PGP
(Pretty Good Privacy)
1. PGP Overview
2. PGP Design: Sending and Receiving Process
3. PGP Usage in Practice
4. PGP Variants
5. PGP Lessons
6. References
Overview

What is PGP?

• **PGP (Pretty Good Privacy)**
  - A software package that performs cryptographic tasks related with email systems
  - Created by Phil Zimmermann in 1991, named Pretty Good Privacy
  - Provides security services and privacy using a series of security techniques including:
    1. Hashing for integrity
    2. Data compression
    3. Symmetric cryptography for confidentiality
    4. Asymmetric cryptography for both authentication and non-repudiation
  - Currently acquired by Symantec Group since 2010
    1. Still source code available (v10) for peer review
    2. PGP variants available such as *OpenPGP, GPG, iPGMail, APG*, and so forth
**Sending Process (Sender = Alice)**

- **Sender (Alice)**
  - **Sender's Private Key** (Session Key)
  - **Hashing** \( H(m) \)
  - **Plaintext** \( m \)
- **Secret Key** (Secret Key)
- **Receiver’s Public Key**
  - **Secret key generation** is done with a pseudo-random generator using IDEA as a building block
- **Ciphertext** encrypted with **Secret key**
  - **Hash value signed by Sender’s private key**
  - **Encrypted Message**
- **Each key pair is generated by RSA mechanism**
  - \( E_{K_{PuB}}(K_s) \) || \( E_{K_s}(m) \) || \( E_{PrA}(H(m)) \)

---

PGP – Pretty Good Privacy
Receiving Process (Receiver = Bob)

1. Extract secret key: \( D_{K_{PrB}}(E_{K_{Pub}}(K_s)) = K_s \)
2. Get the message: \( D_{K_s}(E_{K_{s}}(m)) = m \)
3. Compare the hash: \( D_{K_{Pub}}(E_{K_{PrA}}(H(m))) = H(m) \)

CONFIDENTIALITY with Encryption

AUTHENTICATION & NON-REPUDIATION with Digital Signature

INTEGRITY with Hash
PGP Usage in Practice

- **Mailvelope** in Google extension

  A. Import the public key for the person who you send an email

  B. Check out the imported key ring

  C. Compose the message and then encrypt and send it!


PGP Variants

- **OpenPGP**
  - Became most widely used email encryption standard for nearly all over the world
  - Defined by OpenPGP Working Group of IETF, and proposed Standard [RFC 4880](https://tools.ietf.org/html/rfc4880)

- **GPG (GnuPG or Gnu Privacy Guard)**
  - Complete and free implementation of the OpenPGP standard from GNU’s Project
  - Gpg4win and GPGTools for Windows and Mac OS X platform respectively

- **PGP tools for smart phones**
  - iPGMail for iPhone
  - APG(Android privacy guard) for Android

- **Browser extensions or add-ons**
  - Mailvelope for both Google Chrome and Mozilla Firefox
  - OpenPGP.js (Javascript implementation), known as GPG4Browsers earlier
PGP Considerations, Pitfalls and/or Lessons

- PGP is a great tool to communicate each other secretly, but it could be misused by malicious users including criminals.

- Since there is no way to know whose public key belongs to whom, PGP needs to have the 3rd party to confirm it such as public key infrastructure – PKI. Later version supports this.

- When private key is stored in a personal disk, PGP uses the password to encrypt it. This means public key cryptography might depend on a user’s keystroke. As whole security is as strong as the weakest link, this scheme can be compromised by an elaborate key logger to gain private key. (e.g. United States vs Scarfo in 2001) Therefore, it is very important to maintain a private key securely. This shortcoming is not PGP-only issue but entire PKI.

- To use PGP technique, both ends should install the software implemented standard specification. This means a security bug and/or deliberate backdoor might lead a critical leakage to be exploited.
References

• PGP

  - http://www.openpgp.org/
  - http://www.gnupg.org/
  - http://www.symantec.com/encryption
  - http://www.gpg4win.org/
  - http://ipgmail.com/
  - http://www.thialfihar.org/projects/apg/
  - http://www.ietf.org/rfc/rfc4880.txt
  - http://gpg4browsers.recurity.com/
  - https://webpg.org/
  - http://epic.org/crypto/scarfo.html