

Group contacts management

Problem

Currently for each joining group member two connections are created - one for group communication, and one for member's direct contact. The member's contact connection is created to enable members to communicate with each other directly outside of group, as in "Send direct message" functionality we have in mobile applications, similarly to the same feature available in other messengers.

It works well for small groups where members trust each other, since it allows for immediate communication between members after connections are established on the group join - otherwise group members would have to establish direct connection separately, and often wait for several more asynchronous interactions before being able to send messages. It doesn't work as well in some other communication scenarios, and entails certain problems:

- Larger "public" groups and/or groups where members don't trust each other - such groups' members may want to communicate with other group members only inside the group, and not have a direct contact with each joining member.
- Groups where owner/host doesn't want group members to be able to communicate directly between each other, e.g. inter-business communication, or some other asymmetric scenario where sensitive messages should not be shared between members.

It should be mentioned though that a group connection is no different from a direct connection on the protocol level, and it's entirely possible to use a group connection for direct communication with modified clients, so any change to the existing group protocol we make to SimpleX Chat clients does not fully solve this issue - group's owner/host can't know whether members' clients are unmodified. So probably we shouldn't be taking this scenario into consideration.

- Using a modified client, a host can carry out MITM attack(s) when introducing a new group member to existing members, so group members can't know whether their direct connection is secure.

This problem can be partially softened by allowing members to validate some connection fingerprint out-of-band, or even to "straighten" their connection (replace it with one established by passing secret out-of-band). This can be a separate feature. It requires members to have access to some out-of-band channel and is also not possible entirely in the current group protocol, so anyway it's another scenario

where automatically establishing a direct connection backfires.

Another point to consider is that currently we do not expose the information about direct connection "indirectness" level enough for users to distinguish contacts created via groups (and so may have been compromised by a MITM attack) - we can mark such contacts in UI.

- Users can't delete group member contacts while they have groups that have these contacts as members. Users may want it either because of one of the reasons above, or purely out of aesthetic reasons, e.g. to avoid cluttering chat with unused contacts. It's not a protocol limitation, only a property of existing implementation, so we should be able to change it with some schema/code changes.

Another related problem is that for group members that join via a group link, a contact is created and not even hidden from chat list (unlike introduced members' contacts). This is true for both sides - the joining member, and the host who invites via a group link.

Solution

Out of the listed problems in existing group protocol we could change the fact that direct connections and contacts are created unconditionally, allow to delete group member contacts without deleting groups and hide/disable/delete contacts created via group links.

Establishing direct connections

- We can add a group wide configuration deciding whether to establish direct connections for this group or not, configurable by a group owner.
- Should it be a part of group profile? If so, profile update can be received by group members asynchronously and they should be able to process situations where one has this setting enabled and another not - probably if any one of them has it disabled, direct connection shouldn't be created. E.g., an introduced member can simply ignore `directConnReq` from `XGrpMemFwd`.
- Alternatively or additionally it can be a global user setting.
- If it's a user setting, it can be communicated when sending `XGrpAcpt` (additional field for host to consider in future introductions?), `XGrpMemIntro` (additional field in `MemberInfo`?), `XGrpMemInv`, `XGrpMemFwd` (make `directConnReq` Maybe in `IntroInvitation`?).
- How do we make it backwards compatible? Just send an empty string in `directConnReq` so the attempt to establish a direct connection by the introduced member is failed?

Should there still be an option to request direct connection with member, e.g. by sending a new type of message inside a group connection?

This is a rather complex change if it is to be properly communicated between group members and can be designed / implemented in a separate scope.

Deleting group member contacts

Table `group_members` has `contact_id` foreign key with cascade deletion, options to allow contact deletion:

- When deleting contact, search group members with corresponding contact id and set it to null, also do not delete contact profile and local display name if it had associated members.
- Re-create table with constraint defined as `ON DELETE SET NULL` - the required migration is more complex than the first option and the resulting optimization is unnecessary.

Another option is to mark contact as deleted (could be a dedicated flag) and hide it, and delete the direct connection.

Group link contacts

On the inviting side (the one that created link and auto-accepts group join requests):

- To hide joining contacts:
 - We shouldn't create group invitation chat item in direct chat.
 - Probably a new flag is required so that these contacts are filtered out together with introduced contacts (see `c.conn_level = 0 OR i.chat_item_id IS NOT NULL` filter in `getDirectChatPreviews_`).
 - We should also filter out pending connections in `getContactConnectionChatPreviews_`, probably a separate flag is needed in the connections for that as well.
 - We already filter out respective contact requests out in `getContactRequestChatPreviews_`, see `uc.group_id IS NULL`).
 - User should still be made aware that his client auto-accepted and invited a joining member - it can be done via a new `RcvGroupEvent` chat item, e.g., "invited a new contact X via group link" (should it be created as unread?).
- Interaction between group links and "no direct connections for groups" feature:
 - Delete created contact after member joins group?

- Communicate the setting to the joining member's client, if he still sends messages ignore them. Can be part of group link metadata.

On the joining side:

- Same or similar filtering logic for contact pending connections and contacts can be applied if we include metadata into group link - the fact that it's a group link should be enough.
- Interaction with "no direct connections for groups" feature - group link metadata can include flag that host's contact and direct connection are to be removed after joining. Joining client should respect this flag, otherwise his messages other than required for group join may be ignored by host's client - see above.

The problem that a group link contact is not filtered out is less pressing on the joining side compared to the inviting side as the latter will/may have uncontrolled amount of contacts connected via a group link, when the joining side will only have one per link and it's created on user action, so we may ignore this for the joining side initially.