Digital Signatures

Florin Iucha

Digital Signature

- What is it?
- How does it work?
- How do I get one?
- How do I use it?
- Why do you sign your e-mails?
- Where do I get more information?

What is it?

- A method to confirm the origin of a particular digital artifact (file, e-mail message, etc)
- Similar to the real life signatures
- Of course, both can be forged, obtained under duress...
- “Security is a process, not a product.”

How does it work?

- Digital signatures are intimately connected to the public key cryptography
  - A public algorithm
  - A key pair: a public key and a private key
  - The distinction is not intrinsic, but in the distribution: one distributes the public key as much as possible and protects the private key as much as possible

How does it work? (cont.)

- Dual behavior:
  - What is encrypted with the public key can only be decrypted with the private key
  - Likewise, what is encrypted with the private key can only be decrypted with the public key

How does it work? (cont.)

- If I encrypt a file (*) with my private key and I publish the encrypted file along with the original file:
  - Anybody who has my public key can verify that the file was indeed encrypted with my private key

(*) Usually only a file digest (hash) is encrypted, not the whole file.

Assuming...

- ... that I protected my private key to the best of my knowledge,
- ... that nobody stole my key,
- ... that nobody broke the key generation algorithm,
- ... that nobody installed a trojan program on my computer that signs on my behalf,

How does it work? (cont.)

- Somebody can infer with some degree of probability that I signed the file.

How do I get one?

- PKI: Public Key Infrastructure
- Generate a key pair and
  - get the public key signed by a trusted CA and publish it,
  - or get a group of friends and sign each other key (key singing party) and publish them on a key server.

GPG Tutorial

- Generate a key pair
- Publish the public key
- Sign a file
- Import somebody else's public key
- Verify a signature
- Encrypt a file
- Decrypt a file

Generate a Key Pair

- gpg --gen-key then
- ... then follow the wizard to enter
  - the key type (default is ok)
  - the key size (default is ok)
  - your user name, mailing address and comment

Generate a Key Pair (cont.)

- ... then move your mouse a lot, browse the internet
  - these activities will generate random bits for the key

Publish the Public Key

- gpg --keyserver search.keyserver.net --send-keys “Your Name”

Sign a File

- gpg --armor --sign --detach $filename
  - Armor – generates 7-bit clean file

Import Somebody's Public Key

- gpg –search-keys “Somebody Else”
- Something like this will be displayed

```
florin@bee:~$ gpg --search-keys "Florin Iucha"
gpg: searching for "Florin Iucha" from HKP server subkeys.pgp.net
Keys 1-2 of 2 for "Florin Iucha"
(1)     Florin Iucha <florin@iucha.net>
          1024 bit key 3B90DFE4, created 2000-10-25
(2)     Florin Iucha <fiucha@neta.com>
          1024 bit key 954D59B0, created 2000-02-11
Enter number(s), N)ext, or Q)uit > 1
gpg: key 3B90DFE4: public key "Florin Iucha <florin@iucha.net>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

Import Kernel.org Public Key

- Fast way:
  - gpg --keyserver wwwkeys.pgp.net --recv-keys 0x517D0F0E
- Slow way:
  - Copy the key from http://www.kernel.org/signature.html into a file (kernel.org.key)
  - Import the key:
    - gpg --import kernel.org.key

Verify a Signature

- gpg --verify $sigfile $filename
- For instance, verify a linux kernel patch signature.

```
florin@bee:/alt/downloads/kernel$ gpg --verify patch-2.6.0-test6.bz2.sign
patch-2.6.0-test6.bz2
gpg: Signature made Sat Sep 27 20:53:42 2003 CDT using DSA key ID 517D0F0E
gpg: Good signature from "Linux Kernel Archives Verification Key
<ftpadmin@kernel.org>"
```

Encrypt a File

- gpg -r $recipient -e $filename
  - Assumes that you imported $recipient's public key
  - Will prompt if the key is not trusted.
  - If you want to avoid the prompting, you need to estabilish a trust relationship with the key.
    - You verify the key with the owner.
    - You get somebody you trust to vouch for the key authenticity.

Signing a Key

- You can sign a key with your own private key when you have some degree of certainty that the key belongs to the person
  - You have received the key in person.
  - You know his voice and he read the key over the phone (or the key fingerprint).

Signing a Key (cont.)

- How to sign a key
  - gpg --edit-key $username or
  - gpg --edit-key $keyid
- You will enter the command mode
  - Enter “sign”
  - Enter the degree of trust
  - Enter you key passphrase
  - Enter “save”

Decrypt a File

- gpg --decrypt $file
- You will be prompted for the key passphrase.

Using GPG with E-Mail Clients

- Mutt
  - Copy /usr/share/doc/mutt/examples/gpg.rc to ~/.muttrc.gpg
  - Edit it to specify the fullpath to pgpewrap as most likely /usr/lib/mutt is not in your path

Using GPG with Mutt (cont.)

- Add the following lines to your .muttrc
  - set pgp_sign_as=0x3B90DFE4
  - set pgp_autosign=yes
  - set pgp_replyencrypt=yes
  - set pgp_timeout=1800
- This will automatically sign outgoing messages (both new messages and replyes) and the passphrase will be cached for 30 minutes.

Using GPG with Mozilla

- Available as plugin at http://enigmail.mozdev.org/.
- Couldn't get to work 8^(
- Install Mutt, go two slides back...
- ... or use the x.509 certificates.

Using GPG with Evolution

- Very easy
- Configure in “Tools>>Settings>>Mail Accounts>>Security”
- Just enter your key id and check the desired options (sign, encrypt, trust)

Why do you sign your e-mail?

- Spread awareness of the availability of reasonably good e-mail privacy solutions.
  - People and corporations still use sealed envelopes to exchange messages.
- Spread my key fingerprint on various mailing-lists and get it in many mailboxes.

Why do you ... (cont)

- “Real Men don't make backups. They upload it via ftp and let the world mirror it.”

Web of Trust

- One of the hardest problems in cryptography: key distribution
  - Certificate Authority
    - Hierarchical Model
    - Can be always trusted?
  - Webs of trust
    - Small and disconnected
    - Trust cliques

Upcoming Key Singing Event

- Where?
  - ACM - U of MN Chapter [map] [map]
  - 2-204 EE/CS
  - 200 Union St
  - Minneapolis, MN 55455
- When?
  - Thursday, October 23, 2003
  - 7:00pm to 7:30pm
- http://ry4an.org/keysigning/

More information (books)

- First and foremost Bruce Schneier's (http://www.counterpane.com/) books:
  - Applied Cryptography
  - Secrets and Lies
  - Practical Cryptography

More information (web sites)

- http://www.gnupg.org/
- http://www.pgpi.org/
- http://linux.oreillynet.com/pub/a/linux/2003/09/04/email_pki.html

Questions?