Email Overview

**MUA:** Mail User Agent
Thunderbird, webmail, Pine, …

**MSA:** Mail Submission Agent
SMTP (port 587)
Often same as initial MTA

**MTA:** Mail Transfer Agent
SMTP (port 25)

**MDA:** Mail Delivery Agent
IMAP (port 143), POP3 (port 110), local,

Typical flow:
MUA → MSA → MTA → … → MTA → MDA → MUA
SMTP Transport Example

S: 220 smtp.example.com ESMTP Postfix
C: HELO relay.example.org
S: 250 Hello relay.example.org, I am glad to meet you
C: MAIL FROM:<bob@example.org>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok
C: RCPT TO:<theboss@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: From: "Bob Example" <bob@example.org>
C: To: "Alice Example" <alice@example.com>
C: Cc: theboss@example.com
C: Date: Tue, 15 January 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice.
C: This is a test message with 5 header fields and 4 lines in the message body.
C: Your friend,
C: Bob
C: .
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye
Email/Messaging Security and Privacy Goals

Protect message content

Verify communicating parties’ identities

Fight spam

(subject of future lecture)

Fight phishing

(subject of future lecture)

Hide communication patterns

(subject of future lecture)
Who can read my email?

Adversaries with local or remote access to my devices

Intruders, spouse, administrator, …

Malware, stolen credentials, physical access, …
Who can read my email?

Adversaries with local or remote access to MTAs and other intermediary servers

Intruders, administrators, other insiders, LEAs, …
Who can read my email?

**Adversaries with access to any intermediate network**

Intruders, administrators, other insiders, LEAs, …

Passive eavesdropping, MitM, DNS poisoning, …
Confidentiality Threats Recap:

Stored messages
  *Compromised system (either local user machine or remote email server)*
  Malware, intruder, insider, stolen/lost device, …
  *Compromised authentication*
  Password theft, phone unlock, …

Messages in transit
  Eavesdropping and interception

Displayed messages
  Screendump, reflections, shoulder surfing, …
Securing Email Transit

These days encryption is mandatory for client-to-server email transmission and retrieval

MUA ➔ MSA: STARTTLS (port 587/25), SMTPS (port 465)

MDA ➔ MUA: POP3S (port 995), IMAPS (port 993)

mikepo@capcom:~> nc smtp.gmail.com 25
220 mx.google.com ESMTP i185sm2356739qhc.49 - gsmtp
HELO foo.example.com
250 mx.google.com at your service
MAIL FROM:<mikepo@example.com>
530 5.7.0 Must issue a STARTTLS command first.

MTA ➔ MTA relaying: *a different story...*
STARTTLS: Opportunistic Encryption

Legacy MTAs may not support TLS
   Fail-open design is necessary

MTAs do their best to deliver messages
   A recipient MTA may present a self-signed cert (common in antispam/AV systems)
   There is no PKI for email…

MitM is trivially easy
   STARTTLS command is sent over a plaintext channel (!)
   Analogous to SSL stripping, but in this case the client has no indication that
   downgrade has happened
   Just assumes that the receiving MTA does not support TLS

Message interception is still possible
   Better than nothing: bulk passive eavesdropping not possible
I want to STARTTLS

mikepo@capcom:~> nc aspmx.l.google.com 25
220 mx.google.com ESMTP h126si17458667qhh.29 - gsmtp
EHLO foo.example.com
250-mx.google.com at your service, [128.59.23.41]
250-SIZE 157286400
250-8BITMIME
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-CHUNKING
250 SMTPUTF8
STARTTLS
220 2.0.0 Ready to start TLS
<TLS Handshake>
I want to STARTTLS

mikepo@capcom:~> nc aspmx.1.google.com 25
220 mx.google.com ESMTP h126si17458667qhh.29 - gsmtpl
EHLO foo.example.com
250-mx.google.com at your service, [128.59.23.41]
250-SIZE 157286400
250-8BITMIME
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-CHUNKING
250-SMTPUTF8
STARTTLS
220 2.0.0 Ready to start TLS
<TLS Handshake>

Can be stripped off by a MitM attacker
Facebook STARTTLS Study: May 2014

~60% of all messages sent via encrypted connection

Only ~30% pass strict validation (mostly due to self-signed certs)
Facebook STARTTLS Study: August 2014

~95% of outgoing messages encrypted with PFS and strict certificate validation

Mostly due to changes by big recipient networks (Microsoft, Yahoo)
How much email was encrypted in transit?

Generally speaking, use of encryption in transit increases over time, as more providers enable and maintain their support. Factors such as varying volumes of email may explain other fluctuations.

Outbound

84%
Messages from Gmail to other providers.

View Past:
- 30 days
- 90 days
- 1 year

Inbound

73%
Messages from other providers to Gmail.

View Past:
- 30 days
- 90 days
- 1 year

Download data
A tiny GUI change prompted many networks to deploy STARTTLS
Google, Yahoo SMTP email servers hit in Thailand

*Staff writer*  |  September 12, 2014  |  telecomasia.net

Internet users in Thailand have been hit by a massive man-in-the-middle attack aimed at grabbing email login credentials from fake SMTP servers.

The attack has been verified on Google's and Yahoo's email servers and on two of the country's largest fixed-line ISPs, though preliminary analysis suggests that all SMTP servers are targeted.

The STRIPTLS attack, as it has become known, works by inserting a man-in-the-middle at the ISPs. This is done via a transparent proxy.
On SMTP, STARTTLS and the Cisco ASA

During the course of trying to increase the security of my e-mail while in transit, I was working on enabling TLS in Postfix to opportunistically encrypt connections to SMTP servers. While verifying my configuration, I ran into an interesting issue.

In order to test my configuration out I was sending e-mails to a Gmail address via Postfix, unfortunately I wasn't seeing any logging in Postfix indicating that TLS was being used. So I attempted to investigate whether STARTTLS was actually being advertised by manually connecting to Google's SMTP servers using telnet:

telnet aspmx.l.google.com 25
Trying 200.203.242.222...205:8480:4801:c82:1a...
Connected to aspmx.l.google.com.
Escape character is '^]'.
EHLO example.com
250-mx.google.com at your service,
250 SIZE 35882577
250-8BITMIME
250-XXXXXXXA
250 ENHANCEDSTATUSCODES

Every server I connected to in Google's MX record was not advertising STARTTLS. On a whim, I attempted to connect to Google's SMTP servers from an entirely different network:

telnet 173.194.68.26 25
Trying 173.194.68.26...
Connected to qa-in-f261.e1800.net (173.194.68.26).
Escape character is '^]'.
220 mx.google.com ESMTP 13510881429ct.164
EHLO stomp.colorado.edu
250-mx.google.com at your service, 1
250 SIZE 35882577
250-8BITMIME
250-STARTTLS
250 ENHANCEDSTATUSCODES
DNS Hijacking

STARTTLS stripping is not the only way to intercept email

DNS MX record poisoning: spoofed MX response
  Compromised name server, MotS DNS poisoning, …
  Messages are diverted through the attacker’s mail server

DANE (DNS-based Authentication of Named Entities)
  Allow X.509 certs to be bound to DNS names through DNSSEC
  Trust anchor assertions: domain operator can securely convey information about which certificate authority should be trusted

MTA-STS (MTA Strict Transport Security – RFC 8461)
  Allows recipient domains to tell senders whether they support TLS, how MTAs should validate certificates, and what to do if TLS negotiation fails
  Client-side policy cache provides TOFU-like protection
Gmail making email more secure with MTA-STS standard
April 10, 2019

Posted by Nicolas Lidzborski, Senior Staff Software Engineer, Google Cloud and Nicolas Kardas, Senior Product Manager, Google Cloud

We’re excited to announce that Gmail will become the first major email provider to follow the new SMTP MTA Strict Transport Security (MTA-STS) RFC 8461 and SMTP TLS Reporting RFC 8460 internet standards. Those new email security standards are the result of three years of collaboration within IETF, with contributions from Google and other large email providers.

SMTP alone is vulnerable to man-in-the-middle attacks

Like all mail providers, Gmail uses Simple Mail Transfer Protocol (SMTP) to send and receive mail messages. SMTP alone only provides best-effort security with opportunistic encryption, and many SMTP servers do not prevent certain types of malicious attacks intercepting email traffic in transit.
End-to-End Email Encryption

Two major standards: **PGP** and **S/MIME** (similar, but incompatible)

- Both rely on public key cryptography
- Both support signing and/or encryption
- Main difference: *how certificates are signed*

**Typical workflow**

- Encrypt message with a random symmetric key
- Encrypt symmetric key with the public key(s) of recipient(s)
- Digitally sign a hash of the message

**Metadata still in the clear (!)**

- Email headers, appended “Received:” records, subject line
Pretty Good Privacy

De facto standard for encrypted email

PGP (Phil Zimmermann) ➔ OpenPGP (RFC 4880)
  Gnu Privacy Guard (GPG): GPL implementation

Authentication
  Senders attach their digital signature to the message
  Receivers verify the signature using public-key cryptography

Confidentiality
  Symmetric key encryption
  Random session key generated for each message
  Session key is encrypted with recipient’s public key

Both are typically used on the same message
PGP Encryption

http://www.slideshare.net/rvenkatesh25/network-security-primer
PGP Signed Message Example

From: alice@wonderland.com
Date: Mon, 16 Nov 1998 19:03:30 -0600
Subject: Message signed with PGP
MIME-Version: 1.0
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit
Content-Description: "cc:Mail Note Part"

-----BEGIN PGP SIGNED MESSAGE-----

Bob,

This is a message signed with PGP, so you can see how much overhead PGP signatures introduce. Compare this with a similar message signed with S/MIME.

Alice

-----BEGIN PGP SIGNATURE-----
Version: PGP for Personal Privacy 5.0
Charset: noconv

iQCVAwUBM+OTwFcsAarXHFerAQESJgP/X3noQN57U/6XVygOFjSY51TpvAduPZ8M
aIfalUKCnULLGGxmtsbwRlDWLtcEwG3k+7zXDfx4YxuUcofGJn0QaTlk8b3nxADL0
O/E1vC/k8zJ6aGaPLB7rTlizamG0t5n6/08rpwwVkRB03tmT8UNMAUCgoM02d6HX
rKvnc2aBPFI=
=muAvH
-----END PGP SIGNATURE-----
PGP Additional Features

Compression
Sign ➔ Compress ➔ Encrypt
Compression after encryption is pointless (no redundancy)
Signature does not depend on the compression algorithm

Email Compatibility
Ciphertext contains arbitrary 8-bit octects
Some email systems may interpret some of them as control commands
Solution: base64 encoding (33% space overhead)

Segmentation
Transparent message segmentation and reassembly for very large messages
Segments mailed separately
Encrypted Email: Two Main Challenges

Public key authenticity

Assurance that a public key is correct and belongs to the person or entity claimed

   Ensure it has not been tampered with or replaced by an attacker

Public key discovery

How can we find the public key of a person/entity?

   Especially the very first time we need to contact them
PGP: Web of Trust

Decentralized trust model

- In contrast to the centralized hierarchical model of PKI
- Users create their own certificates

Users validate other users’ certificates, forming a “web of trust”

- No trusted authorities: trust is established through friends
- Adjustable “skepticism” parameters: number of fully and partially trusted endorsers required to trust a new certificate (1 and 3 for GnuPG)
- Key signing parties

Main problems

- Privacy issues: social graph metadata
- Bootstrapping: new users are not readily trusted by others
- When opinions vary, “stronger set” wins: impersonation through collusion/compromised keys
- Scalability: WoT for the whole world?
Hey, I just got home from the party with the IRC folks?

Yeah.

How was it?

I got too drunk, I screwed up, bad.

What happened?

There was a girl.

No idea who she was, don’t even know her name. I was too drunk to care.

And what, you slept with her?

No.

I signed her public key.

Shit, man.
Finding Public Keys

Public PGP key servers

- pgp.mit.edu
- keyserving.pgp.com

Cache certificates from received emails

Integration with user management systems (LDAP)

Ad-hoc approaches

- List public key on home page
- Print on business card
- Exchange through another medium on a case-by-case basis

Association with social profiles/identities

- keybase.io
MIT PGP Public Key Server

Help: Extracting keys / Submitting keys / Email interface / About this server / FAQ
Related Info: Information about PGP /

Extract a key

Search String: ____________________________ Do the search!

Index:  Verbose Index:  

- Show PGP fingerprints for keys
- Only return exact matches

Submit a key

Enter ASCII- armored PGP key here:
Biggest Issue: Usability

Non-trivial setup
   PGP: user is responsible for everything

Key management

Key revocation

Public key fingerprints

Poor mail client integration
   Can lead to catastrophic failures: e.g., Enigmail+Thunderbird silent encryption failure
   (Let alone key discovery and trustworthiness issues)

https://xkcd.com/1181/
Enigmail 1.7 is completely broken for my purposes.

Steps to reproduce the problem:

1) Write an email in TB.
2) Ensure "Force encryption" in Enigmail.
3) Ensure "Force signing" in Enigmail.
4) Recheck encryption and signing settings... OK.
5) Send the email.
6) Look at the received email. OOPS. It is NOT signed and NOT encrypted.

Sorry to say this so directly, but an encryption system, which CONFIRMS to the user in it's graphical user interface on two different places that it will encrypt AND THEN SENDS THE EMAIL WITHOUT ANY ENCRYPTION IN
PLAIN TEXT ... is just the BIGGEST IMAGINABLE CATASTROPHE.

Sorry for my profane language but there is simply no excuse for such
I’VE BEEN POSTING MY PUBLIC KEY FOR 15 YEARS NOW, BUT NO ONE HAS EVER ASKED ME FOR IT OR USED IT FOR ANYTHING AS FAR AS I CAN TELL.

MAYBE I SHOULD TRY POSTING MY PRIVATE KEY INSTEAD.
Runa A. Sandvik
@runasand

Swedish media org @Aftonbladet publishes its GPG private key for a second time (first time was in 2012):

Anders Nilsson @nilssonanders
Sweden's biggest newspaper #Aftonbladet includes their private key in guide to PGP mail them (via @zulin) bit.ly/1FfHACI
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: Mailvelope v1.8.0
Comment: https://www.mailvelope.com

xsFNBFm/2KMBEADbwToJm3BCVEl0eC22HgVEqNEDppXzuD2dgfKuy0M4tx2L
De7GkPjo6AOsw4y18bakLiidp5B0J/AR1vtJIDEmSOF9MRZtcV0UKyA5qV
c9Baf2nAicY7nezkjUWyLCVM60pqSHzo0Ewy2P2jxcT4vDghWmcgfV5X
R+duYld3LtVI+A/5jv326LB16bCNtS/tohW2TOlraNPoCtdhi84Z4tPcyP335
s/a/DZC+EoMD41x1ktyMzk1qEzFvNCs8aRUXy27L01VHeYm16UNKCoEHCn2
2yJxMBCnizzBZKUwcr6ysg97nnq633dn9mfV30PS3szAji0Hvmzg3B/Neo
gzy2daEUJDUBhia0c+9v9F3ZP0ooCjyS0rgyUm/2t3TThA+DnfsUBiq05U
2T0n8x2R1FwxzYNCTkbu5J0vPq8Bft13DsyTJD5Ldp52nqhpVb34erpwuk
qIK0TMru9B4E6q+cNFR32pN1Akj+Hob/TUJwCjPvju2/3gDwgqBh+Q1vC
Nm8vIGNQnZQ30WqNh/UFoh3RJP+WqnDq88NmqBq8I4aNV4u8MqoObd/zrtVX
kAwYHbIzLo925NjFyPuuxHIWCotKen18dzefBAEB81RjYUMCJQCQs+JG8
TJyEesNdK/q8HD5h1kCRZ5MHL1+Ra3z/1+F1iwARAQABZkR1ZG9iZSBQU0I3
VCA8cHnpcnRAYRWvYm/U-Y29tPSlbBewQOAgGLwUCwb/Yr3UJAbEzgAYLCq5G
AwIJEbADB8Kvh3YWBUIA0d0FqI1BAbkBAhsDAh4BADk2a//f+6FpZq4vMLI
Pz6T2QpQr/LX1Z7R7yB0vFWy0WNYx1U1leEoD5Q07HQT6NNMAO51JUs
wFvFOwTaUg16SHRoU1kGtS3RE0qG5HnX4DcBuhKmO5M568pu1z88WYO1N4Up
9V9Puuaue0U4o8ztVH5fBoqurytV805Cq4tUwfnJn7N40h0++291OPJ68B
6+kMuQyG4swmxzSh1ljqGMHc0s/BuI3W+n5+vlxLMN75jJCTjNXR+tGmstdm
RPBoCWo0o+2FwFW0ukLCJyUahp36n8HR13wpr2e0GdhKqRq7t3D4UeAgpqSs
-----END PGP PUBLIC KEY BLOCK-----
S/MIME

Based on standard X.509 certificates
  Analogous operation to TLS: trusted CA sign certificates
  Traditional PKI

Uses MIME to include cryptographic information in the message
  Multipurpose Internet Mail Extensions: extends the format of email messages to support binary attachments, and text in non-ASCII character sets

Works well within corporations
  Certificate distribution through the existing Active Directory infrastructure

Built-in support in most modern email clients
  Seamless interoperability between them
S/MIME Signed Message Example

From: alice@wonderland.com  
Date: Mon, 16 Nov 1998 19:03:08 -0600  
Subject: Message signed with S/MIME  
MIME-Version: 1.0  
Content-Type: multipart/mixed; boundary="simple boundary"

--simple boundary
Content-Type: text/plain; charset=US-ASCII  
Content-Transfer-Encoding: 7bit  
Content-Description: "cc:Mail Note Part"

Bob,

This is a message signed with S/MIME, so you can see how much overhead S/MIME signatures introduce. Compare this with a similar message signed with PGP.

Alice

--simple boundary
Content-Type: application/octet-stream; name="smime.p7s"  
Content-Transfer-Encoding: base64  
Content-Disposition: attachment; filename="smime.p7s"

MITQwYJKoZIhvcNAQcCoIITQDCCDADAQExCzAJBgUrDgMCGgUAAMgCSqGSIb3DQEHAaCCDwnw  
ggnGMI1MwDADAqECAhEQQQRa+DX0FHxJQVHQhFMA0GCSqGSIb3DQEBBAUAMGixETAPBgNVBAcT  
CEludGVybmV0c0RwFQYDVQQKEw5XZXJuU21nbiwgSW5jLjE0MCI1AUECmxMv NyMxVnpzZ24gQ2xh  
c5MgISBDQSA+Iu51G12aWRIYwWgU3Vic2NyW1jcJAvEFw05NzAxMjcwMDAwMDAxFw05ODAxMjcy  
MsU5NT1lNi1Bf3EwRAsGAIUEBwM2IW50ZXJuZXQxFrzAVBgNVBAMrTEl2ZCmlTAwJBIhBmMuMTQw  
MgYDVQQGEyJWTZXJpU21nbiriBDBGFzcyAxAENBICOgSW5kaXZpZHVhCmCTdWJyY3JpYmVJcM5YwRAYD

http://www.slideshare.net/rvenkatesh25/network-security-primer
End-to-End vs. Cloud-to-Cloud

IMAP: one of the oldest “cloud” services!
- Keep messages on the server
- Conveniently access them from multiple devices

Useful cloud-based email features
- Powerful search, collaborative SPAM filtering, …
- Need access to the plaintext (!) Gmail cannot index or filter encrypted messages

Tradeoff: privacy vs. convenience
- Active research on searchable encryption
Encrypted Webmail?

Several recent efforts have focused on transparently combining the convenience of webmail with PGP encryption

Is this really possible in a secure way?

JavaScript crypto is not a good idea

Secure JS code delivery?
Secure key storage?
Secure runtime (it’s a web browser!)?

Google end-to-end: implement cryptographic functionality as part of a browser extension

More control, but still not trivial
After initial excitement, it seems the effort has been abandoned
My Fellow Users,

I have been forced to make a difficult decision: to become complicit in crimes against the American people or walk away from nearly ten years of hard work by shutting down Lavabit. After significant soul searching, I have decided to suspend operations. I wish that I could legally share with you the events that led to my decision. I cannot. I feel you deserve to know what’s going on—the first amendment is supposed to guarantee me the freedom to speak out in situations like this. Unfortunately, Congress has passed laws that say otherwise. As things currently stand, I cannot share my experiences over the last six weeks, even though I have twice made the appropriate requests.

What’s going to happen now? We’ve already started preparing the paperwork needed to continue to fight for the Constitution in the Fourth Circuit Court of Appeals. A favorable decision would allow me resurrect Lavabit as an American company.

This experience has taught me one very important lesson: without congressional action or a strong judicial precedent, I would strongly recommend against anyone trusting their private data to a company with physical ties to the United States.

Sincerely,
Leodar Levison
Owner and Operator, Lavabit LLC

Defending the constitution is expensive! Help us by donating to the Lavabit Legal Defense Fund here.
Lavabit: “so secure that even our administrators can’t read your e-mail”

But they could, if they wanted to…

“Basically we generate public and private keys for the user and then encrypt the private key using a derivative of the plain text password. We then encrypt user messages using their public key before writing them to disk.”

“Because we need the plain text password to decrypt a user’s private key, we don’t support secure password authentication. We decided to support SSL instead (which encrypts everything; not just the password).”

http://highscalability.com/blog/2013/8/13/in-memoriam-lavabit-architecture-creating-a-scalable-email-s.html