IP Multicast Security: Issues and Directions

Thomas Hardjono
Bay Architecture Laboratory, Nortel Networks
3 Federal Street, BL3-03
Billerica, MA 01821
email: thardjono@baynetworks.com

Gene Tsudik
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425, USA
email: gts@ics.uci.edu

Abstract

Security represents one of the major current obstacles to the wider deployment of IP multicast. The present work identifies and discusses various concepts and issues underlying multicast security. A classification of the current issues is provided, covering some *core* problems, *infrastructure* problems, and certain *complex applications* that might be built atop secure IP multicast. Three broad core problem areas are defined, namely fast and efficient source authentication for high data-rate applications, secure and scalable group key management techniques and the need for methods to express and implement policies specific to multicast security. The infrastructure problem areas cover the issues related to the security of multicast routing protocols and reliable multicast protocols. The topic of complex application covers more advanced issues, typically relating to secure group communication at (or above) the session layer which may be built using an eventual secure multicast infrastructure. A brief summary of the relevant developments, including those in the IETF, is provided.

1 Introduction

The phenomenal growth of the Internet in the last few years has provided both the inspiration and motivation for the development of new services, combining voice, video and text "over IP". Although unicast communication have been predominant so far, the demand for multicast communication is increasing, both from the internet service providers and from the content or media providers and distributors. One notable obstacle to wider commercial deployment of IP multicast has been the lack of security for the content (data) being transmitted through IP multicast and for the communication infrastructure underlying IP multicast as a service.

This paper presents a bird's eye view of the current issues surrounding IP multicast security and discusses the current efforts to solve them. The paper also presents open problems and suggests directions for the possible solutions to these problems. We first place IP multicast in the context of security and discuss why security must be built atop the basic IP multicast model. Factors that specifically affect efforts to secure IP multicast are then briefly identified and explained. In order to provide structure to the discussions in this rather broad field, a classification of the problems in multicast security is given. The classification identifies three problem areas: those at the core of the multicast security question, those pertaining to the infrastructures that enable IP multicast and finally those pertaining to the more complex group-oriented applications that may be built upon a secure IP multicast service.

Although the paper does not aim at providing a comprehensive survey of the area of multicast security and the broader group-oriented security, it does provide sufficient references for the reader to follow-up. Some of the more relevant works are described in more detail, either for the purpose of giving an example or because they represent important milestones in the field.

2 The IP Multicast Model and Security

IP multicast allows the delivery of messages to multiple receivers in a convenient and reasonably efficient manner. The IP multicast model was first proposed in the seminal work by Deering [1]. It uses the notion of a *group* of members associated with a given *group address* (a Class-D IP address). A sender simply sends a message to this group address and the network replicates the message at suitable junctions (i.e., routers) and forwards the copies to group members located throughout the network. In order to achieve this convenient replication, routers must maintain some state information about all relevant multicast groups. The maintenance of this state is achieved using a *multicast routing protocol* which creates a logical *distribution tree*. Examples of multicast routing protocols are: DVMRP [2], CBT [3], MOSPF [4] and PIM (Dense and Sparse Modes) [5]. In essence, the distribution tree operates with routers maintaining state information about the interface a multicast packet arrived on and the interface(s) on which it has to be sent out.

In the IP multicast model, a host "joins" a group by using the Internet Group Management Protocol (IGMP) [1, 6] which runs between a host and the subnet router. IGMP (currently version 2) allows the host to indicate to the subnet router that it wishes to receive packets destined for a given multicast group address. Here, the subnet router is interested only in whether or not there is a local receiver for a given multicast group. It does not maintain information about the **membership** of just any group. Thus, the subnet router is only concerned about the "active" groups in its subnet. IGMP version 3 [7] allows a host-receiver to further specify the sources (senders) of a group to which it wishes to receive data. From the perspective of security, the multicast model allows the receivers to remain anonymous, since the subnet routers do not propagate any identification information to the other members of the group.

Figure 1 shows a simplified depiction of the basic IP multicast model. IGMP is running in the subnet routers (R) with attached hosts (H), and only one group is in existence. The members of the group are depicted as circles with thicker lines. The dashed lines indicates the path of traversal of the packets of the multicast group, from the sender to the receivers (hosts) in the various subnets.

The IP multicast model is attractive because it can scale to a large number of members, subject only to the resources available to the underlying multicast routing protocol However, scalability is achievable precisely because of the "anonymous-receiver" behavior inherent in the model, as expressed through IGMP. In other words, scalability is attained because no host identification information is maintained by the routers. Any host in a subnet can join a multicast group without

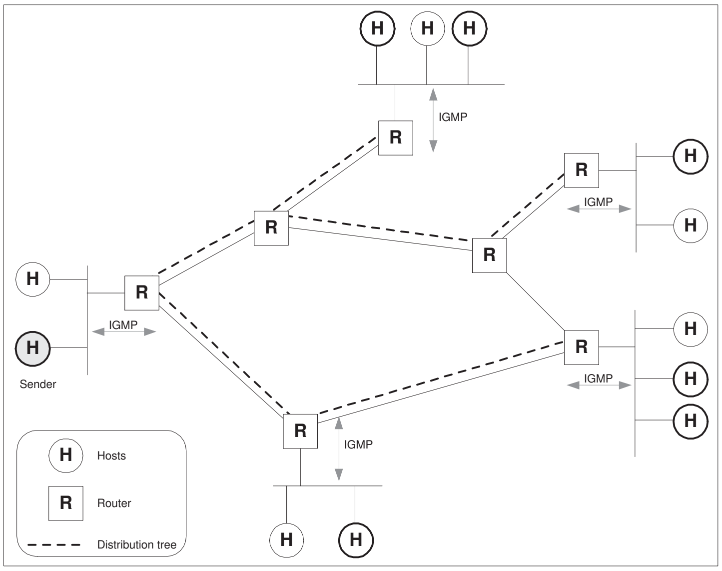

Figure 1: Basic IP Multicast Model

its subnet router passing identification information about the host to other routers upstream in the distribution tree.

Although, from the perspective of security, this lack of host identification information may be viewed at first as being a deficiency, it must be noted that the IP multicast model was never intended to provide secure multicasting. Rather, the IP multicast model allows additional mechanisms and services to be built atop of it. As it turns out, the decoupling of security from the multicast model is advantageous since it allows various security models and architectures to be deployed without affecting the multicast distribution tree. It is also important from the perspective of applications since each application requires different types of host information and other security parameters.

In the next section we will look at some of the factors that influence the security approaches and mechanisms deployable over the basic IP multicast model.

3 Factors in Securing IP Multicast

There are several factors or aspects of IP multicast that influence the approaches and mechanisms used to secure it. Of these, the most relevant factors influencing IP multicast security include the multicast application type, group dynamics, scalability issues and the trust model underlying the multicast security approach:

Multicast Application Type: IP multicast commonly views multicast groups as being either One-to-Many or Many-to-Many. This also corresponds to the type of communication occurring among the group members. Together with the value of the data being transmitted, there exists a spectrum of applications of IP multicast that need to be secured.

As an example, at one end of the spectrum a subscription service may take the form of a One-to-Many multicast from a single source to multiple receivers. Here, the data being delivered may be publicly available (eg. stock market information). Thus, for this subscriber application source authentication of the data is more important than confidentiality.

A second example is the Pay Per View (PPV) service where a group of receivers pay a subscription fee for the program being delivered, analogous to the Pay TV scheme. Although the data itself is not confidential, it carries some value in that the content provider would like to limit access to only the paying subscribers. In this example, encryption of the data may be used to achieve access control, while source authentication may not be as important.

PPV-type applications are also distinguished by the relative obliviousness to the group dynamics. For example, if a Pay-TV subscriber drops out ("turns off the set") the security parameters of the current program broadcast are typically not influenced. Similarly, a new subscriber usually does not precipitate changes in the on-going secure broadcast.

At the other end of the spectrum are the cases which require both confidentiality and source authentication. An example would be a conference call that is implemented over a Many-to-Many multicast. Here each party in the conference would like to know and be assured of the identity of the source of all transmissions in the conference. Since conferencing events are typically limited in membership and are confidential in nature, encryption must be used to achieved the required confidentiality, while methods for source authentication (such as digital signatures) must also be employed.

Another aspect related to the multicast application type is the frequency and rate of data transmission. This aspect is closely related to the performance of the cryptographic algorithms. Thus, for example, continuous streaming video may be afforded a different level of security from the infrequent multicast-based delivery of software update packages, due to the intensive computational requirements of cryptographic operations on streaming video.

Group Size and Group Dynamics: Another important factor affecting multicast security is the size and stability of the group. A multicast group may range in size from a few to tens of thousands of members. The differing sizes affect the mechanisms used to effect security and the scalability of such mechanisms. Security is also influenced by factors such as the frequency of members joining and/or leaving and the average size of the membership change. This, in turn, relates to the application type and, at times, the type of the underlying network. (For example, network perturbations such as those caused by unstable routers can result in frequent group partitions unrelated to explicit membership changes.)

Thus, for example, a multicast group for a PPV service for 100 users may have a different requirements and demands compared to that for 10,000 users. Furthermore, the population distribution of the users and the density of users in certain parts of the internet may affect both the multicast routing protocol being deployed and the security mechanisms used for the multicast group.

Scalability Issues: Scalability in the context of multicast security refers particularly to the ability of the mechanisms implementing the security features to be extended to cover a larger group of members over a wide physical region without too much deterioration in the level of service and performance of the system as a whole. In general, scalability affects almost all facets of networking. However, in the context of security for multicast, scalability pertains more specifically to the delivery and management of the cryptographic keys, and the propagation and management of security-related policies.

Trust Model: When cryptography is employed to provide protection for data, the issue of trust comes to the foreground. The problem concerns the entities that generate, distribute and manage the cryptographic keys and security policies. At the heart of the problem is the need for a *model of trust* underlying the multicast security scheme, that addresses the issues of which entities to be accorded trust to carry out these functions, the level of trust accorded to them, the source of authority, and other related issues.

Although these are the factors which are immediately apparent upon first looking into the problem of security in IP multicast, the list is not exhaustive. Furthermore, some of the factors are inherently multi-dimensional, since they also involve the other abovementioned factors and other less apparent factors and variables. In the next section we begin to address the security issues in IP multicast by identifying some groups of problems which underpin the security solutions for IP multicast and which materialize the factors mentioned above.

4 General Problem Areas in Multicast Security

In order to understand better the various problems surrounding IP multicast security, we divide these problems into three categories, which we refer to as the *core* problem area, the *infrastructure* problem area, and the *complex applications* problem area.

Core Problem Area: The core problem area includes issues of pressing concern, where the solutions are needed in order to solve the broader infrastructure problems [8]:

- Methods for multicast data confidentiality/integrity and source authentication. This is discussed in Section 5.
- Multicast group key management (ie. methods for initializing and adjusting group keying material). Section 6 covers issues in this category.
- Multicast security policies, governing everything from group admission to group key change clauses and type of encryption used. (Section 7).

Infrastructure Problem Area: The infrastructure problem area concerns the issues which are broader in nature and may deploy the solutions defined in the core problem areas. The two foremost infrastructure problem areas are:

- Security of multicast routing protocols (Section 8).
- Security of Reliable Multicast (RM) protocols (Section 9).

Complex Applications Problem Area: The complex applications problem area covers the more advanced issues that might be build upon an eventual secure multicast infrastructure. Among these are:

- Distributed group key generation (key agreement)
- Group and member certification
- Secure communication with the "outside"
- Various authentication flavors (Quality-of-Authentication) of both group as a whole and individual members
- Member non-repudiation
- Member anonymity
- Robustness against attacks and recovery methods

Section 10 discusses some of these issues and possible directions for future work.

These problem areas will be used as the organization of the remainder of this paper and will be discussed in the ensuing sections.

5 Methods for Multicast Data Confidentiality and Authentication

The first core problem area in multicast security concerns the methods used to ascertain the authenticity of a piece of data and the methods used to establish data confidentiality (secrecy), specifically in the context of voluminous data such as within streaming video applications.

Since multicast traffic, like unicast traffic, traverses the so-called "public" Internet, parties that wish to deliver value-carrying data using IP multicast must deploy mechanisms to control access to the data. One method commonly used to implement controlled access is data encryption. The notion here is that data would be cryptographically enciphered at the source (sender) and the decipherment keys would be available to the intended recipients of the multicast data (namely the multicast group members). Thus, although the IP multicast traffic (like other traffic) over the Internet can be intercepted by any party, that data would be useless without the decryption key. The same notion also applies for authentication, where only the holders of the authentication key can authenticate the data sent to the multicast group.

In the context of multicast security it is useful to treat the issue of data confidentiality as separate from data authentication. This is due to the fact that different applications have different requirements and thus apply only one of the two. Thus, for example, the publicly available stock market data being delivered through a multicast group requires source authentication more than it needs confidentiality. On the other hand, a subscriber-based applications (eg. pay-per-view) requires both source authentication and confidentiality.

In the specific context of authentication, it is further useful to distinguish between source authentication and group authentication. The first is typically achieved using public key cryptography, while the later using symmetric key cryptography.

The relevance of source authentication and group authentication becomes apparent when the performance of public key (asymmetric) algorithms and shared key (symmetric) algorithms are taken into consideration, particularly in the context of the high rate of transmission of certain multicast applications. Typically, in software implementations, public key algorithms are several magnitudes slower than shared key algorithms. Thus, the choice between source authentication and group authentication must be weighed against the application type, the computing resources available to the group members and the value of the data being delivered through IP multicast.

Methods that can be used to address the need of authentication and confidentiality of high data rate packets include:

- Application of cryptography to only certain packets at certain intervals.
- Link chaining the authentication of successive packets.
- Threshold authentication, whereby any k out of t packets ($k \leq t$) is sufficient to establish the authenticity of the set of t packets.

- Off-line digital signatures, where both the sender and receiver can perform signing/verification rapidly and efficiently.

Since not all end-users can be expected to have cryptographic hardware or similar assistance for the specific needs of IP multicast, the issue of fast source authentication techniques and corresponding algorithms that are deployable in software remains an open problem. Intermediate or hybrid solutions, such as the digital signing of hashes of several data packets and other stream encryption methods, will remain attractive until such algorithms become standard and are adopted on a wider scale.

6 Multicast Group Key Management

As mentioned previously, since data communicated within a multicast group traverses the public Internet and is therefore subject to tapping or copying by non-members of the group, encryption is the method commonly used to provided access control to the data. In the simplest case, shared-key (symmetric) cryptography is used by the sender/source and the receivers, where the data is encrypted by the sender and decrypted by the receivers. This shared key is commonly referred to as the *group key*, since only members of the multicast group are in possession of the key.

The use of cryptography necessitates the distribution or dissemination of keys, which in this case is the group key. Thus, an additional facet to the general problem of multicast security is the method of distributing keys to the appropriate entities involved in a multicast instance and the management of the keys of over given period of time. A *Group Key Management* (GKM) protocol must not only issue a group key for a new multicast group, but also update (re-key) the existing group key under certain conditions and following the prescribed policies, be those general security policies or multicast-specific policies.

6.1 GKM Requirements

There are a number of requirement that a group key management protocol must satisfy [9, 10].

Scalability: Group key management must be scalable to the scope of: the population being catered for in the multicast group, its varying population densities and behaviors, and its wide geographic distribution. The notion of scalability in this context means that group key management operations should be efficient in resource usage, easily accessible and should minimize delays and other restrictions on the communication within the multicast group.

Independence: Group key management must be independent from both unicast and multicast routing [10]. Protocols that implement group key management must be usable over the various available (and future) routing protocols which might run in different parts of the Internet.

Reliability: The delivery of a group key must be a reliable event, meaning that there should be no doubt as to the status of the delivered key to a recipient (group member). Members of a group must be able to rely on the group key management protocol(s) to deliver the group key to them in a timely fashion.

Security: Group key management must be carried out in a secure fashion, with relevant keys being delivered through a secure channel established to the group members. Such methods must be resistant against a wide range of attacks by both non-member and (even) member attackers. Other supporting keys, or key management keys (km-keys) may be deployed to create a safe passage for the traffic encryption keys used for the bulk multicast data.

6.2 Key Updates

The updates (re-keys) of the group key used within a multicast group are affected by the policies governing the multicast transmission, the periodic key refresh duration, population distribution and membership dynamics, among other factors.

For instance, a multicast group may be governed by the *forward-secrecy* membership policy and the *backward-secrecy* membership policy (or both). These, in turn, govern how key updates are carried out. The forward-secrecy policy may specify that, whenever a member of a group leaves the group, it must be prevented from having further access to the data and group keys of that multicast group. The backward secrecy policy may specify that data communicated within the group before a member joins must remain secret to the new member (even if the encrypted form of that data is public).

Both of the above policies can be enforced by performing a mandatory re-key of the group key upon any change in the group membership.

In general, changes to the group membership can result from new members joining, existing members voluntarily leaving or existing members being revoked (ejected) from the group.

Several factors may influence this approach to re-keying. These include the costs in terms of the computation cycles and the number of exchanged messages, the frequency of membership changes, the population sizes, the existence (or non-existence) of default periodic refreshes, the value of the data and others. For schemes implementing periodic group key refreshes to protect against hostile cryptanalysis, benefits can be gained by aligning re-keying events due to membership changes to those due to periodic re-keying.

6.3 Scalability, Domains and Key Management Keys

Scalability represents an important concern in IP multicast routing protocols, and often separate routing domains are delineated in order to ease network management.

The concept of domains is also applicable to group key management to effect scalability, where members are divided (logically or physically) into domains or subgroups. At least two general types of domains are possible for group key management:

- *Domains according to data encryption*: Here, the domains demarcate regions wherein different group keys are used to encrypt the multicast data. Thus, each domain is associated with a unique group key, and "crypto translations" (decryption using one key, followed by encryption using another key) must be carried out at the domain boundaries. Group members residing within each domain would be in possession of a unique group key (per domain). The work of [11] illustrates this approach. In effect, each domain can be treated independently since it would be associated with a different key.

- *Domains according to key management*: Here, the domains demarcate key management regions, where each region is associated with a different set of key management keys (*km-keys*) for the express purpose of disseminating the common group key. Thus, each domain would manage its own km-keys (eg. institute its own re-key period for km-keys), even though these are used to create safe passage for the common (group-wide) group key from a single key source (eg. key server) to each of the receivers residing in different key management domains.

Combinations of both types of domains are possible; furthermore, other interpretations of domains can be used.

6.4 Architectures for Group Key Management

The work of [12] summarizes three useful architectures for group key management, viewed from the perspective of the logical key arrangements and key relationships. In order to explain these architectures, a useful basic model is shown in Figure 2. The model consists of a Core (or Root) *Key Management Entity* (KME) serving either the group members directly (centralized key management), or serving other Key Management Entities (distributed key management). Although perhaps more complex, the distributed key management approach lends itself more easily to scalability since KME-to-KME secure communication can be established to deliver keys to furthest KMEs. The three architectures are discussed below.

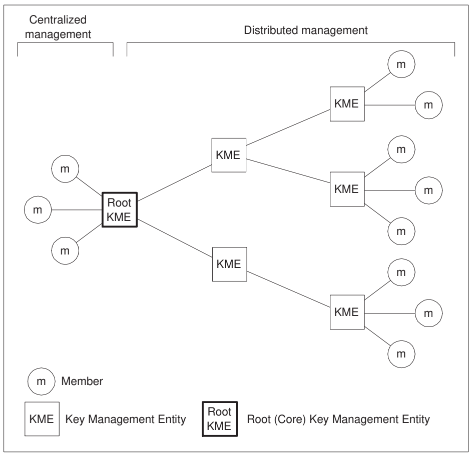

Figure 2: Basic Model for GKM

6.4.1 Pairwise Key Arrangement

In this approach the Root KME shares in a pairwise manner a unique key with each (valid) member of the multicast group. The pairwise secure channel created between the Root KME and each member is then used to deliver a group key. The Root KME carries out this exchange for each member of the group. Although this approach allows the Root KME to be the single point of trust for each member of the group, the approach is cumbersome and may not scale to large numbers of members.

A possible variation of this approach would be to delegate the exchange process to a number of selected “Subroot KMEs”, thereby pushing the computational task to selected group members. This variation, however, requires these selected group members to be trusted. In addition, there is the increased complexity in the task of removing members who are subroots.

Note that the need for a Root KME to share a pairwise unique key with each group member is crucial in any case, since such a key is the basis for the creation of a secure channel between the two, which in turn represents the point-of-departure for other more complex solutions, all of which require some form of “bootstrapping” to start the group key management protocol.

6.4.2 Complementary Keys Approach

The basic idea here is to deliver a set of “complementary variables”, in addition to the group key, to the members of the multicast group. Each member is associated with a variable by the Root KME. However, a member’s variable is never actually given to it. Instead, a member receives the variable of all the other members (except its own variable). This allows the exclusion of any member in the key generation process.

When a member leaves the group and a new group key has to be recomputed, the Root KME will instruct the remaining members to compute the new group key based on all the variables except the variable of the leaving member. Assuming all the remaining members obey this instruction, the effect is that the leaving member is excluded from the key generation process for the new group key. The leaving member will not be able to compute the new group key since it never has possession of the variable associated to it.

This approach is attractive and is reminiscent of secret sharing schemes [13, 14]. However, for correct execution, it assumes that collusion will not occur among members of the group. Furthermore, the cryptographic schemes underlying any such complementary variable approach must be resistant to various attacks to prevent non-members and ex-members from deriving the current group key using other means.

6.4.3 Hierarchical Tree Approach

There are a number of arrangement of keys that are possible for groups of participants in a multicast instance. Two aspects related to key arrangements are the *relationships* among the keys and the *physical distribution* of the keys. As an example, one key may be derived from another through the application of a hash function or through other complex mathematical relationships. In terms of physical distribution, these two keys may reside on a single centralized server or be dispersed on two physically separate servers.

One of the desirable features of group key management protocols is the localization (as much as possible) of the effects of a re-keying event. In other words, a re-keying of one (or a few) members of the group should not affect the other group members too much. To this end the logical division of group members into *subgroups*, arranged in the form of a logical tree, represents a promising avenue towards scalable solutions. Figure 3 shows a logical tree consisting of a number of logical

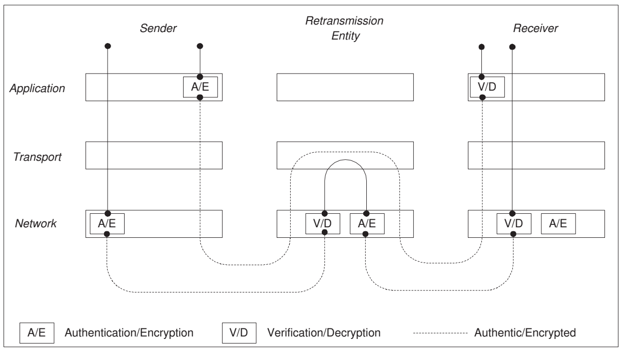

Figure 3: Hierarchical Tree Architecture

nodes. Note that the number of nodes do not necessarily correspond to the number of physical KMEs. Thus, in effect, a physical KME may be responsible for more than key. For example, in Figure 3, the first level of logical node (key) may be implemented in the Root KME, while the second and third level of logical nodes (keys) be implemented by the two second level KMEs.

The aim of the hierarchical tree approach is for each (logical) subgroup to be assigned a unique *subgroup key* for the purpose of delivering and updating the global group key. These subgroup keys, which are known also to the Root KME, allow the Root KME to address subsets of the groups, by enciphering the messages for a given subgroup using the corresponding subgroup key. The resulting ciphertext can then be sent via unicast to individual subgroup members, via the multicast group proper (to the entire group), or via a separate subgroup multicast. In any case, only the holders of the corresponding subgroup key will be able to decipher the ciphertext.

When implemented in a centralized fashion using a single Root KME, the hierarchical tree approach inherently presents more scalability than the other two approaches since subgroups can be tailored to be of varying sizes following to the population density and membership dynamics. However, for multicast groups with a sparse populations spread across wide geographic expanses and for domain autonomy requirements, a distributed KME solution may be preferable to implement (logical) hierarchical tree.

6.5 IETF-Related Activities

The works of [9, 15, 10] have each surveyed to different degrees the various approaches, protocols and solutions proposed for IP multicast security. In the following, we briefly refer to the current efforts being conducted in this area, focusing primarily on the practical rather on the theoretical works. The summary is not meant to be comprehensive and the list of cited works not exhaustive. Hence, the reader is encouraged to follow the references in order to obtain more details on each proposal or solution.

The efforts being conducted in the IETF represents a more engineering view on the problem of multicast, with an eye more on short-term implementable solutions to answer the pressing needs of the industry, both the networking industry and the content industry (eg. media, entertainment), the first seeking to provide services while the later seeking to make use of those services to deliver content to the end-user. Hence, in this sense the standardization efforts in the IETF can be seen as providing the initial foundations at the network and transport layers for more complex applications.

The two general thrusts of efforts in this arena have been focused key relationships and on key dissemination protocols. The first type of works have in mind methods to create mathematical relationships (or hierarchies) among keys that are advantageous – compared to "flat" key relationships – in terms of rekeying. The second type has focused on providing practical protocols to disseminate keys to the members of a multicast group. Both seek to find solutions that are scalable according to the needs of the multicast applications, membership sizes and group dynamics.

- *Key dissemination approaches:*.
  One of the earliest efforts is the *Group Key Management Protocol* (GKMP) of [16, 17] which focuses on the key dissemination. Here each multicast group is assigned a dedicated group controller that shares a symmetric key with each group members. An interesting extension to the notions presented in this work is the possibility of having two tiers of key disseminators, where the second tier is in fact a selected number of group members that are entrusted and have the capability to disseminate keys to the other members.

  The notion of a network entity being ready on-call for services, in this case key dissemination services, is also investigated in the *Multicast Key Management Protocol* (MKMP) [18]. Here, a number of "key distributors" are assumed to be available throughout the network. Each of the key distributors are assumed to be trusted and have the capability to communicate securely with a main key distributor. When a candidate group member is seeking to obtain the group key (namely the key used to decrypt the multicast data) it must "probe" the network for the closest available key distributor. Although the idea of network probing have also been investigated in other resource discovery protocols, in the context of multicast security the issue of trust in such a loosely knit collection of key distributors needs to be further investigated.

  Another early effort is the *Scalable Multicast Key Distribution* (SKMD) approach of [19] in the context of the Core Base Trees (CBT) multicast routing protocol [3, 20]. The approach makes good use of the routing entities involved in CBT, where the core in the CBT protocol is given the additional task of key management. More specifically, since CBT employs what it calls "hard state" in the tree configuration, all routers thus know their on-tree neighbors. Since join requests are forwarded to the core, it becomes the most suitable entity to carry out authorizations, after which the task of authorization and key dissemination can be delegated to the leaf routers at the edge of the tree. Although the approach makes the best use of the features of the CBT protocol, the solution is integrated into the CBT multicast routing protocol itself. Thus, one pre-condition is that a routing domain employs CBT. Should a multicast group span multiple routing domains, each with a differing multicast routing protocol, additional security mechanisms will be required in order to allow for the secure "translation" of data from one domain to another at the borders.

  The mechanism of translations – i.e., data encrypted with one key being decrypted, followed by re-encryption with another key – is also at the heart of the solution proposed for *Iolus* [11]. Here, the idea is that the population of receivers is divided into a hierarchy of subgroups, where subgrouping is done according to the cryptographic keys. Each subgroup has a unique cryptographic key which is used for the multicast data delivered to that subgroup of members. The management of the subgroup keys is by an intermediary-controller that operates under a main controller. The benefit of the approach is that subgroups can be added to the hierarchy, thereby promoting scalability. This would require, however, that the intermediary-controller for a new subgroup be dynamically available, and that the intermediary-controller of a closing subgroup (with no members remaining) be dismissable in a secure manner since it holds sensitive cryptographic keys. In addition, translations must be carried out at the boundary of the subgroups.

Subgrouping is also the basis for the group key management framework of [10]. Reminiscent of intra-domain and inter-domain routing, this framework proposed a two-level hierarchy consisting of one "trunk" region and one or more "leaf" regions. Division into regions is physical, and hence the framework addresses key dissemination. Group members reside within a leaf region, while the trunk region contains only the key managers or key distributors (KD). Each leaf region is catered for by at least one key distributor, and may deploy a different leaf-scoped key management protocol for the members in that leaf region. Since, by definition, the trunk region does not have any group members (receivers), the KD-to-KD key management can be done using a different key management protocol. Furthermore, as it is only a key management framework, [10] does not specify the actual key management protocols to be employed at leaf regions. This is consistent with the view of the Internet as a collection of many autonomous domains with heterogeneous key management requirements (which invariably translates into many potential protocols.)

Following the framework of [10], the *Intra-domain Group Key Management Protocol* (IntraGKMP) [21] proposes a scalable group key management protocol targeted at leaf regions. To enable scalability, each leaf region (i.e., domain) is further divided into "areas", with every host belonging to a single area. There is a single Domain-Key-Distributor (DKD) and many Area-Key-Distributors (AKD) which are responsible for each area. A host only communicates with the AKD of its area. A key for the members of a multicast group in a domain is generated by the DKD and is propagated to the hosts through the AKDs. This scheme presents an interesting new concept: the group key is common to members in the the entire domain, while the control messages for key updates are transferred via the Area-Key-Distributors, using two levels of keys. This method enjoys "the best of two worlds" [9]. First, the data packets need not be re-encrypted in route and can be routed using any multicast routing protocol. Second, the group (or domain) controller need not keep track of all group members; instead, it can keep track only of the AKDs. This facilitates scalability while maintaining independence from the underlying multicast routing. Subgroup multicast into areas by an AKD is achieved using administratively scoped multicast [22, 23].

Other efforts focusing on key dissemination includes the centralized architecture and implementation reported in [24], and a recent group security association agreement protocol (related to key management) in [25].

- *Key relationships:*
  Approaches to provide scalable solutions for key management have also been taken from the perspective of "building-in" features into keys, such that managing the keys used for a multicast group – be that the group key and/or the key management keys – can be done more easily and effectively.

  The work of [12] represents an initial effort to develop "key hierarchies" for the purpose of key management. The initial hierarchical tree approach of [12] presents group members arranged in pairs at the leafs of the tree. The aim of such a hierarchy is to contain the effects of a membership change as far as possible to only the subtree/branch where the member A membership change effects a modification in the keys (logically arranged) between that group member and the root of the tree.

  This notion is extended to a more general arrangement in [26], where the leafs of the logical tree can consist of more than two members. This is the basis for the key dissemination protocol of [21]. However, unlike the physically centralized architecture of [26], in [21] a distributed architecture is used for key dissemination.

Other efforts along these lines have also been reported in [15] in the form of one-way function trees (OFT). This approach is aimed at groups with a large number of members in which computational savings becomes a factor of consideration. Efforts to gain computational efficiency has also been researched in [27] in the form the batching of group membership changes, with the aim of reducing the costs related to the re-keying of the group. Similar efforts along the lines of improving the work of [12] have also been reported in [28].

In general, it is clear that scalable solutions for group key management for IP multicast will be a combination of these two ingredients, namely a logical key arrangement superimposed over a physical key dissemination architecture. Hence, efforts on these two fronts represent an important interest for multicast security.

7 Multicast Security Policies

Similar to other aspects of networking, the correct definition, implementation and maintenance of policies governing the various mechanisms of multicast security is a crucial factor. Those which are directly related to multicast security include the policies for key dissemination, for access control, for the re-keying of group shared keys, and for the actions taken when certain keys are compromised [29, 30].

Other policies may be in place to support the mechanisms used to secure the multicast group. Thus, for example, if a member of a group creates an initial secure channel between itself and a key manager (or key server) using IPsec technology (eg. IKE [31]), then policies governing the pairwise IPsec Security Association [32] and governing the aspects of the key generation must also be in place.

Although there are many possible interpretations of policies in the context of IP multicasting (including those pertaining to multicast traffic shaping), in the context of multicast security two general categories of policies exist:

- *Policies governing group membership of users/hosts.* This covers the aspects of who can join a group, how they are verified/authenticated, under what conditions a user/host can join the group, how a member would be ejected (ie. membership revoked), the minimal resources (eg. cryptographic capabilities) required to join a group, and others.
- *Policies relating to security enforcement.* Some examples of more specific policies include, among others:
  - Policies for initial key dissemination.
  - Policies for subsequent key changes, both related or unrelated to membership changes.
  - Policies to handle errors arising from mechanisms that are security-specific (eg. key server) and from those that are non-security related (eg. routers) which may impact the security of the multicast
  - Policies to address compromises of keys and other sensitive security information.

The possible existence and possible interpretations of policies at different levels demands that the designer of any system to secure multicast develop set of policies which are coherent, free from loopholes or conflicts and which address the possible scenarios to be met by the system.

8 Security of Multicast Routing Protocols

Although not directly affecting the security of the contents (data) and key management in IP multicast, the protection of the multicast routing infrastructure itself is important for IP multicast as a whole. This is due to the fact that the multicast distribution tree is the packet delivery mechanism that carries the (encrypted) multicast data from the source to the receivers over the public Internet. This problem is a subset of the general problem of routing security which has received attention, among others in [33, 34, 35, 36] in the context of unicast routing security.

Many of the solutions devised for data confidentiality and authentication, and those for group key management, are also applicable for securing the multicast routing protocol. These solutions may also be carried over into network management applications that deploy IP multicast to address groups of network devices. Hence, to a large extent, many of the infrastructure problems have solutions derived from those found in the core problems mentioned in Section 4.

In order to understand better the problem of security in multicast routing, it is useful to view the problem from two general points of origin where attacks can be launched:

- *Edge Attacks*: The attacks originates from a host connected to a subnet router at the leafs of the distribution tree. Two types of edge attacks are as follows.
  - *Sender attacks*: Here the distribution tree is attacked by the hosts sending bogus data packets to the group with the correct multicast address, thereby causing the packets to be sent to all receivers in the group. This attack consumes bandwidth, since the packet would be delivered to all host members. Although such attacks are possible also within unicast, the impact is magnified in multicast precisely due to the replication effect within the distribution tree. Such hosts may also send bogus control packets.
  - *Receiver attacks*: Here non-members simply join the group, causing the tree to expand and for multicast traffic to be forwarded to the non-member. Even if the traffic content is encrypted from the source (and thus useless to the attacker), the encrypted packets would still be forwarded regardless, thereby consuming bandwidth. The success of this type of attack is measured by the fact that the tree has expanded and thus consumed resources.
- *Internal Attacks*: The attacks originate from within the tree, either from a compromised router or from a tapped line.
  - *Data attacks*: Here the attacker injects bogus data packets into the data stream. The attack can be aimed at the subtree of members downstream from the point of attack, or the attacker can perform a sender attack with the result of the bogus data packets being received by all members of the group.
  - *Control attacks*: Here the attacker injects bogus control packets destined for other routers participating in the multicast distribution tree. The aim of this attack would be either to confuse and cripple the distribution tree or to influence the behavior of the distribution tree. Note that this kind of control attacks may also originate from the edge.

In practice, control attacks can be subtle, consisting of the attacker influencing the behavior of the multicast distribution tree without necessarily causing denial of service. Such an attack would depend, among others, on whether the multicast routing protocol uses the unicast routing table, on the type of unicast and multicast routing protocols, on the network topology, on the flow of traffic (unidirectional or bidirectional) and on the multicast application type. An attacker who has compromised a topologically strategic router may be able to introduce modifications to the routing table in such a way that a considerable amount of traffic is pulled towards that router. This, in turn, may cause the multicast distribution tree to also pass through the compromised router. At this point, the attacker may use the multicast distribution tree to her/his advantage. The situation is worse if the compromised router happens to be the core or rendezvous point of some multicast routing schemes (eg. CBT or PIM-SM).

Efforts to secure routing protocols, particularly unicast routing protocols, have been underway for a number of years [33]. These include the efforts of [37, 35, 36, 38] to secure the Border Gateway Protocol (BGP) [39], the work of [34, 40] to secure the OSPF unicast routing protocol [41, 42] and the work reported in [43] on the routing policy system security. Other efforts include [44] on link state routing, the work of [45] on distance vector routing protocols, and [46] on the Nimrod routing architecture.

In the context of multicast routing, the earliest efforts are reported in [47] related to the Core Based Trees (CBT) protocol [3]. More recently, the security of the Protocol Independent Multicast (PIM) protocol [5] – particularly the PIM Sparse Mode – has began to be addressed in [48] with the introduction of authentication via IPsec AH [49] on all PIM control messages. The approach is based on using a symmetric key and a keyed hash function, where all PIM routers within a PIM domain share a common symmetric key. Certain entities, such as the BootStrap Router (BSR) and the Rendezvous Point (RP), are also assigned public keys. The BSR, for instance, would then digitally sign the list of Candidate Rendezvous Points (C-RP) which it advertises into its PIM domain.

Since the proposal of [48] intentionally does not address the issue of key management for router-related keys, an accompanying proposal for PIM key management has been put forward in [50]. More recent efforts in the context of hierarchical routing have been reported in [51].

9 Security of Reliable Multicast Protocols

The topic of security in Reliable Multicast (RM) protocols is fairly broad, and hence we only treat it briefly here. For simplicity and convenience, the security of RM protocols at the transport layer is usually treated separately from the security of multicast at the IP layer. However, such a separation is artificial since the two are closely related and may in fact employ the same security mechanisms and policies.

When addressing the issue of security of RM protocols it is difficult to find a single solution for all RM protocols, since RM protocols employ different techniques to provide reliability (e.g., ACK-based, NAK-based, source-retransmission, repair-nodes) and use different entities (e.g., routers, servers, hosts) to implement reliability mechanisms. Thus, each RM protocol will require a unique solution for its security needs.

9.1 Common Requirements

Although RM protocols can differ widely in their general behavior and security requirements, all RM protocols have two (minimal) requirements in common:

- *Authentic control messages:* all control messages exchanged between RM entities must be authenticated.

  RM protocols require that all important control messages exchanged between RM entities be authentic. That is, exchanges of control messages should be protected against possible modifications which may lead to incorrect routing of data and/or other incorrect behavior.

- *Authentic retransmissions:* all retransmitted packets must be authenticated.

  An RM protocol must specify whether a retransmission entity (i.e., repair node) should apply its own authentication features (i.e., digital signature or MAC) whenever it retransmits a lost packet. The type of entity that performs the retransmission (e.g., actual source or repair node) may determine whether that entity can suitably apply authentication.

9.2 Application of Authentication Features

The next issue important to all RM protocols at the transport layer is *where* to place the authentication function.[1]

Authentication at the application layer has the advantage that a retransmission entity needs only to retransmit lost packets without adding its own authentication. The data and the authentication information (i.e., tags) are simply treated as a single payload. In effect, authentication is truly end-to-end from the sender to the receivers, independent of any reliability mechanism. This convenience, however, comes at the cost of IP and transport-layer headers being unprotected. Thus, a receiver may not be able sufficiently distinguish between an original packet being retransmitted from a repair node and an original packet being maliciously replayed.

The second option is to apply authentication at the IP layer (e.g., IPsec AH [32, 49]) to all packets relating to the RM protocol. The main advantage is that both IP and transport headers can be protected. However, this approach may require the retransmission entity to apply its own authentication features to all packets it retransmits. This is particularly the case if the retransmission entity uses IP destination address distinct from the multicast address specified in the original packet. To mitigate the problem, the retransmission entity can either encapsulate the original packet within its own (authenticated) IP packet, or simply strip away and transfer the payload to a new authenticated IP packet.

9.3 Source Authentication vs Group Authentication

Another security issue in RM protocols is the availability and deployment of public key technologies, particularly for message authentication. If public key cryptography is deployed then – regardless of where it is applied (e.g., application or network layer) – all control messages and retransmissions can be source authenticated.

[1] Cryptographic operations are typically placed at either (or both) network or application layers.

On the other hand, if public key cryptography is not deployed (e.g., for performance reasons) then group authentication via symmetric (shared) key is the only remaining avenue. Regardless of where it is applied, group authentication will only provide the receiver with the assurance that a message was sent by a group member. Hence, an honest receiver will not be able to distinguish a retransmission by the proper retransmission entity (either a repair node or the actual source/sender itself) from a retransmission (replay) by a dishonest group member who abuses the group key.

In general, the security of RM protocols exhibits some of the same problems as IP multicast security and, thus, might use the same solutions. These include key management, security policies as well as data authentication and confidentiality. Hence, solutions designed for IP multicast security should be considered in the larger context of potential use for RM protocols at the transport layer.

We can conceive a matrix consisting of the application of either source authentication or group authentication along the X axis against the layer in which they are applied (network or application layer) as the Y axis. Such a matrix would indicate the combinations of approaches possible either for eacg RM-related message or for a limited set of more crucial messages. For example, an RM protocol may deploy group authentication at the network layer for all RM-related messages, but apply encryption (either symmetric or asymmetric cryptography) at the application layer to retransmitted data. Another RM protocol might treat data confidentiality as an end-to-end issue between the sender and the receivers, and deploy only group authentication at the network layer for all control messages coupled with source authentication for all retransmissions by repair nodes. In summary, each RM protocol must identify its security weaknesses and other potential points of attack and address them in the fashion most suited to that particular RM protocol.

On the IETF front, work is underway [52] to address the security requirements of the RMTP-II protocol [53].

10 Advanced Issues in Group Communication

More complex, tighter-coupled group applications typically involve peer groups. (This is contrast to the more centralized or hierarchically organized groups encountered in subscription- or PPV-type settings.) Examples include replicated servers (e.g., database, web, time), audio and video conferencing and, more generally, collaborative applications of all kinds. Unlike other types of multicast groups, peer groups tend to be relatively small in size, at most about one hundred members. Larger groups are harder to control on a peer basis and are typically organized in a hierarchy of some sort. In this context, many-to-many communication pattern is usually assumed.

Due to their inherently more sophisticated and eclectic requirements, the security services in many-to-many peer groups still present a number of research challenges [54].

10.1 Key Management

Since most security services are based on sound key management, the latter must be addressed from the outset. Key management tends to be much more complex than in other multicast settings for a number of reasons.

- A single entity that generates and disseminates keys for a group (or many groups) represents a single point of failure and a likely performance bottleneck.
- If all group secrets are generated in one place, a central entity presents a very attractive attack target for adversaries.
- Some peer group environments are inherently mutually suspicious. For example, consider a group composed of members in different and competing organizations or countries.
- Some group settings are highly volatile often owing to the instability of the underlying network. Therefore, no single entity can be assumed to be present or available at all times.

10.2 Recent Work on Key Management for Peer Groups

In the last decade, several key agreement protocols geared for peer groups were proposed [55], [56], [57, 58], [59, 60, 61], [62]. All except [62] extend the well-known Diffie-Hellman key exchange method [63] to groups of n parties. A brief overview of each mechanism is given below.

We emphasize that, unlike key management approaches for IP multicast, these mechanisms were developed by cryptographers to serve the needs of a generic group-oriented application. Thus, they may be considered by some as being somewhat disconnected from reality. On the other hand, security is the dominating factor here. This is in contrast to more practical considerations such as amenability to easy deployment, scalability and co-existence with IP multicast.

One mathematically elegant approach was proposed by Fiat and Naor [56]. A trusted center T selects a RSA-like modulus $n = pq$ and a *secret* element $\alpha \in Zn^*$ of large multiplicative order (such that it is hard to compute discrete logarithms). For $1 \leq i \leq t$, each party M_i receives $\alpha^{x_i} \bmod n$ from T, with x_i random and relatively prime to x_j (for $j \neq i$). In order to establish a secret group key S, each M_i broadcasts x_i and then, after collecting all messages, computes $S = (\alpha^{x_i})^{x_1 \cdots x_{i-1} x_{i+1} \cdots x_t}$ mod n (i.e. a group Diffie-Hellman key with all the contributions). Drawbacks of this protocol are that 1) it requires a trusted third party (T) and, 2) as shown in [56], two or more parties can collude and recover the secret α.

A very different scheme was presented by Steer et al. in [55]. Given a $g \in Zp^*$ with p prime, each party M_i computes and broadcasts $y_i = g^{x_i} \bmod p$, where x_i is M_i's randomly chosen secret. After receiving all the contributions, M_1 computes the group key:

$$S = g^{x_t g^{x_{t-1} g^{\cdots x_3 g^{x_2 x_1}}}}.$$

Note that to come up with the same group key, the other protocol parties needs to behave differently since the order of exponentiation is right to left, i.e., the protocol is asymmetric. In particular, for $j = 3, \ldots, t-1$, M_j sends $v_j = g^{v_{j-1}^{x_j}}$ to M_{j+1} (where $v_2 = y_1$) and then computes S. The party M_t simply awaits v_{t-1} from M_{t-1} and then computes $S = v_{t-1}^{x_t}$.

Another notable result is due to Burmester and Desmedt [57]. They developed an efficient protocol which executes in only three rounds:

1. Each M_i generates its random exponent x_i and broadcasts $z_i = \alpha^{x_i}$.
2. Each M_i computes and broadcasts $W_i = (z_{i+1}/z_{i-1})^{x_i}$

3. Each M_i computes the group key $S = z_{i-1}^{tx_i} \cdot W_i^{t-1} \cdot W_{i+1}^{t-2} \cdots W_{i-2} \bmod p$

The resulting key is $S = \alpha^{x_1x_2+x_2x_3+\cdots+x_tx_1}$. The protocol is proven secure provided the Diffie-Hellman (DH) problem is intractable. However, there are some important assumptions underlying this protocol. It requires each M_i to broadcast to the rest of the group and to receive $t-1$ messages in a single round. Moreover, the system has to handle t simultaneous broadcasts.

Steiner et al. [59] introduced a class of protocols, called *generic* Group Diffie-Hellman (GDH) key agreement. The entire protocol class has been proven resistant against passive attacks. In brief, [60] shows that *if a 2-party DH key is polynomially indistinguishable from a random value then a t-party DH key is also polynomially indistinguishable from a random value.* Concrete protocols (GDH.2 and GDH.3) were demonstrated in [59]. Follow-ons to this work (all part of the CLIQUES project) include: [60] where membership change protocols are developed, and [61] where group key authentication and group membership integrity services are defined and demonstrated.

Recently, Poovendra et al. [62] came up with a group key agreement method which – unlike the aforementioned Diffie-Hellman based schemes – is based on simple XOR-ing of individual member contributions. This method is both simple and efficient. It also provides information-theoretic security (XOR-ing of random contributions is akin to *One-Time Pad*) as opposed to computational security of DH-based methods. However, its two main drawbacks are: 1) it requires separate provisions for message integrity and confidentiality, and 2) a trusted third party is needed to initialize the key agreement process.

10.3 Impact of Membership Changes on Key Management

In dynamic peer groups membership changes can occur at the granularity of single members: a member leaving (or being excluded) or a new member joining. Moreover, entire groups join and entire sub-groups are excluded; voluntary or otherwise. (The latter is typically caused by network faults and heals.)

Depending on the application policy, membership changes may need to be accompanied by corresponding adjustments to the group keying material. When performing such adjustments, both forward and backward secrecy must must be preserved. Recall that the former means that old keys cannot be known to new members and the latter – that new keys cannot be known to old members. This motivates the need for secure and efficient methods for re-keying (or adjusting the group secret) in the event of a membership change. Although, we emphasize that the actual decision whether a given membership change should result in a re-key is best left to the application policy.

Some groups are fairly static and can afford the expense of re-initializing the group whenever an infrequent membership change takes place. An example would be a group of routers or name servers that provide stable, long-term service.

More dynamic groups need specialized key adjustment protocols for handling membership changes. One very important requirement for these protocols is that they **must be re-entrant**. Re-entrancy here has the same meaning as in the traditional programming practice:

> key management must accommodate membership changes that occur while a key adjustment resulting from an earlier membership change is taking place.

We note that, with the exception of CLIQUES GDH.2/3 protocols, key management mechanisms surveyed above do not make explicit provisions for key adjustment.

10.4 Certification Issues

Group certification is essentially a wide-open problem. It spans issues such as certification of the group itself and certification of group membership for individual members. It also includes revocation of membership and the group public key(s). The latter is particularly important when secure communication with the outsiders (non-members) is desired..

The problems of issuance and maintenance of public key certificates have been well-researched in the recent years. As a result, certification mechanisms, such as X.509 and PGP, are enjoying wide use. However, if we view a tight, collaborative group as an aggregated entity (a composite user of sorts), it is only natural to ask how this entity can be certified. Before answering we first need to consider the reasons for wanting to certify a group:

- A group may be a relatively long-lived entity. For example, a group of experts (in a certain field) might form a long-term group. Anyone wishing to get an expert opinion can ask the group-at-large by sending a query encrypted with the group's public key. This public key must be obtainable from some type of a certificate.
- Outsiders (non-members) may need to communicate with the group or some of its members. In doing so, they need to be assured that communication really emanates from a *bona fide* group member. In other words, group member(s) may need to digitally sign certain communication; this also requires the availability of a group public key certificate.

Group certification cannot be addressed in a satisfactory manner without having to contend with some difficult issues, e.g.:

- Implications of dynamic membership
  Recall that traditional (individual) certificates are typically issued by an off-line Certification Authority (CA). If members frequently join and leave a group, and group keys change as a result, how can group certificates be issued efficiently? Who should issue them?
- Individual vs. opaque certification and authentication
  Should membership certificates be issued to every group member individually thus enabling finely granular group credentials and individual member authentication? Or, should certificates be issued *opaquely* to the entire group thus allowing for the anonymity of individual members?

10.5 Tiered Groups

Some of the complex group applications are actually *few-to-many* rather than *many-to-many* in nature. For example, collaborative visualization application might include many "observer" members but few members authorized to control the visualization instruments. In other words, some groups are two-tiered: they involve a small number of senders and a comparatively large number of receivers. While the senders can be viewed as a collaborative peer group, the entire group is not collaborative. The challenge in this environment is twofold:

- How to provide *scalable* and secure group communication with stronger guarantees for senders and weaker for receivers?

  One relevant recent development is the *InterGroup* protocol proposed by Berket at UCSB [64]. InterGroup targets precisely those multicast groups where receivers greatly outnumber senders while reliable delivery is nonetheless required. However, InterGroup is focused on reliability, not security.

- How to preserve security in a way that treats senders and receivers as different types of security principals? (For example, only senders can participate in key generation and agreement while receivers only obtain a fraction of or, a derivative of, the group key.)

  This issue is more complicated. One potential approach is to extend group key management protocols to perform key agreement only among senders while offering key distribution for the larger group of receivers. The main difficulty here lies not in the actual mechanisms but in the associated policy issues. For example, different actions (key changes) might be appropriate if the population of senders changes as opposed to that of receivers.

10.6 Other Security Services

The primary motivation for obtaining a group key is the resultant ability to communicate **securely and efficiently**. Once a group key is established members can communicate bulk data using symmetric encryption for privacy and keyed hash functions (eg. HMAC [65]) for integrity. There are few obstacles for either of the above; the particulars for data encryption and data integrity are similar to the 2-party case.

Many real-world applications require group members to communicate securely not only among themselves but also with non-members, i.e. the rest of the world. Replicated servers forming a group (eg., time servers or stock market quote servers) may need to communicate with privacy and authenticity to the outside clients. This requires the availability of group public keys.

Assuming that the certification issue can be addressed, it is easy to derive a public key from a group secret. For example, a group secret key can be viewed as a Diffie-Hellman exponent and a corresponding group public key can be computed as simply the residue thereof (i.e., is S is a secret key, $P = g^S modp$ is the public key; for appropriately chosen base g and prime p.) Then, outsiders can establish shared keys with the entire group in a trivial manner using plain Diffie-Hellman key exchange. Similarly, a group secret can be used to derive an El Gamal public key-pair. Outsiders can then use this key in El Gamal public key encryption and communicate in private with the group member(s).

Other, more group-specific services can be envisaged. In particular, a group key composed of secret shares (where each share is known only to one specific group member) can be used to provide a number of useful services as described below.

10.6.1 Member Authentication

As alluded to in Section 4 authentication takes on a different meaning in a peer group context. The following table tries to capture the various authentication flavors:

	Group Authentication	Source Authentication	Anonymous & Unlinkable (but traceable)
Within group	Group secret key	Member secret share	Group signatures
With outsiders	Group public key	Member secret share	Group signatures

Each entry in the table illustrates what is needed to achieve a specific "Quality-of-Authentication" (QoA). In particular, a shared group secret used as input to (for example) a keyed MAC in a well-designed authentication protocol can authenticate a group member to other members. The same secret key in tandem with a corresponding group public key (ie., a secret key used to produce DSA or El Gamal signatures) can authenticate a member to the outside. In either case, the authentication is anonymous and unlinkable since authenticating member demonstrates the knowledge of a secret known to the entire group.

If more granular intra-group authentication is desired, the group secret itself can be thought of as a public quantity. An authenticating member (source) can demonstrate knowledge of its own secret share with respect to the public quantity (the group secret). For example, in case of CLIQUES protocols [61], a member M_i can authenticate by demonstrating possession of its secret share N_i of a group secret $S = g^{N_1 \cdots N_i \cdots N_n} mod p$. It does so by computing a Schnorr signature [66] to prove knowledge of the discrete logarithm of S to the base $g^{N_1 \cdots N_n / N_i} mod p$. (The last quantity is a public intermediate value.) The resultant authentication is *source-specific.*

A similar approach is possible for authenticating *specific* members to the outside. The only difficulty is that the group secret is not available to the verifier (the outside party), hence, the group public key must be used in its place. This, in turn, complicates the authentication procedure. In the example of CLIQUES, the member would need to demonstrate be knowledge of *double* discrete logarithm of α^S to the bases of α and $g^{N_1 \cdots N_n / N_i}$, respectively. (Techniques for this are available albeit at significant cost [67].)

Finally, it might be required to authenticate members in an anonymous and unlinkable manner, however, with a "safety hatch" that would allow the anonymity to be revoked in case of a dispute or misbehavior. For this, a specialized *group signature* [68] construct would be necessary. Although generally expensive to set up, group signatures have a number of important properties and potential applications. For more detail, consult [68], [67], [69], [70], [71], [72].

11 Summary

This paper aimed to provide some insight into the issues related to IP multicast security and to the possible directions of future developments. Three problem areas were identified, namely those problems which are at the core of IP multicast security, those pertaining to the infrastructure security of IP multicast, and the complex applications that may be built upon a secure IP multicast.

The core problem areas that have been identified are the issues relating to fast and efficient source authentication methods (particularly for high data rate multicasting), the issues relating to group key management and those relating to policies – at different levels – specific to IP multicast security.

The infrastructure security problems captures both the security of multicast routing protocols and the security of the Reliable Multicast (RM) protocols. The first is analogous to the unicast routing security problem, which has received much attention in the last few years. Similar to the unicast problems, control packet authentication represents the main security concern voiced by implementors of the protocols. Control packet authentication is also a main requirement of RM protocols. In addition, RM protocols must also address the issue of the sender authentication of retransmitted packets by repair nodes or other retransmission entities.

Although IP multicast provides a basic level of service for group-oriented communication, many applications feature complex behaviors and member interactions. Since such complex behaviors are usually not addressable at the network layer – but represent an integral part of group communication – they were discussed as part of the application layer. Thus, we see IP multicast security as enabling technology for these more complex group-oriented protocol.

Specific attention was given to the activities and developments occurring in the IETF since these are typically practical and have the promise of being the basic foundation for any multicast application-specific solution.

References

[1] S. Deering, "Host extensions for IP multicasting," RFC 1112, IETF, 1989.

[2] D. Waitzman, C. Partridge, and S. Deering, "Distance vector multicast routing protocol," RFC 1075, IETF, 1988.

[3] T. Ballardie, P. Francis, and J. Crowcroft, "Core based trees: An architecture for scalable inter-domain multicast routing," in *Proceedings of ACM SIGCOMM'93*, (San Francisco), pp. 85–95, ACM, 1993.

[4] J. Moy, "Multicast extensions to OSPF," RFC 1584, IETF, 1994.

[5] S. Deering, D. Estrin, D. Farinacci, M. Handley, A. Helmy, V. Jacobson, C. Liu, P. Sharma, D. Thaler, and L. Wei, "Protocol Independent Multicast – Sparse Mode: Motivations and architecture," Aug 1998. `draft-ietf-pim-arch-05.txt` (Work in Progress).

[6] W. Fenner, "Internet group management protocol version 2," RFC 2236, IETF, 1997.

[7] B. Cain, S. Deering, and A. Thyagarajan, "Internet group management protocol version 3," Feb 1999. `draft-ietf-idmr-igmp-v3-01.txt` (Work in Progress).

[8] T. Hardjono, "Secure Multicast Group (SMuG) Reference Framework," March 1999. `http://www.ipmulticast.com/community/smug` (IRTF Work in Progress).

[9] R. Canetti and B. Pinkas, "A taxonomy of multicast security issues," Nov 1998. `draft-canetti-secure-multicast-taxonomy-01.txt` (Work in Progress).

[10] T. Hardjono, B. Cain, and N. Doraswamy, "A framework for group key management for multicast security," Feb 1999. `draft-ietf-ipsec-gkmframework-01.txt` (Work in Progress).

[11] S. Mittra, "The Iolus framework for scalable secure multicasting," in *Proceedings of ACM SIGCOMM'97*, pp. 277–288, ACM, 1997.

[12] D. Wallner, E. Harder, and R. Agee, "Key management for multicast: Issues and architectures," Sept 1998. `draft-wallner-key-arch-01.txt` (Work in Progress).

[13] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[14] G. J. Simmons, "An introduction to shared secret and/or shared control schemes and their application," in *Contemporary Cryptology: The Science of Information Integrity* (G. J. Simmons, ed.), pp. 441–497, IEEE Press, 1992.

[15] D. Balenson, D. McGrew, and A. Sherman, "Key management for large dynamic groups: One-way function trees and amortized initialization," Feb 1999. `draft-balenson-groupkeymgmt-oft-00.txt` (Work in Progress).

[16] H. Harney and C. Muckenhirn, "Group key management protocol (GKMP) specification," RFC 2093, IETF, July 1997.

[17] H. Harney and C. Muckenhirn, "Group key management protocol (GKMP) architecture," RFC 2094, IETF, July 1997.

[18] D. Harkins and N. Doraswamy, "A secure scalable multicast key management protocol (MKMP)," November 1997. (Work in Progress).

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

[20] A. Ballardie, B. Cain, and Z. Zhang, "Core Based Trees (CBT version 3) multicast routing," August 1998. `draft-ietf-idmr-cbt-spec-v3-01.txt` (Work in Progress).

[21] T. Hardjono, B. Cain, and I. Monga, "Intra-domain group key management protocol," Nov 1998. `draft-ietf-ipsec-intragkm-00.txt` (Work in Progress).

[22] D. Meyer, "Administratively scope IP multicast," RFC 2365, IETF, July 1998.

[23] M. Handley, D. Thaler, and D. Estrin, "The internet multicast address allocation architecture," Dec 1997. `draft-handley-malloc-arch-00.txt` (Work in Progress).

[24] R. Canetti, P. Cheng, D. Pendarakis, J. Rao, P. Rohatgi, and D. Saha, "An architecture for secure internet multicast," Feb 1999. `draft-irtf-smug-sec-mcast-arch-00.txt` (Work in Progress).

[25] H. Harney and E. Harder, "Group security association key management protocol," Apr 1999. `draft-harney-sparta-gsakmp-sec-00.txt` (Work in Progress).

[26] C. K. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs," in *Proceedings of ACM SIGCOMM'98*, ACM, 1998.

[27] I. Chang, R. Engel, D. Kandlur, D. Pendarakis, and D. Saha, "Key management for secure internet multicast using boolean function minimization techniques," in *Proceedings of Infocom'99*, (New York), IEEE, March 1999.

[28] R. Canetti, T. Malkin, and K. Nissim, "Efficient communication-storage tradeoffs for multicast encryption," in *Proceedings of Eurocrypt'99*, Springer-Verlag, 1999.

[29] H. Harney and E. Harder, "Logical Key Hierarchy (LKH) protocol," Mar 1999. `draft-harney-sparta-lkhp-sec-00.txt` (Work in Progress).

[30] H. Harney and E. Harder, "Multicast Security Management Protocol (MSMP) requirements and policy," Mar 1999. `draft-harney-sparta-msmp-sec-00.txt` (Work in Progress).

[31] D. Harkins and D. Carrel, "The internet key exchange (IKE)," RFC 2409, IETF, Nov 1998.

[32] S. Kent and R. Atkinson, "Security architecture for the Internet Protocol," RFC 2401, IETF, Nov 1998.

[33] R. Perlman, "Network layer protocols with byzantine robustness," Technical Report MIT/LCS/TR-429, Massachusetts Institute of Technology, October 1988.

[34] S. L. Murphy and M. R. Badger, "Digital signature protection of ospf routing protocol," in *Proceedings of the 1996 Network and Distributed System Security Symposium*, (San Diego), ISOC, 1996.

[35] A. Heffernan, "Protection of BGP sessions via the TCP MD5 signature option," Mar 1998. `draft-ietf-idr-bgp-tcp-md5-00.txt` (Work in Progress).

[36] T. Bates, R. Bush, T. Li, and Y. Rekhter, "DNS-based NLRI origin AS verification in BGP," Feb 1998. `draft-bates-bgp4-nlri-orig-verif-00.txt` (Work in Progress).

[37] S. Murphy, "BGP security analysis," Aug 1998. `draft-murphy-bgp-secr-01.txt` (Work in Progress).

[38] T. Przygienda, "BGP-4 MD5 authentication," Nov 1997. `draft-przygienda-bgp-md5-00.txt` (Work in Progress).

[39] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, IETF, 1995.

[40] S. Murphy, M. Badger, and B. Wellington, "OSPF with digital signatures," RFC 2154, IETF, 1997.

[41] J. Moy, *OSPF: Anatomy of an Internet Routing Protocol*. Addison-Wesley, 1998.

[42] J. Moy, "OSPF version 2," RFC 2328, IETF, 1998.

[43] C. Villamizar, C. Alaettinoglu, D. Meyer, S. Murphy, and C. Orange, "Routing policy system security," May 1998. `draft-ietf-rps-auth-01.txt` (Work in Progress).

[44] R. Hauser, T. Przygienda, and G. Tsudik, "Reducing the cost of security in link-state routing," in *Proceedings of the 1997 Network and Distributed System Security Symposium*, (San Diego), ISOC, 1997.

[45] B. R. Smith, S. Murthy, and J. J. Garcia-Luna-Aceves, "Securing distance vector routing protocols," in *Proceedings of the 1997 Network and Distributed System Security Symposium*, (San Diego), ISOC, 1997.

[46] K. E. Sirois and S. T. Kent, "Securing the nimrod routing architecture," in *Proceedings of the 1997 Network and Distributed System Security Symposium*, (San Diego), ISOC, 1997.

[47] T. Ballardie and J. Crowcroft, "Multicast-specific security threats and counter-measures," in *Proceedings of the Symposium on Network and Distributed Systems Security - NDSS'95*, (San Diego), ISOC, 1995.

[48] L. Wei, "Authenticating PIM version 2 messages," Nov 1998. `draft-ietf-pim-v2-auth-00.txt` (Work in Progress).

[49] S. Kent and R. Atkinson, "IP authentication header," RFC 2402, IETF, Nov 1998.

[50] T. Hardjono and B. Cain, "Simple key management protocol for PIM," Mar 1999. `draft-ietf-pim-simplekmp-00.txt` (Work in Progress).

[51] C. Shields and J. Garcia-Luna-Aceves, "KHIP - a scalable protocol for secure multicast routing," in *Proceedings of ACM SIGCOMM'99*, ACM, 1999. (To appear).

[52] T. Hardjono and B. Whetten, "Security requirements for RMTP-II," May 1999. `draft-ietf-rmtp-ii-sec-00.txt` (Unpublished Work in Progress).

[53] B. Whetten, M. Basavaiah, S. Paul, and T. Montgomery, "RMTP-II specification," Apr 1998. `draft-whetten-rmtp-ii-00.txt` (Work in Progress).

[54] J. Smith and F. Weingarten, "Research challenges for the next generation internet," May 1997. Report from the Workshop on Research Directions for NGI.

[55] D. Steer, L. Strawczynski, W. Diffie, and M. Wiener, "A secure audio teleconference system," in *Advances in Cryptology – CRYPTO'88*, August 1990.

[56] A. Fiat and M. Naor, "Broadcast encryption," in *Advances in Cryptology – CRYPTO'93*, August 1993.

[57] M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system," in *Advances in Cryptology – EUROCRYPT'94*, May 1994.

[58] M. Just and S. Vaudenay, "Authenticated multi-party key agreement," in *Advances in Cryptology – EUROCRYPT'96*, May 1996.

[59] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-hellman key distribution extended to groups," in *ACM Symposium on Computer and Communication Security*, March 1996.

[60] M. Steiner, G. Tsudik, and M. Waidner, "Cliques: A new approach to group key agreement," in *IEEE Conference on Distributed Computing Systems*, May 1998.

[61] G. Ateniese, M. Steiner, and G. Tsudik, "Authenticated group key agreement and friends," in *ACM Symposium on Computer and Communication Security*, November 1998.

[62] R. Poovendran, S. Corson, and J. Baras, "A shared key generation procedure using fractional keys," in *IEEE Milcom 98*, October 1998.

[63] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644–654, 1976.

[64] K. Berket, L. Moser, and P. Melliar-Smith, "The intergroup protocols: Scalable group communication for the internet," in *Proceedings of the 3rd Global Internet Mini-Conference*, November 1998.

[65] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Key hashing for message authentication," RFC 2104, IETF, February 1997.

[66] C. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, 1991.

[67] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," in *Advances in Cryptology – CRYPTO'97*, 1997.

[68] D. Chaum and E. van Heyst, "Group signatures," in *Advances in Cryptology – EUROCRYPT'91*, 1991.

[69] J. Camenisch, "Efficient and generalized group signatures," in *Advances in Cryptology – EUROCRYPT'97*, 1997.

[70] G. Ateniese and G. Tsudik, "Group signatures a' la carte," in *ACM/SIAM Symposium on Discrete Algorithms (SODA'99)*, January 1999.

[71] G. Ateniese and G. Tsudik, "Some open problems and new directions in group signatures," in *Financial Cryptography'99*, February 1999.

[72] H. Petersen, "How to convert any digital signature scheme into a group signature scheme," in *Security Protocols Workshop*, 1997.