linphone

open source VOIP project

Linphone Instant Messaging Encryption

Johan Pascal

FOSDEM 2020

Agenda

- Security requirements
- Protocol overview
- Integration in Linphone group chat with multidevices environment
- Man in the middle attack detection

linphone
open source VOIP project

Linphone

- Is around since 2001
- Is available on android, iOS, Windows, Mac, Linux
- Uses SIP standards for audio, video and instant messaging
- Support group messaging, multiple devices per account

Linphone's team also provides

- Flexisip, an open source SIP Proxy
- A free SIP service sip.linphone.org

linphone
open source VOIP project

Major security requirements for a secure IM :

- Protect content: end-to-end encryption
- Confirm sender and recipient identity: authentication
- Past conversation safe in case of key compromised: forward secrecy
- Recover from compromised key: future secrecy
- Minimal effort from users

First implementation in 2014, based on SCIMP:

- End-to-end encryption and authentication
- Symmetric ratchet provides forward secrecy
- Limited future secrecy
- Users must perform an audio call before exchanging any encrypted message
- Not adapted to group chat (not available in Linphone back in 2014)

linphone
open source VOIP project

Based on the Signal protocol

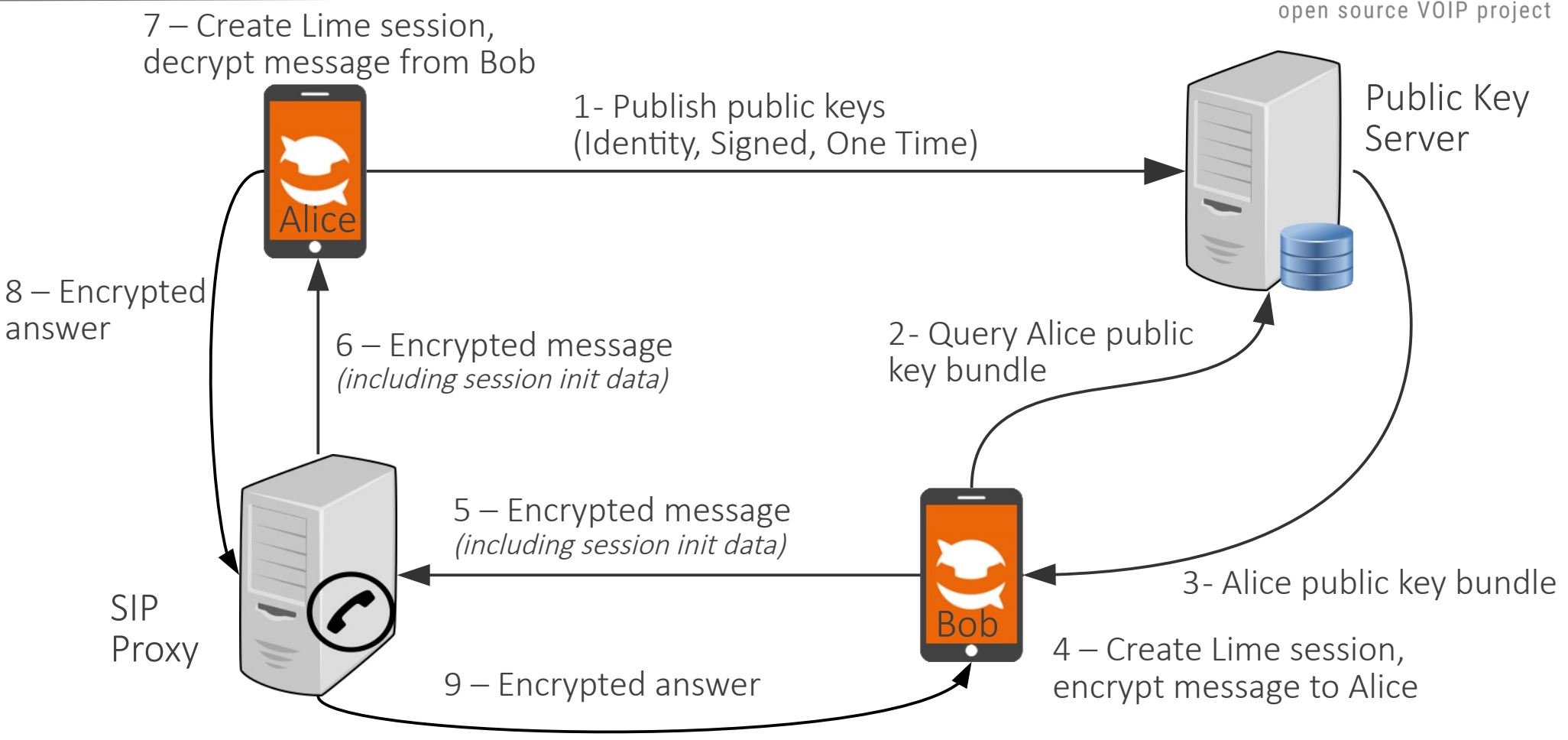
- End-to-end encryption
- Forward and future secrecy
- Asynchronous
- Large deployments
- Open source implementation, well documented: https://signal.org/docs/

Extended to support

- Multiple device per account
- Group chat with future secrecy
- Practical mutual authentication method

Linphone Instant Messaging Encryption: Simplified data flow

linphone
open source VOIP project



linphone
open source VOIP project

Asynchronous key agreement protocol: X3DH

Keys set:

- Public key published on the key server, private key on device only
  - Identity Key (*Ik*): long-term life span, used for both signature(***Sig***) and key exchange (***DH***).
  - Signed PreKey (*SPk*): medium-term life span(weeks), key exchange only.
  - One Time PreKey (*OPk*): discarded after one use in a key exchange.
- On device only, public key in message header
  - Ephemeral Key (*Ek*): used once, private key is discarded immediately.

Bob initiates an exchange: compute a shared Secret Sk

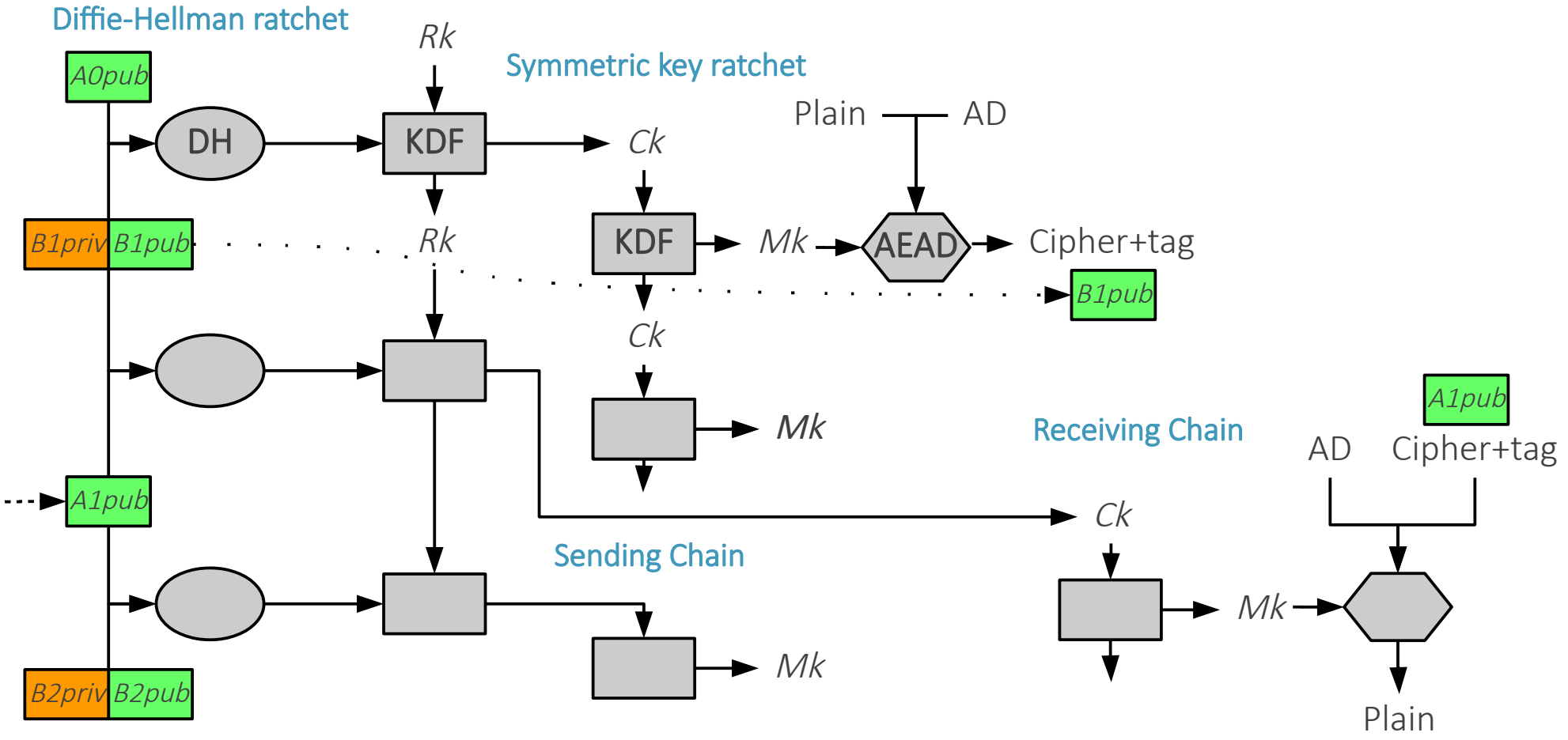
- Fetch *IkA*, *SPkA*, ***Sig***(*IkA*, *SPkA*) and *OPkA* generated by Alice
- Verify ***Sig***(*IkA*, *SpkA*)
- Generate *EkB*
- Compute :
  - *DH1* = ***DH***(*IkB, SpkA*), *DH2* = ***DH***(*EkB, IkA*)
  - *DH3* = ***DH***(*EkB, SpkA*), *DH4* = ***DH***(*EkB, OpkA*)
  - *Sk* = ***KFD***(*DH1,DH2,DH3,DH4*)
- Delete *EkB* private key
- Generate *AD = IkB,IkA,Bob Id,Alice Id*
- Feed encryption protocol with *Sk* and *AD*
- Send *IkB* and *EkB* with the first message to Alice

Alice receives Bob's message and keys

- Compute :
  - *DH1* = ***DH***(*SpkA, IkB*), *DH2* = ***DH***(*IkA, EkB*)
  - *DH3* = ***DH***(*SpkA, EkB*), *DH4* = ***DH***(*OpkA, EkB*)
  - *Sk* = ***KFD***(*DH1,DH2,DH3,DH4*)
- Generate *AD = IkB,IkA,Bob Id,Alice Id*
- Feed decryption protocol with *Sk* and *AD*

linphone
open source VOIP project

Exchange encrypted messages using the Double Ratchet protocol



Lib lime: https://gitlab.linphone.org/BC/public/lime

- Written in C++11
- Flexible use core cryptographic functions
  - support curve-25519 and curve-448 ECDH and Signature
- Signalisation and transport agnostic: can be used with any protocol providing a unique device Id and a connexion to a key server
- Langage bindings to C, java, python

Device identification:

- User identified by a SIP URI: *sip:<username>@<domain>*
- Device identified by a GRUU (RFC5627): *<sip:uri>;gr=urn:uuid:<unique id>*

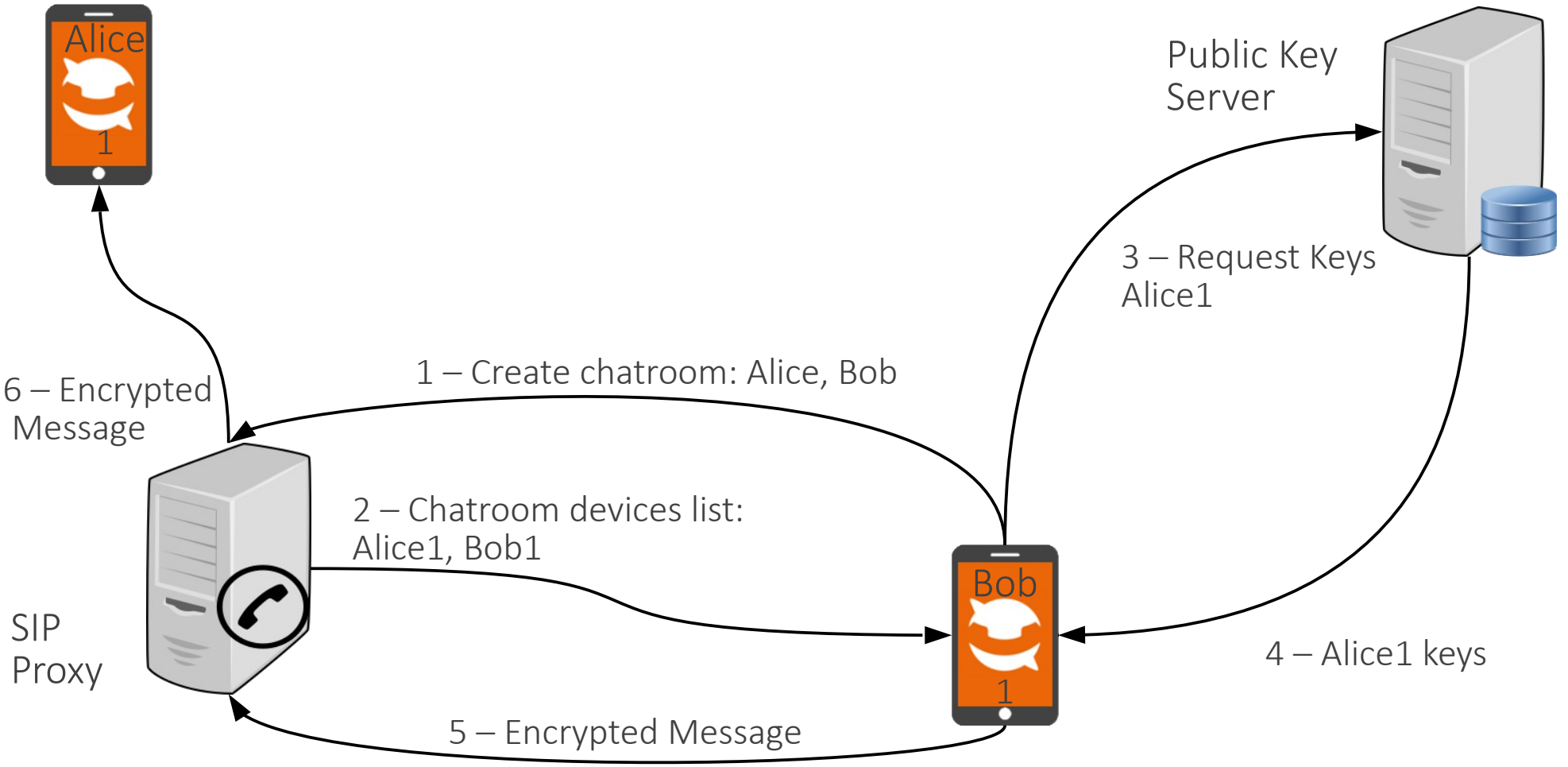
Flexisip SIP proxy, conference server:

- Manage user devices
- Manage group members
- Route encrypted messages

Secure device/server connection:

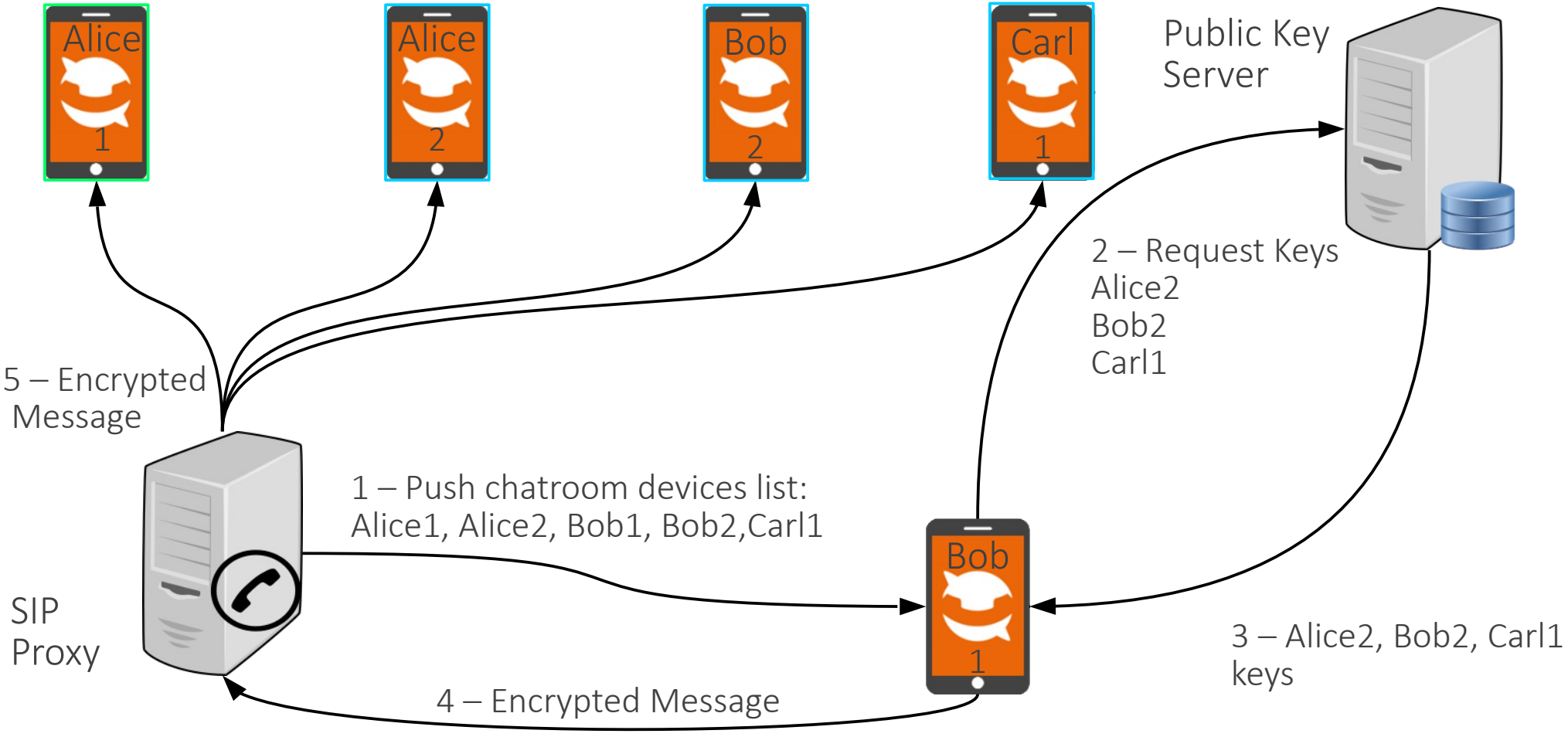
- TLS
- Digest authentication

Linphone Instant Messaging Encryption: multiple devices data flow

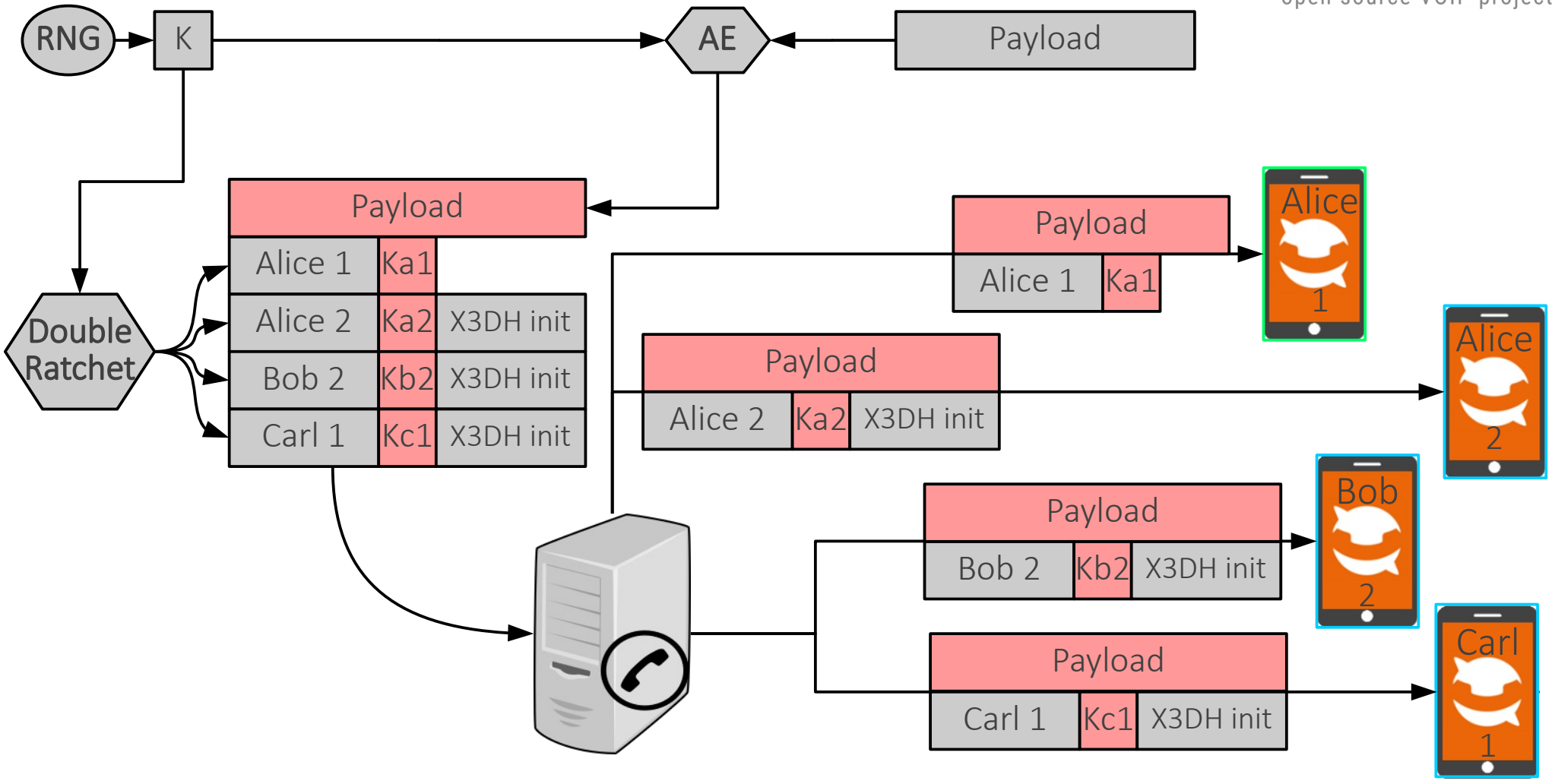


Linphone Instant Messaging Encryption: multiple devices data flow

linphone
open source VOIP project



Linphone Instant Messaging Encryption: Encrypted Message Structure



Pros:

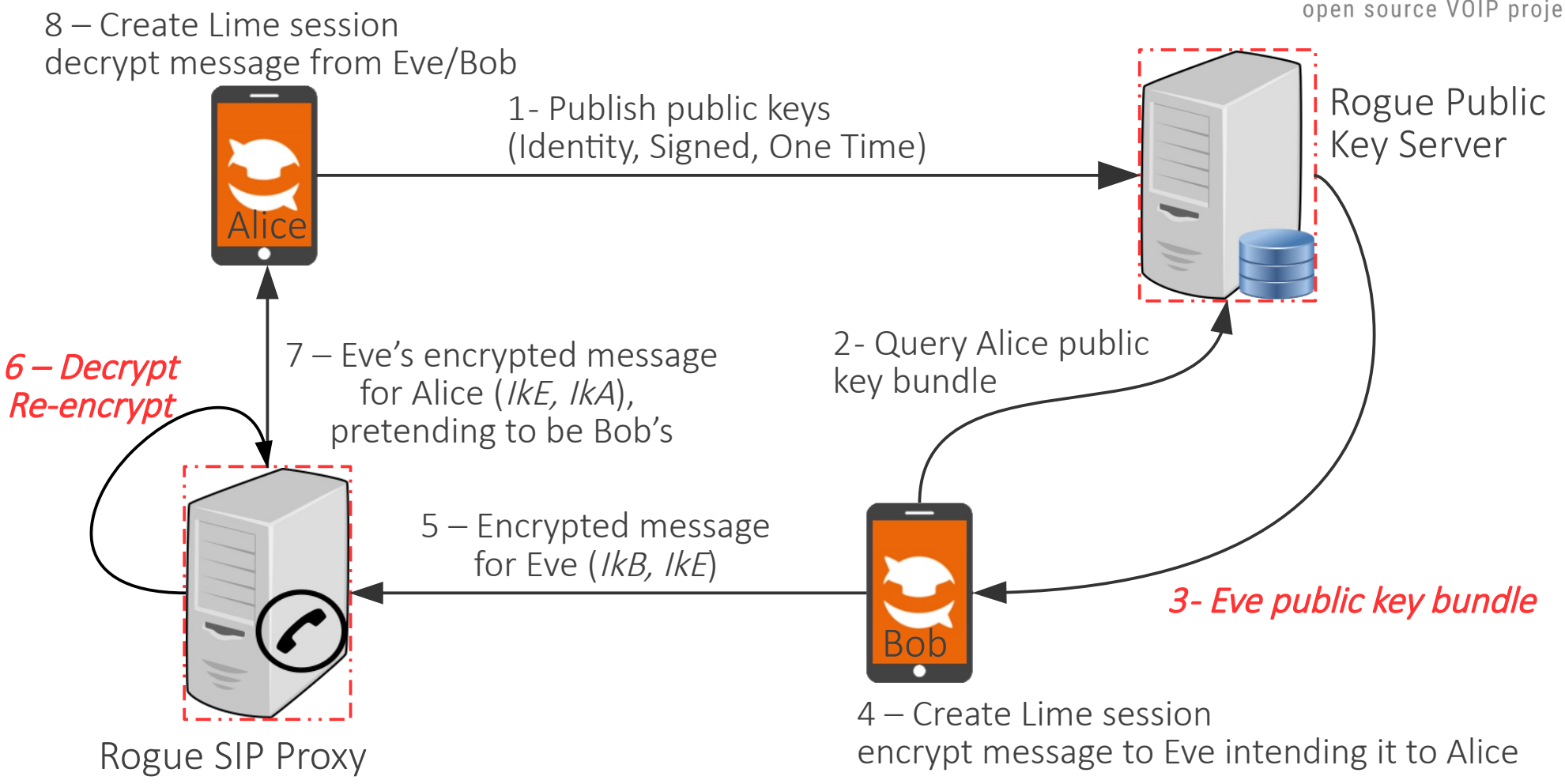
- Group management completely transparent for encryption layer
- Future secrecy maintained
- Group membership fully under device control

Cons:

- Cannot handle massive groups (a few hundred devices is still fine)
- Participant overhead around 100 bytes

Linphone Instant Messaging Encryption: Man-in-the-Middle

linphone
open source VOIP project



linphone
open source VOIP project

Associate Identity Key (*Ik*) with device Id (GRUU):

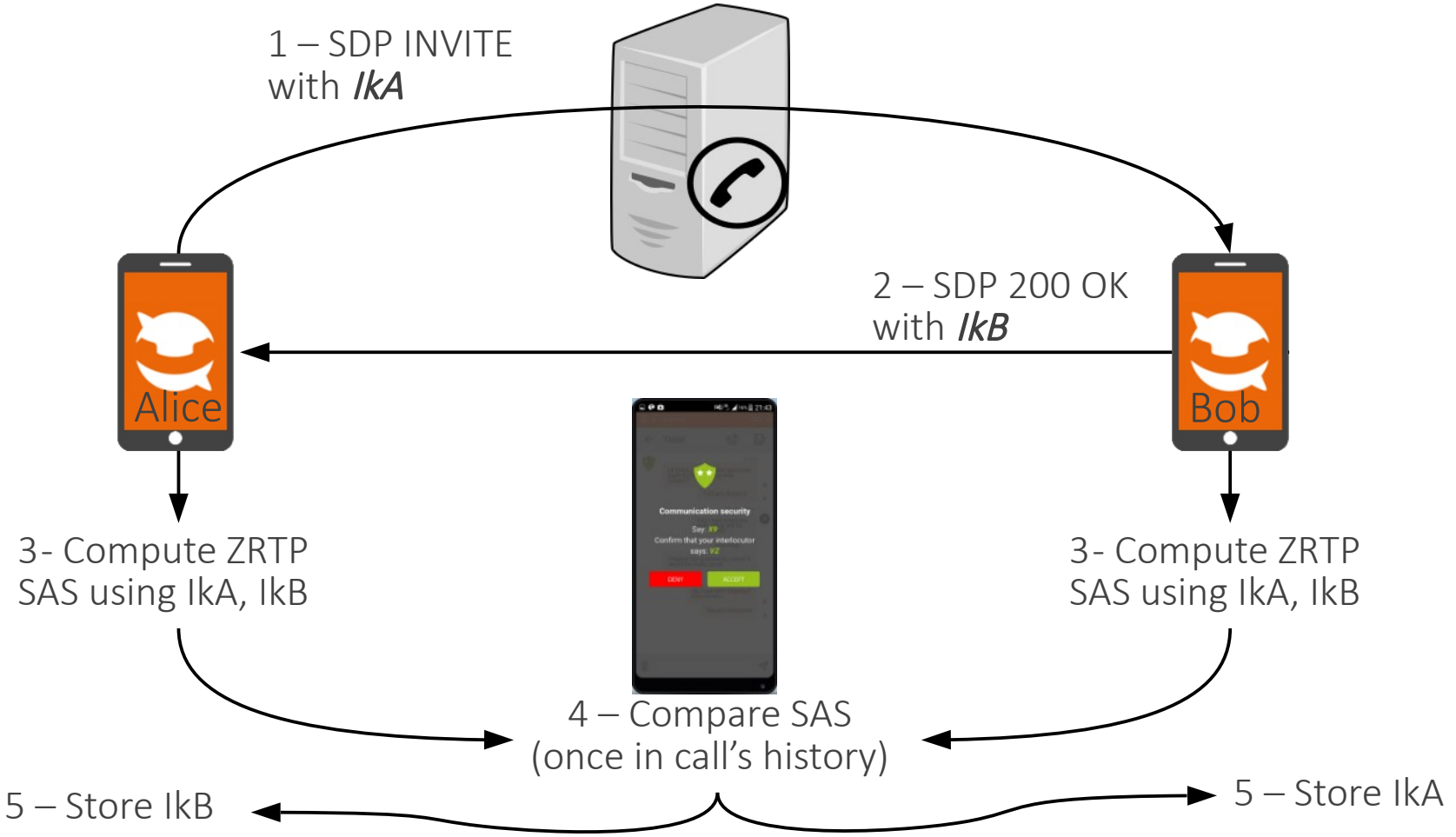
- Signal protocol documentation: "*If authentication is not performed, the parties receive no cryptographic guarantee as to whom they are communicating with.*"

- Implemented solutions:
  - Key fingerprint manual comparison
  - QR code scan on peer device

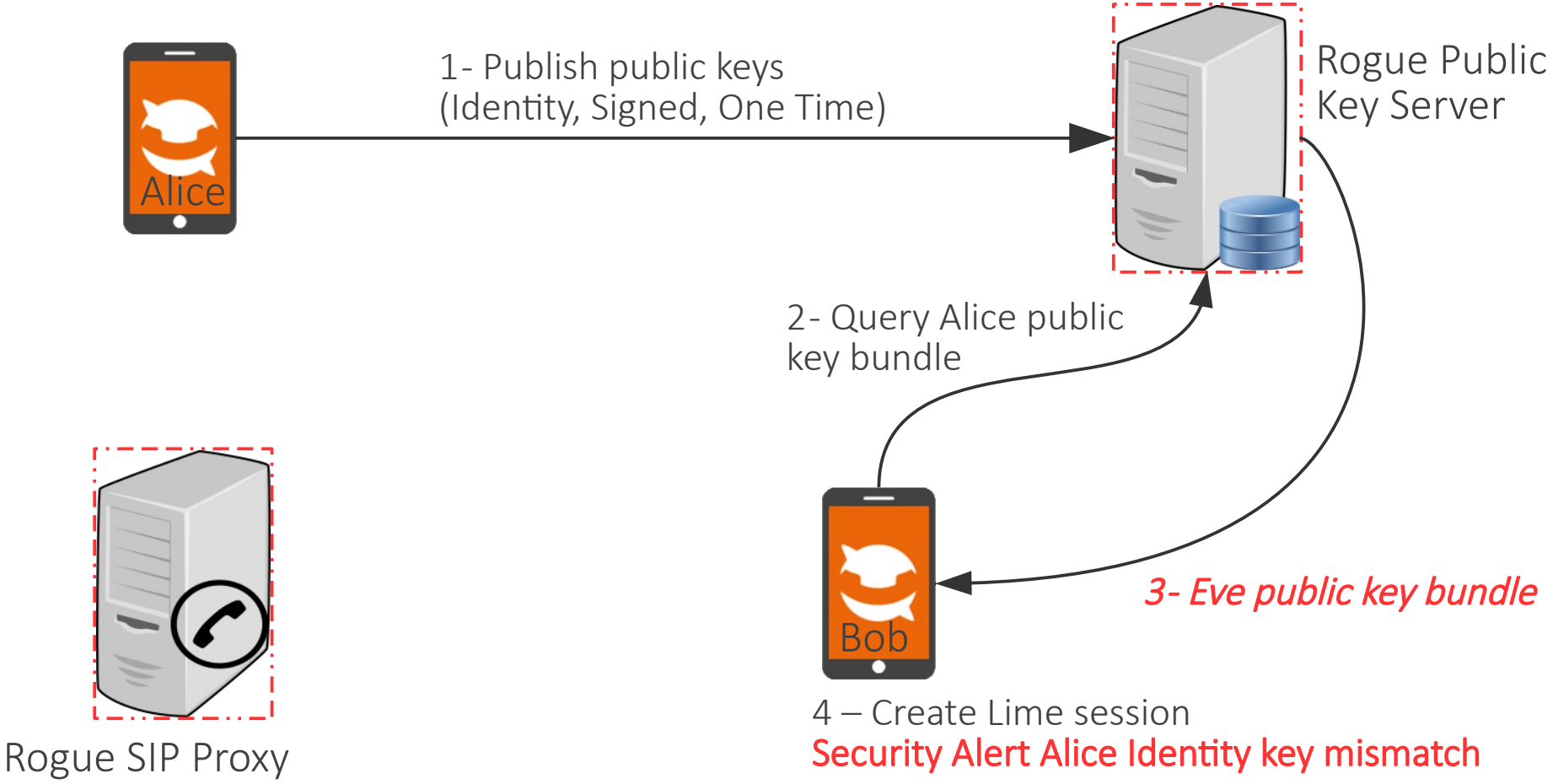
Leverage ZRTP (RFC6189) MitM protection to authenticate peer's Ik:

- ZRTP SAS comparison during audio call detects MitM
- Uses auxiliary secret ZRTP feature
- Is automatic and transparent for users
- Is performed on every call, even before exchanging messages
- ZRTP implementation extends RFC to use ECDH on curves 25519 and 448

Linphone Instant Messaging Encryption: Peer Authentication with ZRTP



Linphone Instant Messaging Encryption: Man-in-the-Middle



Linphone Instant Messaging Encryption: Peer Status

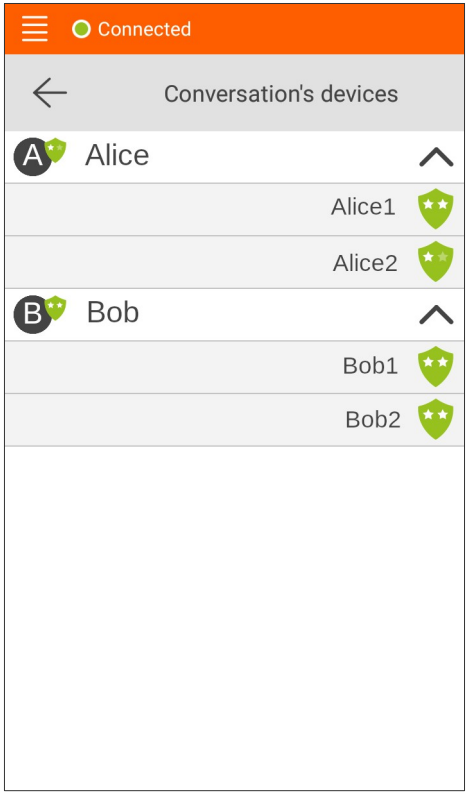
linphone
open source VOIP project

Peer Status :

- Untrusted: identity never confirmed
- Trusted: identity confirmed through ZTRP SAS validation
- Unsafe: Man-in-the-Middle attack detected

Conversation Status :

- Lowest status of all devices involved
- Easy access to all devices included in the conversation and their status



Linphone Instant Messaging Encryption: useful links

linphone
open source VOIP project

- Linphone website: https://www.linphone.org
- Lime corner: https://www.linphone.org/technical-corner/lime
- Lime implementation document:
  https://gitlab.linphone.org/BC/public/lime/blob/master/lime.pdf
- Signal protocol: https://signal.org/docs/

linphone
open source VOIP project

Thank you