Mixminion

a best-of-breed anonymous remailer

(systems track)

Nick Mathewson, Roger Dingledine
The Free Haven Project
{nickm,arma}@freehaven.net

Scope

- Introduction to anonymity
- How we got started
- Introduction to mix-nets
- Contributions
- Lessons learned
- Future work

Anonymity: The idea

Untraceability: hide connection between senders and recipients.

Unlinkability: hide connection between actions by the same sender.

A.K.A. Relationship privacy, traffic-analysis resistance, "security"

Sender vs Recipient anonymity
high-latency vs low-latency systems

Who needs it?

- Private citizens (advocacy, counseling, whistleblowing, reporting, ...)
- Higher-level protocols (voting, e-cash, auctions)
- Government applications (research, law enforcement)
- Business applications (hide relationships and volumes of communication)
  - Is the CEO talking to a buyout partner?
  - Who are your suppliers and customers?
  - Which groups are talking to patent lawyers?
  - Who is visiting job sites?

Project origins

- Let's try implementing our research!
- Why not use deployed mix-networks?
- State of deployed mix-networks: bad! (2001)
  Two incompatible systems, no full specification, known flaws, ugly code.
- The Mixminion project
  - Designs (2003), specifications (2003), software (ongoing)

Mixminion's goals

- Resolve flaws of earlier deployed remailers.
- Conservative design (minimize new design work needed)
- Support testing of future research
- Design for deployment; deploy for use.

Motivation:
The importance of adoption

Anonymity systems rely on network effects more than do other cryptographic systems:

- No users, no anonymity.
- "Safer in the coach seats than riding first class." (?)
- Can't assume a userbase of cryptographers

Consequences

- Software required only for anonymous users: *must* support clear-text delivery
- Must subsume function of earlier systems
- Must work in real-world internet (unsynchronized, unreliable)
- Entire system must be designed, specified

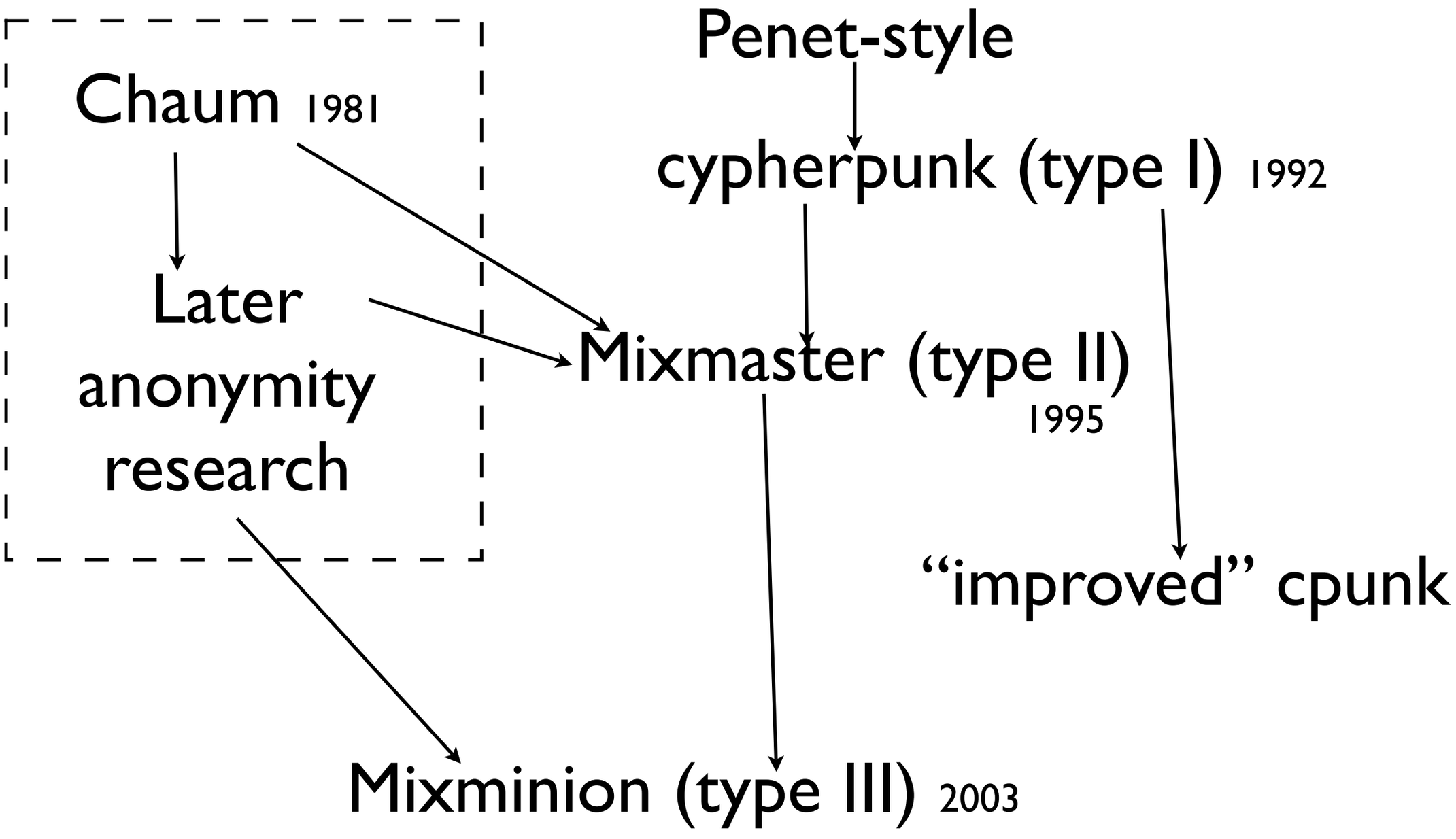
Consequences: Threat model

(Choose for reality, not for security proofs.)

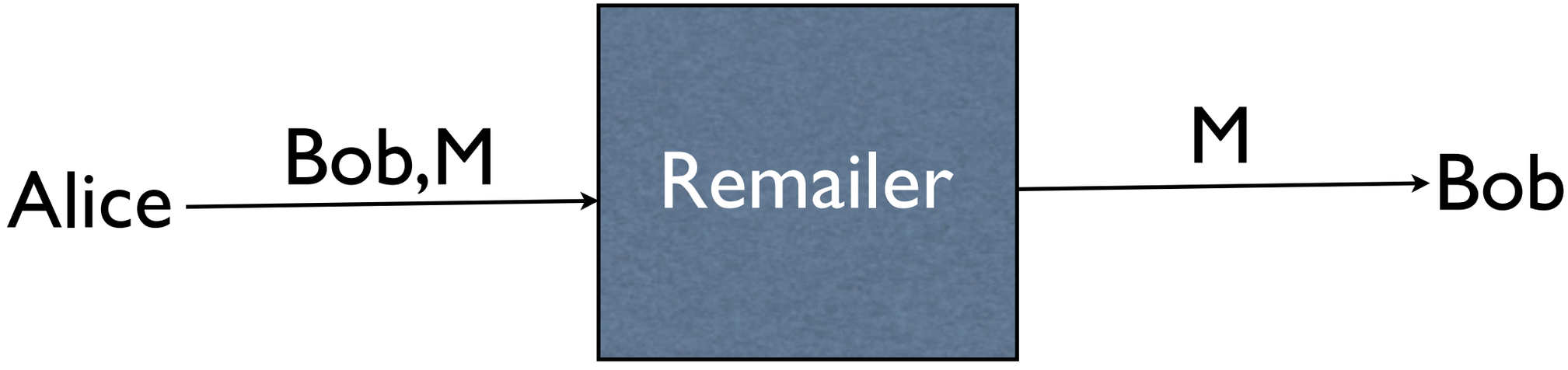
- Global observer: can see all net traffic
- Runs a fraction of the servers on the network
- Can generate or delay traffic

Weak attackers are stopped;
Strong attackers are only delayed.

Deployed remailer systems

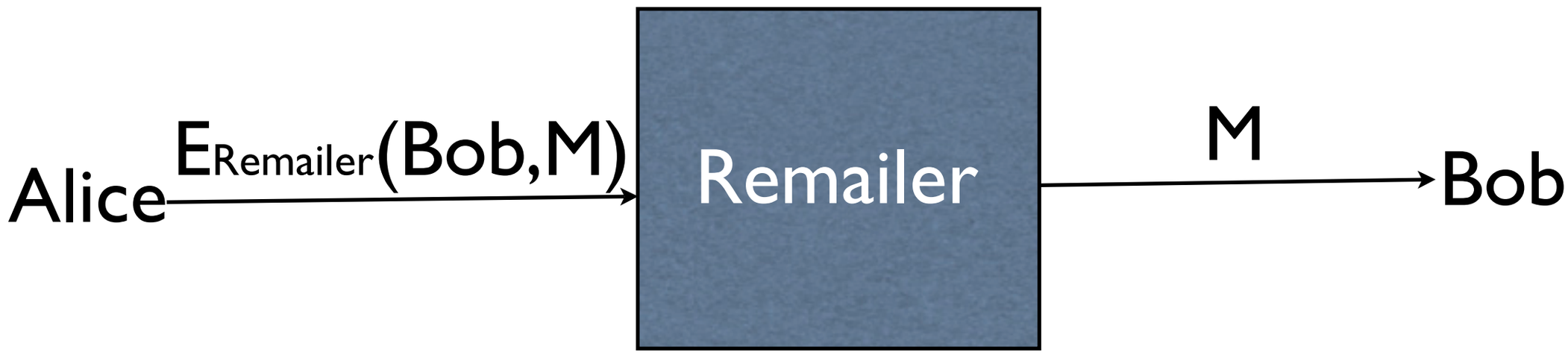


Direct Remailer

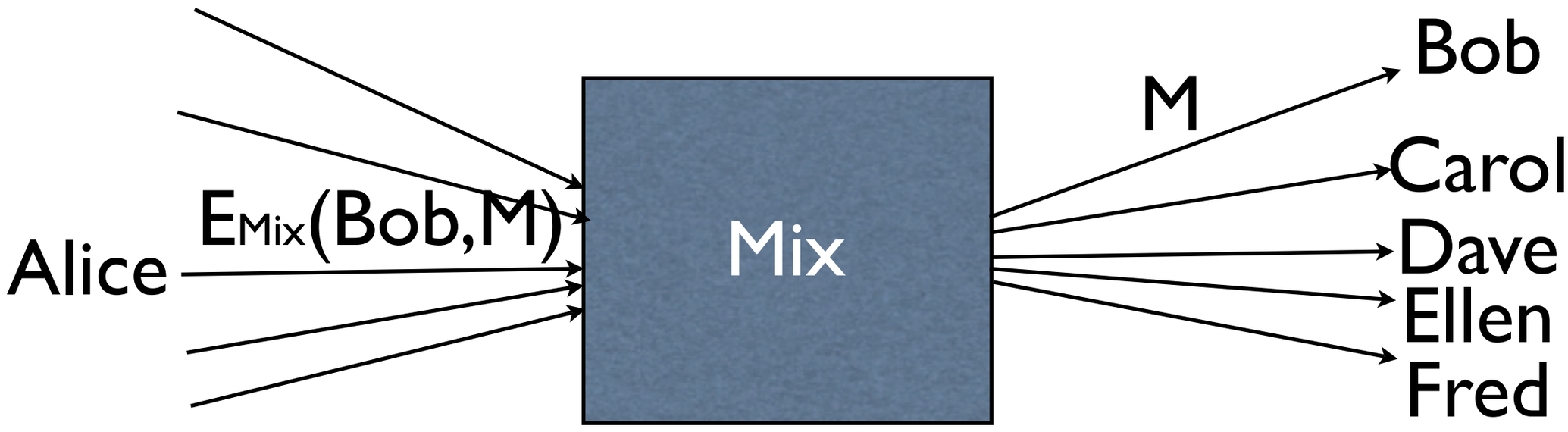


example: anon.penet.fi

Add Encryption

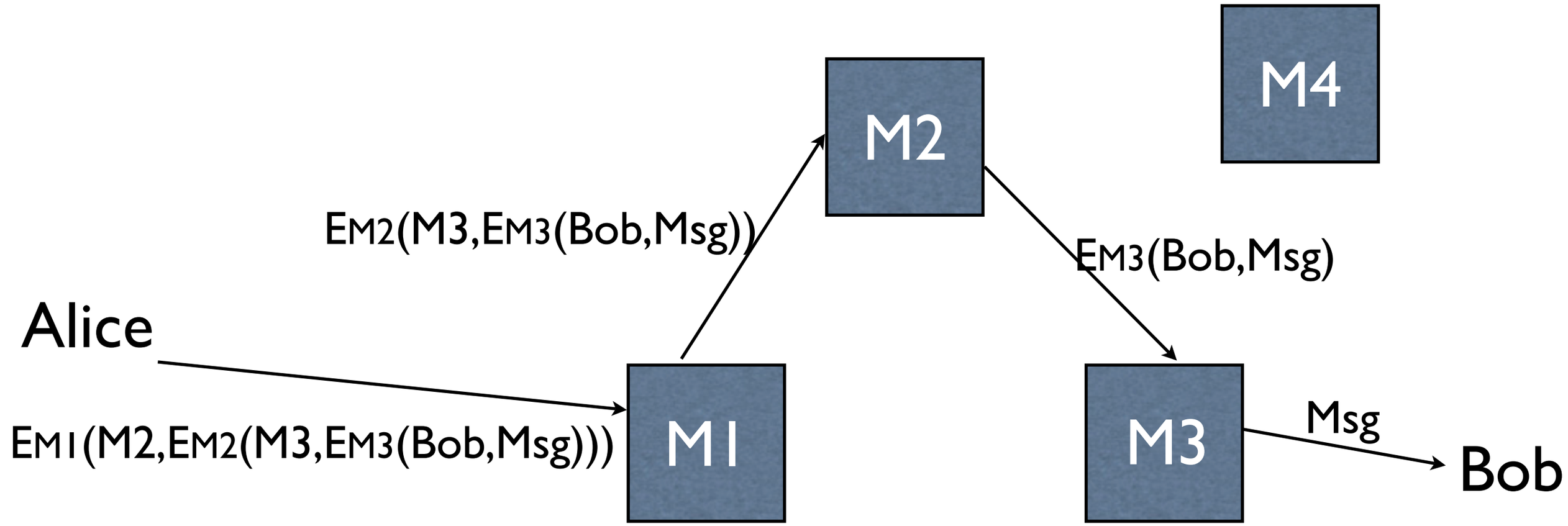


Batch and re-order



(Chaum, 1981)

Mix-nets



(Chaum, 1981)

Flaws of earlier systems (I)

Mixmaster (type II)

- Out-of-band *ad hoc* directories; users partitionable by directory choice.
- Optional link encryption
- Bad code and partial spec (but not any more)
- **No recipient anonymity:** nym users fall back on cpunk

Flaws of earlier systems (II)

“Cypherpunk” (type I)

All the problems of Mixmaster, plus...

- Non-uniform message length
- Distinguishable user options
- Vulnerable to replay attacks
- Reply blocks vulnerable to flooding attacks
- Batching and delaying are optional

And so much, much more

Why are replies hard?

Seemingly:

- Forward messages need integrity checking on routing and payload at each hop
- Replies *can't have* integrity checking on payload at each hop

Must forward and reply messages be distinguishable?

Contributions (I)

- Secure reply blocks
  - Single-use, replay-proof
  - Replies indistinguishable from fwd messages except at recipient

solution: use the LIONESS large-block SPRP construction to ensure that modified data is completely unrecoverable; use two headers with hashes for each; do a Feistel-like step when exchanging headers.

Contributions (II)

- Integrated directory service
  - Enables key rotation (takes months with older systems)
  - Specified, extensible discovery of server capabilities and reliability
  - Coordinate multiple directories

Contributions (III)

- Uniform forward-secure message transfer protocol.
- Simple dummy-traffic policy

Status

- First release: Dec 2002
- First usable release: Jan 2003
- Design published, specification online
- Implementation in progress (35 kloc)
- Now: 29 servers; 12 exits. (each handles ~400 packets per day; most are pings.)

Lessons (I)

- Implementation can drive research:
  - uncovers specification gaps
    (reply recognition)
  - suggests new design problems
    (directory agreement problem)
  - exposes potential security holes
    (retry timing)

Lessons (II)

- Theoretical security is not the whole story
  - With carefully defined transport, network, users, and attackers, we can win in theory...
  - but to win in practice, we must frustrate a real adversary in the real world, even if they would win eventually in theory.

Future work

- Usability and clients
- Directory coordination
- DOS limitation
- Pseudonym service

For more information,
see Mixminion design paper

Mixminion:
Design of a Type III Anonymous Remailer Protocol
(Danezis, Dingledine and Mathewson, 2003)

What about Spam?

- High-latency mix nets are bad for Spam
  - Comparatively high CPU requirements
  - Latency variability makes blocking easy
  - Still need to receive funds nymously
- The real problem is abuse
  - Only one msg needed to annoy a newsgroup
  - Block at users request
  - Support for automatic blocking