September 2003

A novel dynamic key management scheme for secure multicasting

J. Zhang
Macquarie University

V. Varadharajan
Macquarie University

Y. Mu
University of Wollongong, ymu@uow.edu.au

A novel dynamic key management scheme for secure multicasting

Abstract

We propose a new secure multicast scheme based on a novel hybrid key distribution scheme. This scheme meets the requirements described in the Internet Engineering Task Force (IETF) for multicast security architecture. It exhibits certain unique advantages in security services over existing schemes in the area of dynamic group key management. Our scheme allows efficient mechanisms for group members to join and leave a group frequently.

Disciplines

Physical Sciences and Mathematics

Publication Details

This paper originally appeared as: Zhang, J, Varadharajan, V and Mu, Y, A novel dynamic key management scheme for secure multicasting, ICON2003. The 11th IEEE International Conference on Networks, 28 September - 1 October 2003, 391-395.

A Novel Dynamic Key Management Scheme for Secure Multicasting

Junqi Zhang[1], Vijay Varadharajan[1], and Yi Mu[2]
[1] Department of Computing, Macquarie University, Sydney, Australia
[2] School of Information Technology and Computer Science,
University of Wollongong, Wollongong, Australia

Abstract— **We propose a new secure multicast scheme based on a novel hybrid key distribution scheme. This scheme meets the requirements described in the Internet Engineering Task Force (IETF) for multicast security architecture. It exhibits certain unique advantages in security services over existing schemes in the area of dynamic group key management. Our scheme allows efficient mechanisms for group members to join and leave a group frequently.**

I. Introduction

In multicasting, a message is sent from one party to many recipients, or from many recipients to many recipients[1]. Internet Protocol (IP) Multicasting was first proposed and specified in the 1980s. Since the creation of the Mbone in 1992, the interests in IP Multicast has been expanding rapidly. This is because multicasting enables the desired applications to service many users without overloading a network and resources in the server. In general, multicast applications can include both real-time and non-real-time applications which may involve data and multimedia.

Security is essential for data transmission over public networks. As defined in ISO 7498[2], there are several facets to security: confidentiality, integrity, access control, authentication, non-repudiation, and auditing and accountability. There are several protocols widely used to address the unicast security issues, but often they may not be directly extended to a multicast environment. Multicasting introduces some distinct security issues differing to unicast [3][4][5] . First, in general multicasting is more vulnerable than unicast, because transmissions occur over many network channels. A more difficult issue arises due to the multicast group membership being usually dynamic. Users can leave and join the groups, thus making the issue of group management a significant challenge in large scale systems. Also we need to ensure forward and backward secrecy. Forward secrecy implies that whenever a member of a group leaves the group, s/he must be prevented from having further access to the data and keys of that multicast group. Backward secrecy requires that the data communicated within a group before a new member or members join must remain secret to the new member or members. Other multicast security requirements include "1 affects all" scalability and data source authentication. The former requirement implies that the addition or removal of one or more members from a group should not affect other members of the group. The latter addresses the situation where an adversary or a group member poses as a member or another member of the group in sending the data. This requires group member authentication and data origin authentication.

Several schemes have been proposed for secure multicasting over the recent years, which can be classified into three categories [4][6][7]: centralized flat schemes, distributed flat schemes, and hierarchical schemes. Specific ones include manual key distribution, pairwise keying, hierarchical trees, secure lock, distributed registration and Key Distribution (DiRK)[8][9][5]. We will give a brief analysis of these schemes in Section 5 and compare them with our new scheme proposed in this paper.

In this paper, we present a novel secure multicast schemes based on a novel hybrid key distribution scheme. It provides dynamic group key management service that allow group members to join and leave a group frequently. Furthermore our scheme is able to address data origin authentication and group member authentication without introducing specific additional mechanisms. The rest of this paper is organized as follows. Section 2 introduces the IETF multicast security architecture reference framework[10]. Section 3 presents our key distribution scheme. Section 4 proposes our new securing multicasting scheme. Section 5 compares our scheme with the previously proposed schemes. Finally, section 6 provides the concluding remarks.

II. Multicast Security Architecture Reference Framework

This section reviews briefly the multicast security architecture reference framework proposed by the Internet Engineering Task Force (IETF)[10] [11].

A. Reference Framework

A schematic representation of the Draft Multicast Security Reference Framework is shown in Figure 1. This framework is used to classify and specify the functional areas, functional elements (represented by the boxes), and their interfaces (represented by the arrows).

There are three sets of functional entities and three functional areas. The three sets of functional entities are the Policy Server, Group Controller and Key Server (GCKS), Sender and Receiver. The policy server provides functions to create and manage security policies specific to a multicast group. The Group Controller and Key Server (GCKS) provides functions relating to the management of cryptographic keys used by a multicast group.

Based on the number of senders, multicast is divided into two types, 1-to-N and M-to-N. In a 1-to-N multicast, only one sender can transmit data to a group. In a M-to-N multicast, multiple (or all) group members can transmit data to a group.

0-7803-7788-5/03/$17.00

The three functional areas are multicast data handling, group key management, and the multicast security policies.

Multicast data handling covers issues concerning the security-related treatment of multicast data by the sender and the receiver. Typically, the data is encrypted by a group key and authenticated to a multicast group. The data encryption mainly addresses the issue of confidentiality. The data authentication takes two flavors; (1) source authentication and data integrity, (2) group authentication, which guarantees that data was generated by some group member.

Group key management is concerned with the secure distribution and refreshing of keying material. The keying material refers to the cryptographic key belonging to a group, the state associated with the keys and the other security parameters related to the keys. The problems that should be addressed include: (1) member identification and authentication, (2) verification of the membership to groups, (3) establishment of a secure channel between a GCKS entity and the member, (4) establishment of a long-term secure channel between one GCKS entity and another, (5) the changing of keys and keying material, and (6) detection of signaling failures and perceived compromises to keys and keying material.

The multicast security policies provide aspects of policy in the context of multicast security and must provide the rules for operation for the other elements of the Reference Framework. These include the policy creation, high-level policy translation, and policy representation.

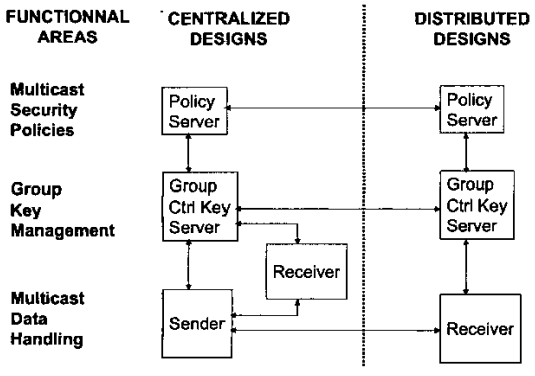

Fig. 1. Multicast Security Architecture Reference Framework

B. Security Service

Several security services are specified for the interfaces of Figure 1. The three security services – multicast data confidentiality, multicast source authentication and data integrity, multicast group authentication – are placed in the functional area of Multicast Data Handling along the interface between Senders and Receivers. Two security services – multicast group member management and multicast key management – are placed in the Group Key Management Area. One security service – multicast policy management – is placed in the Security Policy Management area along the interface between Key Servers and Policy Servers.

C. Group Key Management Architecture

Group Key Management is one of the main aspects of securing multicast. The aim of a group key management protocol is to provide the group members with the up-to-date security association. The Group Security Association Model is shown in Figure 2. The Group Key Management Architecture consists of three protocols: the Registration Protocol, the Re-key protocol, and the Data Security protocol. The Re-key protocol is optional in some cases.

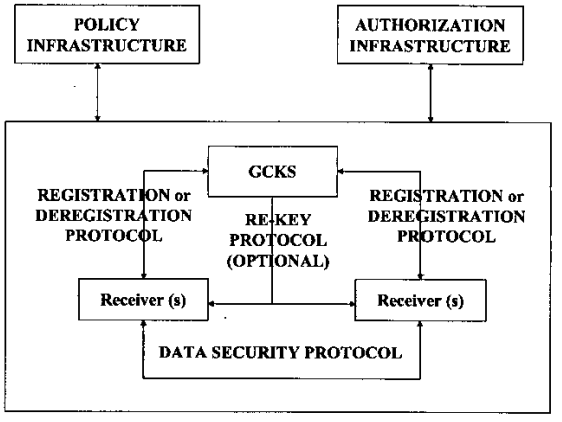

Fig. 2. The Group Security Association Model

There are two types of group key: the KEK (key-encrypting key) and the TEK (traffic-encrypting key). The KEK maybe a single key that encrypts the TEK or a vector of keys that encrypts the TEK and other TEKs. The KEK is established by the Registration Protocol and used by the re-key protocol. The TEK is established by the Re-key Protocol and is used by the Data Security Protocol to protect streams, files or other data sent and received by the Data Security Protocol.

The IETF Group Key Management architecture also provides an implementation diagram. There are several functional blocks to implement the group key management. One is the GKM (Group Key Management) functional block that is used to establish the GSA (Group Security Associations) to use the Registration Protocol and the Re-key protocol. Another one is the CONTROL function that directs the GCKS to establish a group (including "join or leave"). CONTROL includes the authorization subject to Group Policy. CONTROL maybe a telephony signal protocol like SIP. CONTROL could perform the announce functions that can direct group key management using the application programming interface (API). The third functional block is the SECURITY PROTOCOL function that protects the data transmission. It may span inter-networking and application layers. Other function blocks are specific to the operating system (OS), databases such as Security Policy Database (SPD) and Credential Stores (CRED).

III. Key Generation Algorithm

Our approach involves the proposal of a dynamic group key management scheme that enables secure and efficient updating of group members. We achieve this by constructing a public key that is associated with several associated private keys. Our proposal for secure multicasting is based on our earlier work on key distribution described in [12].

A. Preliminaries

The security of our scheme is based on the difficulty of computing discrete logarithms, and the protocols are based on the polynomial functions and a set of exponentials.

Let p be a large prime, $\mathbb{Z}_p^*$ be a multiplicative group of order q for $q|p-1$, and $g \in \mathbb{Z}_q$ for $i = 0, 1, 2, ..., n$ be a set of integers. A polynomial function of order n is constructed as follows: $f(x) = \prod_{i=1}^{n}(x - x_i) \equiv \sum_{i=0}^{n} a_i x^i \bmod q$, where the a_i are coefficients: $a_0 = \prod_{j=1}^{n}(-x_j)$, $a_1 = \sum_{i=1}^{n}\prod_{i\neq j}^{n}(-x_j)$,..., $a_{n-2} = \sum_{i\neq j}^{n}(-x_i)(-x_j)$, $a_{n-1} = \sum_{i=1}^{n}(-x_j)$, $a_n = 1$. Note that $f(x_j) = \sum_{i=0}^{n} a_i x_j^i = 0$. We can use this property to construct a broadcasting encryption system.

Having the set $\{a_i\}$, we can then construct the corresponding exponential functions,

$$\{g^{a_0}, g^{a_1}, g^{a_2}, ..., g^{a_n}\} \equiv \{g_0, g_1, g_2, ..., g_n\}.$$

B. System Setup

The construction of the encryption keys and decryption keys is done as follows:

- Select n distinct random numbers $x_i \in \mathbb{Z}_q$ for $i = 1, 2, \cdots, n$, which form a set X_n and a subset $X_m \subset X_n$.
- Compute $A = \prod_{j=1}^{n}(\prod_{i=0}^{n-1} g_i^{x_j^i}) \bmod p$. Note that A is computed once only. We will see later, a dynamic further updates of the system do not require re-computation of A.
- Select an integer $b \in \mathbb{Z}_q$ and compute its multiplicative inverse b^{-1} such that $bb^{-1} = 1 \bmod q$.
- Compute $\bar{x}_j = b^{-1}\sum_{i\neq j}^{n} x_i^n \bmod q$, for $j = 1, 2, ..., n$.
- Compute $\hat{x}_j = s_j x_j^n$, where $s_j = s_1' s_2' \cdots s_n'$, $s_j s_j' \bmod q = s_j'$, $(s_j, s_j' \in \mathbb{Z}_q)$, and $s_i' \not| s_j'$ for $\forall i, j$.

These values satisfy the equality:

$$A^s g^{sb\bar{x}_j} g^{s\hat{x}_j} = 1, \quad \forall j \in \{1, 2, \cdots, n\}.$$

A is kept by the authorized server and will be used as the encryption key. Since the encryption key is not public, there is no need for us to protect it against any illegal modification.

$\bar{x}_j$ and $\hat{x}_j$ are given to user j as its secret decryption key during the process of its registration. Hence the private decryption key doublet is $(\bar{x}_j, \hat{x}_j)$. Please note that computation of A is a one-time task. The server does not need to modify it during a system update. This is an important feature, since it makes the encryption/decryption processes very efficient (a maximum of 2 or 3 exponential computations).

C. Broadcasting Encryption Protocol

The encryption key A is used to encrypt a session key that is then used to encrypt a message. All members in the group can decrypt the session key and then decrypt the message individually with their private keys. Let us suppose that M is the message to be encrypted and k is a session key.

The protocol is as follows:

- Select an integer $r \in_R \mathbb{Z}_q$.
- Compute $\bar{g} = g^{sr}$ and $\hat{g} = g^{sbr}$.
- Compute the ciphertext $c = E_k(M)$ and $k' = kA^{sr}$, where $E_k(.)$ denotes a symmetric key encryption function.
- Broadcast the 4-tuple $(\bar{g}, \hat{g}, c, k')$ to all subscribers.

To decrypt the session key, the user j computes $k'\hat{g}^{\bar{x}_j}\bar{g}^{\hat{x}_j} = k$. k is then used for the decryption of the message.

IV. Our New Secure Multicast Scheme

In this section, we present our new multicast scheme for 1-to-N multicasting. The GCKS can act as the sender. We concentrate only on the group key management protocol and then discuss the security aspects.

A. Key Registration

The group establishment ladder diagram is shown in Figure 3 [13][12]. Because our scheme involves only 1-to-N multicasting, the sender can act as the controller or the GCKS. The protocol shows how a potential member registers and gets the member key $(\bar{x}_j, \hat{x}_j)$.

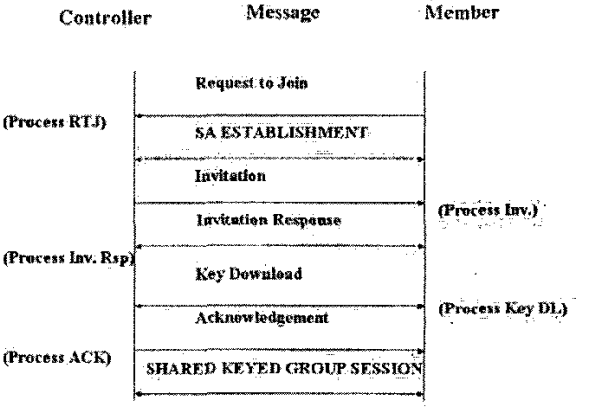

Fig. 3. The Group Establishment Ladder Diagram

The first phase involves a potential member requesting to join the group. The GCKS processes the Request to Join (RTJ) and then establishes the Security Associations (SA). Another mechanism is for the GCKS to invite a prospective member to join the group. The invited prospective member responds to the invitation, and the GCKS processes the response and a agreement is achieved.

The second phase is the key distribution. After the GCKS and the potential member make an agreement, they can use the secure channel such as the IPsec or TLS/SSL to transmit the secret private key to the member.

B. Re-Keying

To remove a member, the GCKS does not need to re-construct the encryption key A. Instead, the GCKS only recomputes s such that s'_γ does not include the member to be removed; the computation is $s = \Sigma_{i=1, i\neq\gamma}^{n} s'_i$. We can still use the protocol above without any modification.

To add new members to the group, the GCKS makes use of an element in the spare set $X_n - X_m$. Recall that we have assumed that the actual number of members is less than the total set. That is, $m < n$ or $X_m < X_n$. Hence to add a new member, the GCKS just simple moves one unused element from $X_n - X_m$ to X_m.

C. Security Keys

As mentioned in section 2, we have two types of group key: the KEK and the TEK. KEK is used for the re-keying protocol and the TEK is used for the data transfer security protocol. In this scheme, there are two ways to encrypt the data. The GCKS with the encryption key (A) acts as both KEK and TEK. In the re-keying protocol, the GCKS recomputes s for an update of the group.

One can also adopt a hybrid approach. The GCKS's encryption key is used for the KEK. The GCKS generates a symmetric key as the TEK to encrypt the message to be transmitted. The message to be transmitted is encrypted with the TEK that is encrypted with the KEK, and is then sent to the assoicated members. The members can decrypt it with their own private key to get the session key TEK, and then use it to decrypt the cipher message.

D. Security Services

As discussed in section 2, the security services are placed in three areas. Three of them are placed in the multicast data handling area. They are the Multicast Data Confidentiality, Multicast Source Authentication and the Multicast Group Authentication. Our scheme uses the asymmetric distributed encryption system; that is, the sender or the members use the asymmetric key. Hence we achieve Multicast Source Authentication and the Multicast Group Authentication. The data is encrypted by the sender's private key, and only qualified members can decrypt it, and hence the Multicast Data Confidentiality is also ensured.

Additional security services in the group key management area include the Multicast Group Member Management and the Multicast Key Management. In our scheme, the GCKS manages the group members and the asymmetric encryption key system, which makes the provision of both these services efficient.

V. Comparison of Our Proposed Secure Multicasting Scheme with Previous Schemes

There are several existing secure multicasting schemes. As we mentioned before, it can be divided into 1-to-N and M-to-N multicast. The existing key distribution scheme can be divided into scalable and non-scalable protocols. The scalable key distribution protocols can be classified as two categories: the Flat scheme and the Hierarchical scheme [4][6]. In this section, we introduce these schemes briefly and then compare our proposed scheme with those schemes.

The flat scheme can be further classified as Centralized Flat schemes and Distributed Flat schemes. Centralized Flat Schemes use a single entity to distribute encryption key to all group members. When a prospective member joins or leaves the group, the group key controller transmits the new session key (TEK) to all members using the secure channel. Examples of this type of schemes are ETM [14], GKMP[15], PDKD [16] and CFKM-DMG[17][18]. In the GKMP (Group Key Management Protocol), the GCKS shares the traffic encryption key (TEK) and the key encryption key (KEK). In the ETM (Elements of Trusted Multicasting), the GCKS sends the encrypted multicast data, and then sends the TEK encrypted with the group members public key. The PDKD (Perfectly Secure Dynamic Conference Key Distribution) uses a secret share scheme for secure group data transmission. The group members compute a common key and any member can identify any other member. In the CFKM-DMG (Centralized Flat Key Management for Dynamic Multicast Group), the GCKS assigns the binary IDs to all members of the group and then generates and 2W KEKs, where W is the number of bits in any member's ID. There is a TEK for all the members of the group.

Distributed Flat Schemes trust all the members equally; the new member can get the encryption key from an earlier joined member and there is no GCKS or manager. There is a distributed version of CFKM-DMG (Centralized Flat Key Management for Dynamic Multicast Group)[17].

The Hierarchical scheme uses a distribution tree to distribute the session key. It can be further classified into hierarchical node based protocols and hierarchical key based protocols. Hierarchical node based schemes use a hierarchy of the nodes to address the scalability issue. Some of this type schemes are SMKD[19], Iolus[20] and DEP[21]. SMKD (Scalable Multicast Key Distribution) protocol uses the Core Based Tree (CBT) architecture for the key distribution. The primary core generates the TEK and the KEK, and then distributes these keys to the secondary core and subsequently to other nodes as they become part of the distribution tree. Iolus proposed the idea of hierarchical subgroup for scalable secure multicasting. The GCKS distributes the secret key to the top-level subgroup. The group security agents (GSA) share a secret key with each of their subgroup members. These secret keys act as the KEK. The TEK is distributed with the multicast data. DEP (Dual Encryption Protocol) also uses the hierarchical subgroup of multicast members to address scalability.

The hierarchical key based scheme uses a hierarchical of key to deal with scalability. Examples of this type of scheme include CTKM[22] and OFT[23]. In CTKM (Centralized Tree-base Key Management) scheme, multicast group members are leaves of the key distribution tree of an arbitrary degree. The internal nodes of the tree represent a KEK. Members share the KEK with the group manager. In the OFT (One Way Function Trees) scheme, each internal node has two children. Each node has a blind version of its key that is

computed with the one-way function. The GCKS generates the blinded key, and the members can compute the rest of the key.

The comparison of the secure multicast schemes is shown in table 1.

TABLE I

COMPARISON OF THE SECURE MULTICAST SCHEMES . n: NUMBER OF MEMBERS, l: NUMBER OF SUBGROUPS, d: DEGREE, c: SIZE OF THE SENDER'S SUBGROUP.

Criteria Property	ETM	GKMP	CFKM	Iolus
Access Control mechanism	Yes	Yes		Yes
Capability certificate	No	Yes		No
No. of keys in the multicast group	$n+1$ $\mathcal{O}(n)$	$2n$ $\mathcal{O}(n)$	$2\log(n+1)$ $\mathcal{O}(\log n)$	$n+l+1$ $\mathcal{O}(n)$
No. of keys managed by sender	2	$2n$	$2\log n+1$	2
No. Of keys at a member	1	2	$\mathcal{O}(\log n)$	3
No. Of keys at SGM	-	-	-	4
Public key /secret key	Public	Secret		Both
Join scalability	Yes	Yes		Yes
Leave scalability	Yes	No		Yes
Data transmission scalability	No	Yes		Yes
1 affects n scalability	Yes	No	No	Yes
No. of message at join	$\mathcal{O}(l)$	$\mathcal{O}(1)$	$\mathcal{O}(\log n)$	$\mathcal{O}(l)$
No. of message at leave	$\mathcal{O}(l)$	$\mathcal{O}(n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(l)$
Total key encryption during data transmission	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(l)$	$\mathcal{O}(l)$
No. of key encryptions at the sender	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(l)$	$\mathcal{O}(l)$

Criteria Property	DEP	CTKM	Our New Scheme
Access Control mechanism	Yes	Yes	Yes
Capability certificate	Yes	No	
No. of keys in the multicast group	$n+l+c+1$ $\mathcal{O}(n)$	$\frac{dn+1}{d-1}$ $\mathcal{O}(n)$	$n+1$ $\mathcal{O}(n)$
No. of keys managed by sender	$c+2$	$\frac{dn+1}{d-1}$	n
No. Of keys at a member	4	$\mathcal{O}(\log dn)$	1
No. Of keys at SGM	5	-	-
Public key /secret key	Both	Secret	Both
Join scalability	Yes	Yes	Yes
Leave scalability	Yes	Yes	Yes
Data transmission scalability	No	Yes	Yes
1 affects n scalability	Yes	No	Yes
No. of message at join	$\mathcal{O}(1)$	$\mathcal{O}(\log dn)$	1
No. of message at leave	$\mathcal{O}(l)$	$\mathcal{O}(\log dl)$	1
Total key encryption during data transmission	$\mathcal{O}(c+l)$	$\mathcal{O}(1)$	2
No. of key encryptions at the sender	$\mathcal{O}(c)$	$\mathcal{O}(1)$	2

An important feature of our scheme is the "1 affects n" scalability. In our scheme, members joining or leaving the group do not affect other members at all. This is a significant issue both in terms of scalability and dynamic group management. This is also advantageous for the sender because s/he needs to do much fewer encryption computations and send fewer messages. Furthermore, it greatly simplifies the KEK transfer process.

VI. CONCLUDING REMARKS

We have presented a new secure multicast scheme based on a novel key distribution scheme. Our scheme possesses several new features that are significant in dynamic group key management. It enables efficient joining or leaving of group members without affecting the rest of the group members. This is an important characteristic when it comes to large scale systems involving several millions of users. Our scheme is computationally efficient, since it involves a relatively few (2 or 3) encryption/decryption operations per update. We believe that the proposed scheme is in general applicable to many other multicasting applications.

REFERENCES

[1] C. Kenneth Miller, *Multicast networking and Applications*, Addison Wesley Longman, Inc., September 1998.

[2] ISO/IEC 7498, Security Architecture, part2, 1994.

[3] T. Ballardie. and J. Crowcroft, "Multicast-specific security threats and counter-measures," in *Symposium on Network and Distributed System Security*. February 1995, pp. 2–16, San Diego, Califonia.

[4] Lakshminath R. Dondeti, Sarit Mukherjee, and Ashok Samal, "Scalable secure one-to-many group communication using dual encryption," *Computer Communications Journal*, November 2000.

[5] T. Hardjono and G. Tsudik, "Ip multicast security: Issues and directions," *Annales de Telecom*, pp. 324–340, July-August 2000.

[6] Lakshminath R. Dondeti, Sarit Mukherjee, and Ashok Samal, "Survey and comparison of secure group communication protocols," .

[7] Marcel Waldvogel, Germano Caronni, Dan Sun, Nathalie Weiler, and Bernhard Plattner, "The versakey framework: Versatile group key management," *IEEE JSAC Special Issue on Service Enabling Platforms For Networked Multimedia Systems*, 17(8),August 1999.

[8] Peter S. Kruus, "A survey of multicast security issues and architecture," in *21st national Information Systems Security conference*, 1998.

[9] Peter S. Kruus and Joseph P. Macker, "Techniques and issues in multicast security," *MILCOM 98*, 1998.

[10] T. Hardjono and B. Weis, "Msec architecture, draft-ietf-msec-arch-00.txt," Oct 2002, Work in Progress.

[11] M. Baugherand, R. Canetti, L. Dondeti, and F. Lindholm, "Group key mangement architecture, draft-ietf-msec-gkmarch-03.txt," Feb 2002, Work in Progress.

[12] Yi Mu and Vijay Varadharajan, "Robust and secure broadcasting," in *Indocrypt 2001, Lecture Notes in Computer Science*, Springer 2001.

[13] H. Harney, A. Colegrove, E. Harder, U. Meth, and R. Fleischer, "Group secure association key management protocol (gsakmp), draft-irtf-smug-gsakmp-00.txt," November 2000, Work in Progress.

[14] L. Gong and N. Shacham, "Elements of trusted multicasting," in *IEEE Intl. Conf. On Network Protocols*, Boston, MA, USA, October 1994, pp. 23–30.

[15] H. Harney and C. Muchenhirn, "Group key management protocol (gkmp) architecture," RFC 2094, July 1997.

[16] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *Information and Computation*, December 1997.

[17] G. Caronni, M. Waldvogel, D. Sun, and B. Plattner, "Efficient security for large and dynamics groups," *Technical Report TIK Technich Report No.41, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology*, February 1998.

[18] I. Chang, R. Engel, D. Kandlur, D.Pendarakis, and D. Saha, "Key management for secure internet multicast using boolean function minimization techniques," in *IEEE INFOCOM*, New York, March 1999.

[19] T. Ballardie, "Scalable multicast key distribution," RFC 1949, May 1996.

[20] S. Mittra, "Iolus: A framework for scalable secure multicasting," in *ACM SIGCOMM*, Cannes, France, September 1997, pp. 177–288.

[21] L. R. Dondeti, S. Mukherjee, and A. Samal, "A dual encryption protocol for scalable secure multicasting," in *The Fourth International Symposium on Computer and Communications*, July 1999.

[22] C. K. Wong, M. Gouda, and S. S. lam, "Securing group communications using key graphs," in *ACM SIGCOMM*, August 1998.

[23] D. A. McGrew and A. T. Sherman, "Key establishment in large dynamic groups using one-way function trees," .