LAB 1 - Fondamenti di crittografia e GPG

Sicurezza e privatezza

The learning objective of this lab is for students to get familiar with elementary concepts in the common cipher, encryption/decryption, keys and OpenGPG. After finishing the lab, students should be able to gain a first-hand experience on basic encryption algorithms as well as public/private keys and trusting mechanism.

Students are invited to solve the following exercises, write the solutions in a .txt file which will contain their name, surname and University ID, and submit it to SPlab@di.unimi.it, with subject: Lab1NameSurname2020. Messages not conforming to these rules will be automatically rejected.

The first five students who will respond correctly to the entire lab will receive a bonus of 0.5pt on the final grade, 0.25pt on the crypto part and 0.25 on the GPG part.

Crytpo

Exercise #1: Warmup with base64

Base64 encoding schemes are commonly used when there is a need to encode binary data that needs be stored and transferred over media that are designed to deal with textual data. This is to ensure that the data remains intact without modification during transport. Base64 is used commonly in a number of applications including email via MIME, and storing complex data in XML or JSON. **Decode the following.**

```
VGhpcyBpcyBiYXNlNjQgZW5jb2RpbmcsIGlzIG5vdCBhIGNoaXBlciBidXQgb25seSBhIHd
heSB0byByZXByZXNlbnQgYmluYXJ5IGRhdGEgaW4gYW4gQVNDSUkgc3RyaW5nLg==
```

Exercise #2: Affine cipher

It is well-known that monoalphabetic substitution cipher (also known as monoalphabetic cipher) is not secure, because it can be inverted quite easy. In this lab, you are given a cipher-text that is encrypted using an affine cipher. **Your job is to find out the original text by implement a bruteforce attack that try all the possible combination.** It is known that the original text is a famous Italian poem. We do keep the spaces between words, so you can still see the boundaries of the words in the ciphertext.

Hint: A good starting point is to check the domain of the encrypted data.

Exercise #3: Stream cipher

A **stream cipher** is an encryption algorithm that encrypts one or more byte of plaintext at time. It uses an infinite **stream** of pseudorandom bits as the key. For a **stream cipher** implementation to remain secure, its pseudorandom generator should be unpredictable and the key should never be reused. The following file secret.txt (inside stream_chiper folder) has been encrypted using the most famous stream cypher algorithm with a key of length 3byte. **Recover the original plaintext.**

Guidelines: as you can see the given text is quite long. So you can use the frequency analysis to find out the plaintext. You know that the key is 3byte long, so every 3byte the encryption of a given letter remain the same.

The original text is a famous English novel.

GPG

Exercise #1: 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 lesson to generate a key** pair (public and private)**. Export and include your public key in the solutions file.**

Exercise #2: SW Integrity

Verifying the integrity of a 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>`".

Exercise #3: 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 with a few words the protocol you set up for verifying their identity and be sure that they will not sign with a faked key.

You can start to sign the attached keys.

Exercise #4: Send a secret message

With GPG you can also share secret messages, if you have anyone else's public key then you can 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 PDF file (paper.pdf) with the SPLab public key and include the encrypted file in the solution.**

Exercise #5: 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 digital_sign directory. Append your digital sign to it and send the file back**. For more help you can visit https://www.gnupg.org/gph/en/manual/x135.html the official GPG manual.