

SPARTA, Inc.
INTERNET-DRAFT
draft-harney-sparta-lkhp-sec-00.txt

Hugh Harney, Eric Harder
SPARTA, Inc., National Security Agency
March, 1999

Logical Key Hierarchy Protocol

Status of this memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

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

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

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

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

Document expiration: August 30, 1999

Abstract

This document presents a Logical Key Hierarchy (LKH) Compromise Recovery (CR) implementation for the key management protocol suggested in the paper "Key Management for Multicast: Issues and Architectures"[10]. The goals of this paper include: defining the CR method, identifying the requirements, defining the operational protocol, and recommending an implementation for an LKH CR implementation.

INTERNET-DRAFT

LKH Protocol

March, 1999

Copyright Notice

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

Contents

1	Background	4
1.1	Security for Multicast	5
2	Compromise Recovery	5
2.1	Definitions	5
2.2	Compromise Recovery Policy	6
2.2.1	Restoration of Secure Network Operations	6
2.2.2	Restrict Compromise Recovery Actions to Authorized Individuals	6
2.2.3	Multiple Co-Located Compromises	6
2.2.4	Secure Compromise Recovery Life-Cycle	7
2.2.5	System Stability After a Compromise	7
2.3	Requirements	7
2.3.1	Generation of LKH Arrays	7
2.3.2	Generation of Support Materials	7
2.3.3	Secure Compromise Recovery	7
2.3.4	Normal Operation Requirements	8
3	LKH CR Protocol Specification	10
3.1	Group Establishment	10
3.1.1	LKH Establishment Protocol	11
3.2	CR Policy and Enforcement	14
3.2.1	Compromise Event	14
3.2.2	Single Message to Exclude Compromised Member	18
3.2.3	New Group Key	18
4	RECOMMENDATIONS	19
4.1	Develop Multicast Framework	19
4.2	Computer Trust Requirements	20
5	Addresses of Authors	22

1 Background

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

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

Group communications can also be obtained using broadcast methods, though these methods tend to be simplex (one-way) in nature. In this case, the sender simply broadcasts his information and each receiver must determine if the message is of interest to them. This approach is inefficient due to the processing required at each broadcast recipient to filter out the wanted from unwanted messaging. Broadcasting is also not practical with some networking technologies (e.g., packet switched).

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

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

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

The decision to create a cryptographic group on the internet is a decision based on the data that is going to be passed across the network and the needs of the communicating group. If the data being passed across the network is extremely important and not time sensitive, the security policy for creation, dissemination, and access control may be stringent. Alternately, if the data is not very sensitive, the security policies of the group may be more relaxed. This is an important distinction because there is a trade-off between security (assurance that the policy is in effect) and performance (time and resources necessary to implement the policy). The job of coordinating that trade-off falls to a management protocol.

1.1 Security for Multicast

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

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

This document presents a Logical Key Hierarchy (LKH) Compromise Recovery (CR) implementation for the key management protocol suggested in the paper ``Key Management for Multicast: Issues and Architectures''[10]. The goals of this paper include: defining the CR method, identifying the requirements, defining the operational protocol, and recommending an implementation for an LKH CR implementation.

2 Compromise Recovery

2.1 Definitions

For the purpose of this document, a group is a gathering of communicating members with a single key. If the group key is compromised, then secure communication must be restored through a recovery action. A compromise occurs when a member of the group can no longer be trusted (e.g. group member loses their key or a group member's key is stolen). When this happens, the group needs to change the compromised keys, without giving

the new keys to the compromised member. This document proposes as LKH CR protocol for this purpose.

2.2 Compromise Recovery Policy

A compromise is an event that makes a trusted member of a group untrusted and untrustable. All keys in that member's possession are considered ``lost``. Many different events may trigger a compromise including: equipment loss, discovery of inappropriate data transfer and theft. Compromising events are defined by the CR Policy. The CR Policy must be defined and understood prior to the compromise. The policy should define what type of compromise requires recovery, the speed with which recovery must be completed, and the level of acceptable risk to the system.

The following sections identify the minimum CR Policy assumptions that would be necessary to support the LKH CR protocol presented in this document.

2.2.1 Restoration of Secure Network Operations

Network operations should be restored with a minimal number of messages and with minimal delay. The goal of recovery is to resume operations in a secure mode quickly and efficiently. The size of the LKH array and the extent of the compromise will determine the number of messages required to recover the LKH.

2.2.2 Restrict Compromise Recovery Actions to Authorized Individuals

The CR process changes the group membership and common group keys. An unauthorized CR action could subvert the group into communicating with unauthorized individuals or be used to deny service to the network. In order to prevent unauthorized CR actions and reduce system vulnerability, only authorized individuals should be allowed to identify that a compromise has occurred, assess the risk, and implement the necessary CR action.

2.2.3 Multiple Co-Located Compromises

The CR process shall provide mechanisms to allow recovery from single- and multiple-entity compromises. Historically, compromises have occurred due to the breach of physical security measures at a particular location. In a group environment, it is possible that several group members will be physically co-located. The CR process should be capable of dealing with multiple co-located compromises.

2.2.4 Secure Compromise Recovery Life-Cycle

The entire life-cycle of the CR process must be secure. This includes the generation of CR materials, establishment of the CR group, execution of recovery from an event, and termination of CR for a group.

2.2.5 System Stability After a Compromise

The outcome of any compromise event and the resulting CR action must leave the group capable of recovering from another compromise.

2.3 Requirements

The following sections present the requirements necessary to support the LKH CR protocol.

2.3.1 Generation of LKH Arrays

The LKH array generation mechanisms, as well as the LKH arrays, must be protected from unauthorized access. The CR Manager will have the only access to these mechanisms. Further, the computers on which these mechanisms reside must also be sufficiently protected from unauthorized access.

2.3.2 Generation of Support Materials

The LKH CR process will be supported by certificates. The certificate registration process must provide mechanisms to ensure that there is unambiguous identification of individuals and authorities. The mechanisms and processes within the certificate registration process must also be verifiable and protected from unauthorized access and disclosure.

2.3.3 Secure Compromise Recovery

The CR process includes the exchange of sensitive materials (LKH key array). To ensure that the compromise recovery process is secure, it must include mechanisms for:

1. Identifying all group members;

2. Identifying all CR agents;
3. Verifying the authority for all sensitive acts;
4. Verifying the integrity of all data exchanges;
5. Protecting all information that could be used to attack the CR system;
and
6. Verifying the assurance level of all CR computer components.

2.3.4 Normal Operation Requirements

The requirements identified in this section support the distribution, management and storage of the LKH arrays prior to a compromise event. These requirements must also be fully satisfied upon completion of a CR action.

2.3.4.1 Minimal Exposure of LKH Arrays

The LKH arrays are sensitive information and, if left unprotected, can be used to attack or compromise the secure group. The use of routers and other network components to distribute the LKH arrays should take place with the understanding that the compromise of the component could lead to group attacks. In order to help minimize the risk of exposure, the LKH arrays should be held by as few computers as possible. Each CR Manager that maintains an LKH array must provide adequate physical, procedural, and computer trust protection mechanisms to protect the array.

2.3.4.2 Authentication of Identities

For all key management actions, the identities of the receiving and sending parties need to be mutually understood. This requirement could be met by verifying the public key of a party during key generation.

2.3.4.3 Verification of Authorization

The management actions supporting CR are critical to group security. The identification and participation authorization of each group member involved in a critical action must be verified. This requires a clear security policy understood across the group. This policy can be static or dynamic based on a policy dissemination mechanism.

2.3.4.4 Computer Security Trust Requirements

The LKH key arrays are critical. If these arrays reside unencrypted on a computer at any time, then that computer must be trusted to protect the group's data. This requirement would be supported by maintaining only the LKH arrays on group member's computers. Each group member's computer must be trusted to protect the group's data.

2.3.4.5 Cryptographic Structure of Groups

It is anticipated that very large groups may choose to implement this LKH CR technique to support their CR process. In order to provide the best management and oversight, the large group should be constructed as the union of multiple smaller groups. Each of the smaller groups will have its own LKH array structures. These smaller LKH array structures will provide the capability to:

1. Localize the CR processes;
2. Generate and maintain smaller LKH arrays;
3. Localize the CR reporting procedures;
4. Localize CR management; and
5. Implement cryptographic gateways to minimize any single group traffic key exposure.

2.3.4.6 CR Message Requirements

There are security requirements placed on CR messages to ensure that the messages are sent and received by the individuals with the appropriate authority levels. CR messages should provide the following information:

1. Source verification;
2. Authority verification;
3. Confidentiality of data from unauthorized access;
4. Verifiable integrity of all data exchanges;
5. Protection of all information that could be used to attack the CR system; and

6. Verifiable assurance levels of all CR computer components.

The CR messages do not have a secure association (SA) with the group and, therefore, confidentiality must be provided within the recovery key schema.

2.3.4.7 Compromise Event Discovery and Reporting

A deliberate compromise event may be difficult to discover because it is in the interest of the attacker to keep the compromise a secret. As long as the compromise remains undiscovered, the attacker will continue to have access to the group's data. An accidental or unintentional compromise will likely be reported as soon as the action is identified. However, regardless of the source of the compromise, the CR Manager must have sufficient mechanisms established to identify a compromise. The CR Manager must then assess the extent of the compromise to identify the necessary CR actions for recovery.

All CR actions will result in a temporary disruption of the group while the group member's identities are verified and the keys are changed and disseminated. A complete discussion of compromise discovery and reporting is outside the scope of this document. Formal compromise discovery and reporting policies should be developed to support this process. The LKH CR technique presented in this document relies on input from a compromise discovery action to identify the compromised group member.

3 LKH CR Protocol Specification

The logical definition of a secure group is multiple members communicating via a common key scheme. The focus of the LKH CR protocol is the CR protocol of this group. The methods used to create, distribute, verify, and authenticate the common group key are outside the scope of this document.

3.1 Group Establishment

A large group can be serviced by several independent CR Agents each controlling a subset of the CR domain. This architecture distributes the processing and communication requirements of CR actions across the group thus avoiding communication and processing bottle-necks.

For example, in an hierarchical tree CR protocol, a 2-layered LKH structure with the CR Manager, ``Member 1``, at the top. The second layer members (1.1 to 1.5) are all subject to CR actions initiated by ``Member 1``. Each of the second tier members themselves have sub-nodes and hence, have LKH

databases.

The top node in the LKH is identified as ``Member 1`` for the discussion in the following sections. ``Member 1`` (Node 1 in the LKH schematic) will act as the CR Manager. The second level nodes act as CR Agents.

3.1.1 LKH Establishment Protocol

The CR Manager and each of the CR Agents, will either create or obtain the LKH databases and distribute the appropriate keys to the members in their domain. These keys are extremely important to the continued secure operation of the group. As such, the distribution of these keys needs to be a secure process. The keys must be kept confidential. All the members need to have an unambiguous identification of any party downloading keys to them. The CR Manager and CR Agents need to have unambiguous identification of the members to which they will distribute keys. The members need to verify that the CR Manager and CR Agents are authorized to act in those roles. Each of these requirements is similar if not exactly equal to the requirements needed to distribute the original group key. To avoid redundancy of action, wherever possible, the CR data will be distributed with the symmetric key.

3.1.1.1 Generation of LKH Array

The CR Manager generates the keys needed to construct a LKH array. There are two methods for generating an LKH array for very large groups. First, the CR Manager could generate a very deep array capable of encompassing all the potential members of the group. Second, the CR Manager can generate a smaller array capable of recovering the first tier of a group. These first tier members can act as delegated CR Agents, each generating LKH arrays for group members in branches below them.

Of the two methods for generating an LKH array for large groups, the delegated CR mechanism provides greater scalability. The issue with this delegated approach is the CR Manager and CR Agents must be identified to the group members prior to the establishment of a group. One mechanism for accomplishing this is the use of a group policy token. The CR Manger and CR Agents could be identified and a single group authority could authorize them.

This method of establishing LKH arrays will be illustrated in the following specification. The CR Manager generates and stores the list of keys.

3.1.1.2 Distribution of LKH Array to Group Members

The distribution of LKH array material should involve a peer-to-peer session association. The security of the group is built from a collection of peer-to-peer access control decisions.

It is important to note that all access control decisions do not need to be made by the CR Manager or any central point. A multicast group can easily be constructed by a number of peer-to-peer access control decisions. The critical issue is to ensure that the access control decisions are made by members authorized to make such a decision for the group.

The identity of group access control points is a matter for group policy. The simplest policy is that one site makes all group access decisions. A more scaleable solution identifies multiple points authorized to make these decisions for the group. An even more scaleable solution allows any group member (with proof that they have been accepted into the group) to make an access control decision for the group. Essentially, known group members could be allowed to vouch for new group members.

In the case of distribution of CR material, the generator of the LKH array could distribute pieces of the array to authorized distribution points within the group for subsequent distribution.

3.1.1.2.1 Establishment of a Secure Association Modern SA protocols like the Internet Secure Association Key Management Protocol (ISAKMP) are suited to this task. The security characteristics of the establishment protocol for the SA should include:

1. Verification of all identities;
2. Validation of public certificates (if used);
3. Creation of a pairwise traffic confidentiality key; and
4. Transfer of identity and certificate information to multicast security management protocol.

Once the SA is established, the multicast security management protocol can use that SA for secure confidential communications.

3.1.1.2.2 Data Structure of LKH Array Download Message When the identities of each side of the SA are known to the multicast security protocol, the multicast security protocol can use these identities along with the verified information on the public certificates to enforce group security policy. The group security policy includes information about authorized

CR mechanisms and distribution authorities. The management protocol needs to verify that the CR Manager is authorized to download the LKH CR arrays. The management protocol will take the identity and authority information verified in the establishment of the SA and make sure that it meets the policy criteria.

In order to verify authority, there has to be something that identifies authorized CR Agents. This could be an internal configuration file or it could be a data structure that dynamically conveys a policy from an authorized source. In the case of a data structure, this policy data structure (token) could be sent with the LKH array.

The data structure of the LKH array and (optional) Policy Token is:

2 bytes	Data ID
16 bytes	Grp ID
8 bytes	Date/Time
1 byte	Sig Alg ID
2 bytes	Cert Infra ID
1 byte	LKH Version
4 bytes	LKH array length
Variable	LKH array
1 byte	(policy token)
4 bytes	(# packets)
Variable	(Packets 1-n)
1 byte	(Pub Cert)
2 byte	(# Packets)
1 byte	(Packets 1-n)
Variable	Sig

where:

Data ID:	Identifies the packet as a CR data item. 1=LKH array, 2=LKH array with policy token, 3=LKH array with policy token and public certificate of signer
Grp ID:	Specifies the group the LKH array will address
Date/Time:	Self-explanatory
Sig Alg ID:	Specifies the signature algorithm used in this data item
Cert Infra ID:	Specifies the certificate infrastructure needed for verifying signature
LKH ver:	Version of the LKH CR protocol
LKH Array Length:	Total number of bytes in the LKH array
LKH Array:	Data

() signifies optional fields

(Policy Token):	Identifies the Policy Token
(# Packets):	Self-explanatory
(Policy Token Packets 1-n):	Self-explanatory
(Pub Cert):	Identifies public certificate data being sent
(# Packets):	Self-explanatory
(Packets 1-n):	Self-explanatory
Sig:	Signature field as identified earlier

3.2 CR Policy and Enforcement

3.2.1 Compromise Event

The CR Manager, designated as Member 1, manages compromises occurring in the second tier of the hierarchy. In the second tier member designations (eg., 1.1), the number to the left of the decimal refers to the CR Manager. The number to the right of the decimal is the unique identifier of the CR Agent. The third tier is comprised of group members who do not act as CR Agents. In the third tier designations (eg., 1.1.5), the first number refers to the CR Manager. The second number designates the second tier CR Agent that owns the domain on which this particular member resides. The third number, in this example 5, is a unique identifier. This numbering scheme is useful for reporting compromises and allows the member designation of the bottom tier member to uniquely identify that member, the CR Manager, and the delegated CR Agent in the hierarchy above.

3.2.1.1 Compromised Discovery

After a compromise is discovered, a compromised report is received by Member 1, the CR Manager. The contents of this message include, at a minimum, the identity of the compromised member and, in the case of a multi-tiered architecture, a path to that member.

The active member in this example is the CR Manager (Member 1). The CR Manager will verify that the CR report was received and is authentic. It will then initiate CR actions.

3.2.1.2 Recovery Protocol

The CR Manager creates the CR message based on the information that was passed within the CR report. The CR Manager sends this message to all members of its domain utilizing the multicast communication address of the group.

The CR message is sent on the group address and to the common key management port routing to the key management application resident at each member. Each member verifies that the CR message is authentic and that the signature on that message comes from a party that is authorized to send a CR message. This authorization decision is based on some known policy that has been previously configured for the group. One mechanism that is useful for configuring group policy is a policy token.

Each CR message will contain a Date/Time stamp. A CR action may only be processed if its Date/Time stamp is later than the Date/Time stamp of the last CR action processed for the group. In the event of multiple CR actions, the CR messages should be processed in ascending order according to their date/time stamp.

After the CR message has been verified, each member will decrypt their portion of the CR message. That single CR message will recover some portion of the compromised group and provide the first tier members the means to the reset group traffic keys to a new secure key. Any member in the path of a compromised member and will be unable to decrypt the new group traffic key.

The LKH shown in Figure 1 represents virtual nodes as letters and members nodes as numerals. Assuming Member 1 is compromised, the symbolic form of the recovery message is:

```
CompHdr{[Sec HdrB(MGK')B], [Sec HdrD(MGK',A')D], [Sec Hdr2
(MGK',C',A')2]}Siglkhc
```

Notation:

CompHdr{} = CR message header for message between {}

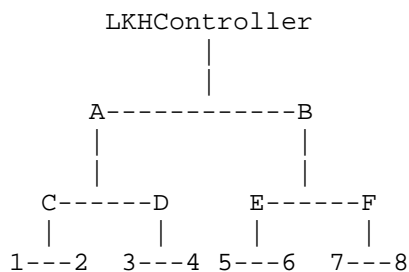


Figure 1: Sample Compromise

[SecHdr*(MGK')]. = Data packet containing a security header that allows the decryption of the data package encrypted in key * (in this case, the data packet contains MGK: Multicast Group Key Prime)

{ }SigXo = Public key signature of data contained within { }, public key to verify is Xo.

The data structure of the CR message follows:

1 byte	CR Msg ID
16 bytes	Grp ID
8 bytes	Date/Time
1 byte	Encr Alg ID
1 byte	Sig Alg ID
1 byte	Hash ID
2 bytes	Cert Infra ID
1 byte	CR Msg Type
1 byte	Tree type
1 byte	LKH Ver
4 bytes	# Packets
Variable	Packets 1-n
Variable	Sig

CR Msg ID: CR Message ID distinguishes this message from other key management messages. Suggest an integer value with: 66=CR.

Grp ID: Group identification value is a 16 byte string that identifies the specific group the CR message is to recover. Suggest this field be the group common name.

Date/ Time: This field specifies the date/time this message was sent.

Encr Alg ID: Encryption Algorithm Identification. This field specifies the exact cryptographic algorithm to be used by this message. An integer value suffices. 1=DES CBC, 2=triple DES.

Sig Alg ID: Signature Algorithm Identification. This field identifies the algorithm used to sign this message. Suggestion 1=DSS.

Hash ID: Hash Identification specifies the hash algorithm used. Suggestion 1=SHA.

Cert Infra ID: Certificate Infrastructure Identification specifies type and location of certificate infrastructures.

CR Msg Type: CR Message Type specifies type of CR message. Suggest: 1=Group Recovery, 2=Individual Recovery, 3=Maintenance, 4=Delete Group Key.

Tree Type: This specifies the mechanism used by the CR process.

LKH Ver: LKH Protocol Version. Suggestion 1=first.

Packets: Number of Packets specifies the number of information data items in the payload of the CR message.

Packets 1-n: Each individual packet will take the following form:
1 byte Packet type
4 bytes Length of packet
Variable Data

Packet type: Specifies the data within the packet. Suggest: 1=encrypted key(s), 2=cryptographic changeover time, 3=unencrypted data.

Packet Length: An integer specifying the number of bytes in the data packet.

Sig: The signature block contains the actual signature in the algorithm specified in the Sig Alg ID data field. It should take the following form:
1 byte Signature format
4 bytes Length of data
Variable Data

Signature format: This field defines the exact data contained in the data field. Suggest: 1=DSS, 2=DSS

Harney/Harder
lkhp-sec-00.txt

draft-hwithapublicrcertificatenofesigningparty.-sparta-
[Page 17]

3.2.2 Single Message to Exclude Compromised Member

The CR Manager has been notified of the compromised status of a tertiary member (eg., 1.1.1). The CR Agent in the compromise path generates a message using keys stored in its database that will exclude the compromised member from receiving the new group key.

```
CompHdr{[SecHdrB(MGK0)B];[SecHdrD(MGK0;A0)D];[SecHdr1:1:2(MGK0; C0;
A0)1:1:2]}SigX1:1
```

The CR message is sent out over the multicast communication address. All nodes in the group and in the subgroup receive that message and each authorized member decrypts the new traffic key. It is possible that the CR Agent could act as a cryptographic gateway for its sub-nodes. A cryptographic gateway changes the cryptographic traffic key for a branch of a group. A message encrypted with the group key of the second tier would come to the CR Agent who would then take all the data and re-encrypt that data in the group key common for its branch.

The CR Agent's CR message will be restricted only to group members in its domain. There are two possible scenarios to support this restriction. In the first, the CR Agent could be a cryptographic gateway and therefore would have a group address for it's branch. The group address could offer a limited distribution option to preclude external transmission. In the second scenario, the CR Agent could establish a special local group address for the branch.

3.2.3 New Group Key

When suborninate CR Agents are used all members in the path of a compromise must be brought back into the secure group. To accomplish this, the CR Manager (Member 1) creates a SA with the delegated CR Agent using a SA protocol like ISAKMP.

Once a SA is established, the CR Manager sends a combination message to the delegated CR Agent (Member 1.1). This combination message tells the CR Agent that one of a its sub-nodes has been compromised and also passes the new secure group key. The CR Agent verifies that the CR Manager is the originator of this message and then updates it's group key. The CR Agent then begins CR actions for his domain.

The symbolic form of the message is:

```
CompHdr[(SecHdr1 (MGK')1, CompRpt)Sig1]1
```

Notation:

CompHdr{} = CR message header for message between {}

SecHdr*(MGK') = Data packet containing a security header that allows the decryption of the data package encrypted in key * (in this case, the data packet contains MGK: Multicast Group Key Prime)

{ }SigXo = Public key signature of data contained within {}, public key to verify is Xo.

The message structure for this combination message is exactly the same as the group recovery message defined in Section 3.2.2. This message utilized a new data packet type, Packet Type 3, to transmit the CR report to the CR Manager.

4 RECOMMENDATIONS

4.1 Develop Multicast Framework

In the interest of standardization and efficiency, it is reasonable to propose a standard multicast security framework that could organize the establishment of security for multicast groups. In such a framework, CR would be a component of the overall security profile for a group. The LKH CR protocol could be an option for supporting CR within a standardized framework architecture.

The LKH CR protocol is part of a complete multicast security management protocol. It provides CR services for a group without respect to the distribution methodologies or underlying communication protocols. The LKH CR protocol does make some assumptions about services provided by the more general multicast security management protocol. This protocol should include mechanisms to support the following requirements:

- Verifiable and understandable security policy;
- Unambiguous identification of members;
- Verification of authority to perform relevant actions; and
- Confidential delivery of information to group members.

These basic services are fundamental to securing groups with keys. A well-orchestrated protocol should incur the overhead of these services once and pass all necessary information to the group members. The LKH CR does not implement these mechanisms due to the assumption that these services are available during secure key establishment. Implementing these services

within the LKH CR protocol is redundant.

4.2 Computer Trust Requirements

Multicast security protocols do not allow each member of a group to verify the identity and authority of every other member of the group. This means that each member of the group must \trust"some other party (group member or not) to verify another group member. This cooperative enforcement of security policy across the group requires a base-level of trust in those verifying authorities.

It is important that every group member, CR Agent, or CR Manager with access to either the group key or the LKH key array (in unencrypted form), is capable of protecting that information from unauthorized disclosure. This requires that all proper rules must be enforced as part of the group security protocol. These computers must also be trusted not to divulge the keys via an unofficial route (e.g., a hacker exploiting a weakness in another application).

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

References

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

5 Addresses of Authors

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

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

Document expiration: August 30, 1999

