Secure (and Insecure) Messaging

CS249i: The Modern Internet
Before we talk about secure messaging... let's talk about insecure messaging
Email Delivery

Alice → SMTP Submission → smtp.uiowa.edu
Email Delivery

Alice → SMTP Submission → smtp.uiowa.edu → MX stanford.edu? → smtp.gmail.com → DNS Server
Email Delivery

Alice → SMTP Submission → smtp.uiowa.edu → MX stanford.edu? → smtp.gmail.com → DNS Server

SMTP Submission

SMTP Relay

smtp.gmail.com
Email Delivery

Alice

SMTP Submission

smtp.uiowa.edu

MX stanford.edu?

smtp.gmail.com

DNS Server

SMTP Submission

SMTP Relay

smtp.gmail.com

smtp.gmail.com

pop3.gmail.com
Email Delivery

Alice

SMTP Submission

smtp.uiowa.edu

MX stanford.edu?

SMTP Relay

smtp.gmail.com

DNS Server

pop3.gmail.com

SMTP Submission

Here is your mail...

Do I have any mail?

Bob
Email Delivery

Alice → smtp.uiowa.edu → smtp.i2.edu → smtp.gmail.com → smtp.google.com → pop3.google.com → Bob
Email Delivery Security

Alice

SMTP Submission

smtp.uiowa.edu

SMTP Relay

smtp.gmail.com

Bob

Do I have any mail?

Here is your mail…
Email Delivery Security

Email is transferred between two organizations across the Internet

Alice

SMTP Submission

smtp.uiowa.edu

SMTP Relay

smtp.gmail.com

Do I have any mail?

Here is your mail...

Bob

pop3.gmail.com
SMTP

SMTP — Simple Mail Transfer Protocol — is the protocol used for transferring email between servers on the Internet.

The protocol was first introduced in 1982. A number of additional extensions were later added in 2008.

As originally conceived, the protocol had no security features.
SMTP Security

Confidentiality. No protection against eavesdropping for mail sent across the Internet. Anyone on path could read your message.

Integrity. Nothing prevented an active attacker from modifying your messages in transit, or spoofing emails as you.

Availability. Little guarantee of uptime or availability of email data (i.e., email delivery).
SMTP Extensions

Several extensions to SMTP were later introduced to provide email security, including STARTTLS, SPF, DKIM, and DMARC.

Their deployment has been largely hidden from sight.
STARTTLS Extension

STARTTLS enables the sender to start an encrypted TLS session when delivering mail. Messages are transferred over the encrypted session.
STARTTLS Protocol

TCP handshake

220 Ready

EHLO

250 STARTTLS

STARTTLS

220 GO HEAD

TLS negotiation

Encrypted email
Opportunistic Encryption

“A publicly-referenced SMTP server MUST NOT require use of the START TLS extension in order to deliver mail locally. This rule prevents the STARTTLS extension from damaging the interoperability of the Internet's SMTP infrastructure.” (RFC3207)

Unlike HTTPS, STARTTLS is used opportunistically. Senders do not validate destination servers — the alternative is cleartext. Many servers do not support STARTTLS.
What name do you validate?

Unlike HTTPS, unclear what name should go on the certificate

**MX Server (e.g., smtp.gmail.com)**
- No real security added
- MITM returns bad MX record

**Domain (e.g., gmail.com)**
- No solution for cloud providers
How much of email is protected in practice?
Yahoo and Hotmail deploy STARTTLS

Today, 92-93% of messages are encrypted

Gmail rolls out indicators

Neither Snow Nor Rain Nor MITM... An Empirical Analysis of Email Delivery Security. IMC’15; Google
Long Tail of Operators (2015)

These numbers are dominated by a small number of large providers

Of the Alexa Top 1M most popular domains:

- 80% support STARTTLS
- 34% have certificates that match mail server
- 0.6% have certificates that match domain

This is the only case where you know you’re sending mail to the right place.
Implications for Mail Providers

Because so many servers still do not support encryption, mail providers are forced to allow mail to be sent unencrypted. Doesn’t that mean that an active attacker can eavesdrop if they can prevent a secure connection?
What’s the simplest way to eavesdrop on connections that use STARTTLS?
I support STARTTLS!

I support STARTTLS!

STARTTLS Protocol

TCP handshake

220 Ready

EHLO

250 STARTTLS

STARTTLS

220 GO HEAD

TLS negotiation

Encrypted email

Me too. I’ll start a TLS connection

Me too. I’ll start a TLS connection
STARTTLS Stripping (1)

TCP handshake →

← 220 Ready →

← EHLO →

← 250 XXXXXXXXXXX →

250 STARTTLS ←

Cleartext Email →

I don't know about XXX...

I support STARTTLS!
STARTTLS Stripping (2)

TCP handshake

220 Ready

EHLO

250 STARTTLS

STARTTLS

??!!?!

XXXXXXX

Cleartext Email
# Attacks in the Wild

<table>
<thead>
<tr>
<th>Country</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tunisia</td>
<td>96.1%</td>
</tr>
<tr>
<td>Iraq</td>
<td>25.6%</td>
</tr>
<tr>
<td>Papua New Guinea</td>
<td>25.0%</td>
</tr>
<tr>
<td>Nepal</td>
<td>24.3%</td>
</tr>
<tr>
<td>Kenya</td>
<td>24.1%</td>
</tr>
<tr>
<td>Uganda</td>
<td>23.3%</td>
</tr>
<tr>
<td>Lesotho</td>
<td>20.3%</td>
</tr>
<tr>
<td>Sierra Leone</td>
<td>13.4%</td>
</tr>
<tr>
<td>New Caledonia</td>
<td>10.1%</td>
</tr>
<tr>
<td>Zambia</td>
<td>10.0%</td>
</tr>
</tbody>
</table>
Are these truly attacks?

<table>
<thead>
<tr>
<th>Organization Type</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corporation</td>
<td>43%</td>
</tr>
<tr>
<td>ISP</td>
<td>18%</td>
</tr>
<tr>
<td>Financial Institution</td>
<td>14%</td>
</tr>
<tr>
<td>Academic Institution</td>
<td>8%</td>
</tr>
<tr>
<td>Healthcare Provider</td>
<td>3%</td>
</tr>
<tr>
<td>Unknown</td>
<td>3%</td>
</tr>
<tr>
<td>Airport</td>
<td>2%</td>
</tr>
<tr>
<td>Hosting Provider</td>
<td>2%</td>
</tr>
<tr>
<td>NGO</td>
<td>1%</td>
</tr>
</tbody>
</table>

Cisco advertises this feature to prevent attacks and catch spam.

Unclear if operators know they’re putting their users at risk.
Authenticating Email

**Sender Policy Framework (SPF)**
Sender publishes list of IPs authorized to send mail

**DomainKeys Identified Mail (DKIM)**
Sender signs messages with their cryptographic key

**Domain Message Authentication, Reporting, and Conformance (DMARC)**
Sender publishes DNS policy that specifies what to do if message validation fails
Authentication for Gmail

Delivered Gmail Messages

- SPF & DKIM: 81%
- SPF: 11%
- DKIM: 2%
- No Auth: 6%

1M Most Popular Domains

**Technology Top 1M**

- SPF Enabled: 47%
- DMARC Policy: 1%

**DMARC Policy Top 1M**

- Reject: 20%
- Quarantine: 8%
- None: 72%
PGP — Pretty Good Privacy

Third-Party Toolkit for encrypting and signing emails originally developed in 1991

Tremendous Usability and Implementation Challenges

Most Recently: "Our attacks allow the spoofing of digital signatures for arbitrary messages in 14 out of 20 tested OpenPGP-capable email clients and 15 out of 22 email clients supporting S/MIME signatures."

“Johnny, you are fired!” – Spoofing OpenPGP and S/MIME Signatures in Emails (USENIX Security 2019)

tl;dr: Do Not Use PGP if you need security.
Pre-authentication vulnerability allowed attackers to dump mailbox content and remotely execute code on Microsoft Exchange Servers.

"Censys observed 251,211 Microsoft Exchange Servers (2013, 2016, or 2019 versions) across the Internet."

Actually, don't use email at all if you really need security.
SMS — Short Message Service

SMS allows sending 140 byte messages as part of the non-data cellular protocols (e.g., GSM, CDMA, HSPA, 4G, 5G)

Messages are sent using the same type of control messages that your phone uses to coordinate with cellular towers for service

No end-to-end security protections — provider sees everything. Messages are stored and forwarded by your provider.
Phone Number and SMS Hijacking

Recent years have seen an increase in social engineering attacks to hijack phone numbers.

An attack this year also showed how it's possible to hijack the SMS capabilities of a phone through a third provider.

NetNumber — company that provides authoritative database of SMS redirections — allows some companies to change routing of numbers.
Alright... more secure alternatives
OTR: Off-the-Record Messaging

Cryptographic Protocol released in 2004 by Nikita Borisov, Ian Goldberg, and Eric Brewer

Alternative to PGP that runs on top of Instant Messaging Clients (e.g., Jabber)

Precursor to many of today's secure messaging protocols

Beyond Encryption and Authentication, introduced new ideas to messaging security:

**Forward Secrecy:** Messages are encrypted with temporary per-message AES keys, negotiated using the Diffie-Hellman key exchange protocol. The compromise of any long-lived cryptographic keys does not compromise any previous conversations

**Deniability:** Messages do not have digital signatures. Anyone is able to forge a message to appear to have come from one of the participants in the conversation.
Signal Protocol

Protocol created by creators of Signal App. Built on good parts of OTR and Silent Circle Instant Messaging Protocol (SCIMP)

Basis for Signal, WhatsApp, Google E2E Encryption

Based on notion of "double ratchet" between each message
Symmetric-Key Ratchet

Different cryptographic key for each message.

**Significant Downfall:** If an attacker gets access to key, then they can decrypt all future messages.
Diffie-Hellman Ratchet
Diffie-Hellman Ratchet

Alice

Private key ➔ Public key ➔ DH ➔ DH output

Bob

Public key ➔ Private key ➔ DH ➔ DH output ➔ Public key ➔ DH

Bob’s DH ratchet step

Bob’s new ratchet key pair
Diffie-Hellman Ratchet

Alice

Public key

DH

Private key

Public key

DH

DH output

Bob

Public key

Private key

DH

DH output

Private key

Public key

DH

DH output

Public key

Private key

Alice’s DH ratchet step
Messaging Layer Security (MLS) Protocol

RFC in actively development that sets out to create a protocol for asynchronous group keying with forward secrecy and post-compromise security

2 Party Solved: "For two parties, this problem has been studied thoroughly, with the Double Ratchet emerging as a common solution [doubleratchet] [signal]."

But group message situation remained unsolved:

Based on earlier work on "asynchronous ratcheting trees", the protocol presented here uses an asynchronous key-encapsulation mechanism for tree structures. This mechanism allows the members of the group to derive and update shared keys with costs that scale as the log of the group size.

Details: https://datatracker.ietf.org/doc/draft-ietf-mls-protocol/