

Internet Research Task Force  
INTERNET-DRAFT  
September 2000

Hugh Harney (SPARTA)  
Mark Baugher (PassEdge)  
Thomas Hardjono (Nortel)  
Expires March 2001

## GKM Building Block: Group Security Association (GSA) Definition

<[draft-irtf-smug-gkmbb-gsadev-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 definition for Group Security Associations (GSA) for the support of group key management in IP multicast security. The reasoning behind the structure of the GSA is discussed, and a definition of the GSA is provided.

### [1](#). Introduction

This document describes a Group Key Management Building Block (GKM-BB) following the Secure IP Multicast Framework and Building Blocks document [[HCB000](#)]. In particular, the current document answers the need for further definition of the multicast key management building block [[HCB000](#)]. The GKM-BB and its relationship with the other building blocks is shown in Figure 1.

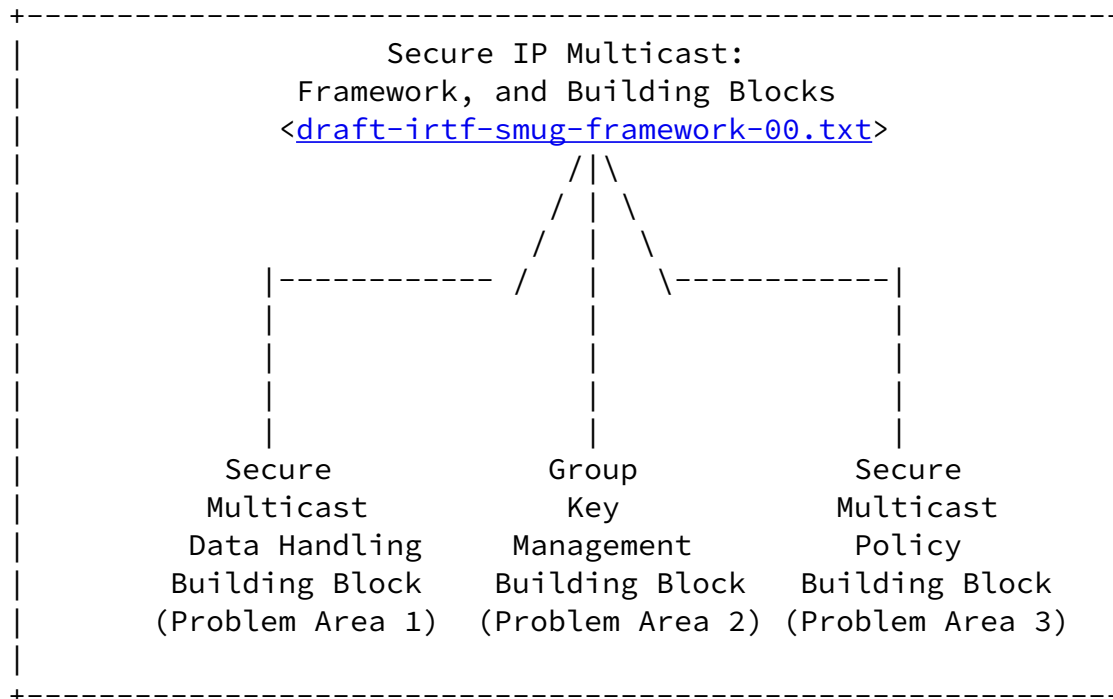


Figure 1: Building blocks defined for Secure IP multicast

### [1.1](#) Group Key Management Building Block and Protocol Instantiations

Within the context of group key management for IP multicast security, the current document seeks to provide the framework for group key management Protocol Instantiations (PI). Following the concept of building blocks in [[HCB00](#)], the PIs implement group key management at various layers of the protocol stack, answering the need of various applications. The relationship between the group key management framework and the PIs is shown in Figure 2.

### [1.2](#) Aim of the GKM Building Block

The aim of the current document is to define a GKM Building Block (GKM-BB) for the establishment of a Group Security Association (GSA) and Group-Key establishment. The notion of a GSA is described in [Section 2](#).

A protocol that can establish and maintain GSAs provides a framework for the design of group key management Protocol Instantiations.

To this end, the current document seeks to:

- Identify the entities involved, and define the functions to be carried-out by these entities.

Harney, Baugher, Hardjono

[Page 2]

---

INTERNET-DRAFT

Group Security Association

September 2000

- Define the concepts and behaviors and explain their motivations.
- Define the constructs in the form of data items exchanged between any two entities involved in the GKM-BB in sufficient detail so as to enable different Protocol Instantiations (PI) to be developed from them.

Above all, the current effort aims to define, develop and evolve the GKM-BB in a manner that conforms to the original intent of the Building-Blocks approach adopted in [[HCB00](#)]. This includes the reuse of existing technology and the introduction of newer technologies to satisfy the requirements of Group Key Management, provide timely delivery of technology through standardization of the GKM-BB, and to validate the practical usefulness of the Building Block for a variety of applications.

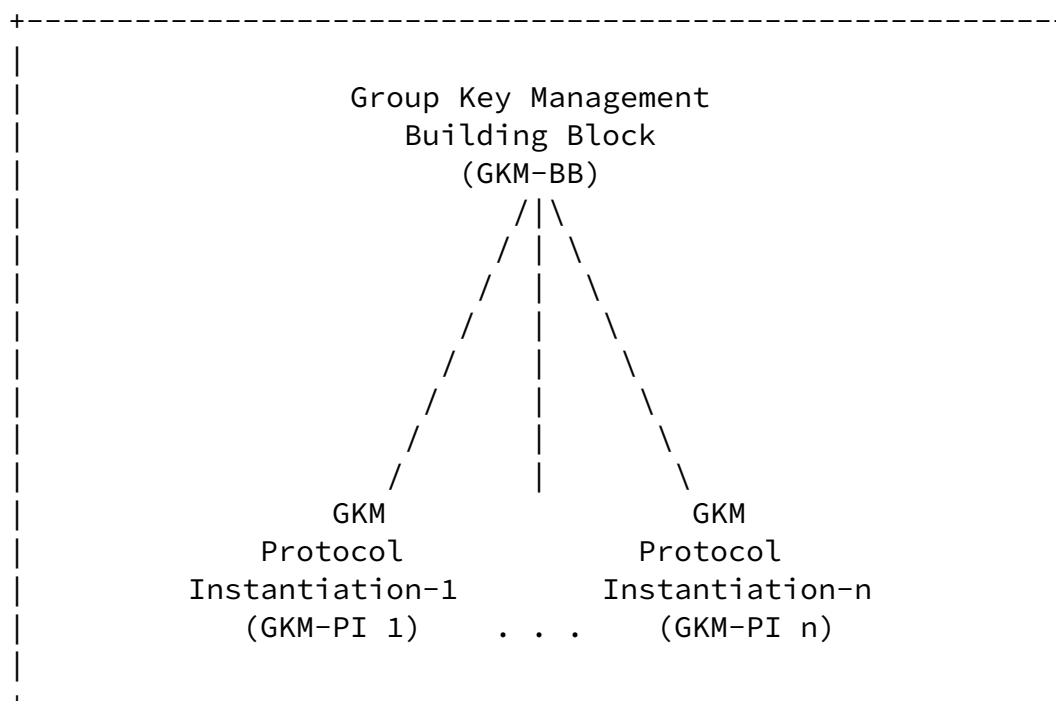


Figure 2: Relationship between the GKM-BB and GKM-PI

### [1.3](#) Elements of the Secure IP Multicast Framework

The Secure IP Multicast Framework and Building Blocks document [[HCBD00](#)] describes a number of entities, which participate in the creation, maintenance, and removal of secure multicast groups. Those that are immediately of concern for group key management and membership management are as follows.

Harney, Baugher, Hardjono

[Page 3]

---

INTERNET-DRAFT

Group Security Association

September 2000

#### [1.3.1](#) Entities and functions

##### Group Controller and Key Server (GCKS)

The GCKS entity embodies both the physical entity and functions of the group controller and the key server. Although two families of functions can be distinguished, namely membership management and group key management, for simplicity both families of functions will be provided by a single physical entity. For any given multicast group, a "Main" or "Root" GCKS must be identified using methods and mechanisms related to a group's initial definition/configuration.

##### Member (Receiver and Sender)

The member is the group member, defined for a particular instance of group communications. The member entity can exist at different layers (eg. user, host, process) and thus must be defined across the group consistently and is best expressed through their corresponding certificates.

Although not directly addressed in the current document, another entity that is involved in group key management is the Remote GCKS. The Remote GCKS expresses the scalability and inter-domain requirements of group key management. A Remote GCKS is identical in functionality to the GCKS. However, in terms of authorization level to perform the management of group keys, a GCKS may possess a different relationship to another GCKS within the same management regime. Examples include a peer relationship among a set of GCKS, and a hierarchical relationship where a Root GCKS is defined and the other GCKS are subordinates of the Root.

## [2](#). GROUP KEY MANAGEMENT REQUIREMENTS AND PROPERTIES

The requirements discussed in this section are for the most part not original but come from a variety of sources. The Internet key management literature is one source. Group key management must operate over packet internetworks, particularly IP multicast internets. Thus group key management has at least some of the properties of Internet key management. Indeed, the very notion of "key management," as distinct from "key exchange," is taken from work done on IPsec. Thus, the Internet key management requirements presented in this memo are gleaned from prior work done on IP security, key management for packet networks, and authenticated key exchange [RFC2409, [RFC2412](#), [RFC2408](#), [RFC2407](#), [RFC2522](#), Kraw96, SDNS88, DVW92]. Our second source of requirements is taken from previous work on multicast security [RFC2627, CP99, HH99a, HCD00].

Harney, Baugher, Hardjono

[Page 4]

---

INTERNET-DRAFT

Group Security Association

September 2000

Group key management requires additional properties beyond those found in the Internet key management work done to date. Group keying material, for example, are not negotiated but sent to and shared by groups of members, which must agree to common policies that are not negotiated [[CP99](#), [HH99a](#)]. Furthermore, the key exchange/distribution architecture is not only peer-to-peer but also operates between key server and key client [[HCB00](#)]. The common and distinct properties of Internet key management and group key management are the subject of this section.

[Section 2.2](#) discusses group key management. [Section 2.1](#) is an overview of internet key management and its applicability to group key management. In both sections we consider the needed properties of a group key management protocol.

## [2.1](#) Internet Key Management and Key Determination

"Authenticated key exchange" is basic to internet key management and key determination protocols, which seek to thwart attacks that may occur on an unsecured network. The types of attacks include man-in-the-middle, connection-hijacking, and reflection/replay attacks, many of which can be combated by mechanisms such as "direct authentication," which integrate authentication into the key exchange, as described in the STS protocol [[DVW92](#)]. Messages that are exchanged as part of a "run" should be chained with authenticatable information, including random data that is contributed by each party in a two-party key exchange. This technique helps ensure that messages received by a peer match what the other peer sent. Work has been done, moreover, to formally prove AKE properties based upon the matching of messages sent and received by

peers in the exchange [[BR93](#)]. When session keys are used to protect exchanges that determine other session keys, "perfect forward secrecy" (PFS) can ensure that "...disclosure of long-term secret keying material does not compromise the secrecy of the exchanged keys from earlier runs" - so long as authentication is linked to the key exchange [[DVW92](#)]. The PFS requirement, however, entails the performance penalty of a Diffie-Hellman exchange, which may not be appropriate for all applications.

The notion of a "selectable level of security" is basic to key management on internetworks, which are composed of diverse communications networks and host computers. In this environment, some applications may tradeoff better security for reduced communications and computing costs. The security choices depend upon application need as well as the capabilities of the hosts and network devices. In order to support heterogeneous network and host devices, Internet key management supports multiple types of exchanges that can be composed in various

Harney, Baugher, Hardjono

[Page 5]

---

INTERNET-DRAFT

Group Security Association

September 2000

ways; some exchanges may support identity protection and provide PFS, for example, while others may not [[Kraw96](#)]. To accommodate diversity, a versatile approach supports a variety of transforms and Diffie-Hellman groups, all of which can be negotiated among communicating entities [[RFC2412](#), [RFC2409](#)]. Internet key management, moreover, supports a "forward migration path" in the protocol so that new algorithms can be introduced, as older methods need to be replaced [[RFC2409](#), [RFC2412](#), [RFC2408](#), [Kraw96](#)].

In fact, the key establishment procedure itself may need to be replaced over time, and the Internet Security Architecture has a key management framework, the Internet Security Association and Key Management Protocol (ISAKMP), which defines an abstract set of exchanges, organized by "modes" and "phases" to provide a selectable level of protection [[RFC2408](#), [Kraw96](#)]. To provide a versatile solution for internet key management, ISAKMP permits alternative authentication mechanisms in its exchanges and is parameterized by a "domain of interpretation" (DOI) in which specific key determination mechanisms are defined through the specification of the name space, policy, specific payloads and, optionally, new exchanges. In this way, ISAKMP is designed to be extended for alternative uses and to allow a forward migration of key exchange protocols and cryptographic transforms. Although the flexibility of their approach may arguably result in more complexity, which may in turn lead to weaker security [[Ferguson & Schneier](#)], the ISAKMP authors recommend the use of ISAKMP as a single key management framework for new uses such as group key management, as well as

transport and application key management [[RFC2408](#)]. New uses can be realized through the specification of a DOI.

ISAKMP achieves its versatility by being more abstract than a key determination protocol since it manages "security associations" (SA) and not just keys. The SA abstraction [[RFC2408](#), [RFC2401](#), [RFC2522](#), [SDNS88](#)] encapsulates keys and information about keys, such as key lifetimes and cryptographic policies, so as to allow all significant aspects of the security to be modified to the needs of the application and environment. In the current Internet Security Architecture, however, SA management is peer to peer as depicted in the Figure 1.

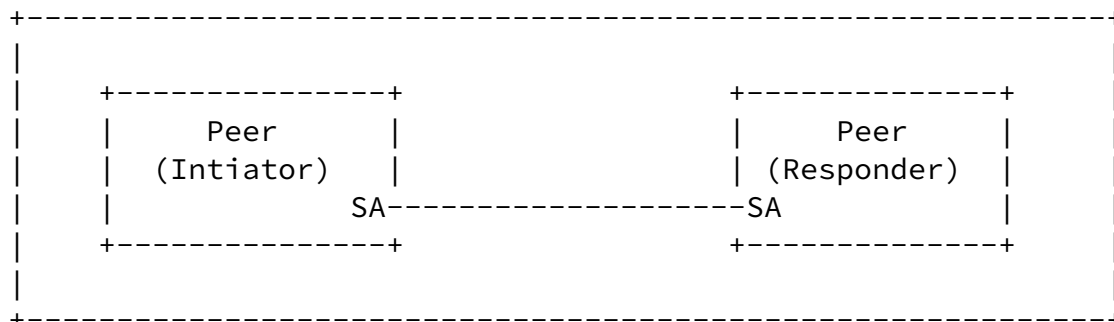


Figure 3: A Security Association between two Internet computers

The SA is defined to be simplex in the Internet Security Architecture [[RFC2401](#)] and is identified by a Security Parameter Index (SPI) [[RFC2401](#), [RFC2522](#)]. SAs are established according to local policy [[RFC2401](#), [SDNS88](#)] using exchanges that are designed to protect against basic key establishment attacks, such as man in the middle, connection hijacking, replay/reflection, and denial of service [[RFC2408](#)]. Although the first three types of attacks are the subject of authenticated key exchange mechanisms, protection against the denial-of-service attack uses a pairwise cookie mechanism [[RFC2522](#)] between peer entities, which appears used in the ISAKMP header for all exchanges [[RFC2408](#), [RFC2409](#)].

Since we assume that group key management must operate across diverse internetworks, particularly IP multicast networks, then at least some of the properties of Internet key management are required for group key management. These properties, broadly stated, are summarized in the points below.

1. Protection against man-in-the-middle, connection-hijacking, replay/reflection, and denial-of-service attacks.
2. Selectable level of security protection in key establishment, such as alternative transforms, optional PFS and identity protection to support heterogeneous internet applications and computers.
3. Alternative authentication mechanisms such as shared key, PKI, and public key to support diverse trust models.
4. Forward migration path for new security mechanisms such as new cryptographic transforms and even new exchanges.
5. A single key management framework to support the establishment of Security Associations according to the local policies of internet host and intermediate systems.

We assume that these properties should be properties of group key management as well. As discussed in [section 2.2](#), group key management has additional needs beyond the five points summarized here.

## [2.2](#) Group Key Management

From the previous section, it is clear that many of the requirements and design features of Internet key management are needed by group key management. In fact, many of the payloads, exchanges, and transforms found in ISAKMP and IKE may be suitable for group key management: Many group key management protocols and algorithms, moreover, such as GKMP, LKH, OFT, GSAKMP, NARK and MARKS assume a unique key for a member, which is established using point-to-point procedures with a key server [RFC2093, [RFC2094](#), [RFC2627](#), BMS99, HH99b, BF99, Bris99]. For the purposes of authenticating a potential group member and initializing it with keys, group-keying material must be "pulled" by an individual client from the server. Group members whose computers are off-line during key updates also must pull keying material to be re-initialized (or to request re-initialization by the GCKS) in a secure, probably point-to-point protocol. Use of IKE, unchanged with the IPsec DOI, however, is out of the question owing to the need to support key distribution in addition to exchange (i.e., an external key is given to



the member by the GCKS), the need for policy distribution rather than policy negotiation, and the use of multicast communications to push key updates to promulgate key changes needed to refresh keys that reach the end of their cryptographic lifetime and to replace keys resulting from changes in group membership. Several algorithms have been proposed to efficiently accomplish group re-key and maintenance [RFC2627, BMS99, HH99b, Bris99]. A versatile group key management building block will support a variety of alternative algorithms to offer a forward migration path when new algorithms are developed or flaws in existing algorithms are uncovered.

The use of a multicast service to "push" key updates and other control messages from the GCKS to members relieves the GCKS of the burden of contacting each member individually to change the key or the configuration of the group [CP99, HH99a, [RFC2627](#), BMS99, HH99b]. In this way, group key management can scale to very large numbers of members. This ability to deploy multicast itself for group key management is attractive for a variety of applications. This property may be superfluous for pure "pay-per-view" sessions where the member is keyed once and never again for duration of the session. But for "subscription" sessions or sessions where keys must be changed, a good multicast application design principles will protect the GCKS from being the target of periodic, and possibly synchronized, requests from large numbers of members attempting to pull keys.

Unlike large-scale subscription groups, short-lived, dynamic groups, which are characterized by relatively small numbers of members, may need group key management to minimize the time it takes to create and add members to a group. Thus, group key management must be able to

Harney, Baugher, Hardjono

[Page 8]

efficiently maintain very large, secure groups, to support large numbers of members, while not precluding fast initialization, maintenance, and destruction for smaller groups that engage in impromptu group communications [CP99, [RFC2627](#), HH99b]. The need to support a range of performance and scalability needs for diverse applications is very much a goal of Internet key management that is shared by group key management.

It is clear, however, that the security associations for group key management are more complex, or at least more numerous, than for Internet key management. Whereas the latter establishes a key management SA to protect application SAs (where a minimum of two are needed to key an Internet application process), group key management requires at least three: There is a "pull" SA between the group member

and the GCKS, a "push" SA between the GCKS and all the group members, and an SA to protect application data from sender-members to receiver-members. In fact, each sender to the group may use a unique key for their data and use a separate SA: there may be more SAs than there are group senders.

Group key management, therefore, uses a different set of abstractions than ISAKMP and IKE. The abstractions used in our Group Key Management Building Block (GKM-BB), however, may be built from the ISAKMP abstractions: in our approach, the Group Security Association (GSA) includes the attributes of the Internet Security Architecture SA, which is succinctly defined as the encapsulation of keys and policies [[RFC2409](#)] as follows.

- o An SA has selectors, such as source and destination transport addresses.
- o An SA has properties, such as an security parameter index (SPI) or cookie pair, and identities.
- o An SA has cryptographic policy, such as the algorithms, modes, key lifetimes, and key lengths used for authentication or confidentiality.
- o An SA has keys, such as authentication, encryption and signing keys.

As is discussed in the next section of this memo, a GSA contains the SA attributes plus some additional ones. As shown in Figure 4 (a), the GSA is a superset of the SA.

- o A GSA has group policy attributes, such as the kind of signed credential needed for group membership and whether the group will be given new keys when a member is added (called "backward re-key" below) or whether group members will be given new keys when a member is removed from the group ("backward re-key").
- o A GSA has SAs as attributes.

Harney, Baugher, Hardjono

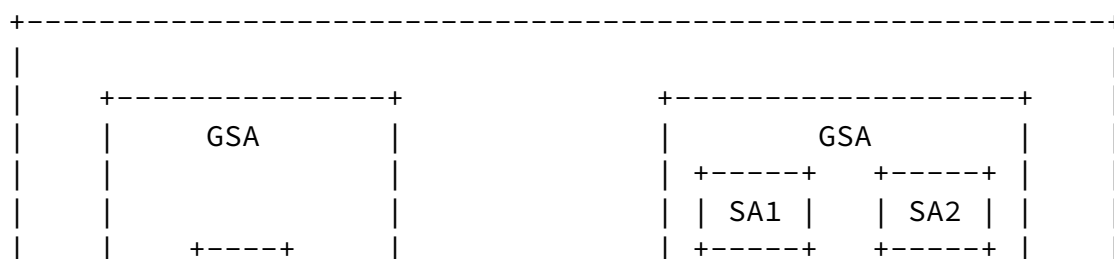
[Page 9]

INTERNET-DRAFT

Group Security Association

September 2000

The final point, a GSA includes multiple SAs, is graphically depicted in Figure 4 (b) and discussed more fully in the next section.



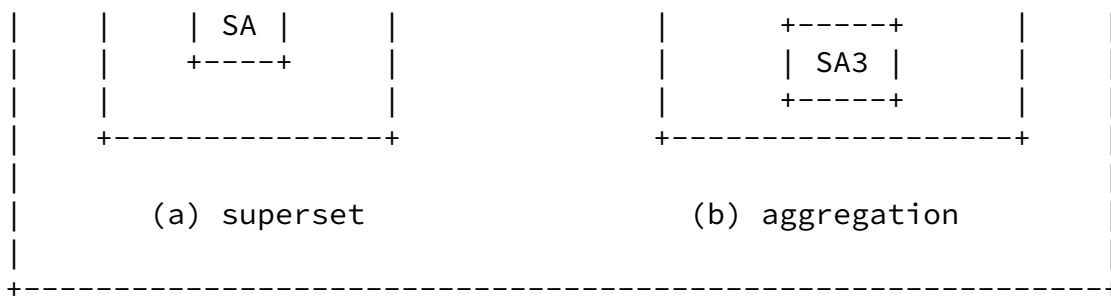


Figure 4: Relationship of GSA to SA

The following lists summarize the desired properties of Internet group key management.

1. The five properties of Internet key management as described in the previous section.
2. Support for the IRTF Secure Multicast Reference Framework having a GCKS that controls access to the group of sending and receiving members according to the group policy it distributes.
3. Support for IP multicast applications where there may be one or more senders to the group who may each have a unique SA to the group or who may each share a common SA to the group.
4. Support for both receiver-initiated "pull" of policy and keying material in addition to server-initiated "push" using a variety of re-key algorithms.
5. Selectable level of performance for group key management, which permits tradeoffs in startup latency, re-initialization complexity, message overhead, join latency, leave latency, and other security-related performance such as transforms.

Group key management requires a protocol with the five properties listed above. The protocol must be capable of establishing security relationships that are not just peer-to-peer but also between GCKS and a group of members (e.g., for re-key) and among sending and receiving

members (e.g., for data protection). This section suggested that these relationships might be built upon group security associations, which in turn build upon the security association concept of IPSec and ISAKMP/IKE, as described in the next section.

### 3. GROUP SECURITY ASSOCIATION: DEFINITION AND EXAMPLES

In this section we describe further the structure of a GSA and provide a definition of a GSA.

### 3.1 Structure of a GSA: Reasoning

There are three categories of SAs aggregated into a GSA in Figure 4(b). We choose this structure to better realize a GSA in our key management environment, the SMuG Reference Framework [[HCBD00](#)]. There is a need to maintain SAs between a Key Server and a group member (either a sender, a receiver or both) and among members. In the SMuG Reference Framework, the Key Server is called the "GCKS," which is charged with access control to the group keys, with policy distribution to client members or prospective members, and with group key dissemination to sender and receiver client members. This structure is common in many group key management environments [HH99a, HH99b, CP99, [RFC2627](#), BMS99, Bris99]. There are two SAs established between the GCKS and the members, and there is an SA established among the sending and receiving members as shown in Figure 5.

The first category of SA (namely SA1 in Figure 5) is initiated by the member to pull GSA information from the GCKS; this is how the member requests to join the secure group or has its GSA keys re-initialized after being disconnected from the group (e.g., when its host computer has been turned off during re-key operations as described below). The GSA information pulled down from the GCKS include the SA, keys and policy used to secure the data transmission between sending and receiving members; this is SA3 in Figure 5. Note that SA3 is a category of SA, and this implies that there may be multiple SAs established between member senders and member receivers - at least as an option. There may exist, for example, a single SA of category SA3 in which all senders share common keys and associated information. Or there may be one or more SAs of category SA3 that are unique to the particular sender. An SA3 security association may be reestablished or have its keys modified through re-key operations, which occur over an SA of category SA2. Keys are pushed through an SA of category SA2 to support subscription groups.

Thus, despite the fact that the data to be protected are multicast, pull exchanges through an SA1 should be unicast or point-to-point key determination exchanges. Some group key management solutions rely

solely point-to-point. Most others combine unicast exchanges for initialization with multicast distribution for re-key. In some cases, such as in a pure "pay-per-session" application, all of the SA information needed for the session may be distributed at the time of registration or selection of a session, i.e. over an SA1; re-key and re-initialization may not be necessary, so there is no SA2. For subscription groups where keying material is changed as membership changes, an SA2 is needed to re-initialize an SA3.

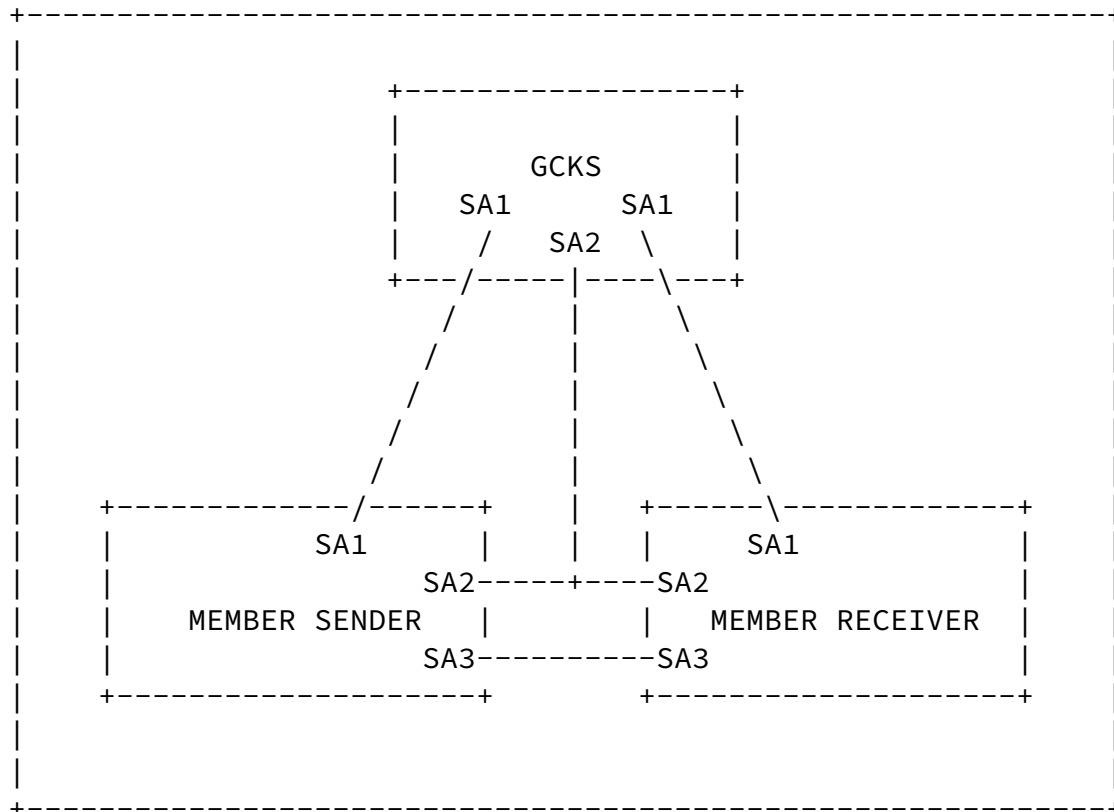


Figure 5: GKM-BB GSA Structure and 3 categories of SAs

### [3.2](#) Definition of GSA

The current GKM Building Block defines a GSA to include an aggregate of three (3) categories of SAs. The three categories of SAs correspond to the three kinds of communications as seen from the point of view of the Receiver (Member). Figure 5 depicts this concept:

- Category-1 SA:

a SA is required for (bi-directional) unicast communications between the GCKS and a group member (be it a Sender or Receiver). This SA is established only between the GCKS and a Member. In the SMuG Reference Framework, the GCKS entity is charged with access control to the group keys, with policy distribution to members (or prospective members), and with group key dissemination to Sender and Receiver members. This use of a (unicast) SA as a starting point for key management is common in a number of group key management environments [HH99a, HH99b, CP99, [RFC2627](#), BMS99, Bris99].

Note that this (unicast) SA is used to protect the other elements of the GSA (such as the other following two categories of SAs), either in a "push" or "pull" model. As such, this SA is crucial and is inseparable from the other two SAs as the definition of a GSA.

From the perspective of one given GCKS, there are as many unique Category-1 SAs as there are members (Senders and/or Receivers) in the group. Thus there may be a scalability concern for some applications, so a Category-1 SA may be used on-demand whereas Category-2 and Category-3 SAs are established at least for the life of the sessions that they support.

- Category-2 SA:

a SA is required for the multicast transmission of key-management messages (unidirectional) from the GCKS to all group members. As such, this SA is known by the GCKS and by all members of the group.

This SA is not negotiated, since all the group members must share it. Thus, the GCKS must be the authentic source and act as the sole point of contact for the group members to obtain this SA.

From the perspective of each participant in a group (GCKS and all members), there is at least one (1) Category-2 SA for the group. Note that this allows for the possibility of the GCKS deploying multiple Category-2 SA for group key management purposes.

- Category-3 SA:

one or more SAs are required for the multicast transmission of data-messages (unidirectional) from the Sender to other group members. This SA is known by the GCKS and by all members of the group.

Similarly, regardless of the number of instances of this third category of SA, this SA is not negotiated. Rather, all group members obtain it from the GCKS. The GCKS itself does not use this category of SA.

From the perspective of the Receivers, there is at least one Category-3 SAs for the member sender (one or more) in the group. This allows for the possibility of including group IDs (GID) in transmission of data packets from the senders in the group.

There are a number of possibilities with respect to the number of Category-3 SAs and the use of GIDs:

- (i) Each sender in the group could be assigned a unique Category-3 SA, thereby resulting in each receiver having to maintain as many Category-3 SA as there are senders in the group.
- (ii) The entire group deploys a single Category-3 SA for all senders, together with the use of GIDs. Receivers would then be able to filter based on the GIDs, whilst maintaining only one Category-3 SA.
- (iii) A combination of (i) and (ii) above.

### [3.3](#) Forward and Backward Rekey

The re-key operation is needed to ensure that messages sent to the group cannot be accessed by a former member whose membership has been revoked by the GCKS; some applications may also require that a member who joins a group be denied access to messages that were sent to the group prior to its membership [[CP99](#), [HH99a](#), [BMS99](#)]. We call the first case, "forward rekey," when a key change is prompted by a member leaving the group, and the latter is called "backward rekey," when a re-key is caused by a new member joining the group. Note that the terms "forward/backward secrecy" and "forward/backward security" have been used in the literature [[CP99](#), [HH99a](#), [BMS99](#), [HH99b](#)].

### [3.4](#) Group-Key Determination Algorithms

In order to efficiently implement re-key so that the complexity of adding and removing members can be done in better than linear time, i.e. without the need for unicast exchanges with each member, a structure may be imposed on the SA2 keying material. The most efficient algorithms to date achieve  $O(\log n)$  complexity for the re-key operation [[WL98](#), [RFC2627](#), [BMS99](#), [Bris99](#)]. In such algorithms, SA2 has a tree structure of keying material (beyond a list of keys as used in IKE), such as a logical key hierarchy [[RFC2627](#)] or one-way function tree [[BMS99](#)]. These approaches do not rely on a trusted execution environment, such as a smart card installed on a member computer, so the member is trusted to

INTERNET-DRAFT

Group Security Association

September 2000

remove itself upon command from a key server [BF99]. Such a 'vertical' approach, such as integrating a smart-card into the re-key operation assumes too much about the operating environment for a general-purpose internet solution. Instead, the SA2 structure (which is dependent on the particular key determination algorithm), must change the keys of the members of the group in a manner that is as efficient and free of collusion [CP99, BMS99] as required by the particular application.

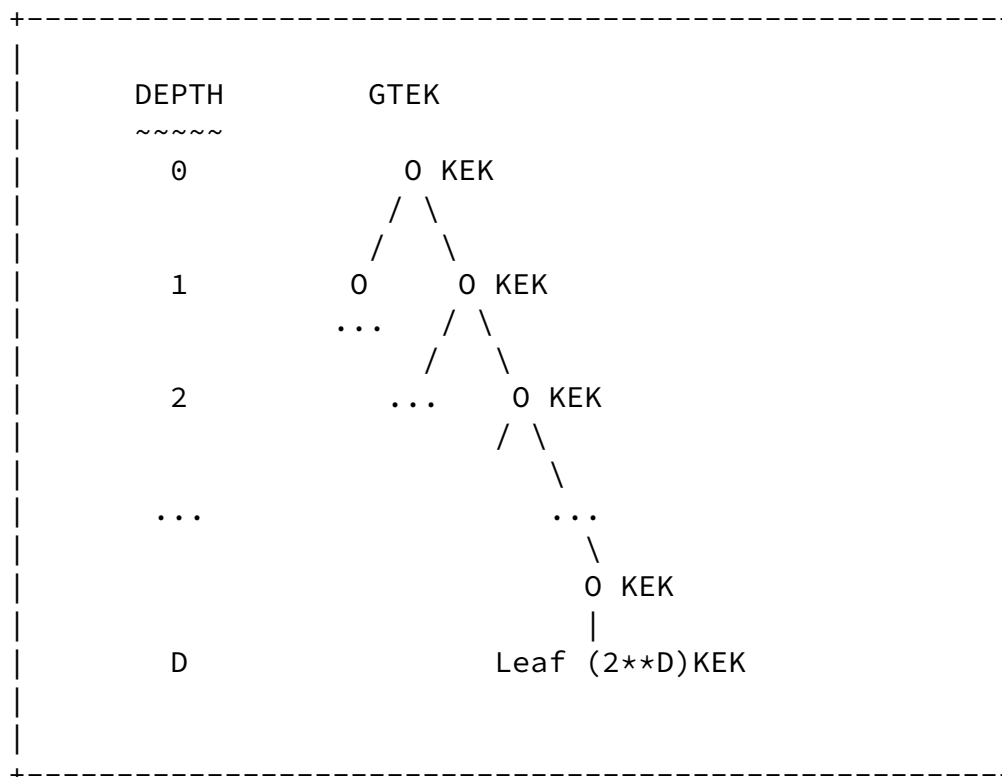


Figure 6: A GSA Key Structure

In RFC 2627 and one-way function trees, for example, each member has a unique leaf key, the knowledge of which is shared only with the key server, which generates the group traffic encrypting key or GTEK (the "net key" in RFC 2627 parlance). In OFT, the key used to encrypt session traffic is the root of structure, which is called the "root KEK" in this memo). The KEK array of Figure 4 is an SA2. The GTEK belongs to an SA3, and the SAs of category SA2 and SA3 are initialized over an SA1, shown in Figure 3. The keys or material to create the keys of SA2 and SA3 are externally generated by the GCKS, which may be determined in a manner similar to the IEXTKEY procedure of OAKLEY [RFC2412].



INTERNET-DRAFT

Group Security Association

September 2000

#### [4.](#) SUMMARY AND FUTURE WORK

This memo contributes to the SMuG Reference Framework and Building Blocks effort [[HCB00](#)], which seeks to develop the foundations for secure multicast standards in the near future. The group key building block will provide a key management solution for Problem Area 2 of the SMuG Reference Framework though this memo only proposed a set of properties and abstractions for group key management. The ultimate goal of this effort is to define the framework so it can be implemented in one or more protocol instantiations. In order to progress to the point of worthy specifications and working implementations, several questions must be answered.

1. What framework should be used for the group key management building block?
2. How many of each category of SA should be allowed in a GSA?
3. What transport should be used for Category-2 SA key management control messages?

The first question asks whether the Internet key management framework, ISAKMP, should be used or whether some invented framework should be used to express, specify, and/or implement group key management.

The second question that must be answered is how many SAs of Category-2 and Category-3 must the group key management framework support? This issue has ramifications for how complex the framework will be in terms of messages and payloads. Multiple Category-3 SAs, for example, may be used to bundle keying material for multiple, related groups such as for multimedia sessions [[RFC1889](#)]. A related question concerns GSA updates: are operations needed to modify existing SAs? Such operations may be very complex and may entail changes to group policy, which may have significant ramifications on access control. Re-key algorithms such as LKH and OFT update SAs by modifying keys. Whereas TLS supports operations to change the cipher, IKE requires that a new SA be created and the old SA deleted as the means by which an SA is modified.

The third question is the transport to be used for Category-2 SA messages which are multicast and which have reliability requirements. Should a reliable multicast services be assumed? Should it be

integrated into the protocol? More consideration is needed on the effects of providing a multicast key management services to groups of members, large and small, static and dynamic.

It is our intention to address these questions in the process of developing a specification for the group key management building block in a subsequent draft.

Harney, Baugher, Hardjono

[Page 16]

---

INTERNET-DRAFT

Group Security Association

September 2000

## REFERENCES

[BF99] B. Briscoe, I. Fairman, Nark: Receiver-based Multicast, Non-repudiation and Key Management, Proceedings of ACM E-Commerce'99, rbriscoe@bt.co.uk.

[BMS99] 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.

[BR93] M. Bellare, P. Rogaway, Entity Authentication and Key Distribution, Advances in Cryptology - Crypto '93 Proceedings, Springer-Verlag, 1993.

[Bris99] B. Briscoe, MARKS: Zero Side Effect Multicast Key Management using Arbitrarily Revealed Key Sequences, Proceedings of NGC'99, rbriscoe@bt.co.uk.

[CP99] R. Canetti and B. Pinkas, A taxonomy of multicast security issues, <http://www.ietf.org/internet-drafts/draft-irtf-smug-taxonomy-01.txt>, August 2000, Work in Progress.

[DVW92] Diffie, P. van Oorschot, M. J. Wiener, Authentication and Authenticated Key Exchanges, Designs, Codes and Cryptography, 2, 107-125 (1992), Kluwer Academic Publishers.

[FS00] N. Ferguson and B. Schneier, A Cryptographic Evaluation of IPsec, CounterPane, <http://www.counterpane.com/ipsec.html>.

[HCB00] T. Hardjono, R. Canetti, M. Baugher, P. Disnmore, Secure IP Multicast: Problem areas, Framework, and Building Blocks, <http://www.ietf.org/internet-drafts/draft-irtf-smug-framework-01.txt>, Work in Progress, Sept 2000.

[HCD00] 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-03.txt>, Aug 2000, Work in Progress.

[HH99a] H. Harney, E. Harder, Multicast Security Management Protocol (MSMP) Requirements and Policy, [draft-harney-msmp-sec-00.txt](http://draft-harney-msmp-sec-00.txt), March 1999, Work in Progress.

[HH99b] 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.

Harney, Baugher, Hardjono

[Page 17]

---

INTERNET-DRAFT

Group Security Association

September 2000

[Kraw96] H. Krawczyk, SKEME: A Versatile Secure Key Exchange Mechanism for Internet, ISOC Secure Networks and Distributed Systems Symposium, San Diego, 1996.

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

[RFC2093] Harney, H., and Muckenhirn, C., "Group Key Management Protocol (GKMP) Specification," [RFC 2093](http://rfc2093.txt), July 1997.

[RFC2094] Harney, H., and Muckenhirn, C., "Group Key Management Protocol (GKMP) Architecture," [RFC 2094](http://rfc2094.txt), July 1997.

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

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

[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.

[RFC2412] H. Orman, The OAKLEY Key Determination Protocol, November 1998.

[RFC2522] P. Karn, W. Simpson, Photuris: Session-Key Management Protocol, March 1999.

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

[SDNS88] H. L. Rogers, An Overview of the CANEWARE Program, 10th National Security Conference, National Security Agency, 1988.

[WL98] C.K.Wong, S.S. Lam, Digital Signatures for Flows and Multicasts, Proceedings of IEEE ICNP'98, October 14-16, 1998.

Harney, Baugher, Hardjono

[Page 18]

---

INTERNET-DRAFT

Group Security Association

September 2000

Authors Address:

Hugh Harney  
SPARTA, Inc.  
Secure Systems Engineering Division  
[9861](#) Broken Land Parkway, Suite 300  
Columbia, MD 21046-1170, USA  
+1 410 381 9400 (ext. 203)  
hh@columbia.sparta.com

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

Thomas Hardjono  
Nortel Networks  
[600](#) Technology Park Drive  
Billerica, MA 01821, USA  
(978) 288-4538  
thardjono@baynetworks.com

