Identity and assembly

Creating a messaging architecture to support human rights

SESSION OPTF

With
KEE JEFFERYS

Intro

Our motivation to creating a messaging architecture to support human rights is based on the assumption that information communications about human rights is highly sensitive, and needs to be protected from state and non-state actors who may try to intercept, block or track down the sources of such information.

There are existing ee2e messengers available to human rights defenders (HRDs). However, most of these messenger protocols leak metadata and often require an email or mobile number to use them.

Therefore our approach was to prioritise plausible deniability as well as encrypting the communications. Our messaging architecture - or the Session protocol - uses Oxen's onion routed SN network to ensure privacy of both sender and receiver, and to make it extremely difficult to anyone to intercept the communications - even in their encrypted state (is this true?) or locate the stored message (is this true?), and most importantly, identify the parties associated with the message.

Today - we will cover:

- The key features of Session Protocol • **2 slides**
- Low-latency service node network that facilitates communications • **1 slide**
- The importance of usability for human rights defenders • **1 slide**
- Remaining technical challenges ahead • **1 slide**
- Other uses for the Session Protocol • **1 slide**
- Links to Open Source repository • **1 slide**
- Q&A

Designing a communication Architecture for human rights

Communication is a basic need for everyone in the world, and increasingly, communication is digital.

We wanted to create a communication architecture which would:

- ***Protect the right to free speech***
- ***Prevent censorship***
- ***Protect the right to privacy***

This would give people a safe space to explore their own personal, private identities, organise and assemble, and work to defend human rights in the face of persecution.

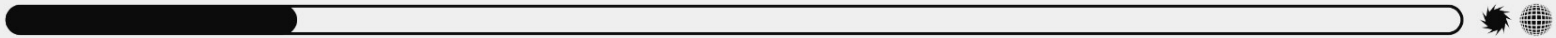
Translating those rights into tech:

The Session protocol

The Session protocol is our design for a messaging app which inherently protects the rights we've discussed. It has

- End-to-end encryption
- Onion-routing
- Decentralised server infrastructure
- Anonymous sign-up

I will discuss these design aspects further



Anonymous Sign Up

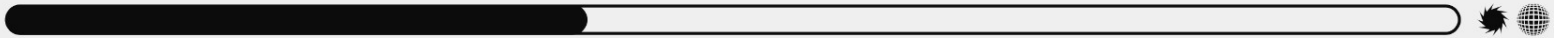
- Each user generates a Ed25519 Public private key pair (Instant)
- Backup account with Mnemonic phrase BIP-39 (Portability)
- Session ID

Your Session ID

5i4f926o448fg45evkl6h1gc38oi75jy9vd64ou2
3ew654v12y3y02iww54s8p9

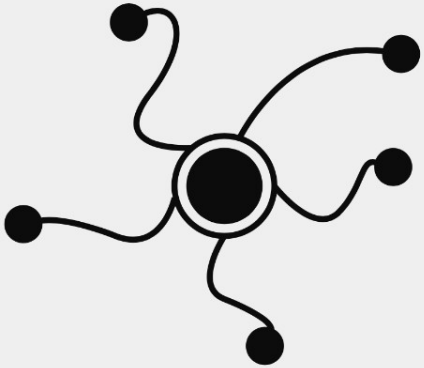
End to End Encryption

- Lib sodium
  - Authentication - crypto_sign()
  - Encryption - crypto_box_sealed()
- Stateless , extremely simple, verifiable
- Suited for decentralised network

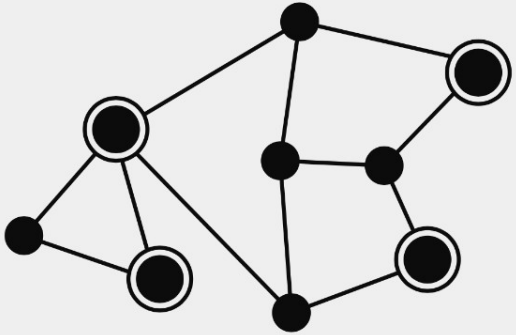


Decentralisation

Decentralising the messenger infrastructure helps solve several issues including trust on first use, central points of failure, and passive surveillance.



Centralised



Decentralised

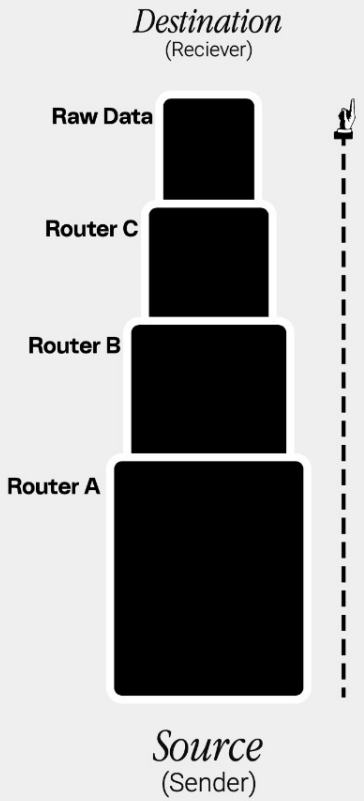
Onion routing

Messages sent over Session are onion routed through a decentralised network of servers.

Nobody (such as your ISP or government) can deduce who you are sending messages to using metadata

The network itself doesn't know who you are sending or receiving messages from

Passive surveillance is more difficult/limited

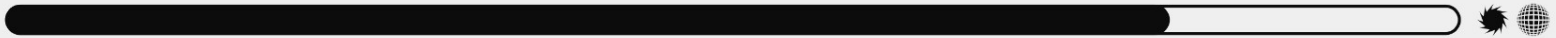


Access & usability

- Open source
- Localisation
- Emulate existing popular application UX
- Remove most confusing options
- Simplicity

Future challenges

- **Replacing Onion Requests with stream based Onion Router (Lokinet)**
  - Onion routed Voice calling
  - Unlimited P2P file transfers
  - Larger attachments
  - P2P messaging
- **Usability enhancers**
  - Stickers & Reacts
  - Performance improvements
- **Technical testing and feedback**
  - Collecting feedback in a privacy respectful manner
  - Understanding what features users interact with or don't interact with



Other uses for The Session protocol

- Instant onboarding
- 2FA
- Anonymous file transfer
- etc ...

Q&A

Message me on Session:
05d871fc80ca007eed9b2f4df72853e2a
2d5465a92fcb1889fb5c84aa2833b3b40

GitHub:
https://github.com/oxen-io