MULTIMEDIA SECURITY IN GROUP COMMUNICATIONS: RECENT PROGRESS IN WIRED AND WIRELESS NETWORKS

Ahmet M. Eskicioglu
Department of Computer and Information Science
Brooklyn College of the City University of New York
2900 Bedford Avenue, Brooklyn, NY 11210
Email: eskicioglu@sci.brooklyn.cuny.edu

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. 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.

For many multicast applications, security is one of the most important requirements. Multicast security includes group membership control, secure key distribution and secure data transfer. 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.

Key Words: multimedia, key management, authentication, multicast security, watermarking.

1. 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. Service types include teleconferencing, pay-per-view, video-on-demand, interactive simulation, software updates and real-time delivery of stock market information. Security is an important requirement for the distribution networks when the delivery includes either confidential or commercial data.

The traditional mechanism to support multicast communications is IP multicast [1]. 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. 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. 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.

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.

2. 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. The four major issues of IP multicast security are [2]:

- *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 cyptographic 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.

2.1 Wired Network Security

2.1.1 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 [3]. 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 [4].
3. *Centralized* schemes, *distributed subgroup* schemes and *distributed* schemes [5].

The above classifications may be combined into one using two criteria: The entity who exercises the control and whether the scheme is scalable or not.

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. 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. 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. Table 1 lists the key management schemes according to the new criteria.

Table 1. Classification of key management schemes

	Scalable	Non-scalable
Centralized Group Control	Wong et al, 97 [6] Caronni et al, 98 [7] Balenson et al, 99 [8] Canetti et al, 99 [9] Chang et al, 99[10] Wallner et al, 99 [11] Waldvogel et al, 99 [12] Banerjee et al, 01 [13] Eskicioglu et al, 02 [14]	Chiou et al, 89 [15] Gong et al, 94 [16] Harney et al, 97 [17] Dunigan et al, 97 [18] Blundo et al, 97 [19] Poovendran et al, 98 [20] Chu et al, 99 [21] Wallner et al, 99 [11] Scheikl et al. 02 [22]
Subgroup Control	Mittra, 97 [23] Dondeti et al, 99 [24] Molva et al, 99 [25] Setia et al, 00 [26] Hardjono et al, 00 [27]	Ballardie, 96 [28] Briscoe, 99 [29]
Member Control	Dondeti et al, 99 [30] Perrig, 99 [31] Waldvogel et al, 99 [12] Rodeh et al, 00 [32] Kim et al, 00 [33]	Boyd, 97 [34] Steiner et al, 97 [35] Becker et al, 98 [36]

Among the schemes listed in Table 1, three scalable schemes, namely, the Centralized Tree-Based Key Management (CTKM) [6], Iolus [23] and DEP [24] are compared through simulation using real-life multicast group membership traces [37]. The performance metrics used in this comparison are (1) the encryption cost at the sender, and (2) encryption/decryption cost at the members and subgroup managers. It is shown that hierarchical node based approaches perform better than hierarchical key based approaches, in general. Furthermore, the performance gain of hierarchical node based approaches increases with the multicast group size.

2.1.2 Periodic Batch Rekeying

In spite of the efficiency of the tree-based scalable schemes for one-to-many applications, rekeying after each join or leave, i.e., individual rekeying, has two major drawbacks: synchronization problem and inefficiency [38].

- *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* [38]. 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. 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.
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.

2.1.3 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 [39]. For a balanced binary tree with n leaves, the distance from the root to any leaf is $\log_2 n$. 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 [39]. The following procedures are used by the server for adding a new member to a group and deleting an existing member from a group.

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.
- *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. 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 [39]. 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.

2.1.4 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 to mutual authentication between the sender and the receiver.

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:

- *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 [40], 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 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 MACs* [9]
- *stream signing* [41]
- *authentication tree-based signature*s [42]
- *hybrid signatures* [43]
- *TESLA* and *BiBa* [44,45,46]

2.1.5 Watermarking

Watermarking (data hiding) [47,48] 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 [49] 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 M-sequence, transform techniques and image-adaptive techniques [50]. 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*. 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.

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, several distinct approaches have been proposed:

- *A different version of video for each group member* [21]: 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 ith watermarked frame in stream j, $j = 0,1$, a different key $K_{i,j}$ is used to encrypt it. The random bit stream determines whether the member will be given K_{io} or K_{i1} 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)* [51]: 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-d_r packets from each transmission group, where d_r 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* [52]: 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 [21,51] 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.

Lastly, the bulk tagging technique described below appears to be a viable approach for controlling access to multicast data.

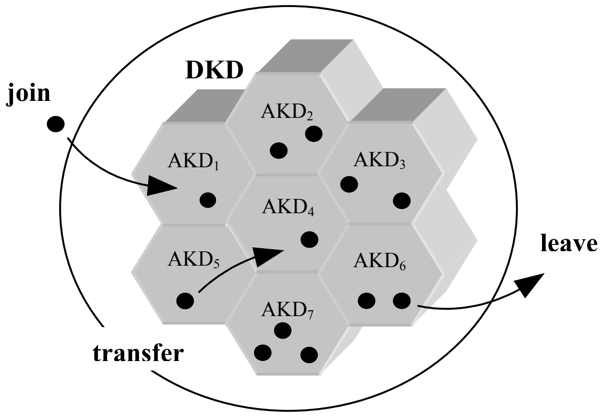
- *Hierarchical tagging and bulk tagging* [53]: Hierarchical tagging allows an artist to insert a different watermark for each of his distributors. Similarly, each distributor can insert a watermark for several sub-distributors. This process can continue until the individual customers receive tagged 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.

2.2 Wireless Network Security

Key management in wireless networks is a more complicated problem because of the mobility of group members [54,55,56]. 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 [3]: *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 1. 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, *inter-area* rekeying algorithms provide the coordination for the transfer of security relationships.



DKD: domain key distributor
AKD: area key distributor

Figure 1. 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 [57].

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_js are in place, the new DEK can be securely multicast among the AKDs and then from each AKD to area members.

Transfer from area i to area j: For member transfer from one area to another, three inter-area rekeying algorithms have been defined:

- *Baseline rekeying (BR)*: The member first leaves the group via area i and then re-joins the group via area j. The data transmission is halted during the distribution of the KEKs and the DEK. In BR, when a member leaves the group, a notification is sent to its current AKD.
- *Immediate rekeying (IR)*: The member initiates a transfer by sending one notification to AKD_i and one notification to AKD_j. Area i performs a KEK_i rekey and area j performs a KEK_j rekey. The only KEK held by a group member is for the area in which it currently resides. Unlike the baseline algorithm, no DEK is generated and data transmission continues uninterrupted. In IR, when a member leaves the group, a notification is sent to its current AKD.
- *Delayed rekeying (DR)*: The member sends one notification to AKD_i and one notification to AKD_j. Area j performs a KEK_j rekey but area i does not perform a KEK_i rekey. AKD_i adds the member to the Extra Key Owner List (EKOL). The EKOL is reset whenever a local rekey occurs. A member accumulates KEKs as it visits different areas. If the entering member has previously visited area j, no KEK_j rekey occurs for j. If the member is entering area j for the first time, a KEK_j rekey occurs for j. To limit the maximum amount of time that KEK_i can be held by a member outside area i, each AKD_i maintains a timer. At $t = T_i$ (a threshold value), the KEK_i is updated and the timer is set to zero. At this point, no group member outside of area i has a valid KEK_i. In DR, when a member leaves the group, a notification is sent to all the AKDs.

Two studies that compare the above algorithms show that delayed rekeying, with reduced communication load and rekeying rate, can improve the performance of key management [56,57]. The first study uses messaging overhead, the KEK rekey rate and the number of KEKs held by a member as the performance metrics. The second study employs rekeying rates, mean number of extra keys and percentage of off-line time to compare the inter-area rekeying algorithms.

3. 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 [24,17,28]. 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:

– Wong et al (CTKM) [6]: 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.

– Dondeti et al (DEP) [24]: Replacement of the KEKs and the group key is a complex and costly procedure, and is expected to be done infrequently.

– Mittra (IOLUS) [23]: 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.

– Eskicioglu et al (CKMSS) [14,58] 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 [39], 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 traced [59]. 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 [60,61,62,63]. 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" (copy-free)). 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 [4].
- 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 [54].
- 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 [55].
- 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.

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.

4. REFERENCES

[1] C. K. Miller, *Multicast Networking and Applications* (Reading, MA, Addison Wesley Longman, Inc., 1999).

[2] T. Hardjono and G. Tsudik, IP Multicast Security: Issues and Directions, *Annales de Telecom*, July-August 2000, 324-334.

[3] L. R. Dondeti, S. Mukherjee and A. Samal, Survey and Comparison of Secure Group Communication Protocols, Technical Report, University of Nebraska-Lincoln, June 1999.

[4] D. Bruschi and E. Rosti, 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.

[5] S. Rafaeli, A Decentralized Architecture for Group Key Management, Computing Department, Lancaster University, England, September 2000.

[6] C. K. Wong, M. G. Gouda and S. S. Lam, Secure Group Communications Using Key Graphs, Department of Computer Sciences, The University of Texas at Austin, Technical Report TR-97-23, July 1997.

[7] G. Caronni, M. Waldvogel, D. Sun and B. Plattner, Efficient Security for Large and Dynamic Groups, Technical Report No. 41, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, February 1998.

[8] D. Balenson, D. McGrew and A. Sherman, Key Management for Large Dynamic Groups: One-Way Function Trees and Amortized Initialization, Internet Draft (work in progress), February 26, 1999.

[9] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor and Benny Pinkas, Multicast Security: A Taxonomy and Some Efficient Constructions, *Proceedings of IEEE INFOCOM*, Vol. 2, New York, March 1999, 708-716.

[10] I. Chang, R. Engel, D. Kandlur, D. Pendakaris and D. Saha, Key Management for Secure Internet Multicast Using Boolean Function Minimization Techniques, *Proceedings of IEEE INFOCOM*, Vol. 2, New York, March 1999.

[11] D. Wallner, E. Harder and R. Agee, Key Management for Multicast: Issues and Architectures, RFC 2627, June 1999.

[12] M. Waldvogel, G. Caronni, D. Sun, N. Weiler and B. Plattner, The VersaKey Framework: Versatile Group Key Management, *JSAC Special Issue on Middleware*, *17*(8), August 1999, 1614-1631.

[13] S. Banerjee and B. Bhattacharjee, Scalable Secure Group Communication over IP Multicast, *International Conference on Network Protocols*, Riverside, CA, November 10-14, 2001.

[14] A. M. Eskicioglu and M. R. Eskicioglu, Multicast Security Using Key Graphs and Secret Sharing, *IEEE International Conference on Networking 2002*, Atlanta, GA, August 26-29, 2002, 228-241.

[15] G. H. Chiou and W. T. Chen, Secure Broadcast Using the Secure Lock, *IEEE Transactions on Software Engineering*, *15*(8), August 1989, 929-934.

[16] L. Gong and N. Shacham, Elements of Trusted Multicasting, *Proceedings of the IEEE International Conference on Network Protocols*, Boston, MA, October 1994, 23-30.

[17] H. Harney and C. Muckenhirn, Group Key Management Protocol (GKMP) Architecture, RFC 2094, July 1997.

[18] T. Dunigan and C. Cao, Group Key Management, Experimental, July 1997.

[19] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung, Perfectly-Secure Key Distribution for Dynamic Conferences, *Information and Computation*, *146*(1), October 10, 1998, 1-23.

[20] R. Poovendran, S. Ahmed, S. Corson and J. Baras, A Scalable Extension of Group Key Management Protocol, Technical Report TR 98-14, Institute for Systems Research, 1998.

[21] H. Chu, L. Qiao and K. Nahrstedt, A Secure Multicast Protocol with Copyright Protection, *Proceedings of IS&T/SPIE Symposium on Electronic Imaging: Science and Technology*, January 1999.

[22] O. Scheikl, J. Lane, R. Boyer and M. Eltoweissy, Multi-level Secure Multicast: The Rethinking of Secure Locks, *Proceedings of the 2002 ICPP Workshops on Trusted Computer Paradigms*, Vancouver, BC, Canada, August 18-21, 2002.

[23] S. Mittra, Iolus: A Framework for Scalable Secure Multicasting, *Proceedings of the ACM SIGCOMM '97*, Cannes, France, September 1997, 277-288.

[24] L. R. Dondeti, S. Mukherjee and A. Samal, A Dual Encryption Protocol for Scalable Secure Multicasting, *Fourth IEEE Symposium on Computers and Communications*, Red Sea, Egypt, July 6-8, 1999.

[25] R. Molva and A. Pannetrat, Scalable Multicast Security in Dynamic Groups, *6th ACM Conference on Computer and Communications Security*, Singapore, November 1999, 101-112.

[26] S. Setia, S. Koussih and S. Jajodia, Kronos: A Scalable Group Re-keying Approach for Secure Multicast, *IEEE Symposium on Security and Privacy 2000*, Oakland, CA, May 14-17, 2000.

[27] T. Hardjono, B. Cain and N. Doraswamy, A Framework for Group Key Management for Multicast Security, Internet Draft (work in progress), August 2000.

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

[29] B. Briscoe, MARKS: Multicast Key Management Using Arbitrarily Revealed Key Sequences, *First International Workshop on Networked Group Communication*, November 1999.

[30] L. R. Dondeti, S. Mukherjee and A. Samal, A Distributed Group Key Management Scheme for Secure Many-to-Many Communication, Technical Report, PINTL-TR-207-99, Department of Computer Science, University of Maryland, 1999.

[31] A. Perrig, Efficient Collaborative Key Management Protocols for Secure Autonomous Group Communication, *International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC '99)*, 1999, 192-202.

[32] O. Rodeh, K. Birman and D. Dolev, Optimized Group Rekey for Group Communication Systems, *Network and Distributed System Security Symposium*, San Diego, CA, February 3-4, 2000.

[33] Y. Kim, A. Perrig and G. Tsudik, Simple and Fault-Tolerant Key Agreement for Dynamic Collaborative Groups, *7th ACM Conference on Computer and Communications Security*, November 2000, 235-241.

[34] C. Boyd, On Key Agreement and Conference Key Agreement, *Information Security and Privacy, Second Australasian Conference*, ACISP'97, Sydney, NSW, Australia, July 7-9, 1997, 294-302.

[35] M. Steiner, G. Tsudik and M. Waidner, Cliques: A New Approach to Group Key Agreement, Technical Report RZ 2984, IBM Research, December 1997.

[36] C. Becker and U. Willie, Communication Complexity of Group Key Distribution, *5th ACM Conference on Computer and Communications Security*, San Fransisco, CA, November 1998.

[37] L. R. Dondeti, S. Mukherjee and A. Samal, Comparison of Hierarchical Key Distribution Schemes, *Proceedings of IEEE Globecom Global Internet Symposium*, Rio de Janerio, Brazil, December 1999.

[38] Y. R. Yang, X. S. Li, X. B. Zhang, S. S. Lam, Reliable Group Rekeying: A Performance Analysis, *Proceedings of the ACM SIGCOMM '01*, San Diego, CA, August 27-31, 2001, 27-38.

[39] M. J. Moyer, J. R. Rao and P. Rohatgi, Maintaining Balanced Key Trees for Secure Multicast, Internet Draft, June 1999.

[40] M. J. Moyer, J. R. Rao and P. Rohatgi, A Survey of Security Issues in Multicast Communications, *IEEE Network*, November/December 1999, 12-23.

[41] R. Gennaro and P. Rohatgi, How to Sign Digital Streams, *Advances in Cryptology – CRYPTO '97*, 1997, 180-197.

[42] C. K. Wong and S. Lam, Digital Signatures for Flows and Multicasts, *Proceedings of IEEE ICNP '98*, Austin, TX, October 1998.

[43] P. Rohatgi, A Compact and Fast Hybrid Signature Scheme for Multicast Packets, *6th ACM Conference on Computer and Communications Security*, Singapore, November 1-4, 1999.

[44] A. Perrig, R. Canetti, J. D. Tygar and D. Song, Efficient Authentication and Signing of Multicast Streams over Lossy Channels, *Proceedings of IEEE Symposium on Security and Privacy 2000*, Oakland, CA, May 14-17, 2000.

[45] A. Perrig, R. Canetti, D. Song and J. D. Tygar, Efficient and Secure Source Authentication for Multicast, *Proceedings of Network and Distributed System Security Symposium*, San Diego, CA, February 8-9, 2001.

[46] A. Perrig, The BiBa One-Time Signature and Broadcast Authentication Protocol, *8th ACM Conference on Computer and Communications Security*, Philadelphia, PA, November 5-8, 2001.

[47] M. D. Swanson, M. Kobayashi and A. H. Tewfik, Multimedia Data Embedding and Watermarking Techniques, *Proceedings of the IEEE*, *86*(6), June 1998, 1064-1088.

[48] F. A. P. Petitcolas, R. J. Anderson, M. G. Kuhn, Information hiding – a survey, *Proceedings of the IEEE*, *87*(7), July 1999, 1062-1078.

[49] F. Hartung and M. Kutter, Multimedia watermarking techniques, *Proceedings of the IEEE*, *87*(7), July 1999, 1079-1107.

[50] R. B. Wolfgang, C. I. Podilchuk and E. J. Delp, Perceptual Watermarks for Digital Images and Video, *Proceedings of the IEEE*, *87*(7), July 1999, 1108-1126.

[51] I. Brown, C. Perkins and J. Crowcroft, Watercasting: Distributed Watermarking of Multicast Media, *First International Workshop on Networked Group Communication (NGC '99)*, Pisa, November 17-20, 1999.

[52] P. Judge and M. Ammar, WHIM: Watermarking Multicast Video with a Hierarchy of Intermediaries, *The 10th International Workshop on Network and Operation System Support for Digital Audio and Video*, Chapel Hill, NC, June 26-28, 2000.

[53] G. Caronni and C. Schuba, Enabling Hierarchical and Bulk-Distribution for Watermarked Content, *The 17th Annual Computer Security Applications Conference*, New Orleans, LA, December 10-14, 2001.

[54] L. R. Dondeti, B. T. Decleene, S. P. Griffin, T. Hardjono, J. Kurose, D. Townsley, C. Zhang and S. Vasudevan, Group Key Management in Wireless and Mobile Environments, Internet Draft (Work in progress), July 2001.

[55] S. P. Griffin, B. T. DeCleene, L. R. Dondeti, R. M. Flynn, D. Kiwior and A. Olbert, Hierarchical Key Management for Mobile Multicast Members, Technical Report, Northrop Grumman Information Technology, 2002.

[56] B. T. DeCleene, L. R. Dondeti, S. P. Griffin, T. Hardjono, D. Kiwior, J. Kurose, D. Towsley, S. Vasudevan and C. Zhang, Secure Group Communications for Wireless Networks, *Proceedings of IEEE MILCOM 2001*, Mclean, VA, October 2001.

[57] C. Zhang, B. T. DeCleene, J. Kurose and D. Towsley, Comparison of Inter-Area Rekeying Algorithms for Secure Wireless Group Communications, Technical Report, University of Massachusetts, June 2001.

[58] A. M. Eskicioglu and E. J. Delp, An Integrated Approach to Encrypting Scalable Video, *IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, August 26-29, 2002.

[59] B. Chor, A. Fiat, M. Naor and B. Pinkas, Tracing Traitors, *IEEE Transactions on Information Theory*, *46*(3), May 2000.

[60] A. Bell, The Dynamic Digital Disk, *IEEE Spectrum*, *36*(10), October 1999, 28-35.

[61] J. A. Bloom, I. J. Cox, T. Kalker, J-P. M. G. Linnartz, M. L. Miller and C. B. S. Traw, Copy Protection for DVD Video, *Proceedings of the IEEE*, *87*(7), July 1999, 1267-1276.

[62] A. M. Eskicioglu and E. J. Delp, Overview of Multimedia Content Protection in Consumer Electronics Devices, *Signal Processing: Image Communication*, *16*(5), April 2001, 681-699.

[63] A. M. Eskicioglu, J. Town and E. J. Delp, Security of Digital Entertainment Content from Creation to Consumption, (Invited paper) *Proceedings of SPIE Applications of Digital Image Processing XXIV*, Vol. 4472, San Diego, CA, July 31-August 3, 2001, 187-211.