Secure collaborations over message boards

Mahalingam Ramkumar*

Department of Computer Science and Engineering,
Mississippi State University,
MS, USA
E-mail: ramkumar@cse.msstate.edu
*Corresponding author

Nasir Memon

Department of Computer and Information Science,
Polytechnic University,
Brooklyn, NY, USA
E-mail: memon@poly.edu

Abstract: We provide a message board model for collaborative systems, and propose an architecture and protocol for securing collaborative applications over message boards. The proposed architecture employs only efficient symmetric cryptographic principles, and low complexity trust modules with each participant. The trust modules are used to protect group secrets and reduce susceptibility to denial of service attacks. We outline an architecture and elements of a protocol for Secure Collaborations over Message Boards (SCMB). The SCMB protocol can serve as a foundation over which a wide range of collaborative applications can be realised.

Keywords: secure collaboration; message board; key predistribution; trust modules.

Reference to this paper should be made as follows: Ramkumar, M. and Memon, N. (2006) 'Secure collaborations over message boards', *Int. J. Security and Networks*, Vol. 1, Nos. 1/2, pp.113–124.

Biographical notes: Mahalingam Ramkumar is an Assistant Professor in the Department of Computer Science and Engineering, Mississippi State University since August 2003. His primary area of research is security solutions under resource constraints.

Nasir Memon is a Professor in the Computer and Information Science Department at Polytechnic University, Brooklyn. His main research interests are in data compression, multimedia security, digital forensics and computer and network security.

1 Introduction

An ever increasing efficacy of 'anywhere/anytime connectivity to anything', is expected to have a significant impact on the effectiveness of collaboration between individuals and organisations, and therefore their productivity. While many such disparate collaborative systems are already in wide use today, like applications catering for connectivity of mobile workforces to corporate resources, blogs, wikis, interactive message boards, and even widely used systems like e-mail, instant messaging and chat, the scope of future collaborative systems will be significantly broader, catering for a wider variety of requirements.

Butler and Coleman (2003) identify five primary models of collaboration based on typical group sizes (the number of collaborators) and the level of interaction between the participants: library, solicitation, team, community and process support models. The library model includes the publish-subscribe systems (Eugster et al., 2000) which have garnered significant attention recently, and digital rights management systems. An example of the solicitation model is methods employed for obtaining feedback from consumers. Perhaps the community model (for example, internet forums) is the fastest growing model. Team models and process support models are primarily used by mobile workforces. One of the more complex collaborative processes in use today are some high profile open-source development projects.

While it may not be possible to have a *one-size fits all* solution for collaborative systems, there are still more similarities than differences between the different models. In all such models, the process of collaboration starts with the formation of interest groups, in response to a solicitation from a participant. All models would involve granting membership privileges by a group controller, and the ability to revoke membership privileges. In most cases the subject and object of exchanges will desire privacy. However, the nature of interactions within a group may be very different for different models, and even different applications that may fall under the same model.

As a general model for interactions between participants, we model collaborative systems as a *Message Board* (MB). The only way that participants interact is through the MB – by *reading-from* or *posting* messages. In general, each post on the MB may consist of the entire message, or a pointer – say a Uniform Source Locator (URL) of some content. The participants employ the MB in order to:

- advertise creation of interest groups
- seek memberships in interest groups
- grant/revoke memberships
- perform intra-group interactions and
- send unsolicited messages.

Farley (1998) considered shared whiteboard models for collaboration, and Murayama et al. (2001), message board systems for very specific purposes. Architectures for complex collaborative systems using SVG[1] based on an event-brokering system, catering for access by a wide range of devices has been discussed by Lee et al. (2002). As event-brokering systems can be realised using message passing interfaces, even complex interactions are possible over MBs.

The main contribution of this paper is an architecture and protocol for interactions over MBs, to facilitate secure collaborative applications. The architecture employs light-weight Trust Modules (TM) that can be plugged into end-users computers like desktop/laptop/PDAs. Henceforth, we shall assume that the end-users device is a PDA.

Every end user (or participant) is issued two IDs – a public ID and a pseudo-ID, and various secrets corresponding to the IDs, some of which are protected by the TM. The participants employ untrusted software running on their PDAs to perform interactions over the MB. The job of the low-complexity TMs is restricted to symmetric cryptographic operations. The proposed *Secure Collaboration over Message Boards* (SCMB) architecture and protocol is generic enough to cater for a wide range of applications like DRM, publish-subscribe systems, e-mail and instant messaging.

The rest of this paper is organised as follows. In Section 2, we provide an overview of SCMB and a justification for the proposed approach for the realisation of collaborative applications. In Section 3, we discuss various key distribution schemes employed by SCMB. Section 4 is a discussion of the architecture and the mechanism of deployment of SCMB. In Section 5, we discuss the use of various security primitives for securing interactions within message boards. Discussions and conclusions are offered in Section 6.

2 SCMB overview, rationale and goals

The SCMB system consists of two independent ID Issuing Authorities (IA), I_U and I_P – who issue public and pseudo-IDs respectively, to all participants. For example, a participant Alice may be issued a public ID A by I_U and a pseudo-ID A' by I_P. However, it is ensured that no one in the SCMB system, including the IAs, can determine that A and A' are the IDs of the same participant. Furthermore *even Alice's TM is not privy to her pseudo-ID A'*, even though her TM is entrusted with the task of protecting some of the *secrets corresponding to her ID A'*, from Alice. Each participant armed with a trust module, can receive SCMB secrets corresponding to both IDs, which will be used for securing interactions over MBs.

The SCMB can support any number of MBs. Each MB is hosted by some service provider.[2] *Gateways* to each MB control write-access to their MBs. Participants with TMs plugged into PDAs interact with the gateways, using untrusted software that runs on such devices. More specifically, we assume that while the end-user has full control over the software that runs on the PDA, the 'SCMB system' and the end-user's TM do not trust the software.

An SCMB participant is eligible to subscribe to any MB. While only subscribers of the MB will be able to provide the authentication information (verifiable by the gateway) before the messages are posted, reading from the MB is open[3] to all. The only way for any two subscribers of the MB to communicate is by posting a message on the board and reading from the board. However, we assume that it may not be possible in general, for any one to 'observe' others during the actual process of posting a message. In practice this would imply that the system would accept packets routed through anonymising networks (Moskowitz et al., 2003). Each post on the MB is timestamped by the gateway.

Within any MB, any subscriber can create any number of interest groups, and any subscriber of an MB can seek memberships in such interest groups. In order to clarify the terminology used in the rest of the paper, note that an *end-user* can be:

1 *participant* in the *SCMB system*
2 *subscriber* of an MB in the SCMB system and
3 *member* or *controller* of *interest groups* within an MB.

Any post could take the form of:

1 broadcasts intended for all subscribers of the MB
2 multicast messages intended for members of a group or
3 unicast messages to specific subscribers.

Furthermore, posts could be solicited or unsolicited. In all cases anonymity of sender and receiver of interactions would be desired with respect to *other* participants[4] in the system. In some cases, even senders and receivers may desire to protect their identities from each other.

2.1 Rationale for the proposed approach

Tolone et al. (2000), Bullock (1998) and Moody and Bacon (2001) have considered Role-Based Access Control (RBAC) in collaborative applications. Opyrchal and Prakash (2001), Wang et al. (2002), Khurana (2005) and Frege et al. (2004) have investigated a variety of security issues in publish-subscribe interaction models. However, while privacy issues are given consideration in the latter (pub-sub), they have not received much consideration in the former (RBAC), and vice versa, perhaps due to the apparently conflicting paradigms of anonymity and access control

(after all, why should anyone cede control to someone unknown?). However, as access control is still possible under looser notions of anonymity like pseudo-anonymity (Hughes and Shmatilov, 2004), SCMB caters for both requirements.

Due to their very nature, resources employed by collaborative systems will need to be accessed from various locations, using various platforms, possibly by a very large number of individuals and computers. Thus apart from inheriting the security issues faced by todays large scale communication infrastructures based on client-server paradigms, many other new issues are introduced in collaborative systems, due to *shared group secrets*. Collaborative systems will also need to be more resilient to attacks, as strategies for mirroring the services provided by such systems can be complex. Furthermore, the downtime of collaborative systems can have a more severe impact.

2.1.1 Trust modules

Due to ever increasing security concerns, it is widely believed[5] (Grawrock, 2006) that solutions based on trust worthy computing modules (TM) will eventually be necessary even for securing more conventional client–server interactions. Furthemore, for any collaborative activity, 'providing assurances' also implies ensuring that users who are assigned some privileges, in the form of group secrets, cannot abuse their privileges, for example, by revealing the group secret(s) to unauthorised entities. Thus the need for TMs that can *protect and use* the group secrets on behalf of the user, is more acute in collaborative applications.

However, in order for solutions based on TMs to have practical acceptance, it should be possible to:

- make them inexpensive and
- provide verifiable assurances to end-users, that such modules cannot violate their privacy – say by surreptitiously sending private information to undisclosed entities.

Providing assurances of trustworthiness, entails effective *shielding* of components from intrusions aimed at *modifying software* or *exposing secrets*. The unfortunate side effect of 'effective shielding' is reduced ability of the devices to dissipate heat. Thus solutions that simultaneouly cater for both (effective shielding and heat dissipation) tend to be expensive (Smith and Weingart, 1998).

Thus by limiting the use of TMs in SCMB to efficient symmetric cryptographic primitives (a few symmetric block cipher operations using a hardware block cipher), and the scope of 'other' tasks to be performed by the TM to be small enough to be handled even by a processor equivalent in capabilities to the microprocessors of the early eighties, we can eliminate the need for proactive measures for heat dissipation. This can simultaneously cater for reduced cost *and* improved trustworthiness of TMs. Furthermore, limiting the *scope* of tasks performed by the TM, can be synergistically employed to provide end-users with the comfort, that TMs (even though they may execute software which are not under the control of the end-user), cannot violate their privacy.

2.1.2 Key distribution schemes

Restricting SCMB to employ very efficient symmetric cryptographic primitives can also reduce susceptibility of SCMB to denial of service attacks. The SCMB makes generous use of an elegant and efficient key distribution scheme, LM-KDS, proposed by Leighton and Micali (1994) for mutual authentication of SCMB participants.

The ability of any participant to organise groups and efficiently control memberships to such groups also calls for efficient *Broadcast Encryption* (BE) schemes. Specifically, such BE schemes should permit sources other than the Key Distribution Center (KDC) to broadcast secrets (as any participant can be a group controller). The SCMB employs a novel probabilistic key predistribution scheme, Asymmetric[6] Random Preloaded Subsets (A-RPS), for BE.

While most BE schemes proposed in the literature can be extended to cater for broadcasts by multiple sources by using asymmetric cryptographic primitives, the A-RPS scheme employed by SCMB caters for this requirement without the need for asymmetric primitives. In addition, we shall see that A-RPS has many desirable properties that make it especially well suited for its use in SCMB.

The SCMB makes very limited use of asymmetric cryptographic primitives. Specifically their use is limited to the deployment phase, and periodically for signing of revocation lists. However, all asymmetric cryptographic operations are performed by the end-user devices (PDA/laptop) – not the TMs.

2.2 SCMB goals

Some of the specific goals of the SCMB are thus:

1 employ only symmetric cryptographic primitives for day-to-day operation, to reduce susceptibility to denial of service attacks

2 provide any subscriber of the message board the ability to organise interest groups, and grant/revoke memberships to other subscribers

3 cater for anonymity of sender and receiver from other participants in the system

4 cater for anonymity of sender and receiver from each other if desired

5 cater for verifiable assurances that correspondences between public and pseudo-IDs of participants cannot be determined by any entity in the SCMB system, *including the end-users TM*

6 restrict TMs to employ only symmetric cryptographic primitives

7 ensure that no packet that *leaves* the end-users device, will be encrypted with a secret that the end-user does not have access to and

8 that the participants will *not* need to employ TMs for their day-to-day interactions.

There are however some exceptions to goals 7 and 8. Under hostile conditions, the MB operators may mandate that the posts include a message authentication code with a secret that is privy only to the TMs. Similarly, depending on the

sensitivity of the collaborative activity, group controllers will be able to enforce policies for their group members concerning the need to employ TMs. Hiding pseudo IDs from TMs is crucial in order to provide assurances to the end user that the TM cannot reveal his/her pseudo ID in such encrypted messages.

3 Key distribution schemes for SCMB

A KDS is a mechanism for distributing secrets to all nodes to facilitate establishment of cryptographic bonds or Security Associations (SA) between the nodes. Such SAs can be one-to-one (e.g. mutual authentication using pairwise secrets), one-to-many (e.g. broadcast authentication) or group security associations (through instantaneous conference secrets or non-instantaneous broadcast encryption). The SCMB employs a simple and novel variant of the LM-KDS for mutual authentication of participants and a novel probabilistic key predistribution scheme, A-RPS, for secure conveyance of group secrets.

3.1 Leighton–Micali KDS

The LM-KDS, based on a master key and a cryptographic hash function $h()$, consists of a KDC and a set of N nodes with unique IDs. The KDC chooses a master key K. Node A (or node with ID A) is provided with the secret $K_A = h(K \parallel A)$. For establishing a session secret K_S with node B (which has the secret $K_B = h(K \parallel B)$), node A performs a look up in a *public* repository, created by the KDC, with $\binom{N}{2}$ entries, for a public value $\Pi_{AB} = h(K_B \parallel A) \oplus h(K_A \parallel B)$, and calculates $K_{AB} = \Pi_{AB} \oplus h(K_A \parallel B) = h(K_B \parallel A)$. Node B can however directly evaluate K_{AB} using its secret K_B. The session secret K_S is now encrypted using K_{AB}. However, for large networks it may not be feasible to maintain a public repository with $\binom{N}{2}$ public values. So the KDC may actually need to be on-line to calculate public values of the form Π_{ij} and provide it to the nodes "on demand".

The main difference between LM-KDS and schemes based on the symmetric Needham and Schroeder (1978) protocol (like Kerberos, which also require a trusted on-line server), is that the values Π_{ij} that a node receives from the server is *not* a secret. Thus nodes do not need to authenticate themselves to the server to receive Π_{ij}s. Further, the KDC is not required to be on-line for *every* communication attempt between i and j – nodes need to access the KDC *only once* (for i to authenticate itself to j for ever in the future). It is also possible for node A to get Π_{Aj}s for a large number of js that node A may desire to communicate with in the future in a single attempt. The LM-KDS can be easily be extended to using multiple master keys - say t such systems used together, with master keys $K^1, \ldots, K^t$. The authentication secret K_{ij} in this case will be $K_{ij} = K_{ij}^1 \oplus K_{ij}^2 \oplus \cdots \oplus K_{ij}^t$.

3.1.1 Privacy protection in LM-KDS

For purposes of mutual authentication, Kerberos like approaches have two disadvantages:

1 the need for a trusted online server and

2 need for active mediation by the server.

In a highly connected world, the first issue is not a serious limitation. The second issue however has implications on the *privacy* of interactions – after all A and B may not wish that the KDC comes to know that A and B interact.

While Leighton and Micali (1994) point out several advantages of LM-KDS over Kerberos, a very important one for our purposes is that LM-KDS does not call for active *mediation* by the trusted server. For the ability of node A to authenticate itself to node B, all that is required for node A is to receive public values from the server, for which as mentioned earlier, A does not need to authenticate itself to the server.

In the SCMB, LM-KDS is used for mutual authentication of participants. However, even though the KDC issues secrets to every participant, and is always available online for providing public values on demand, we wish to ensure that the KDC does not gain any knowledge of the actual identities of the interacting participants. This is achieved by ensuring that for a participant with ID A, the LM-KDS KDC provides secrets corresponding to an ID $A_1 = h(A)$. Specifically, the LM-KDS KDC is not made aware of the actual ID A of the participant.[7] Thus in a scenario where A desires to authenticate itself to B, A can safely obtain the public value $\Pi_{A_1B_1}$ where $B_1 = h(B)$. However as A and B know each other's IDs, the secret $K_{A_1B_1}$ is sufficient for mutual authentication of A and B.

3.2 Managing group secrets with A-RPS

BE (Fiat and Noar, 1994) provides a means of establishing a shared secret between g privileged nodes, out of a set of G nodes, where $g + r = G$, and the r nodes which are *not* provided with the secret are usually referred to as 'revoked' nodes. Specifically, broadcast encryption deals with cases where $g \approx G$ or $r << g < G$.

In the SCMB, BE is realised using a Probabilistic Key Predistribution Scheme (PKPS). Most PKPSs exploit the property of uniqueness of intersections of large subsets. While the earlier of such techniques Gong and Wheeler (1990) and Mitchell and Piper (1995) relied on deterministic strategies for allocation of subsets of keys to every node, Dyer et al. (1995) were the first to point out the simplicity and effectiveness of *random allocation of subsets*.

Ramkumar et al. (2003) define Random Preloaded Subsets (RPS) by two parameters m and k. The KDC chooses an indexed set of secrets $\mathbb{S} = \{K_1, K_2, \ldots, K_m\}$. Every node in the network, is assigned a subset of $k = \xi m$ secrets (or $\xi < 1$). Two nodes will share on an average, $\xi k = \xi^2 m$ secrets. Mutual authentication of two nodes A and B is achieved by deriving a shared secret K_{AB} based on all ξk secrets they share.

Canetti et al. (1999) discussed several source authentication schemes based on random subset allocation. In their basic scheme (where the source is the KDC), to authenticate a message M, the source appends m key based Message Authentication Codes (MAC) – one corresponding to each of the m secrets in $\mathbb{S}$. Any verifier can verify k of the m appended MACs. Canetti et al. (1999) also proposed an elegant extension of their basic scheme to cater for broadcast by external (who are not provided with any of the KPS secrets) sources. For example, such an external source W obtains

m values $\mathfrak{S}_W = \{K_i^W = h(K_i \parallel W)\}, 1 \leq i \leq m$ from the KDC. Note that K_i^W does not provide any information about K_i as long as the hash function $h()$ is secure. Now all broadcasts by W are authenticated with m MACs using the secrets $\mathfrak{S}_W$ – which nodes with subsets of keys from $\mathbb{S}$ can still verify.

3.2.1 Asymmetric RPS

The broadcast *encryption* scheme used by the SCMB, which we refer to as A-RPS, is very similar to Canetti et al. (1990) for broadcast *authentication*. The (m, k) A-RPS scheme employs a simple one way function $F()$, and a cryptographic hash function $h()$. For a node A, $F(A) = \{A_1, A_2, \ldots, A_k\}$ determines the indexes assigned to node A. Now node A is assigned k decryption secrets $\mathbb{S}_A$, and additionally, m encryption secrets $\mathfrak{S}_A$, where

$$\mathbb{S}_A = \{K_{A_1}, K_{A_2}, \ldots, K_{A_k}\} \quad (1)$$
$$\mathfrak{S}_A = \{K_j^A = h(K_j \parallel A)\}, 1 \leq j \leq m$$

In (m, k) A-RPS, where the sender has m encryption secrets and every node has $k = \xi m$ decryption secrets, let us denote by $\mathbb{I}_A$ the set of k indexes between 1 and m (corresponding to which decryption secrets are assigned to A). For BE using A-RPS the sender determines the union $\mathbb{I}^r$ of all such indexes assigned to r nodes (say $R_1, \ldots, R_r$) to be revoked. In other words, $\mathbb{I}^r = \{\mathbb{I}_{R_1} \cup \mathbb{I}_{R_2} \cup \ldots \cup \mathbb{I}_{R_r}\}$. As the source can use encryption secrets with indexes from the set $\{1, 2, \ldots, m\}$, secrets corresponding to indexes $\{1, 2, \ldots, m\} \setminus \mathbb{I}^r$ can be used 'safely' for encrypting the broadcast secret K_b. Apart from various encryptions of the broadcast secret K_b, the broadcast message will also have a header which *indicates the indexes* (between 1 and m) of the secrets used for encrypting K_b.

The performance of BE using A-RPS is identical to that of the BE scheme in Ramkumar (2005), using RPS, where the source is the KDC. While Ramkumar (2005) also investigates performance of BE using various PKPSs, for both broadcasts by KDC and broadcasts by peer nodes, the BE by peer nodes use a different technique which is far less efficient that broadcasts by KDC. However BE using A-RPS, by untrusted sources, has the same efficiency as the scheme for BE by the KDC in Ramkumar (2005).

The *expected* number of indexes in $\{1, 2, \ldots, m\} \setminus \mathbb{I}^r$ is $\bar{q}_e = m(1 - \xi)^r$. In the event (for a given set of r nodes to be revoked) $q_e \approx \bar{q}_e$ is the number of such indexes, the source can use a subset $q \leq q_e$ of the indexes for encrypting K_b – it does not necessarily have to use all secrets it can. The probability p_o that a particular node (that does not belong to the set of r revoked nodes) cannot decrypt any of the q encryptions is $p_o = (1 - \xi)^q$. The total number of encryptions n_e of the broadcast secret required is $n_e = q + gp_o = q + g(1 - \xi)^q$, where the second term, gp_o is the number of nodes for which the broadcast secret may need to be unicast individually. In practice, for small p_o the unicast transmissions may be rarely called for. Apart from the n_e encryptions of K_b, the broadcast message will also have a header which *indicates the indexes* of the n_e secrets used for encrypting K_b.

Generally, the efficiency of BE schemes is measured in terms of the number of encryptions of the broadcast secret n_e required for conveying the group secret for revoking r nodes. For most BE schemes $n_e = \mathbb{O}(r \log(N))$ where N is the network size (number of unique IDs). However some efficient schemes like Noar et al. (2001) which require $n_e/r \approx 2$ have also been proposed. For BE using A-RPS, for small group sizes (less than a million) n_e/r may even be less than one, and about 5 to 6 for large group sizes (billions).

However, the efficiency of A-RPS for any r depends on the value $\xi = k/m$. In general, we need smaller ξ for larger r. Thus a practical solution is to employ several A-RPS schemes with different values of ξ in parallel. Thus we could use say l schemes with various values of m, say $m_1, \ldots, m_l$ and various values k, say $k_1, \ldots, k_l$ (for different values of $\xi_i = k_i/m_i$).

Thus each node will be provided with $\sum_{i=1}^{l} k_i$ decryption secrets and $\sum_{i=1}^{l} m_i$ encryption secrets. In other words, the KDC chooses secrets $\mathbb{S}^1, \cdots, \mathbb{S}^l$ and node A receives authentication secrets $\mathfrak{S}_A^1, \ldots, \mathfrak{S}_A^l$ and verification secrets $\mathbb{S}_A^1, \ldots, \mathbb{S}_A^l$.

The more well-known tree based broadcast encryption schemes in Noar et al. (2001) and Halevy and Shamir (2002) assume that the broadcast is performed by the 'root of the tree' – or only by the KDC who distributes secrets. While they can be extended to support broadcasts by peers, this would need the use of asymmetric cryptographic primitives (Anzai et al., 1999). A-RPS caters for broadcast encryption by any source (with access to the authentication secrets) without the use of asymmetric cryptography. In addition, while most broadcast encryption schemes require that the broadcast *explicitly identify* the list of non-privileged (or revoked) nodes, broadcast encryption using A-RPS permits *concealment of the identities* of the revoked nodes. This is achieved by providing the list of *indexes of the keys used*, for encrypting the broadcast secret K_b. This feature is indeed useful in scenarios where privacy is a crucial issue.

Unlike tree based schemes however, BE using A-RPS places some constraints on the total number of nodes that can be revoked efficiently. Specifically, for tree based schemes the efficiency of the BE scheme (measured in terms of number of encryptions of the broadcast secret needed per revoked nodes) primarily depends on the total network size. For A-RPS however the efficiency will depend on the group size g (or the desired p_o as $p_o\mathbb{O}(1/g)$. Thus in the SCMB where group controllers may control small groups (even though the possible network size – or the limit on the total number of possible participants in the SCMB) may be practically unlimited,[8] A-RPS can be significantly more efficient than tree based schemes. Furthermore, in situations where the *number of secrets* each node needs to *store* is not a serious limitation (as we shall argue is indeed the case for its use in SCMB), A-RPS can be even more efficient.

3.2.2 Shared secrets with A-RPS

Apart from BE, A-RPS can also be used for establishing a shared secret between any two nodes to facilitate mutual authentication. For instance, in a (m, k) A-RPS, for conveying a secret K to B, the source A determines the k indexes $\mathbb{I}_B$ of B's decryption secrets. The secret K is encrypted using the corresponding k encryption secrets of A.

For a choice of $m = (n + 1)k$, the A-RPS scheme (when used for establishing shared secrets) can resist compromise of n nodes with a probability p where $k = e \log(1/p)$ (see Canetti et al., 1999). In other words an attacker who exposes all secrets from n nodes can compromise all k secrets of a fraction p of all nodes. For example, if $k = 256$, $p \approx 10^{-40}$. For $n = 2^{10}$ the total number of encryption keys will be less than 300,000 (about 5 MB of storage for 128 bit keys). Note that for BE using A-RPS the SCMB will employ several A-RPS systems in parallel with different values of m/k. A-RPS systems with large m/k, apart from catering for efficient revocation for large r will also be used for establishing shared secrets.

While the SCMB does not use A-RPS for mutual authentication of participants (LM-KDS is used for this purpose), we shall see that the LM-KDS secrets assigned to nodes for this purpose will not be protected from the users (while A-RPS decryption secrets are). Thus LM-KDS cannot be used for unicasting group secrets (which have to be protected from the users). Thus shared secrets using A-RPS is used for this purpose.

3.3 A-RPS with TMs

The A-RPS decryption secrets are assigned to TMs. All secure computations involving decryption secrets will need to be performed inside the TMs. However the secrets themselves can be stored outside the TM – encrypted with a single highly protected[9] secret – a master secret K_M – stored inside the TM. Thus the decryption secrets assigned to a TM could be stored encrypted in the desktop/laptop/PDA that the TM is plugged into.

The encryption secrets are however of no concern to the TM. They can be protected by the end-user using any means suitable. Even a million 128-bit secrets requires only 16 MB of storage – a trivial requirement even for PDAs. Thus the number of decryption/encryption secrets for A-RPS is not really an issue. For BE using A-RPS (and RPS), efficient operation for any r can be catered for if we increase the number of verification secrets (and authentication secrets) assigned to any device.

In the process of *encryption* of group secrets using A-RPS the TMs have no role to play. At the other end, the end-users PDA determines which of the various encryptions of the broadcast secret can be decrypted by the TM (as the BE messages indicate the indexes of the secrets used for encrypting the group secret). For example, assume that one of the encryptions of the broadcast secret is an A-RPS secret corresponding to index i (or secret K_i), that has been assigned to the TM. In other words, the broadcast includes[10] $K_i(K_b)$. The secret K_i is of course stored encrypted outside the TM – or the PDA has access to $K_M(K_i)$, where K_M is the master secret protected by the TM. The PDA can now provide $K_i(K_b)$ and $K_M(K_i)$ to the TM, which can evaluate K_b. The TM will however not reveal the broadcast secret K_b to the PDA. The secret K_b is encrypted with the master key K_M, and $K_M(K_b)$ is handed back to the PDA for storage.

Later, for all group messages encrypted with some session secret K_s (which is in turn encrypted with the group secret as $K_b(K_s)$), the PDA supplies the TM with $K_b(K_s)$ and $K_M(K_b)$, and the TM returns the session secret[11] K_s to the external device. Note that the group secret is *not* protected from the group controller (who chooses the secret and encrypts it with authentication secrets). It is however protected from group members – only their TM is privy to the secret.

Similarly, when the group secret is unicast to a group member (in which case the group secret may be encrypted with k A-RPS secrets) the PDA provides the TM with the k encrypted secrets. It is important to note that the *TM does not even have to know the ID* for which it holds the $\sum_{i=1}^{l} k_i$ A-RPS decryption secrets. Alice's PDA executes the public function $F()$ of A-RPS (which determines the *indexes* of the secrets used for any SA). The secrets themselves are however only privy to the TMs.

4 SCMB architecture

The SCMB system consists of ID issuing authorities (IA), KDCs (for LM-KDS and A-RPS), message boards with gateways, and participants with trust modules.

Central to the SCMB are the two independent IAs, I_U and I_P. The IA I_U issues public IDs (and corresponding secrets) to participants (end users with TMs). The IA I_P issues pseudo-IDs, and corresponding secrets, to the participants. However, no entity in the SCMB, including the IAs, are privy to links between public and pseudo-IDs. In other words, while Alice may be assigned a public ID A and private ID A' (and some secrets corresponding to both IDs stored in the same TM), no one apart from Alice knows that A and A' are IDs of the same participant.

The IAs I_U and I_P share a secret S_I. After the shared secret is established all communications between the two IAs are cut off.

The LM-KDS KDC T_M (with master secret M), caters for mutual authentication of participants. For a participant with public ID A and pseudo-ID A', the KDC T_M issues secrets corresponding to IDs $A_1 = h(A)$, viz., $M_{A_1} = h(M \parallel A_1)$, and $A'_1 = h(A')$, viz., $M_{A'_1} = h(M \parallel A'_1)$. The KDC T_M however does not have any knowledge of the IDs A or A' (and does not even know that A_1 corresponds to a public ID and A'_1 to a pseudo-ID). The KDC T_M shares a secret S_M with both IAs.

The KDC T_B issues A-RPS encryption and decryption secrets to every participant (corresponding to public and pseudo-IDs of participants). These secrets are used by participants within a message board to control and manage access to group secrets.

Any LM-KDS or A-RPS KDC can be trivially split into multiple independent escrow. However, for keeping the discussion simple, we shall assume single KDC.

The SCMB includes some special participants – the gateways to each MB. While the gateways are also issued public and pseudo IDs their pseudo IDs are also made public. To keep the discussion simple, we shall assume that the SCMB system has only one MB. The gateway to the MB is responsible for providing write access to the MB.

The basic steps involved in inducting a participant in the SCMB consists of:

- S1 Assigning a public ID (by I_U).
- S2 Providing secrets corresponding to public ID, for mutual authentication (by KDC T_M) and broadcast encryption (by KDC T_B).

S3 Subscribing to the MB (using public ID).

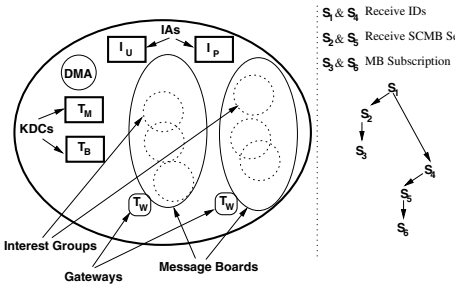
S4 Assigning a pseudo-ID (by I_P).

S5 Providing secrets corresponding to pseudo-ID for mutual authentication and broadcast encryption.

S6 Subscribing to the MB (using pseudo-ID).

The steps involved are (ideally) performed in the sequence S1 to S6 (for reasons that will be explained in the end of this section), as depicted in Figure 1 (right). However, in the following narrative, we follow the sequence S1, S4, S2, S5, S3, S6.

Figure 1 SCMB components and induction of participants



4.1 Assigning public and pseudo IDs (S1, S4)

The IA I_U chooses a master key U. In addition, I_U generates a secure RSA public-private key pair (public key (e_U, n_U), and private key d_U). The public keys (e_U, n_U) of T_U are well advertised. The IA I_P chooses master secret P.

The IA I_U assigns public IDs to all participants. Every participant in the system is assigned a unique public ID - as a one way function of a descriptive string. For instance, Alice, described by a string S_A = 'Alice B. Cryptographer, Anytown, USA' is assigned a public ID $A = h(S_A)$. Similarly a gateway operated by Google is assigned a public ID G corresponding to a string 'Google SCMB Gateway'.

4.1.1 Step 1

Corresponding to the public ID assigned to Alice, Alice's TM[12] A is provided with a secret $U_A = h(U \parallel A)$. Alice can now take physical possession of her TM A. From this point onwards, Alice can authenticate herself (using her TM) as A, over insecure channels to I_U using the secret U_A.

Preparations for Step 4: after obtaining her public ID A, Alice is eligible to obtain a pseudo-ID A' from I_P. However, we wish to ensure that *no one* in the SCMB system, including the two IAs, can determine the correspondence between A and A'. For this purpose, Alice should obtain a certificate from I_U which in effect states that the TM A has been assigned some public ID, however without disclosing the actual ID A. In addition Alice should be able to *verify* that the values provided to Alice by I_A (which will be submitted to I_P for receiving her pseudo-ID) do not reveal any information about her public ID.

Furthermore Alice should not be able to get more than one pseudo-ID. For this purpose, the certificate has to include a *unique* nonce signed by the KDC I_U (so that I_P can verify that an ID has already been assigned for some nonce). However, Alice would like to confirm that the nonce signed by I_U cannot reveal any information about her public ID A. In particular, as I_U and I_P share a secret S_I, it should not be possible for I_U to have the flexibility to choose the nonce. It is also not desirable[13] for Alice to have total freedom in choosing the nonce.

Alice establishes a secure channel by generating RSA key pairs, and requesting her TM to encrypt her public key using the secret U_A. Now Alice chooses a random nonce N_0, and I_U chooses a random nonce N_t. The interactions between Alice and I_U, over the secure channel, are as follows:

$$\begin{array}{lll} \text{Alice} \rightarrow I_U: & N_1 = h(N_0) \\ I_U \rightarrow \text{Alice}: & N_t, N_A, [N_A]_U, U_A(S_{IA}) \end{array} \tag{2}$$

where

$$\begin{array}{ll} N_{A0} = h(N_1 \parallel N_t) & N_A = h(N_{A0} \parallel N_1) \\ [N_A]_U = (N_A)^{d_U} \bmod n_U & S_{IA} = h(S_I \parallel N_A) \end{array} \tag{3}$$

Now Alice (who has access to the RSA public keys of I_U) can convince herself that N_A has indeed been derived from values that I_U does not have full control over, and thus cannot possibly reveal any information about her public ID A. The value $U_A(S_{IA})$ is handed over to her TM, which can determine S_{IA}.

4.1.2 Step 4

To obtain her pseudo-ID, Alice's TM generates $A_S = S_{IA}(N_0)$, and Alice submits

$$M_{AP} = [N_A \parallel [N_A]_U \parallel N_{A0} \parallel A_S] \tag{4}$$

to I_P. Note that:

1 as I_U has freedom in choosing N_t, N_t cannot be disclosed to I_P

2 disclosing N_0 (and hence $N_1 = h(N_0)$) provides an assurance to I_P that Alice did not even have total freedom to choose N_1

3 obviously, neither Alice nor I_U have freedom in choosing N_A

4 the signature of I_U for the value N_A is an indication that Alice (or as far as I_P is concerned the person submitting the four values) has been assigned a public ID.

Note that N_t, N_0, N_A and $[N_A]_U$ are revealed by the TM to Alice (in order for Alice to verify that none of the values she submits to I_P can possibly reveal her identity A). However the secret S_{IA} is still protected from Alice. As N_0 submitted by Alice's TM will be encrypted with S_{IA}, Alice still cannot be sure that the *her TM does not disclose her identity A* to I_P. Alice can however verify that A_S has the same number of bits as N_0, and in addition, the response by I_P will indicate that I_P did indeed receive N_0 (for example, by revealing $N_1 = h(N_0)$) in the clear.

Alice can now be assigned a random pseudo-ID A'. The IA I_P can store N_A to ensure that N_A cannot be re-used by Alice for receiving more than one pseudo-ID. However, for very large network sizes this may be cumbersome. This A' could just be issued as a fixed secret function of N_A, for example $A' = h(N_A \parallel N_0 \parallel X)$ where X is a fixed random quantity

known only to I_P. Thus even if Alice repeats the process all over again, she will still be issued the same pseudo-ID.

Alice (using her PDA) generates a RSA public–private keys. The TM authenticates her public keys by encrypting it with $S_{IA} = h(S_I \parallel N_A)$. Along with M_{AP}, Alice also sends here public keys to T_P. The response by T_P consists of $S_{IA}(P_{A'})$ (where $P_{A'} = h(P \parallel A')$), and A', both encrypted using Alice's public keys. Thus while Alice cannot get access to the secret $P_{A'}$ (only her TM can), the TM is *not* provided with Alice's pseudo-ID A'.

4.2 Receiving SCMB secrets (S2, S5)

In Step S1, I_U also provides Alice with a secret $S_{MA_1} = h(S_M \parallel h(A))$ (where S_M is as secret shared by I_U, I_P and T_M). Similarly, in Step S4 (when Alice receives here pseudo-ID A'), I_P provides Alice with a secret $S_{MA'_1} = h(S_M \parallel h(A'))$. The secrets S_{MA_1} and $S_{MA'_1}$ permit Alice to authenticate herself as $A_1 = h(A)$ and $A'_1 = h(A')$ respectively, to the KDC T_M. The KDC T_M (of a LM-KDS system M with master secret M). Thus Alice receives secrets[14] $M_{A_1} = h(M \parallel A_1)$ (in Step S2) and $M_{A'_1} = h(M \parallel A'_1)$ (in Step S5) from T_M.

Note that T_M has no knowledge of the preimages A and A' respectively. Furthermore, T_M does not know (or care) if A_1 or A'_1 correspond to public IDs or pseudo-IDs. Anyone, without need for authentication, can query the KDC T_M at any time to receive public values of the form $P_{XY} = h(M_X \parallel Y) \oplus h(M_Y \parallel X)$ for any (and any number of) X and Y. However, for mutual authentication of two nodes, say A and C, the secret $K_{A_1C_1} = h(M_{A_1} \parallel C_1)$ can be used as both A and C can derive A_1 and C_1.

The involvement of the IAs I_U and I_P stops with assigning IDs and the corresponding secrets (like U_A and $P_{A'}$) to the participants. As mentioned earlier, SCMB includes some special participants – for whom the pseudo-IDs are also made public:

1 *Gateways*: Gateway G is assigned public ID G (and a secret $U_G = h(U \parallel G)$) and pseudo ID G' (and secret $P_{G'} = h(P \parallel G')$).
2 PKPS KDC T_B, the root of a A-RPS PKPS B used for broadcast encryption. T_B is assigned public ID B and secret $U_B = h(U \parallel B)$, and pseudo ID B' and secret $P_{B'} = h(P \parallel B')$.
3 A deployment monitoring authority (DMA) with a special ID D and D', who is provided with secrets U_D and $P_{D'}$ and *authentication secrets* from B (for broadcast encryption) corresponding to IDs D and D'.

Apart from the secrets provided by the IAs to each participant, the regular participants (end users) are also provided with public values necessary for authenticating themselves with the special participants. For example, TM A is provided with the public value $\Pi^U_{AB} = h(U_A \parallel B) \oplus h(U_B \parallel A)$ for mutual authentication of B (the KDC T_B) and A, and $\Pi^P_{A'B'} = h(P_{A'} \parallel B') \oplus h(P_{B'} \parallel A')$ for mutual authentication of A' and B' (and similarly, public values for authentication of participants with gateways).

Armed with U_A for her ID A, Alice can now authenticate herself to T_B. Thus in Step S2, Alice's TM receives system $\mathcal{B}$ secrets $\mathcal{B}_A$ corresponding to ID A. Specifically, $\mathcal{B}_A$ consists of:

1 $\sum_{i=1}^{l} m_i$ encryption secrets $\mathfrak{S}^1_A, \ldots, \mathfrak{S}^l_A$ and
2 $\sum_{i=1}^{l} k_i$ decryption secrets $\mathbb{S}^1_A, \ldots, \mathbb{S}^l_A$.

At a later time (during Step S5) Alice can approach T_B, authenticate herself as A' using $P_{A'}$ secrets and receive $\mathcal{B}$ secrets $\mathcal{B}_{A'}$ corresponding to ID A'. Note that while T_B knows that A' is a pseudo-ID, there is no way for T_B to determine correspondence between A and A'. Also note that in order to generate the shared secret $P_{A'B'}$ required for step S5, all that the TM has to evaluate is $h(P_{A'} \parallel B')$. Thus Alice's TM does *not* need access to the ID A' to receive the secrets $\mathcal{B}_{A'}$.

4.3 Subscribing to gateways (S3, S6)

Anyone with a valid public ID and/or pseudo-ID is eligible to seek subscription in any of the communities. The gateways for each MB are themselves are KDCs of a LM-KDS deployment. Thus gateway G is the KDC T_W of a system $\mathcal{W}$ with master secret W. The gateway issues two secrets – corresponding to public and pseudo-IDs of all participants seeking subscription. Thus Alice's TM receives the $\mathcal{W}$ secret $W_A = h(W \parallel A)$, and (at a later time), authenticating itself as A' receives $\mathcal{W}$ secret $W_{A'} = h(W \parallel A')$.

While Alice's TM protects the secrets W_A and $W_{A'}$ from Alice, Alice is provided with secrets $W_{AA} = h(W_A)$ and $W_{A'A'} = h(W_{A'})$.

Note that while[15] Step S2 (receiving $\mathcal{M}$ and $\mathcal{B}$ secrets) is not a prerequisite for S3 (seeking subscription in MBs), without $\mathcal{M}$ and $\mathcal{B}$ secrets the participants cannot use the MB.

Obviously, the assumption that T_B and T_W (who issue secrets corresponding to both public and pseudo-IDs) cannot discover correspondences between public and pseudo-IDs will hold only if:

1 participants obtain secrets corresponding pseudo-IDs *well after* they receive the secrets corresponding to public IDs and
2 there are enough number of participants in the system to provide a reasonable amount of ambiguity.

A solution is to limit the SCMB system to use only public IDs till there a large number of participants in the system (and in each MB). Only then nodes will be allowed to contact I_P to receive their pseudo-ID (in other words there may be a considerable amount of time lapse between Steps S3 and S4).

4.4 Security policies

Once all participants have been assigned secrets, the IAs and the KDC T_B have no role to play in the regular operation of the SCMB system. The participants use system $\mathcal{M}$ secrets (and public values) for mutual authentication and $\mathcal{B}$ secrets for broadcast encryption.

4.4.1 Protected and unprotected secrets

Of all secrets assigned to Alice, the following are protected (not revealed to Alice) by the TM:

1 the secrets U_A, S_{IA}, S_{MA_1} issued by I_A, and the secrets $P_{A'}$ and $S_{MA'_1}$ issued by the I_P

2 $\sum_{i=1}^{l} k_i$, $\mathcal{B}$ decryption secrets (corresponding to both IDs, A and A')

3 the group secrets, for example, G_i, which will be decrypted using the $\mathcal{B}$ decryption secrets and

4 secrets issued by the gateway[16] W_A and $W_{A'}$.

However, Alice (using the external device) is responsible for protecting and using:

1 $\mathcal{M}$ secrets M_{A_1} and $M_{A'_1}$ used for mutual authentication of SPMB participants

2 $\sum_{i=1}^{l} m_i$, $\mathcal{B}$ encryption secrets (corresponding to both IDs, A and A'), used for encrypting group secrets by the group controller (the group secret is *not* protected from the group controller) and

3 secrets $W_{AA} = h(W_A)$ and $W_{A'A'} = h(W_{A'})$ revealed to Alice by her TM.

Furthermore, depending on the policies imposed by the group leader, the TM can also reveal $G_i^1 = h(G_i)$ to the end user.

4.4.2 Revocation

The SCMB includes a Deployment Monitoring Authority (DMA) who loosely monitors SCMB deployments for possible security breaches. End-user TMs will be required to periodically respond to some 'challenges' posed by a DMAs. The challenges (say a random nonce) may be posted in message boards as broadcast messages from D and or D'. The responses should consist of encryption of the challenge using various secrets of the form $h(K_i \parallel D)$ to demonstrate to the DMA that TMs are still functional. Thus TMs that have been rendered non-functional due to tampering attempts can be identified. As the number of such non-functional TMs is at least a good indication of threat levels, the DMAs may advice message board operators on appropriate policies to be followed.

Each TM will provide two independent responses for each such challenge – corresponding to the two IDs. Thus it is possible that a public ID of some TM may be revoked while the pseudo-ID is not (or vice-versa). DMAs could broadcast revocation lists. However, revocation lists will be verified by the end-user external devices. Thus DMAs could sign revocation lists using digital signatures.

Periodically the system $\mathcal{B}$ (A-RPS) secrets could be renewed, following the same approach as initial issue of $\mathcal{B}$ secrets. Nodes that have been revoked will not be allowed to take part in renewal. Thus revocation lists can be flushed after each renewal.

The LM-KDS KDC T_M is a possible victim of Denial of Service (DoS) attacks, as T_M can be flooded with unnecessary requests for public values. To mitigate this, the KDC T_M may be protected by a gateway T_G. Similar to the process where I_A and I_P 'direct' T_M to issue secrets for some ID (like A_1 for Alice), they can also direct T_G to issue a secret – say G_{A_1} and $G_{A'_1}$ to Alice's TM. While the $\mathcal{M}$ secrets M_{A_1} and $M_{A'_1}$ issued by T_M are not protected from Alice by her TM, the secrets G_{A_1} and $G_{A'_1}$ are. Thus under hostile conditions requests to the T_M (through T_G), will call for the use of TMs.

5 SCMB message boards

Messages posted on SCMB message boards can take three basic forms:

1 unicast messages from a specific source to a specific addressee

2 broadcast messages that identify the source. Examples of such messages include solicitation of members by participants who create interest groups and

3 multicast messages meant for members of a group.

In each case the identity of the source/addressee could be the public ID or the pseudo ID. However, in all cases we wish to ensure that no one other than the intended addressees will even be able to determine either the identity of source or the intended recipient(s) – even in situations where participants employ their pseudo-IDs.

Apart from verification of authenticity of the source of any post, one of the main functions of the gateway is catering for privacy of interacting parties from other entities in the system. However, note that the gateway does not have any system $\mathcal{M}$ or $\mathcal{B}$ secrets used by participants for mutual authentication and BE.

Any post, when sent by the end-user device to the gateway, will consist of three main parts:

1 source and destination IDs

2 an 'instruction field' and

3 message content (encrypted/authenticated with secrets shared by source and destination).

The first function (authenticating posts) of the Gateway amounts to verifying if the source ID is authentic. All posts by A are authenticated by appending a MAC based on W_A that Alice's TM shares with the gateway (or using $W_{A'}$ if Alice posts using her pseudo-ID). The gateway strips the authentication information, and adds a timestamp to the post before posting it on the message board.

The second function of the gateway is to conceal the identities of the source and destination from participants other than the source and the destination (except for broadcast messages). This is catered for, by sharing a course value of time by all participants in the system. For example, during the interval t, all messages addressed to C (say Charlie) will indicate the addressee as $h(W_{CC} \parallel t)$, where W_{CC} is the secret known to Charlie, his TM C, and the gateway. Thus the post by Alice addressed to C will implicitly 'request the gateway' (using the instruction field) to modify the addressee field from C to $h(W_{CC} \parallel t)$ (as Alice is not privy to W_{CC}). The instruction field can also indicate if the source field (in the message posted by the gateway), and the reminder of the message should be encrypted with W_{CC}.

A special addressee code is reserved for broadcast messages (e.g. messages announcing new groups and soliciting memberships). Likewise, for messages meant for members of a preexisting group, with a shared group secret G_i^1, the addressee could be indicated as $h(G_i^1 \parallel t)$. In this case, the instruction field in Alice's post will direct the

gateway to post the message *without* modifying the addressee field (as the gateway is not privy to the group secret G_i^1 established using B secrets).

Users may also desire to encrypt all fields (except the source) of their messages (e.g. destination and instruction fields), over the link between the participant and the gateway. For messages from A, the secret W_{AA} can be used for this purpose. Such messages will be decrypted by the gateway before they are posted.

End-users could also post messages to the gateway to allow/block unsolicited messages from specific ID (either by explicitly blocking or explicitly specifying IDs, or by blocking all pseudo-IDs etc.) This feature may however be provided only based on a subscription fee as this may call for substantial overheads for the Gateway.

Note that everything that leaves the end-users device (except the MAC appended by the TM to every post) can be verified by the users. During regular operation, depending on the policies imposed by the message board operator, even the secret W_{AA} can be used for the MAC (instead of W_A used by the TM). Similarly, while all content addressed to a group with group secret G_i may be encrypted with G_i^1, depending on the policies enforced by the group controller a MAC based on G_i (to be appended by the TM) may be required.

5.1 Group memberships

In a scenario, where A seeks membership in a group controlled by C, A can respond to C's solicitation with a unicast message. The response from C to A contains the group secret. The group secret is doubly encrypted – first with the shared secret that can be established between C and A using $\mathcal{B}$ secrets (as discussed in Section 3.4.3), and then with the $\mathcal{M}$ secret shared by A and C. While Alice has access to the $\mathcal{M}$ secret, only her TM (using $\mathcal{B}$ decryption secrets for ID A) can decrypt the group secret.

Revocation of membership secrets is achieved using broadcast encryption. For instance for a group controlled by C the secrets $\mathfrak{S}_C^i$, $1 \leq i \leq l$ are used to encrypt a broadcast secret. At the risk of being repetitious, apart from catering for BE by any participant, one of the most important features of BE using A-RPS is that the identities of the revoked members is not explicitly specified in the revocation broadcast – as only the indexes of the keys used for encrypting the broadcast secret need to be specified.

Note that authentication of all broadcast messages is implicitly catered for by the gateway, as the source ID is verified by the gateway. However, additionally, A-RPS can also be used for broadcast authentication[17], if desired.

5.2 Mitigating DoS attacks

One source of DoS attacks on MBs could be external entities who swamp the gateways with 'posts' which however fail authentication tests. Resistance to such attacks calls for very efficient authentication techniques to be used by the Gateway. Note that verification of authentication by the gateway is indeed very efficient as only symmetric cipher operation needs to be performed.

The second class of DoS attacks can originate from internal entities who can authenticate messages, but send messages, possibly even unsolicited, at very rapid rates. Various simple techniques to avoid such attacks are to:

1 charge users per post (different charges for unicast/broadcast/unsolicited messages) or

2 the TMS could limit the rate of posts.

As it may be extremely burdensome for gateways to keep track of usage information, both approaches will call for using the TM to authenticate every post (say by appending a MAC).

6 Discussions and conclusions

In practical incarnations, a public MB can be a database, where each record is a *post*. The gateway could be a high end trust module plugged into a data base server. Reading from the database could be catered for, by banks of servers which do not use TMs and have only read-access to the database.

The use of light-weight authentication by T_W for messages posted on the board is primarily to reduce the risk of DoS attacks on T_W. For the same reason, it is also desirable to keep the *size* of the messages posted as small as possible. To achieve this the message itself may have the bare minimum components necessary to identify the *target* of the message. The rest of the message could just be a *pointer* to the remainder of the message. For instance the pointer itself could be a hash of the remainder of the message.[18]

The message board (or the data base) could thus be thought of as having two 'layers'. The message sent to T_W would contain the hash of the message, and the message itself would (later) be posted by sending the message to other servers. The follow-up message should contain the index of the post, and the message would be posted in 'layer 2' only if the hash of the message matches the value in the layer 1 post. Thus servers without TMs can have read/write access to layer 2. Servers that access the layer 2 database need not share any SCMB secrets with the participants in the system, as follow up messages are not cryptographically authenticated (only their hash is authenticated in a layer 1 post, which can be verified by layer 2 servers).

The layer 1 database can be indexed by two fields – the time stamp and the target of the message, to facilitate easy search for messages. The database itself could be limited to three fields – the two indexed fields and a third field which is a pointer to the rest of the post. For control messages, the 'rest of the post' could be in layer 2. For other messages where the rest of the post could be large-sized content, the layer 2 message itself could be an URL to the location of the content. For DRM and publish-subscribe application models layer 2 messages can be easily extended to cater for peer-to-peer distribution of content (for example the layer 2 URL could be a bit-torrent link).

The basic SCMB system only caters for establishment of session secrets (conveyed using group secrets). For many collaborative applications the session secret could be provided directly to the PDA/laptop, while the group secret is protected by the TM. However for DRM applications the session secret could be a content encryption key. In such scenarios, the session secret will need to be securely provided to DRM enabled devices. This could be catered for by providing secrets to the TMs by the group controller (or in

this case the content provider) to facilitate establishment of shared secrets between TMs and the DRM enabled devices. Alternately, DRM enabled devices could also be integrated into the SCMB system – by issuing them public IDs and corresponding secrets.

6.1 Conclusions

We have proposed a generic message board model for collaborative systems, catering for a wide range of applications like DRM, publish-subscribe models, e-mail and instant messaging, and provided an architecture and elements of a protocol for secure collaborations over MB.

Trusted computers like smart cards will be used extensively in the future, which can be used to protect the group secrets from the members of the group. Furthermore the use of trust modules can also thwart many of the common denial of service attacks by limiting the rate of posts, and may even be able to keep track of usage information for pay-per-post MB. While the use of trust modules can have several advantages, it is essential to provide end-users with assurances that such trust modules will not violate their privacy. One of the consistent philosophies for the proposed approach for SCMB is to derive *synergistic benefits* by limiting the scope of trust modules, viz., reducing their cost, improving their ability to protect secrets (as there may be no practical limit on the extent and type of shielding employed if heat dissipation is not an issue), and providing users with assurances of protection of privacy by ensuring that the TMs will need the bare minimum information necessary to effectively perform their task. Furthermore, SCMB also permits several trade-offs where end-users may not even need to use their TMs on a day-to-day basis, depending on the policies adopted by MB operators and the group controllers.

One of our current research efforts is developing more concrete specifications for the SCMB protocol, which could serve as the foundation over which more complex protocols and applications could be layered.

References

Anzai, J., Matsuzaki, N. and Matsumoto, T. (1999) 'A method for masked sharing of group keys (3)', *IEICE Technical Report, ISEC99-38*.

Bullock, A. (1998) 'SPACE: spatial access control for collaborative virtual environments', PhD Thesis, University of Notingham.

Butler, T. and Coleman, D. (2003) 'Models of Collaboration', *Strategies for Electronic Collaboration and Knowledge Management*, September, Available at: http://www.collaborate.com.

Canetti, R., Garay, J., Itkis, G., Micciancio, D., Naor, M. and Pinkas, B. (1999) 'Multicast security: a taxonomy and some efficient constructions', *INFOCOMM*.

Dyer, M., Fenner, T., Frieze, A. and Thomason, A. (1995) 'On key storage in secure networks', *Journal of Cryptology*, Vol. 8, pp.189–200.

Eugster,P.T., Felber,P.A., Guerraoui,R. and Kermarrec, A-M. (2000) 'The many faces of publish/subscribe', *Technical Report*, Available at: citeseer.ist.psu.edu/649723.html.

Farley, J. (1998) *Java Distributed Computing*, O'Reilly Press.

Fiat, A. and Noar, M. (1994) 'Broadcast encryption', *Lecture Notes in Computer Science, Advances in Cryptology*, Springer-Verlag, Vol. 773, pp.480–491.

Fiege, L., Zeidler, A., Buchmann, A., Kilian-Kehr, R. and Muhl, G. (2004) 'Security aspects in publish/subscribe systems', *International Workshop on Distributed Event-Based Systems (DEBS '04)*, Edinburgh, Scotland.

Gong, L. and Wheeler, D.J. (1990) 'A matrix key distribution scheme', *Journal of Cryptology*, Vol. 2, No. 2, pp.51–59.

Grawrock (2006) *The Intel Safer Computing Initiative*, Intel Press, January.

Halevy, D. and Shamir, A. (2002) 'The LSD broadcast encryption scheme', *Advances in Cryptology: 22nd Annual International Cryptology Conference*, Santa Barbara, CA, pp.18–22, August.

Hughes, D. and Shmatilov, V. (2004) 'Information hiding, anonymity and privacy: a modular approach', *Journal of Computer Security*.

Khurana, H. (2005) 'Scalable security and accounting services for content-based publish/subscribe systems', *ACM Symposium on Applied Computing (SAC)*, March.

Lee, S., Fox, G., Ko, S., Wang, M. and Qiu, X. (2002) 'Ubiquitous access for collaborative information system using SVG, *Proceedings of SVGopen Conference*, July, Zurich, Switzerland.

Leighton, T. and Micali, S. (1994) 'Secret-key agreement without public-Key cryptography', *Advances in Cryptology*, pp.456–479.

Matyas, S.M. and Meyer, C.H. (1978) 'Generation, distribution and installation of cryptographic keys', *IBM Systems Journal*, Vol. 2, pp.126–137.

Mitchell, C.J. and Piper, F.C. (1995) 'Key storage in secure networks', *Discrete Applied Mathematics,* Vol. 21, pp.215–228.

Moody, Y.W. and Bacon, J. (2002) 'A model of OASIS role-based access control and its support for active security', *ACM Symposium on Access Control Model and Technology*, Chantilly, VA.

Moskowitz, I.S., Newman, R.E., Crepeau, D.P. and Miller, A.R. (2003) 'Covert channels and anonymizing networks', *Proceedings of the ACM Workshop on Privacy in the Electronic Society*, Washington, DC.

Murayama, Y., Gondo, H., Nakamoto, Y., Segawa, N. and Miyazaki, M. (2001) 'A message board system on WWW with visualizing function for on-door communication', *34th Hawaii International Conference on System Sciences*.

Needham, R. and Schroeder, M. (1978) 'Using encryption for authentication in large networks of computers', *Communications of the ACM*, Vol. 21, No. 12, December.

Noar, D., Noar, M. and Lotspiech, J. (2001) 'Revocation and tracing routines for stateless receivers', *Lecture Notes in Computer Science*, *Advances in Cryptology*, Springer-Verlag, Vol. 2139.

Opyrchal, L. and Prakash, A. (2001) 'Secure distribution of events in content-based publish subscribe systems', *USENIX Security 01*, August, Washington, DC.

Ramkumar, M., Memon, N. and Simha, R. (2003) 'Pre-loaded key based multicast and broadcast authentication in mobile ad-hoc networks', *Globecom*, San Fransisco, CA, December.

Ramkumar, M. (2005) 'On broadcast encryption with random key pre-distribution schemes', *LNCS, ICISS* 2005, Kolkata, India, December, Vol. 3803, pp.304–316.

Smith, S.W. and Weingart, S. (1998) 'Building a high-performance programmable secure coprocessor', *IBM Technical Report RC21102*, February.

Tolone, W., Ahn, G-J. and Pai, T. (2005) 'Access control in collaborative systems', *ACM Computing Surveys*, Vol. 37, No. 1, March.

Wang, C., Carzaniga, A., Evans, D. and Wolf, A.L. (2002) 'Security issues and requirements for internet-scale publish-subscribe systems', *Hawaii International Conference on System Sciences*, January.

Notes

[1]http://www.w3.org/TR/SVG/.

[2]For example, a 'Google Message Board' or a 'Yahoo Message Board'.

[3]Though catering for unregulated read-access is not a practical necessity, from a security stand-point, the assumption that 'reading from the MB is open to all' caters for posting of messages over insecure channels.

[4]If A posts a message for B no one else apart from A and B should even know that A attempted to communicate with B.

[5]See https://www.trustedcomputinggroup.org/.

[6]Asymmetric RPS however does *not* employ any public key primitives.

[7]As we shall see in a later section, a 'higher authority' instructs the LM-KDS to issue secrets corresponding to a ID $A_1 = h(A)$ to the node with ID A.

[8]The limit is the number of bits used to represent the ID of any node – as each node requires a unique ID.

[9]This approach, of using host master keys to encrypt all other keys dates back to Matyas and Meyer (1978).

[10]We shall denote by $K(M)$ the encryption of a value M using a key K, in conjunction with some block cipher.

[11]In some application scenarios even session secrets may need to be protected from the end user. This issue is briefly discussed in Section 6.

[12]We shall henceforth use A to represent Alice's public ID, and her TM. The context will make the distinction clear. We shall also represent by 'Alice' the person or her PDA.

[13]For instance, if Alice knows signatures for two nonce's N_1 and N_2 she could easily fabricate the signature of T_U for N_1N_2. While there are simple techniques to overcome this problem, it is perhaps safer to ensure that both parties do not have freedom in choosing the nonce.

[14]Alice will receive the secrets M_{A_1} and $M_{A_1'}$ independently – at different times.

[15]Similarly S5 is not strictly a prerequisite for step S6.

[16]In practice there may be multiple sets of secrets corresponding to multiple gateways.

[17]After all A-RPS is exactly identical to the broadcast authentication scheme by Canetti et al. (1999) – only that in the SCMB it is primarily *used* for different purposes.

[18]However, for broadcast messages the same strategies may not be appropriate. They may contain more fields to indicate the classification, key words etc., apart from the source of the broadcast.