

Multicast Security Management Protocol (MSMP) Requirements and Policy

Status of this memo

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

This document is an Internet-Draft. 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/1id-abstracts.txt>

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

Document expiration: August 30, 1999

Abstract

This Internet-Draft describes issues relating to the management of cryptographic keys in support of multicast communications. It describes the functional and security requirements of an electronic key management system for multicast.

Copyright Notice

Copyright © The Internet Society (1999). All Rights Reserved.

1	INTRODUCTION	3
1.1	Desirable Features	4
1.2	Candidate Applications	4
1.2.1	Teleconferencing	5
1.2.2	Broadcast (NNTP, NASA broadcast)	5
1.3	Security for Multicast	6
1.3.1	Securing Multicast Packets	6
1.3.2	Managing Secure Groups	7
2	REQUIREMENTS	7
2.1	Real World Requirements	8
2.1.1	Performance	8
2.1.2	Flexibility/Modularity	9
2.1.3	Scalability	9
2.2	Security Requirements	11
2.2.1	Algorithm Definition	11
2.2.2	Key Generation Definition	11
2.3	Authorization Definition	12
2.3.1	Access Control	12
2.3.2	Key Dissemination Architectures	13
2.3.3	Trust	14
2.3.4	Authorization	15
2.3.5	Rekey Approach	15
2.3.6	Compromise	16
2.4	Protocol Requirements	17
2.4.1	Self Defined	17
2.4.2	Communications Protocol Independent	18
2.4.3	Architecture Independent	19
3	POLICY COMPONENTS	19
3.1	Security Policy	19
3.2	Architecture	20
3.2.1	Operational Policy	20
3.2.2	Key Dissemination Policy	20
3.2.3	Access Control Policy	21
3.2.4	Rekey Policy	21
3.2.5	Compromise Policy	21
4	DESIGN RECOMMENDATIONS	22
4.1	Global Policy Mechanism	22
4.2	Limited Group of Security Mechanisms	22
4.3	Policy Decomposition Of Multicast Protocols	23
5	Addresses of Authors	26

1 INTRODUCTION

Multicasting technology has been a promise for many years now. A blend between unicast (point to point) and broadcast (on sender, many, unidentifiable receivers), multicasting allows a group of participants to communicate efficiently between themselves using public networks. Security has been a key area holding back widespread adoption of multicast.

Group communications can be obtained using unicast methods (e.g., send an e-mail to each participant), but this has an impact on the network infrastructure, requiring sufficient resources to send each message from the sender to each recipient uniquely (an e-mail to 100 addresses requires the sender to actually send 100 messages). Using multicast, information is sent only once into the multicast infrastructure and the infrastructure only creates new messages/packets when needed. Depending upon the networking technologies in use, multicast can be performed with a single message.

Multicasting, in general, provides the capability for information to be disseminated to an identified group of participants efficiently. Multicasting is typically performed by creating a group where participants place information destined for all other participants. This group can be in the form of a newsgroup, IP address, or ATM address.

The security challenge for multicasting is in providing an effective method of controlling access to the group (and it's information) that is as efficient as the underlying multicast. A primary method of limiting access to information is through encryption and selective distribution of the keys used to encrypt group information. Control of the key distribution process provides effective control of the group. The controlling policy for key distribution may differ among groups. For instance, organizations may wish to distribute keys to particular individuals or units based on location or permissions; banks may wish to limit key distribution to particular trusted individuals; or individuals may wish to limit distribution to particular family members. The range of options is limitless.

Establishing this cryptographic group on an internet is not a trivial task. The entire group must converge on a single suite of security mechanisms for data protection. The single cryptographic key must be created and distributed to all members of the group in a secure manner. Some type of access control policy must be enforced as part of the key distribution mechanism. These policies must be created and disseminated to the groups in a manner that can be trusted.

The decision to create a cryptographic group on the internet is based on the data that is going to be passed across the network and the needs of the communicating group. If the data passed across the network is extremely important and not time sensitive, the security policy for creation, dissemination, and access control may be stringent. Alternately,

if the data is not very sensitive, the security policies of the group may be more relaxed. This is an important distinction because there is

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 3]

a trade-off between security (assurance that the policy is in effect) and performance (time and resources necessary to implement the policy). The job of coordinating that trade-off falls to a management protocol.

This paper identifies and discusses the security and key management requirements for cryptographic groups. This includes group creation, group key creation, key distribution, policy creation, policy distribution, access control and group behaviors (management, rekey and compromise recovery). The goal is to craft the specific requirements for a Multicast Security Management Protocol (MSMP).

1.1 Desirable Features

The desirable features for a MSMP include:

- The security management protocol shall operate in a heterogeneous communications protocol environment
- The security management protocol shall provide and utilize all reasonable security mechanisms to provide high assurance to security-relevant management events.
- The security management protocol shall protect the group from all known security attacks pertaining to security management.

1.2 Candidate Applications

In looking to the internet, the Inter-Domain Routing Protocol (IDRP) and the Distance Vector Multicast Routing Protocol (DVMRP) use multicast as a mechanism for parties to relay common information to their peers. Each party both sends and receives information in the multicast channel. As appropriate, a party may choose to leave or join the communication without the express permission of any of the other parties. More interestingly, the multicast internet protocol (IP) model has the receiver telling the network to add it to the distribution for a particular multicast address, whether it exists yet or not, and the sender is not consulted as to the addition of the receiver.

Other applications of multicast communications in the internet (e.g., NASA select broadcasts) can be viewed as implementing the sender model because the sender selects the broadcast time, channel, and content, though not the destinations.

1.2.1 Teleconferencing

Video or audio teleconferencing is one model of multicast communications. Widespread use of the video or audio teleconferencing applications will result in many small groups existing at one time. These groups will be highly dynamic. Individual users may have several applications, or instances of applications, running simultaneously with different keys. Individuals will gain access to groups based on their network address and on personal characteristics (e.g., name, organization, physical location, authorizations) that may be contained in cryptographic certificates. There may even be a secondary mechanism for finer grained access management controlled locally.

1.2.2 Broadcast (NNTP, NASA broadcast)

Another scenario for group keys is a large single keyed group. There are some interesting environmental constraints on key management imposed by the characteristics of extremely large groups (e.g., network news and broadcast). Network news transmissions represent the case of extremely large groups where each recipient receives the same data package. The keying of a secure network news group is complicated by the unidirectional characteristic of the communications. The sheer size of network newsgroups precludes any sort of standard reply from each recipient, as these acknowledgments would easily consume all available network bandwidth for popular groups.

cooperative enforcement of the group security policies would require that all entities enforcing the access control policy were trusted to do so. This requirement may seem difficult to manage. Yet, the group with access to the data decryption key are trusted to protect that data. It seems logical that those same members should be trusted, by default, to protect the key that is protecting the data. Essentially, the group members trusted to protect the data being encrypted are available, and trusted, to enforce the groups' access control policy. The problem devolves to how do we use those members to speed group establishment.

The security of the key is inversely proportional to the number of holders of that key. This observation leads to some potential alternatives for controlling the keys protecting information in such a group. One alternative for large groups is to compose it of smaller groups connected by ``cryptographic gateways''. (1) In principle, if any single endpoint goes

1. Cryptographic gateway refers to a device that is trusted to decrypt and re-encrypt traffic from one ``enclave'' to another. Such a device may be a specialized multicasting gateway, providing security translation service between a local

network and the multicast backbone. Also, such a device may be

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 5]

bad, the compromise is confined to that communication group. In effect, the compromised cryptographic key would have limited utility. The geographical location of that communication net bounds the utility of the key. Systems that require actual broadcast of secure packets (e.g. satellite downfeeds and some cable architectures) could not use the meshed large cryptographic group.

1.3 Security for Multicast

The issue of secure multicast communications for multicast groups has two parts. The first part consists of the mechanisms used to secure the data while it is in transit between the multicast group members. The second part is the management of the security groups. Management in this case, refers to:

- Creation and distribution of keys,
- Enforcement of access control policies, and
- Operational control (e.g., compromise recovery, rekey, identity infrastructure issues).

1.3.1 Securing Multicast Packets

When a group of entities share a cryptographic key, for encryption of data traffic over a multicast address, they all share use of that key. Multicast communications allow any member of the group to encrypt a message and have it decrypted by multiple destinations. The sender ID is included in an IP packet but any member of the group can create a packet with any sender ID making it impossible to unambiguously distinguish the source of the transmission based on the key used to decrypt the transmission. This implies that a separate mechanism must authenticate the source for transmission in a cryptographic group.

Several mechanisms exist that can authenticate individual sources of transmission in a cryptographic group. The most obvious and widely used mechanism is the digital signature. Digital signatures have the advantage of being received by a wide audience and being created by a very narrow audience. They have the disadvantage of taking a long time (as compared to encryption) either to sign or to verify. Depending on the type of communication going on, the time required to use a digital signature may

employed where there are issues of cryptographic releasability, allowing for

groups to be created, that use several cryptographic algorithms.

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 6]

make it impractical.(2) For communications that are not time sensitive, it may be reasonable to apply a digital signature. Network news maybe appropriate for digital signatures.

[1.3.2](#) Managing Secure Groups

The MSMP encompasses all the issues of a cryptographic group. The management of multicast secure groups is most likely an application layer protocol. Each group of members needs an instance of the management application layer protocol. Those protocol instances need to cooperate to successfully enforce the group's policies and provide keys and group management information.

There are many management issues associated with the securing of a multicast group, including:

- Key generation procedures,
- Key distribution to all group members,
- Commonly understood group mechanisms, and
- Orchestrated group actions.

The next section outlines the details of the target requirements for such a group management protocol.

[2](#) REQUIREMENTS

A clear collection and definition of the multicast security and key management requirements will help in the definition of the MSMP. Many people have an idea of how to solve multicast key management problems for specific systems. The requirements presented in this paper were collected from the requirements for multicast key and security management from different types of systems. Several multicast security proposals were also reviewed to include their stated requirements.[[1](#), [2](#), [3](#), [4](#), [5](#), [8](#)]

2. The use of digital signatures for streaming applications may be impractical on a ``packet-by-packet'' basis, though it may be possible to perform a digital signature verification on a periodic basis over ``chunks'' of the previously transmitted stream. Also, the use of a running cryptographic checksum, initialized by an authenticated message

(signed precursor), may also serve this purpose.

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 7]

2.1 Real World Requirements

There are two broad requirements of the real world: efficiency and utility. MSMP must present a useful functionality set for most applications. It must contain enough options to allow it to operate across heterogeneous systems and configurations. Unfortunately, the desire to provide a functional tool set for the widest range of applications conflicts with other concerns, namely efficient use of resources and performance.

2.1.1 Performance

- MSMP shall establish small groups in a few seconds.
- MSMP shall support large groups that never converge.
- MSMP shall support confirmation of group convergence (merge) in large groups, where required by group policy.
- The MSMP should be able to accommodate a variety of convergence states.

2.1.1.0.1 Resource Utilization

It is desirable to perform management operations when they have the least operational impact. It may be useful to describe a performance curve for multicast security management over the life span of a secure group. The set-up phase of the group is where a majority of the management functions should occur. Interactions that would interrupt the group (rekey, compromise recovery, leave, join) should be localized to members of the group requiring the service. These interactions should also be streamlined to minimize the impact on the legitimate group members.

In the group communications of a multicast group, the security management set-up takes place coincident with the group set-up. During normal group communication, the security management of the group is merely a watchdog effort ensuring the group is operating correctly. During a re-key, leave, or join, security management occurs, but it is minimal and localized, if possible. The group communications processing increases if there is a compromise of a group member. If compromise recovery is possible for a group, the security management protocol will become active in keying the compromised individual out of the group. In most instances, a new group is created that excludes the compromised entity. The security management protocol would also support documentation of information for a forensic review of the compromise.

In summary, the MSMP must:

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 8]

- Front load processing requirements (set-up) and
- Provide audit mechanisms.

2.1.2 Flexibility/Modularity

MSMP must be flexible enough to apply to many different environments. It must be modular to easily allow users to adapt the protocol to their environment. One mechanism to create an efficient and highly flexible protocol is to provide a single architecture that supports multiple specialized sub-protocols. To some degree, it may make sense to make protocol "objects" optimized for a particular need.

In summary, the MSMP must:

- Support multiple environments and
- Provide mechanisms for the expansion and optimization for special environments.

2.1.3 Scalability

Scalability refers to the protocols' ability to do two things -- support groups with large numbers of users and support large numbers of individual groups. Unfortunately, these two architectures can be at odds with each other.

2.1.3.1 Many Group Members

Some multicast groups have an extremely high number of sites. Usually, most group members are receive-only and very few are transmit. There are some interesting requirements associated with this type of group. The security protocol may need to operate "out of band" and each individual site will need to correlate keys to the appropriate group address.

There are architectural issues with whether a group like this should even share a single key. Today's architecture relays a single message around the globe. This may not be desirable in the case of a secure group. A key shared by many people really will not protect much information. Of course, it is also true that if a key holder cannot be trusted to protect the key, nor can they be trusted with the information protected by the key.

An alternative architecture to the single key per group is the large

group built up of smaller groups connected by cryptographic gateways.

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 9]

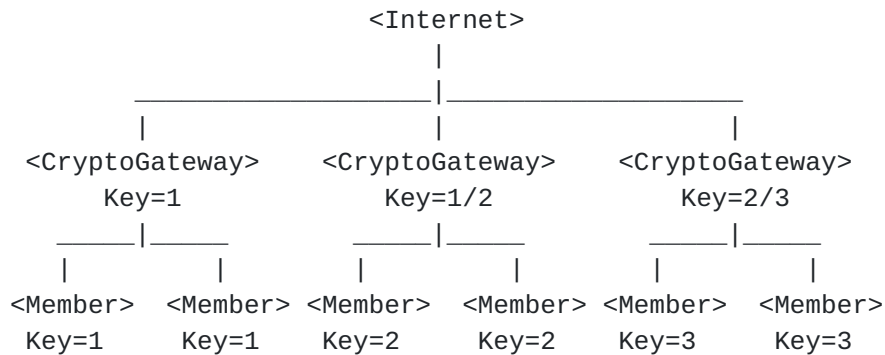


Figure 1: Use of Cryptographic Gateways to Reduce Key Exposure

Figure 1 presents this type of large group. These composite groups have the advantage of limiting the utility of any single cryptographic key and tighter control can be placed on access control by specifying local trusted controllers. The MSMP need do very little to support this type of group. Each local group is feasibly a normal cryptographic group. The cryptographic gateway can either be a local decision or it could be stated in the policy. See Figure 1

In summary, the MSMP must:

- Enforce communications policy defined by group and
- Link single key(s) to individual groups.

2.1.3.2 Large Numbers of Small Groups

Another scalability issue is that the protocol must support a large number of groups each with a fairly small number of members. It seems reasonable to predict that the internet will probably have many IP video or audio teleconferences occurring at any one time.

The scalability issues with this scenario deal with availability of resources. An approach that relies on a central server in establishing groups would likely experience problems as the number of groups increases and, given the dynamic nature of groups, the group's lifespan decreases. That central server would become very busy and a potential single point of failure.

In summary, the MSMP must Support small group proliferation without creating communication or processing overloads.

2.2 Security Requirements

There are security requirements inherently tied to a protocol that manages keys [3,4,5]. The following sections attempt to identify all of these possible requirements. Protocols designed to service a particular environment will have a tailored subset of requirements. However, a generic MSMP must be flexible enough to satisfy these broad requirements.

The use of cryptography to protect data shifts the burden of security to the management of the cryptographic key. In essence, control of the key is equivalent to control of the data, and key management becomes the pivotal point for cryptographic-based security.

2.2.1 Algorithm Definition

Due to the nature of groups, negotiation of cryptographic algorithms is difficult, if not impossible. MSMP must define a common algorithm policy. This could be optimized to the environment.

Alternatively, the Internet Secure Association Key Management Protocol (ISAKMP now IKE) could negotiate the algorithm suite. If ISAKMP was able to completely cover the domain of the potential user base of the MSMP, then ISAKMP would be an adequate solution. The problem arises when a user tries to utilize MSMP without having benefit of a pre-negotiation by ISAKMP. MSMP would have to either negotiate the algorithm suite itself or cause ISAKMP to do so.

In summary, the MSMP should:

- Select cryptographic algorithms based on negotiated group policy, and
- Provide an interface for ISAKMP to establish the cryptographic environment.

2.2.2 Key Generation Definition

Two general mechanisms exist for the generation of cryptographic keys. A cooperative peer exchange [8] of key information can create a cryptographic key. Alternatively, a single entity can use a key generation technology to generate a key by itself.

The choice of which mechanism to use is determined by the security requirements of the group and the communication resources available to the MSMP. For each environment, a policy decision will have to be made. The

MSMP must support both mechanisms as directed by a policy decision.

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 11]

In summary, the MSMP must enforce key generation policy.

2.3 Authorization Definition

The definition of authorization mechanisms, and infrastructure, must be consistent across a MSMP domain. Due to the potential use of authorization tokens and certificates [9,10] for identity within the MSMP, the only way to make correct access control decisions would be to have a common authorization definition. It may be possible to define a mapping between authorization mechanisms to allow heterogeneous authorization infrastructures to interact. However, this mapping mechanism currently falls outside the scope of this security protocol.

In summary, the MSMP must:

- Enforce authorization policy, and
- Support a common authorization definition, and/or
- Support a common mapping of definition between authorization infrastructures.

2.3.1 Access Control

Access control, as it relates to security and key management, denies the key from entities without permission to hold the key. Historically, asymmetric key management protocols have implemented a policy of peer review. Peers cooperate to create the key. They "know" each others' identity based upon information passed. This identity information was previously certified by a trusted third party. Each peer "knew" that it wanted (or was allowed) to create a secure session with the other entity.

In the case of a multicast group, the peer-to-peer relationship for access control is impossible to implement. The number of messages required for every member in a group to identify, verify and perform access control for every other member group is prohibitive in terms of processing and bandwidth. Hence, the access control mechanisms must be different for multicast groups. In a multicast group, it is reasonable for the group owner to define the access control policy.

To summarize, the MSMP must:

- Support configuration of the access control policy,

- Distribute the access control policy to group, and

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 12]

- Verify access control.

Since the MSMP has to control access to the key, it performs the access control decisions. These access control decisions lay the very foundation for group security. There are two different philosophies: rules-based and identity-based access control.

2.3.1.1 Identity-Based

Identity-based access control decisions lend themselves to groups where all participants of the group are known in advance. These decisions are very clean and provide a high degree of assurance that only those group members listed have access to the data. This assumes, of course, that the mechanism for identifying group members is a strong one. Identity in this case can mean individual identities as defined by individual's certificates or it can refer to an IP address of a host machine.

Any identity-based access control policy requires that all access control decision makers have of the list of approved identities. The MSMP must provide a mechanism to disseminate not only the policy, but also the actual list of approved group members to all access control decision points.

2.3.1.2 Rule-Based

Rule-based access control [[9](#),[10](#)] relies on some set of preestablished parameters known about each potential member of the group. A certificate architecture infrastructure provides a framework to make rule-based access control decisions. The asymmetrical signed certificates, signed by a trusted entity, provide information about each individual.

An issue with rule-based access control is that the rule enforcement must be consistent across the entire group. This is easily accomplished if a single point is making all the access control decisions. However, with multiple access control decisions being made by multiple members of the group, the MSMP must provide a mechanism to disseminate the access control rules and access control policy.

2.3.2 Key Dissemination Architectures

Just as there are different levels of data, there are different levels of security (trust in the access control) that apply to a group. There is a natural trade-off between how fast the group can be established and the degree of assurance of the group. These two factors tend to oppose each

other in secure group management.

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 13]

In a small highly secure group, it may be desirable to have a single trusted authority or a small subset of trusted authorities to control access to the group key. This architecture leads to a very tightly controlled group. Such groups have a very difficult time scaling for a large group.

Access control requirements and control of the group, may be relaxed to allow some or all group members to disseminate the key based on the passing of some rudimentary access control rules. This would result in an increase in the speed of establishing an extremely large group.

In either instance, the host making access control decisions to the cryptographic key needs to be trusted to make those decisions. The definition of trust is up to the owners of the data. It could take the form of formal computer security trust levels or it could be defined locally.

In summary, the MSMP must:

- Support configurable key dissemination architectures and protocols, and
- Conform to computer security trust requirements imposed by the architecture.

[2.3.3](#) Trust

The mechanisms used to support and implement a MSMP must be ``trusted'' which means that the mechanisms are responsible for enforcing security and the level of security enforced by the system is dependent on the flawless execution of these resources. If the MSMP must enforce trust policies, it needs to be cognizant of the trust topology of its resources. If a sub-group of routers has the necessary trust mechanisms to protect keys, it is a candidate for a key dissemination protocol. However, this would impose a trust topology on the multicast internet. Use of these trusted routers would need management. The trust levels need monitoring (to verify the trusted state exists), and the list of trusted routers must be available to all entities that desire to create groups.

To summarize, the MSMP must:

- Enforce policy concerning data protection and computer security trust level;
- Maintain verification of the trusted state of "trusted" entities, in accordance with data protection accreditation; and
- Maintain state information for its domain in order to know ``who'' to trust.

[2.3.4](#) Authorization

Multicast groups require authorization of all important security actions. The multicast protocols must provide a mechanism where each group member can verify the identity of the entity asking it to perform important actions and check this identity against a pre-stored list of permissions.

In peer security protocols, the authorization mechanism is relatively simple. Each peer [6, 7] will make the decision to create a secure session with another peer based upon the IP address or user ID of the peer. Since there is direct communication between peers during secure association establishment, there is perfect knowledge of the identity of the communication partner.

In the case of MSMP, there is a requirement for a different authorization mechanism. Group members, in many instances, accept a key as valid without participating in the key's creation. There is a degree of trust on the part of the group member that the key is valid and does indeed belong to the group claimed.

A MSMP could fail if it does not have a full set of authorization mechanisms. The SMKD protocol [8] is designed for core base trees (CBT). The security protocol utilizes CBT routers to disseminate group keys. The CBT routers all undergo a mutual suspicious exchange verifying identities and authorization to receive the key. The group members strongly authenticate themselves to the CBT routers when they request a group key. However, the CBT routers do not strongly identify themselves to the group members. Nor do the new members have information from a trusted source authorizing the router to distribute the group key. In this protocol it is conceivable that a CBT router could become a rogue router. When the group member makes a request to join a group, the rouge router could give it a bogus key for that group and create an entire sub-group with this bogus key. It could trick members of the false group into communicating sensitive information on the bad key. In short, not having a robust authorization mechanism and utilizing the mechanism, could lead to masquerade attacks.

In summary, the MSMP must enforce authorization policy concerning group establishment, key dissemination, rekey and compromise recovery.

[2.3.5](#) Rekey Approach

Traditionally, a cryptographic key was treated as if it had a shelf life. More accurately, a cryptographic key is changed when too much data was protected by that single key. The most straightforward mechanism to achieve this changeover is to cancel the old group and create new group in its place containing all the old members. However, the creation of a cryptographic group, especially a large one, is an arduous task requiring a great deal of

access control decisions, messages, processing and processing resources.

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 15]

This is a time consuming process. In many cases, it is preferable to minimize the disruption of the communication group by sending out a single message that will change the group's key. This process is called rekeying a group.

When the rekey occurs, the single secure message containing the new group key is created. That message is transmitted to the group. Included in that message is some sort of cryptographic changeover time. This time is far enough into the future that most, if not all, of the group members are sure to receive the rekey message prior to changeover time. At that cryptographic changeover time, all group members will switch to the new cryptographic key for the group.

To allow for graceful transition between old and new group keys, there is usually a short period of time when either key decrypts messages. This allows messages that were in transmission, encrypted under the old group key, to be received at their destinations and decoded immediately after the cryptographic changeover time. However, all messages being sent after crypto-changeover time use the new key for encryption.

Usually, only large groups securing critical communications use rekey. The MSMP should support the concept of rekey particularly for critical groups that cannot withstand an interruption in service.

2.3.6 Compromise

For the purpose of this discussion:

- A compromise is the loss of trust in an entity with access to keys. This loss of trust (implies an assumption that the key has been exposed) invalidates the key.
- A compromise is not an administrative decision to remove or replace an entity with access to key. A loss of trust in that entity is not assumed. Administrative decisions do not necessarily imply that the key held by an entity is invalid.

The compromise of a secure group member is a more serious problem than the discovery of a compromised member for pairwise secure communication. In the case of pairwise communication, the secure association is deleted and no further action need take place.

If a group member is compromised, the compromised member needs deletion from the group, but at the same time the other group members need to be able to continue their communications without a disruption of service.

The seriousness associated with disruption of service and the urgency of removing a compromised member is a trade-off.

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 16]

There are several issues dealing with the handling of a compromised group member that could lead to many requirements on the MSMP. The general goal of dealing with a compromised group member is to return the group to a secure state. This compromised entity is denied access to future group information. Normally, one creates a separate group that includes all members of the original group minus the compromised member.

This imposes several management requirements on the security management protocol. The security management protocol must be able to either recognize the compromise of a group member or accept a report that a group member is compromised.

There are at least two separate means for dealing with compromises. One mechanism recently put forth [[10](#)] replaces the compromise recovery keys within the group. These keys split the group in such a manner that it would be easy to send a single message to multiple group members to get them on a new secure group transmission key. This mechanism would reduce the amount of time needed to reconstitute the secure group, after discovery of a compromise. However, this mechanism also requires management of these compromised recovery keys and the storage of compromise recovery by all the group members. Such a compromise recovery mechanism would be extremely valuable in the case of long-term static groups. This is especially true if the communications are critically important.

Another compromise recovery mechanism is simply to cancel the compromised group and create a new group that is exactly equal to the old, minus the compromised member. This mechanism has simplicity on its side, but certainly is slower and causes more disruption to the group communications.

In short, the MSMP must enforce compromise recovery policy as defined at group establishment.

[2.4](#) Protocol Requirements

The multicast security protocol has requirements levied upon it based more in the good design of a protocol rather than focused on the security aspects of the protocol. The following sections attempt to catalog these design goals.

[2.4.1](#) Self Defined

The MSMP should provide a complete tool set for the management of keys and security for cryptographic groups. It should generate and contain all the information the protocol needs to function. The one possible exception could be the certificate's infrastructure, if one is needed. In the case of

the certificate infrastructure, a very good case exists for the utilization of existing infrastructures rather than trying to reinvent it. The MSMP

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 17]

must:

- Provide mechanisms to allow group-wide enforcement of group policy, and
- Support existing certificate infrastructures.

2.4.2 Communications Protocol Independent

The MSMP should be independent of the communication system it is being transmitted over and any protocol that it might be servicing. A majority of the work done in this area has been under the auspices of the IPSEC Working Group. However, the MSMP not only services IP layer security, but will also serve session and application layer security. The MSMP will also reside on hosts serviced by heterogeneous communication protocols. As an application protocol itself, MSMP should be completely divorced from the nature of the communication.

The MSMP should not target a specific communication protocol. However, that does not mean that an option under the MSMP cannot target a specific communication environment. For example, the general protocol could offer an option for those systems that operate solely over ATM or CBT networks. These homogeneous networks offer distinct advantages for a security management protocol. A security management protocol could utilize a trusted backbone of routers [8] to either set groups up more quickly or to ease the recovery from a compromise. The MSMP should offer mechanisms that allow customized protocols.

It is also important to realize that different cryptographic groups, depending on their utilization, have different requirements and natures. For instance, a large IP network may have the luxury to limit the number of endpoints with identical keys, thereby limiting the scope of a compromise. In other systems (e.g. cable system or especially those utilizing satellite downfeeds), there is no capability to limit the scope of compromised keys by limiting the size of key groups. The MSMP must:

- Remain independent of any specific communication protocol or infrastructure;
- Support operation as a source to destination protocol; and
- Support homogeneous systems with optimized solutions.

2.4.3 Architecture Independent

The MSMP should provide multicast security management regardless of the environment it is serving. This protocol should satisfy at least 95 percent of the security architectures that require secure keys.

For example, the MSMP should be able to support networks that push a group key onto the end points and where the end points pull the group. A large group could be built up of multiple cooperatives or it could simply be a large commonly held group of symmetrical keys. Again, the MSMP should be able to satisfy both cases. The MSMP should be configurable to support extremely high security groups, even though they incur a degradation in terms of speed. Conversely, it should be configurable to support groups that trade high security for speed and ease of group establishment.

Obviously, a single scheme for creating secure groups and distributing keys to those end points will not be adequate to satisfy all the different architectures and environments the MSMP will be supporting. A single, universally accepted, protocol construct is required that allows access to sub-protocols optimized for different environments.

3 POLICY COMPONENTS

Security mechanisms and security protocols all enforce some policy or policies within their domain. A clear definition of the enforced policies is critical to the successful design and implementation of a security protocol. The following section attempts to define policies that are being enforced by the MSMP.

3.1 Security Policy

Security policy is a statement of the rules enforced by security mechanisms. There are multiple rules the MSMP will be able to enforce. In a dynamic system, groups define these policies based on the data that particular group will protect.

The security policy can be static, and therefore assumed, or it can be dynamic and tailored to the requirements of the group. A dynamic security policy would allow the group owner to identify one or several key locations as well as authorizing new group members as needed. If MSMP has a dynamic security policy, a mechanism must define and disseminate this policy across the group. The MSMP must understand the policy and verify the authorization of that policy.

A group security policy will make statements about the key the group will

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 19]

share. For example, it is reasonable to see a policy that identifies a key for financial data. The MSMP must implement this policy across the group uniformly.

3.2 Architecture

The more interesting policies MSMP will enforce involve the structure of the group itself. The MSMP will enforce policy roles, key distribution behaviors, access control, rekey, and compromise recovery.

3.2.1 Operational Policy

All of the proposed multicast security protocols [[1](#), [2](#), [8](#)] assumed a structure of the key management protocol itself. A single entity creates the key and makes it available for dissemination to group members. The various proposals disagree about key dissemination, if routers are used to make access control decisions, and how access control decisions are decided.

There is no reason that the MSMP need operate the exact same way as it creates keys for different groups in different environments. A mechanism that conveys to the MSMP the operational policies will facilitate a more dynamic protocol.

3.2.2 Key Dissemination Policy

Another group policy is key dissemination. A single entity may create the keys, but the key can be disseminated to the group members in several manners. One key dissemination policy could be that a single trusted entity performs all key dissemination and associated access control decisions. This single point to dissemination policy is not performance oriented and may not be acceptable for larger groups. Another policy is to delegate responsibility for key dissemination to a subset of routers. This policy assumes trusted routers. The trusted routers must protect the key and make access control decisions in accordance with the sensitivity of the data been protected. Yet another policy, is that any group member disseminates the group key to any potential group member that meets a certain set of criteria.

The particular policy for key dissemination is highly dependent on the sensitivity of the data to be protected. Depending on the data being protected, the same application could have a very different trust requirement placed on the dissemination of key. The MSMP could change its dissemination mechanisms or indeed its utilization of sub-protocols based on

a policy statement about key dissemination and trust requirements.

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 20]

3.2.3 Access Control Policy

Perhaps the most critical policy definition of the group is that of access control. The access control policy defines the user or host that have access to the cryptographic key. This policy can be identity- or rule-based or a mixture of both. In any case, the access control policy must be unambiguously stated so that only authorized group members receive the key.

There are several ways to define access control policy. It can be based on a human identity, IP address, permission parameters, job title, or company name. The requirement is that the parameter be unambiguous and verifiable. The most common mechanism is a certificate. The information in a certificate supports an access control decision because a trusted third party verifies the accuracy of that information.

The MSMP could operate between multiple certificate infrastructures providing there is a policy that clearly stated the acceptable certificate parameters in each infrastructure. In short, the access control policy states who should have access to the keys and the mechanisms used to prove that.

3.2.4 Rekey Policy

As described in earlier sections, a rekey is a useful action when a cryptographic key is of long duration or is protecting a great amount of data. The decision to rekey is appropriate for any particular group and the mechanism that rekey will utilize is the rekey policy.

Rekey involves the creation of the new group key and the creation of a globally acceptable message to disseminate that key to all the current group members. A single group entity needs to coordinate this process. After all there can only be one valid group key at a time. The rekey policy would need to state clearly the individual authorized to perform the rekey, the time of the rekey, and the time allotted for graceful key changeover.

3.2.5 Compromise Policy

Compromise recovery policy involves several decisions. There is the decision whether to pre-place a compromise recovery key hierarchy or to delete and rebuild the group. Another decision, is who has the authority to declare the group compromised and how was that decision communicated to the group.

Perhaps the most difficult part about compromise recovery is discovering the compromise. The rules for discovering a compromise and reporting it are

beyond the scope of this security protocol. However, the MSMP will need to

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 21]

have the capability to accept notification that the group is compromised. How that notification is communicated or utilized by the MSMP is a policy decision. Compromise recovery key structures can be pre-placed in a secure group along with the normal group encryption keys. However, the MSMP must define the nature of the key structures' needs and pass it to the group at the time of group establishment.

4 DESIGN RECOMMENDATIONS

The analysis and review of the MSMP requirements and policies have resulted in two recommendations regarding the direction of the multicast security effort. The first recommendation is to create a globally acceptable policy mechanism that is accepted across the environments and would completely define the cryptographic group. The second recommendation deals with the design and implementation of the MSMP.

4.1 Global Policy Mechanism

One thing that became clear during the analysis of multicast group requirements is that there are many policy decisions involved with group establishment. The multicast environment, unlike the pairwise environment where a peer-to-peer negotiation is uncomplicated, requires more coordination between the group members. Certainly, a single group member can make the cryptographic key. There are multiple ways the MSMP could disseminate cryptographic key to the group. There is the issue of whether or not the group needs a rekey and, if it does, how to orchestrate the rekey. There is the whole issue of compromise recovery orchestration. Many of these decisions are highly dependent upon the sensitivity of the data, the duration of the group, and the criticality of the communication.

There is a strong argument for each of these options. The MSMP should be capable of being configured to satisfy most environmental requirements. Because the entire group needs a common policy and group definition, it makes sense for a single mechanism to provide this information. It would be best if this policy definition mechanism performed all MSMP configuration actions. Hence, one recommended goal for the MSMP is that a single mechanism is defined that will inform the MSMP of the group policy.

4.2 Limited Group of Security Mechanisms

The following recommendation deals with the protocol design and implementation. A small subset of security protocols should be designed and optimized for specific practical environments. These specialized protocols

come from a generally accepted group specification message.

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 22]

This recommendation suggests a highly modularized MSMP with a small, fully optimized, sub-protocol. There are benefits to doing this, including having a universally accepted definition of multicast groups. The end points could then either participate in a group (providing possession of the optimized sub-protocol), go get the appropriate sub-protocol, or not join the group. End systems could load those modules that are relevant to them and ignore all the others. This leads to a protocol structure that works efficiently for specific environments, provides universal protocol recognition, and allows conservation of user resources.

From the point of view of developing an international standard, the modularized approach leads to a highly useful and efficient standard and protocol. A high degree of interoperability exists due to the universally accepted group definition. Each environment could have a the MSMP targeted for that environment. Homogeneous environments could use CBT routers or intermediate routers to distribute to key. Heterogeneous environments could have the end systems generate group keys without the knowledge of the routers. Extremely large unicast networks could utilize unique communication infrastructures like group set-up servers. Extremely high security systems could include a compromise recovery key structure.

4.3 Policy Decomposition Of Multicast Protocols

The following table illustrates how some multicast protocols would decompose into the policy components previously identified. Each protocol makes different assumptions of it's environment and those assumptions lead to different policies. Yet, these policies can be represented using the same decomposition format.

GKMP

Operational - Certificate Infrastructure ID, Group Controller IP address: a.b.c, Group 1st member IP address: a.b.c.d, Group Owner Common Name: X,

Dissemination - Group Controller only (push or pull) or any member

Access Control - Mutual suspicious, IP address list or Rules: IP a.b.*, Common Name: *.acme.com

Rekey - Token required, uses GKEK sent during group establishment

Compromise Recovery - Destroy group, create new, with certificate revocation capability during establishment

SMKD

Operational - CBT routers relays key, routers undergo rigorous

authentication

Harney/Harder

[draft-harney-sparta-msmp-sec-00.txt](#)

[Page 23]

Dissemination - Download from responding CBT router

Access Control - Host sends signature to router

Rekey - NA

Compromise Recovery - Destroy group, create new

The following documents were used in the preparation of this document:

References

- [1] [[RFC 2093](#)] Harney H., Muckenhirn C., and Rivers T., Group Key, Management Protocol Specification, [RFC 2093](#), Experimental, July 1997.
- [2] [[RFC 2094](#)] Harney H., Muckenhirn C., and Rivers T., Group Key Management Protocol Architecture, [RFC 2094](#), Experimental, July 1997.
- [3] [[RFC 2408](#)] Maughan D., Schertler M., Schneider M., and Turner J., Internet Security Association and Key Management Protocol (ISAKMP), [RFC 2408](#), Proposed Standard, November 1998.
- [4] [[RFC 2412](#)] Orman H. K., The OAKLEY Key Determination Protocol, [RFC 2412](#), Informational, November 1998.
- [5] [[RFC 2409](#)] Harkins D., and Carrel D., The Internet Key Exchange (IKE), [RFC 2409](#), Proposed Standard, November 1998.
- [6] SDNS Protocol and Signaling Working Group, SP3 Sub-Group, SDNS Secure Data Network System, Security Protocol 3 (SP3) Addendum 1, Cooperating Families, SDN.301.1, Rev. 1.2, 1988-07-12.
- [7] SDNS Protocol and Signaling Working Group, SP3 Sub-Group, SDNS Secure Data Network System, Security Protocol 3 (SP3), SDN.301, Rev. 1.5, 1989-05-15.
- [8] [[RFC 1949](#)] Ballardie, A., Scalable Multicast Key Distribution, [RFC 1949](#), Experimental, May 1996.
- [9] [[RFC 2459](#)] Housley R., Ford W., Polk T., and Solo D., Internet X.509 Public Key Infrastructure Certificate and CRL Profile, [RFC 2450](#), Proposed Standard, January 1999.
- [10] Wallner, D., Harder E., and Agee R., Key Management for Multicast: Issues and Architectures, Internet Draft, Informational, September 1998.

5 Addresses of Authors

Hugh Harney (point-of-contact)
SPARTA, Inc.
Secure Systems Engineering Division
9861 Broken Land Parkway, Suite 300
Columbia, MD 21046-1170
United States
telephone: +1 410 381 9400 (ext. 203)
electronic mail: hh@columbia.sparta.com

Eric J. Harder
R231 National Security Agency
9800 Savage Road
Suite 6534
Fort Meade, MD 20755
United States
telephone: +1 301 688 0847
electronic mail: ejh@tycho.ncsc.mil

Document expiration: August 30, 1999

