            Group Key Management Protocol (GKMP) Architecture

Status of this Memo

Table of Contents

Abstract

   This specification proposes a protocol to create grouped symmetric
   keys and distribute them amongst communicating peers. This protocol
   has the following advantages: 1) virtually invisible to operator, 2)
   no central key distribution site is needed, 3) only group members
   have the key, 4) sender or receiver oriented operation, 5) can make
   use of multicast communications protocols.

1 Introduction

   This document describes an architecture for the management of
   cryptographic keys for multicast communications. We identify the
   roles and responsibilities of communications system elements in
   accomplishing multicast key management, define security and
   functional requirements of each, and provide a detailed introduction
   to the Group Key Management Protocol (GKMP) which provides the
   ability to create and distribute keys within arbitrary-sized groups
   without the intervention of a global/centralized key manager. The
   GKMP combines techniques developed for creation of pairwise keys with
   techniques used to distribute keys from a KDC (i.e., symmetric
   encryption of keys) to distribute symmetric key to a group of hosts.

1.1 Multicast Communications Environments

   The work leading to this report was primarily concerned with military
   command and control and weapons control systems, these systems tend
   to have top--down, commander--commanded, communications flows. The
   choice of what parties will be members of a particular communication
   (a multicast group for example) is at the discretion of the "higher"
   level party(ies). This "sender-initiated" (assuming the higher-level
   party is sending) model maps well to broadcast (as in
   electromagnetic, free-space, transmission) and circuit switched
   communications media (e.g., video teleconferencing, ATM multicast).

   In looking to apply this technology to the Internet, a somewhat
   different model appears to be at work (at least for some portion of
   Internet multicast traffic). IDRP and Distance Vector Multicast
   Routing Protocol (DVMRP) use multicast as a mechanism for parties to
   relay common information to their peers. Each party both sends and
   receives information in the multicast channel. As appropriate, a
   party may choose to leave or join the communication without the
   express permission of any of the other parties (this begs the
   question of meta-authorizations which allow the parties to
   cooperate). More interestingly, the multicast IP model has the
   receiver telling the network to add it to the distribution for a
   particular multicast address, whether it exists yet or not, and the
   transmitter not being consulted as to the addition of the receiver.

   Other applications of multicast communications in the Internet, for
   example NASA Select broadcasts, can be viewed as implementing the
   sender model since the sender selects the broadcast time, channel,
   and content, though not the destinations.

   It is our intention to provide key management services which support
   both communications (and implied access control) models and operate
   in either a circuit switched or packet switched environment.

1.2 Security for Multicast

   Multicast communications, as with unicast, may require any of the
   security services defined in ISO 7498, access control, data
   confidentiality, traffic confidentiality, integrity/data
   authentication, source authentication, sender and receiver non-
   repudiation and service assurance. From the perspective of key
   management processes, only data confidentiality, data authentication,
   and source authentication can be supported. The other services,
   traffic confidentiality, non-repudiation, and service assurance must
   be provided by the communications protocol, they may rely on
   cryptographic services but are not guaranteed by them.

2 Multicast Key Management Architectures

2.1 Current Operations

   There are several electronic mechanisms for generating and
   distributing symmetric keys to several computers (i.e.,
   communications groups). These techniques, generally, rely on a key
   distribution center (KDC) to act as a go between in setting up the
   symmetric key groups. Military systems, such as BLACKER, STU-
   II/BELLFIELD, and EKMS, and commercial systems, such as X9.17 and
   Kerberos, all operate using dedicated KDCs. A group key request is
   sent to the KDC via various means (on- or off-line) The KDC acting as
   an access controller decides whether or not the request is proper
   (i.e., all members of a group are cleared to receive all the data on
   a group). The KDC would then call up each individual member of the
   group and down load the symmetric key. When each member had the key
   the KDC would notify the requester. Then secure group communication
   could begin. While this was certainly faster then anything that
   requires human intervention. It still requires quite a bit of set-up
   time. Also, a third party, whose primary interest isn't the
   communication, needs to get involved.

   Pairwise keys can be created autonomously by the host on a network by
   using any number of key generation protocols (FireFly, Diffe-Hellman,
   RSA). These protocols all rely on cooperative key generation
   algorithms to create a cryptographic key. These algorithms rely on
   random information generated by each host. These algorithms also
   rely on peer review of permissions to ensure that the communication
   partners are who they claim to be and have authorization to receive
   the information being transmitted. This peer review process relies
   on a trusted authority assigning permissions to each host in the
   network that wants the ability to create these keys. The real beauty
   of these pairwise key management protocols is that they can be
   integrated into the communication protocol or the application. This
   means that the key management becomes relatively invisible to the
   people in the system.

2.2 GKMP-Based Operations

   The GKMP described below, delegates the access control, key
   generation, and distribution functions to the communicating entities
   themselves rather than relying on a third party (KDC) for these
   functions. As prelude to actually distributing key, a few things
   must be assumed (for purposes of this document): there exists a
   "security manager" responsible for creating and distributing to
   parties authentic identification and security permission information
   (The security manager function may be accomplished through a strictly
   hierarchical system (a la STU-III) or a more ad hoc system of

cooperating peer "domain managers," the implementation of the
certification hierarchy is not addressed in this document.);
communicating parties are online for the keys formed and distributed
by the GKMP.

2.2.1 Sender Initiated Operations

This section describes the basic operational concept for multicast
key management for sender initiated multicast support. This model of
multicast communications was the basis for our original work on
multicast key management. From a security viewpoint the sending
application is able to control access to the transmission through
both key distribution and communications distribution (not sending
the transmission to some addresses).

Identification of Group Key Controller -- The originator of the
multicast group creates or obtains a group management certificate
from its certification hierarchy. The certificate identifies the
holder as responsible for generation and distribution of the group
key (Naming standards are not addressed here, the name should reflect
the naming structures appropriate for the supported cryptographic
service. For example, IP-level encryptors should use naming
reflecting "host" identities (IP addresses, or DNS host names), RTP
encryptor would use session names). The originator relays the
membership list to the Group Key Management (GKM) application.

Group Key Creation --   The GKM application, operating on behalf of
the originator, selects one member of the group, contacts it, and
creates a Group Key Packet (GKP). A GKP contains the current group
traffic encrypting key (GTEK) and future group key encrypting key
(GKEK). The GKM application then identifies itself as the group key
controller, which the member validates, under cover of the GTEK.

        Group Key Packet (GKP) = [GTEKn,GKEKn+1]

As part of group key packet formation, usage parameters, appropriate
for the underlying crypto-system, are selected. Unlike normal
parameter negotiation, where common security-level/range, and
services are arrived at, the originator's GKM application selects
these parameters and the member must comply.

Group Key Distribution --   After creation of the GKP, the group
controller contacts each member of the group, creates a Session Key
Package (SKP), validates their permissions (check member's
certificate against group parameters), and create a Group Rekey

Package for that member. A SKP contains a session TEK and a session
KEK for a particular member. A GRP contains the GKP encrypted in a
KEK and signed using the originator's certificate.

     Session Key Package (SKP) = [STEK, SKEK]

     Group Rekey Package (GRP) = {[GKP]KEK} SignatureController

Group Rekey --   When the group needs to be rekeyed, the originating
GKM application selects a member, creates a new GKP, creates a new
GRP (which is encrypted in the previously distributed next GKEK) and
broadcasts it to the group.

This procedure is fairly complex, but other than for the distribution
of site-specific certificates, no centralized key management
resources are needed. The only parties to the key management
communications are the same parties which will be participating in
the group.

2.2.2 Receiver Initiated Operations

This section describes key management operational concept for
receiver initiated multicast communication support. The receiver
initiated model presents some interesting problems from a security
view point since the end-participants are not known a priori. Also,
in a purely receiver initiated application (such as DVMRP), there is
no concept of an "originator" and the participants in the group may
be quite dynamic with participants changing on a minute by minute
basis.

For secure group communications to take place, all members must
obtain the same key. This may be achieved by either using
deterministic key generation techniques (using a secret, shared seed)
or by making one member of the group responsible for creation of the
key. The use of a deterministic key generator presents security
problems, particularly regarding loss of the seed (it compromises
both past and future traffic). The assignment of a member to the
role of key "controller" also presents drawbacks, but these relate to
determining which one should be the controller and the need for each
member to contact him. The remainder of this discussion will look at
how the "controller" concept from above could work in the receiver
initiated case.

Selection of Group Key Controller --   A group member will be made
responsible for initial group establishment and periodic generation
and dissemination of new GRPs. There is no need for the selected
controller to be the controller for all time, but at any one time
only one controller may be active for each group. Selection of

controller may be made through a voting system, by a simple default
(the first to transmit to the group is the controller), or
configuration.

The current controller's identity must be made available to all
members, and potential members, for initial group key load and error
recovery. The information may be relayed by broacast on a key
management "channel," or through a directory service.

Group Key Creation --   The GKP is created and distributed in much
the same way as in sender initiated operations. The controller
creates a GKP with the first group member to initiate contact. The
GKM application then identifies itself as the group key controller,
which the member validates, under cover of the GTEK. Parameter
negotiation is performed and the first group member is keyed.

Group Key Distribution --   After creation of the GKP, as other
members contact the controller, a SKP is created, member permissions
are validated and a GRP is loaded to the member.

For widely distributed groups, a form of distributed dissemination
may be used. Some number of regional GKM applications are enabled
with the ability to validate the permissions of new members and upon
validation send to them the current GKP.(Access control is not
defined in this document, but it is assumed that both hierarchical
and discretionaly (rule-based and identity-based) access control will
be supported.) These regional key distributors perform the same
functions as the controller, except that they do not create the GKP.
This concept can be expanded to the point where all current members
are capable of downloading the GKP, and passing on that capability.

Group Rekey --   When the group need rekeying the procedure would be
identical to the sender initiated case. The controlling GKM
application selects a member, creates a new GKP, creates a new GRP
(which is encrypted in the previously distributed next GKEK) and
broadcasts it to the group.

2.3 GKMP Features

This section highlights areas which we believe the GKMP approach has
advantages over the "traditional" KDC based approaches.

2.3.1 Multicast

Multicast protocols are a growing area of interest for the Internet.
The largest benefit of a multicast protocol is the ability of several
receivers to simultaneously get the same transmission. If the
transmission is of a sensitive nature, it should be encrypted. This

means that the all members of the group must share the same
encryption key to take benefit of the multicast transmission.

To date the only way of setting up a group of symmetric keys is with
the assistance of a centralized key management facility. This
facility would act as a key broker creating a distributing key to
qualified group members. There are several problems with this
centralized concept. These problems give rise to many of the
following motivations for creating a distributed key management
protocol.

2.3.2 Increase the autonomy of key groups

The GKMP proposes to extend the pairwise key paradigm to grouped
keys. This protocol can be integrated into the communication
protocols or applications and can become invisible to the host's
operator. We will use peer review to enforce our security policy.

The GKMP allows any host on a network to create and manage a secure
group. Maintenance of these group keys can be performed by the hosts
interested in the group. The groups themselves will be relatively
autonomous. This simplifies the installation of this technology
allowing more host to use secure multicast communications.

2.3.3 Latency

Latency refers to the time to set-up or tear down or to re-key a
group. In short this corresponds to the length of time it would take
to set-up a multicast address.

The GKMP can allow delegation of group creation authority to any host
in the network. In essence, when a host needs a group it will have
the tools needed to create that group and manage it. Additionally,
since the host only needs to create a single group it can concentrate
on that particular group.

In the current centralized key distribution approach. The group must
be requested from the central site. The central site would process
that request in accordance with it's priority and current workload.
Latencies would develop if the workload of the central site gets
unwieldy or if the communications to the site become overloaded.

2.3.4 Extendibility

One of the problems with a centralized key distribution system is the
concentration of key management workload at a single site. The
process of creating key groups -- key creation, access review,
communication to group members takes time and effort. As the number

of groups on the network grows and the number of group members group.
The workload at that central sight quickly reaches capacity.

GKMP should allow a great number of groups to exist on the Internet
without overloading any particular host. Delegation of the net wide
group creation and management workload places the burden of
maintaining groups on the hosts interested in using those groups.
Not only is this more efficient, but it places the burden in an
appropriate location.

The GKMP distributes the communication requirements to manage groups
across the network. Each group manages the group using the same
communication resources needed to pass traffic. It is likely that if
a communication group can support the traffic of a group, it will be
able to support the minimal traffic needed to management the keys for
that group.

GKMP provides it's own access control, based on signed netwide
permission certificates. This partially disseminates the burden of
access control and permission management. A system wide authority
must assign the permission certificates, but day to day access
control decisions are a GKMP responsibility.

2.3.5 Operating expense

A centralized key distribution site contains, at one time or another,
the keys for the net. This is a valuable target for someone to
compromise. To protect this site physical and procedural security
mechanisms are employed (e.g., guards, fences, intrusion alarms, two
person safes, no-alone zones). These mechanisms do not come cheap.

Allowing the hosts to create and manage their keys eliminates the
need for an on-line centralized key distribution site. The protocol
approach restricts access to the keys to the hosts using them (the
minimal set). Since, the encryption mechanisms will have already
incurred the cost to be physically secured there is no additional
cost levied on the system by the key management system.

2.3.6 Communication Resources

Because a centralized site is involved in creating, distributing,
rekeying, and providing access control for every group, it is
frequently accessed. The communication resources available to this
site often become a bottle neck for the groups. Therefore a big pipe
is usually installed to this facility.

The GKMP proposes delegating most of the key creation, distribution,
rekey and access control mission to the hosts that need the secure
communication. There no longer is a single third party that must be
consulted prior to every group key management action. Hence, the
communications requirements to manage the keys have shifted to the
groups themselves. The need for special high capacity communications
has been eliminated.

2.3.7 Reliability

Delegating key management responsibility to the groups eliminates the
centralized key management site as a single point of failure. The
groups that will use the key are responsible for it. If the
communications system fails for the key management it is also down
for the communications.

The GKMP will attempt to delegate as many functions to the group as
possible. There will be some functions which still need to be
performed outside of the group (granting of privileges). These
functions can still fail. The GKMP will operate on the old set of
permissions. These functions need not be in-line. They are
performed separate from the key management actions and are not
crucial to day-to-day operation.

2.3.8 Security

People are the most risky element for security. A distributed
protocol eliminates many people from the key distribution chain.
This limits "exposure" of the key.

3 GKMP Protocol Overview

3.1 Supporting functions

A secure key management protocol needs a number of supporting
functions, especially in a military environment. The two major
support functions are security management and network group
management. In the commercial world a company could provide these
support functions.

The issue of Security Management is permission management, in a
military environment separation of data occurs along classical
classification lines (i.e., TOP SECRET to UNCLASSIFIED). In the
commercial world these levels are proprietary or need to know access.

Network group management provides an interface to the communications
system and control of network resources. Some entity either a
commercial or military system, the host or network operations center,

   must provide the key management protocol with a list of the group
   members. Also, if the network resources, bandwidth and processing,
   are considered scarce a management structure must allocate them.

3.1.1 Security management

   Security management is a role performed for the entire network. It
   involves netwide issues of permission management, initialization of
   software, and compromise recovery. The GKMP relies on security
   management to operate. Refer to figure 1: Security management view.

   The GKMP must assume trusted handling of the protocol software prior
   and during installation. If the GKMP is to use peer to peer access
   control the system must control the assignment of permissions. These
   permissions must be monitored and updated as needed. Finally,
   overview of these permissions must include the maintenance of a
   Certificate Revocation List.

   Secure start-up We need to control the process of loading GKMP
   software onto a host and initializing it. The protocol needs keys,

   Security Manager --> --> --> --> --> --> --> --> --> --> --> Network
                                 Permissions
                                 Secure Start-ups
                                 Compromise recovery
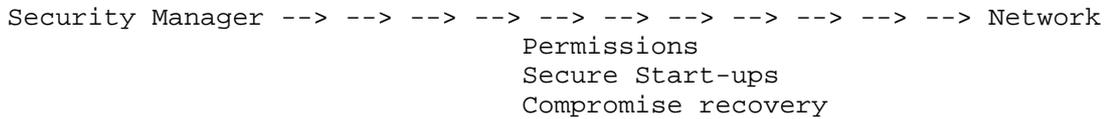
                     Figure 1: Security Management View

   public and private, to operate. It also must have identify
   information of the host on whose behalf it will act.

   There are some life cycle and security concerns with the software
   while in transit, stored, distributed, and installed. A one time
   start-up procedure must verify the identity of the host. Procedural
   and physical identification techniques will verify the identity of
   the host (i.e., the Armed Forces Courier Service (ARFCS) accounting,
   or registered mail). Upon key delivery the security manager logs
   it's receipt and assumes responsibility for the key.

   After proper installation of the software a paper trail verifies the
   recipient. The computer would initiate an association with the
   security management function to initialize the protocol software
   (create a unique public and private key pair for network operation
   and receive network permissions). This activation process uses keys
   distributed with the software (good only for initialization) to
   secure an exchange with the security manager. The host then creates
   a unique public and private pair and sends the public key to the

security manager. The security manager creates a credential that
uniquely identifies the host and it permissions. This credential is
signed by the security management with its private key and can be
verified by all net members with the public key.

Permission management Each host on the network is given a
permissions certificate signed by the security management which
uniquely identify that host and identifies the access permissions it
is allowed. These permission certificates are used by the network
hosts to assign permissions to other hosts.

This process assigns permissions to equipment or human beings in
accordance with their duties. This process involves security
clearances and human judgment therefore it is outside the scope of
this protocol.

The security management function, especially in military operations,
would be responsible for managing permissions and classifications at
each host. In the commercial world, permission management
corresponds to projects or duties.

Compromise recovery management If a group member is found
compromised, the protocol must facilitate the exclusion of the
compromised member and return to secure operations. The security
management function will provide control of compromise recovery.

Usually, physical inspections or accounting techniques find
compromises. These separate systems report the compromise to the key
management system. We must assume the loss of all key resident at
that host. The security management function will rescind the
permission allocated to this compromised host. We create a list of
all know compromised hosts and distribution that list across the
network. Each host is then responsible for reviewing the propriety
of each association and enforcing access control to data.

3.1.2 Group management

The group manager interacts with other management functions in the
network to provide the GKMP with group membership lists and group
relevant commands. The GKMP deals strictly with cryptographic key.
It relies on external communication and network management services
to supply network related information. Primarily, it relies on the
network management service to provide it with the addresses of group
members (if the group is sender initiated).

The GKMP allows an external entity to determine the controller of a
group. The controller of the group should be able to handle the
additional processing and communication requirements associated with
the role. If this is not a necessary function given the
implementation, this assignment of controller duties can be set to
some automated default. However, even if defaulted some external
management entity determines how the role of controller is allocated.

The group manager can receive group progress reports from the group
controller. The GKMP provides a service for the network. It makes
sense that someone in the network is interested in the progress of
this service. The GKMP can provide progress reports. It is up to
the network management to determine the manner and recipient of the
reports. Reference figure 2: Network manager interaction.


```
Group Manager --> --> --> --> --> --> --> --> -->Network Manager
        /\
         |
         |          Commands, Role assignments
         |          Group member list, Reports
         |
        \/
{[Group Controller]     Network}
```
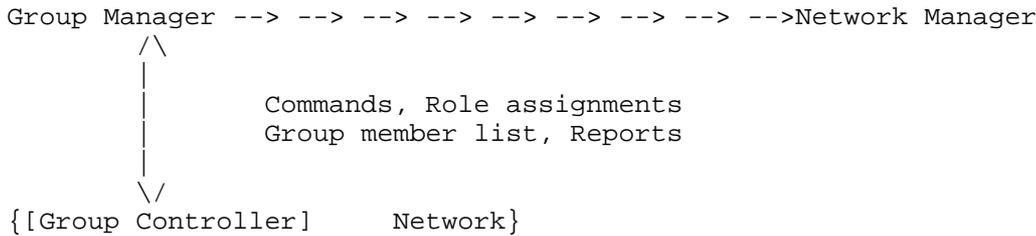
                    Figure 2: Network Manager Interaction

Group to member mapping When the GKMP is implemented in sender
initiated group establishment mode, a list of group member addresses
must be provided as part of the group establishment command. The
GKMP will use these addresses to contact the group members and create
the group.

The creation of groups involves the assignment of a group address,
update of router databases, and distribution of this group address to
the group members. This is a classic function of network management.
The GKMP group controller would be another recipient of this
information.

Protocol role allocation The Group Management Protocol assigns roles
to members of a particular group. These roles are binary one is
either the control over the group or a member of a group. Some
external entity will allocate the identity of the group controller
and group receiver. This is a desirable aspect because some
computers are more capable (i.e., central site, great deal of process
power available to control a group). We allow some external entity
to allocate these roles to individual group members, this is
important in the military application do to the fact that in a

commercial application the allocating authority and group controller
may very well always be the same.

Group key progress reporting The Group Key Management Protocol has
to be able to report to somebody. If we create a group, we should
report it to group requester. Contrarily if we are not able to

Network = {[(Group 1 controller) Group 1 members],
[(Group 2 controller) Group 2 members],
[(Group 3 controller) Group 3 members], }

                 Figure 3: Distributed Group Management

create a group we should report that especially since failure to
create a group at least as a first study will highly correlate with a
failure of the underlying communications. The Group Key Management
Protocol does not have an ability to fix the underlying
communications so the communication management function must deal
with these failures.

3.2 Protocol Roles

Creation and distribution of grouped key require assignment of roles.
These identify what functions the individual hosts perform in the
protocol. The two primary roles are those of controller and
receiver. The controller initiates the creation of the key, forms
the key distribution messages, and collects acknowledgment of key
receipt from the receivers. The receivers wait for a distribution
message, decrypt, validate, and acknowledge the receipt of new key.

One of the essential concepts behind the GKMP is delegation of group
control. Since each host in the network has the capability to act as
a group controller, the processing and communication requirements of
controlling the groups in the network can be distributed equitably
throughout the network. This avoids potential single points of
failure, communication congestion, and processor overloading. Refer
to figure 3: Distributed group management.

3.2.1 Group controller

The group controller is the a group member with authority to perform
critical protocol actions (i.e., create key, distribute key, create
group rekey messages, and report on the progress of these actions).
All group members have the capability to be a group controller and
could assume this duty upon assignment.

The group controller helps the cryptographic group reach and maintain
key synchronization. A group must operate on the same symmetric
cryptographic key. If part of the group loses or inappropriately
changes it's key, it will not be able to send or receive data to
another host operating on the correct key. Therefor, it is important
that those operations that create or change key are unambiguous and
controlled (i.e., it would not be appropriate for multiple hosts to
try to rekey a net simultaneously).

3.2.2 Group receiver

   Simply stated a group receiver is any group member who is not acting
   as the controller. The group receivers will: assist the controller
   in creating key, validate the controller authorization to perform
   actions, accept key from the controller, request key from the
   controller, maintain local CRL lists, perform peer review of key
   management actions, and manage local key.

3.3 Scenarios

3.3.1 Group establishment

   The protocol to establish a group of host that share a cryptographic
   key must create a high quality key, verify that all intended
   recipients have permission to join the group, distribute the key to
   all qualified members, and report on the progress. This process
   consists of two phases: creation of the key and distribution of the
   key. Refer to figure 4: Group Establishment.

   The group establishment process is proceeds in the following manner.
   First, a "create group" command is issued to the group commander.
   The group controller validates the command to ensure it came from an
   authorized commander and the group is within the controller's
   permission range. Next, the controller creates a key. Then that key
   is passed to the group members, after they pass the peer to peer
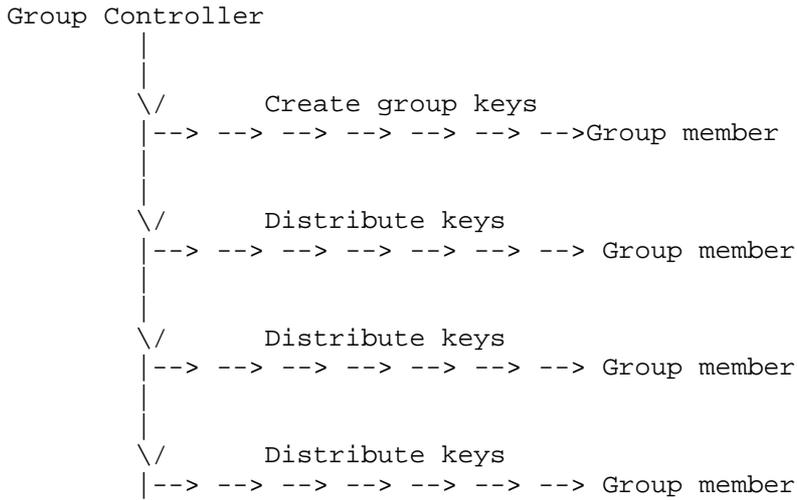   review process.

```
Group Controller
         |
         |
        \/        Create group keys
         |--> --> --> --> --> --> -->Group member
         |
         |
         |
        \/        Distribute keys
         |--> --> --> --> --> --> --> Group member
         |
         |
         |
        \/        Distribute keys
         |--> --> --> --> --> --> --> Group member
         |
         |
         |
        \/        Distribute keys
         |--> --> --> --> --> --> --> Group member
```

                   Figure 4: Group Establishment

Validate command The create group command is signed by the group
commander (they may be the same device). This signature should be
asymmetric in nature. The public key to validate this command can be
sent with the command itself, if the public bound to the identity of
the commander.

The group controller receives the command. It verifies that the
signature, thereby ensuring the message was sent by the claimed
source and the message has not been modified in transit.

Creation of group keys The controller initiates the creation of two
keys for use in the group. The creation of a cryptographic key
requires that the key be sufficiently random. Randomizers, capable
of creating high grade cryptographic key, tend to be hardware based
and are not likely to be practical for this protocol. There are
several established key creation protocols based in software (e.g.,
Diffe-Hellman, FireFly, RSA). All these software based algorithms
involve two hosts cooperating to create a cryptographic key. These
software algorithms are more appropriate for this protocol.

Also important, in the creation of these keys, is verification of the
authorization of the key creation partner. Authorization to posses
the keys include permissions that equal or exceed the group traffic
and identity verification.

Distribution of group keys The controller distributes the group keys
to the net members. The controller must verify the identity and
permissions of each member prior to the key being distributed.

                         Rekey Group
Group Controller --> --> --> --> --> -->{Group (group member 1-n)}

                      Figure 5: Group Rekey

Likewise, the net member must verify the controller's identity,
authorization to perform this action, and permissions.

The key being distributed is the same level as the data that it will
encrypt. Hence, we must encrypt the key during distribution. If no
suitable key exists between the controller and member, a new key must
be created. This new key is cooperatively created between the
controller and net member in a similar manner as the net keys.

The controller creates a message for encryption in the key held
between the controller and member. This message will include key
management information and the keys.

3.3.2 Group rekey

Cryptographic key has a life span. New key must replace "old" key
prior to the end of its cryptographic life. This process is rekey.

Rekey has the advantage of using an existing cryptographic
association to distribute key. Also, there is no requirement to
verify the identity and authorization for the other members.
Identify and authorization are assumed.

A group rekey consists of two stages. First the Group Controller
creates new group keys. Second these "new" keys are sent to the
Group Members in a multicast message. Refer to figure 5: Group
Rekey.

Creation of group keys The controller of the rekey will create the
new keys in exactly the same manner as used during group
establishment.

Distribution of group keys The GKMP creates a message for the group
address. This message uses one of the keys distributed during group
establishment to encrypt the new keys. It also contains an
authorization token identifying the controller as the rekey agent and
new management data. All members of the group using a multicast
protocol (if one exists) accept this message.

The message which rekeys the group encrypts the new keys in the
existing KEK. Since all group members possess the KEK the entire
group can decrypt this message.

The token authorizing the group controller to perform this rekey is
also included. This token is asymmetrically signed by the group
commander. It uniquely identifies the group controller's authority
to rekey this group. It also identifies the group the level of
traffic and rekey interval.

3.3.3 Deletion

It is desirable to be able to delete group members for either
administrative purposes or security reasons. Administrative deletion
is the deletion of a trusted group member. It is possible to confirm
the deletion of trusted group members. Security relevant deletion is
the deletion of an untrusted member. It assumes that the member is
ignore all deletion commands.

Administrative delete Administrative deletion removes the group keys
from trusted group members. This deletion consists of two messages
the first sends a command to the group encrypted in the groups TEK.
The command essentially says: acknowledge receipt and then delete
group keys. This command is signed by the group controller to
prevent unauthorized deletions.

The acknowledgment message is also encrypted under the group TEK and
is sent to acknowledge receipt of the command. We could acknowledge
accomplishment of the command if the net is willing to accept the
burden of creating pairwise keys between the exiting group members
and the group controller.

Compromise recovery Compromise recovery is the deletion of untrusted
group members. This actually involves the creation of an entirely
new group, without the untrusted member. Once the new group is
created, net operations can be shifted to the new group, effectively
denying the untrusted member access to the data.

There is always a trade-off between security and continued net
operations when a member is found to be compromised. The security
first position states that if a member is compromised, the group must
be destroyed and then a new secure group created. However,
operational concerns sometimes out weigh the security concerns. The
operational position is that the group will continue to operate with
the compromised member and will shift to a new secure group when it
becomes available.

The GKMP does not mandate either position. However, the speed and
flexibility of the GKMP does allow a new secure group to be created
quickly. Thereby, restricting the potential damage done by a
compromised member.

Once a member is found to be compromised, that members certificate is
added to a Certificate Revocation List (CRL). The CRL is an
asymmetrically signed piece of data, signed by a security manager.
The list is made up of compromised resource ID's, a version of the
CRL, and perhaps an identifier of the security manager. The CRL is
accessed every time a new key is negotiated. If one of the key
creators is on the CRL the key is destroyed and interaction
terminated.

The idea behind a CRL is each host would keep records of all open
associations and compromised resources. The host would then make
sure that it does not have and will not create a secure association
open with anyone who is on the CRL. The CRL concept of becomes more
complicated in the case of groups. This is because it is not
necessary for every member in the group to know who the other group
members are. Hence, a group member does not have sufficient
information to identify compromised group associations. The GKMP
proposes that the group controllers be responsible for reviewing the
CRL and taking appropriate actions should a group member be
compromised.

Another issue with CRLs is the speed that they can be distributed
across a network. Every time a key is created the cooperating hosts
exchange the version number of their current CRL. If the versions do
not match. The most current version is passed to the host with the
old version. Hence, CRLs propagate when keys are created. If this
is infrequently and there is a single CRL insertion point, the list
may take a few days to move across the net. The GKMP allows a
speedier distribution of the CRL.

The GKMP delegates control of groups to specific group controllers (a
subset of the network). These controllers are responsible for
maintaining the security of the group. If quicker distribution of
the CRL were desired, the CRL generator (security management

function could seed the CRL at these controllers. Controllers are
points of key management activity and are logical CRL staging areas.

4 Issues

What are the unresolved issues with this protocol?

4.1 Access Control

One interesting issue with a grouped key protocol is access control.
This is because we are moving away from having humans in the loop or
having a central authority to check the propriety of the group.

The group protocol must police itself. It must ensure that each
member of a group meets the classic military access control policy (
i.e., a group member's classification level must be higher or equal
to the classification of the group that it's in).

Is allocation of permissions by a higher authority sufficient to
provide access control? Or is a more discretionary mechanism
necessary?

4.2 MLS

A GKMP must be capable of operating in a multi-level secure
environment. The integration of a key management protocol capable of
creating keys of several different classifications with an operating
system capable of operating with multiple classifications in non-
trivial.

Classified label standards needed to be incorporated. The
classification labels used by the key management protocol should
coincide with the labels used by the MLS operating system. These
interoperability issues need to be addressed.

4.3 Error Conditions

A group protocol is more complex than a pairwise protocol hence there
are more possible error conditions. In a pairwise protocol you have
two parties; they must communicate between themselves. It is
relatively simple to define an take care of all the potential error
conditions.

One assumption with any group protocol is the underlying internet is,
to some degree, always broken. The protocol designer has to assume
that messages will be delayed or destroyed in transit, all member
will not receive all multicast messages, and acknowledgment of
actions may not be delivered. This assumption is important if a
protocol uses multicast functions to speed-up actions.

The protocol must provide recovery mechanisms to allow group members
to recover from loss of messages. It must recover in a way that is
transparent to the host and underlying communications network.

For example, there is an issue whether or not we can create an
application layer acknowledgment of multi-cast actions. The issue
deals with the required bandwidth that acknowledgment would take up.
It may be much more friendly to the underlying communications systems
to have each member identify potential errors and correct them in a
pairwise manner. The task of handling error conditions in a key
management protocol is double difficult because many error conditions
can be induced error condition (invoked by a third party trying to
break the security of that system) to retrieve there key that is in
transit or to block the successful dissemination of a key thereby
attacking the system security mechanism.

4.4 Commercial vs. Military

Commercial and military key management differ in many ways.
Commercial Key management protocols tend to emphasize inter-
operability, freedom of action, and performance. Military systems
tend to emphasize security and control of operations.

There will be a difference in cryptographic algorithms. The military
protocol would certainly use high grade encryption because of
protecting classified information. The commercial system would
probably using algorithms. and techniques certified for unclassified
communication systems. The main difference is in the algorithms
length and type.

A military protocol would require more management and structure than
a commercial one. The military has always adopted a hierarchical
communication structure and the commercial system, especially if you
look at the internet, work mainly by anarchist style.

4.4.1 Algorithm Type

Another difference between military and commercial key management is
the type of cryptographic algorithms. The commercial world uses
encryption algorithms like DES and in the future Skipjack. The
military uses other cryptographic algorithms that differ in key

   length and have more restrictions. An example of this would be the
   identification of ACCORDION, as a military key encryption algorithm
   as used in the EKMS program run by NSA.

   Any experiments with a grouped key management protocol must consider
   the differences between military and commercial algorithms. The
   commercial algorithms tend to be quicker to implement, run faster,
   involve less processing time, and allows an unclassified experiment.
   However, we must be careful not paint an unrealistic picture of the
   performance of the protocol based on these commercial algorithms. A
   military algorithm tends to be more cumbersome to process, slow to
   process, require more bandwidth, a lot of unpleasant characteristics
   from the commercial stand point, but allow for a higher grade of
   cryptographic security. One way of dealing with the disparity
   between algorithms is to use the commercial cryptographic algorithms
   and leave the fields the size used by a comparative DOD cryptographic
   algorithms and insert delays to simulate DOD algorithm processing
   times.

4.4.2 Management Philosophy

   Management for a military network is far more structured than a
   commercial network. A military network would restrict the creation
   of network groups, the rekeying of those groups, and access to the
   data contained in those groups. In contrast the commercial world
   would enable any member in the network to create a group and allow
   any other member of the net to join that group.

   The group Key Management Protocol must allow for both these
   architectures i.e., all for very structure command control hierarchy
   and another free form group creation.

4.5 Receiver Initiated Operations

   How do they actually work, what are the performance trades,
   experimentation needed.

   Who is the group leader?

   How do we elect a new leader?

   Will multiple leaders be created?

   Will rule based access control allow fine enough access disgression?

   Methods for distributed GKP/GRP dissemination need to be examined.
   This includes:

   o resolving group identification issues, such as how to notify
      potential members of membership requirements without compromising
      any security-relevant information about the group;

   o approaches for rapidly identifying GKP/GRP sources must be
      developed, such as a "Key ARP" whereby a new member broadcasts
      into the group a request for key service and existing members
      resolve which will provide service; and,

   o Security effects of distributing access control decisions must
      also be reviewed.

5 Security Considerations

   This document, in entirety, concerns security.

6 Addresses of Authors

   Hugh Harney
   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

   Carl Muckenhirn
   SPARTA, Inc.
   Secure Systems Engineering Division
   9861 Broken Land Parkway, Suite 300
   Columbia, MD 21046-1170
   United States
   telephone: +1 410 381 9400 (ext. 208)
   electronic mail: cfm@columbia.sparta.com