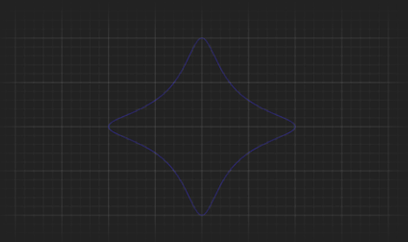
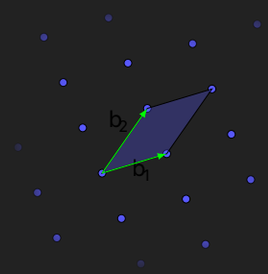
This is an annotated version of my hack.lu (October 2015) slides. This presentation is targeted at hackers and security researchers.
This is NOT a presentation for cryptographers.

Metadata
Theme Color: 5959ff

@FREDERICJACOBS

ACADEMIA





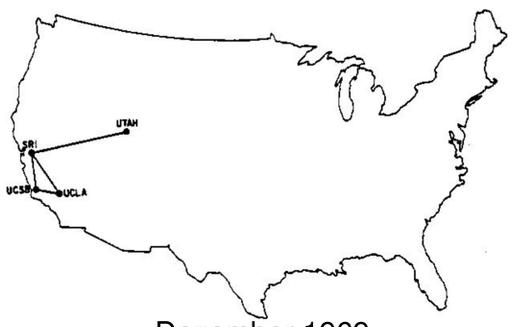
FOSS

I’m Frederic Jacobs. Mainly two things are keeping me busy, academia and FOSS. I have done work with Elliptic Curve Cryptography and Lattices. More relevant to this talk, I’ve been leading the development of Signal (an open-source application that lets you do end-to-end encrypted phone calls and texting).

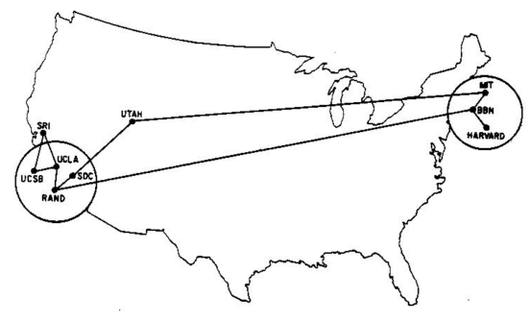
This presentation is about messaging protocols. We are not interested in primitives. In a lot of cases you can swap a primitive for an other but always check with a cryptographer before doing so since in some use cases primitives have different properties than others. For instance, it is perfectly fine to use a form of concatenation for authentication with SHA-3 while using SHA-1 or MD-5 for the same case is going to make you vulnerable to length extension attacks.

The title of this presentation is messaging, let’s set the context for messaging on the Internet.

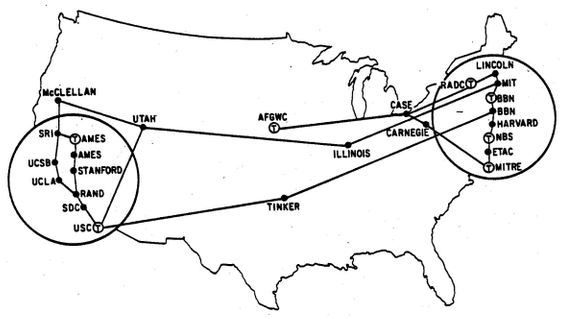
ARPANET



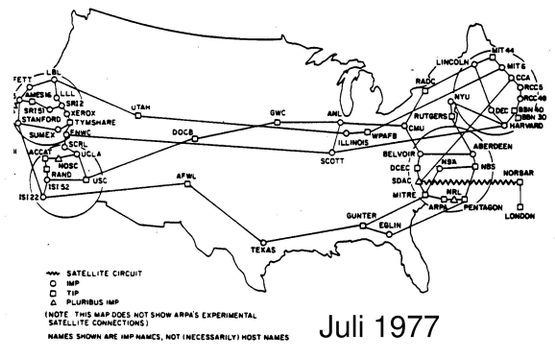
Dezember 1969



Juni 1970



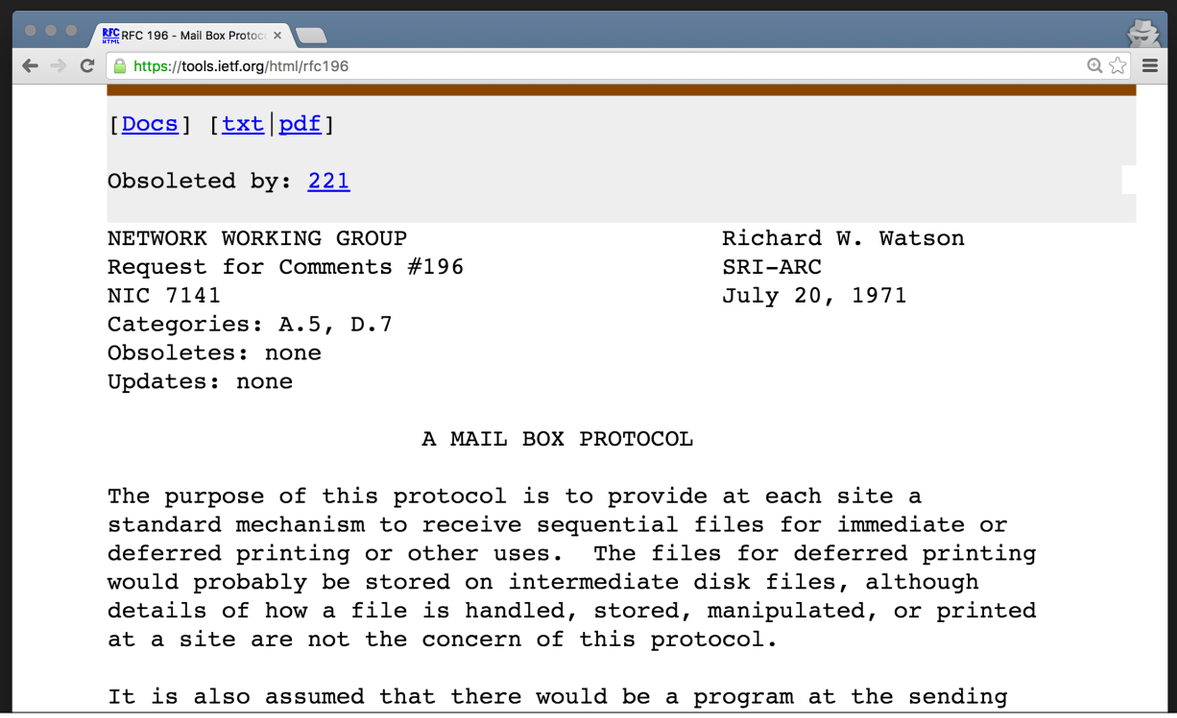
März 1972



Juli 1977

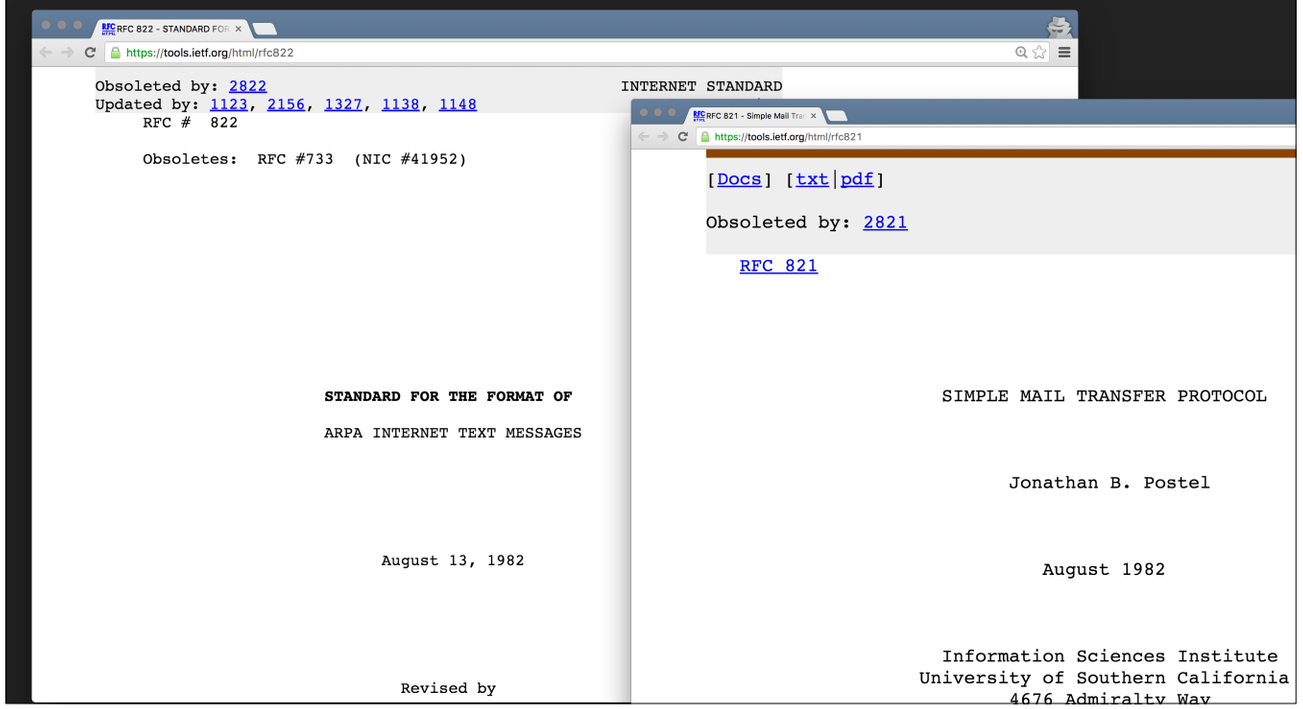
1974 Internet Various forms of one-to-one electronic messaging were used in the 1960s. People communicated with one another using systems developed for specific mainframe computers. As more computers were interconnected, especially in the US Government's ARPANET, standards were developed to allow users of different systems to email one another. SMTP grew out of these standards developed during the 1970s.

1971: A MAIL BOX PROTOCOL



The mail box protocol had most of the ideas that were later standardized as SMTP.
The document doesn’t mention security at all and the protocol is unauthenticated.

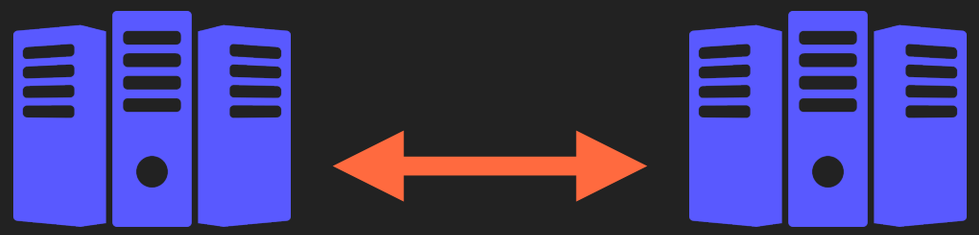
1982: SMTP SPEC IS OUT



Dozens of proposed specifications later (including some like FTPMail based on FTP), SMTP comes to life, roughly 10 years later after the first mailbox protocols! Still not a word about security in the spec and still not addressed lack of authentication (source-spoofable)

https://tools.ietf.org/html/rfc822 <== ARPA Format, legacy

Source IP + POP before SMTP https://en.wikipedia.org/wiki/POP_before_SMTP
https://tools.ietf.org/html/rfc821



⚠ Source-Spoofable

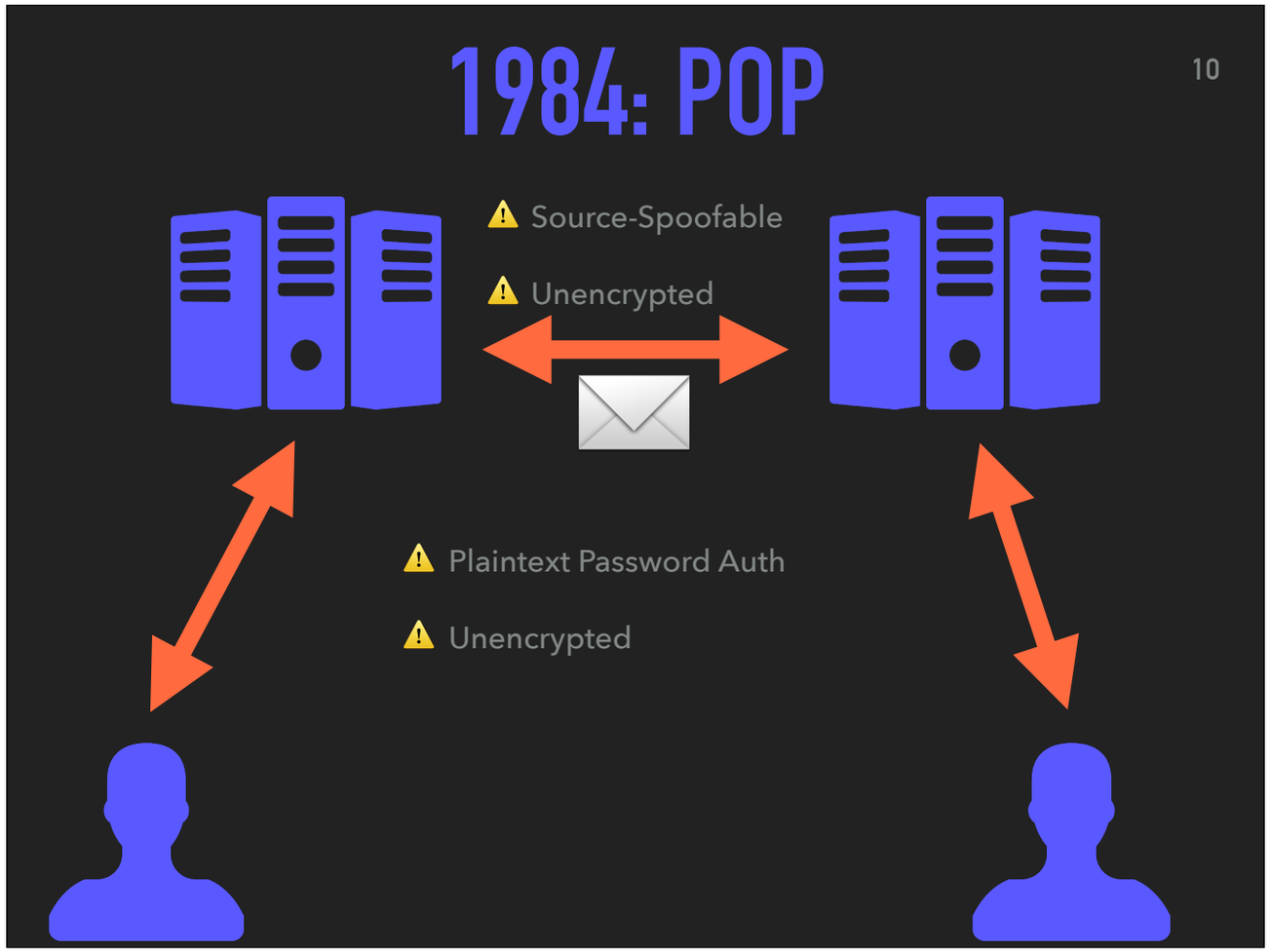
⚠ Unencrypted

EMAIL RETRIEVAL PROTOCOL SECURITY

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

POP arrives, adding attack surface. Plaintext password over unencrypted connection.

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.

It’s interesting that in 1991, end-to-end email encryption appears. Before non-plaintext authentication and years before SSL.

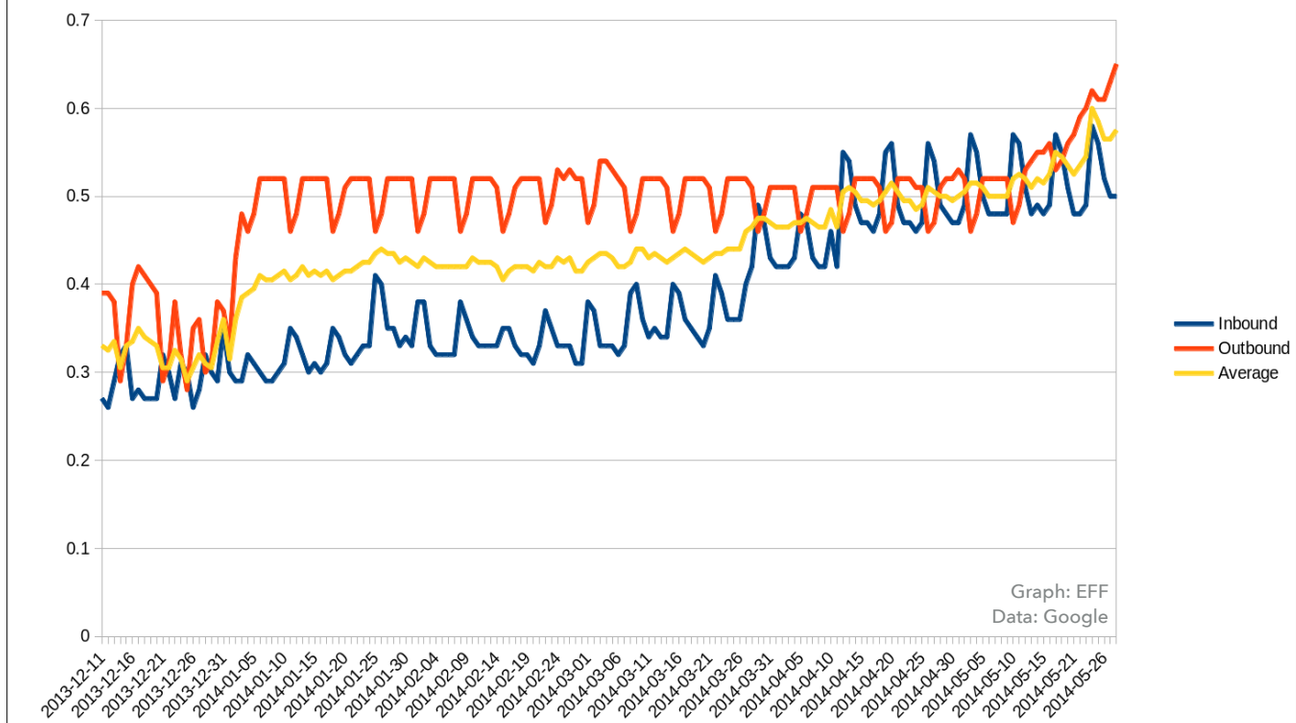
In the same year, IMAP v3 is standardized - still plaintext auth and the spec uses a 5 letter passwords SESAME https://tools.ietf.org/html/rfc1203 :)

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.

Plaintext auth was addressed only in 1994. SSL appeared only in 1995 and was standardized to be applied to SMTP in 1997.

ADOPTION OF SMTPS



Deployment of SMTPS is taking a lot of time. 16 years later, we only saw things moving faster. Thanks Snowden! From 30% to 75% …

Updated numbers were published in a new paper at SIGCOMM http://conferences2.sigcomm.org/imc/2015/papers/p27.pdf

PGP (AND FRIENDS: S/MIME & PEM)

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

Any compromise in key material at any point is a compromise in every message sent to that key or will be sent to that key.

With PGP, if authentication features are desired, document has to be digitally signed.

THE USER EXPERIENCE OF MESSAGING TODAY

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

Messaging needs have changed over time. PGP is not adequate for our modern messaging needs. We want multi-device, group communications, being able to message offline users …

MEANWHILE IN SSH WORLD

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

Interesting ideas came from the SSH world and session based protocols.

OTR

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

The OTR protocol is a session protocol for messaging.

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

Axolotl

Asynchronous with all great features of short lived protocols

Forward secrecy, deniability, conversation integrity …

Can’t we find something that combines best of both worlds? Our mobile communications are intrinsically asynchronous and yet we would enjoy some features from session based protocols.

AXOLOTL

Meet the Axolotl Ratchet - Collaboration between Moxie Marlinspike & Trevor Perrin.

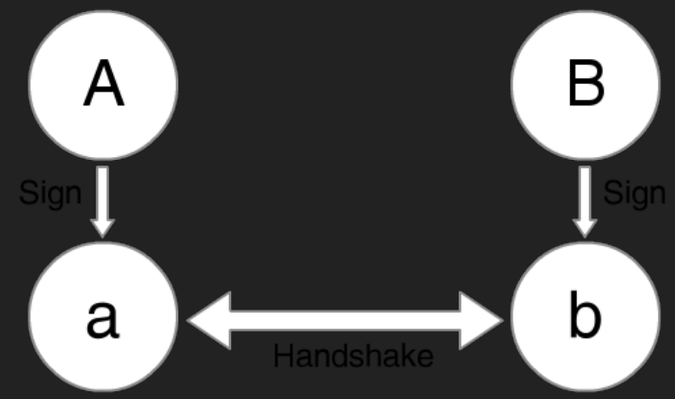
Designed for TextSecure and now increasingly deployed in messaging apps.

DENIABILITY

In most cases, when discussing with someone, you want the other party to be able to verify that messages you're sending were sent by you and not someone else. But you don't want them to be able to prove cryptographically to someone else that you said something. That is deniability.

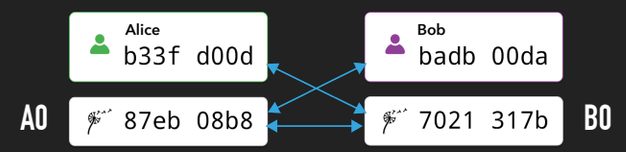
Unlikely to be useful in court though. Especially in group settings but nice to have for a protocol.

OTR HANDSHAKE



How is deniability implemented in OTR? https://whispersystems.org/blog/simplifying-otr-deniability/

3DH–KEY EXCHANGE



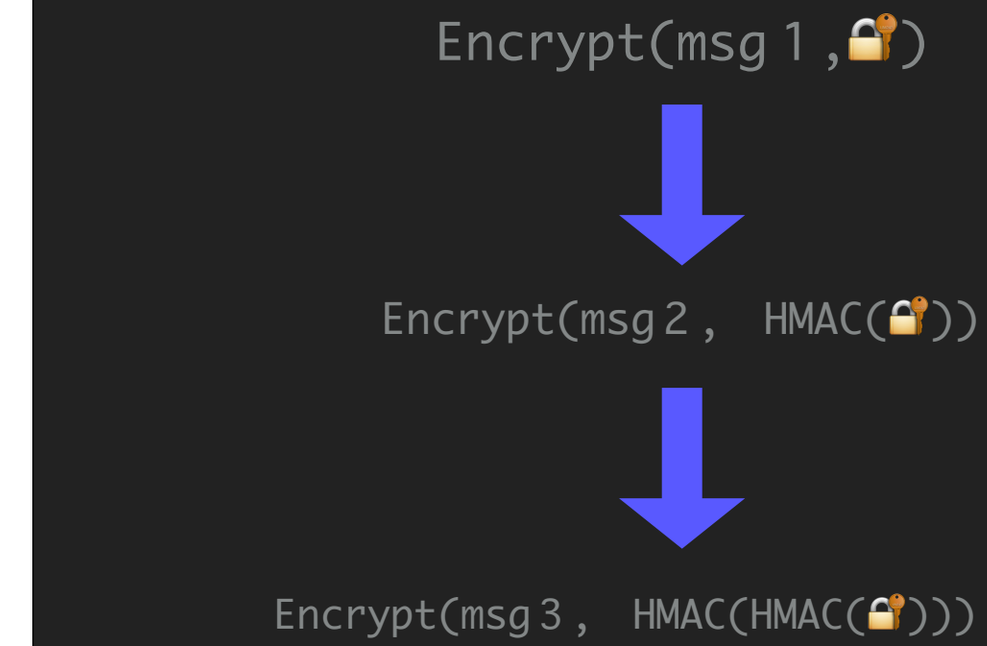
hash(DH(A,B0 8b4 f9af B,A0) || DH(A0,B0))

= Diffie-Hellman

CLOSING THE WINDOW OF COMPROMISE

Ratcheting enforces key rotation. This section is explained in an Open Whisper Systems blog post: https://whispersystems.org/blog/advanced-ratcheting/

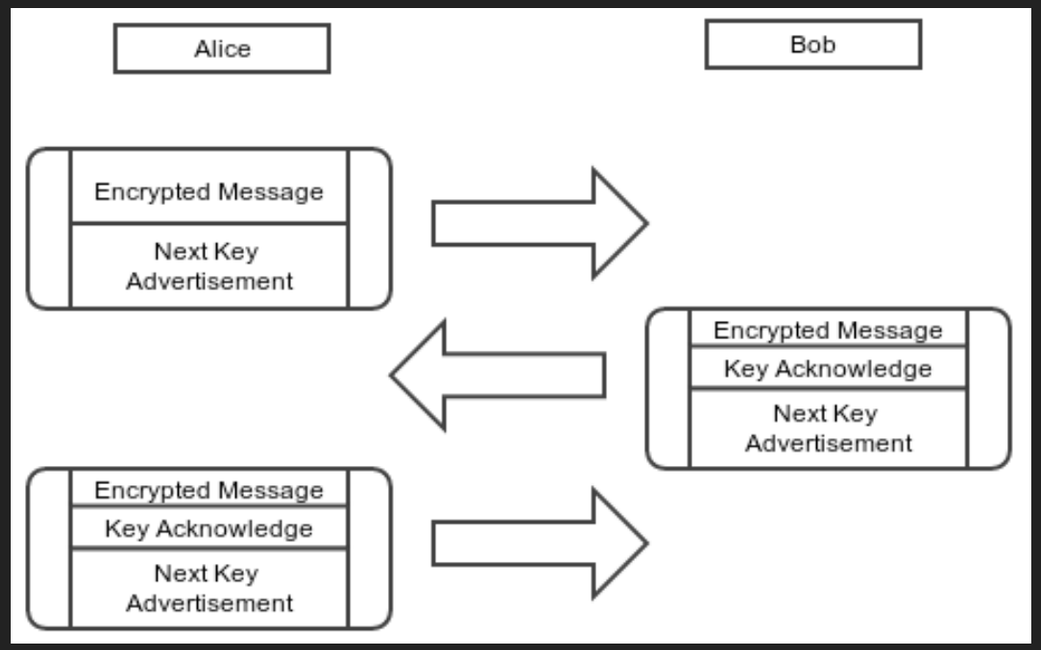
HASH-ITERATED RATCHETS



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

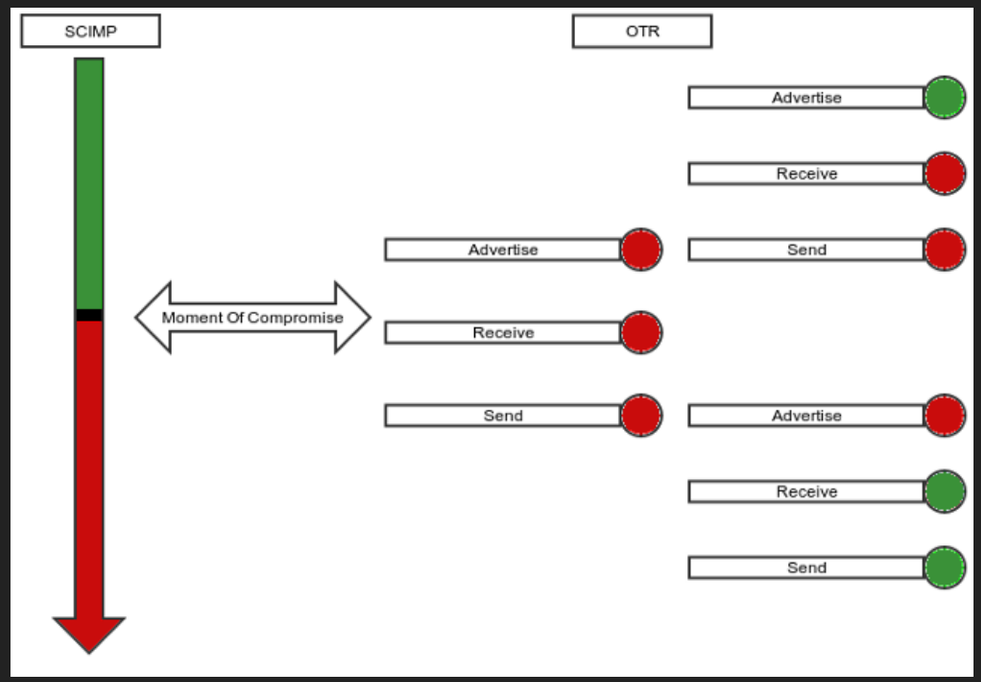
DH RATCHETS


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

DH RATCHETS

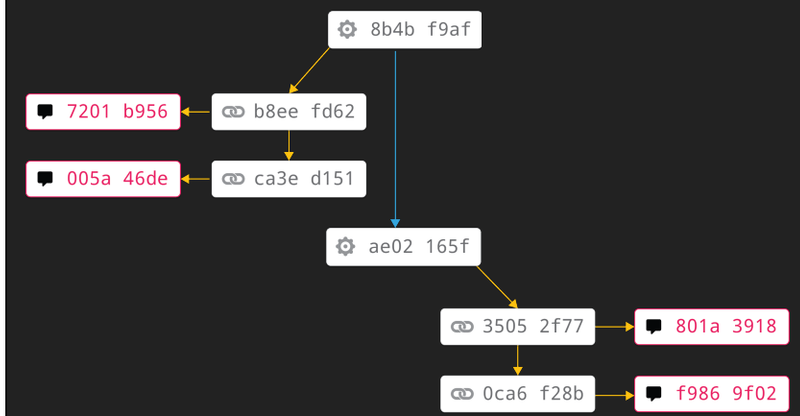
- Provides Perfect Forward Secrecy
- Round trip required to ratchet
- Implemented in OTR
- Self-healing

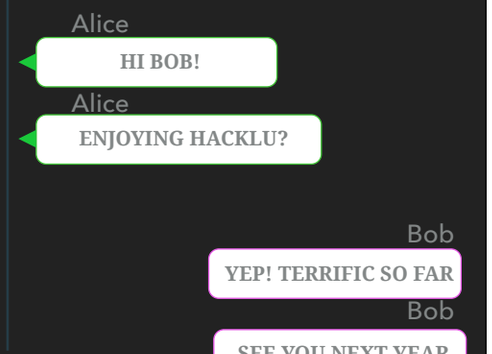
WINDOWS OF COMPROMISE



Hash-Iterated ratchets will limit exposure of previous messages in the case of key compromise but is inefficient against future attacks since no entropy is ever added to the ratcheting and you can deterministically compute all future keys.

AXOLOTL: THE AXAMPLE





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

Note: if you compromise the original key (non-punctured), all messages will be decryptable. No perfect future secrecy.

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.

Multi-device is just a “group chat” except some members have the 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

There exist solutions to transcript consistency, sadly the biggest issue is in UX. How to you represent the different state of messages depending on who got them. How do you make it scale and adapt to latency?

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)

Personally worried about complexity, scalability and latency issues of consensus protocols.
https://equalit.ie/portfolio/np1sec/

SPAM

How do you fight SPAM in fully encrypted systems?

> 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.

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.

Blinded signatures can also be implemented over Elliptic Curves.

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 …