Establishing Security for Group Communication in Multicast Environment

B.V.S.S.R.S.Sastry, K.Akshitha, M.V.Vijaya Saradhi

***Abstract*— Multicast is an internetwork service that provides efficient delivery of data from a source to multiple receivers. It reduces the bandwidth requirements of the network and the computational overhead of the host devices. This makes multicast an ideal technology for communication among a large group of participants. Secure group communications involves many service types include teleconferencing, pay TV and real-time delivery of stock quotes. IP multicast is the traditional mechanism to support multicast communications. Multicast security includes group membership control, secure key distribution, secure data transfer and copyright protection. This paper is an overview of the schemes proposed for group key management, authentication and watermarking in wired networks with fixed members and wireless networks with mobile members.**

***Keywords*- multimedia, key management, authentication, multicast security, watermarking, access control, copyright protection.**

I. INTRODUCTION

The availability of digital technologies and widening Internet bandwidth in recent years have increased the demand for new multimedia services. The Internet service providers are now deploying the new technologies for group communications that allow the participation of many members. Service types include teleconferencing, pay-per-view, video-on-demand, interactive simulation, software updates and real-time delivery of stock market information.[1] Multimedia security is an important requirement for the distribution networks when the delivery includes either confidential or copyrighted data. With the deployment of digital technologies for the reproduction, storage and distribution of content, there is a growing need for the protection of intellectual property.

Content providers (movie studios and recording studios, in particular) have been evaluating the technologies that prevent unauthorized copying in major ways of distribution (satellite, cable and terrestrial systems, the Internet and prerecorded magnetic and optical media). The traditional mechanism to support multicast communications is IP multicast (Miller 1999)[1][3]. It uses the notion of a group of members identified with a given group address. When a sender sends a message to this group address, the network uses a multicast routing protocol to optimally replicate the message and forward copies to group members located throughout the network.

Although the Internet community began discussing architectural issues in mid-80's using Internet Engineering Task Force (IETF) Request for Comments (RFCs), significant activity in multicast IP did not occur until the creation of the Mbone in 1992.[2][4] The Mbone is a set of multicast-enabled subnetworks connected by IP tunnels. Tunneling is a technique that allows multicast traffic to traverse parts of the network by encapsulating multicast datagrams within unicast datagrams. In IPv4, multicast IP addresses are defined by Class D which differs from Classes A, B and C that are used for point-to-point communications. The multicast address space, assigned by the Internet Assigned Numbers Authority (IANA), covers the range (224.0.0.0 – 239.255.255.255). IPv6 has 128 bits of address space compared with 32 bits in IPv4. The Internet Group Management Protocol (IGMP) defines a protocol for multicast enabled hosts and routers to manage group membership information.[5] Developed by the Defense Advance Research Projects Agency (DARPA), the Transmission Control Protocol/Internet Protocol (TCP/IP) connects networks designed by different vendors into a network of networks, i.e., the Internet. It has two transport layers for the applications: The Transport Control Protocol (TCP) and the User Datagram Protocol (UDP). Currently, UDP is the only protocol for IP multicast, providing minimal services such as port multiplexing and error detection. Any host can send a UDP packet to a multicast address, and the multicast routing mechanism will deliver the packet to all members of the multicast group. TCP provides a higher level of service with packet ordering, port multiplexing and error-free data delivery. It is a *connection-oriented* protocol (unlike UDP which is *connectionless*), and does not support multicast applications. MSEC is a Working Group (WG) in the Internet Engineering Task Force (IETF)[7]. Its purpose is to "*standardize protocols for securing group communication over internets, and in particular over the global Internet*." The initial primary focus of the MSEC WG will be on scalable solutions for groups with a single source and a very large number of recipients. The standard will be developed with the assumption that each group has a single trusted entity (i.e., the Group Controller) that sets the security policy and controls the group membership. It will attempt to guarantee at least the following two basic security features:

- Only legitimate group members will have access to current group communication (This includes groups with highly dynamic membership).

- Legitimate group members will be able to authenticate the source and contents of the group communication (This includes cases where group members do not trust each other).

Manuscript Received October 25, 2011.

B.V.S.S.R.S.SASTRY, IT Department, Aurora's Engineering College, Bhuvanagiri, Andhra Pradesh, India,(e-mail: sastry_38@yahoo.com).
K.Akshitha, IT Department, Aurora's Engineering College, Bhuvanagiri, Andhra Pradesh, India(e-mail: koluguri.87@gmail.com).
Dr. M.V.Vijaya Saradhi, IT Department, Aurora's Engineering College, Bhuvanagiri, Andhra Pradesh, India (e-mail: meduri_vsd@yahoo.co.in).

In this paper, we will look at the recent developments in key management, authentication and watermarking for secure group communications in wired and wireless networks. The proposed methods provide solutions to address three different issues of secure multimedia data distribution:

- Controlling access to multimedia data among group members,
- Assuring the identity of participating group members (senders or receivers),
- Providing copyright protection.

Figure 1 depicts some of the challenging questions regarding these issues.

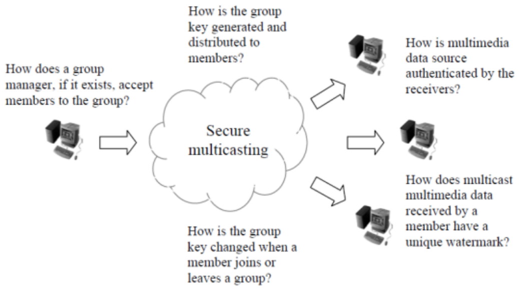
How is the group key generated and distributed to members?
How does a group manager, if it exists, accept members to the group?
Secure multicasting
How is multimedia data source authenticated by the receivers?
How does multicast multimedia data received by a member have a unique watermark?
How is the group key changed when a member joins or leaves a group?

Figure 1. Challenges in secure multicast communications

II. MULTICAST SECURITY

Secure multicast communications in a computer network involves efficient packet delivery from one or more sources to a large group of receivers having the same security attributes[8]. The four major issues of IP multicast security are (Hardjono and Tsudik 2000):

- *Multicast data confidentiality*: As the data traverses the public Internet, a mechanism is needed to prevent unauthorized access to data. Encryption is commonly used for data confidentiality.
- *Multicast group key management*: The security of the data packets is made possible using a group key shared by the members that belong to the group. This key needs to change every time a member joins (leaves) the group for backward access control (forward access control). In some applications, there is also a need to change the group key periodically. Encryption is commonly used to control access to the group key.
- *Multicast data source authentication*: An assurance of the identity of the data source is provided using cryptographic means. This type of authentication also includes an evidence of data integrity. Digital signatures and Message Authentication Codes (MACs) are common authentication tools.
- *Multicast security policies*: The correct definition, implementation and maintenance of policies governing the various mechanisms of multicast security is a critical factor. The two general categories are the policies governing group membership and the policies regarding security enforcement.

In multicast communications, a session is defined as the time period in which data is exchanged among the group members. The type of member participation characterizes the nature of a session. In a *one-to-many* application, data is multicast from a single source to multiple receivers. Pay-per-view, news feeds and real-time delivery of stock market information are a few examples. A *many-to-many* application involves multiple senders and multiple receivers. Applications such as teleconferencing, white boarding and interactive simulation allow each member of the multicast group to send data as part of group communications.

III. WIRED NETWORK SECURITY

A. Key Management Schemes for Wired Networks

Many multicast key management schemes have been proposed in the last 10-15 years. Three classifications from the literature are:

1. *Non-scalable* and *scalable* schemes (Dondeti et al. 1999a). The scalable schemes are in turn divided into three groups: Hierachical key management (node-based and key-based), centralized flat key management and distributed flat key management.

2. *Flat* schemes, *clustered* schemes, *tree-based* schemes and other schemes (Bruschi and Rosti 2000).

3. *Centralized* schemes, *distributed subgroup* schemes and *distributed* schemes (Rafaeli 2000).

We propose a new classification using two criteria - the entity who exercises the control and whether the scheme is scalable or not: *Centralized group control, subgroup control and member control*. a) *Centralized group control*: A single entity controls all the members in the group. It is responsible for the generation, distribution and replacement of the group key[11]. As the controlling server is the single point of failure, the entire group is affected as a result of a malfunction. b) *Subgroup control*: The multicast group is divided into smaller subgroups, and each subgroup is assigned a different controller[13][15]. Although decentralization substantially reduces the risk of total system failure, it relies on trusted servers, weakening the overall system security. c) *Member control*: With no group or subgroup controllers, each member of the multicast group is trusted with access control and contributes to the generation of the group key. Each of the above classes is further divided into *scalable* and *non-scalable* schemes. In the context of multicast key management, scalability refers to the ability to handle a larger group of members without considerable performance deterioration. A scalable scheme is able to manage a large group over a wide geographical area with highly dynamic membership. If the computation and communication costs at the sender increases linearly with the size of the multicast group, then the scheme is considered to be non scalable [14]. *Hierarchical key distribution trees* form an efficient group of proposals for scalable secure multicasting. They can be classified into two groups: *hierarchical key based* schemes and *hierarchical node based* schemes. A hierarchical key based scheme assigns a set of keys to each member depending on the location of the member in the tree. Hierarchical node based schemes define internal tree nodes that assume the role of subgroup managers in key distribution.

B. Periodic Batch Rekeying

In spite of the efficiency of the tree-based scalable schemes for one-to-many applications, changing the group key after each join or leave, i.e., individual *rekeying*, has two major drawbacks:

synchronization problem and inefficiency (Yang et al. 2001).

• *Synchronization problem*: If the group is rekeyed after each join or leave, synchorization will be difficult to maintain because of the interdependencies among rekey messages and also between rekey and data messages. If the delay in rekey message delivery is high and the join/leave requests are frequent, a member may need to have memory space for a large number of rekey and data messages that cannot be decrypted.

• *Inefficiency*: For authentication, each rekey message may be digitally signed by the sender. Generation of digital signatures is a costly process in terms of computation and communication. A high rate of join/leave requests may result in a performance degradation. One particular study attempts to minimize these problems with *periodic batch rekeying* (Yang et al. 2001). In this approach, join/leave requests are collected during a rekey interval and are rekeyed in a batch. The out-of-sync problems are alleviated by delaying the use of a new group key until the next rekey interval. Batch processing also leads to a definite performance advantage. For example, if digital signatures are used for data source authentication, the number of signing operations for J join and L leave requests is reduced from J+L to 1. Periodic batch rekeying provides a trade-off between performance improvement and delayed group access control. A new member has to wait longer to join the group and a leaving member can stay longer with the group. The period of the batch rekeying is thus a design parameter that can be adjusted according to security requirements. To accommodate different application needs, three modes of operation are suggested:

- *Periodic batch rekeying*: The key server processes both join and leave requests periodically in a batch.
- *Periodic batch leave rekeying*: The key server processes each join request immediately to reduce the delay for a new member to access group communications but processes leave requests in a batch.
- *Periodic batch join rekeying*: The key server processes each leave request immediately to reduce the exposure to members who have left but processes join requests in a batch.

A *marking algorithm* is proposed to update the key tree and generate a rekey subtree at the end of each rekey interval with a collection of J join and L leave requests. A rekey subtree is formed using multiple paths corresponding to multiple requests. The objectives of the marking algorithm are to reduce the number of encrypted keys, to maintain the balance of the updated key tree, and to make it efficient for the users to identify the encrypted keys they need [10]. To meet these objectives, the server uses the following steps:

1. Update the tree by processing join and leave requests in a batch. If $J \leq L$, J of the departed members with the smallest IDs are replaced with the J newly joined members. If $J > L$, L departed members are replaced with L of the newly joined members. For the insertion of the remaining $J – L$ new members, three strategies have been investigated (Li et al. 2001; Zhang, Lam et al. 2001).

2. Mark the key nodes with one of the following states: Unchanged, Join, Leave and Replace.

3. Prune the tree to obtain the rekey subtree.

4. Traverse the rekey subtree, generate new keys, and construct the rekey message.

C. Balanced Key Trees

The efficiency of a tree-based key management scheme depends highly on how well the tree remains balanced. In this context, a tree is balanced if the difference between the distances from the root node to any two leaf nodes does not exceed 1 (Moyer et al. 1999a). For a balanced binary tree with *n* leaves, the distance from the root to any leaf is log2*n*. The issue of maintaining trees in a balanced manner is critical for any real implementation of a key management tree. Several techniques, based on the scheme described by Wallner et al, are introduced to maintain a balanced tree in the presence of arbitrary group membership updates (Moyer et al. 1999a). The following procedures are used by the server for adding a new member to a group and deleting an existing member from a group[18].

Given: Each interior node contains four pieces of information: the node key, a boolean key update flag, the distance and direction to the shallowest descendant leaf, and the distance and direction to the deepest descendant leaf.

Procedure for adding a new member

1. Find the shallowest leaf LS of the tree (in case of a tie, any one of the leaves can be chosen).

2. Create a new interior node NI, insert it at the location of LS, and make LS a child of NI.

3. Create a new member node C, and insert it as the other child of NI.

4. Trace the path from node C to the root, and perform the following tasks at each node in the path:

• Update the distance and direction to the shallowest and deepest descendant leaves.

• Set the key update flag to TRUE.

5. Retrace the path from node C to the root, and perform the following tasks at each node that has its key update flag set to TRUE:

• Generate a new node key.

• Create two key update messages for this key, encrypting the first message with the key of the left child node and encrypting the second message with the key of the right child node.

• Digitally sign both messages with the private key.

• Reset the node’s key update flag to FALSE.

6. Update the keys in the same order used in the Wallner et al scheme.

Assuming that the tree is balanced, the following costs are incurred by the above operations for a group size of *n*:

Computation cost:

• *Insertion of new interior node and member node*: O(log *n*), i.e., O(log *n*) time to locate the insertion

point and constant time to create and insert the new nodes.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

• *First trip*: $O(\log n)$, i.e., constant time to update the data in each node, and there are $O(\log n)$ nodes.

• *Second trip*: $O(\log n)$ - similar to the first trip.

Communication cost: 2 $O(\log n)$, i.e., the number of multicast messages sent.

Procedure for deleting an existing member

1. If (the number of leaves = 1) then delete the leaf

else locate the node C of the member to be deleted.

2. Delete C and the interior node P that is the parent of C.

3. Move S, the sibling of C, up to the location formerly occupied by P.

4. Trace the path from the new parent of S to the root, and perform the following tasks at each node:

• Update the distance and direction to the shallowest and deepest descendant leaves.

• Set the key update flag to TRUE.

5. Retrace the path from the new parent of S to the root, and perform the following tasks at each node that has its key update flag set to TRUE:

• Generate a new node key.

• Create two key update messages for this key, encrypting the first message with the key of the left child node and encrypting the second message with the key of the right child node.

• Digitally sign both messages with the private key.

6. Update the keys in the same order used in the Wallner et al scheme.

The computation and communication costs for the operations needed to delete a member are similar to those for member addition. The above cost figures have been obtained with the assumption that the tree is always balanced. This assumption, however, is not completely valid. Although we have complete control over how the tree is edited for new member additions, there is no way to predict the locations in the tree at which the deletions will occur [20]. Hence, it is possible to imagine extreme cases leading to costs that have linear order in the size of the group. Two simple tree rebalancing schemes have been proposed to avoid this cost increase (Moyer et al. 1999a). The first is a modification of the deletion algorithm; the other allows the tree to become imbalanced after a sequence of key updates and periodically invokes a tree rebalancing algorithm to bring the tree back to a balanced state.

D. Authentication

In multicast architectures, group membership control, dictated by security policies, allows access to a secure multicast group. *Member authentication* involves methods ranging from the use of access control lists and capability certificates (Dondeti et al. 1999a) to mutual authentication (Menezes et al. 1997) between the sender and the receiver.

• *Access control lists*: The sender maintains a list of hosts who are either authorized to join the multicast group or excluded from it. When a host sends a join request, the sender checks its identity against the access control list to determine if membership is permitted. The maintenance of the list is an important issue as the list may be changing dynamically based on new authorizations or exclusions.

• Capability *certificates*: Issued by a designated Certificate Authority, a capability certificate contains information about the identity of the host and the set of rights associated with the host. It is used to authenticate the user and allow group membership.

• *Mutual authentication*: The sender and the host authentication each other via cryptographic means. Symmetric or public key schemes can be used for this purpose. A challenging problem in secure group communications is *data source authentication*, i.e., providing assurance of the identity of the sender and the integrity of the data. Depending on the type of multicast application and the computational resources available to the group members, three levels of data source authentication can be used (Moyer et al. 1999b):

• *Group authentication*: Provides assurance that the packet was sent by a registered group member (a registered sender or a registered receiver).

• *Source authentication*: Provides assurance that the packet was sent by a registered sender (and not by a registered receiver).

• *Individual sender authentication*: Provides assurance of the identity of the registered sender of the packet.

In a naive approach, each data packet can be digitally signed by the sender. For group (source) authentication, all members, sender or receiver (all senders), can share a private key to generate the same signature on the packets. Individual sender authentication, however, requires each sender to have a unique private key. Although digital signature-based authentication per packet is desirable as a reliable tool, it exhibits a poor performance because of lengthy keys and computational overhead for signature generation and verification. Recent research has led to more efficient authentication methods, including

• *multiple Message Authentication Codes (MACs)* (Canetti et al. 1999)

• *stream signing* (Gennaro and Rohatgi 1997)

• *authentication tree-based signature*s (Wong and Lam 1998)

• *hybrid signatures* (Rohatgi 1999)

• *TESLA and BiBa* (Perrig et al. 2000; Perrig et al. 2001; Perrig 2001)

A Message Authentication Code (MAC) is a keyed hash function used for data source authentication in communication between two parties (sender and receiver). At the source, the message is input to a MAC algorithm which computes the MAC using a key K shared by both parties. The sender then appends the MAC to the message, and sends the pair {message|MAC} to the receiver. In an analysis of the generalization of MACs to multicast communications, it is shown that a short and efficient collusion resistant multicast MAC (MMAC) cannot be constructed without a new advance in digital signature design (Boneh et al. 2001).

E. Watermarking

Watermarking (data hiding) (Swanson et al. 1998; Petitcolas et al. 1999) is the process of embedding data into a multimedia element such as image, audio or video.

This embedded data can later be extracted from, or detected in, the multimedia for security purposes. A watermarking algorithm consists of the watermark structure, an embedding algorithm and an extraction, or a detection, algorithm. Watermarks can be embedded in the pixel domain or the transform domain. In multimedia applications, embedded watermarks should be invisible, robust and have a high capacity (Hartung and Kutter 1999). Invisibility refers to the degree of distortion introduced by the watermark and its affect on the viewers or listeners. Robustness is the resistance of an embedded watermark against intentional attacks and normal A/V processes such as noise, filtering (blurring, sharpening, etc.), resampling, scaling, rotation, cropping and lossy compression. Capacity is the amount of data that can be represented by an embedded watermark. The approaches used in watermarking still images include: least-significant bit encoding, basic Sequence, transform techniques and image-adaptive techniques (Wolfgang et al. 1999). As video watermarking possesses additional requirements, development of more sophisticated models for the encoding of video sequences is currently being investigated. Typical uses of watermarks include *identification of the origin of content, tracing illegally distributed copies and disabling unauthorized access to content*.[15] Requirements and characteristics for the digital watermarks in these scenarios are different, in general. Identification of the origin of content requires the embedding of a single watermark into the content at the source of distribution. To trace illegal copies, a unique watermark is needed based on the location or identity of the recipient in the multimedia network. In both of these applications, watermark extraction or detection needs to take place only when there is a dispute regarding the ownership of content. For access control, the watermark should be checked in every authorized consumer device used to receive the content. Note that the cost of a watermarking system will depend on the intended use and may vary considerably.

The *copyright protection* problem in a multicast architecture raises a challenging issue. All receivers in a multicast group receive the same watermarked content. If a copy of this content is illegally distributed to the public, it may be difficult to find the parties responsible for this criminal act. Such a problem can be eliminated in a unicast environment by embedding a unique watermark for each receiver. To achieve uniqueness for multicast data, two distinct approaches are feasible:

1. multiple copies of content, each with a different watermark, are created to allow the selection of appropriate packets in distribution,

2. a single copy of unwatermarked content is created to allow the insertion of appropriate watermarks in distribution.

The following proposals are variations of these two approaches:

• *A different version of video for each group member* (Chu et al. 1999): For a given multicast video, the sender applies two different watermark functions to generate two different watermarked frames, d*i,w0* and d*i,w1*, for every frame *i* in the stream. The designated group leader assigns a randomly generated bit stream to each group member. The length of the bit string is equal to the number of video frames in the stream. For the *i*th watermarked frame in stream *j*, *j* = 0,1, a different key *Ki,j* is used to encrypt it. The random bit stream determines whether the member will be given *Kio* or *Ki1* for decryption. If there is only one leaking member, its identification is made possible with the collaboration of the sender who can read the watermarks to produce the bit stream and the group leader who has the bit streams of all members. The minimum length of the retrieved stream to guarantee a c-collusion detection, where c is the number of collaborators, is not known. An important drawback of the proposal is that it is not scalable and two copies of the video stream need to be watermarked, encrypted and transmitted.

• *Distributed watermarking (Watercasting)* (Brown et al. 1999): For a multicast distribution tree with maximum depth *d*, the source generates a total of *n* differently watermarked copies of each packet such that $n \geq d$. Each group of *n* alternate packets is called a transmission group. On receiving a transmission group, a router forwards all but one of those packets to each downstream interface on which there are receivers. Each last hop router in the distribution tree will receive *n-dr* packets from each transmission group, where *dr* is the depth of the route to this router. Exactly one of these packets will be forwarded onto the subnet with receivers. The goal of this filtering process is to provide a stream for each receiver with a unique sequence of watermarked packets. The information about the entire tree topology needs to be stored by the server to trace an illegal copy. A major potential problem with watercasting is the support required from the network routers. The network providers may not be willing to provide a security-related functionality unless video delivery is a promising business for them.

• *Watermarking with a hierarchy of intermediaries* (Judge and Ammar 2000): WHIM Backbone (WHIM-BB) introduces a hierarchy of intermediaries into the network and forms an overlay network between them. Each intermediary has a unique ID which is used to define the path from the source to the intermediary on the overlay network. The Path ID is embedded into the content to identify the path it has traveled. Each intermediary embeds its portion of the Path ID into the content before it forwards the content through the network. A watermark embedded by a WHIM-BB identifies the domain of a receiver. WHIM-Last Hop (WHIM-LH) allows the intermediaries to mark the content uniquely for any child receivers they may have. Multiple watermarks can be embedded using modified versions of existing algorithms. The above two “fingerprinting” schemes (Chu et al. 1999; Brown et al. 1999) require a certain number of video frames in order to deduce sufficient information about the recipient whereas WMIN requires only one frame since the entire trace is embedded into each frame. A serious overhead for this scheme, however, is the hierarchy of intermediaries needed for creating and embedding the fingerprint.[16] Lastly, the two techniques described below appear to be viable approaches for copyright protection and access control, respectively.

• *Hierarchical tagging and bulk tagging* (Caronni and Schuba 2001): Hierarchical tagging allows an artist to insert a different watermark for each of his distributors. Similarly, each distributor can insert a watermark for several sub-distributors. This process can continue until the individual customers receive tagged content identifying the artist and all the distributors in the chain. In practice, however, more than a few layers of watermarks may reduce the visual quality to an unacceptable level. With bulk-tagging, the distributor creates multiple, tagged versions of the data. The contents are hidden using cryptographic techniques, and distributed as a single data set.

Each customer receives the same data set, performs some preprocessing and retrieves only the tagged data prepared for him. A simple approach is described to show the feasibility of bulk-tagging for images. It requires registration with the producer and the delivery of keys to decrypt the consumer’s individually tagged copy. The preprocessing required by the client device creates a weakness in system security as the individual tag is used for access control only. If the decryption keys are recovered for one consumer, the content would become available in-the-clear, and there would be no trace to the illegal distributor.

IV. WIRELESS NETWORK SECURITY

Key management in wireless networks is a more complicated problem because of the mobility of group members (Dondeti et al. 2001; Griffin et al. 2002; DeCleene et al. 2001). When a member joins or leaves a session, the group key needs to change for backward confidentiality and forward confidentiality. Since secure data cannot be communicated during the rekeying process, an important requirement for a key management scheme is to minimize the interruption in secure data communications. Mobility also allows the members to move to other networks without leaving the session. The existence of a member whose position changes with time adds another dimension of complexity to the design of rekeying algorithms. A common approach in designing a scalable multicast service is to use a hierarchical structure in group key distribution. The hierarchical key management schemes fall into two major groups (Dondeti et al. 1999a): *Logical* hierarchy of keys and *physical* hierarchy of servers. These schemes divide the key management domain into smaller areas in order to distribute the processing workload. Members of the multicast group belong to a key distribution tree having a root at the sender. In *hierarchical key based* schemes, the set of keys kept by a member is determined by the location of the member in the tree. In *hierarchical node based* schemes, internal tree nodes assume the role of subgroup managers in key distribution. For mobile members, the latter approach is more appropriate. Consider the mobility framework in Figure 2. All the members in the group belong to a “*domain*,” denoted by the collection of pentagons, managed by a *Domain Key Distributor* (DKD). The domain is divided into several independent “*areas*,” each managed by an *Area Key Distributor*. An area is defined in such a way that member movement within an area does not require any rekeying, and a join/leave is handled locally by an *intra-area* rekeying algorithm. When a member moves between the areas, *interarea* rekeying algorithms provide the coordination for the transfer of security relationships.

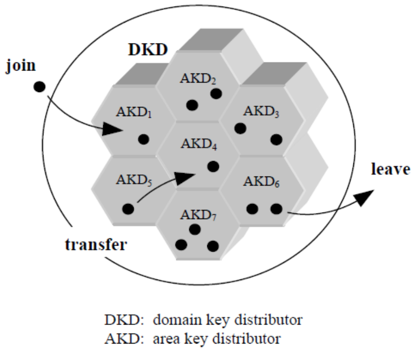

Figure 2. Mobility framework

The DKD generates the *data encryption key* (DEK) for the session and distributes it to all AKDs. Each AKD is responsible for distributing the DEK to its members. As the distribution of the DEK has to be secure, it is protected by a local *key encryption key* (KEK). For intra-area rekeying, several approaches, including the hierarchical key based schemes, can be used. We will now summarize the three operations: join, leave and transfer (Zhang, DeCleene et al. 2001).

Joining the group via area i: When a member joins the group via area *i*, it sends a signaling message to AKD*i* to notify AKD*i* of its arrival. AKD*i* creates a new KEK*i* and securely distributes it to area *i* existing members and the new member. Once the new KEK*i* is in place, the new DEK can be securely multicast among the AKDs and then from each AKD to area members.

Leaving the group via area i: When a member leaves the group via area *i*, all AKDs, *j*, for which the departing member holds a valid key KEK*j* must be notified. A new KEK*j* is created and securely distributed to remaining members for all areas, *j*, for which the departing member holds a valid key KEK*j*. Once the new KEK*j*s are in place, the new DEK can be securely multicast among the AKDs and then from each AKD to area members.

V. OPEN ISSUES AND CONCLUSIONS

A number of schemes has been proposed for secure distribution of the group key to multicast group members. The architectures for wired networks can be extended to wireless networks by addressing the mobility of group members. Our conclusions and the current open issues in multicast security (wired or wireless) include the following:

Wired networks

• Some of the group key management schemes address the problem of join secrecy, i.e., preventing the joining member from having access to past communications, but propose no efficient solutions for leave secrecy, i.e., preventing the leaving member from having access to future communications (Dondeti et al. 1999b; Harney and Muckenhirn 1997; Ballardie 1996). Both types of secrecy are essential in a complete key management scheme, and each should be provided in a scalable and inexpensive way.

• Many multicast applications may require frequent group key updates without waiting for rekeying after joins or leaves. An example is multimedia content, e.g., a 2-hour movie, in a pay-per-view application. In conditional access systems, which protect A/V data in satellite and cable distribution networks, the content descrambling key changes every few seconds to increase robustness against cryptanalytic attacks (the period is normally between 2-10 seconds). The content providers may require the same level of security in multicast applications as well. Most key management schemes do not include efficient rekeying algorithms. The workload may vary substantially in different schemes, as shown below:

− CTKM (Wong et al. 1997): The number of messages the group manager has to send is equal to the number of children of the group manager. For each child, the message would contain the new group key encrypted with the node key belonging to the child.

− DEP (Dondeti et al. 1999b): Replacement of the KEKs and the group key is a complex and costly procedure, and is expected to be done infrequently.

− IOLUS (Mittra 1997): The new subgroup key for each subgroup is multicast encrypted under the old subgroup key. This creates a chain of ciphertexts which is a major cryptanalytic weakness. A compromise in one link would result in the recovery of all the keys used in the following links.

− CKMSS (Eskicioglu and Eskicioglu 2002; Eskicioglu and Delp 2002; Eskicioglu et al. 2003): Only an activating share is multicast to the entire group in-the-clear. The activating share is used by the members to derive the new group key.

• In hierarchical key based schemes, join and leave operations may result in an imbalanced tree over time. There has been some work in tree balancing (Moyer et al. 1999a), but this topic has not received much attention, probably because the tree-based approaches are relatively new.

• Data source authentication is a major issue in multicast security. Most of the proposed authentication mechanisms are based on MACs and digital signatures. Current research is focused on scalable solutions for the three levels of authentication.

• Hierarchical key distribution schemes are compared in a study using the encryption/decryption cost as the performance metrics. This comparison shows the performance advantage of hierarchical node based schemes which increases with the size of the multicast group.

• In large multicast groups, it is very difficult to achieve security. Secure distribution of the group key is only a part of the solution and does not address key compromises inside the group. Detection of traitors is therefore an important requirement in applications where the source of the leak needs to be raced (Chor et al. 2000). A traitor in this context is an authorized user who allows unauthorized parties to obtain content.

• Encryption and watermarking are two groups of technologies used in developing technical solutions for the copy protection problem in DHNs (Bell 1999; Bloom et al. 1999; Eskicioglu and Delp 2001; Eskicioglu, Town et al. 2001). The former, the *first line of defense*, makes the content unintelligible through a reversible mathematical transformation based on a key. The latter, the *second line of defense*, inserts data directly into the content at the expense of imperceptible degradation in quality. Depending on the purpose of the embedded watermark, there is an essential difference between the functionalities of the consumer electronics devices:

− *Copyright protection*: The open literature on watermarking has so far focused on copyright protection for which the receiver does not have to assume an active role in responding to the watermark. When a dispute arises regarding the ownership of content, the watermark needs to be detected or extracted by authorized entities such as the legal institutions.

− *Access control*: The use of watermarking for content protection has been the subject of prolonged discussions at the Copy Protection Technical Working Group (CPTWG) meetings in California in the last few years. The three industries (information technology, consumer electronics and motion picture) have agreed in principle to implement a watermarking system in DVD playback and recording devices. According to a set of principles, the playback and recording devices will detect and respond to watermarks representing the Copy Generation Management System (CGMS) bits (“11” (copy-never), “10” (copy-once), “01” (no-more-copies), and “00” (copyfree)). If an unauthorized copy is detected, the playback device will prevent the playback of the copy and the recording device will refuse to make a next generation copy. Time will tell if the multimedia content will be required to be watermarked for copy protection or access control purposes in multicast applications.

Wireless networks

• Hierarchical node based schemes are the natural choice to develop inter-area rekeying algorithms. The level of trust assigned to the nodes determines the amount of work performed by the entities participating in secure group communications (Bruschi and Rosti 2000).

• The domain that defines a group is made up of a number of disjoint areas, each with its own intra-area rekeying algorithm. The size and definition (logical or geographic) of the areas depend on such factors as the network architecture, the application type (military, commercial, etc.) and operational arrangements (Dondeti et al. 2001).

• The current key distribution protocols assume mobile members and fixed key distributors (KD). If key distribution services are hosted on mobile networking environments, KD mobility will present new challenges. When an AKD moves, for example, its members will have to find a new AKD for coverage. Dynamic allocation of KDs is an active research area (Griffin et al. 2002).

• Inter-area rekeying algorithms are compared in two studies which consistently show, as expected, the performance gain of delayed rekeying. This is achieved by allowing a member to accumulate multiple area keys and to reuse them when he returns to the areas previously visited.

Encryption based technologies may provide sufficient multimedia security for a given application with appropriate key management and authentication methods. It appears that practical use of watermarking lies in the area of copyright protection, particularly because of the cost of implementing a watermarking system for the purpose of access control. In spite of several years of research and testing, the Interim Board of Directors of the DVD Copy Control Association (DVD CCA) decided not to select a watermarking system for copy protection before ending its term in the summer of 2002.1 The task of determining the next steps has been inherited by the new board.

Multicast security is a relatively new research area. With more comparative studies and efficient techniques, we will move toward mature technologies to protect group communications in a variety of applications. Maturity will imply efficient schemes for key management, authentication and traitor detection in wired and wireless networks as well as robust watermarking algorithms with sufficient capacity to carry the information needed for copyright protection or access control.

ACKNOWLEDGMENT

We would like to thank Dr.M.V.Vijaya Saradhi, HOD, Department of Information Technology for providing his utmost support in preparing the paper

REFERENCES

1. Balenson D, McGrew D and Sherman A (1999) Key Management for Large Dynamic Groups: One-Way Function Trees and Amortized Initialization. Internet Draft (work in progress), February 26, 1999.
2. Ballardie A (1996) Scalable Multicast Key Distribution. RFC 1949, May 1996.
3. Banerjee S and Bhattacharjee B (2001) Scalable Secure Group Communication over IP Multicast. International Conference on Network Protocols, Riverside, CA, November 10-14, 2001.
4. Bauer M, Canetti R, Dondeti L and Lindholm F (2002) Group Key Management Architecture, Internet Draft, IETF MSEC WG, February 2002.
5. Becker C and Willie U (1998) Communication Complexity of Group Key Distribution. 5th ACM Conference on Computer and Communications Security, San Fransisco, CA, November 1998.
6. Bell A (1999) The Dynamic Digital Disk. IEEE Spectrum, Volume 36, Number 10, October 1999, pp. 28-35.
7. Bloom JA, Cox IJ, Kalker T, Linnartz JPMG, Miller ML and Traw CBS (1999) Copy Protection for DVD Video. Proceedings of the IEEE, Vol. 87, No. 7, July 1999, pp. 1267-1276.
8. Blundo C, De Santis A, Herzberg A, Kutten S, Vaccaro U and Yung M (1997) Perfectly-Secure Key Distribution for Dynamic Conferences. Information and Computation, December 1997.
9. Boneh D, Durfee G and Franklin M (2001) Lower Bounds for Multicast Message Authentication. Proceedings of Eurocrypt 2001, Lecture Notes in Computer Science, Vol. 2045, Springer-Verlag, 2001, pp. 437-452.
10. Boyd C (1997) On Key Agreement and Conference Key Agreement. Information Security and Privacy, Second Australasian Conference, ACISP'97, Sydney, NSW, Australia, July 7-9, 1997, pp.294-302.
11. Briscoe B (1999) MARKS: Multicast Key Management Using Arbitrarily Revealed Key Sequences. First International Workshop on Networked Group Communication, November 1999.
12. Brown I, Perkins C and Crowcroft J (1999) Watercasting: Distributed Watermarking of Multicast Media. First International Workshop on Networked Group Communication (NGC '99), Pisa, November 17-20, 1999.
13. Bruschi D and Rosti E (2000) Secure Multicast in Wireless Networks of Mobile Hosts and Protocols and Issues. ACM Baltzer MONET Journal, Special Issue on Multipoint Communication in Wireless Networks, 2000.
14. Canetti R, Garay J, Itkis G, Micciancio D, Naor M and Pinkas B (1999) Multicast Security: A Taxonomy and Some Efficient Constructions. Proceedings of IEEE INFOCOM, Vol. 2, New York, March 1999, pp. 708-716.
15. Caronni G, Waldvogel M, Sun D and Plattner B (1998) Efficient Security for Large and Dynamic Groups. Technical Report No. 41, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, February 1998.
16. Caronni G and Schuba C (2001) Enabling Hierarchical and Bulk-Distribution for Watermarked Content. The 17th Annual Computer Security Applications Conference, New Orleans, LA, December 10-14, 2001.
17. Chang I, Engel R, Kandlur D, Pendakaris D and Saha D (1999) Key Management for Secure Internet Multicast Using Boolean Function Minimization Techniques. Proceedings of IEEE INFOCOM, Vol. 2, New York, March 1999.
18. Chiou GH and Chen WT (1989) Secure Broadcast Using the Secure Lock. IEEE Transactions on Software Engineering, 15(8), August 1989, pp. 929-934.
19. Chor B, Fiat A, Naor M and Pinkas B (2000) Tracing Traitors. IEEE Transactions on Information Theory, Vol. 46, No. 3, May 2000.
20. Chu H, Qiao L and Nahrstedt K (1999) A Secure Multicast Protocol with Copyright Protection. Proceedings of IS&T/SPIE Symposium on Electronic Imaging: Science and Technology, January 1999.

AUTHORS PROFILE

B.V.S.S.R.S. Sastry has been graduated with B.Tech in 2009 from Aurora's Engineering College, Bhongir, Andhra Pradesh, India. He is currently pursuing M.Tech from Aurora's Engineering College, Bhongir, Andhra Pradesh, India. Contact him at **sastry_38@yahoo.com**

K.Akshitha has been graduated with B.Tech in 2009 from Royal Institute of Technology and Science, Chevella, R.R.Dist, Andhra Pradesh, India. She is currently pursuing M.Tech from Aurora's Engineering College, Bhongir, Andhra Pradesh, India. Contact her at **Koluguri.87@gmail.com**

Dr. M.V.Vijaya Saradhi is Currently Professor in the Department of Information Technology (IT) at Aurora's Engineering College, Bhongiri, Andhra Pradesh, India. He received the Ph.D degree in Computer Science & Engineering from Osmania University (OU), Hyderabad, Andhra Pradesh, India. He is a life member of various professional bodies like MIETE, MCSI, MIE, and MISTE. You may contact him at **meduri_vsd@yahoo.co.in**