PGP & S/MIME Email Security Demo

2025-10-25

PGP & S/MIME Email Security Demo

Prepared for Applied Information Assurance Projects
Date: October 2025

---

1. Executive summary

This project demonstrates end-to-end email protection using OpenPGP (GnuPG) and S/MIME in a controlled lab. It covers key generation, signing, encryption, verification, and trust models, emphasizing practical usability and operational controls for secure email exchange.

> Scope: Local lab mailboxes and client installations (Thunderbird). All keys and test accounts are laboratory-only and sanitized for publication.

---

2. Key concepts

- **OpenPGP / GPG** — user-centric keypairs, web-of-trust or keyserver discovery; flexible but requires user-driven trust decisions.
- **S/MIME** — X.509 certificates issued by a CA; easier enterprise provisioning but depends on trusted PKI.

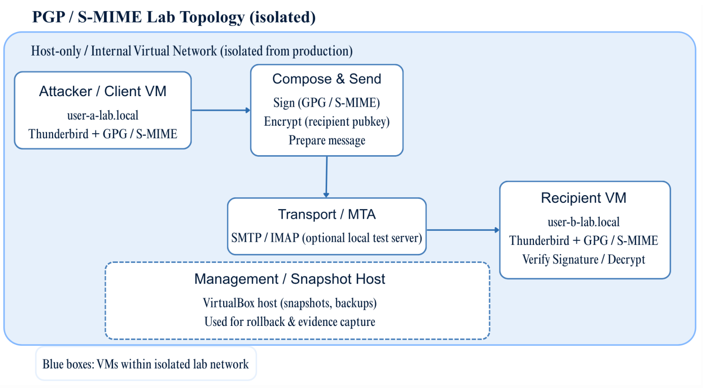
Both provide confidentiality and integrity for message bodies; note that standard headers (From, To, Subject) are usually not encrypted.

---

3. Lab topology & test setup

- Attacker/Client VM: `user-a-lab.local` (Thunderbird + GPG)
- Recipient VM: `user-b-lab.local` (Thunderbird + S/MIME cert or GPG key)
- Optional: Local SMTP/IMAP server for realistic delivery

The PGP / S-MIME lab was implemented in an isolated virtual network to simulate secure email exchange between two test users. As shown in Figure 1, the environment consists of three primary virtual machines: the *Attacker / Client VM* (`user-a-lab.local`), the *Transport / MTA* server, and the *Recipient VM* (`user-b-lab.local`). This configuration enables end-to-end testing of message signing, encryption, and verification while remaining completely detached from production systems.


PGP / S-MIME Lab Topology (isolated)
Host-only / Internal Virtual Network (isolated from production)
Attacker / Client VM
user-a-lab.local
Thunderbird + GPG / S-MIME
Compose & Send
Sign (GPG / S-MIME)
Encrypt (recipient pubkey)
Prepare message
Transport / MTA
SMTP / IMAP (optional local test server)
Recipient VM
user-b-lab.local
Thunderbird + GPG / S-MIME
Verify Signature / Decrypt
Management / Snapshot Host
VirtualBox host (snapshots, backups)
Used for rollback & evidence capture
Blue boxes: VMs within isolated lab network

PGP/S-MIME Lab Topology (isolated)

Figure 1 — PGP / S-MIME Lab Topology (isolated)

---

4. High-level workflows (sanitized)

To execute the demonstration, both participants generated key pairs, exchanged public keys or certificates, and used Thunderbird to compose, sign, and verify messages. The sanitized command sequence used for these steps is shown in Figure 2 (see Section 5), which details the OpenPGP and S/MIME operations performed through GPG and OpenSSL utilities. These commands were executed only within the isolated environment, using placeholder addresses and self-signed certificates.

GPG — key generation (example)

```
# interactive guided generation (safe)
gpg --full-generate-key

# or non-interactive (replace placeholders)
gpg --batch --gen-key <<'EOF'
Key-Type: RSA
Key-Length: 3072
Name-Real: Test User A
Name-Email: test-a@lab.local
Expire-Date: 1y
%commit
EOF
```

Export & share public key

```
gpg --armor --export test-a@lab.local > test-a.pub.asc
# recipient imports
gpg --import test-a.pub.asc
```

Sign & encrypt example (local files)

```
# sign a file
gpg --armor --sign --output message.sig --detach-sig message.txt

# encrypt for recipient (assumes recipient's public key imported)
gpg --armor --encrypt -r test-b@lab.local -o message.asc message.txt

# decrypt (recipient)
gpg --decrypt message.asc > message.txt
```

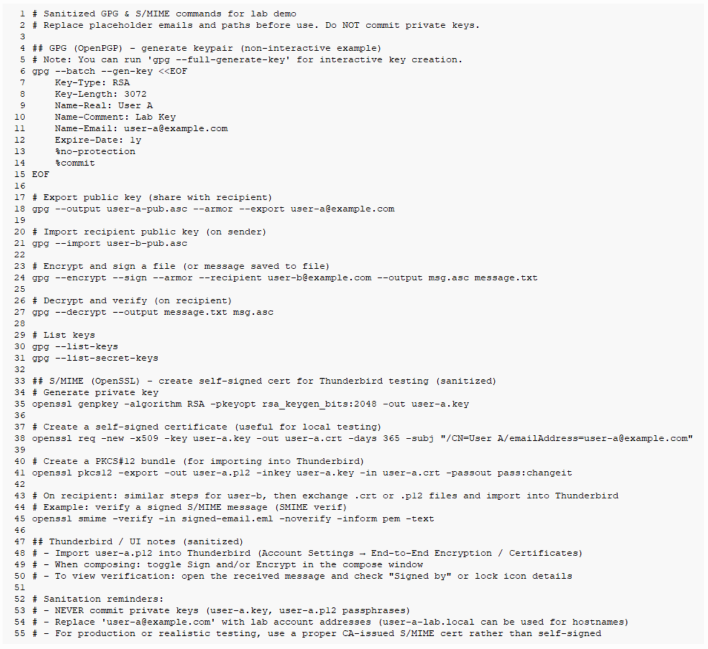
S/MIME — certificate creation (lab CA)

- Create an internal CA, issue client certificates, install into Thunderbird. (Do not publish private keys.)

---

5. Thunderbird client flows (visual)

- Generating/importing keys or importing S/MIME certificates.
- Composing a message: toggle Sign and/or Encrypt.
- On receipt: UI indicates Verified signature or Encrypted message; shows certificate details.



```
# Sanitized GPG & S/MIME commands for lab demo
# Replace placeholder emails and paths before use. Do NOT commit private keys.

## GPG (OpenPGP) - generate keypair (non-interactive example)
# Note: You can run 'gpg --full-generate-key' for interactive key creation.
gpg --batch --gen-key <<EOF
    Key-Type: RSA
    Key-Length: 3072
    Name-Real: User A
    Name-Comment: Lab Key
    Name-Email: user-a@example.com
    Expire-Date: 1y
    %no-protection
    %commit
EOF

# Export public key (share with recipient)
gpg --output user-a-pub.asc --armor --export user-a@example.com

# Import recipient public key (on sender)
gpg --import user-b-pub.asc

# Encrypt and sign a file (or message saved to file)
gpg --encrypt --sign --armor --recipient user-b@example.com --output msg.asc message.txt

# Decrypt and verify (on recipient)
gpg --decrypt --output message.txt msg.asc

# List keys
gpg --list-keys
gpg --list-secret-keys

## S/MIME (OpenSSL) - create self-signed cert for Thunderbird testing (sanitized)
# Generate private key
openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out user-a.key

# Create a self-signed certificate (useful for local testing)
openssl req -new -x509 -key user-a.key -out user-a.crt -days 365 -subj "/CN=User A/emailAddress=user-a@example.com"

# Create a PKCS#12 bundle (for importing into Thunderbird)
openssl pkcs12 -export -out user-a.p12 -inkey user-a.key -in user-a.crt -passout pass:changeit

# On recipient: similar steps for user-b, then exchange .crt or .p12 files and import into Thunderbird
# Example: verify a signed S/MIME message (SMIME verif)
openssl smime -verify -in signed-email.eml -noverify -inform pem -text

## Thunderbird / UI notes (sanitized)
# - Import user-a.p12 into Thunderbird (Account Settings → End-to-End Encryption / Certificates)
# - When composing: toggle Sign and/or Encrypt in the compose window
# - To view verification: open the received message and check "Signed by" or lock icon details

# Sanitation reminders:
# - NEVER commit private keys (user-a.key, user-a.p12 passphrases)
# - Replace 'user-a@example.com' with lab account addresses (user-a-lab.local can be used for hostnames)
# - For production or realistic testing, use a proper CA-issued S/MIME cert rather than self-signed
```

Sanitized commands

Figure 2 — Sanitized GPG & S/MIME Commands for Lab Demo

This figure shows the sanitized command-line workflow used for PGP and S/MIME testing. All email addresses and key data are placeholders. Do not commit private keys or passphrases.

---

6. Usability & trust considerations

- Key discovery: use secure channels / key fingerprints out-of-band. Avoid blindly trusting keyserver imports.
- Revocation & expiry: publish and test revocation procedures (revocation certificates for GPG).
- Certificate trust: S/MIME in enterprises is manageable via internal CA or MDM distribution. Beware of CA compromise risks.

7. Operational guidance

- Backup private keys securely (encrypted storage, HSMs for enterprise).
- Use passphrases and consider smartcards or hardware tokens for private key protection.
- Train users on fingerprint verification, phishing resistance, and attachment handling.
- If deploying at scale, automate certificate provisioning (SCEP/AD integration or MDM).

8. Limitations & privacy notes

- Email metadata (subject, recipients) is typically visible and must be considered in threat models.
- Encrypted attachments and inline vs PGP/MIME compatibility can affect interoperability.
- Publishing examples uses sanitized keys and truncated fingerprints to avoid exposure.

9. References & resources

- GnuPG documentation: https://gnupg.org
- Thunderbird encryption guide: https://support.mozilla.org
- RFC 8551 (OpenPGP), RFC 5751 (S/MIME)
- NIST guidance on cryptographic key management

Download Report (PDF)

```
```