Folks,

Below is the final agenda for the next SMuG meeting in IBM Watson.

```
     Date:       6 October (Friday)
     Location:   IBM Watson, New York
                 Building: Hawthorne I
     Time:       9:30am - 5:00pm
     Directions: http://www.watson.ibm.com/watson/direct.html
```

```
SMuG Meeting -- October 6, 2000, IBM Watson.
-----------------------------------------

9:00 - 10:00   Presentations
                 - Ran/Thomas -- status of SMuG, BOF/WG, issues (20 min)
                 - R. Yang/utexas (20 min)
                 - Cathy Meadows (20 min)
                 - Others

10:00 - 12:00  Data Handling Building Block
                 - Progress report (Canetti)
                 - Specific Issues & Discussion

12:00 - 13:00  Lunch

13:00 - 15:00  GKM Building Block
                 - Progress report (Hardjono)
                 - Specific Issues & Discussion

15:00 - 15:30  Break

15:30 - 17:30  Policy Building Block
                 - Progress report (McDaniels/Harney)
                 - Specific Issues & Discussion
```

Please email me if there are any corrections to be made to
the agenda.

cheers,

Ran & Thomas

Summary
~~~~~~~

1. Agenda bashing - Ran Canetti, Thomas Hardjono

Thomas recommended that people who have not gone to secureMulticast.org visit the site. He also reported on the IETF BOF at the San Diego meeting: He asked for two hours, which puts us towards the back of the queue. WG thought we should request two one-hour sessions from the IESG.
~~~~~~~

2. A Lower Bound on the Communication Cost of Secure Group Key Management, Yang Richard Yang, University of Texas at Austin http://www.cs.utexas.edu/users/yangyang/research/TechReports

Keygem is a group key management system. The focus of our work is on performance and our design goal is to achieve highly scalable operation. Keygem's system components consists of Registration and Key Servers. The latter has Rekey encoding and Rekey transportation components. Rekey encoding takes joins and leaves and forwards this to rekey transportation in a way to optimize communication costs.

Previous work includes Key graph, LKH, OFT showed upper bound of O(ln(n)). Canetti et. al. proved a lower bound of c>=bn**(1/b) - b, where c is the leave communication costs and b+1 denotes maximum number of keys any member holds.

We consider first a system model that captures requirements and which is important to consider the lower bound. Backward and forward secrecy are among our security requirements that affect group performance and scalability when members join and leave. Thus there are 3 types of security requirements. Backward, forward, or both. Also, in our model, there is only one key server that implements forward secrecy and all users in group share a common group key gi. When updating keys the key server uses one KEK. We make the following assumptions: There is only one key server, the Key server implements forward secrecy, all users in the group share a common group key gi. When updating keys, the key server uses one key k to encrypt another key k', the adversary has access to all past communications. However, it can access one key k if it has received k from the key server, or k is encrypted by a key k' that it has access, etc.

In our model, the rekey ecryption graph captures all important communications features of rekey. Thus there is a sequence of directed graphs Gi, where Gi captures the communication costs of rekeying the first i requests. The nodes of Gi are either user nodes or key nodes (individual or normal); edge is a user node to its individual key node; a keynode k to key node k' if key server has send k' by encrypting it in k. The subgraph, Si, of Graph Gi: Si consists of user node u when u is in the group (join and not left) and all nodes on path from (leaf) node u to (root)node gi, the group key. When modeling leaves, the key graph has the property: Si subset Gi subset Gi+1. Si has the functions i(x), c(u), n(x) and s(x). i(x) is the indegree of node x. c(u) is the cost of user node u, which is the sum of I(x) from u to group (root) key. C is the maximum of all c(u). n(x) as the number of user nodes reachable to x. s(n) >= NlnN ==> C >= lnN. We use this to show that the wasted communication costs after a user leaves has per request communication cost of W(ln(n)).

Extending the systems model to allow several types of KEKs. Another extension is to add anti-collusion and still prove lower bound.

We have proven that amortized per request communication cost is LnN and that one needs to relax security requirements to further reduce communication costs. We are continuing work on keygem system to: consider communication costs as a property to evaluate transportation overhead; consider batch processing to further improve performance; to set the output of the encoding component to transport component block

size.

3. Formally Specifying Group ISAKMP, Catherine Meadows, NRL

We developed a formal specification and performed an analysis of GDOI using the NRL protocol analyzer (NPA) to evaluate this protocol early on in the standards process to gain maximum impact. We want to speed up acceptance of GDOI by using formal analysis to find and remove bugs and ambiguities and to provide evidence of soundness. Also want to learn more about strengths and limitations of the NPA.

NPA is a formal-methods tool for verifying security properties of crypto protocols and finding potential attacks. The user specifies the protocol in terms of communicating state machines. The user then proves a set of lemmas to limit size of search space; specifying an insecure state and use NPA to search backwards to see if successful attack can be found. To date the specification has been done but not the analysis.

The user must first supply an executable state machine specification, which is useful in identifying ambiguities, inconsistencies, etc.

GDOI uses a group key distribution center, and it uses an IKE phase 1 to distribute category 1 SAs. GDOI has a customized phase 2 to distribute category 2 SAs (may also distribute cat 3 SAs).

Phase 2 and Key Management datagram were shown.

The protocol starts with GCKS creating a group and a group key. After that, a group member may request to join the group by initiating a Phase 2 exchange; GCKS responds by completing protocol and may expel members by not issuing new KEKs to a member.

An example state transition shown.

Among its limitations, the NRL protocol analyzer was not built for analysis of data structures of arbitrary size. So we assumed each KEK is a single key. Assumed that no more than one SAT was sent in Phase 2. We further assumed one SAK or one SAT in key management datagram. And we did not include spec of IKE phase 1 (which is absolutely huge). GDOI may want to include information from other RFCs to make the document more self contained. Two substantial questions about KMD. First, the SIG payload definition was unclear over what fields were included in the signature and whether the signature was done first. We also found an ambiguity in the SEQ payload definition. We Found a redundancy between the use of SEQ and the sequence number field in the SAK.

Our conclusion is that the state machine spec is a helpful way to identify potential problems in GDOI early, which did not require any special properties of NPA or crypto protocol analysis. What seemed most helpful was concreteness and unambiguity of the NPA specification. Plan is to continue with spec and consultation with authors.

Discussion: Mark said that several problems were identified and removed through use of this tool. Ran asked about the use of KE_I and KE_R and PFS.

4. GDOI - Mark Baugher, PassEdge

Mark described the group management protocol that he has been working on with Brian Weis and Thomas Hardjono. GDOI 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 GDOI is intended to extend an IKE implementation to support secure groups.

The first SA in a GDOI GSA is created through a IKE Phase 1 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 LKH and One-Way Function Trees use a tree of KEKs 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 keyserver.

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.

Ran Canetti and Cathy Meadows identified a few issues that have been incorporated into the recently-posted GDOI draft, draft-irtf-smug-gdoi-00.txt. Ran's comments led to the replacement of the SIG with a POP (Proof of Possession) as the signature was applied to a nonce and not to the whole message. Cathy identified problems with SIG and SEQ definitions in addition to several smaller issues that the authors addressed in the draft.

We are looking forward to further review of the current draft by NRL. The authors are also considering how public-key encryption can be used to simplify the exchanges and plan to have two, interoperable implementations by 1Q2001.

5. Suggested changes to GDOI - Ran Canetti, Watson

Ran suggested to keep IKE as it is to set up an IPSEC tunnel. Then do a two message exchange between an entity above IKE. First message is <ID,POP[CERT]> and return message is <SA,KD>. Thus the GDOI would be extended to run over a security transport protocol such as IPSEC.

Advantages are that it is
- o more modular and flexible
- o use existing, tested, deployed IKE without change
- o Isolate the complexity of IKE from the complexity of MIKE (also work on other transports as well)
- o using TCP inside IPSEC tunnel provides reliability for free
- o easier experimentation and evolution
- o cost of ipsec tunnel is amortized over many communication events
- o separate MIKE control (which is complex and asymmetric) from IKE
- o one more round trip of communication

Ran's idea was to make the so-called GDOI Phase 2 more generic so that it can run over different security transports. There was a lot of discussion regarding the pro's and con's of this approach, which will change the GDOI focus which has been solely on extended IKE. Mark suggested that the authors consider this for the next revision of the draft and report back to the WG.

6. Group Policy - Hugh Harney, SPARTA

Hugh proposed that the SmuG Policy Building Block may use ASN.1 as it provides flexibility, standard notation. Also, ASN.1 is used by many PKIs. He noted that building blocks provide standard categories and interoperability switches, but building blocks do not try to solve all cases. Use of ASN.1 does not satisfy the requirements for all environments.

The Group Policy team has developed the following structure for capturing group policy information.

```
Group Token (choice)     tokenID
  GSAKMP                      version
  ANTIGONE                    protocol ID
  Other                       timestamp
                         authorizations
                              GroupOwner
                              gCKS (and subordinates)
                              rekey control
                         access control
                              permissions
                              access
                         mechanisms
                              dataComm
                              unicast
                              groupManagement
                         signature block
                              sigInfo
                              sigData
```

Example:
- tokenID version could be GSAKMPv1 or ANTIGONEv2
- authorizations would reference certs, public keys

mechanisms (cat1, cat2, cat3)

The rationale for this structure and an ASN.1 specification of it will be posted soon in a forthcoming draft.

7. MESP - Ran

Ran gave an overview of MESP and AMESP. MESP is an extension of ESP, and AMESP is the concept of AMESP at the application layer. Both accommodate use of a group source authentication algorithm, such as TESLA. If you are doing TESLA, you're probably going to put that on the outside (in app layer) and do the group secrecy and group auth at the network layer.

There was a discussion of putting the source authentication as a separate transform, but this was not ultimately desired.

The question was asked if the sender has to be dependant on the receiver operations? The answer was no, that the sender can determine whatever interval is reasonable (e..g, 300ms)

The question was asked if QoS is affected by the queuing of packets for TESLA? The answer was yes, very affected. To mitigate this, you can have multiple chains to deal with delays, where each chains are different intervals. But you have to predict the delays.

It was remarked that it would be nice to get rid of the time delays from TESLA. Panjab replied that Dan Bonae of Stanford determined that you have to have a signature. You need fast and effecient signatures. He's worked a combination of one-time signatures and ordinary signatures. But the trade off is that the signature blocks are very large.

8. Wrap Up - Thomas and Ran

SmuG WG participants should watch the list regarding the upcoming IETF BOF. We will decide over the mailing list whether SmuG will meet at the next IETF meeting in San Diego.

Attendees
~~~~~~~~~

Thomas Hardjono	harjono@nortelnetworks.com
Ran Canetti	canetti@watson.ibm.com
Pete Kruus	pkrruus@nai.com
Brian Weis	bew@cisco.com
Isidor Kouvelas	kouvelas@cisco.com
Cathy Meadows	meadows@itd.nrl.navy.mil
Mark Baugher	mbaugher@passedge.com
Patrick McDaniel	pdmcdan@eecs.umich.edu
Hugh Harney	hh@sparta.com
Pahkaj Rohatgi	rohatgi@watson.ibm.com
Yang Richard Yang	yangyang@cs.utexas.edu
Pau-Chen Cheng	pau@watson.ibm.com
~~~~~~~~~