Volume: 04 Issue: 06 Pages:1772-1777 (2013) ISSN : 0975-0290

Analysis of Adaptive Group Key Management Protocol Using Network Simulator 2

Mr. Rahul B. Mapari

Department of CSE, MIT, Aurangabad (MS),

Email: rahulmapari@gmail.com

Mr.Prashant M. Adhao

Department of IT, SGGS Institute of Engineering and Technology, Nanded (MS), India.

Email: p.adhao@gmail.com

----------ABSTRACT----------

IP Multicast is increasingly used as an efficient communication mechanism for group oriented applications. This success urged the development of security mechanisms for that communication model. In dynamic multicast groups, new members may join or current members may leave the group dynamically. Due to dynamic nature of these multicast groups, the group key is needed to be changed dynamically to maintain group key confidentiality. Existing group key management protocols do not take into consideration the dynamicity of group members. In this paper we first classify proposed protocols for secure multicast and discuss their non suitability for dynamic group. This leads to inefficient solutions for real multicast session. We then propose an efficient protocol, called AKMP, which maintain good performance by adapting the key management process with the membership frequency during the multicast session. A simulation result shows that AKMP is more efficient than existing protocols.

Keywords – ***backward secrecy, dynamic group, forward secrecy, secure multicast, traffic encryption key.***

Date of Submission: April 24, 2013 Date of Acceptance: May 21, 2013

I. INTRODUCTION

IP multicast is an efficient communication mechanism for group-oriented applications such as video conferencing, interactive group games and video on demand. IP multicast saves bandwidth by sending the source traffic on a multicast tree that spans all the members of the group. Security concerns for IP multicast are very complex because of the important number of communicating participants. This raises the problem of group communication confidentiality and thus group key management. Group communication confidentiality requires that only valid users could decrypt the multicast data even if the data is broadcast to the entire network. We assume in what follows that data is encrypted to ensure confidentiality using asymmetric cryptosystem. In this case, a symmetric key is used to encrypt data by the source and to decrypt it by receivers. It is generally called Traffic Encryption Key (TEK). The confidentiality requirements can be translated mainly into two key distribution rules [18]:

1) Forward confidentiality: Users that left the group should not have access to any future key. This ensures that a member cannot decrypt data after it leaves the group.

2) Backward confidentiality: A new user joining the group should not have access to any old key. This ensures that a member cannot decrypt data sent before it joins the group.

1. Rekeying Process

In order to meet the above requirements, a re-key process should be triggered after each join/leave. It consists in generating a new TEK and distributing it to the members including the new one in case of a join or to the residual members in case of a leave. This process ensures that a new member cannot decrypt old multicast data and prevents a leaving member from eavesdropping future multicast data. Unfortunately, no security mechanisms were foreseen in the initial multicast architectural design [6]. A lot of work has been done these last years to cope with this limitation. We propose to classify current group key management proposals into two approaches:

2. Approach A

In this approach all group members share a unique single symmetric key called the Traffic Encryption Key (TEK). This TEK is used by the source to encrypt multicast packets and by the receivers to decrypt them. This approach is mainly used within a centralized architecture where a single key server is responsible for generating and redistributing the new TEK whenever a member joins or leaves the group. Protocols within this approach do not meet scalability requirements since the number of transmitted messages to update TEK is proportional to n, where n is the number of group members [12][20]. This is known as the «1 affects n» phenomenon [19] where a single group membership change (join or leave) results in a re-keying process that disturbs all group members to update TEK. In addition, the use of a single key server leads to a bottleneck problem during TEK

Volume: 04 Issue: 06 Pages:1772-1777 (2013) ISSN : 0975-0290

distribution and suffers from a single point of failure. Some distributed solutions are proposed to share re-keying process among different entities and thereby to cope with scalability, bottlenecks and fault tolerance issues, but they remain suffering from the «1 affects n» phenomenon [14] [4]. *Approach B*

In order to cope with the approach A drawbacks («1 affects n», scalability, bottlenecks), in this approach the multicast group is divided into multiple subgroups. Each subgroup shares a local TEK managed by a special entity: the subgroup controller. The set of protocols proposed within this approach (called also: hierarchical key management protocols) are more scalable than centralized protocols. They also attenuate the «1 affects n» problem. In fact, if a member joins or leaves the group, only the concerned subgroup updates its local TEK. However, this improvement is not for free: as subgroups have different TEKs, multicast packets should be decrypted and re-encrypted by subgroup controllers whenever they pass from a subgroup to another. A common drawback of both approaches A and B is that they are not flexible regarding the dynamicity of the group: for multicast applications with few membership changes, protocols using approach A would fit better than others since re-keying would not be frequent and decryption / re-encryption overhead changes, protocols supporting approach B would fit better because of the attenuation of the «1 affects n» phenomenon and a better scalability. Moreover, the frequency of membership changes is generally unpredictable for a large range of multicast applications. Hence, none of both approaches A and B can be the best during all the secure session lifetime.

In this paper we propose to take advantage of both approaches A and B by dynamically adapting the key management process with respect to the frequency of membership changes. To do so we present a new protocol called the Adaptive Key Management Protocol. Within the same secure multicast session, AKMP begins with using a single TEK (approach A behaviour) and dynamically constructs subgroups with different local TEKs (approach B behaviour) whenever the join/leave frequency exceeds a certain threshold.

II. ADAPTIVE GROUP KEY MANAGEMENT PROTOCOL

The main idea of AKMP is to meet approach A as long as no frequent membership change is depicted by group members and to switch to approach B whenever members show certain dynamicity. What distinguishes AKMP from other hierarchical solutions [17] [19] [7] is restricting decryption / re-encryption process to the sub-networks that are subject to high dynamicity. In AKMP environment, we have several AKMP enabled routers. The protocol begins with a single group that shares a unique TEK. This group is initially managed by one AKMP router. During the multicast secure session, if an AKMP router detects a local dynamicity, it initiates a subgroup with an independent local key. To do so, this AKMP router generates and distributes the local key to the members in the constructed subgroup. This key is called a Downstream Key (DK). Then, the router decrypts received packets using its parent AKMP-router key (this key is called the Upstream Key (UK) and re-encrypts the packets downstream using DK. We say that this AKMP-router has switched from an inactive state to an active state (or switched to a decryption / re-encryption process).This way, AKMP reduces decryption / re-encryption overhead to the minimum while attenuating the «1 affects n» problem. Each AKMP-router Ri holds:

- A dynamicity-evaluation function I.: this measures the level of dynamicity of the group. This function could be based on a probabilistic model of the group dynamicity. In this paper we will use the following dynamicity-evaluation function:

if mcf > d then { switch to decryption/re-encryption proc}

f_i=*true;*
else
f_i=*false;*
end

mcf : membership frequency change
d= threshold

- An upstream key UK_i and a downstream key DK_i: the UK_i is used to decrypt upstream packets in order to re-encrypt them downstream using DK_i. Note that if UK_i=DK_i then R_i is not involved in decryption / re-encryption process.
- A pair of private/public keys (K_i^{-1}/K_i) for secure exchanges between AKMP-routers.

AKMP does not require that all routers should be AKMP enabled. All AKMP-routers are grouped in a virtual AKMP tree (Fig.1). Only the AKMP-enabled routers are involved in security mechanisms described below.

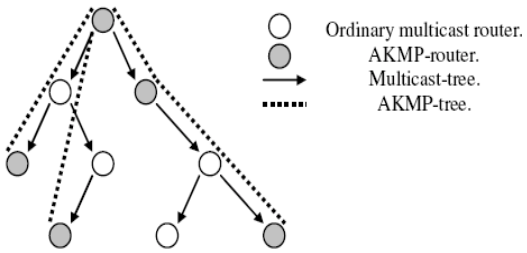


Fig.1. AKMP-tree concept

1. *Initialization*

Group controller (GC) is responsible for announcing the secure session. It also authenticates the AKMP routers and the members before their registration.

2. *Notation*

[a → b: m] - "a" send message "m" to "b"
[{m}$_k$] - "m" encrypted with "k"
[<m>k^{-1}] - "m" signed using the private key k-1

3. *AKMP dynamic aspect*

As AKMP is an adaptive solution, the state of AKMP routers is dynamic. Each AKMP-router could be in one of the three states (active, inactive, or waiting new UK):

- Active: an AKMP-router is "active" if it is involved in decryption / re-encryption process.

Volume: 04 Issue: 06 Pages:1772-1777 (2013) ISSN : 0975-0290

- Inactive: an AKMP-router is "inactive" if it does not assure decryption / re-encryption process.
- Waiting new UK: an AKMP-router is in this state if it waits for a new UK from an upward AKMP-router which is already involved in decryption / re-encryption process.

When an AKMP-router detects a membership change that yields f_i true, it becomes "active" (if it is not already "active") and thus generates a new DK_i and switches to decryption / re-encryption process as shown in Figure 3.3. The new DK_i will be considered as the traffic encryption key for the members attached to R_i and as an upstream key UK for the child routers of R_i. Thus DK_i is distributed by R_i to the attached members (resp. the residual attached members) and the child routers using script 1 (resp. script 2) as shown in Figure 2. A "DK_UPDATED, oldDK_i" message is sent to the parent AKMP-router (as shown in Figure 2) in order to update and distribute its DK_j in the case of oldDK_i=DK_j. Note that this latter update is necessary to ensure backward and forward confidentiality and that it is not a concern since it happens only for the first time that the child AKMP router switches to decryption / re-encryption process.

If f_i remains false after a membership change, R_i forwards this membership change upstream asking a new UK (it sends "NEW_UK_REQUEST" message to its upstream AKMP-router) and changes its state into "WAITING_NEW_UK" (as shown in Figure 3). This request is forwarded upstream until it reaches the first active AKMP-router R_j, it is then discarded by the latter which generates a new DK_j. R_j distributes then the new DK_j (inside a special message "UPDATE_UK") using script 1 (as shown in Figure 2) and each child AKMP-router R_p reacts to the message according to the state chart of Figure 3depending on its state. When R_i receives DK_j it resumes its "inactive" state (as shown in Figure 3) and, initializes UK_i and DK_i to DK_j and distributes it using script 1 or 2 depending on whether the membership change is a join or a leave.

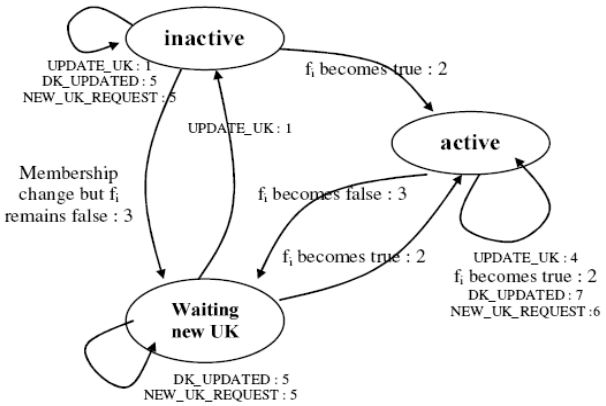
Script 1:
for j=1..nb_attached_members
$R_i \rightarrow u_j : \{newDK_i\}_{Kuj}$.

for j=1..nb_child_AKMP-routers
$R_i \rightarrow R_j : <UPDATE\_UK, \{newDK_i\}_{Kj}>_{Ki^{-1}}$.
$Ri \rightarrow$ PARENT_ROUTER:$<DK\_UPDATED, oldDK_i>_{Ki^{-1}}$.

Script 2 :
/* Distributing $newDK_i$ to residual members, u_j is the leaving member */
for p =1..nb_attached_members, p#j
$R_i \rightarrow u_p : \{newDK_i\}_{Kp}$
for p =1..nb_child_AKMP-routers
$R_i \rightarrow R_p : < UPDATE\_UK, \{newDK_i\}_{Kp}>_{Ki^{-1}}$
$R_i \rightarrow$ PARENT_ROUTER:$<DK\_UPDATED, oldDK_i>_{Ki^{-1}}$.

Fig.2. Distributing New D_k

If an active AKMP-router detects that there is no longer dynamicity among its attached members, it stops decryption / re-encryption process and rejoins "inactive" state. A new DK(s) is (are) generated and distributed according to the above process. This manner of dealing with membership changes adapts easily to partial dynamicity and fits better to applications with unpredictable membership behavior.



Legend : Event : Action
1 : DK_i = newUK.
UK_i = newUK.
Action 2.
2 : Distributes newUK using script (1 or 2) depending on whether the membership change is a join or a leave.
3 : sends NEW_UK_REQUEST upstream.
4 : UK_i = newUK.
5 : forwards the message upstream.
6 : generates and distributes new DK_i using script 1.
7 : if oldDK_j = currentDK_i then action 6.

Fig. 3. AKMP-router state chart

III. RESULTS AND DISCUSSIONS

In this section, we provide an overview of our simulation model and some of the results we obtained by comparing AKMP with centralized solution.

1. Simulation Model

All schemes discussed in this paper have been tested using Network Simulator *NS* version *2.34*. *NS* give the facility of defining our own protocols. For defining our own protocol, classes and their functionalities are to be defined using C++. Parallel classes are then need to be defined using OTcl (Object-oriented Tool Command Language) and linking can be done between the OTcl classes and C++ classes.

2. Results

Different test scripts have been written for different sizes of groups as 9, 27, 81, 243, 729 and 1001 members. Trees with 9, 27, 81, 243, 729 members are complete trees whereas those with 1001 members are incomplete trees. These scripts have been written for all CtrMcast and CtrMcastAKMP protocols.

These test scripts are given as input to *ns*. The overall run-time of these scripts is computed by using the time difference between the start of execution of the script and

Volume: 04 Issue: 06 Pages:1772-1777 (2013) ISSN : 0975-0290

end of execution of the script. For getting time, the date command with +%r option is used.

Scripts are used for testing time required for join as well as leave operation for all the group sizes. In all these scripts, node labelled '0' is treated as a Group Controller. The internal nodes in the tree are logical; but while simulating the topologies they have been assumed as real nodes.

After simulation the overall runtime for join operation is summarized in table 1 and is briefed in figure 4.

TABLE 4.1
OVERALL RUNTIME OF JOIN OPERATION (IN SECONDS)

No. of Members	CtrMcast (Time in sec)	AKMP (time in Sec)
9	1	0
27	1	0
81	2	1
243	8	3
729	120	80
1001	162	120

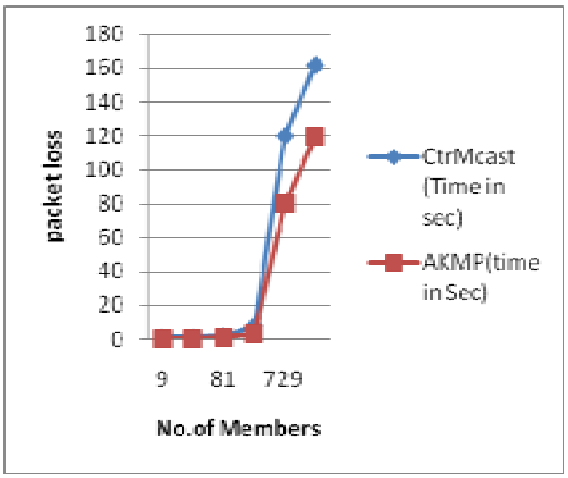


Fig.4. Chart showing performance analysis of Join operation

Using above simulation, overall runtime for leave operation is summarized in table 2 and is briefed in figure 5.

TABLE 4.2
OVERALL RUNTIME OF LEAVE OPERATION (IN SECONDS)

No. of Members	CtrMcast (Time in sec)	AKMP (time in Sec)
9	1	0
27	1	0
81	3	1
243	10	7
729	122	72
1001	164	129

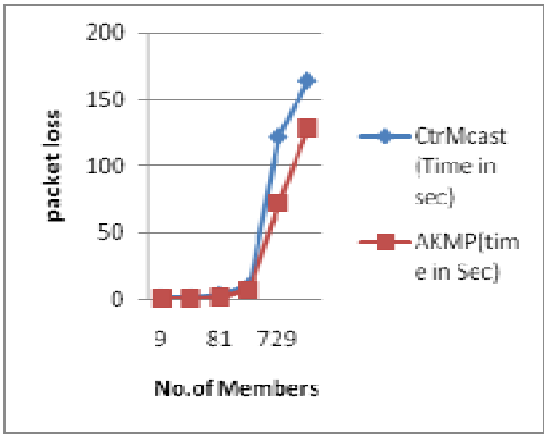


Fig. 5. Chart showing performance analysis of Leave operation

Using above simulation, Overall packet loss for join leave operation is summarized in table 3 and is briefed in figure 6.

TABLE 4.3
OVERALL PACKET LOSS OF JOIN LEAVE OPERATION (IN SECONDS)

No. of Members	Packet Loss CtrMcast	Packet Loss AKMP	Packet Loss Proposed Scheme
9	29	19	9
27	39	26	16
81	44	30	24
243	49	40	29
729	69	59	39
1001	89	74	51

Fig.6. Chart showing Comparison of performance analysis of Join Leave operation with Packet Loss Parameter

Volume: 04 Issue: 06 Pages:1772-1777 (2013) ISSN : 0975-0290

IV. CONCLUSION

In this paper, Approach A and Approach B group key management protocols are analysed. A proposed scheme is based on combination of Approach A and Approach B protocols. All schemes are simulated using ns2 simulation tool. When dynamicity in the multicast group is increased, greater overall computational and communication cost (run-time) is required for join/leave operation because of problems such as "1 affects n", decryption and re-encryption overhead.

From performance analysis of all the above schemes for computation and communication cost (i.e. overall run-time) of join / leave operation in dynamic multicast group, it has been observed that overall run-time is minimum for the AKMP.

In comparison with the approach A, approach B and adaptive group key management protocol, performance analysis shows that proposed scheme reduces overall packet loss during the message transfer.

As a result, we can conclude that Adaptive Key Management Protocol takes advantage of both approaches A and B by dynamically adapting the key management process with respect to the frequency of membership changes. Which reduced overall run time cost and packet loss.

As a future scope, the work can be done to reduce the value of parameters like bandwidth, end to end delay, and jitter.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers and the editor for their constructive comments. The authors would also like to thanks to management of MIT, Aurangabad for continuous guidance and support.

REFERENCES

[1] K. Almeroth and M. Ammar. "Collecting and modeling the join/leave behavior of multicast group members in the Mbone" , In the proceedings of the Symposium on High Performance Distributed Computing, Syracuse NY, 1996.

[2] K. Almeroth and M.Ammar. "Multicast group behavior in the internet's multicast backbone (Mbone)" , IEEE communications Magazine, 1997.

[3] A. Ballardie, "Scalable Multicast Key Distribution", RFC 1949 May 1996.

[4] Ghassen Chaddoud, Isabelle Chrisment, André Shaff , "Dynamic Group Communication Security", 6th IEEE Symposium on computers and communication, 2001.

[5] Chung Kei Wong, Mohamed Gouda, Simon S. Lam, "Secure Group Communications Using Key Graphs", ACM SIGCOMM' 98, Columbia, 1998.

[6] S. Deering,"Host Extensions for IP Multicasting", RFC 1112. Stanford University, August 1989.

[7] Lakshminath R. Dondeti, Sarit Mukherjee, Ashok Samal, " Scalable secure one- to-many group communication using dual encryption", Computer Communications, ELSEVIER Science 2000.

[8] Lakshminath R. Dondeti, Sarit Mukherjee, Ashok Samal, "Comparison of Hierarchical Key Distribution Schemes" , In Proceedings of IEEE Globcom Global Internet Symposium, 1999.

[9] Lakshminath R. Dondeti, Sarit Mukherjee, Ashok Samal,"Survey and Comparison of Secure Group Communication Protocols" , Tech Rep, University of Nabraska-Lincoln. 1999.

[10] Thomas Hardjono, Gene Tsudik, "IP Multicast Security: Issues and Directions", Annales de telecom 2000.

[11] Peter S. Kruus,"A survey of multicast security issues and architectures", 21st NISSC Proceedings, Virginia (USA). 1998.

[12] H. Harney, C. Muckenhirn, "Group Key Management Protocol Architecture", RFC 2094 July 1997.

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

[14] Rolf Oppliger, Andres Albanese, "Distributed registration and key distribution", Proceedings of the 12th International Conference on Information Security IFIP SEC'96, Greece, 1996.

[15] H. F. Salama, "Multicast Routing for real-Time Communication on High-Speed Networks", PhD thesis, North Carolina State University,1996.

[16] Sandro Rafaeli, "A Decentralized Architecture for Group Key Management", Computer Department Lancaster University, September 2000.

[17] Clay Shiels, J. J. Garcia-Luna-Aceves,"KHIP—A scalable protocol for secure multicast routing", Proceedings of ACM SIGCOMM' 99. 1999.

[18] Jack Snoeyink, Subhash Suri, George Vorghese,"A Lower Bound for Multicast Key Distribution", IEEE INFOCOM 2001.

[19] Suvo Mittra, "Iolus: A Framework for Scalable Secure Multicasting", In Proceedings of ACM SIGCOMM'97, Cannes, France, 1997.

[20] D. Wallner, E. Harder, R.Agee, "Key Management for Multicast: Issues and Architecture", RFC 2627, National Security Agency Jun 1999.

[21] R. Waxman, "Routing of multipoint connections", IEEE Journal on Selected Areas in Communications, December 1988.

[22] Hatem Bettahar, A. Bouabdallah, Yacine Challal, "AKMP: An Adaptive Key Management Protocol for secure multicast", Computer Communications and Networks, 2002. Proceedings, Eleventh International Conference on 14-16 Oct. 2002 .

[23] William Stallings,"Cryptography and Network Security- principles & practices", Third Edition.

Volume: 04 Issue: 06 Pages:1772-1777 (2013) ISSN : 0975-0290

[24] B. Schneier, “Applied Cryptography”, Second Edition, John Wiley & Sons Inc., 1996.

[25] Kevin Fall, Kannan Vardhan, “The ns Manual (formerly ns Notes and Documentation”, The VINT Project, October 19, 2008.

[26] D.R. Stinson and T. van Trung, “Some New Results on Key Distribution Patterns and Broadcast Encryption”, Designs, Codes and Cryptography, vol. 14, pp. 261-279, 1998.

[27] D.R. Stinson,“On Some Methods for Unconditionally Secure Key Distribution and Broadcast Encryption”, Designs, Codes and Cryptography, vol. 12, pp. 215-243, 1997.

[28] C. Blundo, L.A. Frota Mattos, and D.R. Stinson, “Trade-Offs between Communication and Storage in Unconditionally Secure Schemes for Broadcast Encryption and Interactive Key Distribution”, Advances in Cryptology—Proc. 16th Ann. Int’l Cryptology Conf. (CRYPTO ’96), pp. 387-400, 1996.

[29] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, “Perfectly Secure Key Distribution in Dynamic Conferences”, Advances in Cryptology—Proc. Workshop Theory and Application of Cryptographic Techniques (EUROCRYPT ’93), pp. 471-486, 1993.

[30] B .A. Forouzan, “Cryptography and Network Security”, Second Edition, Mcgraw Hill Publication, pp.001 - 025

[31] Yacine Challal, Hamida Seba, “Group Key Management Protocols: A Noveltaxonomy”, International Journal Of Information Technology Volume 2 Number 1 2005 Issn:1305-2403.