layout: layouts/article.html title: "SimpleX announces SimpleX Chat v1" date: 2022-01-12 preview: Major protocol changes address all design mistakes identified during concept review by an independent expert.

permalink: "/blog/20220112-simplex-chat-v1-released.html"

SimpleX announces SimpleX Chat v1

Published: Jan 12, 2022

The most private and secure chat and application platform

We are building a new platform for distributed Internet applications where privacy of the messages and the network matter. SimpleX Chat is our first application, a messaging application built on the SimpleX platform.

What is SimpleX?

There is currently no messaging application which respects user privacy and guarantees metadata privacy - in other words, messages could be private, but a third party can always see who is communicating with whom by examining a central service and the connection graph. SimpleX, at it's core, is designed to be truly distributed with no central server. This allows for enormous scalability at low cost, and also makes it virtually impossible to snoop on the network graph.

The first application built on the platform is Simplex Chat, which for now is terminal (command line) based with mobile apps in the pipeline. The platform can easily support a private social network feed and a multitude of other services, which can be developed by the Simplex team or third party developers.

What's new in v1?

Stable protocol implementation

All releases from v1 onwards will be forwards and backwards compatible.

Message encryption has been completely re-engineered to provide forward secrecy and recovery from break-in.

SimpleX Chat v1 now uses:

- double-ratchet E2E encryption using AES-256-GCM cipher with X3DH key agreement using 2 ephemeral Curve448 keys to derive secrets for ratchet initialization. These keys and secrets are separate for each contact, group membership and file transfer.
- in addition to double ratchet, there is a separate E2E encryption in each message queue with DH key exchange using Curve25519 and NaCl crypto-box - separate E2E encryption has been added to avoid having any cipher-text in common between message queues of a single contact (to prevent traffic correlation).
- additional encryption of messages delivered from servers to recipients, also using Curve25519 DH exchange and NaCl crypto-box - to avoid shared cipher-text in sent and received traffic (also to prevent traffic correlation).

Improved user and server authentication and transport

SimpleX now uses ephemeral Ed448 keys to sign and verify client commands to the servers. As before, these keys are different per message queue and do not represent a user's identity.

Instead of ad-hoc encrypted transport we now use TLS 1.2+ limited to the most performant and secure cipher with forward secrecy (ECDHE-ECDSA-CHACHA20POLY1305-SHA256), Curve448 groups and Ed448 keys.

Server identity is validated as part of TLS handshake - the fingerprint of offline server certificate is used as a permanent server identity which is included in server address, to protect against MITM attacks between clients and servers.

SimpleX also uses tls-unique channel binding in each signed client command to the server to protect against replay attacks.

Changes in protocol encoding

We switched from inefficient text-based low level protocol encodings, that simplified early development, to space and performance efficient binary encodings, reducing protocol overhead from circa 15% to 3.7% of transmitted application message size.

Learn more about Simplex

Further details on platform objectives and technical design are available here.

SimpleX Chat client can be used in the terminal on all major desktop platforms (Windows/Mac/Linux) and also on Android devices with Termux.

SimpleX also allows people to host their own servers and own their own chat data. SimpleX servers are exceptionally lightweight and require a single process with the initial memory footprint of under 20 Mb, which grows as the server adds in-memory queues (even with 10,000 queues it uses less than 50Mb, not accounting for messages).

We look forward to you using it!

We look forward to your feedback and suggestions - via GitHub issues or via SimpleX Chat - you can connect to the team with `/simplex` command once you run the chat.