

Internet Engineering Task Force
INTERNET-DRAFT
draft-irtf-smug-mcast-policy-00.txt
May 2000

Patrick McDaniel (U.of Michigan)
Hugh Harney (Sparta)
Peter Dinsmore (NAI Labs)
Atul Prakash (U.of Michigan)

Multicast Security Policy
<draft-irtf-smug-mcast-policy-00.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

Security is increasingly becoming a concern in applications built on multi-party communication. Centrally, protection of the application content from non-authorized or malicious parties is the fundamental goal of any security policy specification. This draft seeks to illuminate the design space of secure multicast communication policy. The security requirements of existing application policies are intended to be addressable by these policy dimensions. It is from an understanding of policy design space that the mechanisms for policy specification and enforcement can be derived.

Table of Contents

1. Introduction

Although security policy is an oft mentioned component of security infrastructures, no single definition has been found to address the needs of all parties and environments. One working definition defines policy as:

``.. the group security relevant behaviors, access control parameters, and security mechanisms.'...' [HCH+00]

This definition best fits the viewpoint of policy as defining how

security defines group behavior, who are the entities allowed to participate, and which mechanisms will be used to achieve mission critical goals. For the purposes of this document, we will accept these dimensions as defining the relevant properties of secure multicast policy.

A group security context is the set of keys, members, protocols, and algorithms used to secure a singular group communication. A policy defines (directly or indirectly) how the security context should be derived. A policy framework defines the entities and protocols used to define, interpret, negotiate, and distribute a secure group policy.

Once an understanding of what constitutes a policy is obtained, the natural question of policy specification arises. This question speaks to the requirements of both the representation and interpretation of policy, and the means by which policy is distributed and authenticated.

The central aims of this document are the definition of a group model and an investigation of the dimensions along which policy may be defined in secure multiparty communication. The group model should be comprehensive and flexible; the requirements of arbitrary group communication security should be expressible and achievable. In this model, we attempt to identify those dimensions of policy relevant to secure multicast. This document does not seek to identify specific solutions or mechanisms for providing group security policy specification and enforcement, leaving this to other future documents.

The remainder of this section describes the major components and challenges of secure group policy. The following two subsection describe a major distinction between group and local policy upon which the model defined in this document rests. The issues of policy specification and negotiation are briefly described in subsequent sections.

1.1 Group Policy

A group policy defines the behavior of the group. Included in this definition is the types of security guarantees provided to the participants, the ways in which the guarantees are provided and the definition of the relationships between the group participants.

1.2 Local Policy

Obtained from the local environment, a local policy defines the requirements and credentials of the local entities. Upon receiving the group policy, both the group and local policies should be reconciled. Irreconcilable conflicts (see negotiation below) requires the local host abstain from participation in the group.

In addition to specifying minimal standards for group behavior, the

local policy should identify which identities or credentials should be accepted as authoritative. In the case of identities, an authentication method should also be defined.

1.3 Policy Specification

Once agreement has been reached on the policy dimensions that will be supported by a policy framework the issue of specification arises. A policy specification language defines both how a policy is represented and the rules with which the representation is interpreted. The following text outlines several design goals of the policy representation.

The policy language should be unambiguous. Because a policy will be interpreted by host software, the mapping of the policy to mechanism and credentials must be deterministic.

The policy language should be succinct. The costs associated with policy distribution is likely to be a key determinant in the success of the policy framework. As such, the language should be designed to be represented in as small a electronic format as possible.

The policy language should be clear. A requirement of the SMuG architecture is that policy definitions must originate from security personnel which may or may not be directly involved in application development. Thus, the ability to relate the representation to real world objects is a goal. Where possible, a policy representation should be human readable.

1.4 Policy Negotiation

Reconciliation, possibly through participant negotiation, of local and group policy is a key task of the SMuG framework. A open issue is the extent to which a group policy may be altered during this negotiation. Secure multicast groups may be large and extend over several administrative domains. Thus, the potentially competing requirements of group members must be weighed in creating a single, coherent group security policy.

1.5 Related Documents

This informational document is intended to motivate the design of the Secure Multicast Research Group [SMuG] secure multicast architecture [HCBP99,CCP+99]. Specifically, this document intends to identify potential policies that may be supported by the SMuG secure multicast framework through the policy framework (problem area 3). This should also serve as an informal policy requirements specification for the management of keying material (problem area 2) and multicast data handling (problem 1) layers of the SMuG architecture. A more detailed document defining the requirements of the policy framework is the subject of a future draft. A taxonomy of issues relating to the SMuG framework can be found in [CP99].

A secondary goal of this document is a requirements definition for the framework policy specification language. In this, we intend to outline a general model which can be used as the basis of policy specification. While the specifics of the policy language are not stipulated, we expect that future documents will realize this model in defining an electronic policy representation.

1.6 Document Organization

The remainder of this document is as follows. Section 2 defines a group model through a specification of the actions and roles of group participants. Section 3 describes the potential dimensions on which policy may be defined, and identifies those policies required for secure multicast communication.

2. Group Model

This section outlines the group model under over which we define policy, and is much motivated by the emerging GSAKMP protocol [HCH+00]. In this model, the defining characteristic of a group is a statement of rights associated with roles assumed by group members.

A role defines the rights and responsibilities of any group member assuming that role. In gaining access to the group, a group member must assume one or more roles. External entities may serve as roles within the group. In this view, the task of the policy framework is the specification and distribution of role definition, the means and specifics of role access control, and rules guiding the realization of these role defined activities in the underlying mechanisms.

To motivate (one definition) of roles, we present an enumeration of activities relating to the management of a group security context.

##	Action	Description
--	-----	-----
1	key creation	right to create a session key, or to generate rekeying material
2	key dissemination	This right allows a member to distribute keying material.
3	rekey action initiation	right to initiate a group rekey.
4	key access	right to gain access to the session key
5	policy creation	right to create/assert a group policy
6	policy modification	right to modify the group policy
7	grant rights	right to grant rights to members/entities external to the policy

8	authorize member	right to authorize/state authenticity of group member
9	admit member	right to admit a member to the group
10	eject member	right to remove a member from the group
11	audit group	right to monitor access control messages or membership information

Tables 1 and 2 describe the set of access rights and roles assumed by group participants. This description of roles and rights is intended to be informational. Particular instantiations of the SMuG framework may define roles and rights as appropriate for their purposes. We describe the meaning of each role in the following subsections.

#	Role	Access Rights
1	group owner	5, 6, 7, 11
2	group key authority	1, 2, 3, 4, 11
3	group membership authority	8, 9, 10
4	member	4

Table 1 - Role Access Rights

The following subsections describe the purpose, rights, and responsibilities of the group roles defined in Table 1.

2.2.1 Group Owner

The group owner is the initiator of the group and controls the policy for the group. The group owner is the entity that states rules for admittance and determines the behavior of the group (policy). Also, it must identify the authorities that perform the various management duties for the group. Note that the group owner may or may not be a member of the group.

2.2.2 Group Key Authority

The group key authority is the controller of keying actions within the group. As such, this entity creates and coordinates the distribution of all session keys and related keying material. The entities to which keying material will largely be driven by information received from the group membership authority. Note that

the group key authority need not be a member of the group or distinct from the group owner.

2.2.3 Group Membership Authority

The group membership authority is the controller of membership actions within the group. As such, this entity authorizes and admits the members of the group. The means used to perform these actions is dependent on the policy stated by the group owner. Note that the group key authority need not be a member of the group.

2.2.4 Member

A group member is a participant in the group. The right to access the session key implies the ability to both send and receive messages within the group. Obviously, the member is required to be a member of the group. Note that a member specifically does not have any rights to monitor the group control messages or membership. If needed, these rights may later be granted through the definition of a new role.

3. Policies

3.1 Group Policy

A group policy defines group services and participant roles to be implemented by the group. The central goal of the SMuG policy framework (problem area 3) is to provide services for specification, distribution, and negotiation of the group policy.

As defined by roles (see section 2.2 above), specification of the group policy is to be performed by some authorized entity. The group policy is to be distributed to joining members by policy distribution points (see [CCP+99]). The mechanism and architecture of the specification and distribution mechanism is beyond the scope of this document.

The following subsections define several dimensions along which a group policy may be defined.

3.1.1 Rekeying Policy

A common strategy to support secure group communication among trusted members is to use a common symmetric session key (e.g. GKMP [HMR97a,HMR97b], GSAKMP [HCH+00], Antigone [MPH99], DCCM [DBH+00]). An important policy issue for a group communication application is deciding when a session must be rekeyed, i.e., the old session key is discarded and a new session key is sent to all the members. This policy is likely to drive much of the key management activities of the key management protocols [HBH00].

A rekeying policy defines how and when session keys are created and (re)distributed. The management of the session keys is a central

determinant of the security afforded by the resulting session. As such, the rekeying policy drives many subsequent policy and mechanism related decisions.

Associated with each rekeying policy is a number of properties which define the security guarantees being provided to the group. Several properties of the session keying include:

session key independence - no meaningful information about one session can be derived from another.

perfect forward secrecy (PFS)- the property that a session key provides no meaningful information about future session keys.

membership forward secrecy (MFS) - the property that a member leaving the group cannot obtain meaningful information about future group communication.

perfect backward secrecy (PBW) - the property that a session key provides no meaningful information about past session keys.

membership backward secrecy (MBS) - the property that a member joining the group cannot obtain meaningful information about past group communication.

failure secrecy - the property that a failed process can not continue to actively or passively participate in the group. The means in which failures are detected and reported is beyond the scope of this document.

compromise secrecy - the property that a compromised process can not continue to actively or passively participate in the group. The means in which compromises are detected and reported is beyond the scope of this document.

limited lifetime - the property that a session key has maximum lifetime (which may be measured in time, bytes transmitted, or some other globally measurable metric of group communication).

Similarly, the mechanism used to create session keys may have the following properties:

contributory keying - the property that each group member participate in the creation of the session key.

centralized keying - converse to contributory keying, this property requires that (only) one or more trusted parties contribute to the creation of the key.

These properties are realized in some combination of key creation algorithm and rekeying protocols. The definition of these algorithms, protocols, and the (policy to mechanism) mapping function is beyond the scope of this document. The Antigone system

[MPH99] investigates the means by which policy may be mapped into mechanisms.

Support for some or all of the rekeying policies defined in this section is a requirement of the SMuG Policy framework.

A representation of a rekeying policy may be the set of properties which the rekeying mechanism is required to provide. Each property may require additional policy specification and mechanisms. To illustrate this point, we describe the features and requirements of two policies mentioned above.

3.1.1.1 Membership Forward Secrecy

A membership forward secrecy policy is useful in secure conferencing applications. The content of conferencing application is often driven by the members of the group. (e.g. The content of a sales meeting may need to be protected from suppliers who have exited the session.)

In membership forward secrecy, The group is required to protect content from members of past security contexts. Therefore, any rekeying mechanism and protocol supporting an MFS policy must provide the following features:

- a) protection of the LEAVE process (reliable, authenticated, and timely)
- b) rekey after every group member LEAVE
- c) provide PFS rekeying

It is immediately obvious MFS may be both be difficult to provide and expensive. Thus, MFS policies may be incompatible with large or highly dynamic groups. Support for MFS in the secure multicast framework may not be required.

3.1.1.2 Limited Lifetime

Limited lifetime rekeying can be useful in a secure on-line subscription service. Paying members would periodically be sent a new key that is valid until the next subscription interval. The GKMP [HMR97a,HMR97b] protocol implements a time-sensitive rekeying policy (albeit without the session key independence required by the this example). Limited lifetime rekeying without independence provides (primarily) protection against cryptanalysis of the session key. The MARKS system [Bri99] provides an efficient means of supporting limited lifetime rekeying within arbitrarily large groups.

Where limited lifetime rekeying is used, the metric used to measure key lifetime and the threshold at which rekeying is required must be stated in the group policy. Mechanisms implementing this policy must support the measurement of key lifetime and periodic PFS rekeying.

Limited lifetime rekeying is supported by the vast majority of existing secure multicast and group communication frameworks. The mechanism requirements of this policy are simple and strait-forward. Finally, to avoid cryptanalysis of session keys, periodic rekeying is a good security practice. Thus, support for limited lifetime rekeying is a likely requirement of the SMuG policy framework.

3.1.2 Access Control Policy

An access control policy states the identities/credentials, rights and responsibilities of each member of the group. Access control in our current group model is defined by the roles assumed by group participants. The specification, meaning, and mechanisms for assuming these roles is as defined above in section 2.2.

Support for role defining access control policy is a requirement for the SMuG policy framework.

3.1.3 Data Security Policy

The canonical security policy, a data security policy states the security guarantees provided to application level messages. This policy is likely to directly or indirectly state the data transforms defined by the SMuG problem area 1 group [CRC00] used to secure group messages. Several data security guarantees include;

confidentiality - Guarantee stating that no member outside the group can obtain the contents of a group message.

integrity - Guarantee stating that any modification of a group message during transmission is detectable by the receiver.

group authentication - Guarantee stating that a received message was transmitted by some member of the group. This is typically a byproduct of other (data security) guarantees.

source authentication (or sender authentication) - Guarantee stating that the sender of a message can be uniquely identified. Providing this guarantee in an efficient and scalable way is an open issue. However, recent developments by Perrig et. al. [PSTC00] outline several promising solutions for providing efficient sender authentication.

non-repudiation - Guarantee stating that a sender should not be able to falsely deny sending a previously transmitted message.

anonymity - Guarantee stating that the originator of a message cannot be ascertained by receivers (or by outside parties).

The cryptographic algorithms and used to provide these guarantees have varying strength and performance characteristics. As such, a data-security policy should be able to state the algorithm(s) that may used to provide data security. A result of the selection of

cryptographic algorithms leads to the following kinds of secrecy:

ephemeral secrecy - the data be protected for (only) a short period after transmission. That is, the algorithm should prevent easy access to content, but strong guarantees are not required. One example, from [CP99] is:

[... to maintain ephemeral secrecy when transmitting a video it is sufficient to encrypt only the low-order Fourier coefficients in an MPEG encoding.]

long-term secrecy - requirement that the transmitted data be protected for an indefinite period after transmission. This requires strong cryptographic algorithms.

A "cipher-suite" is one or more cryptographic algorithms used to implement data related guarantees. The policy specification should allow, at a minimum, cipher suite definitions that support the specification of acceptable algorithm parameters and modes. Additionally, the suite definition should indicate the guarantees for which the suite should be used.

Support for some, if not all, data security policies is a requirement of the SMuG policy framework. A definition of the supported cipher suites should be developed by the multicast data transform specifications (level 1).

3.1.4 Member-Data Policy

A member-data policy indicates the availability of group membership information, states guarantees of the accuracy of this information, and identifies the mechanism used for its distribution.

Identification of the membership within a group session is an important requirement for a large class of applications. As evidenced by a number of group communication systems, achieving strong guarantees for the availability and correctness of group membership can be costly. Several member-data policies worth considering are:

best-effort member-data - In this policy, membership data will be delivered as available. No guarantees about the accuracy or timeliness of this information are provided. However, due-diligence should be expended in providing accurate membership data.

positive member-data - This policy guarantees that all members in the membership data are actively participating in the group. That is, a listed member is guaranteed to be receiving data and has not failed (see below, in Failure Policy 3.1.6, for a definition of a process failure).

negative member-data - This policy guarantees that every member of

the current security context is listed in the membership data.

perfect member-data - This policy guarantees that all members in the membership data actively participating in the group, and that every member of the current security context is listed in the membership data. That is, both positive and negative member-data is provided.

A related policy is confidentiality of group membership. In general, hiding the group membership information from members and non-members is difficult to do in current networks. This is primarily because the ability to monitor messages on the network allows access to the source and destination of packets (in case of unicasts) and at the multicast tree (in case of IP multicasts). In mounting this traffic analysis attack, an adversary may deduce a close approximation of group membership.

It is unclear if member-data policies are within the scope of the SMuG framework. However, we note that the availability of accurate membership information is a pre-requisite of some reliable multicast solutions being discussed by the Reliable Multicast Research Group [RM].

3.1.5 Compromise Policy

A compromise policy identifies the types of compromises to be detected, the means by which they are reported, and the mechanism used for recovery. Compromise related algorithms may or may not be protocol and keying algorithm dependent. Further investigation of the requirements of these policies is required.

It is unclear if compromise detection and recovery is within the scope of the SMuG framework.

3.1.6 Failure Policy

A failure policy defines what kinds of (member) failures are to be detected and the mechanisms used for failure detection, reporting, and recovery. The definition of a failure policy should be derived from a process "crash model". As defined in [Mul93], traditional crash models include;

fail-stop - The failed process immediately and permanently stops sending and receiving messages. The vast majority of secure group and multicast frameworks and protocols assume a fail-stop failure model, if any.

message-omission - The failed process will not receive or send (omit) an arbitrary number of messages.

Byzantine - The failed process can exhibit any behavior whatsoever. A failed process in a Byzantine failure model should be assumed to be actively attempting to circumvent the security of

the group. As demonstrated in RAMPART [Rei94] system, known mechanisms providing protection from Byzantine failures are both expensive and complex. Note that mechanisms used to combat Byzantine failures are often similar to compromise recovery algorithms.

Systems supporting failure detection and recovery techniques found in survivability and distributed systems literature typically do not address security. Thus, the integration of these services with a security infrastructure requires careful design and analysis. It is unclear if these policies should be addressed by the SMuG framework.

3.1.7 Domain Dependent Policy

A domain dependent policy dictates ways in which the security context may be effected by forces external to secure multicast services. Such a policy would specify the modification of group behavior in response to the observation of an external event or state.

An example of a domain dependent policy is the modification of group access rights during a launch window. At the Kennedy Space Center (NASA), monitoring devices on the space shuttle continuously transmit data to a number of monitoring applications. Outside a launch window, the applications may alter configuration or test devices as needed. During the launch window, these devices transmit monitoring data, but access to configuration and testing interfaces is prohibited. One group (domain dependent) policy supporting the requirements of this environment would state that no application should be able to send to the group during the launch window. Another policy would outright prohibit the (testing) applications from participating in the group during a launch window.

Developing an enumeration of all potential domain dependent policies is infeasible. Thus, if supported, flexible interfaces for reporting external events and state must be provided. An open issue are the security requirements for the event detection and state assessment mechanisms. A second issue is the support of domain dependent roles (such as the application role in the above example).

At a minimum, it seems necessary for the secure multicast group to provide interfaces to secure group related activities. Such activities may include (but are not restricted to); initiate rekeying, member ejection, and compromise recovery.

3.2 Local Policy

A local policy states the security and performance requirements of the local infrastructure on the SMuG framework. The mechanism used for the specification and distribution of local policy is beyond the scope of this document.

The following subsections define several dimensions along which a

local policy may be defined.

3.2.1 Infrastructure Policy

An infrastructure policy identifies the locally trusted entities and indicates which mechanisms may be used to obtain other credentials. In this, a statement of what groups the host is allowed to join may be implicitly specified.

Typically, this policy is used to identify the identity, location, and mechanism of locally held credentials (e.g. long term keys). It is from these credentials that access to the group services will be likely be obtained. An example policy of this type may identify a file in the local filesystem that contains a long term key. Note that the means in which the contents of the file is interpreted must also be specified.

Conversely, the infrastructure will also state the means in which the identities and credentials received from the group will be verified (.e.g. location of locally trusted CA). The identity of trusted framework components (e.g. policy distribution points) may also be specified.

Support for infrastructure policy is a requirement of the SMuG framework.

3.2.2 Policy Requirements

This policy state the minimum services a host will accept. In this, it may state the following requirements of any group:

trusted entities - This policy states which parties may assume roles within the group. For example, one policy may state that a local process may only join groups whose key controller is a specific host (e.g. antigone.citi.umich.edu).

data security policies - This policy requires that groups provide particular data security policies.

others - In general, any policy expressible in the group policy should be able to be stated as a policy requirement.

Policy requirements are likely to drive any policy negotiation process. Converging on a set of services that meet the requirements of all members is an open issue.

Support for policy requirements is a requirement of the SMuG framework.

4. References

[Bri99] Bob Briscoe, "MARKS: Zero Side-Effect Multicast Key Management Using Arbitrarily Revealed Key Sequences", In Proceedings

of First International Workshop on Networked Group Communication, November 1999.

[CCP+99] R. Canetti, P-C. Cheng, D. Pendarakis, J.R. Rao, P.Rohatgi and D. Saha, "An Architecture for Secure Internet Multicast", Internet Engineering Task Force, February 1999, draft-irtf-smug-sec-mcast-arch-00.txt (Draft).

[CP99] R. Canetti and B. Pinkas, "A Taxonomy of Multicast Security Issues (updated version)", Internet Research Task Force, April, 1999, draft-irtf-smug-taxonomy-01.txt (Draft).

[CRC00] Ran Canetti, Pankaj Rohatgi, and Pau-Chen Cheng, "Multicast Data Security Transformations: Requirements, Considerations, and Prominent Choices", Internet Engineering Task Force, May 2000, draft-data-transforms.txt (Draft).

[DBH+00] P. Dinsmore, D. Balenson, M. Heyman, P. Kruus, C Scace, and A. Sherman "Policy-Based Security Management for Large Dynamic Groups: A Overview of the DCCM Project", In Proceedings of DARPA Information Survivability Conference and Exposition (DISCEX '00), pages 64-73, DARPA, January 2000.

[HBH00] H. Harney, M. Baugher, and T. Hardjono, "GKM Building Block: Group Security Association (GSA) Definition", Internet Engineering Task Force, February 2000, draft-irtf-smug-gkmbb-gsedef-00.txt (Draft).

[HCBP99] T. Hardjono, R. Canetti, M. Baugher and P. Dinsmore, "Secure Multicast: Problem Areas, Framework, and Building Blocks", Internet Engineering Task Force, October, 1999, draft-irtf-smug-framework-00.txt (Draft)

[HCH+00] H. Harney, A Colegrove, E. Harder, U. Meth, and R. Fleischer, "Group Secure Association Key Management Protocol", Internet Engineering Task Force, May 2000, draft-harney-sparta-gsakmp-sec-01.txt (Draft)

[HMR97a] Harney, Hugh, Carl Muckenhirn, and Thomas Rivers, "Group key management protocol (GKMP) architecture," Request for Comments (RFC) 2094, Internet Engineering Task Force (July 1997).

[HMR97b] Harney, Hugh, Carl Muckenhirn, and Thomas Rivers, "Group key management protocol (GKMP) specification," Request for Comments (RFC) 2093, Internet Engineering Task Force (July 1997).

[MPH99] P. McDaniel, A. Prakash and P. Honeyman, "Antigone: A Flexible Framework for Secure Group Communication", In Proceedings of the 8th USENIX Security Symposium, pages 99-114, August, 1999

[Mul93] Sape Mullender. Distributed Systems. Addison-Wesley, First edition, 1993.

Tel: +1-978-288-4538
