Sicurezza e Privatezza

LAB 2 – GnuPG and John the ripper

The learning objective of this lab is for students to get the basics of symmetric and asymmetric encryption using GPG tool. After finishing the lab, students should be able to manage keyring, exchange keys and handle encrypted files. Another focus of this lab is password cracking with John the ripper.

Students are invited to solve the following exercises, compress all solutions in a zip file and submit it to upload.di.unimi.it. The name of the zip file should be **`Lab2NameSurnameUniversityid`** (eg. Lab2MatteoZoia856378.zip). Files not conforming to these rules will be automatically rejected.

The first ten students who will respond correctly to the entire lab will receive a bonus of 0.5pt on the final grade.

Exercise #1: SW Integrity

Verifying the integrity of software you download is important to ensure that your software hasn't been tampered with. Your task is to download the GPG stable source code and verify its signature, a screenshot has to be included in the solution file containing the line stating with "`Good signature from <person> [unknown]`".

Exercise #2: Key generation

For using GPG you first need to create a public and private key pair. The public key is so-called because you will share it with others so they can use it to send you secret information, on the other side the private key is used to decode any information encoded with your public key. **Following the steps in the video to generate a key** pair (public and private)**. Export and include your public key in the solutions file.**

Exercise #3: Send a secret message

More deeply, you must have anyone else's public key if you want to send them encrypted data. Any people that uses GPG has two keys, public and private one. Your public key is used by other people to encrypt information they want to send you, and when you receive an encrypted message from someone, you use your private key to decrypt it. You're the only person who can decrypt the secret message because you're the only one who has the private key, with the passphrase that unlocks it. **The student has to encrypt the given secret_file.pdf with the following public key and include the encrypted file in the solution.**

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQENBF6LQrIBCACu4GXs3d9sqWJCP7Bp0hVqNYhTrTOu131GSFKlCKhMLgZoVR1c
8FjRceER9B6UOtY3VN2OVLR9c+XtzJiqkPJpkmDqanzazfn66OQ0jc97olD0C6NO
CYxfP5PRIjd+ZR7j/QQaatWDE72wCUyoDqYRPZa/gdIXwApZD83i5PdDBeiz2rLU
sJIWRjcmGw4qJnU4+V8gtUB1KavMqBQED8IfhJxs/BpU3UblAuTZg5i/GkyrFyYU
4XeXG1eZ36OZOMQthtSwpzOvOqWhTJnYPqg5SVMnZMrHGKuzt3/MhfPtdB1NTXen
LvZfRcb6K6MeJGIzrHX+/u7lE/jrCeFqQe2vABEBAAG0OFNQTGFiIHVuaW1pIChT
aWN1cmV6emEgZSBwcml2YXRlenphKSA8U1BsYWJAZGkudW5pbWkuaXQ+iQFOBBMB
CAA4FiEEZzLDlWbYqno9Cq36isrcEJSO8I0FAl6LQrICGwMFCwkIBwIGFQoJCAsC
BBYCAwECHgECF4AACgkQisrcEJSO8I3Z4gf9Gqw7DHDfVQqOWfdBrfl2uF7Pkv/l
DmlLjZPgTEU2MWnIDfFMFZ6/n+XbmaTrRU5meHLshYn20QOf6VTiKuIptQWU6FJP
```

```
jY0Ooicwvf+ZD41ixn8/UnceAJW0Is1iWBJq42+mgOzI+H+lTfiW0wGs6nnckHU9
85Omj5qlLO95ulD0ndBym17p3ciE9BYy2s3A02TStpzi24fr6EybTOv7N5aDofik
u0u+ZHxBwTWytIJyvRj71WHBT28Bjs9W8sU0RF676DMK/NRePW3p1ktSVGOK09cu
qxVl7m1hZ1tOTPScmB5yIy+qTuflShg1c8ws2RAt0XfXiHB6WYghP4rDNrkBDQRe
i0KyAQgA103gfnnY6HCUM+RWhHcYbSUTcv5ILXO1KJ7fujmheP5xWrgIqipk0mlz
qmAQ7LXv4aTxe54xKj6WBGTq1pxRR72cGIPFVU7Ub64I86xmuwXFKloRyA+CkWqt
ZOV3o/CNUbwOstt3OgDyofB3rLL8J9vPbRqqz8TvbSpbQobwsWLra8KEIoRPeBwJ
H7S0jk9WougA1AuPDwYuIv89vMYd5Mzst9f7ZpdBh9C8Ke0rT850VxWMRf8ekuZB
GWB2EvAwrpSEwlfyGhsuAU6LdVNpOfDRETFA9Q0n+4A18dwVWs4YmVHFY3bY0Es4
AK3OGTvJ7h2xr9lK2x6p0+eXqgFEbQARAQABiQE2BBgBCAAgFiEEZzLDlWbYqno9
Cq36isrcEJSO8I0FAl6LQrICGwwACgkQisrcEJSO8I1EOAf8DmRR68EYFM4wbhB2
inhjOpu01ssXQG1ILRU9NVwkzcDRRTD3eD8t65QLUGyTS+xVQv/tKeVYZNu9nWmj
hb3TQuyV40b5ehdFL3HfU8d3NMxyDZ/zmTbs9zV+t/J8H5J/1lzKCLI8mL7hJvDV
Pf8xnpMQQKd7hU+BAa7EDZivxzzDywTte1OkG6kjy1WNuPk2kq8IOqPhqund/aQ+
TeDjs2bEFslqOpP4xK0CT3PJInYmtKNL2hoTSBbT/idhQDWqnr3R5fhPCybKbvtw
jfXZkX0hDDyuDVFXiphT5AuuKbbiRNGXHv8fk+AtkIhjy1jdw5i8MGPlKhPsUEfz
PqAqDg==
=lAU7
-----END PGP PUBLIC KEY BLOCK-----
```

Exercise #4: Digital signature

A digital signature is a cryptographic technique used to validate the authenticity and integrity of a file or a message. For signing a document, the first step is generating the hash and then encrypt it with the private key. The reason for encrypting the hash instead of the entire message is that a hash function can convert an arbitrary input into a fixed length value, which is usually much shorter. Any change in the data (also a single data character) will result in a different value. Anyone can validate the integrity of the data by using the signer's public key to decrypt the hash and check if it matches. **Your job is to find the signed PDF inside the lab02_digitalsign directory. Append your digital sign to it and send the file back**.
Also, you need to sign the file encrypted in the exercise 3 with a detached sign and attach the signature file.

Exercise #5: Key signing

The level of trust of your private key is related to the level of trust of persons who signed it. Invite some of your colleagues, those you best trust, to sign your public key. Describe also the protocol you set up for verifying their identity and be sure that they will not sign with a faked key.

Password cracking

Exercise #6: Single cracking

John the Ripper is a password cracking software used for password strength testing. It combines a number of cracking mode into one tool, autodetects password hash types, and includes a customizable cracker.
The simplest way you can use for cracking password through John is “Single cracking” mode, it will use the **login names**, **"GECOS" fileds**, **"Full Name" fields**, and **users' home directory names as candidate for passwords**, also with a large set of permutations rules applied. “Single craking” is the mode you should always start cracking with.
Inside lab02_simplecracking directory there is a unix password file, you need to **crack all the password in it**.

Exercise #7: Wordlist

As you know, not all users use passwords crackable via “single mode”, many of them use passwords that are not connected with user's personal information fields. In this case the “single crack” mode is not usefull. Trought “wordlist” mode, John uses a dictionary of words and it **compares the hashes of the words present in the dictionary with the password hash**. We can use any desired wordlist, John also comes with a password.lst which contains most of the common passwords.
The students has to **crack all the passwords** inside the lab02_wordlistcracking directory **with wordlist mode** (a simple dictionary is already included).

Exercise #8: Bruteforce

When a password is not common there are less chance to find it into a dictionary, in this case John’s “incremental” mode implements a brute force attack that try all password hashes in a given wordspace (eg. lowercase). Recall that this attack can only be successfully completed if the password length is small; we can assume that cracking with this mode will never terminate because of the number of combinations being too large (actually, it will terminate if you set a low password length limit), and you'll have to interrupt it earlier. **The student has to crack all password in the file lab02_bruteforce directory using john incremental mode.**

Note: The password maybe lowercase a-z or 0-9 digits, not both in the same time. All passwords have a maximum length of 6 characters.

Exercise #9: Going on the net

Find and download a password file by googling as well as some related dictionaries, use John and report the results.