Cybersecurity: Pretty Good Privacy

Ozalp Babaoglu

ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

What is PGP?

- Freeware (at least in its initial version) developed by Philip R. Zimmermann
- Based on cryptographic techniques, both symmetric and asymmetric (as well as hybrid)
- Used mainly to keep information (files, email) hidden from "indiscreet eyes"
- Implements certificates
- Implements digital signatures

A bit of history

- In 1991, an anticrime legislation is proposed in the US senate that requires producers of cryptographic tools to include "trap doors" in their products (predecessor to the Clipper Chip)
- Before the law is approved, Philip R. Zimmermann develops and distributed PGP

A bit of history

- Soon after, Zimmermann is accused of violating patent laws because PGP makes use of RSA technology
- Furthermore, the US government accuses him of having violated ITAR (*International Traffic in Arms Regulations*) that places trade restrictions on cryptographic technologies as well as software that implement them
- There is public outcry and a legal defense fund is established
- The charges are dropped in 1996

More recent history

- During the FBI investigation, Zimmermann and his team work on new versions of PGP
  - PGP 3 introduced public-key algorithms (e.g., DSA —*Digital Signature Algorithm*) not restricted by patents
- 1996 — "PGP Corporation" is born to commercialize PGP
- 2010 — PGP Corporation acquired by Symantec
- 2019 — Symantec acquired by Broadcom Inc.

More recent history

- In 1997, Zimmermann proposes the standard "OpenPGP" to the IETF
- Last version in 2007 (RFC 4880)
- GPG (*Gnu Privacy Guard*) is the implementation of OpenPGP by the Free Software Foundation
- Many other implementations exist
  - Perl Crypt::OpenPGP
  - OpenPGP.js
  - PHP OpenPGP
  - Java Portable PGP
  - ...

More recent history

- Main factors contributing to PGP success:
  - Freely available worldwide
  - Relying on "secure" algorithms
  - Not developed or controlled by governments or other institutions
  - Standardized (OpenPGP)
  - Widely implemented and supported, multi-platform

PGP functions

- Generation and management of keys
- Encryption/Decryption and Sign/Verify of digital documents
- Creation of "self-decrypting archives (SDAs)
- Permanent removal of files and directories from disk
- Creation of VPN (Virtual Private Network)

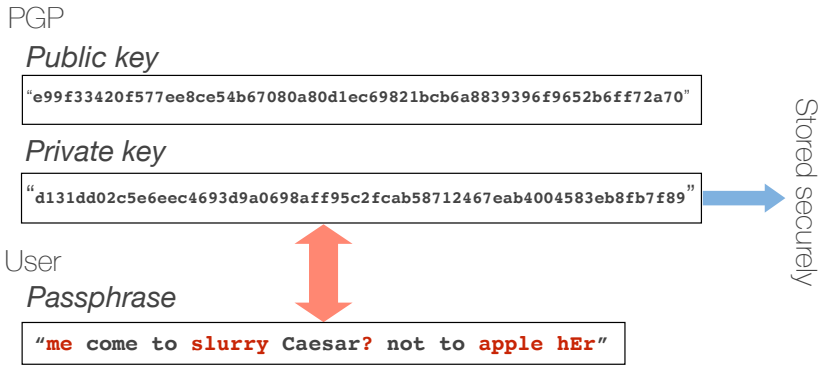
Keys in PGP

- PGP implements both secret-key and public-key protocols
- Informally:
  - An 80-bit secret key is equivalent to a 1024-bit public key
  - A 128-bit secret key is equivalent to a 3000-bit public key
- 56-bit (or less) secret keys are considered insecure
- 128-bit secret keys are considered secure today but may become insecure in the future (with quantum computers)
- 256-bit secret keys can be considered secure

PGP Keyrings

- PGP stores the keys in two files called ***keyrings***, one for public keys and one for private keys
  - users' keys are stored in the ***user keyring***
  - public keys of recipients are stored in the ***public keyring***
- PGP private keys are stored in encrypted form
  - Each private key is encrypted using a hash of a ***pass phrase*** as the secret key
- A pass phrase is longer than a password, and thus more secure (at least in theory)

Passphrase



Passphrase: serves when the user has to refer to her private key. Easy to remember. Must not be forgotten

PGP: Encryption

- Sender A:

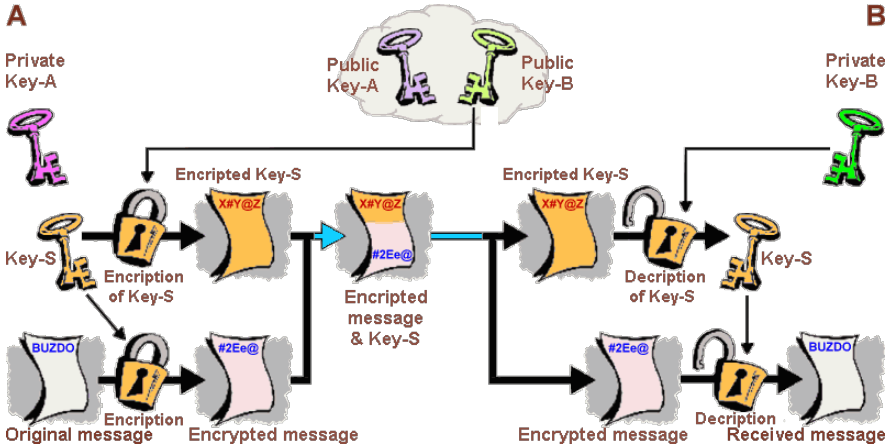
1. generate a secret key K_S that will serve as the ***session key***
2. encrypt the document using K_S
3. obtain the public key $K_B[pub]$ of the receiver
4. encrypt K_S using $K_B[pub]$
5. send document encrypted with K_S and the secret key encrypted with $K_B[pub]$

PGP: Decryption

- Receiver B:

1. decrypt the received cryptogram using own secret key $K_B[priv]$ to obtain the (secret) session key K_S
2. use the session key K_S to decrypt the document

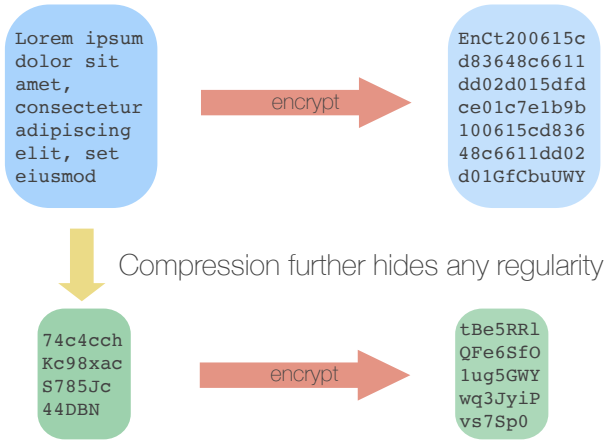
PGP: Overall scheme



Compression

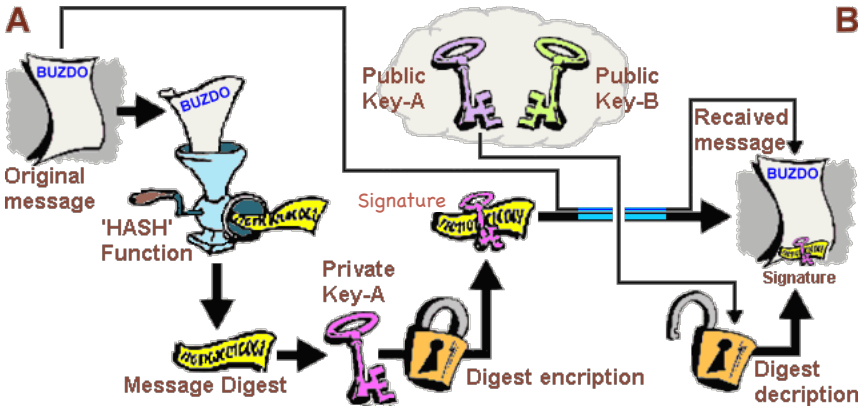
- PGP uses compression to further hide the encoded message
- Before encrypting, the plaintext message is *compressed*
- After decrypting, the resulting message is *decompressed*

Compressione



- Advantages: statistical attacks more difficult, smaller dimension

PGP: Digital signatures



Algorithms used in PGP

- Private-key (symmetric):
  - CAST
  - Triple-DES
  - IDEA
  - Two Fish (AES)
- Public-key (asymmetric):
  - RSA
  - ElGamal (in more recent versions of PGP o GPG)
  - DSA (Digital Signature Algorithm)
- Hash:
  - SHA1 (Secure Hash Algorithm)

PGP: Private-Public key pair example

```
********************************************************************
* WARNING: This file is a backup of your secret key. Please keep it in *
* a safe place.                                                    *
********************************************************************
The key backed up in this file is:
pub   1024D/3D0C8FF8 2009-04-26
      (key_size_in_bits/IDENTIFIER Creation-date-key)
      Key fingerprint = 98FF 763A 6AF6 AEF6 1B33  C8BC F719 447B 3D0C 8FF8
      (checksum of certain parameters of the public key)
uid                  Drago Radic (For the purposes of 'alphabet') <tupko@buzdo.com>
sub   1024g/A133AFBD 2009-04-26

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.10 (MingW32)
mQGiBEn0MvMRBACg5QL4VYjutTntcz0Xgnvy4MdF7yXofTJsg5faGVkWsEW/ZwNe
QoVIZrWdls2wiXHkxLQ/1ehcMeo5bJdkthJYuw1jr2noDsC9zdhtw68X9vDbQdwi
A7S+FLB88HsNbXfueKgeyhGXh5qlmBOzXNdwi/MR5X8hft1iimem4JaMhwCgynHq
Jcu68U5HmZg=
=EbV+
-----END PGP PUBLIC KEY BLOCK-----

-----BEGIN PGP PRIVATE KEY BLOCK-----
Version: GnuPG v2.0.10 (MingW32)
lQHhBEn0MvMRBACg5QL4VYjutTntcz0Xgnvy4MdF7yXofTJsg5faGVkWsEW/ZwNe
QoVIZrWdls2wiXHkxLQ/1ehcMeo5bJdkthJYuw1jr2noDsC9zdhtw68X9vDbQdwi
PQyP+NuBAJ4m0clvfiiVVy96quvbDrq/ajgLJgCZAWgQ/jC7wB4Hd3XGWiGuysE2
j6M=
=eYIR
-----END PGP PRIVATE KEY BLOCK-----
```


Anatomy of PGP Certificates

- PGP version number
- Public key of U, properties of the key (length, creation algorithm, date of creation, validity of the key, …)
- Identity information for U: name, last name, place and date of birth, photo, …
- Self-signature: public key of U is signed using the private key of U
- Preferred symmetric ciphers (Es: CAST, IDEA, Triple-DES)
- Other signatures …

Certificate formats in PGP

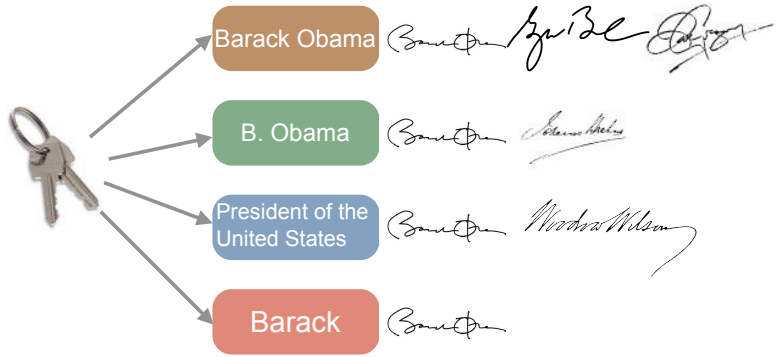
- PGP recognizes two different certificate formats:
  - Certificates in native PGP format
  - Certificates in X.509 format (which is an international standard: in theory, all applications must handle them but in practice the fact that there are many variants of X.509 makes this goal difficult to achieve)

PGP vs X.509

PGP	X.509
No Registration Authority	Registration Authority
Self-signed certificates	Certificates signed by a CA
Multiple identities	Single identity
Multiple signatories to attest the validity of the certificate	Single signatory to attest the validity of the certificate

PGP Certificates

- May contain multiple public key/identify pairs, each signed multiple times



Trust and Validation

- Unless you receive a certificate directly from its owner, you have to rely on on other factors that it is *valid*
- You ***trust*** people, PGP ***validates*** certificates
- Trust models
  - Direct trust
  - Hierarchical trust
  - Web of trust

Trust in X.509

- Based on chains of trust among entities that are reputed to be CAs (hierarchical trust)
- The (blind) trust we place on root-level CAs must be acquired through reputation, experience, operational competence and other non-technical aspects
- Anyone claiming to be a CA must be a trusted entity and we must believe that it is secure and correct

Trust in PGP

- In PGP, any user can act as a CA and sign the certificate of another user (becomes an "introducer" of that key)
- You consider a certificate to be *valid* only if you *trust sufficiently one or more of the introducers* of the certificate
- "Web of trust"
- Non symmetric
- Non transitive

Trust in PGP

- You trust yourself (reflexive)
- You assign one of the following levels of trust to other subjects:
  - ***Complete*** trust
  - ***Marginal*** trust
  - ***Untrusted*** (no trust)
- If you assign ***complete*** trust to someone, this effectively makes them a CA

Validation in PGP

- PGP considers a certificate "valid" based on
  - how other subjects judged the certificate
  - the level of trust you have assigned to those subjects
- You add your signature only to those certificates that you can verify in person
- This creates a decentralized, dynamic "reputation system"

Validation in PGP

- PGP will consider a certificate ***valid*** only if it has either
  - One signature from a ***completely*** trusted user, or
  - Two (or more) signatures from ***marginally*** trusted users

SDAs and PGP Shredder

- A *self-decrypting archive* (SDA) is an archive that can be opened by anyone, even those who have not installed PGP
  - An SDA includes an executable (to open the archive)
  - SDAs can be only opened on the same platform on which they were created
- SDAs can only be protected by pass phrases
  - a secure way to communicate the passphrase is needed!
- PGP *shredder* is a tool to securely delete files