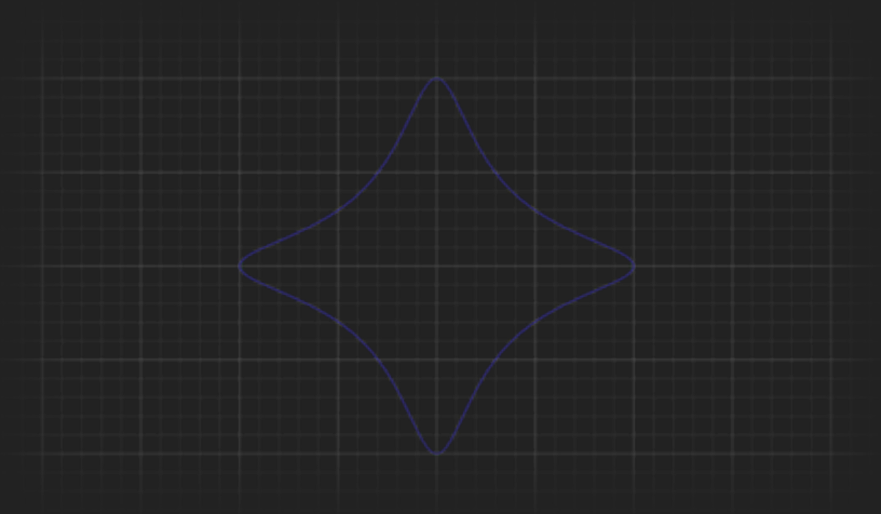
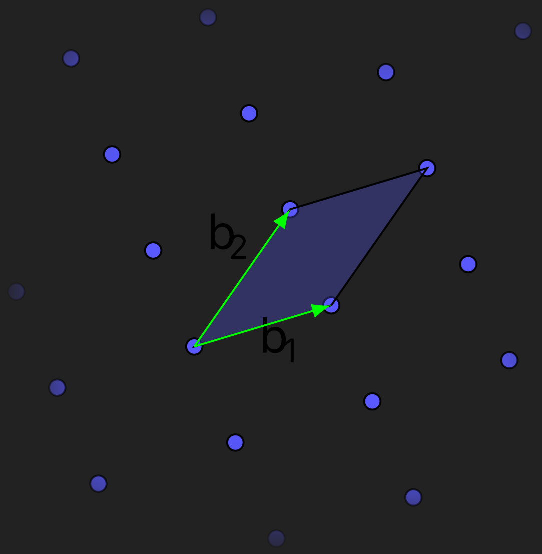
HACK.LU - @FREDERICJACOBS

ADVANCES IN SECURE MESSAGING

@FREDERICJACOBS

ACADEMIA





FOSS

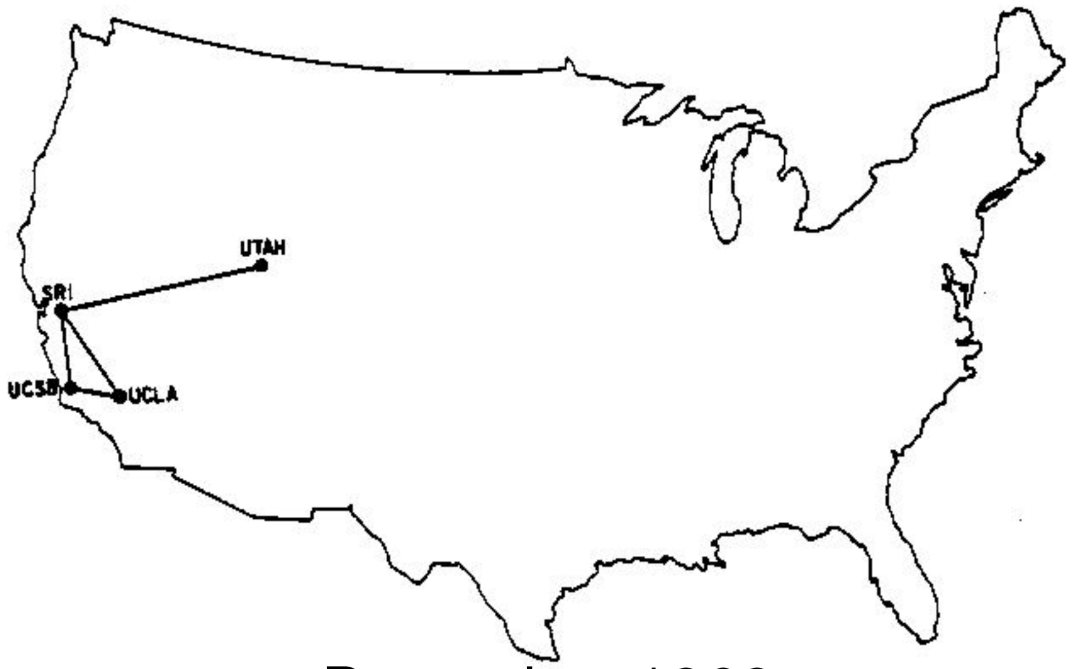
⚠️

TALK ABOUT PROTOCOLS, NOT PRIMITIVES

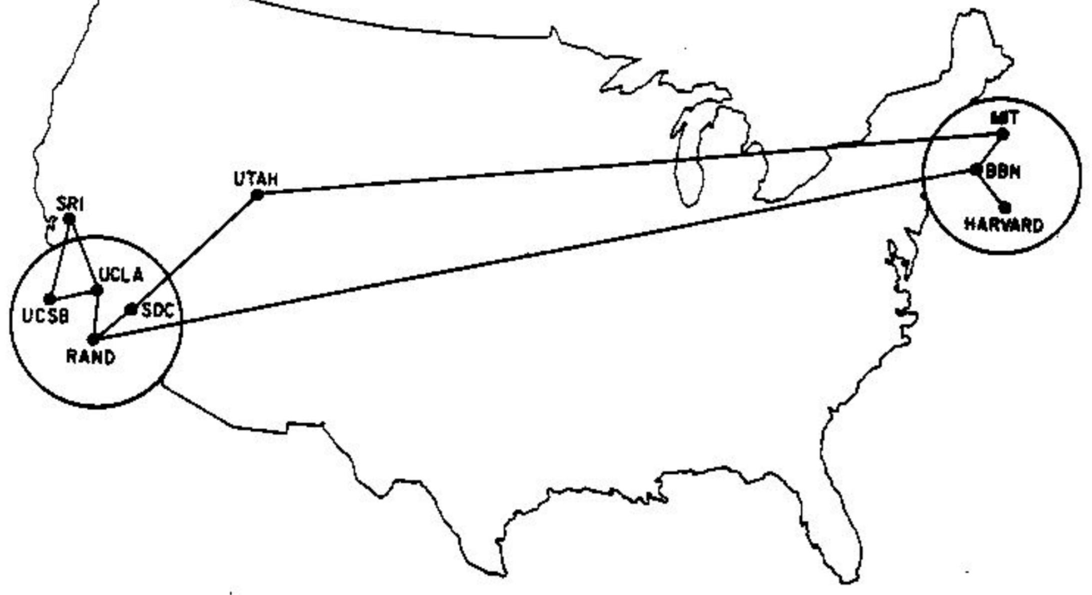
DEFINING THE CONTEXT FOR

MESSAGING

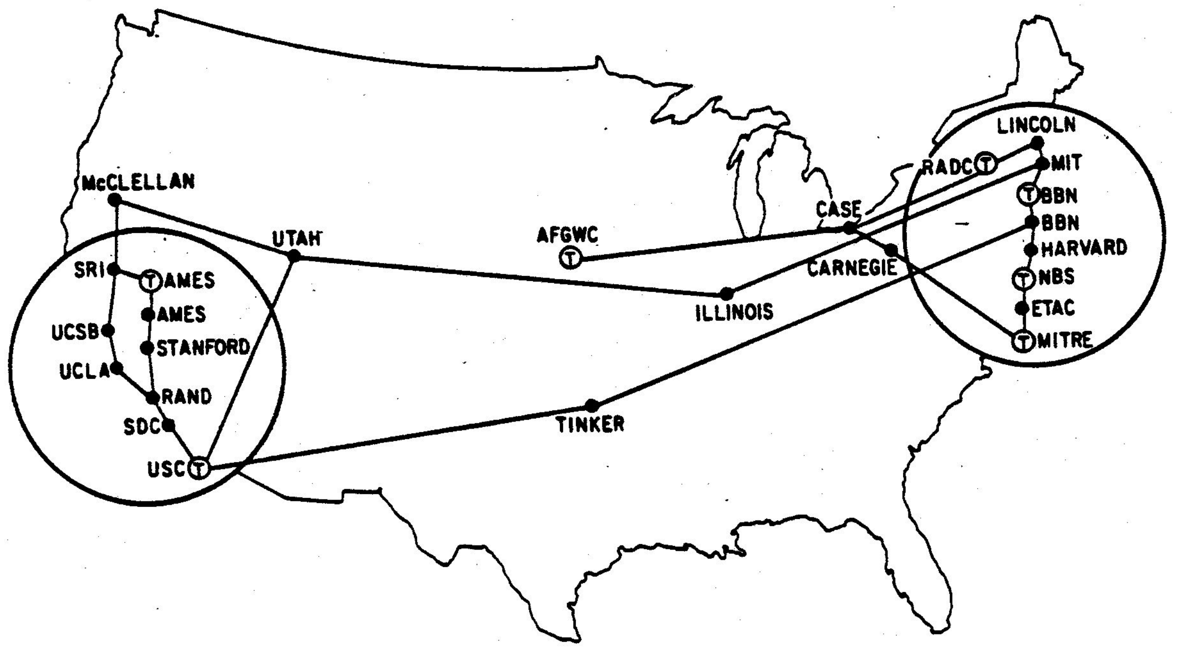
ARPANET



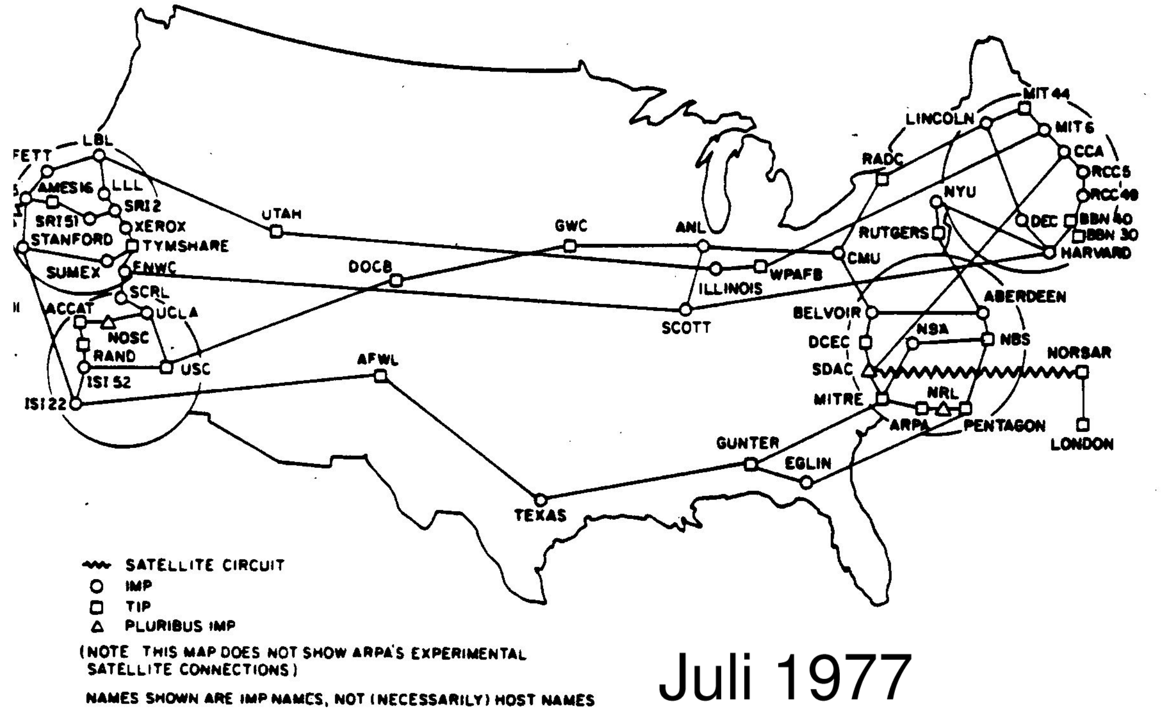
Dezember 1969



Juni 1970

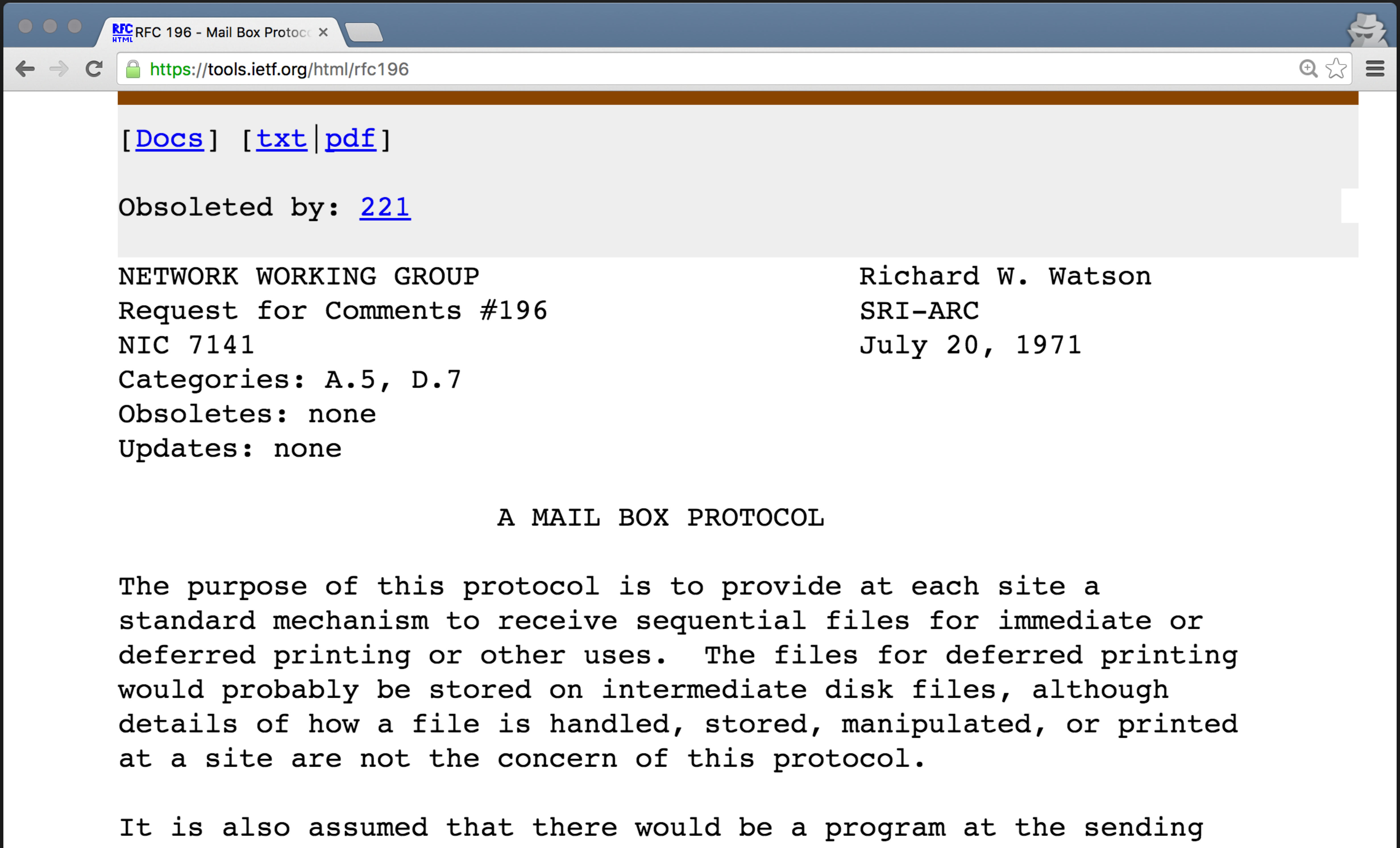


März 1972



Juli 1977

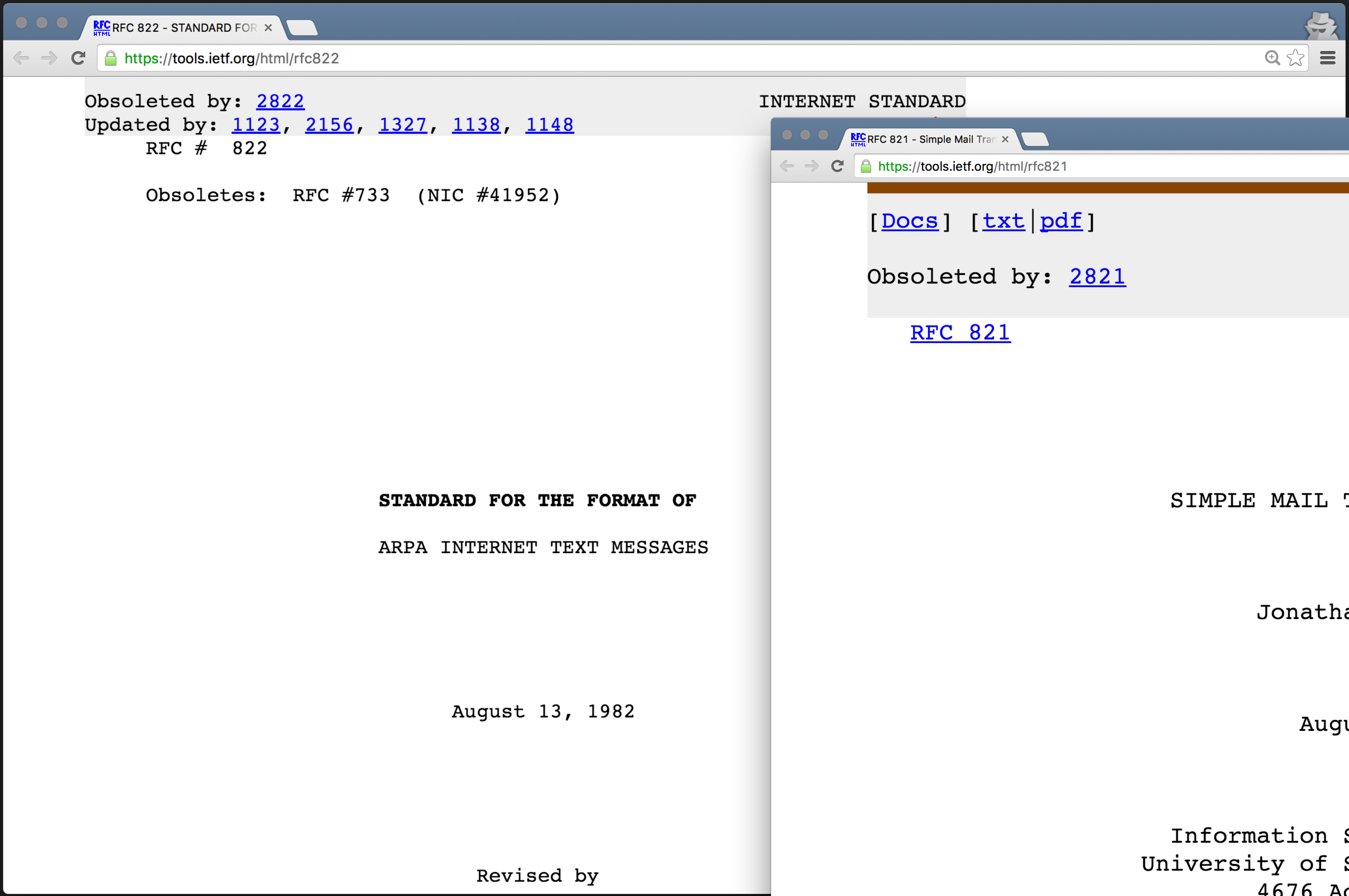
1971: A MAIL BOX PROTOCOL


RFC 196 - Mail Box Protoc

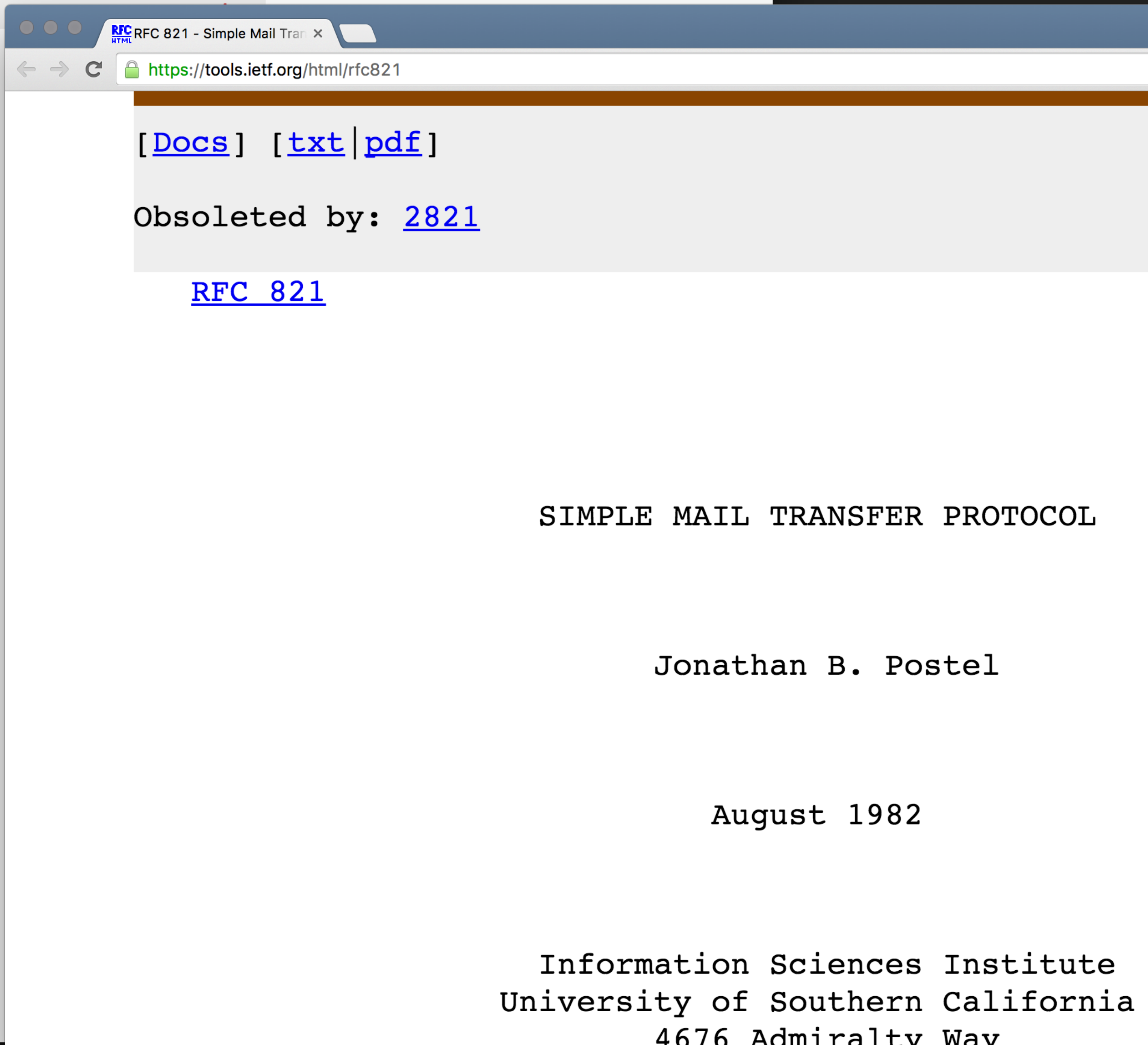
https://tools.ietf.org/html/rfc196

```
[Docs] [txt|pdf]

Obsoleted by: 221

NETWORK WORKING GROUP                        Richard W. Watson
Request for Comments #196                    SRI-ARC
NIC 7141                                     July 20, 1971
Categories: A.5, D.7
Obsoletes: none
Updates: none

                      A MAIL BOX PROTOCOL

The purpose of this protocol is to provide at each site a
standard mechanism to receive sequential files for immediate or
deferred printing or other uses.  The files for deferred printing
would probably be stored on intermediate disk files, although
details of how a file is handled, stored, manipulated, or printed
at a site are not the concern of this protocol.

It is also assumed that there would be a program at the sending
```

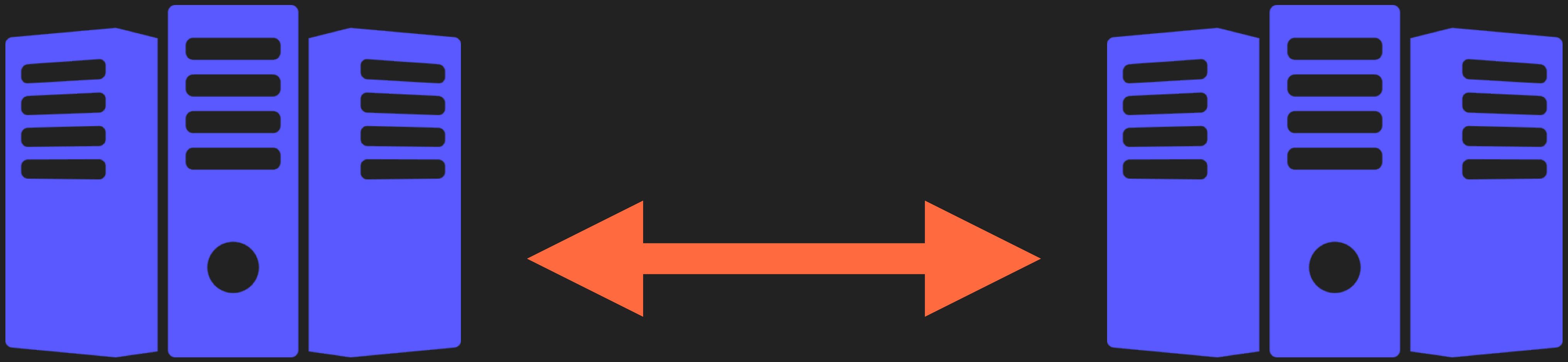
1982: SMTP SPEC IS OUT



```
Obsoleted by: 2822                                    INTERNET STANDARD
Updated by: 1123, 2156, 1327, 1138, 1148
     RFC #  822

     Obsoletes:  RFC #733  (NIC #41952)









                        STANDARD FOR THE FORMAT OF

                        ARPA INTERNET TEXT MESSAGES






                              August 13, 1982





                                Revised by
```



```
Obsoleted by: 2821

   RFC 821







                          SIMPLE MAIL TRANSFER PROTOCOL



                              Jonathan B. Postel



                                 August 1982



                          Information Sciences Institute
                        University of Southern California
                              4676 Admiralty Way
```



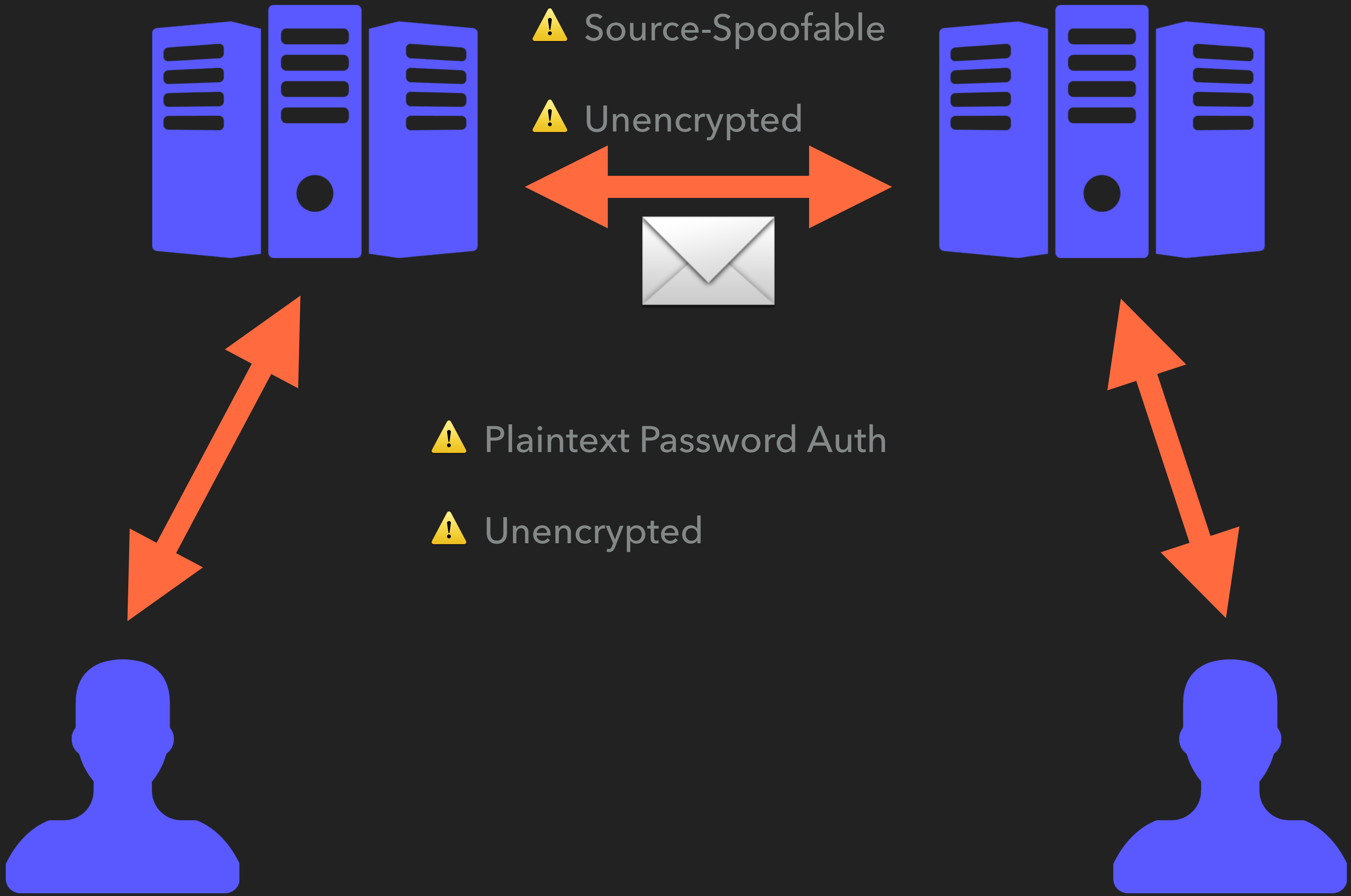
⚠️ Source-Spoofable

⚠️ Unencrypted

EMAIL RETRIEVAL PROTOCOL SECURITY

- 1984: Post Office Protocol (POP) allows remote email retrieval.
  - Plaintext information retrieval
  - **Plaintext** password authentication over **plaintext** network protocol

1984: POP


⚠️ Source-Spoofable
⚠️ Unencrypted
⚠️ Plaintext Password Auth
⚠️ Unencrypted

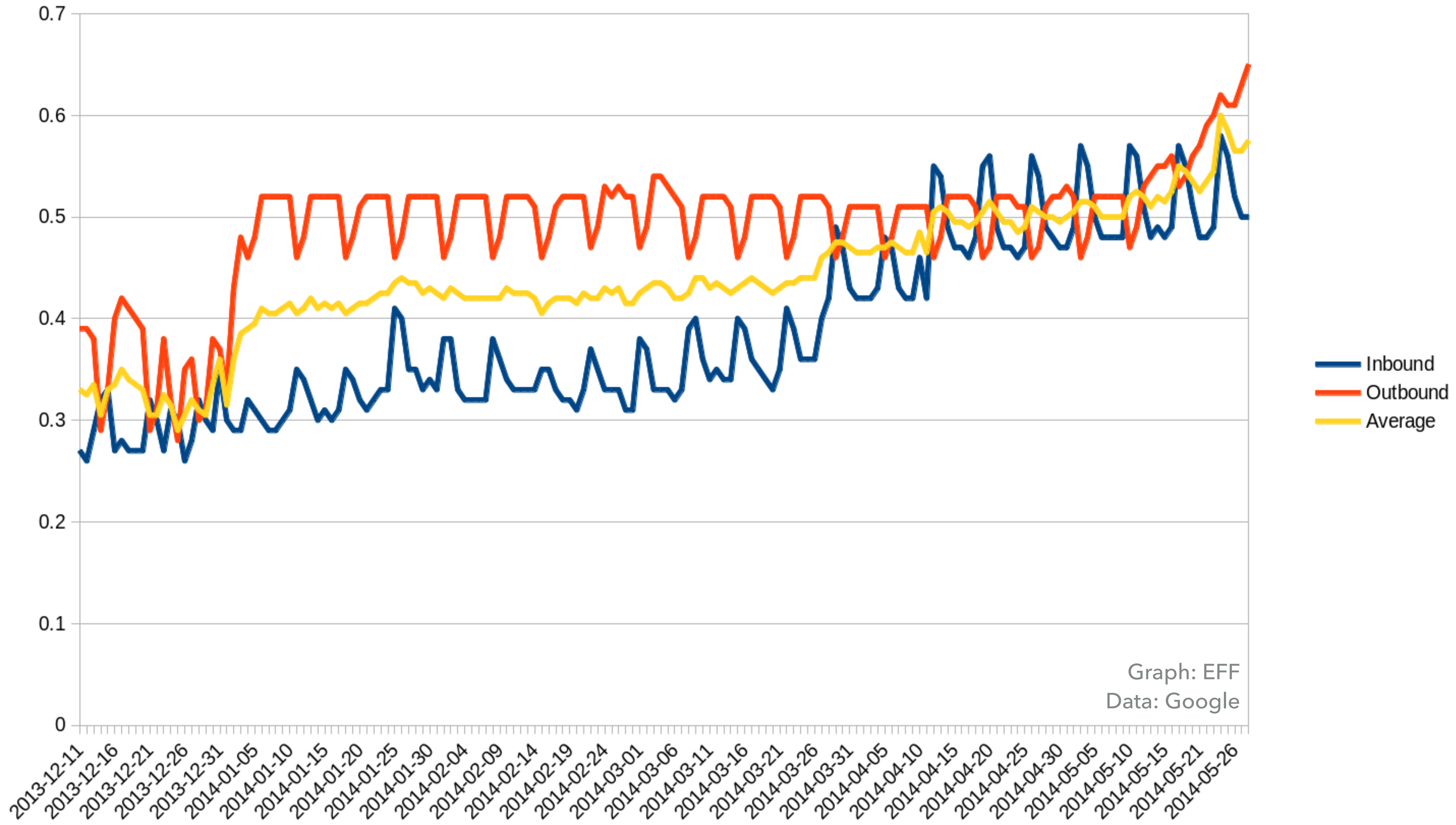
1991: PGP

- End-to-End Cryptography predates standardized transport security and non-plaintext auth for email protocols.
- Same year, IMAP v3 comes out. Still plaintext auth.

DEEP PROTOCOL INSECURITY ADDRESSED AFTER PGP

- 1994: OTP and Kerberos support in IMAP/POP
- 1995: Authentication for SMTP, SSLv2 is released
- 1997: SMTPS is standardised.

ADOPTION OF SMTPS



PGP (AND FRIENDS: S/MIME & PEM)

- Works in asynchronous environments
- Lacks forward/future secrecy
- Lacks deniability
- Complicated setup and usage

THE USER EXPERIENCE OF MESSAGING TODAY

- Multi-device
- Group paradigm is growing (Slack, Facebook Groups, WhatsApp Group Chats …)
- Ability to message offline users

MEANWHILE IN SSH WORLD

- Short-lived sessions (ephemeral keys)
- TOFU
- Use of Diffie-Hellmann primitives

OTR

- Forward secrecy via a ratcheting ephemeral key exchange
- Fewer ways to shoot yourself in the foot
- Synchronous
- Single device protocol

MESSAGE PROTOCOLS	SESSION PROTOCOLS
Examples : PGP, S/MIME	Examples: OTR, SSL, SSH
Asynchronous	Synchronous
Lacks: conversation Integrity, forward secrecy, deniability	Short-lived session

Axolotl

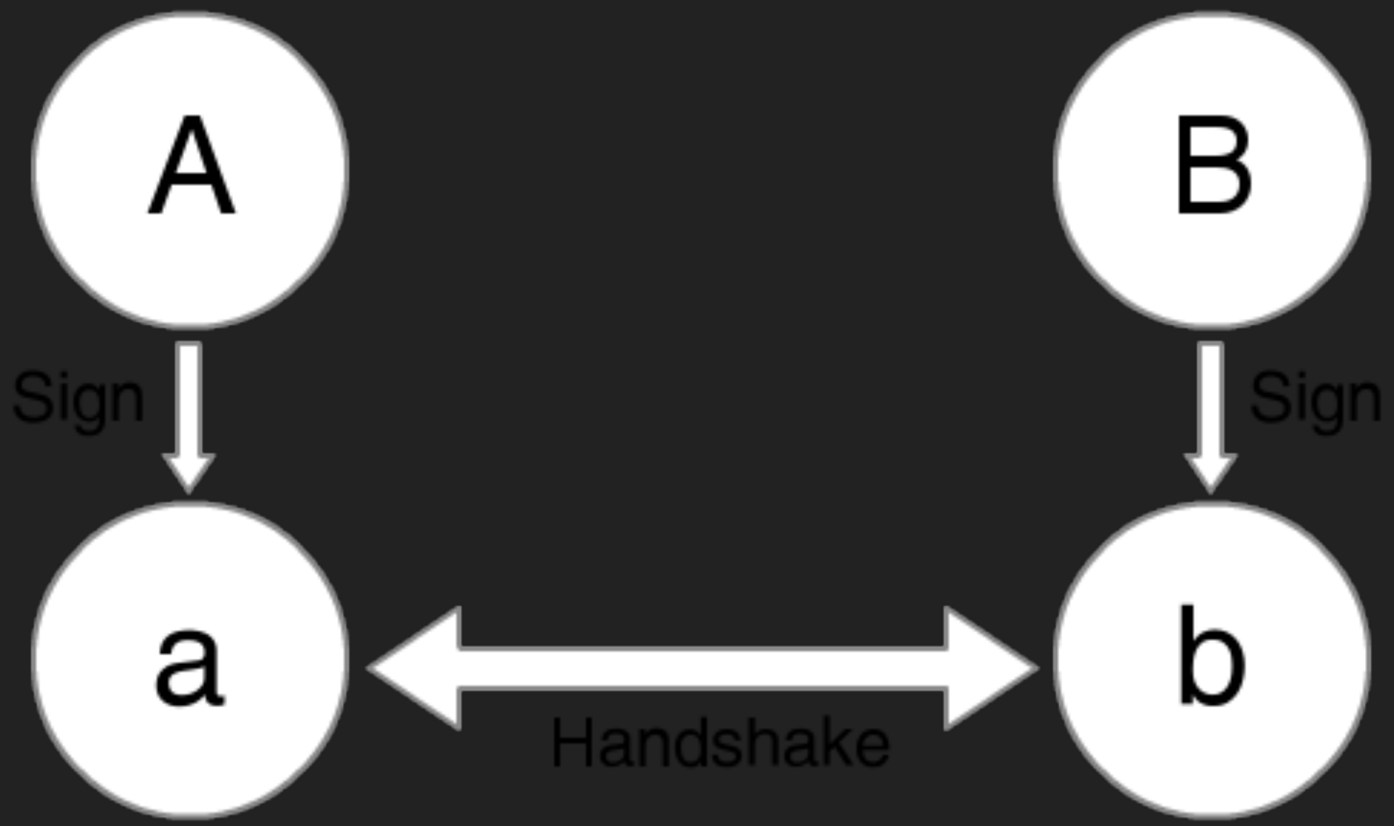
Asynchronous with all great features of short lived protocols

Forward secrecy, deniability, conversation integrity …

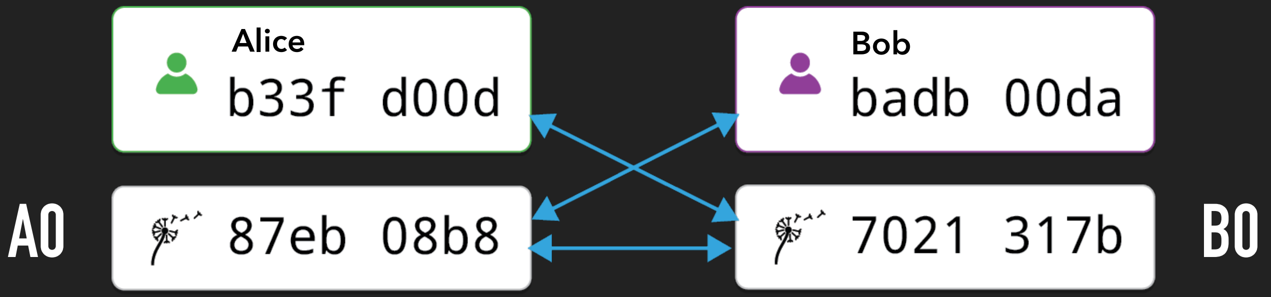
AXOLOTL

DENIABILITY

OTR HANDSHAKE



3DH-KEY EXCHANGE



hash(DH(A,B0 8b4 f9af B,A0) || DH(A0,B0))

🟦 = Diffie-Hellman

CLOSING THE
WINDOW OF COMPROMISE

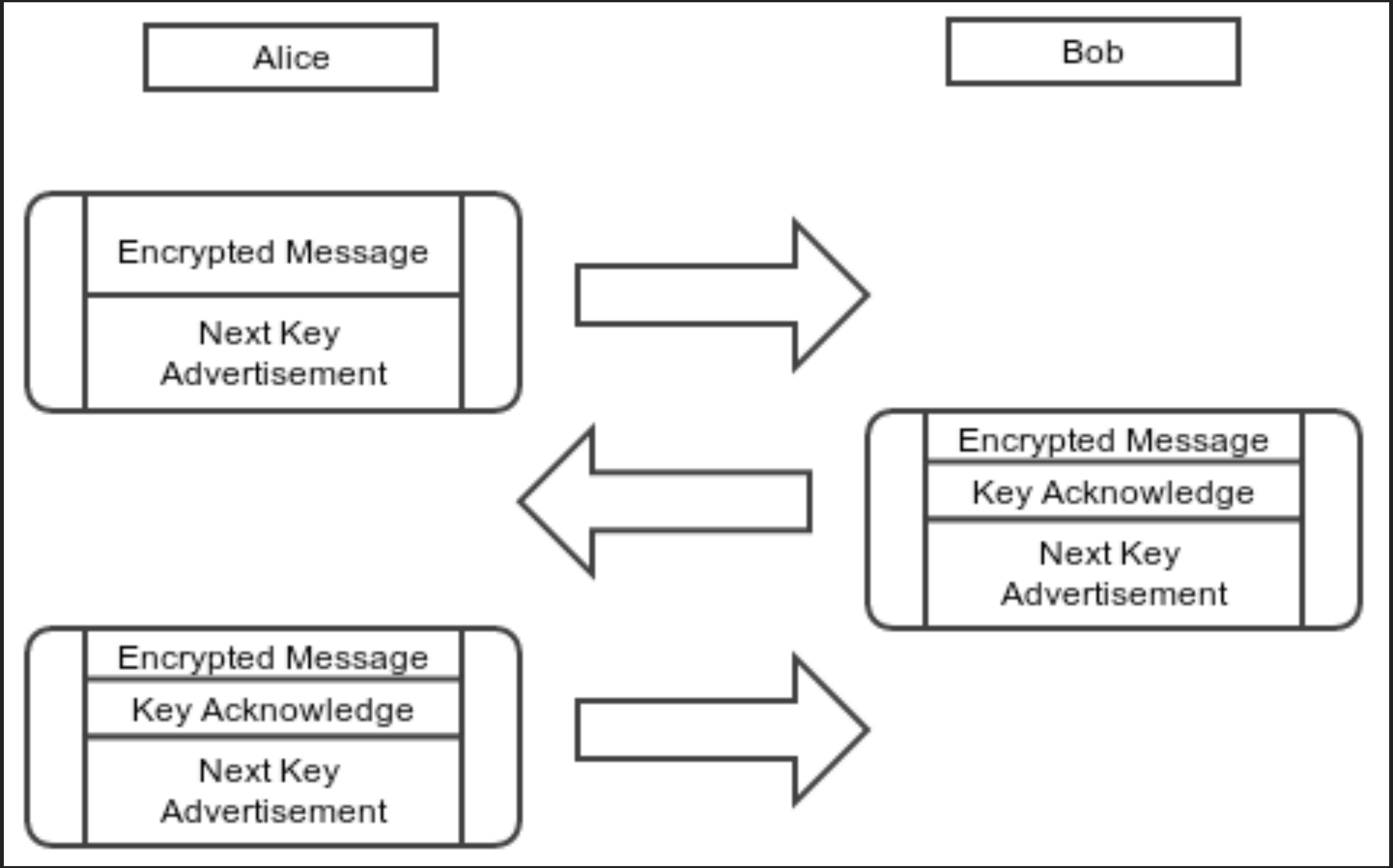
HASH-ITERATED RATCHETS

```
Encrypt(msg1,🔐)
        ↓
Encrypt(msg2, HMAC(🔐))
        ↓
Encrypt(msg3, HMAC(HMAC(🔐)))
```

HASH-ITERATED RATCHETS

- Provides Perfect Forward Secrecy
- Simple implementation, no round trip required
- First important use, the SCIMP protocol by Silent Circle
- Any key compromise will compromise all future messages

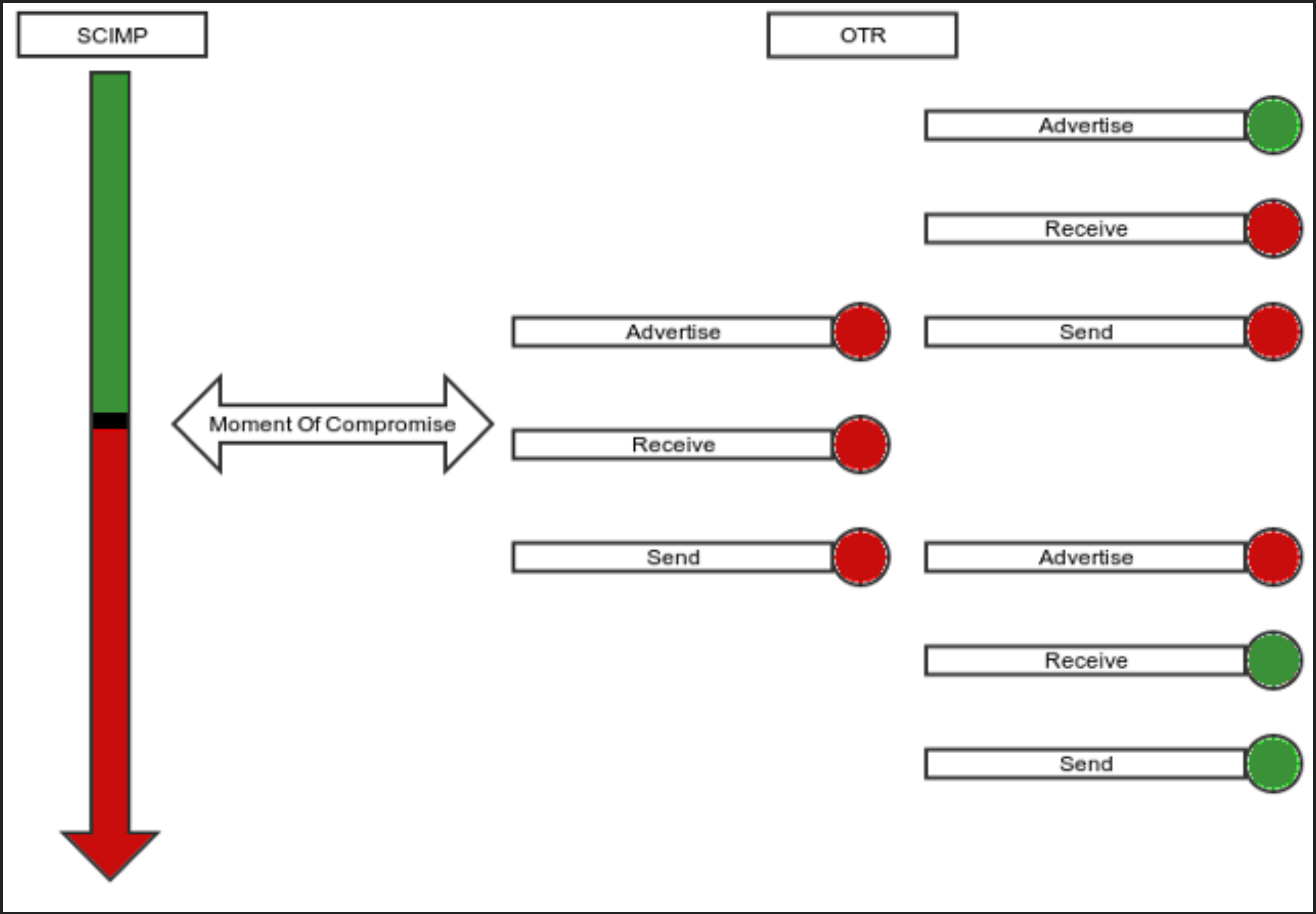
DH RATCHETS


Alice
Bob
Encrypted Message
Next Key Advertisement
Encrypted Message
Key Acknowledge
Next Key Advertisement
Encrypted Message
Key Acknowledge
Next Key Advertisement

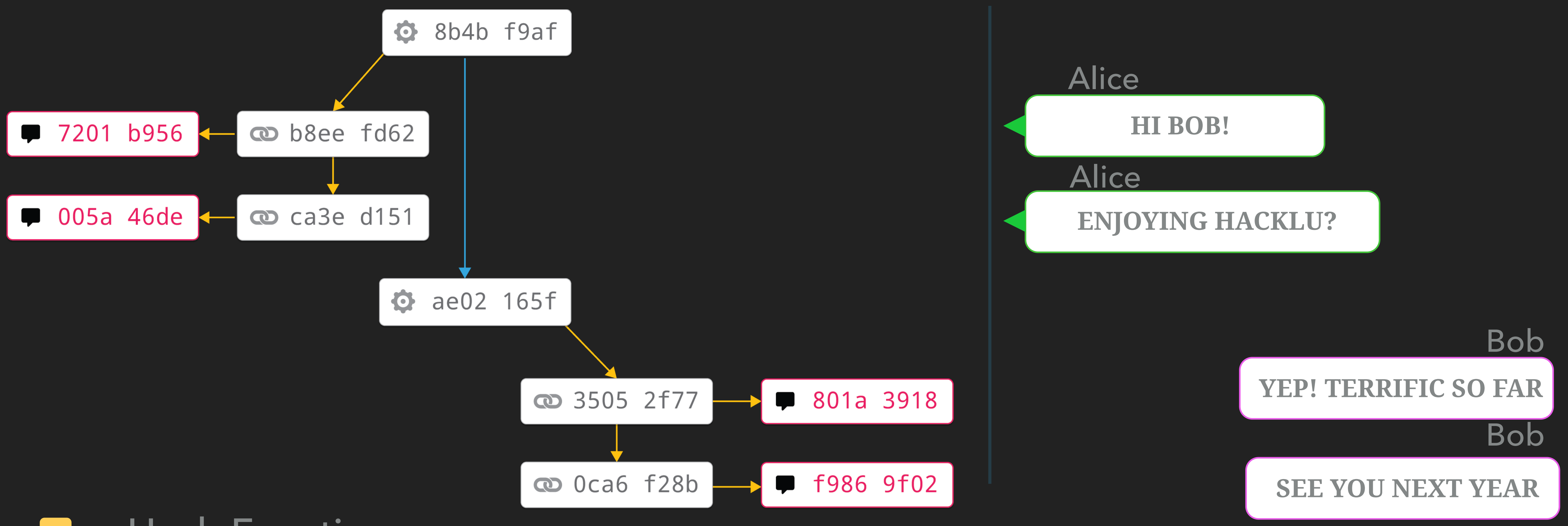
DH RATCHETS

- Provides Perfect Forward Secrecy
- Round trip required to ratchet
- Implemented in OTR
- Self-healing

WINDOWS OF COMPROMISE



AXOLOTL: THE AXAMPLE



🟨 = Hash Function
🟦 = Diffie-Hellman Exchange

FORWARD SECURE ASYNCHRONOUS MESSAGING FROM PUNCTURABLE ENCRYPTION

- Recent paper by Matt Green & Ian Miers (2015)
- New concept of puncturing tags of a "key" to achieve PFS

MULTI-DEVICE

MULTI-DEVICE PROTOCOLS

- Example implementation: Identity key provisioning using QR code
- The ratcheting case is like having two sessions with same identity key.

👪 GROUP MESSAGING

2009: mpOTR PAPER BY IAN GOLDBERG

- Goals:
  - Plausible Deniability
  - Consensus
  - Confidentiality
- Like OTR, synchronous protocol
- Complex protocol, no reference implementations

N-TIMES SENDING PROTOCOL

- Frequently used
- Generates large amounts of cipher text
- No transcript consistency

2014: N+1SEC

- Developed by eQualit.ie with support from the Open Technology Fund and Cryptocat
- Primarily designed for synchronous use cases (making assumptions about transport)

SPAM

Reputation systems require the ability to read *all* email. It's not good enough to be able to see only spam, because otherwise the reputations have no way to self correct. The flow of "not spam" reports is just as important as the flow of spam reports. Most not spam reports are generated implicitly of course, by the act of not marking the message at all.

Mike Hearn on Messaging Crypto Mailing List (05-2014)

Reputation contains an inherent problem. You need lots of users, which implies accounts must be free. If accounts are free then spammers can sign up for accounts and mark their own email as not spam, effectively doing a sybil attack on the system. This is not a theoretical problem.

Mike Hearn on Messaging Crypto Mailing List (05-2014)

ISSUES WITH REPORT-BASED SPAM FILTERING

- Since reputation systems need to know both good and bad messages, it knows who you are messaging with.
- Can’t know if report is honest or not since it can’t verify that users aren’t cheating.

Spam filters rely quite heavily on security through obscurity, because it works well. Though some features are well known (sending IP, links) there are many others, and those are secret. If calculation was pushed to the client then spammers could see exactly what they had to randomise and the cross-propagation of reputations wouldn't work as well.

Mike Hearn on Messaging Crypto Mailing List (05-2014)

WITH SPAM IN MIND

HOW CAN WE REDUCE METADATA

CLIENT FEDERATION OVER HIDDEN SERVICES

- Requires to be online or use of a bouncer
- Provides NAT traversal “for free”. Useful for direct connections without relays including calling case.

POND – PANDA

Since the bandwidth of this system is so low, a user can trivially be incapacitated by a small number of messages. Because of this, we make the system closed: only authorised users can cause a message to be queued for delivery. This very clearly sets Pond apart from email. There are no public addresses to which a Pond message can be sent. Likewise, it's no longer true that the network is fully connected; if you send a message to two people, they may not be able to reply to each other.

Pond Technical Overview

BLINDED SIGNATURES

- Introduced in 1982 by David Chaum while trying to design digital anonymous cash
- Properties:
  - Signer knows nothing about the correspondence between the elements of the set of stripped signed matter s′(x) and s′(c(x))
  - Only one stripped signature can be generated from each thing signed by signer
  - Anyone can check validity

BLINDED SIGNATURES - EXAMPLE

- User chooses x at random and gives c (x) to the signer.
- Signer signs c (x) by applying the signing function and returns the signed matter s' (c (x)) to provider.
- User strips signed matter by application of c', the inverse of the commutative function c, yielding c'(s'(s(x)))) = s'(x)
- Anyone can check that the signature is valid.

BLIND SIGNATURES APPLIED TO RATE LIMITING

- Server still needs to know recipient for routing purposes
- Sender can drop message in “mailbox” of recipient without authenticating by providing a valid signed message.
- Requires anonymity at the network layer (by the use of Tor or similar to prevent easy correlations).

STATE OF MESSAGING PROTOCOLS

- Interesting areas of research
  - Usability of fingerprints and authentication methods
  - Group chat protocols with transcript consistency
  - Spam in fully anonymous and encrypted systems with publicly reachable addresses
  - …

HACK.LU - @FREDERICJACOBS

THANKS! QUESTIONS?

REFERENCES

- Modern Crypto Mailing List
- Open Whisper Systems Blog
- History of the Internet - Wikipedia
- RFCs … many RFCs …