Internet Engineering Task Force
INTERNET-DRAFT
draft-ietf-rmt-pi-track-security-01.txt
April 5, 2001
Expires Sep 4, 2001

T. Hardjono (Nortel)
B. Whetten (Talarian)

Security Requirements For TRACK

<draft-ietf-rmt-pi-track-security-01.txt>

Status of this Memo

Abstract

This document discusses the security issues within the TRee-based ACKnowledgement (TRACK) reliable multicast protocol, and identifies some constraints and requirements for security provisions for this protocol. Based on the constraints and requirements, the document proposes a separation of data packet confidentiality and authentication, from transport layer protection. It proposes that TRACK be primarily concerned with group authentication of control and data packets, to protect against attacks on the transport infrastructure. It proposes that data confidentiality and source authentication be provided separately from this low level group authentication, ideally at the application level. We show that this is particularly important for TRACK, because of the requirement that the interior control nodes only OPTIONALLY have access to the data packet payload.

Specifically, the current work RECOMMENDS that data and control packet authentication be provided using IPsec-based authentication at the network layer. This approach allows an interior control node to authentically retransmit a lost data packet (which remains encrypted under the separate data-encryption key) to its own children (a set of Receivers), while making use of the IPsec features, such as protection against replay attacks.

This document then provides a specific proposal for how group keys SHOULD be divided up among group members, for control and data packet authentication. While providing some rationale for divorcing this proposal from that of source authentication and data confidentiality, it does not provide a specific proposal for those pieces.

1. Background: The Multicast Security Problem

The problem of multicast security can be divided into three general areas of concern:

- Data Encryption and Source Authentication. The method used to encipher or scramble the multicast data, and verify the identity of the sender of this data.
- Key Distribution. The method used to securely distribute group keys and keying material to the members of a group, for use in decrypting or authenticating the data or control packets.
- Infrastructure Protection. Mechanisms used to protect the multicast infrastructure itself, and to minimize the ability of an intruder to deny service to legitimate users.

The security of reliable multicast protocols falls primarily into the third category of problems. Complete denial of service protection must start at the network level (i.e. IP Multicast), with controls placed on senders from overloading the network with brute force "spamming", as well as with authentication of control packets, to keep users from corrupting the state of the IP Multicast protocols. A transport protocol needs to address the same issues, checking to make sure that senders are not sending more data than they are allowed (such as with enforceable congestion control), and authenticating control packets, to protect the protocol state. Control packet authentication is particularly important in TRACK, because of its use of interior control nodes (Repair Heads, or RHs) to increase scalability.

An OPTIONAL extension to the requirement of infrastructure protection is that of infrastructure privacy. Some applications require that the headers of the network packets be encrypted, to provide protection from network analysis attacks.

2. Conventions Used in this Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [B97].

3. Independence of RM Security

The security of reliable multicast (RM) protocols is part of the larger problem of the security of the multicast infrastructure, which also consists of the security of the multicast routing protocols.

Since RM protocols and multicast routing protocols exist at different layers in the protocol stack, and since different RM protocols may be employed with different multicast routing protocols, it is useful from a security perspective to treat these two security problems separately. In addition, although in many instances the topology of RM infrastructure may coincide with that of the multicast routing protocol, such symmetry cannot be assumed for all cases.

Similarly, from a design perspective, the problem of securing the data stream (e.g. through content encryption) should be separated from the issue of securing reliable multicast protocols.

Although we treat RM-security as an independent problem from other multicast security problems, this does not preclude using the solutions in other areas in order to solve the security needs of RM. For example, the use of IPsec technology at the IP layer to authenticate multicast routing protocol control-packets can also be used to authenticate RM control-messages. However, the instance of deploying IPsec in both cases MUST be distinguished and treated separately.

4. TRACK Overview

TRACK arranges Receivers (R) into local regions, where each region is assigned to a Repair Head (RH). These groups are arranged hierarchically as a tree rooted at a Sender (SD), with the RHs representing the nodes of the tree, and the Receivers as the leaves. The Receivers send periodic control messages (called ACKs or NACKs) to their parent RH, selectively acknowledging the packets they have received, and requesting the ones they have missed. Retransmission is then performed by the parent RH. Each RH sends their control messages to the RH at the next level up the hierarchy. This process is repeated until the messages reach the sender, informing it of the status of the group, and notifying it when it is allowed to advance its transmission window. The RHs aggregate the selective positive acknowledgements from the receivers, and suppress the redundant negative acknowledgements, in order to solve the ACK/NACK implosion problem.

A RH maintains a local multicast group to just its children, and subscribes to the local multicast group of its parent. A RH uses this local multicast group for retransmissions to its children, which also provides suppression of other negative retransmission requests for that packet at other children.

TRACK distinguishes between a data channel and a control channel. A data channel is a global multicast group created using the underlying multicast routing protocol. A control channel is the interconnected topology of control nodes, for handling error recovery and positive packet acknowledgements.

In order to obtain data packets from the Sender, a Receiver in a given TRACK region MUST join the multicast group (i.e. the data channel). The RH of that region MUST join every multicast group that its descendants have joined. Note that the RHs are not responsible for forwarding the data packets multicast by the Sender, since that data stream is propagated by the underlying multicast routing protocol.

The figure below illustrates a TRACK tree with multiple control nodes.

```
                       -------> SD (Sender node)----->|
                      ^^^                             |
        TRACKs      /  |  \    Control                |
         and      /    |    \    Tree                 |
        NACKs   /      |      \                       |
              /        |        \     (Repair         |
            /          |          \    Head           |
          /            |            \  nodes)         v
        RH             RH            RH  <------------|
        ^^            ^^^            ^^               | Data
       / |           / | \           | \              | Channel
      /  |          /  |  \          |  \             |
     /   |         /   |   \         |   \            v
    R    R        R    R    R        R    R  <---------
               (Receiver Nodes)
```

5. TRACK Protocol Security Issues and Requirements

This section details the security requirements for TRACK. These requirements include general multicast transport requirements, as well as some requirements specific to TRACK.

5.1 Background

In addressing the security issues specific to TRACK, it is useful to consider the general aspects of security relating to reliable multicast.

- Layer in which security is applied:
  The two layers in which security mechanisms are deployed are typically the network layer and the application layer. In the network layer the protocol that is the most commonly used is the IPsec protocol, which provides authentication and/or encryption. In either case, with IPsec the transport headers (and IP headers) are protected. When authentication and/or encryption is applied at the application layer, neither the transport headers nor the IP headers are protected.

- Types of authentication:
  - Source-authentication: If public key (asymmetric) cryptography is deployed, where only the sender knows the secret-half of the public key pair, then unique "source-authentication" can be established.
  - Group-authentication: If shared key (symmetric) cryptography is deployed and the key is shared by more than two parties, then only "group-authentication" can be established. This means that a receiver in a group is only certain that the entity that sent the message is in possession of the symmetric-key, and is thus assumed to be a member of the group sharing the key.

In the following, the TRACK specific requirements are further elaborated.

5.2 Authentication of Control Messages

As stated above, the directly relevant security concern for TRACK is protection of the multicast infrastructure, particularly of the control tree, in order to provide protection against replay or other attacks which seek to corrupt the state of the transport protocol. The authentication of control messages exchanged among TRACK protocol entities represents the minimal security mechanisms necessary to do so.

Two types of authentication mechanisms can be adopted, corresponding to the two basic types of cryptosystems. In the context of reliable multicast, throughput and latency is typically of high importance, and group-authentication based on symmetric cryptography appears to be preferable.

Given this, the efficiencies of symmetric key based authentication appear to outweigh the benefits of public key based authentication. There are potentially cryptographic schemes that can provide the unique source-authentication of public key cryptography while providing the performance characteristics of symmetric key based authentication (e.g. efficient digital signing of the hash-value of several data packets). However, at the present time, for the general case of infrastructure protection, the complexities of these options appear to outweigh the benefits.

Thus, in summary, for TRACK protocol control-messages, explicit group-authentication at the IP layer SHOULD be deployed using symmetric cryptography. Although a number of technologies are available, we propose specifically IPsec-based authentication using a keyed-hash function [KA98b,MG98a,MG98b] due to its growing use and availability.

We denote the symmetric key used for control message authentication as the InteriorNodeKey. The InteriorNodeKey is a symmetric key shared by all RHs and the Sender within a given TRACK hierarchy. The key is independent of any data stream, and is used to authenticate control packets exchanged among the RHs/Senders.

5.3 Non-Decipherability of Data Packets by RHs

TRACK requires that a Repair Head Node (RH) join all of the multicast groups that its descendants have joined. For TRACK it is preferable that authentication methods based on a symmetric key be deployed due to performance reasons. This may be achieved explicitly, such as by using a keyed hash function, or implicitly using encryption (where a successful decryption implies the ciphertext is both unmodified in transit and was generated by a holder of the symmetric key).

However, TRACK has a further requirement, namely that the RHs be OPTIONALLY prevented from reading the multicast data. More specifically, we perceive that RHs may be administered as part of a reliability service offered by third parties such as ISPs. These third parties may refuse the ability to decipher data packets in order to avoid the legal ramifications of having access to the data contents. Thus, from the ISP perspective, TRACK SHOULD allow them to prove to the content-owner that they do not posses the means to alter the contents transmitted through the multicast groups.

Given the above requirement of the RHs and the need for fast authentication, we propose:

- Data stream confidentiality, using either a symmetric or asymmetric key, SHOULD be separated from data authentication, using a symmetric key (i.e. explicit group-authentication).

- Data stream confidentiality SHOULD be conducted at the application layer, while data authentication, using a symmetric key, SHOULD be conducted at the network layer.

- Since the RHs MAY be prevented from reading the multicast data, two (2) different keys SHOULD be deployed corresponding to the needs of data stream confidentiality and data group-authentication.

5.4 Authentication of Data Retransmissions

In TRACK, retransmissions of data packets always come from a child's parent, which may be either the original source or a RH node. This local recovery is an important tool for increasing the scalability and latency of a protocol. In the context of security, it raises the question as to what authentication methods should be used on these packets.

(a) If source authentication (using public key cryptography) and data confidentiality (including implicit group authentication) is applied at the application layer, the RH can simply replay the data (i.e. payload) unmodified to the querying receivers via local multicast.

(b) If, however, explicit group authentication (using a symmetric key) was applied at the network layer (e.g. using IPsec), then the RH could not simply replay the packet due to restrictions at the IP layer. Thus, in this case the RH would have to re-apply the group-authentication.

Since the retransmission is via multicast to a subgroup, then the RH can either use the existing group-shared symmetric key or use a separate symmetric key only for the subgroup of its children. We propose the later approach be OPTIONALLY supported, which means

that a RH and its children (Receivers and in some cases other Repair Heads) could have to share a separate symmetric key for explicit group authentication at the IP layer.

5.5 Keys for Data Confidentiality and for Authentication

As mentioned above, we propose the separation of data stream confidentiality using a symmetric key encryption (effecting an implicit group-authentication) from data authentication using a symmetric key and a keyed hash function (i.e. explicit group-authentication).

We now denote the symmetric key for data stream confidentiality at the application layer as the GroupDataKey. Only the source and valid receivers will have a copy of the GroupDataKey, which is delivered to them through the appropriate Group Key Management (GKM) protocol that identifies and verifies the members individually. In the case where the RHs are not permitted to read the multicast data, they MUST be prevented by the GKM protocol from obtaining the GroupDataKey.

We denote the symmetric key for explicit group-authentication at the network layer as the GroupAuthKey. The GroupAuthKey is distinct from the GroupDataKey. For TRACK, we propose, where feasible, the use of IPsec with keyed hashing at the network layer to provide explicit group-authentication using the GroupAuthKey. Unlike the GroupDataKey, the GroupAuthKey is known by all entities involved in the multicast. This includes all interior node entities (RHs), the Sender and the Receivers.

5.6 Authenticity of NACK and other Control Packets

In TRACK, a RH responds to a NACK from one its children (typically a Receiver) by re-transmitting the lost packet via local multicast. This basic behavior can be open to abuse by an attacker who injects spurious NACK messages towards the RH, causing a local multicast to all children of the RH. In itself this is a waste of bandwidth and may result in a denial of resource to the group members. Other control packets such as group membership requests, could directly impact the state of the group, and could also be used in denial of service attacks.

To counter these types of attack, the control messages themselves SHOULD be authenticated by the RH. Digital signatures using public key cryptography could be applied to the control messages. However, this approach would be inefficient due to the high CPU cost of public key encryption. Also, it would require creating a separate security association with each child of the RH.

Instead, we propose that NACK and other control messages from a child (Receiver) to its RH be protected using symmetric IPsec based authentication, where feasible. This requires the two parties to first establish a Security Association (SA) and a shared symmetric key. The symmetric key is uniform over a subgroup of receivers (i.e. those under the RH).

5.7 Fault Recovery

If a child's connection to a RH or Sender fails, TRACK provides automatic mechanisms for failing-over to another RH or to the Sender. This reconnection needs to happen quickly, so that the child can rejoin the data stream before too much data has been missed to recover from. If a child needs to get a new key for that RH or Sender, this can be a bottleneck. Given that the key distribution infrastructure may be centralized, and a majority of receivers may need to fail over at the same time, this presents a major opportunity for network congestion.

TRACK entities are expected to usually have the addresses of one or more backup nodes. When implementing security features, each child SHOULD keep the key for its primary backup parent. Optionally, it MAY need to keep the keys for each of the backup parents it is using.

5.8 Implementation with Different Levels of OS Protection

A TRACK protocol can be implemented in (at least) one of three ways.

- Application level. Most implementations of TRACK for the near future are expected to operate in the application level of the OS, running on top of UDP.
- Kernel. As TRACK becomes bundled with standard operating systems, it is expected to become a kernel module, and run directly over IP.
- Virtual machine. Some TRACK entities (particularly senders and receivers) will be run in virtual machines, such as when implemented as a Java applet.

TRACK protocol security SHOULD accommodate all three of these options. This raises the following issues.

- Application Level. One advantage of application level implementations is their flexibility. These implementations could use either IPsec routines, the application layer security functions, or both.

- Virtual Machines. There are issues trying to use IPsec with virtual machines such as Java, which have to date hindered the support of IPsec through native Java applets. TRACK SHOULD be able to OPTIONALLY use only application level security.
- Kernel Implementations. As a TRACK protocol becomes bundled with operating systems, it is expected that IPsec will also become bundled with the OS. To avoid having to use less trusted software in the application level, the TRACK protocol SHOULD be able to use a kernel level security system (such as IPsec) for its transport level security needs.

5.9 Separate Regional Protection

TRACK is expected to be used for content distribution from a few senders to many receivers. In the case of applications that distribute critical data to many different organizations, it is not enough to trust all receivers. For example, a market data feed provider could be using a TRACK protocol to distribute live market data feeds to competing financial institutions. In this situation, the data feed provider needs to be able to protect individual companies from corrupted control packets from other customers (which could be generated either intentionally, or more likely, unintentionally) which would cause denial of service.

As we have proposed, a TRACK protocol MAY provide separate regional keys for the group-authentication of control packets sent from the receivers of different RHs. This allows the authentication of control-messages for each set of receivers (part of one customer) to be done separately (from another set of receivers, as part of a different customer). Since for each set of receivers a different key is used, this limits the ability of a customer to perpetrate denial of service attacks against other customers.

5.10 Piracy of Pay-Per-Use Data

A common scenario for TRACK involves pay-per-use data distribution, such as live market data, pay-per-view video signals, or paid subscriptions to software updates. In this scenario, a receiver cannot be trusted to not give its group keys to outside entities that are trying to get free service. We mention this requirement since it is different than point-to-point security. However, this requirement is the responsibility of the application level security.

6. Architecture Recommendations

The previous section detailed some of the specific requirements and issues for TRACK protocol security, along with some individual recommendations on handling each one. Given those requirements, we propose the following architecture recommendations for implementing security with TRACK.

6.1 Separation of Security Responsibilities

As detailed above, TRACK is primarily concerned with protection of the network infrastructure, rather than with issues such as data confidentiality and source-authentication. Therefore, we RECOMMEND that TRACK SHOULD provide:
   (a) group authentication of control packets, and
   (b) OPTIONAL group authentication of data packets
TRACK MAY choose to provide:
   (c) OPTIONAL privacy of data and of control packet headers.
To accomplish this, we RECOMMEND that, where feasible, TRACK use IPsec technology at the network layer, while letting application level security perform additional functions as needed. For implementations that do not have access to IPsec, and are not implemented as part of the OS, application level security can be used instead--although this of course risks incompatibility with other implementations.

The separation of group-authentication of data from both data source-authentication and data-confidentiality is tightly coupled to the choice of recommending IPsec for the group authentication. These two choices are motivated by the following:

(a) Non-Decipherability of data by interior control nodes: it is a requirement of TRACK that in some deployments, its control entities (RHs) be unable to decipher the data packet. Thus, the GroupDataKey for data encryption and GroupAuthKey for group-authentication SHOULD be distinct. It is not sufficient to simply use an identical key (for the GroupDataKey and GroupAuthKey) and to instruct the TRACK protocol entities to avoid deciphering the data packets. It SHOULD be provably shown that the interior control nodes do not have the ability to decipher the data packets (even if they wish to do so).

(b) Use of IPsec technologies: considerable effort has been invested in developing the IPsec architecture and protocols [KA98a, KA98b], and a growing number of vendors are supporting IPsec. The IPsec suite offers a number of features, including some protection against replay attacks.

(c) Multicast IPsec: the IPsec architecture has intentionally allowed the use of IPsec for IP multicast without changes to the basic constructs within the IPsec suite. Currently, work is proceeding towards the establishment of a standard mechanism to select the Group Security Association (Group SA or GSA) for multicast and a method to disseminate the GSA to the valid members of the group [HCM98,HH99a].

(d) Appropriate Level: since the primary purpose of transport level security is to secure the infrastructure at a transport level, using a network or transport level security protocol allows each to be implemented together--either in the OS, or in the application layer.

6.2 Division of Responsibilities

Given this fundamental division between application and transport responsibilities, we divide the security responsibilities in to four parts.

Network Responsibilities (IP and IP Multicast)

- Admission controls on senders--to protect against "brute force" spamming attacks.
- Authentication of routing control packets--to protect the routing infrastructure from denial of service attacks.

Transport Responsibilities (TRACK)

- Protection of the control messages from replay attacks and other denial-of-service attacks.
- Optional: protection of data packets from replay attacks.
- Optional: encryption of data and control headers to minimize network analysis by attackers.

End to End Responsibilities (Application)

- Source authentication--to verify the authenticity of data, and provide OPTIONAL non-repudiation of data.
- Data encryption--to provide data confidentiality.

Key Management Infrastructure

- Distribution of transport and network layer keys: authentication of individual hosts, and distribution of keys to those hosts
- Application level key distribution: authentication of individual processes, and distribution of keys to those processes
- Optional: periodic rekeying--group keys periodically need to be changed, both after a certain time limit has expired, and/or after the group membership changes.

The figure below shows how these components relate to each other. TRACK can be used without any additional security at the IP/IP Multicast level, although this will not provide full protection from denial of service attacks. TRACK will be able to be used on top of UDP or raw IP/IP Multicast. A TRACK protocol can use either IPsec or application level security for its network security requirements, although we RECOMMEND using IPsec wherever possible.

```
                       +--------------------+       +----------+
+--------------+<----->|    Application     |<----->|  App Sec |
|              |       +--------------------+   |==>|          |
|              |       |       TRACK        |<--|   +----------+
|     Key      |<----->+          +---------+   |-->|          |
|  Management  |       |          |   UDP   |       |   IPsec  |
|              |       +--------------------+       |          |
|              |<=====>|  IP, IP Multicast  |<=====>|          |
+--------------+       +--------------------+       +----------|

  <===> Optional
```

For example, when a data packet is to be sent to the multicast group, the Sender/Source first (optionally) enciphers the data packet using the GroupDataKey above the RM/transport layer. It is then passed to the RM/transport layer, which attaches the necessary RM headers. The result is then passed down to the IP layer where IPsec authentication is established (using the GroupAuthKey).

A Receiver in the multicast group would be in possession of both the GroupDataKey and the GroupAuthKey, and thus will be able to first authenticate the data packet using the GroupAuthKey, and then continue to decipher the data packet using the GroupDataKey.

A Repair Head Node (RH) will possess the GroupAuthKey (but not the GroupDataKey), and thus will only be able to authenticate the packet

using the GroupAuthKey using IPsec. After verifying the authenticity of a received data packet, a RH will be able to retransmit the (enciphered) data packet to its children, either via unicast or region-based local multicast. A retransmission of a lost data packet from a RH will be authenticated using a SubgroupAuthKey (see below) which is a symmetric key shared by a RH and all its children Receivers only.

Again, although the current work proposes the use of unicast IPsec and multicast IPsec at the network layer, it does not preclude the use of other authentication technologies at the network layer or at the RM/transport layer. Such technologies, however, will have to address much of the same issues faced by IPsec, including prevention of replays, the creation and maintenance of state (i.e. "Security Associations") associated with the GroupAuthKey, the Sender and Receiver(s), and other features and supporting mechanisms. It is precisely the growing availability of IPsec that motivates the current work to choose IPsec for network layer authentication for both data and control packets.

6.3 TRACK Keys

In general, each node in the hierarchy MUST be able to authenticate itself to the key management entity/server, before it will be allowed to receive any of the below keys. We assume the implementation of a key management infrastructure, which interfaces with the RHs, as well as the Senders and Receivers.

We propose that this key management system be responsible for distributing the following TRACK protocol keys:

- GroupDataKey:
  The GroupDataKey is the unique symmetric key for data encryption shared by all members of a multicast group, excluding the interior tree entities. Typically, one GroupDataKey is associated with one multicast group. The GroupDataKey is used to provide access control to the data packet by way of the Sender/Source enciphering the data packet. Since only the Receivers hold the copy of the GroupDataKey, only the Receivers will be able to decipher the data packets. This is an OPTIONAL application key, which does not directly concern the TRACK transport.

- GroupAuthKey:
  The GroupAuthKey is the unique symmetric key shared by all members of a multicast group, including the interior control nodes. One GroupAuthKey is associated with one multicast group. The purpose of

the GroupAuthKey is to provide authentication of the (possibly enciphered) data packets. In the context of IPsec authentication, this can be achieved using a keyed hash function, such as HMAC-MD5-96 [MG98a] and HMAC-SHA-1-96 [MG98b].

- SubgroupAuthKey:
  The SubgroupAuthKey is the unique symmetric key shared only by entities within a given local region, consisting of a RH and its children (consisting of one or more Receivers, and possibly one child RH). The SubgroupAuthKey is used by the parent in a local region to provide group-authentication for the (lost) data packets (still enciphered under the GroupDataKey) which are retransmitted to the Receivers in the region via local multicast. The SubgroupAuthKey is also used by the entities in a region to group-authenticate control messages that are exchanged with each other. Similar to the GroupAuthKey, we propose the use of IPsec based authentication via keyed hash function.

  Note that for region-based retransmission of lost packets and for control-packet authentication, the SubgroupAuthKey is used instead of the GroupAuthKey (not both).

  Note that if a RH happens to be a child within a region and at the same time a parent within its own region, then that RH will hold two distinct SubgroupAuthKeys corresponding to the two regions.

- InteriorNodeKey:
  The InteriorNodeKey is a symmetric key shared by all interior control nodes within a given TRACK hierarchy. The key is independent of any data stream, and is used to authenticate control packets exchanged among the RHs. Should a child-RH request a retransmission of a lost data packet from its parent-RH, then the parent-RH will deliver the (encrypted) lost packet to the child-RH, authenticated using the InteriorNodeKey.

  Before the child-RH retransmits this lost data packet to its own region, it MUST first authenticate the packet from the parent-RH using the InteriorNodeKey. It MUST then use its own SubgroupAuthKey of the region headed/parented by that child-RH to provide authentication for the retransmitted data packet.

7. Limitations

The proposed security architecture has certain limitations. These include:

(a) Brute Force Attacks. At the transport level, no admission controls can be put in place to throttle a sender which is generating lots of spurious packets to a multicast address. This is the requirement of the network level. At the present time, no accepted standard exists for doing so at the IP Multicast level.

(b) Key Corruption. The recommended group key architecture makes a careful tradeoff between the need to distribute many keys, and the need to localize the effects of a node which is compromised. This proposal allows a local receiver to perpetrate denial of service attacks to its local RH, and the receivers served by that RH.

(c) Privacy. In order to prevent network traffic analysis attacks, the group keys can be used with IPsec to encrypt the packets sent to the group, in addition to doing packet authentication. However, it must be recognized that this is not a general solution for data privacy. In particular, the group keys can easily be passed from a valid receiver to an unauthorized receiver, to enable piracy of pay-per-use services. This is reasonable, as data privacy is not considered part of the scope of TRACK.

(d) Multicast IPsec. Although currently IPsec is generally implemented for pair-wise (one-to-one) communications between one sender and one receiver, the design of IPsec itself allows for usage in IP multicast. Currently the Security Association (SA) definition requires the Security Parameter Index (SPI) to be selected by the receiver [KA98a]. However, since in IP multicast a group address may be associated with multiple receivers, the existing method of selecting the SPI must be re-interpreted. Hence, in the context of "Multicast IPsec", a pre-defined entity (e.g. the source, or the key server/manager) MUST first create the Group-SA (including selecting the SPI) and deliver the Group-SA to all the members of the group (by either the "push" or "pull" paradigm). Thus rather than being a modification to the IPsec specification, this requirement simply means that additional protocols are needed to establish a shared Group-SA. One possible approach for the Group Key Management (GKM) protocol is to also deliver the Group-SA (and other keying material) to the receiver at the same time it delivers the GroupDataKey [HCM98].

8. Summary

In summary, in the current work we have proposed for TRACK the separation of data stream confidentiality using a symmetric key (i.e. implicit group-authentication) from data authentication using a symmetric key (i.e. explicit group-authentication). Data stream confidentiality using a symmetric key SHOULD be conducted at the application layer, while data authentication using a symmetric key SHOULD be conducted at the network layer. Since the RHs MAY be prevented from reading the multicast data, two (2) different symmetric keys SHOULD be deployed corresponding to the needs of data stream confidentiality and data group-authentication.

This proposal follows a number of requirements, some of which are specific to TRACK. The use of group-authentication at the network layer is:

- for protection of the transport and IP headers.
- to allow a RH to authentically retransmit lost packets to a destination address (unicast or local multicast) different from the original multicast group address.
- to allow a separate symmetric key encryption to be applied (at the application layer) in order prevent the RHs from reading the data.

We assume encryption for confidentiality (using the GroupDataKey) has been applied above the transport layer by the sender/source, in order to prevent a RH from decrypting the data. The GroupDataKey is only available to the group members, excluding the interior control nodes.

We propose the use of another key (GroupAuthKey) to provide group-authentication from the source/sender at the network layer using symmetric key cryptography. The GroupAuthKey is known by the members of the group, as well as the interior control nodes.

For the retransmission of lost packets to regions within a group, either via unicast or local multicast, we propose the use of a SubgroupAuthKey (instead of the GroupAuthKey) which is known only to entities within a region (RH and its children). The SubgroupAuthKey is also used by the entities in a region to group-authenticate control messages that are exchanged with each other.

9. References

[B97] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[HCM98] T. Hardjono, B. Cain and I. Monga, "Intra-Domain Group Key Management Protocol", work in progress, draft-irtf-smug-intragkm-00.txt, September 2000.

[HH99a] H. Harney, A. Colegrove, E. Harder, U. Meth, R. Fleischer, "Group Security Association Key Management Protocol", work in progress, draft-ietf-msec-gsakmp-sec-00.txt, March 2001.

[KCWP00] M. Kadansky, D. Chiu, J. Wesley, J. Provino. "Tree-based Reliable Multicast (TRAM)", work in progress, draft-kadansky-tram-02.txt, January 2000.

[KA98a] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol", IETF, RFC 2401, November 1998.

[KA98b] S. Kent and R. Atkinson, "IP Authentication Header", IETF, RFC 2402, November 1998.

[MG98a] C. Madson, R. Glenn, "The Use of HMAC-MD5-96 within ESP and AH", IETF, RFC 2403, November 1998.

[MG98b] C. Madson, R Glenn, The Use of HMAC-SHA-1-96 within ESP and AH", IETF, RFC 2404, November 1998.

[R92] R. Rivest, "MD5 Digest Algorithm", RFC 1321, April 1992.

[RSA93] RSA Laboratories, "PKCS#1: RSA Encryption Standard", Volume1.5, No. 1993.

[WBPM98] B. Whetten, M. Basavaiah, S. Paul, T. Montgomery, "RMTP-II Specification". Work in progress, draft-whetten-rmtp-ii-00.txt, April 8, 1998.