGK) scheme can be used to achieve non-colluding, storage-communication optimal group key management. Moreover, EGK supports dynamic subgroup communication and each member can setup a secure conference with other members in an ad hoc way.

In EGK, a group controller (GC) is responsible for key generation and distribution and the group data are encrypted by a GK. Each group member (GM) is assigned a unique n-bit ID. For each GM, GC also generates and distributes a set of n = logN secrets, which are one-to-one mapped to the bits in the GM's ID. Note that, although different GMs may share common bits in their IDs, the pre-distributed secrets are different generated by using different random numbers. As a result, different GMs cannot combine their secrets that are masked by different random numbers. The set of pre-distributed secrets is denoted as a GM's private key.

Whenever GMs are removed from the group, GC will multicast an encrypted key-update message. Only the remaining GMs are able to recover the message and update GK as well as their private keys. To achieve storage-communication optimality, we use the similar method of Flat Table scheme.

A minimized boolean function in the form of sum-of-product expression (SOPE) is calculated based on the IDs of remaining GMs, in order to minimize the number of encrypted key update messages. A remaining GM can combine n pre distributed secret shares in his/her private key to decrypt a key-update message.

EGK is the first work that achieves storage-communication optimality with constant message size and immune to collusion attack. It outperforms existing group key management schemes in terms of communication and storage efficiency. We must note that in [8], the authors utilized the ciphertext policy attributed based encryption (CP-ABE [7]) scheme to implement FT so that it is secure against collusion attack. As mentioned in [8], the size of each message is very large and grows linearly on the number of attributes in the access policy [8], [7]. In EGK, the message is substantially reduced to a constant size.

Based on the storage-communication-optimality, EGK also supports dynamic subgroup communication efficiently. In existing MGKD, group members can only participate in the overall group communication protected GK, which is distributed in one-to-many manner. EGK allows each GM to initialize a subgroup communication with any subset of GMs in many-to many manner. The number of required messages is minimized. Only GMs within this subgroup can securely communicate with each other. Overall, the main contributions of EGK are presented as follows:

- With any number of removing GMs, the number of encrypted key-update messages is information theoretically minimized to O(logN).
- The size of each message in encrypted key-update message is constant.
- The communication overhead of adding GMs is O(1),i.e., only one multicast message is required.
- The storage overhead of GC and GM is O(logN) even if GC does not store IDs of GMs.
- EGK is collusion resistant and provides forward and
- backward group key secrecy.
- EGK supports dynamic subgroup communication efficiently.

4. Constructions Of EGK:

A. Group Setup

We describe how the GC sets up the multicast group. First, GC chooses a bilinear group G0 of prime order p with generator g. Also, GC chooses a publicly known oneway function H. Then, it chooses two non-trivial random numbers $\alpha, \beta \in Z* p$. For simplicity, we can map the universe of bit-assignments U to the first |U| members of $Z*$ p, i.e.,the integers 1, 2, . . . , |U|. Finally, for each bit-assignment.B ∈ U, GC chooses a non-trivial random number yB ∈ Zp. We denote this set of 2n random numbers by YB = {yB0, yB0, . . . , yBn−1, yBn−1}.GC publishes the group public parameter GP = {G0, e, g,H}. The group master key: MK = {β, gα, gβ, e(g, g)α, YB} is well protected by the GC.

B. GM Joining and Key Generation

When a new GM u joins the group, u needs to set up a secure channel with the GC using either a pre-shared key or public key certificates. GC then checks whether the GM is authorized to join in the group. Once the checking is passed, GC assigns a unique ID and a set of bit assignments Su to u. Once u is admitted to the group, GC runs key generation algorithm KeyGen(MK,Su) to generate private key SKufor u, where MK is the group master key and Su is the set of bit-assignments in u' ID. The algorithm first chooses a non-trivial random number $r \in Z* p$. Then, it computes $g\alpha+r\beta$. Finally, for each bit-assignment $B \in Su$, the KeyGen()algorithm calculates a blinded secret share gryB. The outputted private key SKu : {D = $g\alpha+r\beta$, $\forall B \in Su$: DB = gryB}.

If u is the first GM in the group, GC will generate an initial SEK and sends the private key {SKu, SEK} to the new GM u through a secure channel. If u is not the first joining GM, to preserve backward secrecy, GC generates another random key SEK_ and multicast {SEK_}SEK. Each GM other than u can decrypt the message and replace SEK with SEK_.

Finally, GC sends {SKu, SEK_} to the new GM u through a secure unicast channel. In the join process, besides the unicast communication, GC only needs to multicast one message, i.e.,{SEK_}SEK. Thus, the communication overhead for GMs join is O(1).

C. Encryption and Decryption

We present how GC can encrypt a message with a set of bit-assignment S, so that only GMs whose IDs satisfy S can decrypt the message. For example, in a three-bit-ID group, if a ciphertext is encrypted by using bit-assignment S = {B0,B1},GMs with IDs 010 and 011 can decrypt the ciphertext.

1) Encryption: ENC(GP,MK,S,M) encryption algorithm takes inputs of the system master key MK, the group parameter GP, a set of bit-assignment S, the message M, and returns the cipher text CT. In EGK, only the GC can perform ENC since MK is the system master key. Given the set of bit-assignment S, it is easy to find a YS = _B∈S yB. For example, if S = {B0,B1,B2}, YS = yB0 + yB1 + yB2 .

After calculating YS, the ENC() algorithm generates a non-trivial random number $t \in Z* p$. Then, the algorithm computes C0 = Me(g, g)αtYS , C1 = gβtYS , C2 = gt. Thus,the ciphertext is as: CT : {S,C0 = Me(g, g)αtYS, C1 =gβtYS, C2 = gt}.

2) Decryption: On receiving the CT, GMs whose ID satisfied the bit-assignment S associated with the ciphertext,can decrypt the CT by performing decryption algorithmDEC(GP, SK,CT).

The DEC() algorithm first checks whether the GM u is eligible to decrypt the message by testing whether $Su \subseteq CT.S$,where CT.S represents the bit assignments associated with the ciphertext CT. Then, for each bit assignment $B \in CT.S$,the algorithm use u's pre-distributed secret shares DB = gryB to compute F = _B∈CT.S gryB = gr_B∈CT.S yB = grYCT.S .

Next, the algorithm computes A1 = e(C1,D) = e(g, g)(α+r)tYCT.S and A2 = e(C2, F) = e(g, g)rtYCT.S .

Then the algorithm divides A1 by A2 and get A3 = A1/A2 = e(g, g)αtYCT.S , which blinds the plaintext in ciphertext.Finally, the algorithm unblinds the ciphertext by calculating C0/A3 = M. [4]

5. RESULTS AND CONCLUSIONS:

We run several simulations under Linux, using the network simulator NS2 version ns-allinone-2.26. The simulation environment is composed of:

area: 500*500 meters.

number of nodes :50 - 100.

simulation duration: 1000s.

Comparing EGK with LKH in the group size of 100 GMs, and the number of messages required are shown in Figure 1, respectively. In the comparison,

we consider the cases of 5%, 25%, 50% IDs are not assigned (i.e., do not care value). As a comparison, the message number curve of LKH is also plotted. We can see that EGK performs better than LKH .

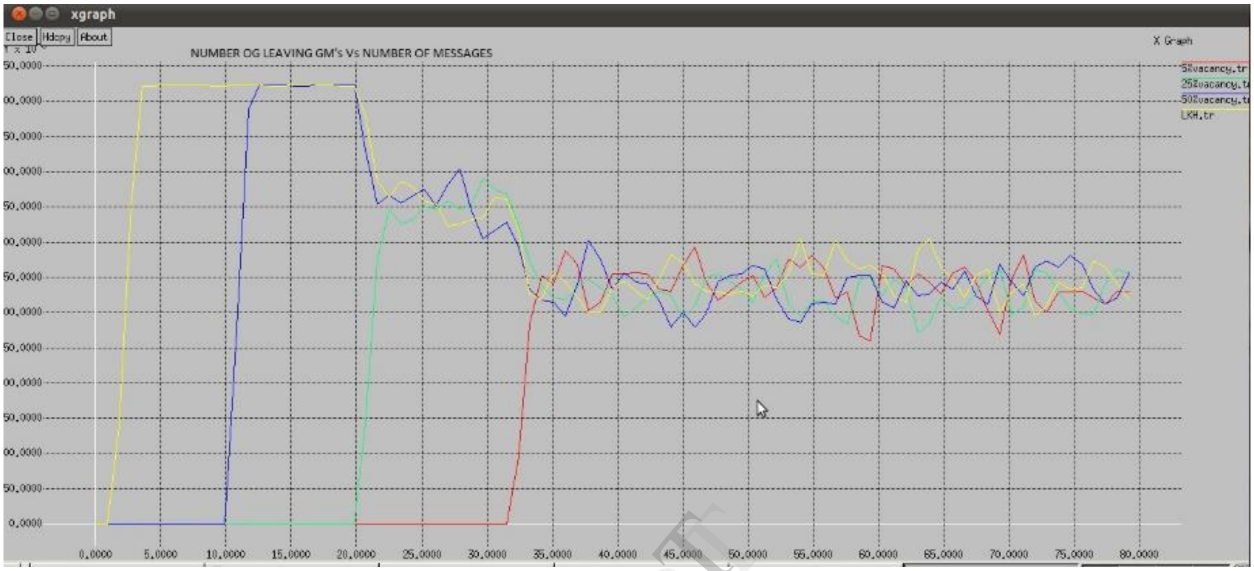

Figure 1. Number of leaving GM's Vs Number of messages

REFERENCES:

[1]Chung Kei Wong,, *"Secure Group Communications Using Key Graphs",*IEEE Feb 2000

[2] S. Benson Edwin Raj, J. Jeffneil Lalith , "*A Novel Approach for Computation-Efficient Rekeying for Multicast Key Distribution*" IJCSNS , VOL.9 No.3, March 2009.

[3]S. Rafaeli and D. Hutchison, "*A survey of key management for secure group communication*," ACM Computing Surveys *(CSUR)*, vol. 35, no. 3,

[4].Zhibin Zhou and Dijiang Huang ,"*An Optimal Key Distribution Scheme for Secure Multicast Group Communication,",* IEEE INFOCOM 2010.

[5]Lihao Xu, Cheng Huang, "*Computation Efficient Multicast Key Distribution*," IEEE Trans. Parallel And Distributed Systems, Vol 19, No. 5, May 2008.

[6]Mohamed M. Nasreldin Rasslan, Yasser H. Dakroury, and Heba K. Aslan "*A New Secure Multicast Key Distribution Protocol Using Combinatorial Boolean Approach*" ,International Journal of Network Security, Vol.8, No.1, PP.75–89, Jan. 2009

[7]J. Bethencourt, A. Sahai, and B. Waters, "*Ciphertext-Policy Attribute-Based Encryption*," *Proceedings of the 28th IEEE Symposium on Security and Privacy,"*2007(Oakland).

[8] A. Fiat and M. Naor, "*Broadcast Encryption, Advances in Cryptology-Crypto93*," Lecture Notes in Computer Science, vol. 773, pp. 480–491,1994.

[9]Daniele Micciancio and Saurabh Panjwani, "*Optimal Communication Complexity of Generic Multicast Key Distribution*", IEEE/ACM Transactions on Networking (2008).