IRTF SMuG Meeting at 46th IETF (Washington DC, 7 November, 1999)
~~~~~~

Scribe: Mark Baugher

List of attendees:

Mark Baugher	mbaugher@intel.com
Bob Briscoe	rbriscoe@jungle.bt.co.uk
Ran Canetti	canetti@watson.ibm.com
Dah Ming Chiu	chiu@east.sun.com
Peter Dinsmore	peter-disnmore@nai.com
Ross Finlayson	finlayson@live.com
Thomas Hardjono	thomas_hardjono@baynetworks.com
Dan Harkins	dharkins@networkalchemy.com
Hugh Harney	hh@sparta.com
Roger Kermode	roger.kermode@motorola.com
Amit Kleinmann	akleinmann@ndsisrael.com
Isidor Kouvelas	kouvelas@cisco.com
Mike Luby	luby@dfountain.com
Patrick McDaniel	pdmcdan@eecs.umich.edu
Cathy Meadows	meadows@itd.nrl.navy.mil
Sasha Medvinsky	smedvinsky@gi.com
Piers O'Hanin	piers@cs.ucl.ac.uk
Dimitris Pendarakis	dimitris@watson.ibm.com
J. R. Rao	jrrao@us.ibm.com
Sanjeev Setia	setia@cs.gmu.edu
Rajitha Sumanaselara	rajitha@cisco.com
Yongguang Zhang	ygz@hrl.com

Agenda

9:30-10:00 Introduction (Ran & Thomas)
- Summary of progress
- BB draft
- Issues regarding BB draft
- Other general SMuG matters

10:00-11:30 Presentations
- Pat McDaniel (20 minutes)
- Bob Briscoe (20 minutes)
- Hugh Harney (20 minutes)
- Sanjeev Setia (20 minutes)

11:30-1:00 BB area-1 discussion
1:00-2:00 Lunch
2:00-3:30 BB area-3 discussion
3:30-3:45 Break
3:45-5:15 BB area-2 discussion

1. Ran, present agenda, skip intro and move to presentations.
   Area 1 (Data Handling), Area 3 (Policy Management), Area 2
~~~~~~

(Key and group management) - note that 3 and 2 are reversed.

2. Presentation:
Multicast Security Policy Definition, Pat McDaniel, H.Harney, Dinsmore, Atul Prakash, November 7th, 1999
3 efforts: Policy creation (mostly today), representation, and translation into problem area 1 and 2 services. Main objectives are to define 'what we mean by policy.' 2 kinds of issues (1) access control (member rights) and (2) group policy.

Guiding principles: Implementable, coherent, useful, and simple ... but expressive.

Local Policy: There are local and group policies; local policy is a file or data structure local to group member; what is the infrastructure, local credentials and trusted parties. Policy requirements: What groups can a local join, what are the conditions, e.g., source authenticity

Group Policy: GSAKMP definition of group policy; access control policy is how to assign rights to members (we propose roles) and the group security policy itself.

Access Control: What rights we provide to group members. List shown includes key creation, key dissemination, rekey action init, key access, policy creation, policy modification, grant rights, authorize member, admit member, eject member, audit

Roles: What roles members perform in the group; unlike object based systems; think of as collection of rights. Role policy defines roles and mapping rights <<--> roles, and then identifies access control by saying who may assume which roles. Currently defined roles: group owner, group key authority, group

Access Control proposal in table form.

Group Security Policy (mandatory) in rekeying policy (pfs), limited lifetime. Data security policy (privacy, group auth). Access control policy. Roles are collection of access rights. {discussion on what access control is used to do; this framework should a view into roles and access controls at a point in time}

Group security poicy (potential) in member data policy (avail) and distribution about membership info and claims about accuracy), compromise policy, failure policy, domain depenedent policy.

Issues: refine rights, roles. refinement of policy dimensions, policy specification, negotiation, policy architecture (protocols, etc.).
/docs/draft-irtf-mcast-pol-def.txt

3. Presentation:
Bob Briscoe, "MARKS: zero side-effect mcast key management using arbitrarily revealed key sequences"; Key mgt problem viewed in

2 dimensions (horizontal & vertical).
The goal: to do key management forgroups with dynamic membership, without re-key messages.
People join and leave over varying amounts of time with a key tree at each point in time. Since rekeying is expensive, an alternative
is Application Data Unit (ADU) approach which has a 'crypto period' during which some non-members receive all or part of the ADU (see taxonomy of large-scale multicast req, Bagnell).
Solution then becomes to rekey at ADU boundary; but you still have a lot of work to do if you are changing membership at each rekey.
Looks like OFT, but does not need to be hashed, just generated.

mcast key management: more solutions require reliable multicast.
MARKS redefines the problem to help with solution; arbitrary eviction, but pre-planned, works well with pre-payment. The key graph gets rotated along the time axis rather than member access; with a graph per member.

loose coupling to senders: distributed key managers with unicast set up between KMs, R's and S's, which does not require reliable multicast - assumes members know in advance when they are going to leave.

Two blinding functions from one and a tree is made out of them starting from the top rather than the bottom with the key sequence in time running along the bottom leaves (e.g., you 'sell' with the blinding of various parts of the tree.

algorithm to re-seed individual keys shown (briefly). BHT per ADU key calculation. << 2 hashes per key (#branches/#leaves) efficiency figures shown, where N is the number of ADU's, not members - complete de-coupled from number of members. As secure as a hash chain; max attacker gain;

Variations include multi-sender multicast; combination with other schemes; unplanned eviction; satermarking.

Biggest imitationn is collusion, arbitrage. Strength of hash chain of length D gaes the hash strength.

Questions/comments:
1. why a tree? to scale in size of key vector logarithmically

Watermark without smartcard...

wider context: not just multicast; can be used in other evironments; consortium sourcing flexinet, Mware to combine security services; declarative: cf. LSMA requirements taxonomy draft-ietf-lsma-...

Summary: no limit on Marks scalability, completely decoupled with no reliable mc required, low setup and running costs, no reliable multicast re-keying. Arbitrary eviction - unplanned more difficult than planned.

Drawback: makes life easy for pirates. A single key (or small number of keys) known in advance are sufficient for decrypting the entire transmission. In particular, a user can subscribe to the entire duration, get the root key and publicize it. This doesn't happen if the key changes periodically.
(There are some counter-measures in the paper.)
further info: Mware project at
<http://www.labs.bt.com/people/briscorj/papers.html#MARKS>

4. Presentation:
H.Harney looked at work that policy subgroup has done trying to fit this work into SMuG. policy subgroup has many options (pfs, CR, etc.), and this is a short overview of the simplest model for setting up a group and its associated policy.

Basic concept has 2 key servers each with one member with a person at key server 1. tradeoff between conf and speed per set of data - a human decision; there are requirements on data protection, group setup, behavior requirements (rekey, CR). Person makes decision, communicates to key server that allows members to get keys; for scalability purposes we want several key servers. Q: double-ended arrow between KSs?
HH asserts that there is some interaction between key servers.

Minimal policy: IDs (policy, group version), Access control (IP, DN), Crypto spec (data protection, groupo establ spec. -- transforms, etc., remote keyserver auth (IP, DN), policy auth data (signature, public cert)

This is a candidate, minimal set. Stake in the sand on what a minimal policy definition would be.

5. Sanjeev Setia, Kronos. Group re-keying for forward and backward confidentiality, several approaches can be defined in two types of schemes; in one scheme there is a key hierarchary and in the other approach is divide and conquer to reduce the overhead by subdividing the group (IOLUS, intra-domain GKMP).

Group rekeying overhead has 2 overheads - individual rekey and frequency of rekeying (depends upon group size and membership dynamics) - we're looking at latter

LKH, OFT addresses scalablity of individual rekeying but not the freqency of rekeying. Iolus rekey freq depends on size and dynamics of rekeying

Our problem: reduce frequency of rekeying, which becomes the bottleneck when rekeying happens frequently. How long is a key to be valid? how long before group key needs to be changed?

graph, #subscribers by avg-tunein- time. in PPV, you don't want to rekey upon membership change. In subscriber model, when does the bottleneck arise. simple analytical model: If sub population is large, then even if members join in for 600 seconds then you will rekeying very often - if they were not staing that long, then rekeying would become too frequent to be feasible. graph shows how large and dynamic a group can be.

{discussion on what is the upper bound, how it is found, and what is practical or impractical or desirable}

Kronos solution: periodic rekeying where frequency of rekeying to be independent of group size and membership dynamics. A distributed scheme can be used in conjunction with IGKMP that divides domains into 'areas.' There is a single group key that does not need to be re-enciphered.

Distributed framwork has DKD and AKD (area key distributor). All AKDs can rekey at the same time if you agree on a lifetime: Each AKD generates the same group key at the same time and multicast it to members in that area. and then use LKH or OFT for groups within the area.

Kronos operation: all akds rekey at same time, using NTP. all akds generate same group key (they share two secrets (K an R0) where r1 =Ek(Ro) and Ri+1=Ek(Ri). reestablish K and R0 periodically.

Pros/Cons: advantages are that this is scalable way to support periodic rekeying with overhad for app being predictable and bounded. {don't know how long messages will be} {another issue is forcing periodic rekey, event-driven is another approach, not clear that you can force one model on everyone}. disadvantage include distributed trust.

Performance evaluation of Ionus, Kronos and LKH. Metrics were join/leave latency, data latency, and time between rekeys. LKH (algorithm that rekeys upon group change).
is limited by frequency of rekeying; Iolus has lowest join/leave latency but data latency is higher. Kronos join/leave is related to rekey period but can get comparable join/leave latency comparable to Iolus.

Conclusion: don't ignore scalability restriction caused by the frequency of rekeying, Kronos approach is scalable, http://www.cs.gmu.edu/nsetia/kronos.pdf

6. Ran - SMuG framework for research and for standardizing specific protocols in IETF WGs. Brief overview of Reference Diagram. Defined a set of functional building blocks for sub-protocol problems dealing with different issues. I. Data handling (BB1 data encryption, BB2 group auth, BB3 source auth), II Group and Key management (BB4 group membership and BB5 Key Management), III Policy management (BB6).

7. Ran - Area 1 data handling.

On the need for 3 'independent' transforms: group auth, encryption, source auth. why do 3? why not join group and source? Order of transforms [[[[data] s.a.] enc] g.a.]. Other issues include layer of application, in/out of kernel, reliability assumed, app-specific.
We made group auth separate because it can be easily done in the network layer, whereas source auth requires more heavy lifting. encryption can be used in the IP layer as an option.
{comments: in diagram, is source authentication encrypted? Ran said 'no'. (diagram only depicts order of transforms. usually there is no need to encrypt the s.a. fields.)
In the key mgt schemes, there is no assumption that the receivers need to know; the group manager. NB: a sender has the group key and thus is a member of the multicast group even if it is not a member of the IP multicast group}
{Question: what do we recommend? Some discussion on what/how SMuG recommends things. NB: this group should not try to solve too broad a problem or it will end up with too many solutions. One approach is that if you have this application makeup, then we recommend this type of solution. Comparison with RM, which has three tracks.}
{Ran: when we agree to the building blocks, for example, then we can have different realizations of each building block, that address different scenarios. Once we have concrete protocols we can discuss what/how to push to IETF? Comparison with IPSec. HH says that if we thought about the scenarios we want to solve so when we hand it off to IETF it is clear what the purpose is. Bob: We are still early in conceptualizing the problem and potential solutions}
{MacD: MBone application document cites application scenarios}
{Where should transforms be implemented: Layer 3?
People want to do an IPSec-like solution - agreed.
Last year draft gave an undetailed treatment on IPSec including SA def., transform definitions, etc.
Probably need something like AH/ESP authentication for multicast; not assuming reliability in L3. New AH for m'cast.
How to standardize L4+ schemes? HH: generic solution that can be put into various applications?
One way to go is to establish a new WG for L3 and possibly layer 4+ - Ran
Source authentication with and without reliability both need to be provided - at the transport or application layer.
different flavors of source auth: In network layer and in L4+. 3 flavors: network, app (reliable), app (unreliable) question raised over whether app is significantly different from transport.

answer: one difference is that in app layer each application/process does its own transforms/crypto. in transport layer the

transform/crypto is done by kernel. this has both advantages and diadvantages.

{
issue of single source multicast (e.g., express, etc.)

not likely to help the problem of source authentication.
}
S.A. I. network layer
II. application layer with reliability
III. app layer without reliability

Rohatgi, McCarthy sit at III. we have nothing for II.
stream signature scheme takes advantage of II although
III applies to II.
There are no universally good solutions for I, multi-macs
don't offer unconditional guarantees in face of collusion.

- o Layer4 looks like app layer.
- o also, II, we are talking about streaming, what is the unit (sequence of packets, NPDUs, APDUs, sets thereof, files)
- o if we could do I, then we would have the entire problem solved.
- o another issue is whether S.A. is done inside the kernel or outside the kernel. Should this be done on a per-node basis versus on a per-app basis? in the kernel?

{ Bob question: is this group's charter on mechanisms or frameworks. A: mechanisms.

Data Handling
Confidentiality (encryption) in the network layer, but not mandated since it can be done in both areas.
Should encryption be done at the application layer?

1. media specific ciphers
2. authenticated execution environment
3. user-specific confidentiality

Wrap up: Need to have:

- o G.A. in network layer
- o S.A. in network layer, partial solutions abound
- o S.A. in application layer, ala some of the drafts
- o confidentiality in network and application layers

8. Problem Area 3: Policy, Patrick McDaniel

i. PA3 - Policy Mgt
Requirements

Scenario - mm distribution
- multiple key server

Future:

ii. Hugh's foil

- o Policy ID
  - group id
  - version
- o Access Control (authorization to get the key)
  - IP

- DN
- o Crypto spec
  - Data protection
  - group establishment spec
    - 3DES/HMAC...
    - DSS
  - remote key server auth
    - IP
    - DN
- o Policy auth data
  - signature
  - public cert

iii. PD's Group Security Policy (Mandatory) such as rekeying
{PD: rekeying is not mandaory}
{MB: issue with using pfs in two different ways}

Thomas: Need a comprehensive requirements document

Patrick: requirements, schenario (mm distri bution, multiple key servers (minimal group); future: Access control description, policy definition, investigate other policy efforts.

Hugh: start with two foils from this morning

Dah Mingh: why do we need a policy server?

HH: need a rationale for why we need a policy server

PM: need a rationale for Area 3

PD: group policies are more problematic than point-point
{discussion on adding and changing Hugh's foil , also see above

HH: Access control talks about criteria that protocol uses to determine whether or not someone should get key... it is access control to the key. Access control and authentication are discusse ;

JR Rao:

Policy is...

...membership management

...data management

...key server management

Bob: do we need to invent any new security policy for m'cast

HH: there are specific distinctions for m'cast

Bob: we don't need to specify unicast procedures between key managers

AK: 3 issues is access control (authentication, authorization) data, key management (length of keys, pfs, PFS)

PD: there are three components, three policy areas, notice that we are really talking about the policy issuing from each of our three components.

JR Rao: what are we not trying

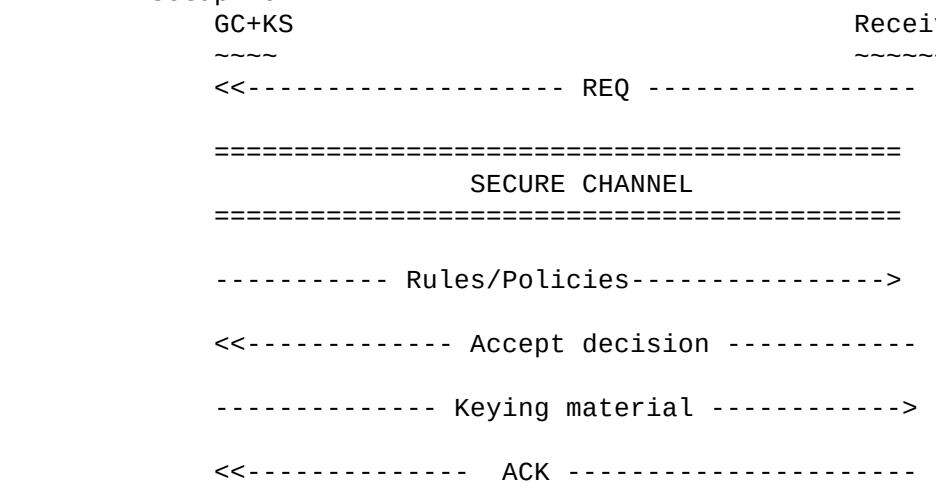
PM: do we want to open up for applications to provide their own policies?

Amit: we are looking for something very simple and this is the options and parameters for operation of the basic components of this systme.

HH: Need to start with a simple subset of policy to get some immediate traction.

Amit: What I expect is all of the options and parameters and a mapping to values.
HH: small subgroup
MB: should we organize by Ref Diagram?
HH: we should use common terminology. reqs document says 1) what are the components, 2) what are the options (?) and then perhaps do with AK says
conlcusion" PD will rework draft, hand over foil to Thomas for key management group to get some input on the alternatives, options and possible values.

9. Area 2, Thomas
setup foil:



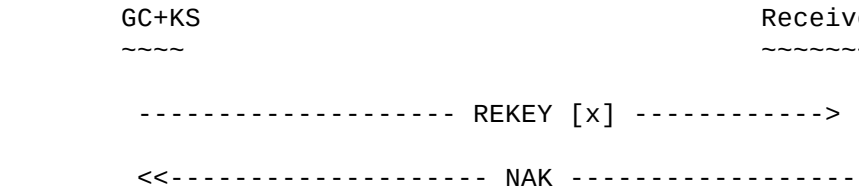
Ran: first exchanges with group membership while the latter are with key server.

Bob: a lot of rules or policies are no longer needed once things gets standardized

Thomas: what if receiver is a GC+KS; e.g., MKMP has this issue
MB: are the servers inter-domain or intra-domain?
Thomas: assuming intra-domain.
HH: are we assuming one crypto groups.
Thomas: but in the inter-domain or intra-domain case we will basically have the same exchange
MB: yes. and there are several variations such as inter vs. intra-domain, crypto grateways (which are necessary if the crypto policies are altered.
*** we have applied this flow to two scenaria ***

re-keying foil:
Issue: do we send to the data mulricast group, or to a special multicast group?

MB: discussion on advantages of using a special address for keys versus always using the data address.



(1) unicast
(2) separate group
(3) data group

thomas: do we say that RM is called to do this?
Ran: cannot place such a restriction.
Question: what is the purpose of rekey
HH: rekey, CR distinctions, accomodate dynamic membership
Ran: may need to develop our own reliability mechanism
Thomas: work through the Wallner example
Ran: could put the KM in every data packet
Amit: there are two issues, synchronization and reliability; solve the former with SPI or some tag and solve latter through RM protocol if available
Patrick: so how do we solve the need to reliably send keys?
HH: without RM we will do periodic retransmission, FEC, NAK approaches;

RM folks seem to agree that much of RM provides mechanisms that are not needed;
Thomas: A lightweight RM
Ran: Should it be done inside or outside of the data channel.
Mike: This is well suited to Luigi FEC
Patrick: Someone needs to come back with a report on RM solutions for re-keying
Questions by chairs:

- How many plan to attend a meeting in Australia IETF?

  about half of the people present (12 people) rais hands.

- Where shall we meet in January?
  agreed on West coast, end of january, either San Diego or Bay Area.
- How many plan to come?

  Again, about half the people raised hands.
