Rakuten Viber

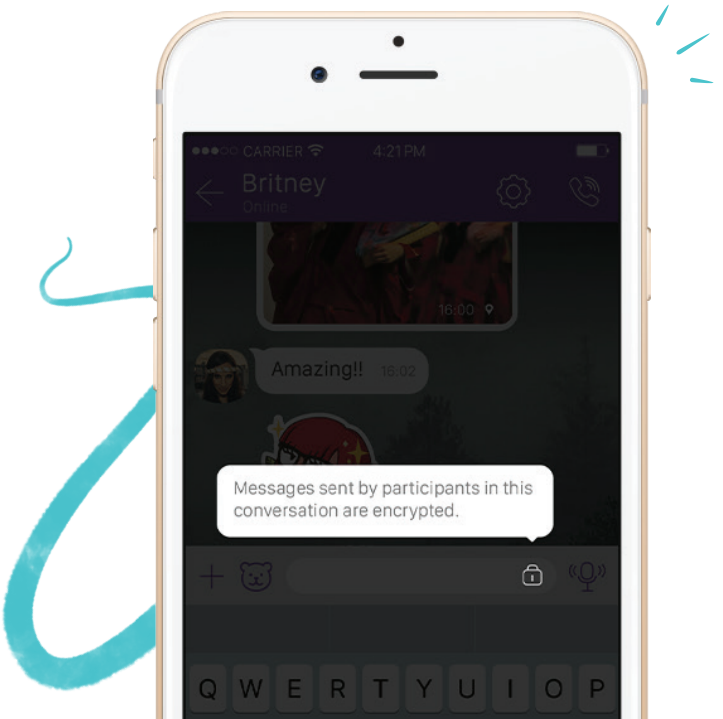
VIBER ENCRYPTION OVERVIEW

INTRODUCTION

This document provides a technical overview of the security protocol implemented by Viber.

Starting with Viber 6.0, all of Viber's core features are secured with end-to-end encryption: calls, one-on-one messages, group messages, media sharing and secondary devices. This means that the encryption keys are stored only on the clients themselves and no one, not even Viber itself, has access to them.

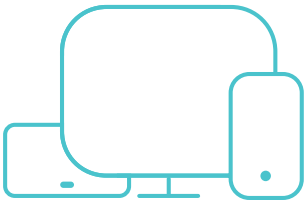
Viber's protocol uses the same concepts of the "double ratchet" protocol used in Open Whisper Systems Signal application, however, Viber's implementation was developed from scratch and does not share Signal's source code.



TERMS

Viber Account – a collection of devices registered with Viber for the same phone number. An account is made up of one primary device and (optional) unlimited secondary devices. Messages sent or received by any device is displayed on all other registered devices of the same account, from the time of their registration and onward (past history cannot be seen).

Primary device – a mobile phone running iOS, Android or Windows Phone, and registered to Viber's service using a phone number managed by a mobile operator.



Secondary device – typically an iPad, Tablet or Desktop PC, which must be linked to a primary device.

ID Key – a long-term 256-bit Curve25519 key pair used to identify a Viber account. The ID key is generated by the primary device only.

PreKeys – a set of medium-term Curve25519 key pairs used to establish one-on-one secure sessions between devices. PreKeys are generated separately by each device in an account.

Session – a two-way secure virtual link between two specific devices. Secure sessions are automatically established between all devices of the same account, and between any two devices of different accounts that exchange messages. A secure session with a different account is identified in Viber by a gray lock in the conversation screen.

Trust – a state between two accounts which indicates that the peer account is authenticated and that no man-in-the-middle attacker has interfered with the session established between any two devices in the account. Users can establish trust by following the authentication procedure described below. A trusted session is identified in Viber by a green lock.

Breach of trust – a state between two accounts that occurs when the ID Key of a trusted peer has changed after trust was established. This can happen legitimately if the peer user has replaced his primary device, reinstalled Viber, or if keys were lost due to storage malfunction. However, it can also happen if a man-in-the-middle is eavesdropping on the conversation. A breached session is identified in Viber by a red lock.

PREPARATIONS FOR SESSION SETUP

During installation, every primary Viber device generates a single 256-bit Curve-25519 key-pair called ID Key. The private part of the key is stored on the device, while the public part of key is uploaded to the Viber servers. Secondary devices (PCs and Tablets), receive the private key from the primary device via a secure method described in "Secondary Device Registration", below.

Additionally, every Viber client generates a series of PreKeys. Each PreKey has two 256-bit Curve-25519 key-pairs called Handshake Key and Ratchet Key. The private keys are all stored on the device, while the public keys are uploaded to the server. The server holds several keys per device and requests more when they drop below a configurable threshold. Each request causes the client to generate additional keys and upload the public parts to the server.

SECURE SESSION SETUP

A session needs to be established only once between every two Viber devices wishing to communicate securely. Once a session exists, it can be used to send an unlimited number of messages in either direction.

To send a secure message, secure sessions must exist between the sending device and all the recipient's devices, as well as between the sending device and all the sender's other devices. So for example, if user A that has a mobile phone and a PC registered to Viber under the same account wishes to communicate with user B that has a mobile phone and a PC, secure sessions must be established between each pair of devices.

Sessions between devices of the same account are established when the devices are registered. Only a single session is required between any two devices, and that session can be used to synchronize any number of conversations with other Viber accounts.

In order to establish a session with a different account, the device ("Alice") wishing to establish a session with a peer ("Bob") sends a query to the Viber server with the recipient's phone number. The server responds with the peer's public ID Key and a series of peer public PreKeys, one per each device registered on "Bob's" account. "Bob's" devices are not required to be online when this happens.

"Alice" device then generates two 256-bit Curve-25519 key-pairs as its own handshake and ratchet keys, and derives a Root Key, as follows:

RootKey = SHA256 (DH(IDAlice, HSBob) || DH(HSAlice, IDBob) || DH(HSAlice, HSBob))
The RootKey is then used to derive a session key using:

TempKey = HMAC_SHA256(RootKey, DH (RatchetAlice, RatchetBob))
New RootKey = HMAC_SHA256(TempKey, "root")
SessionKey = HMAC_SHA256(TempKey, "mesg")
DH indicates the use of Elliptic-Curve Diffie-Hellman key-exchange algorithm. HS indicates Handshake Key.

The different strings passed to the HMAC functions ensure that even if the session key is compromised, the root key cannot be derived back from it.

"Alice" then sends to "Bob" a session start message containing its own public ID, an identifier of "Bob's" pre-key that was used for this session, and its own public handshake and ratchet keys. When "Bob" goes online and retrieves this message from its inbox, it can reconstruct the same Root and Session keys using the same DH procedure.

It should be noted that if a session is established with a peer's primary device but cannot be established with a secondary device (for example, because it is offline and out of PreKeys on the server), the conversation will still be secured with end-to-end encryption, but that secondary device will not be able to decrypt those messages and thus, will not be able to display them.

EXCHANGING MESSAGES

A device sending a message to a target user needs to encrypt this message for every session with every device that the target user has. To accomplish this, an ephemeral one-time 128-bit symmetric key is generated and is used to encrypt the message body using Salsa20 encryption algorithm. This ephemeral message key is then encrypted using each recipient's session key. The sender device sends a unified message to the server, containing one encrypted cyphertext and a set of encrypted ephemeral keys. A server-side fan-out slices this message and delivers the relevant parts to each target device.

The two devices take turns at advancing the session keys in a process called ratcheting. Each time the direction of the conversation changes, the device whose turn it is randomly generates a new Ratchet key pair, and once again performs the following sequence:

TempKey = HMAC_SHA256(RootKey, DH (RatchetAlice, RatchetBob))
New RootKey = HMAC_SHA256(TempKey, "root")
SessionKey = HMAC_SHA256(TempKey, "mesg")
With Ratchetthis_device being the private part of the newly derived key-pair. Alongside each message, the public part of the Ratchetthis_device is also sent. The recipient runs DH with its last private ratchet together with the sender's public ratchet.

This double-ratchet accomplishes two things: first, the continuous ratcheting provides forward and backwards secrecy so even if the keys are compromised, past and future messages cannot be decrypted. Second, the algorithm maintains the authentication of the peer, because the DH chain of the root keys started with both devices' ID Keys. If the ID key of the peer is trusted at any point, the entire chain can be trusted.

ENCRYPTED CALLS

Each side of a call generates an ephemeral 256-bit Curve25519 key-pair. The public part is signed using the device private ID Key and is exchanged between the two devices during call setup phase. The other side authenticates the request using the peer's public ID key.

Each device verifies the signature, and performs DH calculation to derive a one-time session key. The session key is valid only for the duration of this specific call, and is not saved in non-volatile storage.

The RTP stream of the audio or audio/video call is converted to SRTP and encrypted via Salsa20 algorithm using the session key.

PHOTO, VIDEO AND FILE SHARING

The sending client generates an ephemeral, symmetric Salsa20 key and encrypts the file. The encrypted file, together with an HMAC signature, is uploaded to Viber servers. An additional MD5 signature is calculated over the encrypted data and sent together with the file in order to allow the storage server a simple way of verifying transmission integrity regardless of the end-to-end encryption.

The sender then sends a Viber message to the recipient with the ID of the uploaded file and the encryption key. This message is end-to-end encrypted using the encrypted session between the sender and recipient.

The recipient generates a download link from the file ID, downloads the encrypted file and decrypts it using the key.

SECURE GROUPS

All members of a secure group share a common secret key (a symmetric Salsa20 encryption key) which is not known to Viber or to 3rd parties.

For new groups, this shared secret is generated by the group creator and sent to all participants using the secure one-on-one sessions described above. For non-secure groups that were created with past Viber versions, this secret is generated by the first member who sends a message to the group chat after all group members have upgraded to the secure version. In the Viber application any group member can add additional participants to a group. These participants will receive the secret from the group member that added them.

The group secret is ratcheted forward using a HMAC-SHA256 with every message sent. Each group message contains a sequence number that indicates the number of times that the hash function has been invoked. Different clients always pick up where the last message has left off and continue the hashing chain from that point, so keys are not reused. Forward secrecy is maintained by the one-way hashing algorithm; even if the key is compromised, past conversations cannot be decrypted. Past keys are discarded by the client and not saved to not-volatile storage, except for a short window of past hashes which is used in race conditions when two or more participants write to the group simultaneously.

SECONDARY DEVICE REGISTRATION

A key feature in the Viber ecosystem is the concept of secondary devices. A secondary device is a PC, an iPad or a Tablet which is paired with a user's mobile phone and sees the same message history of both incoming and outgoing messages.

Viber's end-to-end encryption on secondary devices works as follows:

Encryption is done separately for each device. If some user A sends a message to user B that has two devices, then user A needs separate end-to-end sessions with the two devices and encrypts the data twice, each one using a different set of keys.

Authentication is done just once for the entire account. If user A trusts user B, the trust is automatically applied to all of user B's devices, not just one.

The authentication is accomplished by sharing the private ID Key between all devices of the same account. The ID Key is generated only by the primary device, and is transmitted to the secondary devices during registration in a secure method, as follows:

The secondary device generates an ephemeral 256-bit Curve-25519 key-pair.
The device then generates a QR code containing the device's "Viber UDID" (a publicly accessible unique device ID generated by Viber) plus the public part of the ephemeral key pair.
The user uses his primary device to scan the QR code. The primary device also generates a 256-bit Curve-25519 key-pair and performs DH calculation with the public key from the QR. The result is hashed using SHA256 to create a shared secret.
The primary then encrypts its own private ID key with this secret and sends it and the public part of the ephemeral key through the Viber servers to the target device, identified by its UDID as read from the QR code. The cyphertext is signed using HMAC-SHA256.
The secondary device receives the message, performs the same DH and hash to obtain the same secret, and uses it to decrypt the primary private ID Key.
The ID key is part of the DH chain that creates the shared secrets for the one-on-one sessions. Therefore, without the correct ID, the secondary device cannot participate in any secure conversations that the primary device is part of.

AUTHENTICATION

Authentication provides a means of identifying that no man-in-the-middle attacker is compromising the end-to-end security. In the Viber application, authentication is done in the context of a Viber audio (or audio/video) call.

When in a call, each user can tap on the lock screen to see a numerical string which is computed as follows:

Both devices perform DH calculation using their own private ID key and the peer public ID key as published during call setup phase.
The DH result is hashed using SHA256 and trimmed to 160 bits.
Those 160 bits are then converted to a string of 48 decimal characters (0-9).
Both parties should see the same string of numbers and can compare them by reading them to the call participant. If both sides hear the other side read out loud the same string of numbers as they see on their own screen, this gives a very high degree of certainty that the ID keys have not been tampered with and that no man-in-the-middle exists for this conversation.
The ID key verification protects both the secure calls and secure 1-1 chats. In calls, the ID key is used to sign the key exchange DH message. In 1-1 chats, the ID key functions as the root of the DH chain leading to the shared secret generation. In turn, 1-1 session protects group sessions because the group keys are exchanged over 1-1 sessions.