Secure and anonymous messaging

Project presentation

Dario Pavllo

École polytechnique fédérale de Lausanne

January 10, 2018

1 - Introduction

Goal: build a robust, scalable messaging system that provides end-to-end encryption, anonymity, and spam prevention.
Why? People are putting less trust into governments; increasing demands for privacy.
Short list of features:

- Public chat room and private messaging, with authentication and integrity.
- Anonymous identities using self-signing names.
- End-to-end encryption in private messages. Only end users can decrypt contents.
- Spam prevention through proof-of-work

Motivation: messaging services like Whatsapp or Telegram already implement some of these features, but they are **centralized**. We want to build our system in a decentralized setting.

2 - Anonymous identities

- Idea borrowed from Tor hidden services (e.g. `blockchainbdgpzk.onion`) and Bitcoin wallet addresses.
- New user generates a public/private key pair (RSA-2048).
- Users are identified by a name derived from the public key (e.g. `alice4jfj49dkalp`). Base32 of the first 80 bits of the SHA-256 hash of the public key.
- Easy to verify identities (names are **self-signing**).
- No name system, no key directory, fully decentralized. Intrinsically resistant to MITM and impersonation.
- No consensus required for registering new names.

3 - Private messaging

- `alice4jfj49dkalp` wants to send a message to `bob3fk1f94olfpoz`.
- Alice encrypts the message with Bob's public key, and signs it using her own private key.
- Nodes distribute the message without being able to see the content.
- Bob decrypts the message with his private key and verifies it with Alice's public key.
- Secure against attacks: all nodes validate signatures; public keys are verifiable.
- Additional privacy/security: RSA-OAEP padding scheme, conflict handling.

4 - Proof-of-work

- Clients must solve a crypto puzzle before sending a new message.
- This provides a rate-limiting mechanism against spam.
- A nonce is appended to the message. The SHA-256 of the message + nonce must start with N leading zeros.
- **Only the original sender computes the nonce!** Intermediary nodes verify it and store it along with the message.

Message ID	From	To	Content	Signature	PoW nonce
1234	`alice4jfj49dkalp`	`bob3fk1f94olfpoz`	Encrypted binary data	256 bytes	16 bytes

5 - Message exchange

- Public key sent as public message (first message)
- Full nodes store the entire database of messages

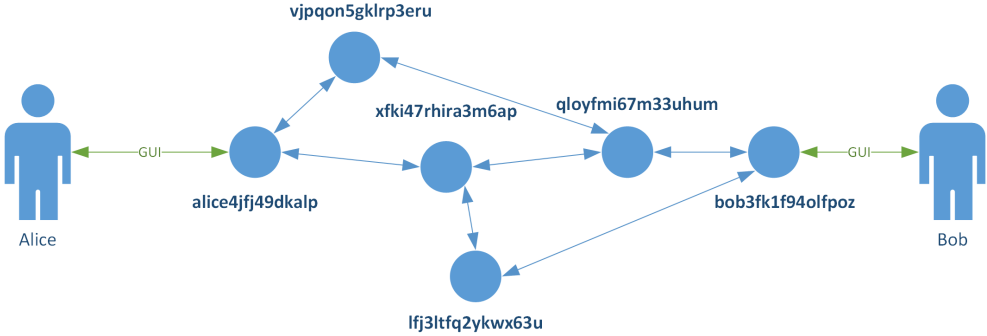

Figure: Diagram that illustrates how users communicate.