

Internet Engineering Task Force
INTERNET-DRAFT
[draft-ietf-ipsec-gkmframework-03.txt](#)
August 2000

T. Hardjono (Nortel)
B. Cain (Mirror Image)
N. Doraswamy (Photonex)

Expires: February 2001

A Framework for Group Key Management for Multicast Security

<[draft-ietf-ipsec-gkmframework-03.txt](#)>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Abstract

This document provides a framework for group key management for multicast security, motivated by three main considerations, namely the multicast application, scalability and trust-relationships among entities. It introduces two planes corresponding to the network entities and functions important to multicasting and to security. The key management plane consists of two hierarchy-levels in the form of a single "trunk region" (inter-region) and one or more "leaf regions" (intra-region). The advantages of the framework among others are that it is scalable, it has reduced complexity and allows the independence in regions of group key management.

INTERNET DRAFT

August 2000

Table of Contents

1.	Scope of Document and Philosophy	3
2.	Group Key Management: Background	5
3.	Group Key Management: Issues	6
3.1	Multicast Application Types.	7
3.1.1	One-to-Many Multicast Applications	7
3.1.2	Many-to-Many Multicast Applications.	8
3.2	Size and Distribution of Group Members	8
3.3	Scalability of Protocols and Membership Management	9
3.4	Independence of GKM Protocols	10
3.5	Trust-Relationships	10
3.6	Group Authentication and Sender Authentication.	11
3.7	Identities and Anonymity.	11
3.8	Access Control.	12
3.9	Membership Verification	13
3.10	Failure of Systems	13
3.11	Other Issues	14
4.	Framework: Basic Model.	15
4.1	Basic Model	15
4.2	Trunk-Keys and Leaf-Keys.	17
4.3	Interpretations of Regions.	18
4.4	Security Associations and Secure Channels	18
4.5	Advantages of the Framework	18
5.	Example of Framework Application.	19
5.1	One-to-Many Multicast Example	19
5.1.1	Scope of Leaf Regions	19
5.1.2	Location of Key Managers.	20
5.1.3	Advertising Key Managers.	21
5.2	Many-to-Many Multicast Example.	21
5.2.1	Location of Key Managers.	21
5.2.2	Scope of Leaf Regions	21
5.2.3	Advertising Key Managers.	22
6.	References.	22
7.	Authors Addresses	23

INTERNET DRAFT

August 2000

1. Scope of Document and Philosophy

This document proposes a framework for group key management (GKM) for multicast security on the Internet. The objective of the framework is to foster the development of an Internet-wide solution while encouraging innovations in solving the many problems that are related to multicast security. Since multicast security has many complex facets, related to multicast technologies and security technologies respectively, the following two-pronged approach is recommended corresponding to a two level hierarchy:

1. Encourage the growth and evolution of novel secure solutions for group key management within pre-defined key management "regions" ("domains") whose scope is determined on a per-case basis. Regions can be defined to be size of subnets, autonomous systems (AS), or larger. This will allow for the development of independent and innovative solutions that are addressed for specifically for such regions, taking into consideration the multicast application being employed.
2. Encourage secure, simple, consistent and stable interactions between the key management regions that implement the various group key management solutions. This will allow for the development of innovative "inter-region" ("inter-domain") solutions that can consistently and securely tie together the various regions deploying "intra-region" ("intra-domain") group key management protocols.

By defining "regions" of group key management, various schemes can be used for each region independent of one another, with only the requirement that they can interact with a common, and simple, "inter-region" group key management protocol. In this way, the need of a single all-encompassing scheme for Internet-wide group key management will be removed. This will allow for different region-scoped group key management schemes to be developed concurrently while an "inter-

region" scheme and architecture is being developed.

The aim of this document is to describe a simple framework for group key management that addresses some issues specific to multicast security. In doing so, the framework relies on existing security technologies, notably IPsec and its related protocols for unicast key management. It is not the aim of the document to specify the details of a group key management scheme or architecture. Nor is it an objective to specify the details of the interactions of group key management schemes between regions that implement them. On the region level, the goal is to develop basic GKM requirements, while allowing maximum freedom for the development of solutions. On the Internet-wide scale, the aim is to identify basic GKM functions and facilitate the development of a protocol (or enhancement of an existing GKM protocol) that allows relatively simple interactions between regions of group key management.

Hardjono, Cain, Doraswamy

[Page 3]

INTERNET DRAFT

August 2000

The framework proposed in this document approaches the multicast security problem, and more specifically, the group key management problem, by first introducing two planes corresponding to the network entities and functions pertaining to multicasting and to security. The first plane, called network infrastructure plane, encompasses the entities and functions that define the network, which in the case of IP multicasting includes the various protocols (eg. routing protocols) and the entities that implement them (eg. routers, hosts). The second plane, called the key management plane, encompasses the entities and functions of the network define and establish security in the network, which in the case of IP multicasting includes security-related protocols (ie. GKM protocols, IPsec and its related protocols, cryptosystems) and entities that implement them (eg. key generators, key managers, policy server, routers).

Within the key management plane two hierarchy of regions are introduced, namely one "trunk region" and one or more "leaf regions". The trunk region is bounded by certain key manager entities and does not contain any member hosts (senders/receivers). All member hosts are defined to exist within leaf regions, each of which is associated with (at least) one key manager entity. The purpose of introducing leaf regions and a trunk region is for the framework to inherently promote scalability by allowing regions to be defined according to the available entities and protocols in underlying network infrastructure plane and according to the multicast application under consideration.

Since the current framework also aims at promoting the clear identification of trust-relationships that exists (both explicitly and implicitly) among entities in the network that are involved in securing multicast, it has identified two general multicast application types that have differing trust-relationships. These are the One-to-Many multicast applications and the Many-to-Many multicast applications.

From the security perspective, the identification of the two multicast application types is aimed at distinguishing the different possible mappings between the two planes, which in turn determines the trusted entities involved in securing the multicast transmission and also determines the trust-relationships among the entities. This process in turn determines the jurisdictions over the key managers in the two respective multicast applications, and thus the physical locations of the key managers.

The jurisdiction over the key managers and the locations of the key managers will then determine to a large extent the applicable "intra-region" group key management within each leaf region and the suitable "inter-region" solution that binds the leaf regions together securely.

2. Group Key Management: Background

The topic of group-oriented security has been researched for over two decades now, particularly in the area of cryptology in the context of secure conferences (eg. [1-4]), secret-sharing (eg. [5-7]) and digital multi-signatures (eg. [8]). Most of the results of such research has been theoretical, which are valuable in the long term, but which are too difficult or too computationally-intensive to be implemented as a solution to the current pressing multicast security needs.

The IETF has provided security standards for the Internet by introducing the IPsec standard and its related technologies [9]. Although these technologies satisfy to a large extent the needs of secure communications in the Internet, they are aimed mainly at unicast transmissions between one sender and one receiver.

More recent efforts to address the security needs multicast have

taken the form of group key management protocols with the aim of securely delivering a common key to all members of a multicast group.

Having such a key allows group members to encipher the traffic within the multicast group. Thus, the group key also affords membership-enforcement by only allowing key holders to decipher the multicast traffic. A sender must encipher all traffic that it sends to the group.

The advantage of having a group key is that a sender avoids having to encipher traffic individually for each receiver. However, related to this is the issue of re-keying of the group key should a member ceases membership or a new host takes-on membership.

Although group-authentication is implicitly provided through the possession of the key, sender-authentication must be provided through other means (eg. signature of individual sender). Examples of GKM protocols can be found in [10-13]. Some, if not all, of these proposals suffer from one drawback or another, the most common being scalability in the context of re-keying.

The work of [11] employs a Group Controller which works in conjunction with group members in creating and delivering keys to the members. The group controller also performs checking on the permission of candidate members. The use of a centralized entity to control key management presents limitations from the perspective of scalability.

The problem of scalability is directly addressed by the work of [12], where a hierarchical ordering of subgroups is employed to limit the effects of re-keying. The key management at different levels of the

hierarchy is carried-out by different controller entities. Thus, when re-keying occurs to a member within a subgroup, only the members in that subgroup will be affected. Although the scheme points to an attractive direction in terms of limiting the effects of re-keying, it suffers the drawback of needing a decryption/re-encryption of traffic as it enters or leaves a subgroup. (This drawback can be limited to a certain extent if the controller entities perform their decryptions and re-encryptions using cryptographic hardware).

The work of [10] follows from the work on the Core Based Trees (CBT) multicast routing protocol. Here, the idea is to employ the core of the tree to distribute keys to candidate members, who must contact

certain routers that are connected to the core. These routers then carry-out membership checks and key distribution to the candidate members. Although the scheme maybe scalable, it is dependent on CBT as the multicast routing protocol and hence poses difficulties when used with other multicast routing protocols.

The recent proposal of [13] addresses the scalability problem by separating the key generation entity from the key distribution entity. The key distribution entity can be dynamically added by requesting their participation. Authority would then be delegated to such key distribution entities together with access control lists. Candidate members can then request membership and a copy of the group key from the key managers. Although promising and can be hierarchically organized, the extent of the scalability of the approach remains to be seen due to the problem of re-keying in the case of hosts joining/leaving.

One promising direction has been the recent work of [14]. Here, a logical tree of keys is created at a server that generates all the keys and coordinates the key management among the members of the group. The members are divided into subgroups, each being assigned a key. Depending on the need, the keys at different levels in the logical tree would then be applied. Although attractive, as it stands the scheme of does not scale well due to the dependence on a centralized server for all aspects of key management. More specifically, the need of the server to hold the private-key of each member may prove to be too burdensome for the server.

3. Group Key Management: Issues

There are a number of issues related to multicast security in general, and more specifically group key management. These issues are listed (non-exhaustive) in the following and are discussed in the ensuing sections.

- Multicast application types
- Size and distribution of members
- Scalability of protocols and membership management
- Independence of GKM protocol
- Trust-relationships

- Group authentication and sender authentication
- Identities and anonymity
- Access control and membership verification
- Security and practicality of protocols
- Failure of systems
- Denial of service (DOS) attacks

[3.1](#) Multicast Application Types

The current framework recognizes that an all-encompassing solution for multicast security is difficult, if not impossible, to achieve (and even undesirable) due the various multicast applications that exist today and may emerge in the future. To this end, two general multicast application types have been identified in the effort to provide a common ground for discussing the issues related to multicast security and group key management.

[3.1.1](#) One-to-Many Multicast Applications

The first multicast application type covers the cases where the multicast group has one sender and multiple receivers. Transmission is unidirectional from one sender to many receivers. The receivers are assumed to be passive consumers of the data, while the single sender is the producer of the data. The initiator of the group is assumed to be its owner, and for simplicity it is also assumed to be the sender. Examples this multicast application includes Pay Per View (PPV) programs (eg. Internet TV, Radio, Video) and other real-time data (eg. news, stock prices, etc).

The One-to-Many multicast applications correspond to the cases where the transmitted data carries an immediate value for which the receivers must also be subscribers. Hence, they would be of interest to commercial companies seeking to use multicast as a medium for transmitting data over the Internet to as wider audience as possible.

Two general cases exist with respect to the data being transmitted. In the first case, the group is concerned more about the authenticity and integrity of the data, and not so much its confidentiality. An example would be subscriptions to publicly available data (eg. stock market data, government publications). These desired effects can be achieved using public key techniques and message integrity techniques, leaving the data itself readable to non-members.

Of more concern here is the second case, where the aim is to prevent non-members from accessing the data. An example would be subscriptions to subscribers-only transmissions (eg. pay per view Internet TV). Here, encryption techniques can be used for controlling access to the data.

It is in the interest of the initiator/sender to ensure that only legitimate members (subscribers) of the group obtain access to the contents of the multicast, by encrypting the contents. It is also in the interest of the initiator/sender to ensure that only legitimate members of the group obtain a copy of the encryption key. Consequently, it is in the interest of the sender/initiator to be in control over the entities that implement group key management (eg. key managers).

[3.1.2](#) Many-to-Many Multicast Applications

The Many-to-Many multicast application type refers to the case where the relationship between the sender and receiver(s) is equal (democratic) and where the data is of immediate value only to the members of the group. Every member of the multicast group is both a sender and a receiver. An example of this multicast application would be conferencing.

Membership of the group maybe Open or Closed. In the Open Many-to-Many multicast anyone can join the conference provided that the identity of the member is known. In the Closed Many-to-Many multicast only a predefined number of members can join, and the identities of the members are known in advance.

The aim in the Many-to-Many multicast application type is to prevent non-members from accessing the data. Hence, encryption (in addition to message authenticity and integrity techniques) is used for controlling access to the data that is of immediate value only to the members of the group.

It is in the interest of all members of the group to ensure that only legitimate members obtain access to the contents of the multicast. Consequently, it is in the interest of members to select the entity it controls under its jurisdiction to participate in the group key management.

[3.2](#) Size and Distribution of Group Members

The IP multicast model is attractive because its membership can extend throughout the Internet, subject only the capability of the multicast routing protocol and the availability of resources. One of

INTERNET DRAFT

August 2000

the main attractive points about the IP multicast model is that, in effect, a large numbers of host members can be reached without the sender needing to know the size of the group and the distribution of the group.

In the context of security, however, the identity of the communicating parties is an inherent requirement of authentic and confidential communications. In IPsec it is the basis of the security association between two communicating parties, which leads to the secure key-agreement between them. Hence, in GKM protocols that employ secured unicast communications, the size and distribution of the members become issues that have a direct impact on the scalability of the protocols.

The effectiveness of a group key management protocol and its underlying multicast routing protocol is dependent to a certain extent on the size of the group and on the distribution of the group (dense or sparse). Related to this is also the issue of the frequency of changes to the membership, which may lead to the need of re-keying and other changes in the security parameters of the group.

[3.3](#) Scalability of Protocols and Membership Management

One of the primary issues in group key management is that of the scalability of the protocols it employs. Often these protocols rely on one (or few) security-managing entity (eg. key server) that is assumed to be trusted by all other entities in the entire Internet. Furthermore, the protocols often require host members to communicate securely with the entity (unicast). Not only does the entity (or entities) become a bottleneck in the scheme, it also becomes the "best" point of attack by intruders since it necessarily holds security parameters pertaining to the host members.

Also impacting scalability of protocols is the method used to perform a re-keying of the group key due a host member leaving the group or a new one joining. Re-keying of the group key involves a new group key being delivered to the (affected) members of the group. Ideally, a protocol should strive to minimize the number of affected host members in the case of re-keying and to minimize the number of messages exchanged during the re-key process, particularly if secure

(authentic and confidential) unicast messages must be exchanged. The work of [12] on the Iolus system represent a conscious attempt address this problem by specifically designating areas that would be affected by a re-keying event due to changes in the membership of the group.

Other issues related to membership management include specifying how the decision regarding members joining or leaving (or ejection) is reached. This issue is largely dependent on the multicast application being employed.

[3.4](#) Independence of GKM Protocols

Regardless of the scope of a group key management protocol, such a protocol must be independent of (or decoupled from) the underlying multicast routing protocol, thereby allowing it to be used in conjunction with various multicast routing protocols. This, however, does not exclude the use of GKM protocols that are tightly coupled with a given multicast routing protocol should it be chosen for certain areas or organizations in the Internet.

For a GKM protocol to be independent from multicast routing protocols, the GKM protocol must not rely on the structures (eg. distribution tree) and mechanisms inherent to any particular routing protocol. A GKM protocol must also be separate from the session advertisement protocol (eg. SAP). However, sufficient information about a group and its related GKM protocol security parameters must be advertised in order for a host wishing to become a member to engage in the GKM protocol.

[3.5](#) Trust-Relationships

Many group key management protocols for multicast security have proposed the use of certain entities to manage security-related information and parameters without specifying:

- on what basis such an entity is accorded trust
- who accorded the trust to the entity
- under whose jurisdiction (administrative or otherwise) the entity resides

- who else in the Internet is assumed to trust that entity

The problem of trust-relationship is a difficult one and several factors influence it, among others:

- the multicast application and the definition of regions may influence one another, which may also influence the applicable group key management protocols
- the distribution of the members over the Internet may influence which entities are trusted by the members (eg. a member may only trust entities physically within its country)

- the availability (or lack) of a certification infrastructure that allows for certificates specifying trust to be widely accessible on the Internet and for delegations to occur
- historical records about attacks to certain areas or organizations on the Internet may deter host members from trusting entities in that area/organization

Research on this issue has been continuing for a number of years. However, a practical approach embodying trust-relationships specifically for multicast security on the Internet has yet to be proposed.

In the long term, one possible solution to this problem may consist of the codification within certificates of the trust between organizations in the manner of Service Level Agreements (SLA). Such certificates may then be the basis for accepting or rejecting entities that manage security-related information and parameters.

[3.6](#) Group Authentication and Sender Authentication

In schemes in which a group key is used to encrypt the group traffic to afford membership control the decipherability of a multicast packet implies its origination from one of the members of the multicast group. That is, group authentication is achieved at the same time as data confidentiality.

This level of authentication, although sufficient for some multicast applications, may not be enough for other applications in which the precise identity of the sender of the multicast packet needs to be known by the receivers of the packet. That is, sender authentication must be provided in addition to group authentication.

One simple approach to sender authentication within a multicast group would be for each member of the group to digitally sign the messages it sends, before the message is enciphered using the group key. This approach requires the use of public key cryptography, and depending on the multicast application, it may also require the existence of a public key infrastructure for its scalability.

[3.7](#) Identities and Anonymity

As referred to previously, one of the attractive points about the IP multicast model is that, in effect, a large numbers of host members could be reached without the sender knowing identity of the receivers. IGMP Membership-reports are used by the receivers to

report about memberships. The presence or absence of (at least) a member determines whether the router joins onto the multicast distribution tree. The identity of a member (eg. IP address) is never relayed by the router to the sender, and hence the sender never knows the identity of the receiver.

In secure communications between a sender a receiver, the identity of each of the communicating parties is an important parameter which must be convincingly verified by the other. This is typically achieved by resorting to certificates that embody the identity, supported by a certification infrastructure. In the context of multicast at the network layer the certificate for a host must contain the Distinguished Name (DN) or other equivalent unique identification information corresponding to the host. This verifiable identity becomes the basis for a host being admitted into a multicast group and for the host to be given the group key through the appropriate GKM protocol.

Since the current framework views IPsec and its related technologies

for unicast security as the building blocks for multicast security (and that IPsec requires the identities to be known) it makes little sense to discuss anonymity of hosts at the network layer (or lower). The issue of anonymity is better addressed at the application layer. It must be pointed-out, however, that anonymity does not imply non-identification. That is, even in systems that feature anonymity (eg. electronic payment systems) a unique pseudonym [15] is used to identify one user from another. It is the mapping of the pseudonym to the user's personal information that must remain secret.

[3.8](#) Access Control

The IP multicast model allows for any host to become a member of the group simply by requesting to join the group. Other group members may not necessarily be aware of the existence of other members in the group.

Although the IP multicast model may be attractive in its native form to some applications, from the perspective of security such unlimited membership may be undesirable. The current framework views access control policies and their implementation to be an issue tightly related to the multicast application type.

Similar to the independence (decoupling) of the group key management protocol from the underlying multicast routing protocol, the current framework proposes the independence of the group key management protocol from the access control model and implementation. This does not, however, preclude the possible developments (or extensions) of multicast routing protocols that exhibit some form of (limited) access control.

[3.9](#) Membership Verification

Related to access control and member identities is the need for membership verification at the network layer. More specifically, membership verification refers to the ability of a member of a multicast group to request information and self-verify the constituents of the group. Although this functionality may not be necessary (or even undesirable) in certain multicast applications (eg. pay per view transmissions), it may be highly desirable for

other applications (eg. conferencing).

Solutions to the membership verification issue have been suggested in the context of cryptographic conference key distribution schemes, in which membership verification is in-built into the scheme itself or is a major feature of the scheme (eg. [2, 16]). However, the complexity of these cryptography-based solutions may point to the application layer as being the best place for them to be implemented.

In general, at the very minimal membership verification can be achieved by a trusted party (eg. group initiator) vouching for a membership-list by digitally-signing it and distributing it to all group members. Such a trusted party may be one from which new members must obtain group-keys when they join the group, and hence will in fact hold security information pertaining to each member of the group.

Another possible approach is to deploy a special membership verification protocol, much in the spirit of IGMP, which reports in a secure fashion about the membership identities within a given subnet or a larger area. Such an approach will be related to the existence of a certification infrastructure and have to address the issue of the trustworthiness of the entities (eg. router) than implement the membership verification protocol.

[3.10](#) Failure of Systems

It is a requirement that when a network entity (eg. host or a router) carrying security parameters fails, it must not divulge or allow the security parameters that it holds to be compromised in anyway. The

security of the entity must not be lessened when the entity experiences failure. Hence, the entity must exhibit a "fail-closed" behaviour with respect to security.

Although to a large extent this issue is one related to systems implementation, the awareness of potential vulnerabilities that may exist when an entity boots-up or fails should lead protocol designers and implementers to take this matter into consideration when developing hardware and software for network elements.

3.11 Other Issues

There are other issues related to the security of multicast and to group key management. These are listed as follows (not exhaustive):

- Denial of service (DOS) attacks
- Authenticity of multicast routing exchanges
- Non-repudiation of group membership and key possession
- Frequency of periodic re-keying
- Tamper-proof storage on network entities (eg. routers)
- Secure boot-up of multicast-related network entities

4. Framework: Basic Model

Although there are a variety of multicast security issues that must be resolved, the current framework is motivated by three main considerations, namely the multicast application, scalability and trust-relationships among entities. The framework aims at being as general as possible while remaining practical and useful.

The framework proposed in this document approaches the multicast security problem, and more specifically, the group key management problem, by first introducing two planes corresponding to the network entities and functions pertaining to multicasting and to security. The first plane, called "network infrastructure plane", encompasses the entities and functions that define the network, which in the case of IP multicasting includes the various protocols (eg. routing protocols) and the entities that implement them (eg. routers, hosts). The second plane, called the key management plane, encompasses the entities and functions of the network define and establish security in the network, which in the case of IP multicasting includes security-related protocols (ie. GKM protocols, IPsec and its related protocols, cryptosystems) and entities that implement them (eg. key generators, key managers, policy server, routers).

Within the key management plane two hierarchies (levels) of regions are introduced, namely one "trunk region" and one or more "leaf regions". The trunk region is bounded by certain key manager entities and does not contain any member hosts (senders/receivers). All member hosts are defined to exist within leaf regions, each of

which is associated with (at least) one key manager entity. The purpose of introducing leaf regions and a trunk region is for the framework to inherently promote scalability by allowing regions to be defined according to the available entities and protocols in underlying network infrastructure plane and according to the multicast application under consideration.

Although the current framework proposes a two-level hierarchy (namely a trunk region and leaf regions) in the key management plane, it does not preclude the use of a single-level arrangement. In a single-level arrangement the framework essentially consists of a large leaf-region. The usability of a single-level region from the perspective of scalability would be dependent on the multicast application type and the size/distribution of the group members.

[4.1](#) Basic Model

Network infrastructure plane:

This view is a physical/topological view. The Internet is seen a collection of autonomous systems (AS), some being Stub ASs and others being Transit ASs (ie. ISPs) and various backbone connections.

This plane identifies the entities and functions that define the network, which in the case of IP multicasting includes the various protocols (eg. routing protocols) and the entities that implement them (eg. routers, hosts).

Key management plane:

This plane encompasses the entities and functions of the network that promote and implement security in the network. For multicast security these include security-related protocols (ie. GKM protocols, IPsec and its related protocols, cryptosystems) and the entities that implement them (eg. key generators, key managers, policy server, routers).

Key management regions:

This plane also introduces the logical structure consisting of a key management "trunk region" and one or more key management "leaf regions" (Figure 1). This size/scope of these regions is determined by multicast application in question.

Key Managers (KMs):

One important entity in the current model is the Key Manager (KM) entity. Two kinds of KMs are assumed to exist, namely Border KMs and Non-Border KMs.

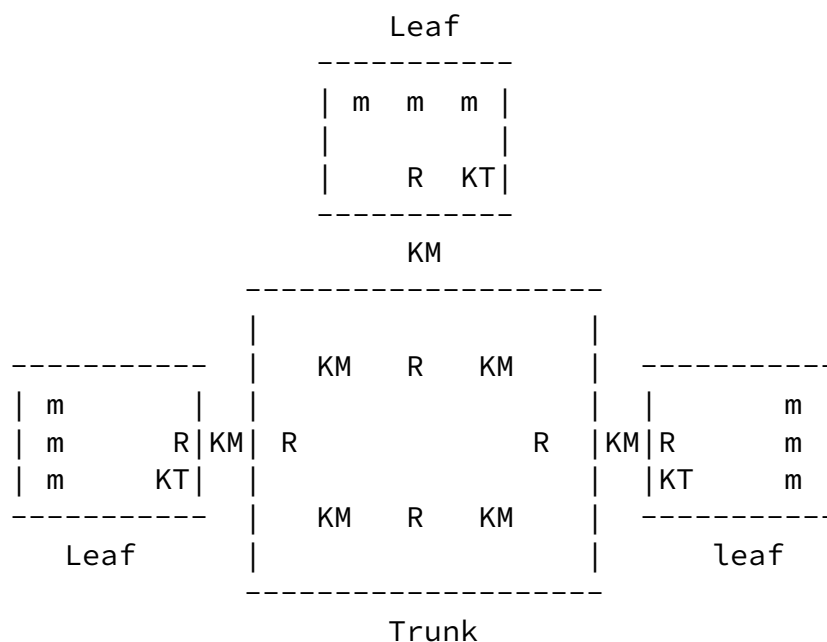
The trunk region is bounded by Border KMs and does not contain any member hosts (ie. no sender/producer or receiver/consumer of the multicast traffic). Each leaf region is associated with (at least) one Border KM.

Non-Border KMs may reside inside the trunk region or inside leaf regions. However, they do not participate in the definition of the boundary (scope) between the trunk region and leaf regions.

Member hosts are defined to exist within leaf regions. A leaf region is defined to contain a member host (at least one) and one (or more) multicast-capable routers.

As an example, the mapping of the two planes may result in an instance where a leaf region maps to a Stub AS and the trunk region maps to a set of Transit ASs.

If there are multiple KMs then a method of election for the principal KM for the region is assumed to be employed (the method of election beyond the scope of this document).



INTERNET DRAFT

August 2000

Key Translators (KTs):

Another important entity in the current model is the Key Translator (KT) entity. The function of the KT entity is to "translate" payload (which can be either multicast data or the group key) from being encrypted under one key to another key. The process of decrypting payload (ciphertext) using one key and enciphering the resulting plaintext using another key must be atomic, reliable and tamper-free. Each leaf region is associated with one (or more) KT entity. Such a KT entity may be implemented in the form of a fast router or server, containing high-capacity cryptographic hardware or software. The translation may be applied to multicast data or used for key management purposes (eg. delivering a group key).

[4.2](#) Trunk-Keys and Leaf-Keys

In the current framework each key management region (trunk region and leaf regions) is assumed to be associated with a different cryptographic key. A leaf region is assumed to be associated with a unique leaf-key, while the trunk region is associated with a trunk-key.

The trunk-key is only known among the Border KMs. The trunk-key is generated through an (inter-region) group key management protocol in the trunk region among the Border KMs.

Leaf-keys are generated through the local group key management (GKM) protocol (such as [13]), whose scope of key distribution is defined to be limited to the size of the leaf region.

Since the Border KMs demarcate the boundary between the trunk region and the leaf regions, the Border KM associated with a given leaf region also holds a copy of the leaf-key of that leaf region. The Border KM associated with a leaf region is assumed to be involved in the local GKM protocol of that leaf region.

In summary, a Border KM associated with a leaf region holds a copy of the trunk-key which it shares only with the other Border KMs and it holds a copy of the leaf-key of the leaf region with which it is

associated. A Border KM does not share the copy its leaf-key with entities outside its associated leaf region.

For simplicity, in the remainder of the work the term "key manager" or "KM" will refer to Border KMs. For any multicast group, the KM (ie. Border KM) associated with the leaf region of the group initiator will be called the Initiator KM (IKM). The leaf region where the initiator resides will correspondingly be called the Initiator Leaf, while the other leaf regions will be referred to as Remote Leafs. The KM associated with a Remote Leaf will be referred to as the Remote KM (RKM).

[4.3](#) Interpretations of Regions

From the point of view of the application of cryptographic keys (namely trunk-keys and leaf-keys), the logical structuring into regions (trunk region and leaf regions) leads to (at least) two possible interpretations:

(a) Regions for delivering a group key:

The region-based key management to create secure channels for the purpose of the distribution of a (group-wide) group key. The group key is then applied to the multicast data in the group from Source/Sender to the Receiver (end-to-end), without translations. The keys to create this channel must in-turn be securely managed and are treated on a per-region basis. Thus, the regions pertain to the secure channels.

(b) Regions for delivering multicast data:

Here, each region applies a different key to the multicast data as it transits across regions (from the sender's leaf region into the trunk region, and finally into the receiver's leaf region). Hence, translations in the form of decryption and re-encryption of the multicast data at borders of regions occur.

These two interpretations do not contradict the scalability requirement, since they both still produce the desired result of limiting the effects of re-keying to that of regions. Combinations of the two interpretations can also be conceived.

[4.4](#) Security Associations and Secure Channels

A primary assumption in the current framework is that all security-sensitive communications between entities is carried-out through "secure channels" (with mutual authentication, data confidentiality and data integrity). The secure channels are based on security associations (SA) and are implemented using IPsec and its related technologies. Also assumed in the use of certificates (and thus the existence of a certification infrastructure) that underlies the establishment of security associations and thus secure channels.

[4.5](#) Advantages of the Framework

The current framework has a number of advantages. These are discussed in the following.

Scalable: By design the framework promotes scalability since it allows new leaf regions to be added, independent of existing leaf regions and independent of the population of members in each leaf region.

Reduced Complexity: By employing a limited level (2 level) of key management regions, the complexity of key management for multicast security is greatly reduced. Each leaf region can perform its leaf-scoped key management functions independent of other leaf regions. Key management among the KMs in the trunk region can be performed independent of the key management in leaf regions.

Long life of trunk keys: Due to the fact that the trunk region employs a different key (trunk-key) from the leaf regions, the KMs need not re-key the trunk-key immediately when a member of a multicast group in the leaf region leaves or is ejected. The KM can keep its copy of the trunk-key even after its associated leaf region ceases to have any members of the multicast group.

Grafting new members: Since the KM associated with a leaf region does not need to immediately discard its copy of the trunk-key after the associated leaf region ceases to have any members, the KM is ready for the grafting of the new members in the associated leaf region. The KM does not have to obtain a copy of the trunk-key associated with a multicast group every time its previously non-empty leaf region becomes devoid of members, and then becomes populated again.

Independent Re-key Period: The trunk region and the leaf regions are free to set their own periodic re-key period. Varied opinions have been voiced in the field of computer and network security regarding the need of periodic re-keying in any scenario where communicating parties share a common "private" (symmetric) key. This need is due to the increased vulnerability of frequently used keys to cryptanalysis.

[5. Examples of Framework Application](#)

As mentioned previously, the size/scope of regions is influenced by multicast application in question. The mapping between the network infrastructure plane and the key management plane also defines the entities involved in the multicast instance, most notably the Key Managers (KMs). The physical location of the KMs and the jurisdiction under which it functions is dependent on the multicast application in question. In the following, two examples are given based on the two multicast application type previously identified.

[5.1 One-to-Many Multicast Example](#)

[5.1.1 Scope of Leaf Regions](#)

The One-to-Many multicast application corresponds to the situation where:

- the group has one sender and multiple receivers
- the multicast traffic carries direct value to non-members
- the attacker's gain lies in illegally re-distributing the traffic to the widest audience
- example: Pay Per View (PPV) and other subscriber services

It is therefore in the interest of the initiator/sender to ensure that only legitimate members (subscribers) of the group obtain access to the contents of the multicast. Hence, it is in the interest of the sender/initiator to be in control over the entities that implement group key management.

Since the initiator/sender is the producer of the data, and effectively the owner, and since there can be a variety of configurations involving other parties (eg. ISPs), the initiator/sender itself must:

- define the scope/size of each leaf region
- define the physical location of the key managers
- define the trust-relationships among the entities involved in securing the multicast

The initiator may choose to define leaf regions to be the size of an autonomous system, a larger region composed of several autonomous system, a geographic state, geographic region of the country, or larger. It may define leaf regions to be a function of the accessibility provided by an Internet Service Provider or similar organizations.

[5.1.2](#) Location of Key Managers

The issue of selecting key managers that can be accorded trust is largely determined on whether the initiator (producer) has control (directly or indirectly) over the entity being the key manager:

Direct control: the initiator may choose to have several key managers (eg. server farm), all physically within its own leaf region. Each key manager would be associated one (remote) leaf region.

Indirect control: the initiator may choose to employ other parties trusted to provide a given level of service based on some agreement (ie. outsourcing). This is analogous to the concept of trusted certification organizations where the notion of trust is codified in the form of legally-binding contractual agreements, in such a way that it is economically disadvantageous for a trusted certification organization to cheat. In the current context, the trusted third party can be organizations such as Internet service providers (ISPs) to which the initiator/sender is directly connected, trusted certification organizations or other organizations offering security management services.

[5.1.3 Advertising Key Managers](#)

The current framework is not concerned about how key managers are advertised, but rather about what information is advertised about the key managers. The list of available key managers can be made known to hosts wishing to become a member through session advertisements (ie. SAP/SDP) using one of two methods:

The advertisement carries not only the identity of the Initiator KM (IKM), but also the list of the available KMs associated with the multicast group.

The advertisement carries only the identity of the IKM. Interested hosts must send the join-request to the advertised IKM that will then forward it to the appropriate KM (ie. IKM or RKM).

[5.2 Many-to-Many Multicast Example](#)

[5.2.2 Location of Key Managers](#)

The Many-to-Many multicast application corresponds to the situation where:

- the group members are both senders and receivers
- the multicast traffic carries indirect value to non-members
- the attacker's gain lies in providing the contents of the multicast to a limited audience
- Example: Confidential company conference meeting

The distribution of rights and obligations within the Many-to-Many multicast application type is more democratic. It is in the interest of each member to maintain the security of the multicast. Hence, it is in the interest of each member to select the most trustworthy entity under its jurisdiction to be the KM associated with the member's leaf region and for that entity to be securely administered.

[5.2.2 Scope of Leaf Regions](#)

Here the implication is that the key manager associated with a leaf region should be under the jurisdiction and administration of that leaf region. This further implies that for Many-to-Many multicast application type the most suitable size of a leaf region may be that of an autonomous system (AS) corresponding to the members' organization. Only by its own organization administering the key manager can a member be assured that its interests are best served.

[5.2.3](#) Advertising Key Managers

As mentioned previously, the current framework is not concerned about how key managers are advertised, but rather about what information is advertised about the key managers. The session advertisement (ie. SAP/SDP) for the Many-to-Many multicast application type must always carry the identity of the IKM.

Depending on the openness of the membership of the group (ie. open or closed membership), upon creating a new multicast group the initiator host must provide the Initiator KM (IKM) with additional information:

Open Many-to-Many: since anyone can join the multicast provided that the identity of the member is known, the initiator provides an Access Control List (ACL) for the group.

Closed Many-to-Many: since only a predefined number of members can join, the initiator provides the IKM with a list of allowable members.

A host wishing to join must send the join-request (containing the its identity) to the RKM it selects. The RKM in turn will forward the request to the IKM together with the identity of the RKM. It is then up to the IKM to decide membership using on the identities of the host and its associated RKM.

6. References

- [1] I. Ingemarsson, D. T. Tang, and C. K. Wong, "A Conference Key Distribution System," IEEE Transactions on Information Theory, vol. IT-28, pp. 714-720, 1982.
- [2] K. Koyama and K. Ohta, "Identity-based Conference Key Distribution Systems," presented at Advances in Cryptology - CRYPTO'87 (Lecture Notes in Computer Science No. 293), 1987.
- [3] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman Key Distribution Extended to Group Communications," presented at Proceedings of the 3rd ACM Conference on Computer and Communications Security, New Delhi, 1996.
- [4] M. Burmester and Y. Desmedt, "Efficient and Secure Conference Key Distribution," presented at Security Protocols (Lecture Notes in Computer Science No. 1189), 1996.
- [5] A. Shamir, "How to share a secret," Communications of the ACM, vol. 22, pp. 612-613, 1979.
- [6] 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., 1992, pp. 441-497.
- [7] Y. Zheng, T. Hardjono, and J. Seberry, "Reusing Shares in Secret Sharing Schemes," *The Computer Journal*, vol. 17, pp. 199-205, 1994.

INTERNET DRAFT

August 2000

- [8] T. Okamoto, "A Digital Multisignature Scheme Using Bijective Public-Key Cryptosystems," *ACM Transactions on Computer Systems*, vol. 6, pp. 432-441, 1988.
- [9] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," IETF, [RFC 1825](#) 1998.
- [10] T. Ballardie, "Scalable Multicast Key Distribution," IETF, [RFC 1949](#), 1996.
- [11] H. Harney and C. Muckenhirn, "Group Key Management Protocol (GKMP) Architecture," IETF, [RFC 2094](#), July 1997.
- [12] S. Mitra, "The Iolus Framework for Scalable Secure Multicasting," presented at Proceedings of ACM SIGCOMM'97, 1997.
- [13] D. Harkins and N. Doraswamy, "A Secure Scalable Multicast Key Management Protocol," IETF, IETF Draft [draft-ietf-ipsecond-00.txt](#), November 1997.
- [14] C. K. Wong, M. Gouda, and S. Lam, "Secure Group Communications Using Key Graphs," presented at Proceedings of SIGCOMM'98, 1998.
- [15] D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Communications of the ACM*, vol. 24, pp. 84-88, 1981.
- [16] K. Ohta, T. Okamoto, and K. Koyama, "Membership authentication for hierarchical multigroup using the extended Fiat-Shamir scheme," presented at Advances in Cryptology - Proceedings of EUROCRYPT'90 (Lecture Notes in Computer Science No. 473), Aarhus, Denmark, 1990.

[7](#). Authors Addresses

Thomas Hardjono
Nortel Networks
[600](#) Technology Park Drive
Billerica, MA 01821, USA
Email: thardjono@baynetworks.com

Brad Cain
Mirror Image
[49](#) Dragon Court

Woburn, MA 01801
Email: brad.cain@mirror-image.com

Naganand Doraswamy
Photonex
8C Preston Court
Bedford, MA 01730
Email: naganand@photonex.com