Lazy handshake messages

MLS interim 2019-1

Multi-device modes

- Each device has its own leaf
  - Adding and removing members results in multiple handshake messages
  - The roster is a list of devices, not members
  - Some logic is pushed to the application layer

Multi-device modes

- Leaves represent a member, devices are in a sub-tree (inspired by ART)
  - Adding and removing devices results in an update handshake message
  - The roster is a list of members, not devices
  - Additional logic is required to handle the sub-tree
  - Bonus: Users have a secure channel between their devices for free

Motivation

- Currently the biggest pain point is adding a device to an account
- This is specific to MLS, other messaging system don’t have strong notions of groups
- If the penalty is too big, this might hinder the adoption of MLS

Current costs

- An update message has to be generated in every group
- The cost is O(log n) to O(n) per group
- There might be hundreds of groups
- NB: Many groups might be inactive

Lazy handshake messages

- The following is limited to sub-tree multi-device
- We only need to look at the Update handshake message
- We consider a new variant: LazyUpdate

LazyUpdate

1. The sender creates a new leaf node (NLN), but doesn't "hash it up".
2. The public key of the NLN is sent to the group in a new HS message (LazyUpdate)
3. Every member in the group replaces the old leaf node (OLN) of the sender with the new leaf node (NLN)
4. Every member blanks the sender's direct path

Next steps

- We could repeat this: multiple LazyUpdates could be accumulated
- Before anything else happens: someone must issue a regular Update
  - The member that sent the LazyUpdate or
  - Anyone

Benefit

- At the time of addition, a client only needs to compute a key pair and sign it
- The bulk of the cost is payed later, possibly by someone else
- The postponed cost might decrease, if LazyUpdates are accumulated
- Decisions of when to pay the cost is up to vendors, various strategies might exist

Misc

- For the 1 device = 1 leaf mode, Add and Remove could also have lazy counterparts
- There might be some overlap between LazyAdd and UserAdd