Lecture 15: Email Security and Secure Messaging

TTM4135

Relates to Stallings Chapter 19.
Stallings does not cover secure messaging

Spring Semester, 2025

Motivation

- Email remains one of the most widely used forms of electronic communication but is often sent without end-to-end security
- Instant messaging is increasingly popular and has been built with good security
- Both use cryptography extensively but in practice have very different security propeties

Outline

Email architecture

- Message user agent (MUA) connects client to mail system. Uses SMTP to send mail to message submission agent (MSA) and POP or IMAP to retrieve mail from message store (MS).
- Message handling system (MHS) transfers message from MSA to MS via one or more message transfer agent (MTA)
- Simple message transfer protocol (SMTP) is mail transmission protocol defined in RFC 5321
- Today it is very common to use *webmail* which is a browser interface to an online email client. Note that SMTP and POP/IMAP are still used to send and receive email

Email architecture in a picture

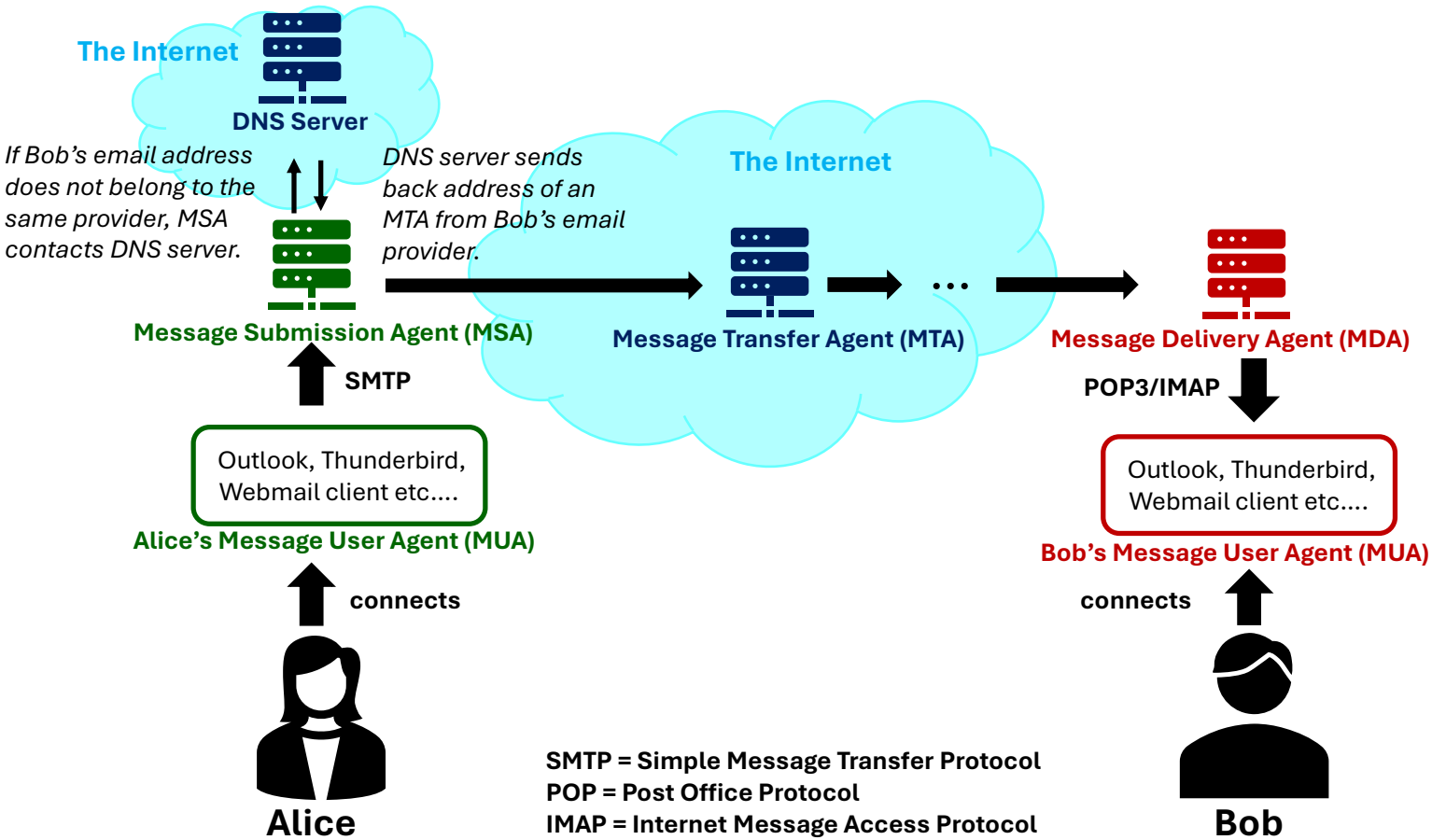

Figure: Email architecture

Security threats against email

- We may consider threats in the usual ‘CIA’ categories
- Email content may require confidentiality or authentication
- Availability of the email service may be threatened
- Metadata in header information is a significant source of attacker information

Spam

- Unsolicited (bulk) email
- A cheap form of advertising?
- Common vector for phishing attacks
- Countermeasures typically use email filtering
- Phishing with more accurate targeting (spear phishing) is harder to filter

Link security and end-to-end security

- Security may be provided between different agents in the mail system on a link-by-link basis using protocols such as STARTTLS and DKIM
- Alternatively it may be provided from client to client (end-to-end) using protocols such as PGP and S/MIME
- Both have their advantages and disadvantages. Ideally both are used.

Link security and end-to-end security in a picture

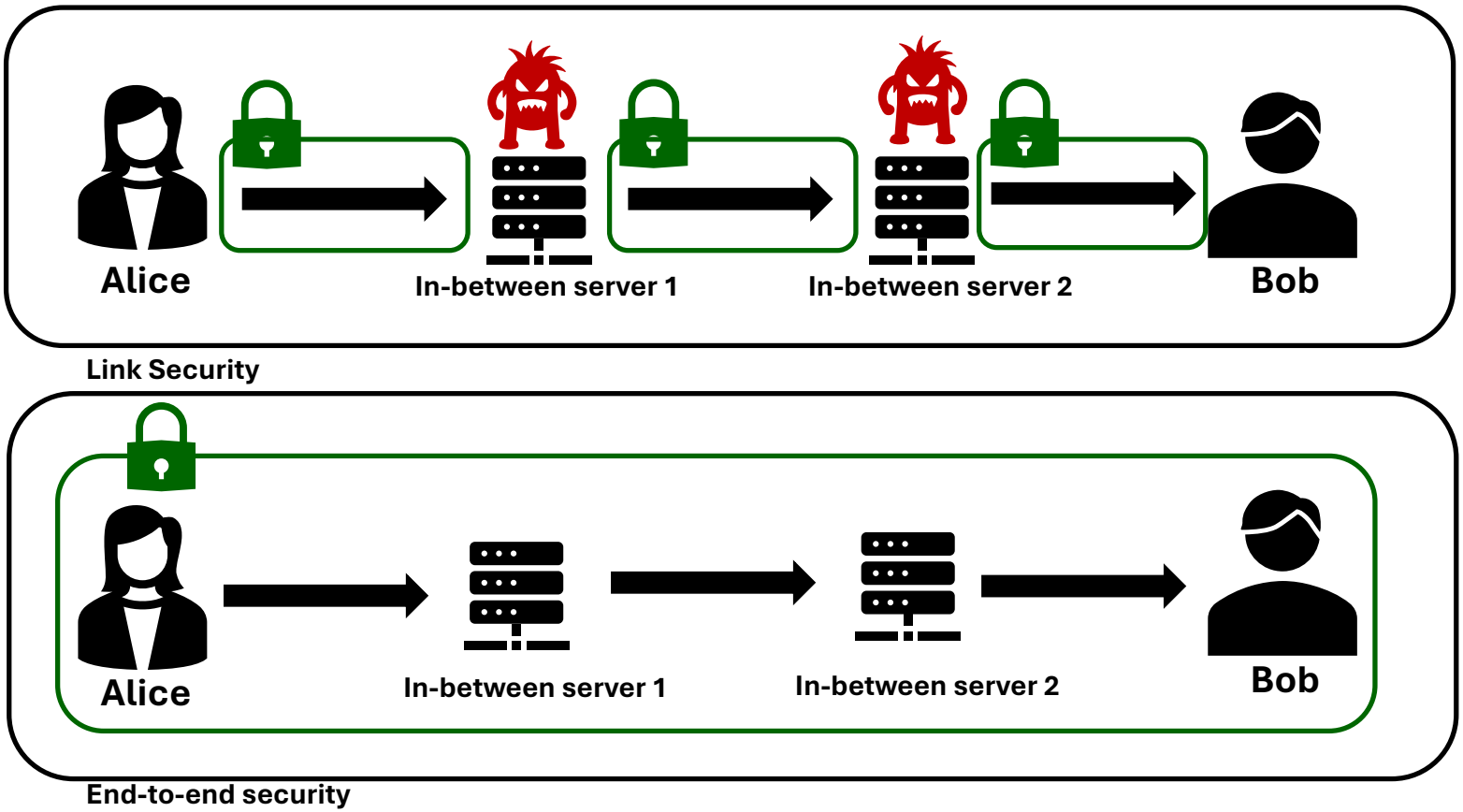

Figure: Link Security vs End-to-end Security

STARTTLS

- Extensions to mail protocols SMTP, POP and IMAP to run over TLS connections
- Provides link-by-link security, not end-to-end security
- *Opportunistic* use of TLS security (encryption) — use it if possible
- Defined for IMAP and POP3 (RFC 2595) and for SMTP (RFC 3207) amongst other protocols
- Widely used by prominent email providers including Gmail and Microsoft Outlook
- Vulnerable to so-called STRIPTLS attacks – attacker interrupts TLS negotiation and connection falls back to plaintext transmission.

DomainKeys Identified Mail (DKIM)

- Standardised in RFC 6376 (2011)
- Allows sending mail *domain* to sign outgoing mail using RSA signatures (currently supported signature algorithm)
- Receiving *domain* can verify origin of mail
- Widely used by prominent email providers including Gmail
- Helps prevent email spoofing and hence reduce spam and phishing
- Example on next slide shows 2048 bit RSA signature on message, coded in base64
- Public verification key of sending domain retrieved using DNS

Example DKIM signature

```
v=1;   // Version
a=rsa-sha256;  // Algorithm
c=relaxed/relaxed; // Header/body canonicalization (format)
d=easychair.org;  // Domain claiming origin
s=default;        // Selector subdividing namespace
t=1677503401; // Timestamp
h=Content-Type:Date:From:Subject:Sender:From;
// Signed header fields
bh=L56upQ4J/BTdlVqCi3PP+Ab67CIehSnUzUFm1aRFEIg=;
// Hash of the body part
b=cS0GpBApvz1YTNs93xkduJgryOnEp/l/t+TAvRFb0HL16ACrttSdnN
UoMVT1se1ZxPpqff9DaAW5DSeBrm5CQUfJvnf8Q7e2ZvJGukpJiiRn
NfNCVy5TIxI5N1oDXCeUT8q kn/YcyxzOjpF+8mmzFo4aK/5NQD/jT1
/Ydfwl/jegHB0c9+rNPHgtlJd7ANOc+GNgS XCHIYL4jhMTnCN4VNM
sgBLQMhFcfU0rWbNaX6Z37r9PwvEli+MpXzYHL68do9skO8B O60Y
Z9MOG8vI1ara40DIuRTVdK3d45geYOTy3rp55VbKC/kY4AKMCCwm
dFgMl75KY7 f5QCpWUhpogEQ==      // Signature
```

DKIM public keys

- The 'd=' and 's=' parts of the DKIM signature specify domain and selector
- The relevant public key is in the DNS record for the host defined by the host name:
  `[selector]._domainkey.[domain]` where
  - 's=' value is the selector
  - 'd=' value is the domain
- In the example header above the nslookup would be:

```
nslookup -type=txt default._domainkey.easychair.org
```

Take-up of DKIM and STARTTLS

- In February 2023, Gmail was using STARTTLS for around 90% of both outgoing and incoming emails. (https://www.google.com/transparencyreport/saferemail/?hl=en)
- A 2020 study found that just under 60% of emails included a DKIM signature: Georgios Kambourakis and others. *What Email Servers Can Tell to Johnny: An Empirical Study of . . . Email Security*, IEEE Access, July 2020.
- The same study noted around 97% usage of STARTTLS

History of PGP

- Originally product of one person — Phil Zimmermann
- Subject of widely reported export restriction controversy
- OpenPGP standard, specified in RFC 4880, allows for interoperable implementations
- GnuPG (GPG) is an open implementation.
- PGP corporation acquired by Symantec in 2010

Photo: User Matt Crypto on en.wikipedia

Email processing

- Protection of email message contents
- Hybrid encryption — a new random “session key” is generated for each object (message) and the session key is encrypted with the long-term public key of recipient
- Signing using RSA or DSA signatures
- Compression using Zip
- Coding using base-64 to ensure that binary strings can be sent in email body

PGP encryption

- Session keys are encrypted using asymmetric encryption. OpenPGP requires support for ElGamal encryption and recommends also to support RSA encryption.
- Encryption of message text using symmetric key encryption – OpenPGP requires support for 3DES with three keys (168 bits in total) and recommends also AES-128 and CAST5. Other algorithms are also defined.
- Compression is applied before encryption
- Encryption can be applied independently of signing (no requirement for authenticated encryption)

PGP signatures

- Plaintext message is optionally signed with sender's private key
- OpenPGP standard requires support for RSA signatures
- DSA signatures also defined
- RSA signed messages are hashed with SHA1 (support required in standard) or other SHA2 hash functions

OpenPGP PKI

- Used in PGP email security
- Includes ID, public key, validity period and a *self-signature*
- No certification authorities — keys can be signed by anyone
- Various key servers used to store keys, such as `https://keys.openpgp.org/`
- Often known as the *web of trust*

Web of Trust in Pictures

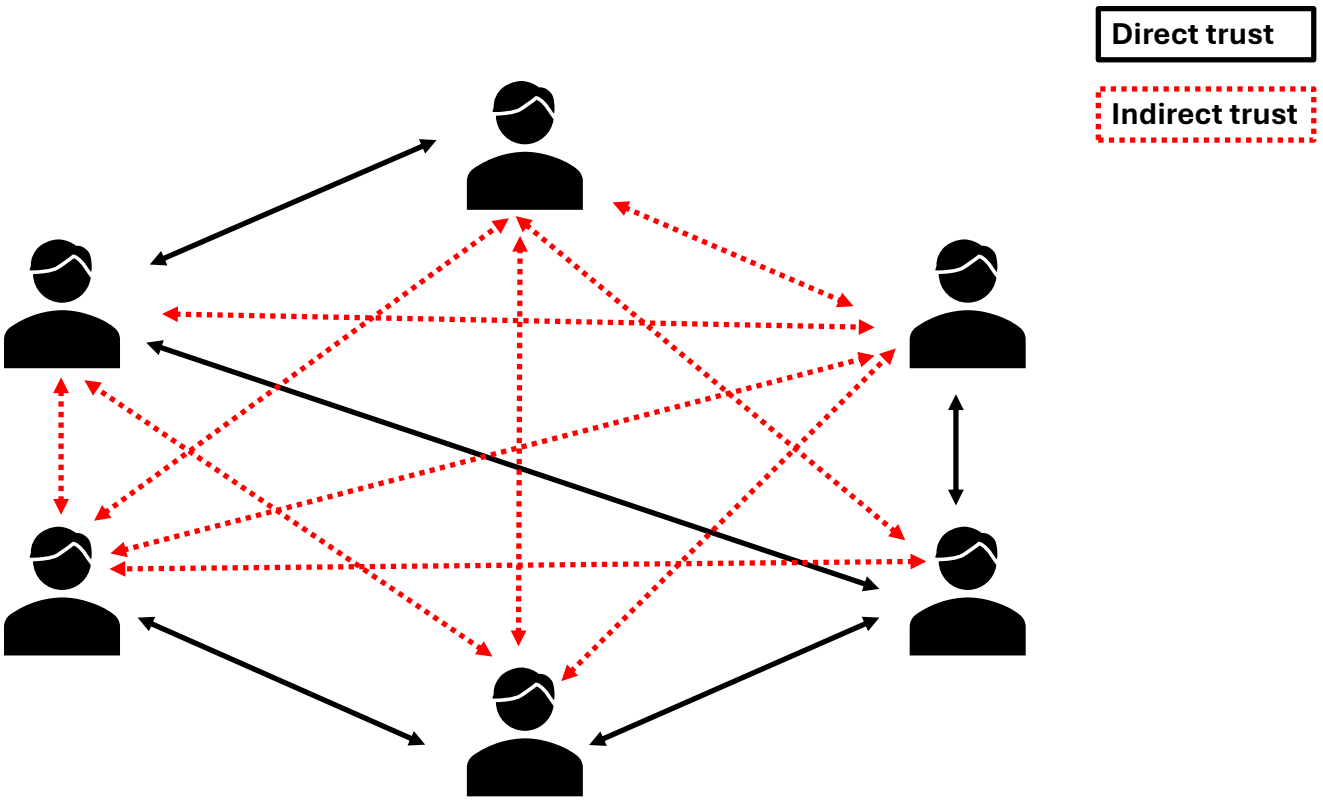

Figure: Schematic Representation of the Web of Trust

Central Authority in Pictures

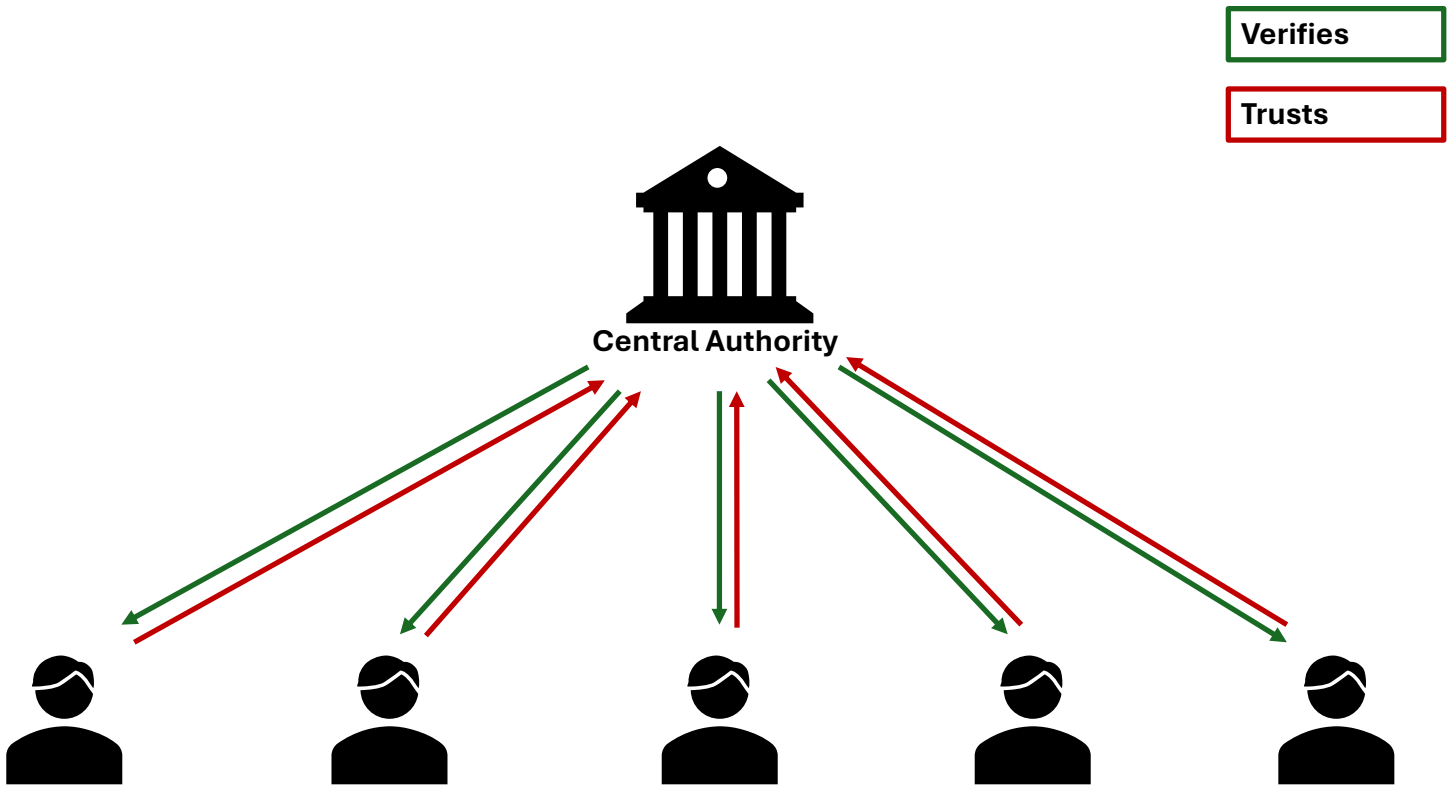

Figure: Schematic Representation of a Central Authority

Usability

- Can we expect the average user to understand public key cryptography?
- Is it possible to design a PGP interface that helps users to operate PGP correctly and safely?
- See: Alma Witten and J. D. Tygar, *Why Johnny can't encrypt: A Usability Evaluation of PGP 5.0*, 1999
- Follow-up studies show that newer PGP versions are still hard to use
- Typical problems:
  - Generating new keys securely
  - Moving keys between devices
  - Renewing keys when they expire

Take-up of PGP

- Plugins available for many popular mail clients and for webmail interfaces (Mailvelope, OpenKeyChain) (see list at https://www.openpgp.org/software/)
- Some mailer servers, such as ProtonMail, provide compatibility but manage your private key for you
- The key server https://keys.openpgp.org/about was launched in June 2019 and has currently around 350000 keys

Criticisms of OpenPGP

- Outdated cryptographic algorithms still used: SHA1, CAST, Blowfish, ...
- No support for SHA3 or authenticated encryption such as GCM
- A lot of metadata is available to an eavesdropper including
  - file length
  - encryption algorithm used
  - key identity of recipients
- No forward secrecy
- Does not support streaming mode or random access decryption

S/MIME

- Similar security features to PGP but different format for messages and not interoperable
- Requires X.509 format certificates instead of web of trust
- Supported natively by most popular mail clients

Differences between email and messaging

Email and messaging have obvious similarities but also important differences

- Most instant messages are part of an interactive conversation which extends over many messages and a long time
- Proprietary servers are typically used to manage accounts and dedicated applications are used

Messaging security

- The standard CIA security services are important as usual
- Forward secrecy is important especially for long sessions — achieved using *medium-term* public keys stored at the server
- Desirable also to have *post-compromise security* (self-healing): an attacker who obtains a long-term key should be locked out again after communication resumes

Messaging security standards

- There is no standardized (secure) messaging protocol
- Different apps do security in different ways - with varied levels of success (see Wikipedia comparison)
- Snapchat, Discord: no End-to-End encryption
- (Facebook) Messenger: End-to-end encryption since April 2024.
- iMessage, and Whatsappare (allegedly) secure.
- Telegram only offers encrypted chat, if a secret chat is opened, the normal chats are not by default encrypted. Additionally, they use a custom encryption protocol.
- Signal is generally considered the most secure and is open source

Attacks on Telegram

Four Attacks and a Proof for Telegram*

Martin R. Albrecht[1], Lenka Mareková[2], Kenneth G. Paterson[3], and Igors Stepanovs[3]

[1] King's College London
martin.albrecht@kcl.ac.uk
[2] Information Security Group, Royal Holloway, University of London
lenka.marekova.2018@rhul.ac.uk
[3] Applied Cryptography Group, ETH Zurich
{kenny.paterson,istepanovs}@inf.ethz.ch

31 March 2023

Figure: Paper from 2023 that shows major attacks on Telegram

Signal protocol

- Signal server sets up initial authentication of user and registers initial public keys
- Public keys at the server are used to set up initial communication between users
- Key exchange uses elliptic curve Diffie–Hellman
- AES in CBC mode with HMAC (SHA256) used for message protection
- Protocol is used in Signal app and claimed also to be in WhatsApp and Facebook Messenger (closed source)

Ratcheting

- A ratchet is a device which is easy to move forward but blocked from moving backward
- Signal uses a new unique message key for every message exchanged, known as *continuous key exchange*
- When successive messages sent in the same direction the message key is updated with a *symmetric ratchet* by applying a function such as HMAC
- When a new message is returned in the opposite direction a new Diffie-Hellman ephemeral key is used to compute the new message key: this is the *Diffie-Hellman ratchet*
- Many more details in the online specification: `https://signal.org/docs/specifications/doubleratchet/`

Group messaging

- No good alternative for Diffie-Hellman is known in the mutli-party case
- Signal uses a simple key distribution method for group messaging
- Currently a research effort is under way to develop Messaging Layer Security (mls) standard: https://datatracker.ietf.org/wg/mls/about/