NTNU | Norwegian University of Science and Technology

EMAIL SECURITY AND SECURE MESSAGING

TTM4135 - Lecture 15

Tjerand Silde

26.02.2026

Motivation

- Email remains one of the most widely used forms of communication but is often sent without end-to-end security. We will study practical solutions.
- Instant messaging is increasingly popular (e.g. Signal and WhatsApp) and frequently designed with end-to-end security in mind.
- Both applications rely heavily on cryptography, yet their real-world security properties and setup differ significantly.

Contents

Some Notions

- **MUA** - Message User Agent; a client application for reading and sending email (e.g., Outlook, Gmail).
- **MTA** - Mail Transfer Agent; an agent that transfers email messages between hosts using the Simple Mail Transfer Protocol (SMTP).
- **MSA** - Message Submission Agent; an agent that receives mail from MUA and cooperates with MTAs.
- **IMAP** - Internet Message Access Protocol; standard protocol for retrieving email over TCP/IP.
- **POP** - Post Office Protocol; alternative to IMAP for retrieving email.

Email Architecture

- MUA uses SMTP to send mail to the MSA and POP/IMAP to retrieve messages from message store (MS).
- Message Handling System (MHS) transfers messages between MTAs.
- SMTP is defined in RFC 5321. Webmail is now common, but SMTP and IMAP/POP still underpin the communication.

Security Threats Against Email

- Threats may be categorized as:
  - Confidentiality
  - Integrity
  - Authentication
- Email content may require confidentiality or sender authentication.
- Availability of email services can be attacked.
- Metadata (headers) reveals valuable information to an attacker.

Spam

- Unsolicited bulk email.
- Often used as a cheap form of advertising.
- Common attack vector for phishing.
- Countermeasures include filtering.
- Spear-phishing (targeted) is harder to filter.

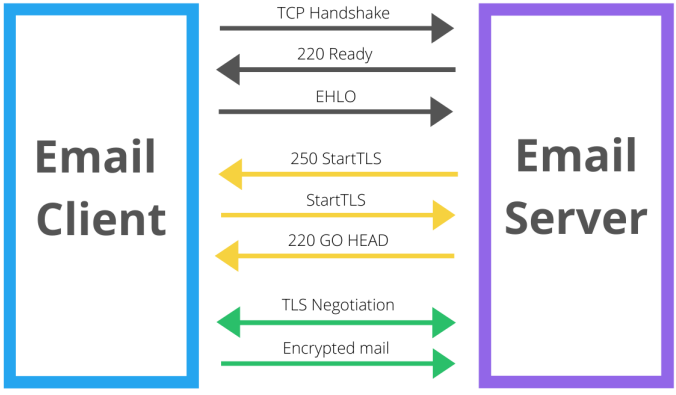
Link Security vs. End-to-End Security

- Link security (STARTTLS, DKIM) protects communication between servers.
- End-to-end security (e.g., PGP, S/MIME) protects client-to-client messages.
- Both approaches have strengths and weaknesses.

STARTTLS

- Extensions to SMTP, POP, and IMAP enabling TLS encryption.
- Provides link-by-link protection, not end-to-end.
- Opportunistic: falls back to plaintext if TLS unavailable.
- Defined in RFCs 2595 (IMAP/POP3) and 3207 (SMTP).
- Widely deployed in major user agents (Gmail, Outlook).
- Vulnerable to STRIPTLS downgrade attacks.

STARTTLS Diagram



DomainKeys Identified Mail (DKIM)

- Email authentication by adding digital signatures to outgoing messages.
- Standardized in RFC 6376 (2011).
- Sending domain signs email using RSA.
- Receiving domain verifies authenticity.
- Helps prevent spoofing, spam, and phishing.
- Public keys retrieved via DNS.

Example DKIM Signature

```
v=1; a=rsa-sha256; c=relaxed/relaxed;
d=easychair.org; s=default; t=1677503401;
h=Content-Type:Date:From:Subject:Sender:From;
bh=L56upQ4J/BTdlVqCi3PP+Ab67CIehSnUzUFm1aRFEIg=;
b=cSOGpBApvz1YTNs93xkduJgryOnEp/l/t+TAvRFbOHL16ACrttSdnN ...
```

DKIM Public Keys

- DKIM uses “d=” (domain) and “s=” (selector) fields.
- DNS record: `[selector]._domainkey.[domain]`
- Example lookup:

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

- Email clients allow "View source" to inspect DKIM headers.

DKIM and STARTTLS Deployment

- In 2023, Gmail used STARTTLS for $\sim 90\%$ of email traffic.
- A 2020 study showed:
  - $\sim 60\%$ of emails carried a DKIM signature.
  - $\sim 97\%$ of emails used STARTTLS.

History of PGP

- Developed by Phil Zimmermann.
- Sparked export-control controversy.
- Standardized as OpenPGP in RFC 4880.
- GnuPG (GPG) is an open-source implementation.
- PGP Corp. acquired by Symantec in 2010.

Email Processing Under PGP

- Hybrid encryption using a per-message session key.
- Signatures using RSA or DSA.
- Compression (ZIP).
- Base64 encoding for safe transport.

PGP Encryption

- Session keys encrypted with asymmetric encryption (ElGamal required, RSA recommended).
- Message text encrypted symmetrically (OpenPGP requires 3DES; recommends AES-128, CAST5).
- Compression applied before encryption.
- Signing and encryption are independent.

PGP Signatures

- Optional signing using sender’s private key.
- OpenPGP requires RSA signature support.
- SHA-1 required (legacy), SHA-2 recommended.

OpenPGP PKI

- Used in PGP-secured email.
- Keys contain ID, public key, validity, and self-signature.
- No CAs: any user may sign another user's verification key.
- Public key servers such as `https://keys.openpgp.org`.
- Known as the "web of trust", where you trust people other trust.

Usability Challenges

- Many users struggle with public-key cryptography.
- "Why Johnny Can’t Encrypt" (Witten & Tygar, 1999).
- Later studies show persistent usability issues:
  - Secure key generation
  - Key transfer between devices
  - Renewing expired keys

Take-up of PGP

- Plugins exist for many mail clients and webmail (e.g., Mailvelope).
- Services like ProtonMail manage private keys on behalf of users.
- `https://keys.openpgp.org` (launched 2019) hosts $\sim$ 350,000 keys.

Criticisms of OpenPGP

- Outdated algorithms remain supported (SHA-1, CAST, Blowfish).
- No support for SHA-3 or authenticated encryption (e.g., GCM).
- Metadata leaks:
  - File length
  - Encryption algorithm used
  - Recipient key identities
- No forward secrecy.

S/MIME

- Provides similar functionality to PGP.
- Uses X.509 certificates (CA-based trust), not web of trust.
- Supported natively by major email clients.

Contents

Email vs. Messaging

Email and messaging share characteristics, but differ significantly:

- Instant messages are interactive and long-lived.
- Messaging platforms use dedicated apps and proprietary servers.

Messaging Security

- CIA security requirements apply:
  - Confidentiality
  - Integrity
  - Authentication
- Forward secrecy important for long-term conversations.
- Post-compromise security (self-healing) is desirable.

Messaging Security Standards

- No universal messaging standard.
- Apps vary greatly in security.
- Discord: no end-to-end encryption.
- Facebook Messenger: E2E default only since 2023.
- iMessage, WhatsApp claim E2E security.
- Signal is considered the most secure.

Signal Protocol

- Server registers user identities and public keys.
- Initial communication set up through server-mediated public keys.
- Key exchange uses elliptic-curve Diffie-Hellman.
- AES-CBC + HMAC-SHA256 for message protection.
- Used by Signal, and reportedly WhatsApp / Messenger.

Ratcheting

- A ratchet allows forward-only progress.
- Signal uses a new message key per message (continuous key exchange).
- *Symmetric ratchet*: update via HMAC per same-direction message.
- *DH ratchet*: new ephemeral DH key when direction switches.
- See `https://signal.org/docs/specifications/doubleratchet`.

Group Messaging

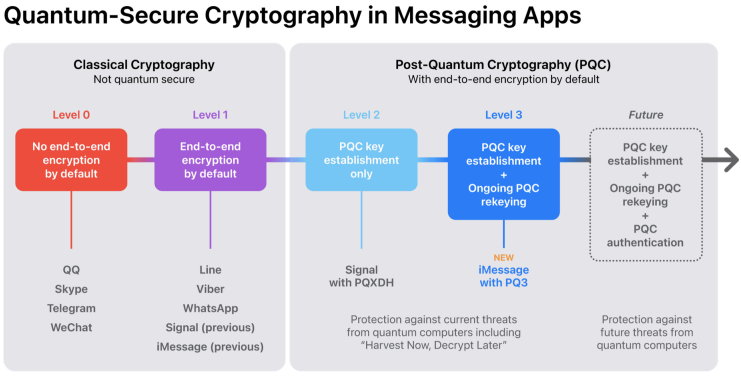
- No efficient multi-party Diffie-Hellman known.
- Signal uses simple key distribution for groups.
- IETF's MLS standard under development:
  `https://datatracker.ietf.org/wg/mls/about`

Contents

Signal's PQXDH Protocol (2023)

- Counters "harvest now, decrypt later" threats.
- PQXDH = Post-Quantum Extended Diffie-Hellman.
- Provides post-quantum forward secrecy.
- Still relies on classical discrete log for mutual authentication.
- Documentation at signal.org/docs/specifications/pqxdh/pqxdh.pdf.

Apple's iMessage with PQ3



Note: This comparison evaluates only the cryptographic aspect of messaging security, and therefore focuses on end-to-end encryption and quantum security. Such a comparison doesn't include automatic key verification, which we believe is a critical protection for modern messaging apps. As of the time of this writing, only iMessage and WhatsApp provide automatic key verification. The iMessage implementation, called Contact Key Verification, is the state of the art – it provides the broadest automatic protections and applies across all of a user's devices.

Figure: https://security.apple.com/blog/imessage-pq3

Questions?