

Internet Research Task Force
INTERNET-DRAFT
[draft-irtf-smug-framework-01.txt](#)
September 2000

Thomas Hardjono (Nortel)
Ran Canetti (IBM Watson)
Mark Baugher (PassEdge)
Peter Dinsmore (NAI)
Expires March 2001

Secure IP Multicast: Problem areas, Framework, and Building Blocks

<[draft-irtf-smug-framework-01.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 foundation for the research work done as the Secure Multicast Group (SMuG) of the IRTF. The document begins by introducing a Reference Framework and problem areas, and proceeds to identify functional building blocks for a secure multicast solution. The identified building blocks, their definition and realization will be the subject of future work of SmuG.

1. Introduction

Securing IP multicast communication is a complex task that involves many aspects. Consequently, a secure IP multicast protocol suite must have a number of components that address different aspects of the problem.

Internet-Draft

September 2000

This document proposes a reference framework for secure IP multicast protocol suites and defines a breakdown to functional building-blocks for such protocol suites. The proposal is a product of discussions at the Secure Multicast Group (SmuG) of the IRTF, and is addressed at the SmuG group as a basis for further discussion. In particular, this document may serve as a basis for considering the issues that have been addressed so far in the SMuG IRTF community, the issues that have yet to be addressed, the issues that need further in-depth attention, and technologies that may be ready for consideration by an IETF working group.

Sections [2](#) and [3](#) present the problem areas, reference framework and briefly discusses the functional entities, and interfaces, which are identified by the framework. [Section 4](#) presents the breakdown of the interfaces to individual functional building blocks. [Section 5](#) presents the conclusion.

[2](#). Problem Areas in Secure Multicast

In order to begin to address the problems in securing IP multicast, we identify three problem-areas and define a reference framework expressing these problem areas. The three problem-areas are:

Problem-Area-1: Multicast data handling. This area covers problems concerning the security-related treatments of multicast data by the sender and the receiver. This problem-area is further discussed in [Section 2.1](#).

Problem-Area-2: Management of keying material. This area is concerned with the secure distribution and refreshment of keying material. This problem-area is further discussed in [Section 2.2](#).

Problem-Area-3: Multicast security policies. This area covers aspects of policy in the context of multicast security, taking into consideration the fact that policies may be expressed in different ways, that they may exist at different levels in a given multicast security architecture and that they may be interpreted differently according to the context in which they are specified and implemented. This problem area is further discussed in [Section 2.3](#).

[2.1](#) Multicast Data Handling (Problem-Area-1)

In a secure multicast group the data typically needs to be:

1. Encrypted using the group key, mainly for access control and possibly also for confidentiality.
2. Authenticated, for verifying the source and integrity of the data. Authentication takes two flavors:

Secure IP Multicast

[Page 2]

Internet-Draft

September 2000

2.1 Source authentication and data integrity.

This functionality guarantees that the data originated with the claimed source and was not modified en route (either by a group member or an external attacker).

- ### 2.2 Group authentication.
- This type of authentication only guarantees that the data was generated (or last modified) by some group member. It does not guarantee data integrity unless all group members are trusted.

While multicast encryption and group authentication are fairly standard and similar to encrypting and authenticating point-to-point communication, source authentication for multicast is considerably more involved. Consequently, off-the-shelf solutions (e.g., taken from IPSec [[RFC2406](#)], TLS [[TLS](#)]) may be sufficient for encryption. For source authentication, however, special-purpose transformations are necessary. See [[CP99](#)] for further elaboration on the concerns regarding the data transforms, on present solutions and remaining challenges.

[2.2](#) Management of Keying Material (Problem-Area-2)

The term "keying material" refers to the cryptographic key belonging to a group, the state associated with the keys and the other security parameters related to the keys. Hence, the management of the cryptographic keys belonging to a group necessarily requires the management of their associated state and parameters. A number of solutions for specific problems must be addressed. These may include the following:

- Methods for member identification and authentication.
- Methods to verify the membership to groups.

- Methods to establish a secure channel between a KS+GC entity and the member, for the purpose of delivery of shorter-term keying material pertaining to a group.
- Methods to establish a long-term secure channel between one KS+GC entity and another, for the purpose of distributing shorter-term keying material pertaining to a group.
- Methods to effect the changing of keys and keying material
- Methods to detect and signal failures and perceived compromises to keys and keying material

The needs related to the management of keying material must be seen in the context of the policies that prevail within the given circumstance.

[2.3](#) Multicast Security Policies (Problem-Area-3)

Multicast Security Policies must provide the rules for operation for the other elements of the Reference Framework. While much of the work for the Multicast Security Policy area is focused in the Policy Controller, there are potential areas for work in the application of policy at the Group Controller element and the member (sender and receiver) elements. While there is already a basis for security policy management in the IETF between the Policy Working Group and the IP Security Policy Working Group, multicast security policy management will extend the concepts developed for unicast communication in the areas of:

- Policy creation,
- High-level policy translation, and
- Policy representation.

Examples of work in multicast security policies include the Dynamic Cryptographic Context Management project [[Din](#)], Group Key Management Protocol [[Har](#)], and Antigone[[McD](#)].

Policy creation for secure multicast has several more dimensions than the single administrator specified policy assumed in the existing unicast policy frameworks. Secure multicast groups are usually large and by their very nature extend over several administrative domains, if not spanning a different domain for

each user. There are several methods that need to be explored for the creation of a single, coherent group security policy. They include a top-down specification of the group policy from the group initiator and negotiation of the policy between the group members (or prospective members). Negotiation can be as simple as a strict intersection of the policies of the members or extremely complicated using weighted voting systems.

High-level policy translation is much more difficult in a multicast group environment, especially when group membership spans multiple administrative domains. When policies are specified at a high level with a Policy Management tool, they must then be translated into more precise rules that the available security mechanisms can both understand and implement. When dealing with multicast communication and its multiple participants, it is essential that the individual translation performed for each participant result in the use of a mechanism that is interoperable with the results of all of the other translations. Typically, the translation from high-level policy to implementation mechanisms must result in the same mechanism in order to achieve communication between all of the group members. The requirement that policy translation results in the same mechanism places constraints on the use and representations in the high-level policies. It is also important that policy

negotiation and translation be performed as an integral part of joining a group. Adding a member to a group is meaningless if they will not be able to participate in the group communications.

Multicast security policies must represent, or contain, more information than a traditional peer-to-peer policy. In addition to representing the security mechanisms for the group communication, the policy must also represent the rules for the governance of the secure group. Policy must be established for the basic group operations of add and remove, as well as more advanced operations such as leave, rejoin, or resync.

3. Problem Scope and Reference Framework

This section considers the complex problems of multicast security in the context of a heuristic device, the Reference Framework diagram, shown in Figure 1. The Reference Framework is used to classify problem areas, functional elements, and interfaces. The Reference Framework defines

the building blocks and suggested program of work for research and standardization, which is presented in a later section of this paper.

[3.1](#) A Reference Framework

Based on the three broad problem-areas, a reference framework is proposed (Figure 1). The reference framework attempts to incorporate the main entities and functions relating to multicast security, and to depict the inter-relations among them. At the same time it also tries to express the complex multicast security question from the perspective of problem classification (i.e., the three problem areas), from the perspective of architectures (centralized and distributed), of multicast types (1-to-M or M-to-N), and protocols (the exchanged messages).

The aim of the reference framework is to provide some general context within which problems can be identified and classified (as being under a given problem-area) and the relationships among the problems can be recognized. Note that some issues span more than one so-called problem-area. In fact, the framework encourages the precise identification and formulation of issues that involve more than one problem-area or those which are difficult to express in terms of a single problem area. An example of such a case is the expression of policies concerning group keys, which involves both the problem-areas of group key management and multicast policies.

When considering the reference framework (Figure 1) it is important to realize that the singular "boxes" in the framework do not necessarily imply a corresponding singular entity implementing a given function.

Rather, a box in the framework should be interpreted loosely as pertaining to a given function related to a problem-area. Whether that function is in reality implemented as one or more physical entities is dependent on the particular solution. As an example, the box labelled "Key Server" must be interpreted in broad terms as referring to the functions of key management. Similarly, the Reference Framework acknowledges that some implementations may in fact merge a number of the "boxes" into a single physical entity.

The reference framework can be viewed horizontally and vertically. Horizontally, it displays both the entities and functions as singular boxes, expressing each of the three broad problem-areas. Vertically,

it expresses the basic architecture designs for solutions, namely a centralized architecture and a distributed architecture.

The protocols to be standardized are depicted in Figure 1 by the arrows that connect the various boxes. See more details in [Section 4](#), below.

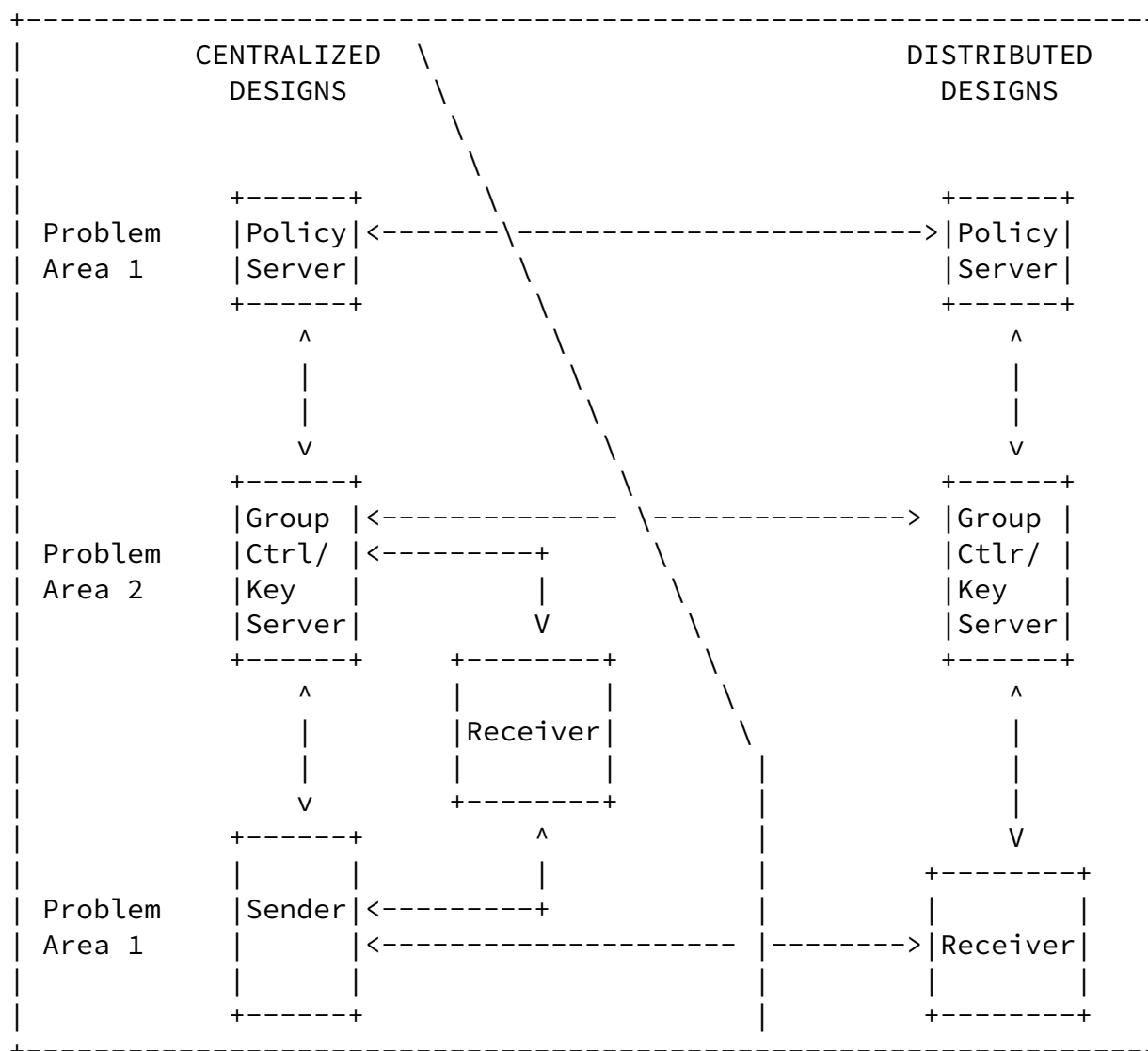


FIGURE 1: PROPOSED REFERENCE FRAMEWORK FOR SECURE MULTICAST

[3.2](#) Elements of the Reference Framework

The Reference Framework diagram of Figure 1 contains boxes and arrows. The boxes are the functional entities and the arrows are the interfaces between them. Standard protocols are needed for the interfaces, which

support the multicast services between the functional entities. There are three sets of functional entities in both centralized and distributed designs as discussed below.

[3.2.1](#) Key Server and Group Controller

The Key Server and Group Controller (KS+GC) represent both the entity and functions relating to the issuance and management of cryptographic keys used by a multicast group, which is subject to the user-authentication and authorization checks conducted on the candidate member of the multicast group.

In a distributed architecture the KS+GC entity also interacts with other KS+GC entities to achieve scalability in the key management related services. In such a case, each member of a multicast group may interact with one or more KS+GC entity (say, the "nearest" KS+GC entity, measured in terms of a well-defined and consistent metric). Similarly, in a distributed architecture a KS+GC entity may interact with one or more Policy Servers, also arranged in a distributed architecture.

We remark that the Key Server (KS) and the Group Controller (GC) have somewhat different functionality and may in principle be regarded as separate entities. Currently the framework regards the two entities as one "box" in order to simplify the design, and in order not to mandate standardization of the protocol between the KS and the GC. It is stressed that the KS and GC need NOT be co-located. Furthermore, future designs may choose to standardize the protocol between the GC and the KS, without altering other components.

[3.2.2](#) Sender and Receiver

The Sender is an entity that sends data to the multicast group. In a 1-to-N multicast group only a single sender is allowed to transmit data to the group. In an M-to-N multicast group many (or even all) group members can transmit data to the group.

Both Sender and Receiver must interact with the KS+GC entity for the purpose of key management. This includes user-authentication, the obtaining of keying material in accordance with some key management policies for the group, obtaining new keys during key-updates, and obtaining other messages relating to the management of keying material and security parameters.

The influence of policies on both Senders and Receivers is seen as coming indirectly through the KS+GC entities, since the event of joining a multicast group is typically coupled with the Sender/Receiver obtaining keying material from a KS+GC entity. This does not preclude the direct interaction between the Sender/Receiver and the Policy Server.

The reference framework displays two Receiver boxes corresponding to the situation where both the Sender and Receiver employ the same KS+GC entity (centralized architecture) and where the Sender and Receiver employ different KS+GC entities (distributed architecture).

[3.2.3](#) Policy Server

The Policy Server represents both the entity and functions used to create and manage security policies specific to a multicast group. The Policy Server interacts with the KS+GC entity in order to install and manage the security policies related to the membership of a given multicast group and those related to keying material for a multicast group.

The interactions between the Policy Server and other entities in the reference framework is dependent to a large extent on the security circumstances being addressed by a given policy.

[3.2.4](#) Centralized and Distributed Designs

The need for solutions to be scalable to large groups across wide geographic regions of the Internet requires the elements of the framework to also function as a distributed system. This implies that a KS+GC entity must be able to interact securely with other KS+GC entities in a different location. Similarly, Policy Servers must interact with each other securely to allow the communication and enforcement of policies across the Internet.

[4.](#) Building Blocks

This section breaks down the secure multicast problem to functional building blocks. The breakdown uses the Reference Framework presented in [Section 2](#) to identify high-level building blocks.

A "multicast security building block" provides a multicast security service, such as multicast data confidentiality, or it provides a service to multicast security, such as an application programming interface (API). Specifications for an API or multicast confidentiality protocol, however, are building blocks that can be directly

Internet-Draft

September 2000

implemented. This paper describes "functional building blocks," which are at a higher level of abstraction than algorithm or protocol specifications. Our goal is to identify functional building blocks that will focus work on algorithms, protocols, and other specifications that can be implemented to solve multicast security problems. In our approach, a functional building block is mapped to specific boxes or interfaces in the Reference Diagram of Figure 1. This section identifies which functions are performed where in Figure 1 without defining a specific realization of a functional building block.

As an example, an algorithm such as Diffie-Hellman key exchange is one possible realization of a key management functional building block. An algorithmic building block is at a lower level of abstraction than a functional building block. Protocols or APIs that can be directly implemented in computer hardware or software may in turn, realize algorithms. The Internet Key Exchange [[RFC2409](#)], for example, realizes Diffie-Hellman and ISAKMP [[RFC2408](#)]; the TLS Handshake Protocol [[RFC2246](#)] realizes Diffie-Hellman and RSA.

Thus, our treatment of functional building blocks for multicast security is relatively abstract and will only be of practical use when functional building blocks are realized as algorithms and protocols. We propose to focus the work on multicast security algorithms and protocols along the lines of functional building blocks as a problem-solving strategy. [Section 4.1](#) motivates the building blocks approach to multicast security by listing three reasons for using this approach followed by three criteria by which to evaluate multicast security building blocks. [Section 4.2](#) considers some functional building blocks for multicast security.

[4.1](#). Motivation

A common approach to solving a complex problem is to subdivide the problem into manageable "blocks". Here, each block must serve a well-defined function and its relationship with other blocks must be clearly defined. Besides being more manageable, the approach inherently has a number of advantages including use and re-use of the functional block independently of the whole, and the ability to combine different blocks to satisfy multiple functions. There are a number of risks to the building blocks approach [[RMT](#)]:

1. Delayed development, which results from the need for additional work to develop ways to combine independent building blocks.
2. Increased complexity, which is caused by too many building blocks having too many interfaces.
3. Reduced performance, which may be caused by too much modularization.

4. Abandonment of prior work, which results from attempts to develop robust, general solutions.

These risks were identified for the multicast file transfer work. The fourth risk addresses an historic fact in the development of multicast file transfer that is not true for multicast security. It may be true, however, that an attempt to develop general solutions for diverse requirements, such as IP multicast and application-layer multicast security requirements, may retard the development of particular solutions. Although we have restricted our attention to functional building blocks in this paper, as described in [section 4.0](#), there may be algorithms that solve common problems in both application-layer multicast and IP multicast. The IETF IPsec working group, moreover, has already produced a key management protocol that aims to provide for both application-layer multicast and IP multicast services, if properly extended [[RFC2408](#)].

Despite the above-mentioned risks, there are at least three important benefits to multicast security in applying the building blocks approach.

1. Re-use of publicly-reviewed cryptographic protocols: Even the best cryptographic mechanism can be effectively attacked at the protocol level [Schneier p. 473], so we seek to benefit from the re-use of proven technologies, which is inherent in a building-blocks approach to cryptographic protocols.
2. Timely delivery of needed technology: Using multicast building blocks, we hope to standardize specific technology, such as multicast confidentiality, independently of other security services that may take longer to specify for a great variety of uses, such as multicast source and data authentication.
3. Robust support for a variety of application environments: Good building block definitions will permit the combination of individual building blocks to flexibly add security services to

IP multicast or application-layer multicast traffic.

4. Simplicity in the proof of correctness and working of each building block as a separate block.

To make the notion of building blocks more concrete, consider the example of the independence of protocols in the IPsec suite. Certain IPsec protocols such as AH [[RFC2402](#)] and ESP [[RFC2406](#)] perform their security functions independently of other protocols in the IPsec suite, such as Internet Key Exchange (IKE), which provides security association [[RFC2401](#)] and key management services to AH and ESP. As a result of this independence, a compliant ESP implementation can be used today to provide IP multicast confidentiality despite the fact that an IKE security association (SA) is unique to a pair of communicating

endpoints and is unsuitable for managing multicast group keys. If ESP uses an IPsec SA having a multicast address, however, it effectively supports IP multicast confidentiality since there is no requirement that an SA used by ESP be established by IKE (though this might be a reasonable policy for some environments). Thus, other applications, such as a future ISAKMP variant may be used to establish the keying material needed for an IP multicast ESP service. For this to work, however, an application-programming interface may be needed for updates to the host security association database - an API such as PF_KEY [[RFC2367](#)] may serve this purpose although the definition of an SA may need to be extended for multicast [[HH](#)]. Taken together, ESP and PF_KEY can provide a useful multicast security service - IP multicast confidentiality. The capacity to provide a useful security service is one important criterion for a multicast security building block, which is realized in an algorithm, protocol, API, or by other means. In cases where authentication of multicast packet sources is required, however, ESP is probably unsuitable: ESP source and data authentication use a symmetric key, which is a shared secret among all members of the particular multicast group [[RFC2403](#), [RFC2404](#)]. An alternative approach of using an asymmetric signature algorithm [[RFC2406](#)] would generally be too slow for real-time multicast flows.

The mechanisms used in ESP for authentication and integrity, therefore, will not authenticate individual senders to a multicast group. Just as some applications may need only a subset of multicast security services, others will require a "Whole Protocol" [[RMT](#)]. A multicast security building block should be able to be combined with other building blocks to provide additional security services. Without this

property, the building block is little more than an incomplete solution to the general problem. Thus a second criterion for a good multicast security building block is that it can be combined with other building blocks to provide additional security services. A good building block for IP multicast confidentiality can be combined with other building blocks for IP multicast source authentication, data authentication (integrity), and additional security services.

A good example of the building blocks approach is the work being done on multicast packet-level source and data authentication within the SMuG community. One output of this effort is a draft specification on multicast packet-level authentication for RTP applications [[McCarthy](#)], which is a proposed RTP Profile [[RFC1889](#)]. The "RTP Profile for Source Authentication and Non-Repudiation of Audio and Video Conferences," proposes to efficiently authenticate the sender and verify the integrity of multicast packets by applying a digital signature over the hash of a sequence of packets [[Wong](#), [McCarthy](#)]. Although practical experience is needed to evaluate this protocol, it illustrates the use of a multicast security building block. A successful multicast source or data-packet authentication building block should be applicable to other applications such as SDP/SAP [[RFC2327](#), [SAP](#)]. Indeed, technology

that solves source and data-packet authentication for real-time multicast application traffic should be considered for IP multicast traffic as well - at least the algorithm, if not the protocol. Thus, a third criterion for a multicast security building block is its applicability to IP multicast and application-layer multicast security.

The preceding discussion has established a set of three criteria for good multicast security building blocks.

1. A building block provides a flexible security service. A protocol that realizes a building block should be standardizable independently of other building blocks.
2. Building blocks can be combined in a framework to provide a set of multicast security services that amount to a "Whole Protocol" for multicast security.
3. Building blocks can be applied to both IP multicast and application-layer multicast security; good multicast security building blocks can be adapted for both protocol and data security.

As discussed above, useful solutions may not satisfy all three criteria, but we expect that the most promising proposals for standardization would satisfy more than a single criterion. Thus, our criteria are suggested as measures of goodness for a functional or protocol building block.

In addition to the demands of productive use and standardization, the building blocks approach allows the identification of certain problems that are still ill understood and thus ill defined. In the context of the SMuG IRTF group, we expect that the building blocks approach will help focus our research mission and facilitate our standards objectives. By adopting this approach early, SMuG can avoid the near-impossible task of extracting building blocks from mature protocols and can positively influence multicast standards work that may occur in IETF working groups.

The building blocks approach also allows the sharing of standardized technologies with working groups within the IRTF and the IETF. For example, certain blocks developed with the SMuG IRTF group may be useful and deployable by the Reliable Multicast Transport (RMT) Working Group in their efforts to secure the RMT protocols. The same blocks may also be used to secure other application protocols (e.g., RTP), multicast routing protocols, and be applied to other areas where both multicast and security services are needed.

[4.2](#). Functional Building Blocks

Referring to our Reference Diagram, this section identifies functional building blocks for designated interfaces of Figure 1. In this section, distinct functional building blocks are assigned to specific interfaces. For example, multicast source authentication, data authentication, and confidentiality occur on the multicast data interface between Senders and Receivers in Figure 1. Authentication and confidentiality services may also be needed between the Key Server and key clients (i.e., the Senders and Receivers of Figure 1), but the services that are needed for multicast key management may be unicast as well as multicast. Multicast Key Management is a separate function and has a separate building block. A functional building block for multicast security, therefore, identifies a specific function

along one or more Figure 1 interfaces.

This paper does not attempt to analyze the trust relationships, detailed functional requirements, performance requirements, suitable algorithms, and protocol specifications for IP multicast and application-layer multicast security. Instead, we propose these tasks as future work that will occur as the functional building blocks are further defined and realized in algorithms and protocols.

We identify a set functional building blocks in the following sections. This preliminary list of building blocks is intended to serve as a basis for discussion at the SMuG working group. We anticipate that work done on the several high-level, functional building blocks described below will lead to the specification of lower-level building blocks that can be implemented to provide needed multicast security services.

[4.2.1](#) Multicast Data Confidentiality

This functional building block handles the encryption of multicast data at the Sender's end and the decryption at the Receiver's end. This building block presumably may apply the keying material that is provided by Multicast Key Management in accordance with Multicast Policy Management, but it is independent of both.

An important part of the future work on the Multicast Data Confidentiality building block is in the identification of and motivation for specific ciphers that should be used for multicast data. Obviously, not all ciphers will be suitable for IP multicast and application-layer multicast traffic. Since this traffic will usually be connectionless UDP flows, stream ciphers may be unsuitable though hybrid stream/block ciphers may have advantages over some block ciphers. Those working on this functional building block will need to evaluate the real-time and other requirements of multicast senders and receivers, and recommend a small set of promising ciphers and data

protocols for IP multicast and application-layer multicast data confidentiality.

Regarding application-layer multicast, some consideration is needed to the effects of sending encrypted data in a multicast environment lacking admission-control, where practically any application program can join a multicast event independently of its participation in a

multicast security protocol. Thus, this building block is also concerned with the effects of multicast confidentiality services, intended and otherwise, on application programs in all senders and receivers.

In Figure 1, the Multicast Data Confidentiality building block is placed in Problem Area 1 along the interface between Senders and Receivers. The algorithms and protocols that are realized from work on this building block may be applied to other interfaces and Problem Areas of Figure 1 when multicast data confidentiality is needed.

[4.2.2](#) Multicast Source Authentication and Data Integrity

This building block handles source authentication and integrity verification of multicast data. It includes the transforms to be made both at the Sender's end and at the Receiver's end. It assumes that the appropriate signature and verification keys are provided via Multicast Key Management in accordance with Multicast Policy Management as described below. Work done by members of the SMuG community suggests that this is one of the harder areas of multicast security based on the connectionless and real-time requirements of many IP multicast applications. There are classes of application-layer multicast security, however, where offline source and data authentication will suffice. As discussed in [section 4.1](#), not all multicast applications require real-time authentication and data-packet integrity. A robust solution to multicast source and data authentication, however, is necessary for a Whole Protocol solution to multicast security.

In Figure 1, the Multicast Source and Data Authentication building block is placed in Problem Area 1 along the interface between Senders and Receivers. The algorithms and protocols that are produced for this functional building block may have applicability to building blocks in other Problem Areas that use multicast services such as Multicast Key Management.

[4.2.3](#). Multicast Group Authentication

This building block provides a limited amount of authenticity of the transmitted data: It only guarantees that the data originated with (or was last modified by) one of the group members. It does not guarantee

authenticity of the data in case that other group members are not trusted.

The advantage of group authentication is that it is guaranteed via relatively simple and efficient cryptographic transforms. Therefore, when source authentication is not paramount group authentication becomes useful. In addition, performing group authentication is useful even when source authentication is later performed: it provides a simple-to-verify weak integrity check that is useful as a measure against denial-of-service attacks.

The Multicast Group Authentication building block is placed in Problem Area 1 along the interface between Senders and Receivers.

[4.2.4](#) Multicast Group Membership Management

This building-block describes the functionality of registration and de-registration of members. Registration includes member authentication, notification and negotiation of security parameters, and logging of information according to the policies of the group controller and the would-be member. (Typically, an out-of-band advertisement of group information would occur before the registration takes place. The registration process will typically be invoked by the would-be member.)

De-registration may occur either at the initiative of the member or at the initiative of the group controller. It would result in logging of the de-registration event by the group controller and an invocation of the appropriate mechanism for terminating the membership of the de-registering member (see [Section 4.2.5](#)).

This building block also describes the functionality of the communication related to group membership among different GC+KS servers in a distributed group design.

In Figure 1, the Multicast Group Membership building block is placed in Problem Area 2 and has interfaces to Senders and Receivers.

[4.2.5](#) Multicast Key Management

This building-block describes the functionality of distributing and updating the cryptographic keying material throughout the life of the group. Components of this building may include:

- GC+KS to Client (Sender or Receiver) notification regarding current keying material (e.g. group encryption and authentication keys, auxiliary keys used for group management, keys for source authentication, etc).

Internet-Draft

September 2000

- Updating of current keying material, depending on circumstances and policies.
- Termination of groups in a secure manner, including the multicast group itself and the associated keying material.

Among the problems to be solved by this building block is the secure management of keys between Key Servers and Clients, the addressing issues for the multicast distribution of keying material, and the scalability or other performance requirements for multicast key management [[RFC2627](#), [BMS](#)].

To allow for an interoperable and secure IP multicast security protocol, this building block may need to specify host abstractions such as a group security association database (GSAD) and a group security policy database (GSPD) for IP multicast security. The degree of overlap between IP multicast and application-layer multicast key management needs to be considered. Thus, work on this functional building block must take into account the key management requirements for IP multicast, the key management requirements for application-layer multicast, and to what degree specific realizations of a Multicast Key Management building block can satisfy both. ISAKMP, moreover, has been designed to be extensible to multicast key management for both IP multicast and application-layer multicast security [[RFC2408](#)]. Thus, multicast key management protocols may use the existing ISAKMP standard's Phase 1 and Phase 2 protocols, possibly with needed extensions (such as an ISAKMP Domain of Interpretation for IP multicast or application-layer multicast security).

This building block also describes the functionality of the communication related to key management among different GC+KS servers in a distributed group design.

Multicast Key Management appears in both the centralized and distributed designs as shown in Figure 1 and is placed in Problem Area 2.

[4.2.6](#) Multicast Policy Management

This functional building block handles all matters related to multicast group policy including membership policy and multicast key management policy. Indeed, one of the first tasks in further defining this functional building block is identifying the different areas of

multicast policy. Multicast Policy Management includes the design of the policy server for multicast security, the particular policy definitions that will be used for IP multicast and application-layer multicast security, and the communication protocols between the Policy Server and the Key Server. This functional building block may be realized using a standard policy infrastructure such as a Policy

Decision Point (PDP) and Policy Enforcement Point (PEP) architecture. Thus, it may not be necessary to re-invent a separate architecture for multicast security policy; we expect that this work will evaluate use of the products of IETF efforts in the areas of network and security policy. At minimum, however, this functional building block will be realized in a set of policy definitions, such as multicast security conditions and actions.

The Multicast Policy Management building block describes the functionality of the communication between an instance of a GC+KS to an instance the Policy Server. The information transmitted may include policies concerning groups, memberships, keying material definition and their permissible uses, and other information. This building block also describes communication between and among Policy Servers. Thus, the Multicast Policy Management building block is placed in Problem Area 3, along the interface between Key Servers and Policy Servers. Group members are not expected to directly participate in this building block. However, this option is not ruled out.

[5. Conclusion](#)

As stated in [section 4](#), the ultimate goal of developing multicast security building blocks is to produce better specifications that can be standardized as expeditiously as possible. [Section 2](#) classifies the problems we seek to solve along the lines of Problem Areas. And [Section 3](#) presents a heuristic device, the Reference Framework, to facilitate investigation and standardization of multicast security building blocks. [Section 4.1](#) motivates this approach as having three distinct advantages. [Section 4.1](#) also advances some criteria for evaluating algorithms, protocols, and other specifications for the six functional building blocks that are proposed in [section 4.2](#). We recommend that work be undertaken to elucidate the requirements and functions of each of the building blocks that are proposed in [section 4.2](#). This work should be closely followed by analysis of algorithms and the

specification of protocol building blocks.

We expect that the partitioning of multicast security services along the lines of functional building blocks, as described in this document, has a number of advantages: The building blocks approach should encourage the development of particular technology for particular problems and foster the development of individual pieces of a whole multicast security protocol. Rather than delayed development, which is the first problem identified by RMT [RMT] above, work on the individual building blocks should proceed in parallel despite some obvious dependencies.

For a given building block, one dependency exists between parts of the building block that deal with a different building block (Centralized Design) and parts that deal with different instances of the same building block (Distributed Design). This dependency, however, does not prevent work on the Centralized Design to proceed, while at the same time the requirements and functions needed for distributed designs are considered by others who are working on this problem independently in SMuG.

A second dependency exists between the Multicast Confidentiality and the Source and Data Authentication building blocks. The protocols and packet formats for encrypted data transmission must accommodate the needs of source authentication and data integrity. Cryptographic protocols tend to be modular, however, and follow a building block approach. To draw an example from IPSec, the AH and ESP protocols are specified independently but are capable of being applied to any packet flow.

We believe that de-coupling multicast security services such as confidentiality and authentication/integrity can produce better protocols more quickly. And we expect to draw on a large body of experience of developing transport and network security protocols to avoid the risks and pitfalls of the building block approach while reaping its advantages for multicast security research and standardization.

[BMS] D. Balenson, D. McGrew, A. Sherman, Key Management for Large Dynamic Groups: One-Way Function Trees and Amortized Initialization, <http://www.ietf.org/internet-drafts/draft-balenson-groupkeymgmt-oft-00.txt>, February 1999, Work in Progress.

[CP99] R. Canetti and B. Pinkas, A taxonomy of multicast security issues, <http://search.ietf.org/internet-drafts/draft-irtf-smug-taxonomy-01.txt>, April 1999, Work in Progress.

[Din] Dinsmore, P., Balenson, D., Heyman, M., Kruus, P., Scace, C., and Sherman, A., "Policy-Based Security Management for Large Dynamic Groups: An Overview of the DCCM Project," DARPA Information Survivability Conference and Exposition, To Be Published.

[HCD] T. Hardjono, B. Cain, N. Doraswamy, A framework for group key management for multicast security, <http://www.ietf.org/internet-drafts/draft-ietf-ipsec-gkmframework-00.txt>, July 1998, Work in Progress.

[Har1] Harney, H., and Muckenhirn, C., "Group Key Management Protocol (GKMP) Specification," [RFC 2093](#), July 1997.

Secure IP Multicast

[Page 18]

Internet-Draft

September 2000

[Har2] Harney, H., and Muckenhirn, C., "Group Key Management Protocol (GKMP) Architecture," [RFC 2094](#), July 1997.

[HH] H. Harney, E. Harder, Group Secure Association Key Management Protocol, <http://search.ietf.org/internet-drafts/draft-harney-sparta-gsakmp-sec-00.txt>, April 1999, Work in Progress.

[IKEdraft] D. Harkins, D. Carrel, The Internet Key Exchange (IKE), <http://www.ietf.org/internet-drafts/draft-ietf-ipsec-ike-01.txt>, May 1999, Work in Progress.

[McCarthy] L. McCarthy, RTP Profile for Source Authentication and Non-Repudiation of Audio and Video Conferences, [draft-mccarthy-smug-rtp-profile-src-auth-00.txt](#), May 1999, Work in Progress.

[McD] McDaniel, P., Honeyman, P., and Prakash, A., "Antigone: A Flexible Framework for Secure Group Communication," Proceedings of the Eight USENIX Security Symposium, pp 99-113, August, 1999.

[RFC1889] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, January 1996.

[RFC2014] A. Weinrib and J. Postel, IRTF Research Group Guidelines and Procedures, [RFC 2014](#), October 1996.

[RFC2246] Dierks, T. and C. Allen, The TLS Protocol Version 1.0, [RFC 2246](#), January 1999.

[RFC2327] M. Handley, V. Jacobson, SDP: Session Description Protocol, April 1998.

[RFC2367] D. McDonald, C. Metz, B. Phan, PF_KEY Key Management API, Version 2, July 1998.

[RFC2401] S. Kent, R. Atkinson, Security Architecture for the Internet Protocol, November 1998

[RFC2402] S. Kent, R. Atkinson, IP Authentication Header, November 1998

[RFC2403] C. Madson, R. Glenn, The Use of HMAC-MD5-96 within ESP and AH, November 1998.

[RFC2404] C. Madson, R. Glenn, The Use of HMAC-SHA-1-96 within ESP and AH, November 1998.

[RFC2406] S. Kent, R. Atkinson, IP Encapsulating Security Payload (ESP), November 1998.

[RFC2407] D. Piper, The Internet IP Domain of Interpretation for ISAKMP, November 1998.

Secure IP Multicast

[Page 19]

Internet-Draft

September 2000

[RFC2408] D. Maughan, M. Shertler, M. Schneider, J. Turner, Internet Security Association and Key Management Protocol, November 1998.

[RFC2409] D. Harkins, D. Carrel, The Internet Key Exchange (IKE), November, 1998.

[RFC2627] D. M. Wallner, E. Harder, R. C. Agee, Key Management for Multicast: Issues and Architectures, September 1998.

[RMT] B. Whetten, L. Vicisano, R. Kermode, M. Handley, S. Floyd, Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer, <http://www.ietf.org/internet-drafts/draft-ietf-rmt-buildingblocks-00.txt>, June 1999, Work in Progress.

[Schneier] B. Schneier, Applied Cryptography, Second Edition, John

Wiley, 1996.

[SAP] M. Handley, C. Perkins, E. Whelan, Session Announcement Protocol, <http://www.ietf.org/internet-drafts/draft-ietf-mmusic-sap-v2-01.txt>, June 1999, Work in Progress.

[SAPPK] P. Kirstein, G. Montasser-Kohsari, E. Whelan, SAP Security Using Public Key Algorithms, <http://www.ietf.org/internet-drafts/draft-ietf-mmusic-sap-sec-04.txt>, September 1998, Work in Progress.

[Wong] C.K.Wong, S.S. Lam, Digital Signatures for Flows and Multicasts, Proceedings of IEEE ICNP'98, October 14-16, 1998.

[WP96] A. Weinrib and J. Postel, IRTF Research Group Guidelines and Procedures, [RFC 2014](#), October 1996.

Authors' addresses:

Thomas Hardjono
Advanced Networks
Nortel Networks
[600](#) Technology Park Dr.
Billerica, MA 01821

(978) 288-4538
thardjono@baynetworks.com

Ran Canetti
IBM Research
[30](#) Saw Mill River Rd
Hawthorne, NY 10532
(914) 784-7076
canetti@watson.ibm.com

Mark Baugher
PassEdge
[20400](#) NW Amberwood Drive
Beaverton, OR 97006, USA
(503) 466-8406
mbaugher@passedge.com

Peter Dinsmore
NAI Labs
[3060](#) Washington Road,
Glenwood, MD 21738
(443) 259-2346
Pete_Dinsmore@NAI.com

Expires March 2001
Secure IP Multicast

[Page 21]

Thomas Hardjono email1: thardjono@yahoo.com
 email2: hardjono@nortelnetworks.com
 Tel: +1-978-288-4538
