Multicast Re-keying

- consider the setting of a long-lived dynamic group, say $\mathcal{U}$, with single source broadcast
- an on-line subscription service is one example of this
- the *TA* wants to broadcast to every user in the group, but members may join or leave the group over time
- communications to the group are encrypted with a single *group key*
- every user has a copy of the group key
- users may also have additional LL-keys, that are used to update the system as the group evolves over time
- the system is initialized in a *key predistribution phase*, during which the *TA* gives LL-keys and an initial group key to the users in the network

Re-keying Operations

- when a new user joins the group, that user is given a copy of the current group key, as well as appropriate long-term keys (*user join operation*)
- when a user U leaves the group, a *user revocation operation* is necessary to remove the user from the group
- the user revocation operation will establish a new group key for $\mathcal{U}\backslash\{U\}$ (this is sometimes called *re-keying*)
- in addition, updating of LL-keys may be required as part of user revocation operation

Properties of Multicast Re-keying Schemes

Criteria used to evaluate multicast rekeying schemes include the following:

- communication and storage complexity (and trade-offs), for example:
  - size of broadcast, and
  - size (and number) of secret LL-keys required to be stored by users
- security WRT coalitions of revoked users
- anonymity of revoked users
- flexibility/efficiency of user revocation operations, for example:
  - users must be revoked one at a time, or
  - multiple user revocation is possible (up to some specified bound)

Properties of Multicast Re-keying Schemes (cont.)

- flexibility/efficiency of user join operation, for example:
  - any number of new users may be added easily to the system, or
  - entire system must be reinitialized to add new users (one-time system)
- efficiency of updating LL-keys, for example:
  - no updating required (LL-keys are static), or
  - keys can be updated "efficiently", or
  - entire system must be reinitialized after a user revocation (one-time system)

Some Possible Approaches

Some possible approaches to multicast re-keying include the following:

- blacklisting schemes (using $(1, w)$-cover-free families)
- general broadcast encryption schemes
- re-keying schemes based on threshold schemes (Naor-Pinkas)
- tree-based schemes, such as the logical key hierarchy (LKH)

Using a $(1, w)$-CFF to Revoke a Set of w Users

- LL-keys are distributed according to an incidence matrix derived from a $(1, w)$-CFF(v, n)
- let $|F| = w$, and let K denote the new group key for $\mathcal{U} \backslash F$
- for every
$$i \notin \bigcup_{U_j \in F} keys(U_j),$$
the TA computes $y_i = e_{k_i}(K)$ and broadcasts y_i
- recall that the following properties hold:
  1. no user $U_j \in F$ can compute K (even if the users in F combine all their information)
  2. if $|F| \leq w$, then every user $U_h \notin F$ can decrypt K, because
$$keys(U_h) \not\subseteq \bigcup_{U_j \in F} keys(U_j)$$

Comments

- LL-keys are static in this scheme
- each user stores $O(\log n)$ keys and the broadcast has size $O(\log n)$
- new users can be added to the scheme after it is initialized only if $|\mathcal{U}| < n$
- anonymity is possible, because each user U_j only needs to know the keys in the set $keys(U_j)$ (the entire matrix M does not need to be public)
- if it is desired, users can be revoked in stages (up to a total of w revoked users over some period of time) – see next slide
- however, there doesn't appear to be a convenient way to update LL-keys, so the scheme must be re-initialized after w users have been revoked

Revoking Users in Stages

- suppose we want to revoke the users in F_i at stage i, $1 \leq i \leq T$
- we assume that $|F_1| + \cdots + |F_T| \leq w$ and $F_1, \ldots, F_T$ are disjoint
- let K_i denote the group key at stage i, $1 \leq i \leq T$
- the T group keys are broadcast as follows:

Algorithm: $\textsc{Revoke}(F_1, \ldots, F_T; K_1, \ldots, K_T)$

$F \leftarrow \emptyset$

for $i \leftarrow 1$ **to** T

$$\textbf{do } \begin{cases} F \leftarrow F \cup \{F_i\} \\ \textbf{if } |F| \leq w \\ \quad \textbf{then } \text{broadcast the group key } K_i \text{ to } \mathcal{U} \backslash F \\ \quad \textbf{else } \text{quit} \end{cases}$$

Revocation Using General BES

- LL-keys are static in this scheme
- each user stores $O(n)$ keys and a broadcast has size $O(\log n)$
- users can be added to the scheme after it is initialized only if $|\mathcal{U}| < n$
- in general, anonymity is not possible
- any number of users can be revoked at any given time
- there is no limit on the total number of users that can be revoked
- however, coalitions of more than w revoked users can compute group keys

Basic Naor-Pinkas Scheme

- denote $n = |\mathcal{U}|$
- the TA uses a $(w+1, n)$ Shamir threshold scheme to construct n shares, say $s_1, \ldots, s_n$, of a new group key, K
- every user U_i is given the share s_i in the intialization phase
- let $|F| = w$ be the set of users to be revoked
- in order to activate the "pre-positioned" group key K, the TA broadcasts the w shares s_i for all $U_i \in F$
- every user *not in* F now has $w+1$ shares, which permits K to be computed
- the users in F are not able to compute F, since they (collectively) hold only w shares

Comments

- LL-keys are static in this scheme
- each user stores $O(1)$ keys and the broadcast has size $O(w)$
- users can be added to the scheme after it is initialized by creation of new shares of K (shares are just evaluations of the secret polynomial at new x-values)
- anonymity is possible, if x-co-ordinates corresponding to the shares are broadcast when the new group key is activated
- it is possible to revoke $w' < w$ users by broadcasting the shares of the w' revoked users, along with $w - w'$ newly created shares that do not correspond to any user
- however, the basic scheme does not allow users to be revoked in stages

A "Re-usable" Version of the Naor-Pinkas Scheme

- suppose that keys and shares are defined in a subgroup G of ${\mathbb{Z}_p}^*$, having prime order q, in which the Decision Diffie-Hellman problem is intractable
- let α be a generator of G
- the TA constructs shares of $K \in \mathbb{Z}_q$ using a $(w+1, n)$ Shamir threshold scheme implemented in $\mathbb{Z}_q$
- recall that K can be reconstructed using the formula

$$K = \sum_{j=1}^{w+1} b_j s_{i_j} \bmod q,$$

where the b_j's are coefficients defined as follows:

$$b_j = \prod_{1 \leq k \leq w+1, k \neq j} \frac{x_{i_k}}{x_{i_k} - x_{i_j}}$$

Re-usable Scheme (cont.)

- then it follows that

$$\alpha^K = \prod_{j=1}^{w+1} \alpha^{b_j s_{i_j}} \bmod p$$

- hence, for any r, it follows that

$$\alpha^{rK} = \prod_{j=1}^{w+1} \alpha^{r b_j s_{i_j}} \bmod p$$

- suppose that the TA broadcasts α^r along with w "exponentiated shares" $\gamma_j = \alpha^{r s_{i_j}}$ $(1 \leq j \leq w)$
- a non-revoked user, say $U_{i_{w+1}}$ can compute his own exponentiated share, as follows:

$$\gamma_{w+1} = \alpha^{r s_{i_{w+1}}} = (\alpha^r)^{s_{i_{w+1}}}$$

Re-usable Scheme (cont.)

- then $U_{i_{w+1}}$ can compute α^{rK}:

$$\alpha^{rK} = \prod_{j=1}^{w+1} (\alpha^{r s_{i_j}})^{b_j} \bmod p$$

- α^{rK} is the new group key
- by choosing random values of r for each broadcast, a series of new group keys can be set up
- as in previous schemes, coalitions of more than w revoked users can compute group keys
- we would prefer a rekeying scheme that permits an aribtrary number of users to be revoked
- achieving this property apparently requires updating LL-keys when users are revoked