08 - Pretty Good Privacy

Funzioni:

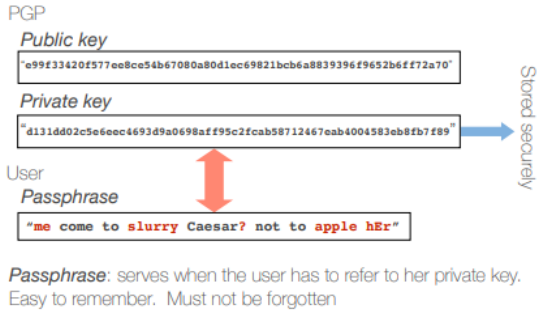
- Generazione e gestione di chiavi
- Criptazione/decriptazione e firma/verifica di qualsiasi documento digitale
- Creazione dei "self-decrypting archives" (SDAs)
- Rimozione permanente di file e directory dal disco
- creazione di VPN

Chiavi in PGP

PP implementa protocolli sia a chiavi segrete che a chiavi pubbliche.
una chiave segreta a 80 bit è uguale a una pubblica a 1024 bit, una segreta a1 28 bit = 3000 bit pubblica
chiavi da 56 bit o meno sono considerate non sicure
128-bit sono sicure oggi ma non si sa nel futuro
1256-bit sono sicure

PGP salva le chiavi in due file chiamati **keyrings** uno per le chiavi pubbliche e uno per quelle private

- Le chiavi degli utenti sono salvate in **user keyrings**
- le chiavi pubbliche sono slavate nei **public keyrings**
  PGP private keys sono salvate in forma criptata usando un hash con una **pass phrase** come chiave segreta.
  Una pass phrase è più lunga di una password quindi più sicura.



PGP: Encryption

Sender A:

1. Genera una chiave segreta K_S che serve come **session key**
2. cripta il documento usando K_S
3. ottiene la chiave pubblica $K_B[\text{pub}]$ del ricevente
4. cripta K_S usando $K_B[\text{pub}]$
5. Invia il documento criptato con K_S e la chiave privata criptata con $K_B[\text{pub}]$

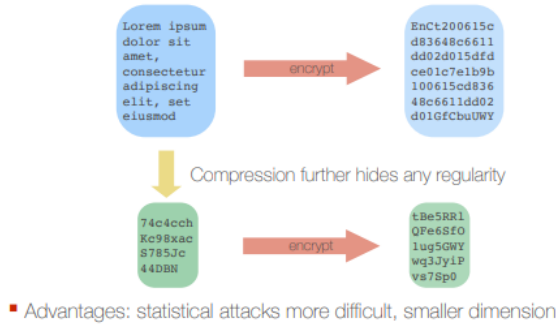
Ricevente B

1. decripta il cryptogrammo ricevuto usando a sua chiave secreta $K_B[\text{priv}]$ per ottenere la (segreta) session key K_S
2. usa K_S per decriptare il documento

Compression

PGP usa la compressione per nascondere ancora di più il messaggio codificato
Prima di criptare il plaintext esso viene compresso

Dopo la decriptazione il messaggio estratto viene decompresso



Algoritmi usati in PGP:
Chiave privata (simmetrica):

- CAST
- Triple-DES
- IDEA
- Two Fish (AES)

Chiave pubblica:

- RSA
- ElGamal
- DSA

Hash:

- SHA1

Anatomia di un certificato PGP

```
********************************************************************
* 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+NuBAJ4m0c1vfiiVVy96quvbDrq/ajgLJgCZAWgQ/jC7wB4Hd3XGWiGuysE2
j6M=
=eYIR
-----END PGP PRIVATE KEY BLOCK-----
```

PGP version number
Chiave pubblica di **U**, proprietà della chiave (lunghezza, algoritmo di creazione, data, validità della chiave)
Identity information di **U**: nome cognome, luogo e data di nascita ...
Self-signature: chiave pubblica di U è firmata usando la chiave privata di U
Cifrario simmetrico preferito(Es: CAST, IDEA, Triple-DES)
Altre firme...

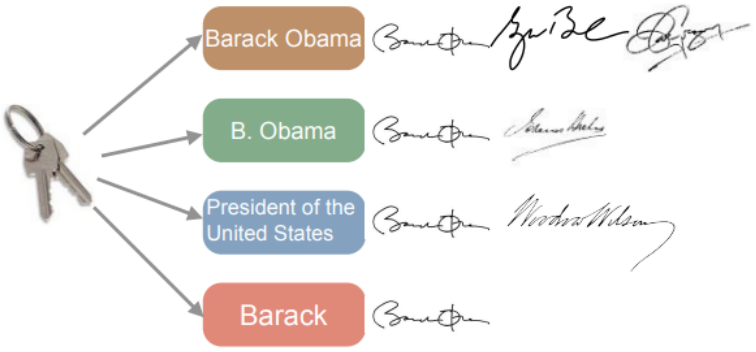
PGP riconosce due formati di certificato:

- In formato PGP nativo

- In formato X.509 (standard internazionale)

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

Potrebbe contenere più coppie key/identify ognuna firmata più volte:



Almeno che non si riceva il certificato dal proprietario si deve fare affidamento ad altri fattori per ritenerlo valido

- Tu ti fidi delle persone, PGP valida i certificati.
- Trust models: Direct trust, Hierarchical trust, Web trust.

Trust in X.509

basato su una catena di trust tra entità che sono reputate ad essere CA (Hierarchical trust)
La (cieca) fiducia che si esprime al CA al livello root deve essere guadagnata tramite esperienza, competenza...
Chiunque affermi di essere un CA deve essere un entità trusted.

Trust in PGP

In PGP ogni utente può essere un CA e firmare il certificato di altri utenti (diventa un introducer di quella chiave)
Si considera un certificato valido solo se *ci si fida abbastanza* di uno o più introducers di quel certificato.
Web of trust, no simmetrie, no transitività

- Tu ti fidi di te stesso (riflessivo)
- Assegni uno dei seguenti livelli di fiducia agli altri soggetti:
  - Fiducia **completa**
  - Fiducia **marginale**
  - **Inaffidabile**
- se assegni fiducia completa a qualcuno, lo si rende effettivamente un CA

Validation in PGP

PGP considera valido un certificato basandosi su:

- come gli altri soggetti giudicano il certificato
- il livello di fiducia che l'utente ha assegnato a quei soggetti
- Aggiungi la tua firma solo a quei certificati che puoi verificare di persona
- Questo crea un sistema di certificati decentralizzato e dinamico.

Quindi PGP considera valido un certificato quando:

- è firmato da un utente su cui si ha **completa fiducia** oppure
- due o più firme di utenti di cui si ha **fiducia marginale**

SDAs and PGP Shredder

Un self-decrypting archive (SDA) è un archivio che può essere aperto da chiunque, anche chi non ha installato PGP

- Un SDA include un eseguibile (per aprire l'archivio)
- SDA può essere aperto solo sulla stessa piattaforma in cui è stato creato
- SDAs sono protetti solo da pass phrase
- PGP shredder è uno strumento per l'eliminazione sicura di file