IRTF SMuG WG August 2000 Meeting Summary

Agenda

1. Agenda bashing - Ran, Thomas
2. Multicast Security Policy Building Blocks - Patrick
3. Group Policy Building Blocks - Hugh
4. Administrative matters - Ran
5. Multicast Data Transforms - Ran
6. TESLA - Adrian
7. Group ISAKMP - Mark
8. Next Steps - Thomas, Ran

1. Agenda bashing - Ran Canetti, Thomas Hardjono

Ran and Thomas welcomed the WG members and proposed the above-listed agenda to the group, who accepted the agenda.

2. Problem Area 3: Multicast Security Policy Building Block - Patrick McDaniel

Patrick, Hugh Harney, Pete Dinsmore and Andrea Colgrove are working on multicast security policy definition and multicast security policy requirements. Group security policy is the security-relevant parameters and facilities used to implement a secure group, such as a secure multicast group. It must answer how security directs group behavior, who are the participants, and what mechanisms are used. Specifically, Group Policy must define the following.

- a unique identification of the group and each participant
- Two sets of mechanisms
  Policy Area1 - what data transforms are to be applied?
  Policy Area2 - when/how group is going to be re-keyed?
- authorization - such as credentials needed to join the group
- access control - in what capacity can previously-identified authorities participate in the group
- verification: each policy must present evidence of its validity (e.g., sigs)

We are working on 5 Multicast Security Building Blocks.

- Identification
- Mechanism
- Authorization
- Access Control
- Verification

Hugh will describe each of these in more detail.

3. Group Policy Building Blocks, Hugh Harney

This presentation covers two topics.

- The 5 policy building blocks

- The relationships to the Key Management building block.

The first block is Identification. Identification is the unique mapping of policy to group: must be able to identify policy to multiple secure groups under an single IP address for secure multicast groups.

The second block is Authorization. Authorization reflects permissions for security relevant actions, enforces the policy system for structure and mechanisms, and forms trust relationships for the group. Authorization assignment is application specific: Authorization will differ depending on the environment and use.

The third block is Access Control. Access control defines group membership criteria. The Policy system provides structure to enforce key access control.

The fourth block is Mechanisms. This is system specific. Multicast IPSec, for example, uses SPD and SAD entries, as well as source authentication transforms as mechanisms for implementing policy.

The fifth block is Verification. Freshness, integrity, and origin are important elements of Verification.

The GCKS, Member (sender and receiver), and Remote GCKS are the SMuG Key Management Building Block includes three entities, the GCKS, Group Member (GM, sender and receiver), and Remote GCKS. Each of these are explicit roles. Group owner (GO) and local policy authority are implied roles.

Here is a mapping of group policy to the group key management building block.

- policy creation - GO
- policy modification - GO
- grant rights - GO
  capability to give information to another entity outside the group
- key creation - GCKS
- key dissemination - GCKS, remote GCKS
- rekey action initiation - GCKS, remote GCKS
- authorize memberships - GCKS, remote GCKS
- admit member - GCKS, remote GCKS
- eject member - GCKS, remote GCKS
- audit group - GCKS, remote GCKS
- key access - GCKS, remote GCKS, GM
- verify authorities - GCKS, remote GCKS, GM

DISCUSSION:

A question was asked regarding the role of policy server of the SMuG Reference Configuration and its interface vis a vis key management? Hugh answered that the policy BB sub-group is defining the policy payloads, protocol, bindings, etc. that entail policy exchanges between the "policy server" and "GCKS" as well as between the GCKS and members. Further discussion concerned the nature of the protocol exchanges. The group key management protocol carries policy. It is not clear what protocol exchanges will occur between policy servers. Best approach may be to define a simple example, 1:N broadcast application, with no negotiation and local policies.

There was some discussion on carrying group policy information in announcement protocols and the issue of having multiple secure groups associated with a

multicast group address or vice versa. An opinion was expressed that secure groups should use session identifiers in place of IP addresses. A "secure group" is defined as a group of participants who have access to a secret and the group may be identified by some name associated with this secret.

4. Administrative matters.

Ran reported that we are looking for someone to host web page and mailing list. smug@cs.umass.edu is the mailing list. Carsten Borman volunteered to help with the web page.

5. Problem Area 1: Data Transforms, Ran Canetti

The draft document was distributed to group prior to the last SMuG WG meeting, remarks were collected and fed back into a revised draft, which was subsequently posted to ietf-drafts. The team working on MESP, AMESP and TESLA feel that these are ready to move to IETF. This presentation reviews MESP and AMESP.

Three services are offered by the multicast data transforms

- GA is group authentication based on MAC technology that uses a symmetric key shared by the group.
- SA is source authentication that uses one of several schemes for authenticating a source that is sending to the group.
- GS is group secrecy ("secrecy" may not be the right word since a movie may require access control rather than some privacy service) based upon symmetric encryption using a shared group key.

There are several deployment considerations for these services.

- Should transforms include only a single BB or how should the BBs be aggregated
- What are the interactions with other application protocols such as reliable multicast (RM) protocols? We believe that SA is easier if RM is provided, FEC-based RM needs SA on individual packets/frames, and retransmission-based RM needs the encryption to be done "above" the RM transform unless repair-nodes need access to group secrets.
- There are several considerations for placement in communication layers. Source authentication is often (but not always) computationally intensive. Large messages/packets are preferable for many or most SA algorithms and state kept across messages/packets, which is problematic for internetwork operation. TRAK states that source authentication should be in the application layer
- Data encryption is not much of an issue as standard encryption exists and can be used as is done in existing protocols (ESP, SSL/TLS), it's not constrained to message size and there's no need to keep state across messages/packets, and data encryption can be done at any layer
- Order of application of the GA, SA and GS services has a couple of considerations. Essentially any order is sound cryptographically (but for non repudiation, it's better to do SA before encryption). GA is easy but weak.

We recommend a proposed design that has

- Two identical transforms. One transform is in the IP layer and one above IP layer.
- Each transform can provide full SA, GS and GS functionality
- One transform is the IP-layer transform: MESP, which is an extension of ESP.

DISCUSSION: it might be cleaner to encapsulate a new protocol id encapsulated inside ESP; issue is support for RMT and TRAK. Brian suggested an alternative encapsulation that chains the secondary ESP having, say, SA, to be pointed to by the IPSEC ESP and to point to transport header. Adrian mentioned that the next header points to a encrypted header and so this seems to be a problem.

o The second transform is the application-layer transform AMESP, which is a bump in the stack that takes an RM message and adds AMESP header and then hands it back to RM so that the RM transform can run without having a security transform beneath it.

o There remains the issue of application of order of the GA, GS and GA transforms within MESP and/or AMESP. There are three usage patterns

i. Everything in application layer such as
  + amesp with ga[enc[sa[data]]]
  + amesp with sa[enc[data]]]
  + null MESP

ii. Everything in IP layer

iii. SA in app, GA+GS in IP layer
  + amesp with sa[data] (null encrupt + external auth)
  + mesp with ga[gs[data]]

iv. There are a variety of valid combinations, e.g. SA in IP GS in app

Note that in many source authentication applications, the external auth data is variable length (or set to some inefficient maximum).

DISCUSSION: The question was asked: "Do RMT people like the bump in the stack approach of AMESP?" And the answer from those in the meeting seemed to be "In general, yes." It was recommended that WG participants read ietf-rmt-trak-pi-security-requirements.txt.

6. TESLA, Adrian Perrig

TESLA is a multicast source authentication algorithm that overcomes the inefficiencies of using asymmetric cryptography on a packet-by-packet basis by amortizing the signature over a sequence of packets that use a low-cost message authentication code for each packet. TESLA's efficiencies come from delayed authentication of a chain of packets that have MACs computed by a chain of keys. TESLA has the following properties.

- o low computation and communication overhead
- o perfect robustness to packet loss
- o unidirectional data flow
- o no sender side buffering
- o delayed authentication
- o high authenticity guarantee

TESLA provides much of the functionality of digital signatures, which are impractical today for most packet or message-based authentication. TESLA cannot provide non-repudiation. TESLA also makes several assumptions about the particular group.

o The sender and receiver are loosely time-synchronized on the order of a single RTT. The receiver, moreover, knows the RTT dispersion of the group.

o It's possible to use digital signature for bootstrapping

o There exists a cryptographically secure prf and mac

o TESLA may be bootstrapped using some digital signature system

Sender setup proceeds with the definition of an interval having a beginning time and an interval duration. During operation, the sender generates a key chain using pseudo-random function G and it is said that the sender commits to the key chain and reveals (signs) one element of the chain of keys after some disclosure delay. As each key in the chain can be derived from the disclosed key, the sender is thereby authenticated.

Receiver bootstrapping requires loose time synchronization where receiver knows the RTT dispersion of the group, beginning time of a specific interval, the interval duration, the key chain commitment (from the keys used in the MACs of individual packets or messages), and the disclosure delay. As part of its bootstrap procedure, the receiver needs the public key of the sender.

During sender to receivers operation, the per-packet overhead is as low as 20 bytes/packet (if we skip all optional fields), as it is not necessary to disclose each key in each packet because one can derive each key from its successor in the key chain. The authentication information then of a message, P2, is the key from the preceding interval K1, and the contents used in the hash of P2, called D2. The receiver stores the packet arriving during the disclosure interval to ensure that the Security Condition is preserved: An attack may succeed if attacker can get keys before a receiver gets the packets disclosing the keys. Thus, the sender and receiver are weakly time-synchronized (+- delta) to preserve the security condition for packet p: The reciver is certain of authenticity when packet p arrives before sender discloses Kp. The packet gets dropped if the security conditions not satisfied.

The TESLA transform applications include authentication in MESP and AMESP header (TESLA is suitable for both internetwork, transport and application-layer services). That is, it can be used in both or either the internal authentication or external authentication header of MESP/AMESP. TESLA is also designed to serve in the source authentication field of the Reliable Multicast Transport ALC. protocol.

7. Problem Area II: Group ISAKMP, Mark Baugher

Mark described the group management protocol that he has been working on with Brian Weis and Thomas Hardjono. Group ISAKMP applies many of the concepts, messages, and payloads of GSAKMP to an ISAKMP framework. This work is a "domain of interpretation" for ISAKMP for Group Key Management; it is a Group DOI or GDOI for managing a structure of inter-dependent security associations as a group security association (GSA). The GDOI establishes SAs to protect one or more group secrets among a group of principals; these secrets protect key encrypting keys, traffic encrypting keys, or data shared by group members.

The first SA in a GDOI GSA is created through a unicast exchange between sender and each receiver as is done with GKMP and GSAKMP. A second SA is optional for re-keying of the group and for group membership maintenance using hierarchical algorithms such as Logical Key Hierarchy and OFT. This second type of SA (called a "Category 2 SA" in GDOI parlance) is one-way and is suitable for "push" over a multicast connection (though unicast service may be used as well). The group secret used for maintaining group membership is the Key Encrypting Key (KEK). The group secret used to control access to group data is called a Traffic Encrypting Key (TEK). Just as a tree of KEKs is used by LKH and One-Way Function Trees for group maintenance; refresh of the TEK is accomplished by encrypting it in the KEK. Thus the KEK is used for access control to the TEK

without requiring costly unicast exchanges with each member and a central key server.

The TEK thus protects traffic between the sender and receivers and it's the keying material of the third type of SA in a GDOI GSA (the Category 3 SA). Establishment of the TEK for streams or files is the goal of the GDOI. It is possible to establish the TEK for internetwork, transport and application-layer services. The GDOI allows the TEK to be established solely using a point-to-point exchange between the GCKS and the member. The Key Management datagram may be used for pushing a TEK, encrypted in a KEK.

The authors plan to post the draft prior to the next SMuG WG meeting. NRL has agreed to review the draft prior to posting. The posted draft will leave LKH and OFT support as TBD, the authors also plan to incorporate the work of the group policy team in a future revision. Support for MESP, AMESP and TESLA will be in the first revision of the draft.

DISCUSSION: Ran was concerned that the GDOI cannot run over IPSec, SSL/TLS, ssh, and other security protocols but only over an IKE phase 1. Ran suggested that we work on a "resolution document." Mark thought that there was no need for a resolution document since one of the main things that distinguishes GDOI from GSAKMP and the generalized exchange in the GKMBB draft is that GDOI uses the ISAKMP framework. Thomas said that he was planning of posting an updated version of the generalized key management building block draft and that might allow further experimentation with different group key management within the context of a generic set of messages and payloads.

8. Next steps - Ran, Thomas

GKMBB team should work towards consistency of Group ISAKMP and GSAKMP once Group ISAKMP draft is reviewed and published. We would like to have implementations of the key management, (A)MESP, and TESLA in the Fall. Some members of smug will likely conduct a secure multicast BOF at the next IETF meeting.

DISCUSSION: A couple of the participants from RMT voiced support for the progress the SMuG has made and said they feel good about the output of the IRTF SMuG WG to date.

Attendees

Thomas Hardjono	hardjono@nortelnetworks.com
Tanja Zseby	zseby@fokus.gmd.de
Cathy Meadows	meadows@itd.nrl.navy.mil
Mark Baugher	mbaugher@passedge.com
Armin Haken	armin@digitalfountain.com
Brian Weis	bew@cisco.com
Michael Luby	luby@digitalfountain.com
Adrian Perrig	perrig@acm.org
Peter Dinsmore	peter_dinsmore@nai.com
Andrea Colgrove	acc@columbia.sparta.com
Hugh Harney	hh@sparta.com
Carsten Borman	
Brian Whetten	whetten@talarian.com
Lakshminath Dondeti	ldondeti@nortelnetworks.com

Roger Kermode roger.kermode@motorola.com
Aidan Williams aiden.williams@motorola.com
Patrick McDaniel pdmcdan@eecs.umich.edu
Ran Canetti canetti@watson.ibm.com