SKSChat

Description

SKSChat is an Instant Secure Messaging System, that enables sending and Receiving cyphered text messages in real time from Android smartphones using a Wi-Fi data network or a mobile data network, 3G, 4G, LTE, etc.

The System has been designed with emphasis on privacy and security.

By default, the application encrypts the message database on the user's device, as well as both ends of the communication, in order to protect all messages that are sent to other users.

SKSChat applies encryption by default, using an algorithm of the Double Ratchet type.

The Secure Instant Messaging server is installed in a customer-controlled location

SKSChat depends on a centralized server that in addition to routing messages also facilitates the automatic exchange of user public keys.

All communications between the client and the server are protected by a communication channel encrypted with TLS. The identifiers are only kept on the servers for the time required to transmit each message. The servers do not keep records on who called who or when.

The end-to-end encryption protocol is designed to prevent third parties (including System administrators) from accessing plain text messages or calls. What's more, even if the encryption keys from a user's device are physically compromised, they cannot be used to decrypt previously transmitted messages.

Encription

SKSChat uses an extension to the XMPP (Extensible Messaging and Presence Protocol) protocol for multi-client end-to-end encryption, that uses the “Double Ratchet” algorithm, to provide multi-end to multi-end encryption, allowing message synchronization securely on multiple clients, even if some of them are offline.

After an initial key exchange, continuous renewal and maintenance of ephemeral session keys is managed. It combines a cryptographic ratchet based on the Diffie-Hellman (DH) key exchange and a ratchet based on a key derivation function (KDF) such as a hash function and, hence, is called double ratchet.

The System offers FS (Forward Secrecy) and guarantee of non-repudiation, so that past sessions are protected against future compromise of secret keys and passwords.

Combined with a public key infrastructure for the retention of pre-generated single-use keys (pre-keys), it allows the initialization of messaging sessions without the presence of the remote pair (asynchronous communication). The use of the triple Diffie-Hellman (3-DH) key exchange as an initial key exchange method improves the denial properties. An example of this is the signal protocol, which combines the double ratchet algorithm, the shortcut keys and a 3 DH link protocol.

The System provides confidentiality, integrity, authentication, participant consistency, destination validation, forward secrecy, backward secrecy (also known as future secret), preservation of causality, unlink of messages, repudiation of messages, repudiation of participation and asynchronousness. It does not provide anonymity preservation and requires servers for message relay and storage of public key material.

In addition to the point-to-point encryption of messages, the server communicates with the client using an encrypted channel provided by TLS.

Functionality

Other capabilities of SKSChat are as follows:

- It enables group conversations.
- It enables private messaging within a group.
- It encrypts by default.
- Allows sending of images.
- Allows sending of video.
- Allows messages in alternative formats to be sent.
- Allows audio messages to be sent.
- Allows location to be sent (in development).
- Notifies message receipt and message read.
- It shows when the other party is typing.
- It enables screen theme selection (light / dark)
- Server listening port configuration is posible (done at server).
- User profile photo can be added.
- Notification can be dissabled.
- History can be deleted.
- A conversation can be terminated.
- Image reshape and resize is possible.
- Text size can be managed in all the app.
- History can be exported to SD card.

Server

Taking as a reference a system to serve 400 users and knowing that the system does not save messaging information, calls or attachments, only a medium-low range server is required to support operations, even considering a significant increase in the number of users.

As an example, an HP DL380 server with 16 GB of dual source RAM and an array of two 1 Tb HDDs in RAID 1 configuration should be enough, with Red Hat (preferred) or Fedora Operating System.

If you require High Availability, or Continuous Availability, the server can be installed on a Stratus ftServer 2810 (Fault Tolerant / Continuous Availability Server solution with built-in Hot pluggable redundant modular components / Intel Xeon processor E5-2630v4, 2.2 GHz, 10 Core per processor, 25MB Cache per processor, Intel Hyper-Threading technology / 8GB DDR4 RAM / 1 300 GB 10K RPM SAS Disks, 1 1.2 TB 10K RPM SAS Disks / DVD-RW / Microsoft Windows 2012 or Latest / 99.999 uptime / Zero loss of data / Zero Failover time / in-flight data protection).