GPG

Pretty Good Privacy / Gnu's Privacy Guard

Fatih Turkmen

Slides borrowed from Frank Brokken

Today

- Problems with clear text information storage or exchange (mostly in the context of email).

- GPG/PGP features and use, and some issues.
  - OpenPGP (not PGP which is trademark of Symantec)

Dangers of Common Practices

- Clear-text information storage or exchange
  - No guarantee for confidentiality
  - No guarantee for integrity

- GPG/PGP you to encrypt and sign your data and communications.

PGP/GPG

- 1991: Phil Zimmermann published PGP
- 1997: PGP was deemed `legal', after years of investigation by the US government of presumed criminal activities by Phil et al.
- Some links:
  - http://www.philzimmermann.com
  - http://www.gnupg.org (/gph/en/manual.html)
  - http://www.pgpi.org

❗ What do PGP and GPG stand for?

- Were you paying attention ☺ ?

PGP/GPG

- Consider the following by Phil Zimmerman:

> *When encryption is outlawed, only outlaws have encryption*.

PGP/GPG

- The PGP/GPG Public Key Infrastructure (PKI)
  - Publicly (widely) known public key
  - Privately kept private (or secret) key

❗ What are the “key” ingredients of PKI?

PGP/GPG

- The PGP/GPG Public Key Infrastructure (PKI)
  - Publicly (widely) known public key
  - Privately kept private (or secret) key
  - Passphrase protects access to the private key

PGP/GPG

- The PGP/GPG Public Key Infrastructure (PKI)
  - Publicly (widely) known public key
  - Privately kept private (or secret) key
  - Passphrase protects access to the private key
  - Well-defined public key infrastructure is available
    - Many key servers available
    - Many clients available

PGP/GPG

- The PGP/GPG Public Key Infrastructure (PKI)
  - Publicly (widely) known public key
  - Privately kept private (or secret) key
  - Passphrase protects access to the private key
  - Well-defined public key infrastructure is available
  - Software is free

PGP/GPG

- The PGP/GPG Public Key Infrastructure (PKI)
  - Publicly (widely) known public key
  - Privately kept private (or secret) key
  - Passphrase protects access to the private key
  - Well-defined public key infrastructure is available
  - Software is free
  - No known practically feasible way to subvert

PGP/GPG - features

Allows for:

- **Confidentiality**:
  - Encrypt your own sensitive data
  - Only the intended recipient(s) can read the information
- **Integrity**:
  - Digital signatures are invalid when the document is modified

PGP/GPG – features

Features:

- **Non-repudiation**:
  - Verify the authenticity of the sender
- **Web of Trust**:
  - The signer/sender of the information is known.

❗ What is non-repudiation?

PGP/GPG – software

- The gpg software may be downloaded from
  - http://www.gnupg.org/

- For Unix/Linux all standard distributions offer the software (or you can compile it yourself)

GPG – How?

Summary of actions with GPG

- **Once**
  - Create your own key-pair (private/public)
  - Send it to key servers
- **Regularly**:
  - Encrypt information using someone’s (i.e., recipient) public key
    - Ensuring the confidentiality of the information
  - Sign information using your own secret key – Ensuring the integrity of the information
    - Yields non-repudiation
- **Every now and then**:
    - Sign someone's public key, building your `web of trust'

GPG – How?

Check Configuration settings

- See here for Ubuntu: https://riseup.net/en/security/message-security/openpgp/best-practices#putting-it-all-together

Among others;

- Find the file gpg.conf (if it does not exist, create), make sure it contains:

```
personal-digest-preferences SHA256
cert-digest-algo SHA256
default-preference-list SHA512 SHA384 SHA256 SHA224 AES256 AES192 AES CAST5 ZLIB BZIP2 ZIP Uncompressed
```

One line!

GPG – How?

Once

- Create your own key-pair (private/public)
- Before you do, see for passphrases:

https://www.iusmentis.com/security/passphrasefaq/

```
gpg  --gen-key
```

- The secret key is protected with a **passphrase**
  - Protection by length, avoid plain biographical data
  - If you lose your passphrase, you're lost...

Also Note: GnuPG actually uses a signing-only key as the master key, and creates an encryption subkey automatically.

Read here: https://wiki.debian.org/Subkeys

GPG – How?

- Once
  - Create your own key-pair (private/public)
  - Two entries are created (if you use gpg key generation):
    - private-keys-v1.d - contains your secret key(s)
    - pubring.kbx - contains all your public keys

Good to know (Revocation Certificate!!):

```
gpg --output ~/gpgrevocation.crt --gen-revoke …
```

GPG – How?

- Once
  - Sending your keys to keyservers

```
gpg --send-keys --keyserver http://pgp.surfnet.nl/ …
```

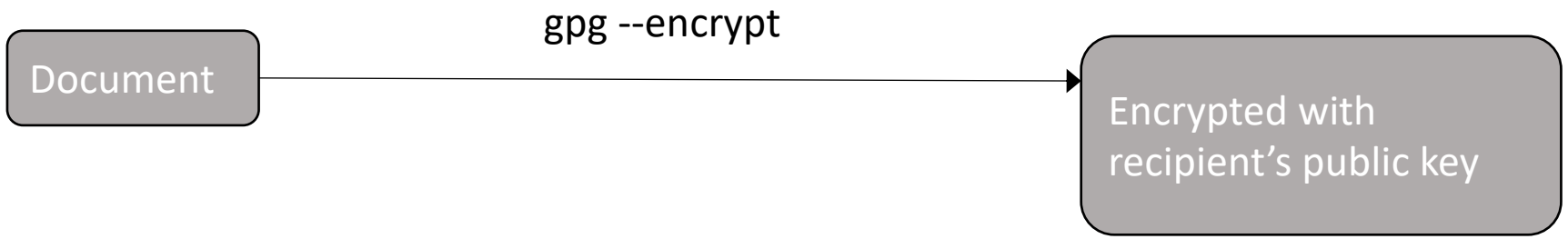
  - To see your key fingerprint/s

```
gpg –fingerprint …
```

GPG – How?

- Daily
  - Encrypting Information:

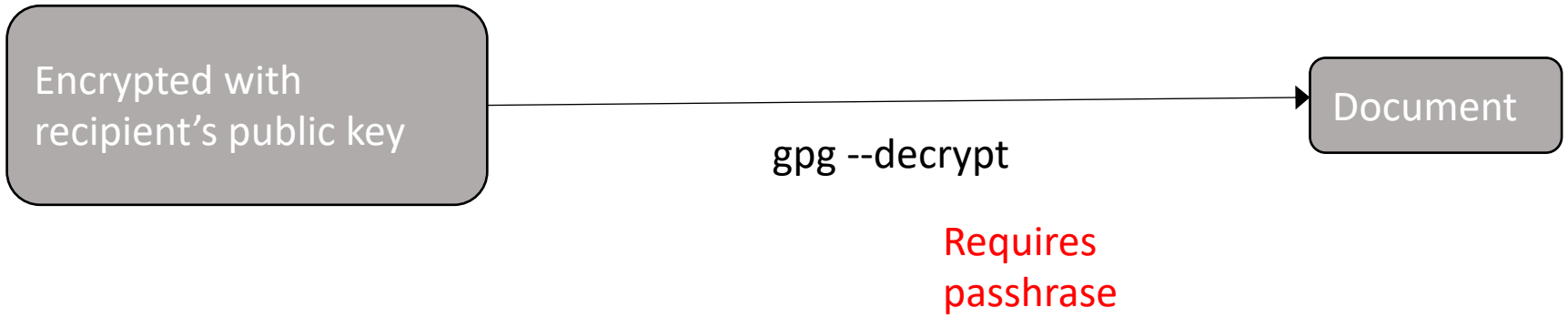
```
gpg --encrypt < original > encrypted
```



GPG – How?

- Daily/Regularly
  - Decrypt Information:

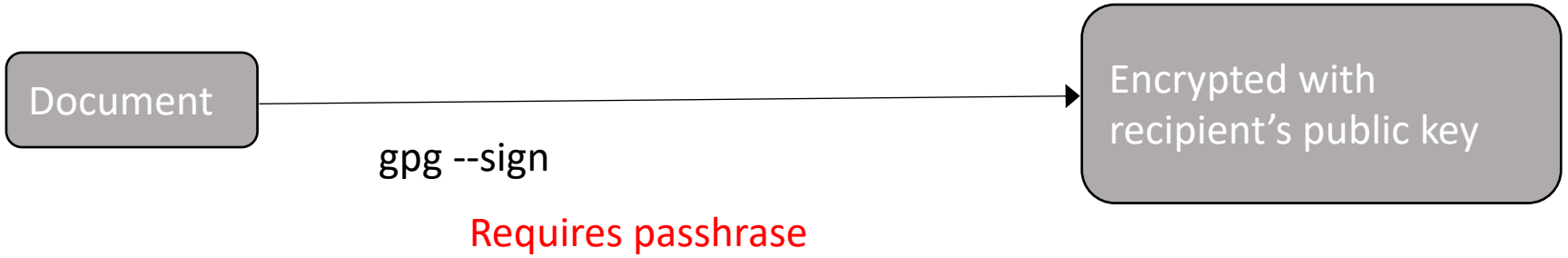
```
gpg --decrypt < encrypted > original
```



GPG – How?

- Regularly
  - **Sign** Information (requires your secret key)
  - Flavors
    - Plain signature (implies encryption (with. Sec. key)): **--sign**
    - Clear text signature : **--clearsign**
      - Creates *filename.asc*
    - Detached signature: **--detach-sign**
      - Creates *filename.sig*

gpg --sign < original > encrypted



GPG – How?

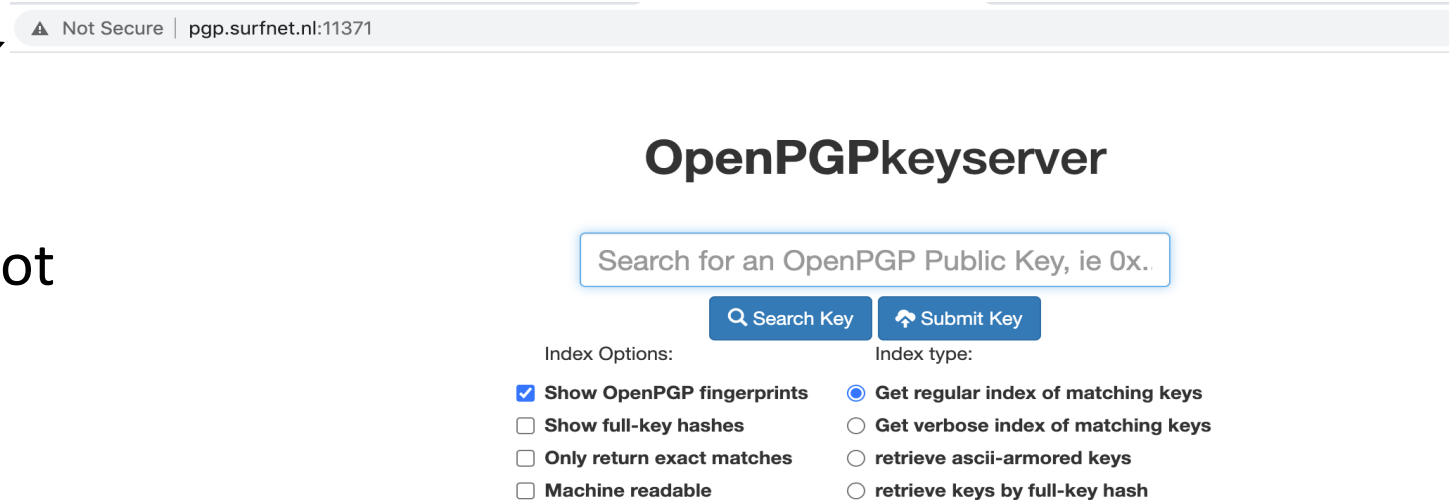
- Regularly
  - Verifying signed information (you must have the signer's **public** key)
    - Plain signature
      - **gpg –verify signed-file**
    - Clear text signature
      - **gpg –verify filename.asc**
    - Detached signature
      - **gpg –verify filename.sig**

PGP Issues/Concerns

- **How to get the public key of someone you don't know?**
- Authenticity issues:
  - PKI is susceptible to MiM unless properly used.
  - Web of Trust is considered to be failure (?) because it is very difficult to implement not only technically but also socially
    - There are additions to OpenPGP to patch it, e.g. https://inversegravity.net/2019/web-of-trust-dead/
  - Public keys (of malicious parties) may be retrieved from key servers and used
  - See for more details: https://gist.github.com/rjhansen/67ab921ffb4084c865b3618d6955275f

PGP – Some Notes

- Key server: e.g., http://pgp.surfnet.nl:11371



❗ How come HTTP and not HTTPS?

- Https is *not* required (though good to have), as the authentication verification depends on:
  - the public key's cryptographic hash
  - the signatures attached to the public key.

PGP - Some Notes

- The public key's cryptographic hash

gpg --fingerprint …

- Hash size: 40 hex chars, 160 bits.

```
pub   rsa3072/0x4977F4633C849F70 2020-10-21 [SC] [expires: 2022-10-21]
      Key fingerprint = D034 D44D 8A5F 3E51 21B2  63ED 4977 F463 3C84 9F70
uid                 [ultimate] InfoSec2020 <infosec2021@rug.nl>
sub   rsa3072/0x6572C88AF697F8A2 2020-10-21 [E] [expires: 2022-10-21]
```

Course Evaluation: 5 – 10 minutes

That's All