layout: layouts/article.html title: "SimpleX Chat v5.2 released: message delivery receipts" date: 2023-07-22 image: images/20230722-receipts-full.png previewBody: blog_previews/20230722.html

permalink: "/blog/20230722-simplex-chat-v5-2-message-delivery-receipts.html"

SimpleX Chat v5.2 released: message delivery receipts

Published: July 22, 2023

What's new in v5.2:

- message delivery receipts – with opt-out per contact!
- filter favorite and unread chats.
- more usable groups:
  - view full original replied message in info.
  - share your address with your contacts and group members via your chat profile.
  - search new and existing members.
- stability improvements:
  - keep your connections working after restoring from backup.
  - restart app (Android) or reconnect servers (iOS).
  - more reliable switching of the receiving address.
  - more stable message delivery.
- other improvements:
  - better disappearing messages.
  - prohibit message reactions.

Platform evolution:

- problems of public groups:
  - why not all messages are received.
  - how to cope with these problems.
  - when will public groups be more usable.
- what about read receipts?

What's new in v5.2

Message delivery receipts

Most messaging apps add two ticks to sent messages – the first one to show that the message is accepted by the server, and the second – that it is delivered to the recipient's device. It confirms that the network is functioning, and that the message is not lost or delayed. SimpleX Chat now has this feature too!

In some cases it may compromise recipients' privacy, as they show that the recipient is online, so we made sending delivery receipts optional – it can be disabled separately for each chat profile or contact. For the new chat profiles this feature is enabled by default.

To avoid compromising your privacy, sending delivery receipts is disabled for all your existing chat profiles. The first time you start the app after the update, you will be offered to enable them for all visible profiles, and they can be enabled later via Privacy and Security settings.

Filter favorite and unread chats

You can now mark your contacts and groups as favorite, to be able to find them faster. With filter enabled, you will only see favorite chats, chats that contain unread messages and also any unaccepted group invitations and contact requests.

More usable groups

Active SimpleX Chat users know how broken the current group experience is, and that we plan some major overhaul of the groups protocol – more on that below. In the meanwhile, we added some simple features that make groups in their current state a bit more usable.

What is this in reply to?

A major problem is that you can see replies to the messages you've not seen before - this would happen both when you just join the group, and didn't connect to most other members, and also when other new members join the group and they didn't yet connect to you – so literally all the time, and the bigger the group gets, the worse it becomes. While this problem cannot be solved without major group protocol changes, at least there is now ability to see the original message that was replied to via the message information.

How to connect to this member?

To simplify direct connections with other group members, you can now share your SimpleX address via your chat profile, and group members can send you a contact request even if the group does not allow direct messages.

How to find a member in the list?

Large member lists (and also the long lists of contacts, if you have many of them) become hard to scroll through, so now there is a search in both lists.

Stability improvements

This version fixes many long-standing problems with the message delivery, failed connections with group members (that also contributed to group fragmentation), and reduces traffic in groups (beta users experienced a

traffic spike because of this fix, but it doesn't affect the final v5.2 release). It's not the end of the road to making SimpleX Chat as stable as mainstream messengers, but it is a big improvement.

Please report the cases when messages are not delivered – delivery receipts should help with that.

Messages failed to decrypt? Problem solved!

Previously, a growing number of users had the issue when after restoring the chat database from backup, messages from some contacts failed to decrypt and were showing an error in the app.

This happens due to double ratchet protocol protecting the integrity of end-to-end encryption after the compromise - post-compromise security. The protocol logic does not allow to use the old version of the database to decrypt the message.

v5.2 added the extension to the messaging protocol allowing to negotiate the new ratchet keys in such cases - both with the contacts and the group members. This requires a user action, and it resets the security code verification status for this contact or member – you need to verify it again to have the additional protection from man-in-the-middle attacks.

The negotiation of the new ratchet keys still happens via the end-to-end encrypted messages, as the protocol has two layers of end-to-end encryption, so it cannot be compromised by the messaging relays.

You may still lose connection if you or your contact changed the receiving address after you made the backup, so make sure to make a new backup after any receiving address changes.

Reconnect the servers

While v5.2 solved many message delivery issues, there may be some others, but they are usually resolved with app restart. It was difficult to fully restart Android app, as there is a continuosly running background service for notifications that doesn't restart with the app. Now Android app has both Restart and Shutdown buttons that take background service into account.

On iOS you can now pull down the list of conversations to reconnect to all relays without restarting the app.

Better disappearing messages

You can now send a separate disappearing message if the chat preferences allow them, but do not have any time to disappear enabled – this applies both to groups and to contacts. You can also set the time to disappear up to 12 months.

Prohibit message reactions

While most people like message reactions, some conversations make them inappropriate - you can disable them now.

Platform evolution

Problems of public groups

As I wrote above, the major problem is that not all messages are received by all members, or, at least, they may be substantially delayed. Additional problems are various scenarios when the list of members gets out of sync for different members.

How to cope with these problems?

It really helps to only use one link shared with the members to join the group - the one created by the client that is most frequently online, ideally always online. This is sometimes confusing, as any group admin can create another group link, and share it with the members, and if this admin is not online, the new member won't be able to join.

We will add new group features to manage fragmentation - there will be an option to fix the connection with the member that you failed to connect to by passing the link out-of-band. This can be particularly helpful for stable groups of 20-50 people where it's important to see all messages.

In the long term, the only way to make groups usable is to move to a new design. We considered several options.

Why not hosted groups with MLS?

Initially, we considered the design with the dedicated servers, potentially self-hosted, that host groups. This design would require adopting MLS (or similar) protocol for group-wide key agreement. Unfortunately, this design is not sufficiently resilient and easier to censor than decentralized design. Also, MLS protocol is very complex to implement, requires a centralized component, and reduces forward secrecy. So we decided against this approach.

Why not fully decentralized groups?

We also considered rumour-mongering protocol, where all members are equal and participate in message dissemination. The problem with this approach is that it adds a lot of traffic for all members, even those who mostly read messages. Also, it still requires establishing a fully connected graph, and with large groups it becomes prohibitively expensive and unreliable, given that many members join public groups for a limited time.

Members host the groups

We are now considering a middle-ground - the design where the owners and admins host the group, synchronising the state between them, receiving and re-broadcasting the messages between all members. This puts a higher burden on these members, but these clients can be hosted in the cloud, and also group owners have a larger incentive to maintain group integrity. At the same time, this design is better for the rest of the group members, as they don't need to establish connections with all other members, only with a limited number of "hosting" members, and it also better protects their privacy, due to the lack of direct connections between most members.

This approach avoids the need for a group-wide key agreement protocol, as hosting members are expected to have access to all content anyway, so pair-wise ratchets are sufficient. At the same time the content remains end-to-end encrypted, and protected from the outsiders.

This approach also simplifies moderation - the message that needs to be removed simply won't reach the members before it is moderated (in case of automatic or policy-based moderation).

Discovery and content search in such groups will be provided via a dedicated discovery server that will participate in the group, provide an always-online client, and also automatic content moderation functionality - a possible approach to moderation is described here.

We really look forward to your feedback on this design.

What about read receipts?

We have an approximately equal number of users who ask us to add receipts, and those who ask not to add them, even as optional.

While read receipts provide some convenience to the message senders, they introduce a lot of stress for the recipients.

As one of the users in the group wrote it: "The existence of read receipts in other platforms is exhausting and is often a source of undue stress. I have to make a decision to read something and let someone know that I have read something and decided not to respond or merely didn't have the time to respond. The outcome of that is a complex social negotiation with non-theoretical social fallout as a consequence. All in all, it's an invasion of privacy of being able to read things at the pace of the individual as opposed to the pace dictated by others... Most people don't need a read receipt, so leave it to a group of individuals to decide if having read receipts make sense to them for their workflow".

Also read this post about the damage from read receipts and other invasive features, like typing and presense notifications.

There is also no discounting that the presense of read receipts functionality, even as opt-in, creates a social pressure to enable them, with the same consequences – there are many scenarios when they become non-optional in

some relationships. So many users belive, and we share this view, that it is better not to have these features at all. We will be re-assessing this view.

SimpleX platform

Some links to answer the most common questions:

SimpleX Chat security assessment.

How can SimpleX deliver messages without user identifiers.

What are the risks to have identifiers assigned to the users.

Technical details and limitations.

How SimpleX is different from Session, Matrix, Signal, etc..

Visit our website to learn more.

Help us with donations

Huge thank you to everybody who donated to SimpleX Chat!

We are prioritizing users privacy and security - it would be impossible without your support.

Our pledge to our users is that SimpleX protocols are and will remain open, and in public domain, - so anybody can build the future implementations of the clients and the servers. We are building SimpleX platform based on the same principles as email and web, but much more private and secure.

Your donations help us raise more funds – any amount, even the price of the cup of coffee, makes a big difference for us.

See this section for the ways to donate.

Thank you,

Evgeny

SimpleX Chat founder