End-to-End Secure Messaging

Trevor Perrin

Stanford Security Seminar

Some projects

- TextSecure
  - Text messaging for smartphones
  - Moxie Marlinspike (Open Whisper Systems)
  - https://whispersystems.org/
- Pond
  - Email-like messaging that resists traffic analysis
  - Adam Langley
  - https://pond.imperialviolet.org/

General Approach

- Message protocols (PGP, S/MIME)
  - Asynchronous, long-lived conversations
  - Problems: Conversation integrity, forward secrecy, deniability
- Session protocols (OTR, SSL, SSH)
  - Synchronous, short-lived sessions
- Blend (TextSecure, Pond)
  - Asynchronous, long-lived sessions

General Approach - infrastructure

- Use where needed:
  - Contact discovery
  - Mailbox servers
  - Posting async handshake messages
  - Anonymity networks
  - Transparency logs

- But don’t trust it

Problems

- Basic
  - Contact and Key Discovery
  - Authentication
  - Handshaking
  - Forward-secrecy ratcheting
- Advanced
  - Unobservability
  - Multi-party
  - Multi-device

Contact and Key Discovery

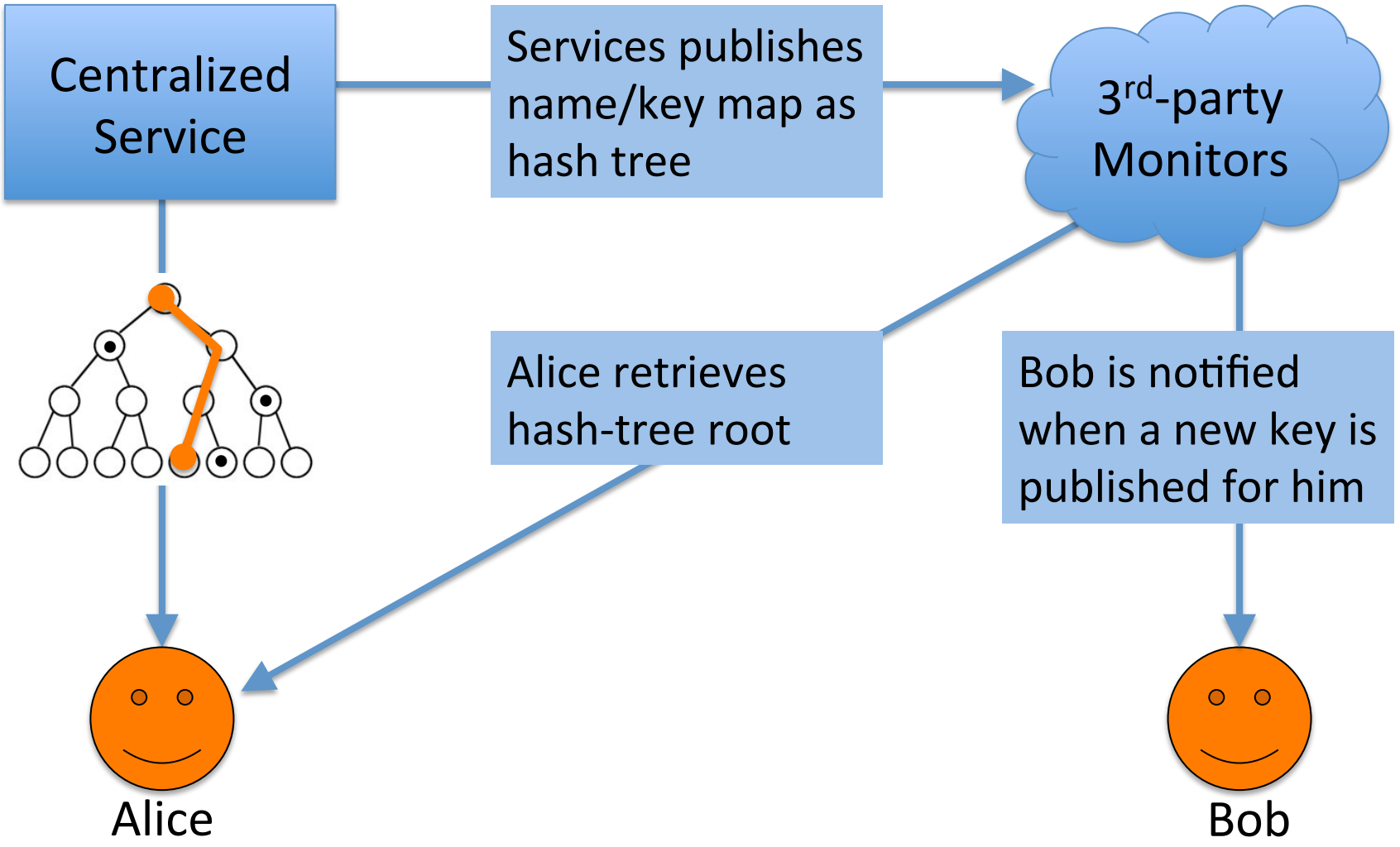
- Manual
  - "Hi, I use PGP, here's my key"
- Signalling
  - Email headers, whitespace tagging, etc.
  - Hard to integrate; gives incomplete view
- Server lookup
  - "Here's all my contacts, who can I encrypt to?"
  - Private Information Retrieval?

Authentication

- Key continuity
  - E.g. TOFU, then warn on changes
- Key fingerprints
  - QR codes work well; other encodings need study

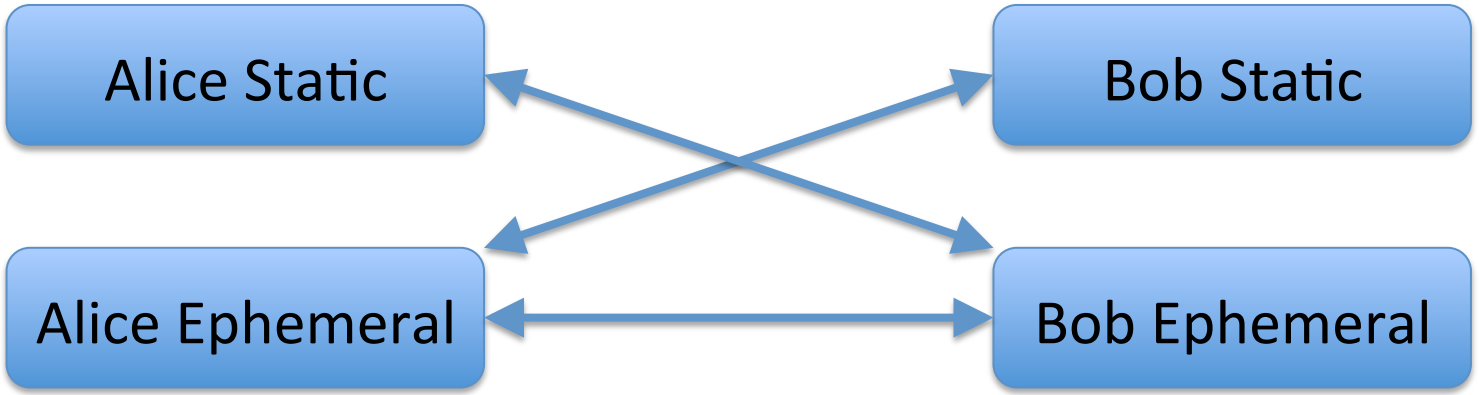
    (hex vs base32 vs words vs sentences vs images…)
- Short Auth Strings? Shared Knowledge Qs? Certificate Transparency?

Certificate Transparency (adapted)



Handshaking

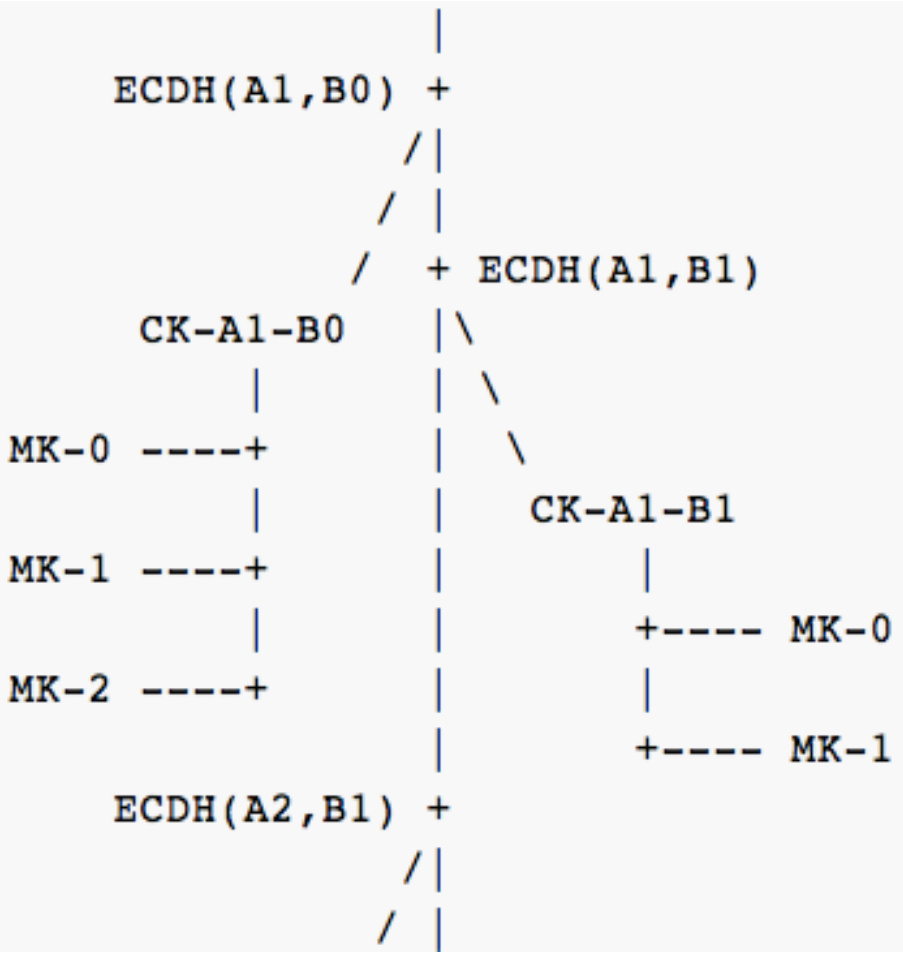
- Publish ephemeral “pre-keys”
- Alice can send after fetching Bob’s pre-key
- Needs async-friendly, deniable key agreement
  - E.g. “TripleDH”:



Forward Secrecy Ratcheting

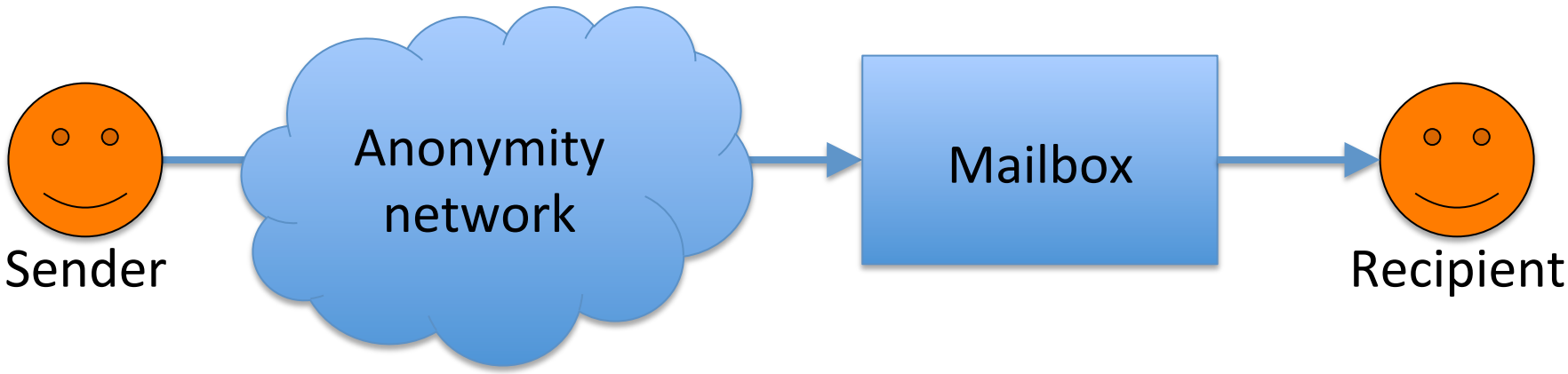
- Symmetric-key ratchet
  - Replace key after processing each message (e.g. SCIMP)
  - Secure deletion?
- DH ratchet
  - Replace key on exchange of new DH values (e.g. OTR)
- "Axolotl" ratchet
  - Combines symmetric + DH ratchet
  - Supports out-of-order messages, and header encryption (e.g. Pond)

Axolotl Ratchet



Advanced Topics

Unobservable message transport

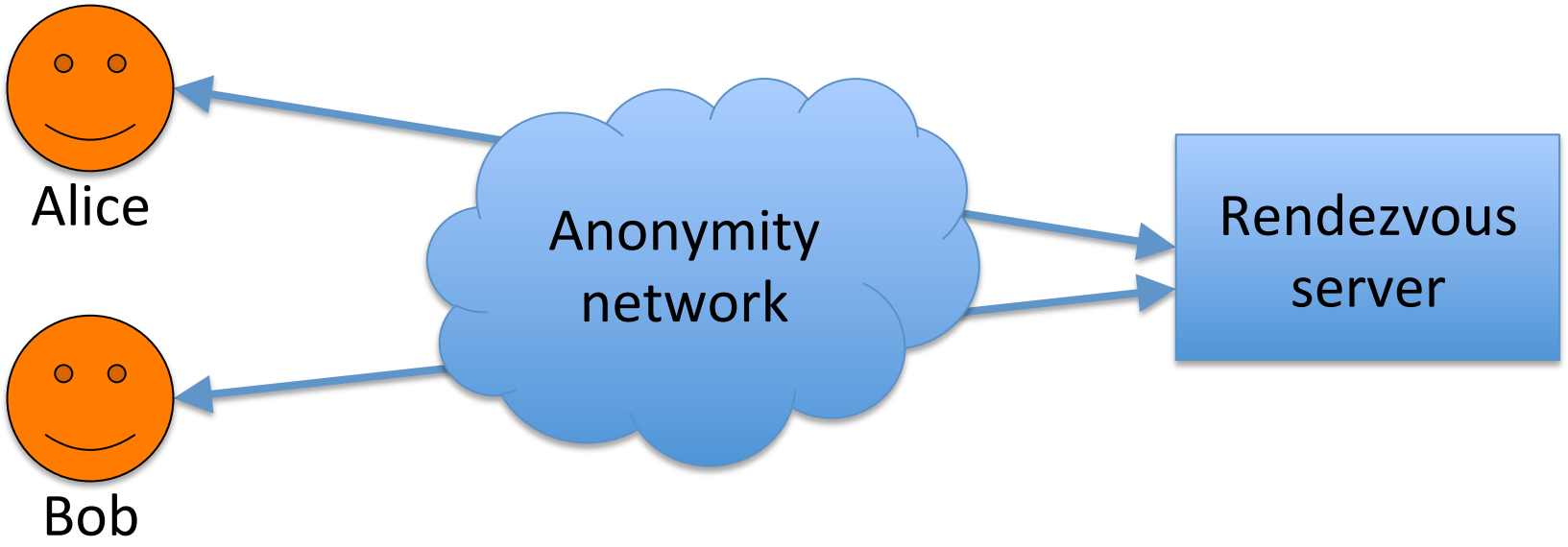


- Mailbox authenticates messages but can't distinguish senders
- Recipient can recognize and revoke senders
- Group sigs vs. one-time signing keys

Unobservable bootstrapping

- How to exchange message-transport secrets?
- Use conventional messaging
  - Reveals who, but not when / how much
- Face-to-face (Bluetooth, QR codes)
- Online rendezvous based on shared secret

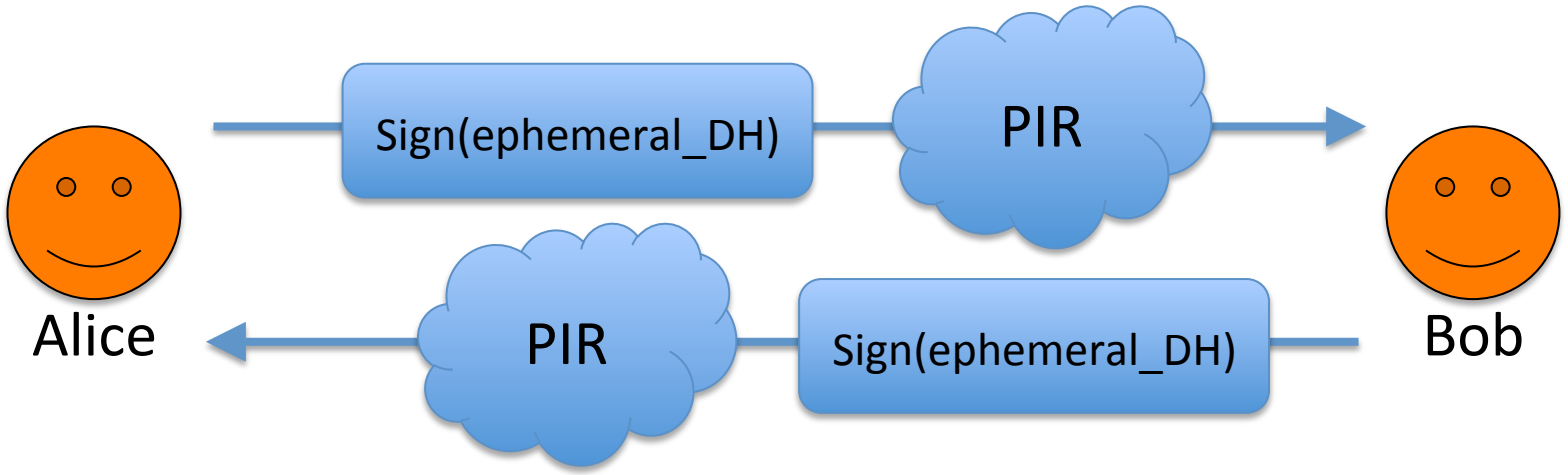
Unobservable online rendezvous



- Alice and Bob share a secret, use to derive meeting slot at rendezvous server ("PANDA")

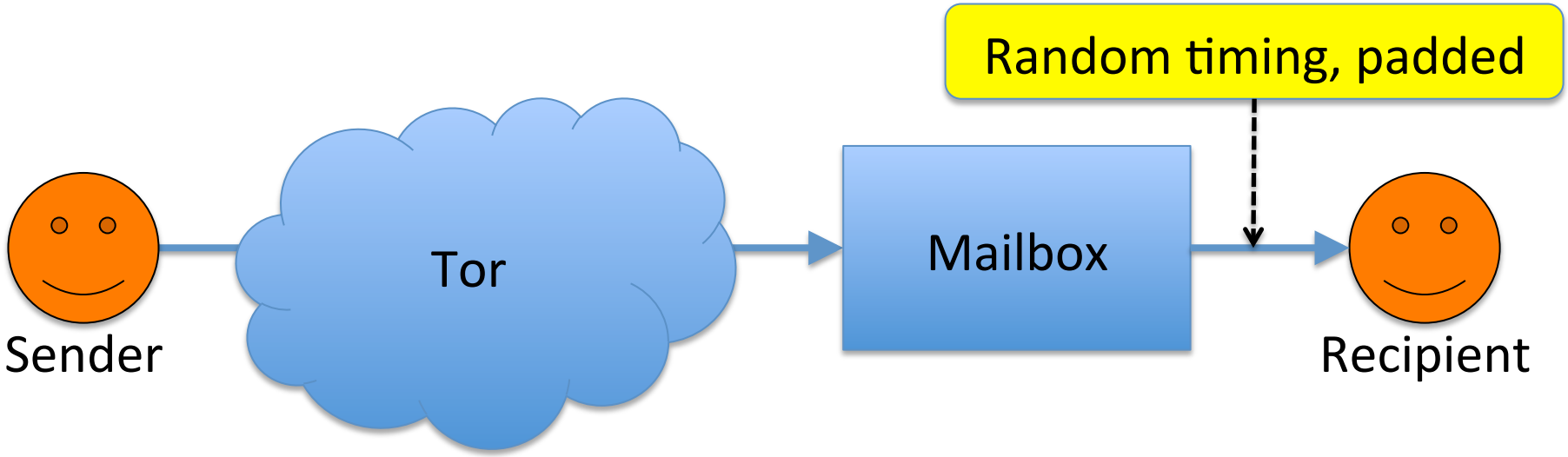
- Attacker could observe parties sending rendezvous traffic; perhaps mask w/dummy traffic?

Online rendezvous via fingerprints?



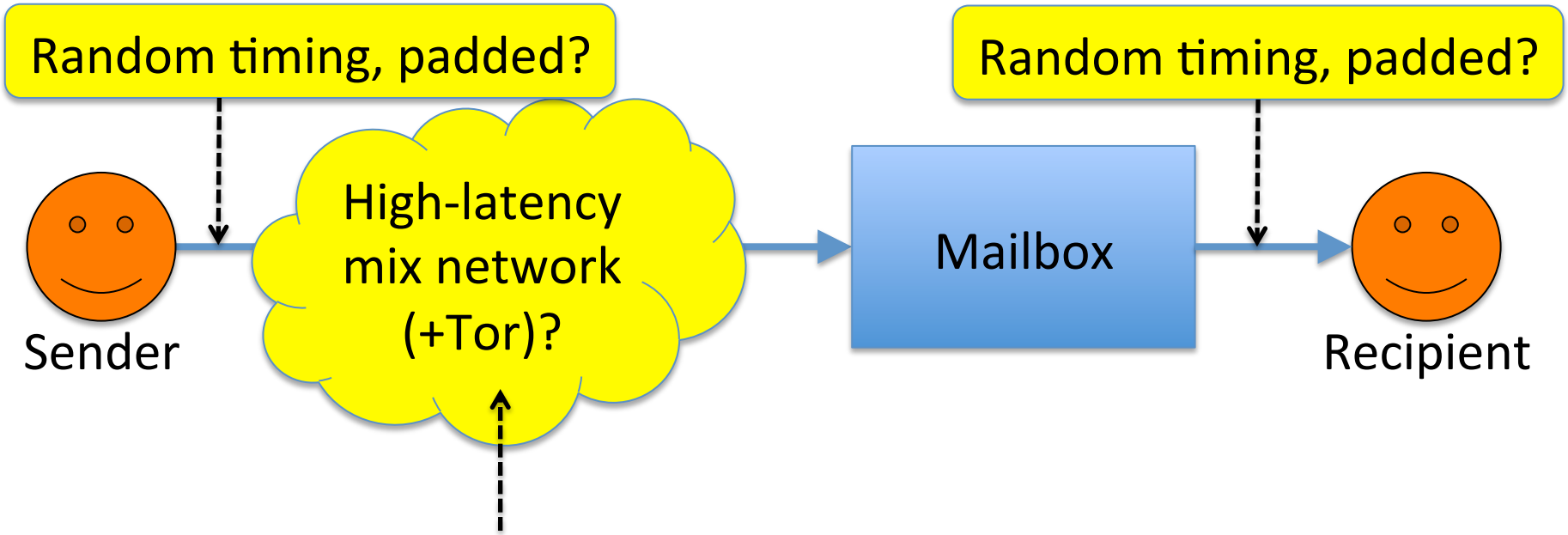
- Exchange fingerprints instead of secrets
- Use fingerprints to lookup short-lived DH keys

Anonymity net = Tor (Pond)



- Tor vulnerable to in/out correlation
- Sender/recipient correlation broken at mailbox/ recipient end
- Sender/mailbox correlation remains (only ~1 mailbox server at present)

Anonymity net = high-latency mix??



- Breaks sender/mailbox correlation
- Traffic-flow measures at sender and receiver could mask send/receive volume

Multi-party: key agreement

- Group Key Agreement + Signatures
  - mpOTR = deniable signing keys
  - More handshaking; smaller messages
- Pair-wise
  - Less handshaking; larger messages; better ratcheting
- Answer may depend on context
  - Broadcast? More bandwidth up than down? Mailbox servers?

Multi-party: new attacks

- Different messages could be sent to different recipients
- Messages could be re-ordered to change their context
- Messages could be deleted or delayed
- **Result:** Messages (or their absence!) misunderstood due to manipulated context

Multi-party: transcript consistency

- Messages could declare their "causal predecessors" and a hash over them

- Lots of details:
  - Displaying partially-ordered messages?
  - Detecting silenced users / delayed messages?
  - Handling join / leave?
  - What amount of delay / reordering is tolerable?

Multi-device

- Build on multi-party, treating each device as separate party
- Reveals number of devices and when they're being used
- Alternatively, sync ratchet between devices?

Thanks!

- These projects (and others!) need your help
- Lots of ways to participate:
  - https://github.com/whispersystems/textsecure/
  - https://github.com/agl/pond
- Messaging mailing list
  - https://moderncrypto.org