Key management and multi-layer IPSEC for satellite multicast

H Cruickshank, M. P. Howarth, S.Iyengar and Zhili Sun

Centre for Communication Systems Research, University of Surrey, Guildford, UK

Email: H.Cruickshank@surrey.ac.uk

ABSTRACT

Satellites are also ideally suited for delivery of multicast applications. However secure multicast over satellites is a challenging problem. One important step toward the correct solution for end-to-end security is the integration of security architectures between satellites and IP terrestrial networks.

This paper presents a secure group management and key distribution architecture based on the current activity in the IETF on IPSEC and securing group communications. The paper presents the Logical Key Hierarchy as a group key distribution system, which makes full use of the satellite broadcast capabilities. We propose and analyse an interworking solution between multi-layer IPSEC and LKH that reduces key management traffic while enabling interworking with performance enhancing proxies (PEPs) used on satellite links.

In this interworking solution, between ML-IPSEC and LKH, the end users are put into one branch of the LKH tree and the satellite terminals are put into another branch. The root key in the LKH tree can be used for securing the transport header (TCP or UDP) and a branch key acts as data key to secure the data content for the end users. The proposed scheme is scalable, in that the rekey effort varies with $\log N$, and efficient, in that for user departures the number of rekeys required is in the region of half that of two separate tree hierarchies.

1. INTRODUCTION

Demand continues to grow for broadband networks capable of supporting applications such as multimedia and information distribution, and one important component of a communications architecture that can support these services is multicast. However, terrestrial IP multicast has only slowly been deployed, due to the complexities of wide scale networks that include large numbers of multicast-enabled routers. This situation is expected to continue at least for the foreseeable future, restricting accessibility to multicast content for most potential European users. In contrast, a satellite service could simplify multicast deployment and operations/maintenance, since a single satellite hop (using only a small number of multicast enabled routers) would provide uniform delivery across the whole footprint of the EC [1] [2] .

The process of securing and performing key management for unicast connections is well understood [3] , but multicast security is more complex. In principle, a multicast connection can be regarded as a set of unicast connections, but this approach does not scale well for large groups, especially at the scales expected in satellite systems. Protocols that manage the process of distributing keys in a multicast environment are under development [2] [5] .

The principal actors in multicast key management are the group controller (GC) and group members (GMs). The former is responsible for creating and distributing keys and rekeying (to maintain security) as appropriate; the group members are entities with access to the group keys. The GC need not be co-located with the multicast data source. The multicast group may need to be rekeyed for any of a number of reasons:

(1) The group key is usually updated regularly (typically every few seconds or minutes) to reduce the probability of successful cryptanalysis of the encrypted traffic.

(2) The group key may also need to be changed on demand if it is determined that the key has been compromised.

(3) Rekeying may be required when a new member joins the multicast group. This ensures that the member cannot decrypt encoded traffic sent prior to their joining (backward secrecy).

(4) Rekeying may be required when an existing member departs from the multicast group. This ensures that the member cannot decrypt encoded traffic sent after they leave (forward secrecy).

For large multicast groups that have frequent membership changes the cost of rekeying can be significant, since satellite resources are expensive. Scalable rekeying is therefore an important problem that needs to be considered in order to support secure communications for large dynamic groups. We now consider rekey techniques for each of the four functions listed above.

Several techniques exist for rekeying (1) and (3) above: two options are for the new group key to be encrypted with either (a) the old group key, or (b) a separate "control" key negotiated during session establishment. For (2) and (4) above a different rekeying approach is required since the old key is known by at least one user who is no longer to be a recipient of the multicast transmission. We now consider options for this rekeying.

A number of multicast key management approaches have been developed with the objective of improving the scalability of group secure associations, by ensuring that parameters grow more slowly than the group size, *N*. Parameters considered include group controller encryption effort, memory requirements, network traffic, and group members' decryption effort and memory requirements. One example of such techniques is the logical key hierarchy, LKH [14] , [15] .

2. LOGICAL KEY HIERARCHY (LKH)

Logical Key Hierarchy (LKH) is a mechanism for security key management within a group of entities, providing the ability to initialise the group with a common key and then to rekey the group as required (RFC2627). It is thus of particular application in secure multicast communications.

Logical Key Hierarchy (LKH), uses a set of keys arranged in a tree structure to reduce the cost of rekeying. For a tree of outdegree *k* and depth *d*, the number of rekeys transmitted on a member compromise is reduced from $N = k^d$ (for a flat system) to $k \log_k N - 1$. The system is also robust against collusion, in that no set of users together can read any message unless one of them could have read it individually. Improvements to LKH for the specific case of binary trees (*k*=2) have also been proposed in one-way function trees [6] [7] [8] : both these approaches reduce the number of rekeys required in the event of compromise of a user from $2\log_2 N - 1$ to $\log_2 N$.

We introduce LKH by considering a simple flat key management system that can be used to share a single key, 'A', so that it is known to the GC and all GMs but to no other entities. The flat key management system consists of *N* pairwise keys each shared between the group controller and one of the *N* group members (Fig. 1). Each of these pairwise secure associations is represented by a circle and the group key is represented by the box labelled 'A'. If the group key is changed the new group key has to be encrypted with each user's unique pairwise key and then unicast to that user; each of these encrypted keys is represented by one of the lines drawn in Fig. 1. Thus for *N* users a total of *N* encrypted keys are generated and transmitted across a network.

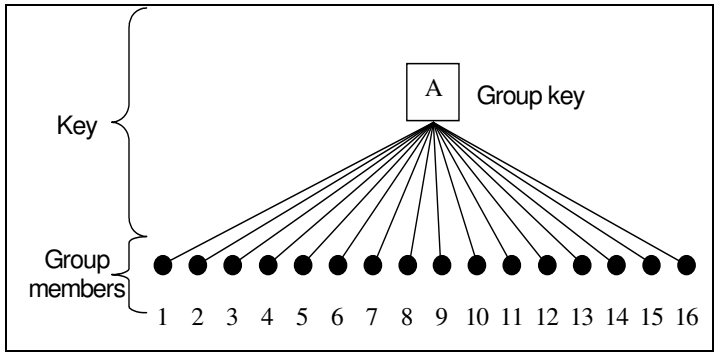

Fig. 1: Key hierarchy: *N* pairwise keys

We contrast this with LKH, where a tree of keys is used to share a single key 'O' so that it is known to the GC and all GMs but to no other entities. In Fig. 2 the keys are labelled A through O, the circles again represent the pairwise keys, and the lines each represent encrypted keys sent across the network, as we shall now see. Suppose now that User 11 needs to be deleted from the multicast group. Then all of the keys held by User 11 (keys F, K, N, O) must be changed

and distributed to the users who need them, without permitting User 11 to obtain them or anyone else who is not entitled to them. To do this, we must replace the keys held by User 11, proceeding from the bottom up.

The server chooses a new key for the lowest node (not the leaf, for which a unicast secure association exists between the GC and the GM), and then transmits it encrypted with the appropriate daughter keys. Thus for this example, the first key replaced is Key F, and this new key will be sent encrypted with User 12's unique pairwise key. The second key replaced is Key K, which is sent encrypted with the newly replaced Key F (for User 12) and also sent encrypted with key E (for Users 9 and 10). Key N is then sent encrypted in the newly replaced Key K (for Users 9, 10, and 12) and also encrypted in key L (shared by Users 13 through 16). Finally, Key O is replaced, and this new key is sent encrypted in the newly replaced Key N (for Users 9, 10, and 12 through 16) and also separately is encrypted in key M (shared by Users 1 to 8). Since we are proceeding from the bottom up, each of the replacement keys will have been replaced before it is used to encrypt another key.

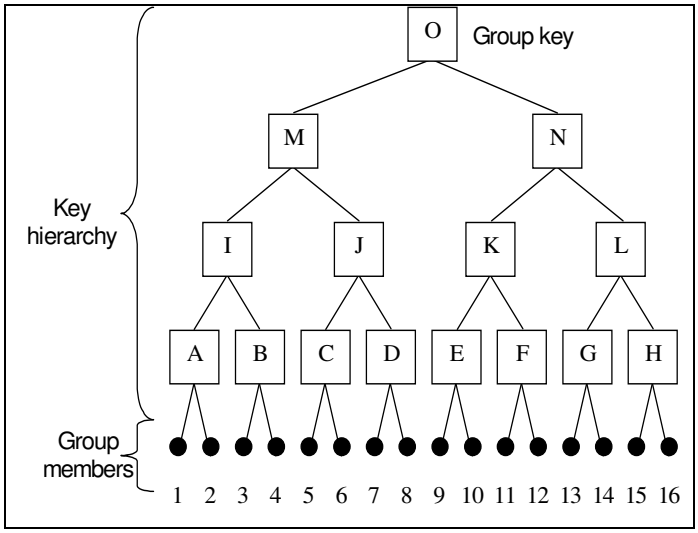

Fig. 2: Logical Key Hierarchy

The seven keys sent represent a significant saving on the 16 keys that would need to be transmitted using the flat key system of Fig. 1. We briefly write these keys as $\{F\}_{12}\ \{K\}_E\ \{K\}_F\ \{N\}_K\ \{N\}_L\ \{O\}_M\ \{O\}_N$. In general, the number of transmissions required is the sum of the degrees of the replaced nodes. In a *k*-ary tree of depth *d*, this is a total of $kd - 1 = k \log_k N - 1$ transmissions.

The group traffic encrypting key (GTEK), used to encrypt data traffic, may, depending on the group security policy, either be key O (Fig. 2), or it may be separately encrypted using key O and transmitted to all group members.

3. MULTI-LAYER IPSEC

The use of traffic encryption at the network layer can be incompatible with performance-enhancing modules used on satellite links. The transport layer may be either unicast (TCP or UDP) or multicast or reliable multicast. Some basic rules for TCP optimization techniques used in satellite communications and the implications they might have for IPSEC have been pointed out in [10] . However, if the optimization techniques involve intermediate routers or gateways and these require read or write access to the transport layer header or encapsulated data, IPSEC cannot be used without some kind of adaptation. We therefore now proceed to consider a mechanism that permits network layer traffic encryption with satellite performance-enhancing modules.

Work on multi-layer IPSEC has been carried out for example by Hughes Network Systems [11] . It has also been considered for mobile networks [13] , and optimization of multicast over satellites is still a research issue. Multi Layer-IPSEC defines a security relationship that involves not only the sender and the receiver of a security service, but also selected intermediate nodes along the traffic stream. The IP datagram is divided into several zones and different protection schemes are applied to each zone. Fig. 3 illustrates the use of ML-IPSEC in a satellite environment [12] .

Individual security relationships can be used to cover each zone of the IP datagram, and then build a new type of secure association (SA), called a composite SA (CSA). Thus the transport data is encrypted using key K2, while the transport header is encrypted using key K1.

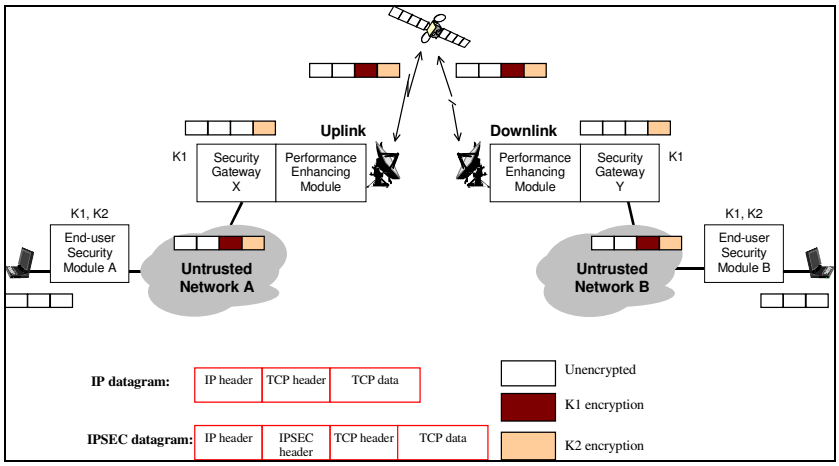

Fig. 3. ML-IPSEC in a satellite context

For simplicity the analysis is initially illustrated here in terms of a secure unicast connection. For multicast, the analysis is identical, although there are potentially many downlink gateways. A user in Network A wishes to establish a secure link with a node in Network B. End-user security module A establishes secure associations with end-user security module B and also with the satellite gateways, X and Y. When user A sends an IP datagram, it is encrypted using key K1 to encrypt the transport header and key K2 to encrypt the transport content. On receipt by the uplink terminal the transport header only is decrypted by security gateway X, any performance enhancing functions such as header compression or spoofing are performed, and the transport header is re-encrypted. The datagram is then transmitted over the satellite link, where security is assured by the encryption. At the satellite downlink (security gateway Y), the transport header can be decrypted using K1 and any performance enhancing functions can be performed. The header is re-encrypted by security gateway Y using K1 and forwarded. The datagram is fully secured in transit over the untrusted network B. At end user B the datagram is decrypted using both K1 and K2.

In summary, it can be seen that the security entities at the ends of a connection (i.e. at the source A and at each destination B of a multicast transmission) need both group keys K1 and K2. However, intermediate security gateways that are responsible for performance-enhancing functions only need access to group key K1 to enable them to read and if necessary change the transport header.

The security gateway needs access to key K1, and the end user(s) need access to keys K1 and K2. In order to achieve this, both the satellite terminal operator and the end user need to trust a secure third party. The trusted third party is responsible for generating and distributing the keys, and this forms the basis for the end-to-end security between users. The model also implies a limited trust between the user and the satellite terminal operator, where the latter is trusted to have access to the transport headers.

3.1. ML-IPSEC and LKH Interworking

We now present an extension of the logical key hierarchy discussed in Section III that provides an efficient and scalable key management system for multicast ML-IPSEC. The two group keys K1 and K2 could be managed using two separate logical key hierarchies, but a saving can be made by integrating them into a single hierarchy as follows. Fig. 4 shows the proposed key hierarchy for a set of users U1 to U9 and a set of intermediate gateways G1 to G4.

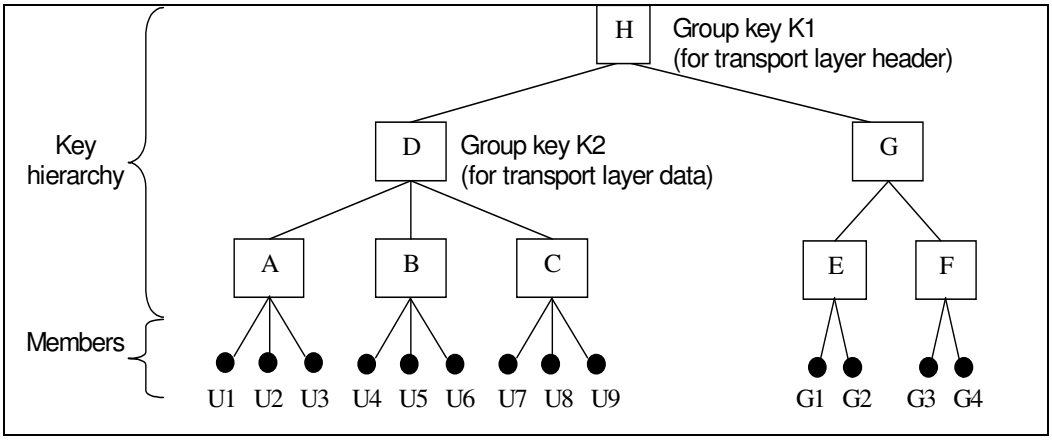

Fig. 4 Integrated LKH tree for ML-IPSEC

All users and intermediate gateways are members of the multicast group. In this illustration, the nine users are in a subtree of outdegree $k_U = 3$, and the four gateways are in a subtree of outdegree $k_G = 2$. The root, key H, has two children, irrespective of the values of k_U and k_G. As with the LKH of Section III, Keys K1 and K2 could either be the LKH keys H and D shown in Fig. 4, or they could be separate keys, encrypted using H and D respectively and transmitted to members. The group key K1 used to encrypt the transport layer header is at the root of the tree, and the group key K2 used to encrypt the data is one of the root's two child keys. Recalling that in a logical key hierarchy each member only knows the keys that lie on the path from the member's leaf node to the root, it can be seen that users have access to both K1 and K2, while gateways only have access to key K1. In the event of a gateway compromise, as part of the normal LKH rekeying we transmit $\{K1\}_{K2}$ and so users can still decrypt the transport header. For the users located behind the compromised gateway, it is assumed that a protection path exists via one of the other gateways so that they can still receive the multicast traffic.

If there are N_U users and N_G intermediate gateways, then the cost of rekeying for this single integrated tree is as follows. For a user depart, the rekey cost in keys is:

$$R_{U-\text{integrated}} = k_U \log_{k_U} N_U + 1 \qquad (1)$$

For a gateway depart the rekey cost is:

$$R_{G-\text{integrated}} = k_G \log_{k_G} N_G + 1 \qquad (2)$$

This compares with rekey costs for two separate trees (Fig. 5) as follows.

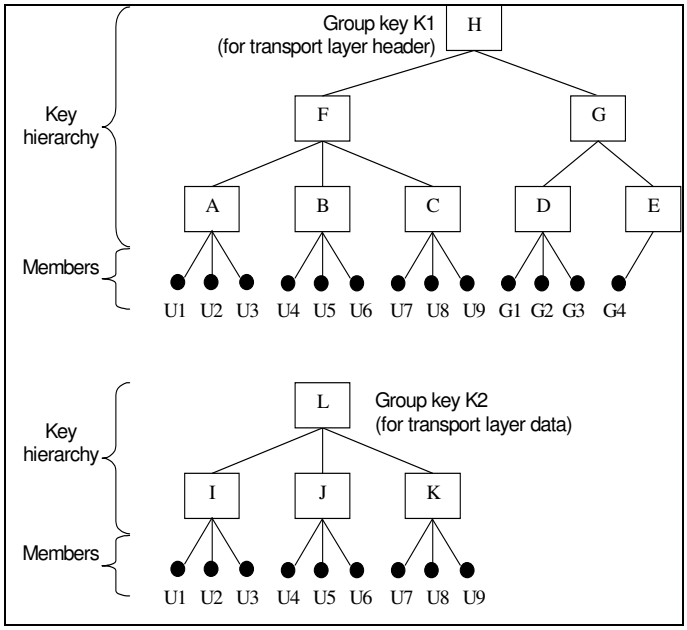

Fig. 5. Separate LKH trees for transport header and data

The combined LKH tree for key K1 has $N_U + N_G$ members; let its outdegree be k_{comb}. The LKH tree for K2 has N_U members, and we assume its outdegree to remain at k_U. Then, for a user depart, the rekey cost to rekey both keys K1 and K2 is:

$$R_{U-separate} = \{k_{comb} \log_{k_{comb}} (N_U + N_G) - 1\} + \{k_U \log_{k_U N_U} - 1\} \quad (3)$$

and for a gateway depart, the rekey cost is (to rekey K1 only):

$$R_{G-separate} = k_{comb} \log_{k_{comb}} (N_U + N_G) - 1 \quad (4)$$

The rekey costs on a user departure and a gateway departure are shown respectively in Fig. 6 and Fig. 7. These show that for a user rekey, the rekey cost is almost halved for values of $N_U \geq N_G$ and is reduced still further in the unlikely event that the number of gateways exceeds the number of users. For a gateway departure, the integrated LKH saving compared to the separate LKH rekey cost is particularly high for $N_U >> N_G$, that is, when there are a relatively small number of gateways.

The gateways would normally be expected to be less volatile than the users. In this case, the optimum outdegree of the user subtree may be selected using the analysis described in Section III using the outdegree of the user subtree, k_U.

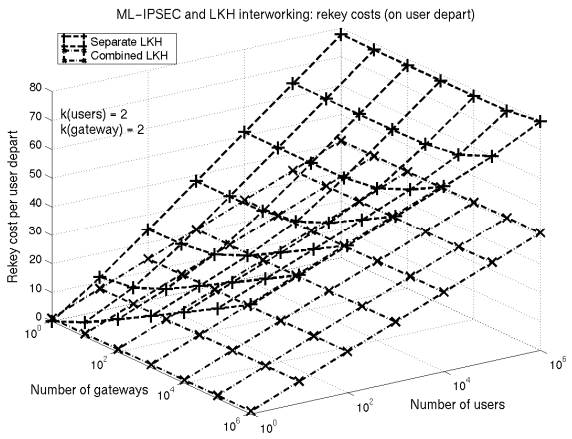

Fig. 6. ML-IPSEC and LKH interworking: rekey costs on user depart

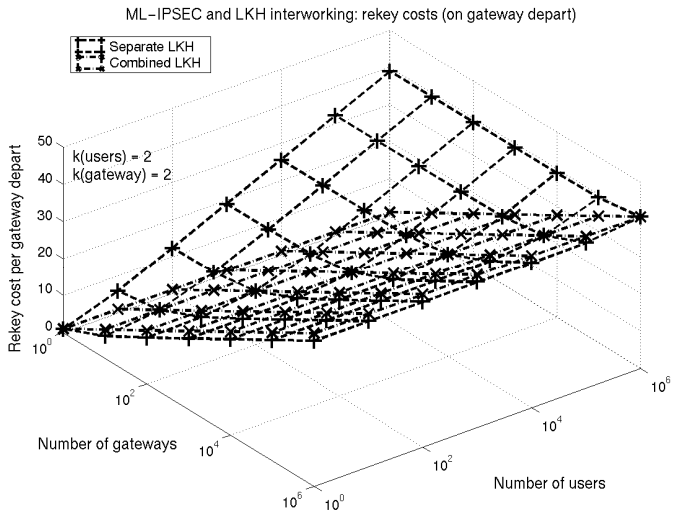

Fig. 7. ML-IPSEC and LKH interworking: rekey costs on gateway depart

4. CONCLUSION

This paper presents a security solution for secure group management over satellites. This solution is based on IETF activity in the MSEC working group on LKH key distribution architecture. This paper has presented an interworking solution between ML-IPSEC and LKH that supports the implementation of performance-enhancing modules for satellites. In the proposed approach, the end users are placed in one branch of the LKH tree and the satellite terminals or gateways are located in another branch. The root key in the LKH tree can be used for securing the transport header and a branch key acts as GTEK and secures the data content for the end users. The proposed scheme is scalable, in that the rekey effort varies with $\log N$, and efficient, in that for user departures the number of rekeys required is in the region of half that of two separate tree hierarchies.

5. ACKNOWLEDGEMENTS

This work was supported by the EU Information Society Technologies GEOCAST project, IST 11754 and ESA project with LOGICA-CMG on secure multicast over satellites.

6. REFERENCE:

[1] M. Annoni, et al (TILAB); H. Cruickshank, M. Howarth and Z. Sun (CCSR, University of Surrey, UK). "Interworking between Multi-Layer IPSEC and secure multicast services over GEO satellites". COST-272 meeting, paper number TD-02-016-P, June 2002.

[2] Z. Sun, H. Cruickshank, S. Iyengar and M. Howarth, "IP Multicast over Satellites – Technology Challenges", *Proceedings of the 20th AIAA International Communication Satellite Systems Conference and Exhibit*, Montreal, Canada, 12-15 May 2002.

[3] D. Harkins and D. Carrel, "The Internet Key Exchange," IETF RFC2409, Nov. 1998.

[4] H. Orman, "The OAKLEY key determination protocol," IETF RFC2412, Nov. 1998.

[5] H. Harney, A. Schuett and A. Colegrove, "GSAKMP Light," IETF Internet Draft, work-in-progress, draft-ietf-msec-gsakmp-light-sec-01.txt, Jul 2002, expires Dec 2002.

[6] D. Balenson et al., "Key management for large dynamic groups: one-way function trees and amortized initialization", IETF Draft, work-in-progress, draft-balenson-groupkeymgmt-oft-00.txt, Feb 1999.

[7] M.J. Moyer et al., "A survey of security issues in multicast communications," *IEEE Network*, Nov 1999, pp.12-23.

[8] R. Canetti et al., "Multicast security: a taxonomy and some efficient constructions," *Proc. IEEE INFOCOM* 1999, pp. 708-716.

[9] B. Schneier, "Applied cryptography," John Wiley & Sons, 1996.

[10] G. Noubir and L. von Allmen, "Security Issues in Internet Protocols over Satellite Links", *IEEE Proc. 50th Vehic. Tech. Conf.* 1999, pp.2726-2730.

[11] Yongguang Zhang, "Multi-layer Internet Security for satellite & wireless networks", Hughes Research Laboratories, HRL Technical Report 99-611, 1 Dec. 1999.

[12] M. Annoni et al., "Interworking between multi-layer IPSEC and secure multicast services over GEO satellites," COST-272 symposium, document TD-02-016-P, Thessaloniki Greece, 20-21 June 2002.

[13] N. Assaf et al., "Interworking between IP security and performance enhancing proxies for mobile networks," *IEEE Comms. Magazine*, Vol. 40, No. 5, May 2002, pp.138-144.

[14] D. Wallner, E. Harder and R. Agee, "Key management for multicast: issues and architectures," IETF RFC2627, June 1999.

[15] C.K. Wong, M. Gouda and S.S. Lam, "Secure group communications using key graphs," *IEEE/ACM Trans. Networking*, Vol. 8 No. 1, 2000, pp.16-30.