Learn PGP

SIPB Cluedump, 19 October 2016

Anish Athalye (aathalye), Merry Mou (mmou), Adam Suhl (asuhl)


HOW TO USE PGP TO VERIFY THAT AN EMAIL IS AUTHENTIC:
LOOK FOR THIS TEXT AT THE TOP.
REPLY
-----BEGIN PGP SIGNED MESSAGE-----
HASH: SHA256
HEY,
IF IT'S THERE, THE EMAIL IS PROBABLY FINE.

Overview

1. Theoretical PGP / Intro to Security
2. Practical PGP - Installation, Usage, Demo
3. Do it yourself!

THEORETICAL PGP!

The Problem

I want to send and receive email from Bob and make sure:

- No one else can read OR modify its contents
- I trust that Bob is Bob, and Bob trusts that I am me

The Solution

Use **public key cryptography** to achieve **end-to-end encryption**

Public Key Cryptography (tl;dr edition)

- Each person has a (private key, public key) pair
- Two basic sets of operations:
- Encrypt / Decrypt

```
encrypt(plaintext, public key) -> ciphertext
decrypt(ciphertext, private key) -> plaintext
```

- Sign / Verify

```
sign(data, private key) -> signature
verify(data, signature, public key) -> ok?
```

Trust Models

- How do you know that you can trust a key?
- Centralized (e.g. HTTPS)
    - Certificate Authority hierarchy
- Decentralized (e.g. PGP)
    - Web of Trust

Finding and Trusting Keys

- **Public key certificate server**: databases of public keys of people (of unverified identities), so that you can send encrypted messages (signed with their public key) to them.
- **Your keyring**: list of public keys whose owners' identities are verified by you
- **Web of trust**: "web" of public keys whose owners' identities are verified, usually via in person contact (e.g. PGP key signing party)

Using your Key

- Signing someone else's key to show that you trust that their key belongs to them (e.g. during a PGP key signing party)
- Signing your software release so that people can verify that you wrote it and that it hasn't been modified
- Signing your email and attaching the signature with it so that the receiver can verify that you wrote it and that it hasn't been modified
- Encrypting your email and sending the ciphertext so that no one else can tamper your message (often you'll also sign the email)
- Encrypting and signing your backups in Amazon S3 to make sure that Amazon can't mess with your data and can't read your secret files

Limitations of PGP

(aka, ways the NSA can still get you)

(aka, why PGP is only "pretty good" privacy)

- Endpoint security
- Metadata (e.g., subject line)
- User error
- Scalability

PRACTICAL PGP!

PGPGP

- OpenPGP - the standard https://tools.ietf.org/html/rfc4880
- PGP - 1st implementation of OpenPGP standard, now commercial proprietary software owned by Symantec
- GPG - GNU Private Guard, open source implementation of the OpenPGP standard, the thing everyone uses

(But in reality, people use all three terms interchangeably)

Installing PGP: Mac OS

- Graphical clients also available (https://gpgtools.org/)
- Or, for CLI client only, `brew install gpg2`
- EFF tutorial on installing https://ssd.eff.org/en/module/how-use-pgp-mac-os-x
- With Apple Mail: https://gpgtools.org/

Installing PGP: Linux

- `apt install` gnupg (or "pacman -S gnupg", etc.)
- https://ssd.eff.org/en/module/how-use-pgp-linux

Installing PGP: Windows

- ¯_(ツ)_/¯
- https://ssd.eff.org/en/module/how-use-pgp-windows
- How to download GPG4Win + Thunderbird integration
- tested, once this is installed, the CLI client works (in `cmd.exe`)

Basic Operations

- Create your key
  - `gpg --gen-key`
- Figure out your key fingerprint
  - `gpg --list-secret-keys --fingerprint`
- Upload key to keyserver
  - `gpg --keyserver pgp.mit.edu --send-key <fingerprint or key id>`

Basic Operations (continued)

- Download a key from the keyservers
  - `gpg --recv-key <fingerprint or key id>`
- Sign a blob of text
  - `gpg --clearsign`
- Sign and encrypt a blob of text
  - `gpg --sign --encrypt --armor`
- Decrypt a blob of text
  - `gpg --decrypt`
- Verify a signature
  - `gpg --verify`

Signing keys

- Verify the person's identity (e.g., with photo ID)
- Download their key from a keyserver:
  - `gpg --keyserver pgp.mit.edu --recv-key <their fingerprint or key id>`
- Sign the key
  - `gpg --sign-key`
- Check with the person that the fingerprint you see is correct before saying "yes"!
- Send their signed key to the keyserver
  - `gpg --keyserver pgp.mit.edu --send-key <their fingerprint or key id>`

Encrypt/decrypt your email

- Thunderbird: Install Enigmail
    - https://ssd.eff.org has tutorials for setting up for setting up Enigmail with Thunderbird
- Apple Mail
    - https://gpgtools.org/
- Gmail:
    - Write email in a text editor
    - Encrypt and sign the email
    - Paste it into gmail

Other key maintenance stuff

- Backing up your key (`.gnupg` directory)
- Key expiration
- Revocation certificates (make one!)
- Key comments

Resources

- Slides at git.io/learn-pgp
- GPG Best Practices https://riseup.net/en/security/message-security/openpgp/gpg-best-practices
- EFF Surveillance Self Defense https://ssd.eff.org/
- Come to SIPB's PGP Key Signing and Movie Night Party on 10/27 7pm in the SIPB office (W20-557)!
    - Bring your laptop and a government ID