
SMuG.0 8/28/98
Chicago, IL (Co-located at IETF 42)

"SMuG v0 since not yet approved as research group
by IESG, but hopefully the next meeting will be v1."

Chairs:
Dan Harkins <dharkins@cisco.com>
Ran Canetti <canetti@watson.ibm.com>

Attendees (no particular order):
Jamie Jason <jamie.jason@intel.com>
Thomas Hardjono <thardjono@baynetworks.com>
Brad Cain <bcain@baynetworks.com>
Shlomo Kip <skipnis@ndsisreal.com>
Cathy Wittbrodt <cathy@corp.home.net>
J.R. Rao <jrrao@watson.ibm.com>
Pankay Rohatgi <rohatgi@watson.ibm.com>
Rob Adams <adams@cisco.com>
Ross Finlayson <finlayson@live.com>
Dah Ming Chiu <chiu@east.sun.com>
Brian Whetten <whetten@julia.gcast.com>
Mark Handley <mjh@east.isi.edu>
Bob Quinn <rcq@ipmulticast.com>
Pau-Chen Cheng <pau@watson.ibm.com>
Hugh Harney <hh@columbia.sparta.com
Debanjan Saha <debanjan@watson.ibm.com>
Ran Aneth

Dan Harkins noted that the proposed RG charter wAS sent out
to the SMUG mailing list.
- Dan read charter
- Summary: we will create a taxonomy of problems (applicaton
  requirements), and outline the solutions needed.

Question (Q): Why is this a closed meeting? Dan answered that
it's not really closed; we won't kick people out. It's just
that we wanted to avoid filling the room with those with
just a passing interest.

- some discussion of whether or not to have it open and co-located
- Mark Handley suggested we have some connection with activities
of the reliable multicast (RM) research group (RG) of
the Internet Research Task Force (IRTF), since they are so closely
linked (and secure multicast may need RM). General agreement.
(Ed Note: see http://www.east.isi.edu/RMRG/ for more information)

Ran Canetti: I would like the group to come up with a protocol to
solve some subset of the problem
- Q: creating a protocol should not be an absolute goal, but may
be a by product
+ general agreement

  + general agreement that using/creating some solutions to reference would be helpful, if not necessary
  + Mark H. said in RM RG we have decided on many and we are identifying portions that can solve - we are creating a strawman
  + Dan: we will come up with a strawman for single center vs. distributed key center and send it off to IETF for a protocol definition
- Mark H: RM group is to define/understand the problem and then spawn an IETF work group (WG)
- Ran: We will create a roadmap of components, APIs, architecture
- Q: Is there a publication mechanism for RM RG?
- Mark: Now RM just has everything on a website, but eventually we'll do an RFC to describe the conclusions of the RG from which WG work can be initiated
- Dan asked for volunteers to do a website, and perhaps takeover the SMUG mailing list. No responses.
- mailing list membership is gated now (manually) by Dan. Mail to mailer@cisco.com with "subscribe smug <email address>" in the body. Mail list address is smug@external.cisco.com

---

Ran Canetti - A Taxonomy of Multicast Security Issues
draft-canetti-secure-multicast-taxonomy-00.txt

The goal is to define taxonomy to characterize the problems, and next to make suggestions as to possible solutions

Multicast group parameters
- size
- membership dynamics (dyanmic or static)
- life-time
- number of senders (known identiies or not)
- volume of traffic

<< none of above deal with security >>

Multicast security requirements
- Access control
- Authentication
  + src (individual, group)
  + data
- Secrecy
  + long term (strong)
  + ephemeral (short term) (weak)
- Anonymity
  + from group members
  + from outsiders
- Availability

Q: There's also receiver access control.
Ran: we should NOT use data to do authentication
Two aspects (from senders vs. receivers point of view):
- who can make sense of the data
- can clients authenticate sender

Q: I would also add that just having a key should NOT be sufficient

Q: There's two ways to do access control

- One requires a change to multicast model (e.g. IGMP Authentication,

  draft-ishikawa-igmp-auth-01.txt) We should state that we do not want to change the multicast model (should be in the charter). It should not be done at the network layer. Should be independent of the routing algorithm.
- Dan to Bob (note taker): Did you get that? Bob: Yes.

Trust Issues
- trust in a single center- trust intermediaries

Levels of trust:
- to relay keys
- to generate kys
- to correctly encrypt/authenticate
  + to decrypt/reincrypt using a different key

Our main goal is to distribute trust.

Q: I am not sure I agree with that statement
A: We are dealing with minimizing risk, essentially. Could design system with a single center that does all exchanges. Works as long as system is secure. Then you could decentralize that system, and create other versions.
Q: As long as you don't want to make the statement that by distributing, security is stronger
A: No, we don't mean that. Just that we could use third parties (e.g. Certificate Authorities).
Q: this may not be a technical issue, but a business issue. On a distributed network, the transit networks don't want any liability for content.
Q: I would add minimal exposure key (trusting people who have the key is not strong enough).

Performance Parameters
- setup time
- member enrollment/revocation overhead (and rekeying)
- time to authenticate + encrypt (sender)
- time to verify and decrypt (receoiver)
- communication overhead
- length of keys
Q: there's another issue of scalability in general

Benchmark Scenario
- single senders with strong machines
- many recipients (100K - 1M+) with weak machines
- dynamic, bursty membership
- weak/ephemeral secrecy needed

- strong authenticity needed
- examples
  + pay per view movies
  + net cast
  + stock quotes

At IP Multicast meeting/workshop in NYC a couple months ago (ISPs in attendance) said this is the most significant case from their perspective.

2nd Example
  + -medium small groups (100s - 1K)
- many senders (unknon in advance)
- slowly changin membership
- strong secrecy (sometimes)
- strong autheticity
  + may not need to know who is "talking"
- examples
  + games
  + video conferences
  + lectures, classes
  + town hall meetings
  + distributed simulations
Q: I would possibly put the first three in the one-to-many category
Q: The secrecy requirements are very different for the two examples.
Ran: There are many more than two, really

Two algorithmic problems
- individual (source) authentication
- member enrollment/revocation

Q: We don't have a user identity, so we are doing host/interface authentication, and in that case we have IPSec available.

Source Authentication
- Problem: how to verify identify a sender? (shared key is not enough)

Possible solutions
- Signatures
  + expensive computationally
? could sign every nth packet
? use stream signatures
? use signatures with cheap verification
- A different approach: MAC with multiple shared keys (i.e. not scalable)
  + more efficient for small/medium sized groups
  + secure only against limited-sized coalitions (a consipiracy of users could forge keys ...though increasing the number of members/MACs can help to thwart this, it adds to overhead)
  + sender know all keyseach receiver has portion of key and verifies/generates only some of the MACs
  + You can avoid linear growth (scalable problems) by creating pools, doing some combinatiral work with keys

Q: We should not consider computation overhead a major challenge
(since Moore's law is still in effect, it is less of an issue all
the time)
Q: We should segment the problem into two big ones:
- distribution of the group key (PIM and Sandra Mooke (sp?))
- authentication

Membership Deletion
- when member joins/leaves, key has to change
  + how do you distribute the updated key?
- Solutions:
  + encrypt new key with old key -- > insecure (there's actually

    a draft that says a leave should give key back to sender and

    erase their copy (yea, right, sure we'll do that!)
  + encrypt new key individyually for each member --> too much work
  + use many sub-domains, each with its own key -> have to re-encrypt
    each message
  + use tree scheme, saving on ecryptions (depth of tree rekeyed,
    not number of receivers)

Member deletion by Wellner (reference??), et al (RSA)

Suggested Design Principles
- make security module independent o the routing moduel (done change
routing of data packets, and make it router protocol independent
- separate of app layer models (key mgmnt) from IP layer modules
  (packet transport)
- use existing mechanisms as much as possible (IKE, AH, reliable
  multicast routing, etc).
Dan: these should probably be additions to our charter

Suggested Basic modules/architecture (30,000 foot view)
- Multicast center
  + membership ontrol
  + key mgmnt
- sender module
  + key mgmnt
  + packet transformation -> at IP layer
- Receiver module
  + key mgmnt
  + packet transformation -> at IP layer
--> most of this happens at the application layer (except where noted)
--> this needs reliability

Q: We don't want to limit ourselves to IP for network layer. We
don't want to get too close to the network layer.
Q: There are different domains of this problem that have different
requirements
Q: The most significant policy issue is whether perfect forward
secrecy is a requirement
Q: I'm seeing a requirement to be able to revoke a membership for

someone untrusted
Q: The current pay per view security is not fine granularity - rekeyed about once per day
Q: There are also other special cases to consider (e.g. last ten minutes of a football game)
Q: We don't necessarily need reliability.

Debanjan Saha - Prototype Toolkit

Designing Prototype requirements
- large groups
- extensible

Overall architecture shown (two tiered trust and data distrubution)
- Distributed controller & reflector with local members
- Group owner
- No mechanism for accounting and billing
Q: Do you pay the group owner or local distribution?
A: Depends. Probably group owner.

Domain Architecture
- data plane and control domain are conceptually separate
- control plane is the focus here
- can handle either one-to-many or many-to-many

Control messages - different aspects
- cleint initiated
  + registration, join group, leave group
- conroller initiated
  + expel client

For perfect forward secrecy, you need to change keys when member leaves, but alternative is periodic key changes.

Control Messages
- uses RSA for authentication (using SSL)
- key updates are on reliable multicast channels (using SRM)

Showed message flows
- for group controller joins
- for cleint member joins

Currently they have a version of Wellner scheme (reference???), and update keys periodically
- showed update session key format
- can support multiple auxiliary keys, if needed

Data Plane
- encryption/decrypt is transparent
- socket-like send/receive API
  + can be turned on/off sing a flag
  + facilitates partial encryption

Architecture block chart shown for sender and another for shown
for client.

Q: Is there a way to verify that you have the updated key?
A: It is done periodically and there's a way to request an update

Planned demo at IETF 42 (Ed Note: Was it here??) and RSA Conference
Applications
- Stock distribution
  + authentication
  + real-time low data rate, reliable
- Audio Video
  + authentication
  + real-time, high data rate, unreliable

Thomas Hardjono - Multicast Security Framework
draft-ietf-ipsec-gkmframework-00.txt

Motivations and basic notions
- A single Internet wide Group Key Management (GKM) protocol is
  difficult.
- Our requirements
  + scalabliltiy
  + routing independent
  + business reuirement
- Divide problem of multicast security into regions

  + intRA region

  + intER-region
  + there's a draft by Dave Thaler for routing interoperability

    (Ed Note: see draft-thaler-multicast-interop-03.txt)
- Analogous to routing protocols
- Need a simple framework & interoperability rules for multiple
   solutions

Basics for framework
- multicast app type (many-2-many, one-2-many)
- scalability considerations
- <missed others>

Key management plane
- single trunk region"" (a logical unit: an AS)
- One or more leaf "regions"
- boundary demarcated by key manager (KM) entities
- trunk region - no member hosts (only KM entities)
- Leaf Regions
  + member hosts defined to exist in leaf regions
  + associated with (at least) one KM
Q: Why can't a receiver also distribute keys? Why must the KM not
   be a member host?
A: They can, this is a logical description

Key Management Plane illustration shown
- three clouds: leafs on each end of a trunk cloud
- KM between clouds
- other KMs *inside* trunk key
- KT's in leafs

Interpretation of Regions
- Key Mgmnt zones
  + to create secure channels for delivery of group keys
- Key Application zones
  + each region has different keys
  + region applies key tp payload
Q: key application zones may be necessaty for political reasons
  between international boundaries
- Combination Key Mgmnt and Application Zones

Advantages of framework
- Promotes scalability
  + leaf region added as required
- Reduces complexity
  + 2 levels
  + single purpose Internet solution becomes unnecessary
- Independent trunk region key mgmnt protocols
  + longer lifetime of trunk key
  + ease of grafting new (or resurrected) leafs
  + independent re-key periods (app dependent0

 Brian Whetten - Existing RM Applicatons

Application taxonomy
- Unreliable vs reliable
- Real-time vs non-real timne

Even things previously unreliable, we are seeing a need for
reliability to provide QoS

Key driving forces are ISPs and contet providers
- pay per use
- value added services
- one-to-many is the key app
- user tracking and billability

App Types
- low bandwidth real-time data (stock)
- 256Kb multimedia
- File transfer

Access control is the Big Thing.

Globalcast uses a suite of RM protocols:
- RMTP-II
- PGM

- G-SRM (hasn't sold a single copy yet)
- RMP

Typical Deployment Scenarios
- Uses a "Repeater" (a local distribution point)
- Can have different protocols in diferent "repeater regions"
(so it's like a caching mechanism)
- They do integration between the proxies (at the upper levels)
  so you can have a local proxy for whatever application you are
  doing (and that way the lower layer can be whatever ...but messes
  up the per-user authentication and billing)
- The goal is for ISPs to be distribution points, but they don't
  want any security liability ...which presents a problem for
  this architecture

We believe the 1st solution is the simplest: one key per group.
- we do it at either IP or app layer
Q: Which is better?
A: Seems that data at IP layer and key mgmnt at App layer

If you have reliable multicast you can do rekeying easier
- most apps use access control initially, then a static key

Multicast group ingress and different one (or unicast) egress
- makes problem more complex

Brian welcomes any input to security issues
- to protect against Denial of Service (DoS) attacks, specifically

Shlomo Kip: I have a list of security issues and scenarios,
  and would be willing to write them up.

Dan Harkins - Multicast Key Mgmnt Protocol (MKMP)

- draft available (presented at IPSec meeting) by Dan Harkins,
Cisco and Naganand Doraswamy, Bay Networks
(Ed Note: I was unable to locate the draft)

- There's a copyright notice attached

MKMP features
- scalable
  + key distribution delegated
  + acess control enforced
  + key distribution authrnicated
  + key dsiribution provides "live-ness"

Announcement of group done in SAP/SDP

- key owner sends announcements with list of (Photuris-like) cookies

- Routers (network entities) are distributing keys

- message sent with Router Alert (RA), see RFC 2113
  + RA capable routers will pay attention and parse packet
Q: Using router alert is one implementation
Q: You should take out RA and just have router join group address
- Only routers already in distribution tree become GKDs
- GKDs must prevent themselves rom being pruned

At first, members get key from sender, but then as multicast (router) tree is formed, members get key from local distribution points. Basically a flood and prune mechanism. Those that are authorized
Q: Using TTL is bad for scoping

Dah Ming Chiu, Sun
Flexible Reliable Multicast Service

Java Reliable Multicast Service
- TRAM - Tree-based ReliAble Multicast transport (Ed. Note: See London RM meeting presentation http://www.east.isi.edu/RMRG/london/kadansky.ps.gz)
- Includes API
- Does multicast address allocation (MALLOC)
- Security
- Channel (security and other mgmnt)

Security Requirements
- Some apps don't need security
- Data confidentiality
- Sender authentication, data integrity and reliability
- One-time key distribution before session
- Re-keying (periodically, or upon membership changes) for long-lived sessions

Framework illustration shown with sender and receiver authentication
- Senders and Receivers, with Channel Manager (policy config & online registration)

Second Framework illustration shows data-signing separate from encryption

Challenges
- Efficient key distribution (or agreement)
- Collusion resistance
- Management of a large group - registration, access control, initial key distribution

---

Send comments, additions, corrections, questions regarding these notes to Bob Quinn <rcq@ipmulticast.com>