DPROT LAB WORK 3: Hashed ElGamal, Hybrid Encryption & HMAC

Ismael Douha Prieto

MCYBERS - Q1 2020/21

1 Introduction

In this laboratory work, I developed three scripts which allows to generate a Diffie-Hellman long-term key pair and to encrypt and decrypt using hybrid encryption; Hashed ElGamal as public key encryption scheme and AES-128-CBC as private key encryption key. In addition, HMAC is used as message authentication method.

2 Key generation

The long-term key generation is done by the script ***gen_key.py***. To use it, is enough typing the following command on your terminal:

```
1 $ python3 gen_key.py alice
```

This, what it does is generate two files, *alice_pkey.pem* and *alice_pubkey.pem* which contains the public and private long-term key for alice respectively. This key pair is generated using Diffie-Hellman algorithm with the third cyclic group defined by RFC5114 and stored in PEM format. In case that this cyclic group is not in the directory where the script is executed stored in a file called *param.pem*, it generates and it stores in the directory.

3 Encryption

The encryption is performed by the script ***encrypt.py***. It should be used in the following way:

```
1 $ python3 encrypt.py alice 'Message to encrypt'
```

With this, what we achieve is to encrypt using alice's public key (alice_pubkey.pem), which should be in the same directory as the script is, the message passed between quotes as second parameter.

The result is the generation of a file called *ciphertext.pem*, which contains the ephemeral public key in PEM format, necessary to decrypt the message, because the common secret key used to encrypt was derived from the public key of alice and the private part of the ephemeral key (generated in this script), the Initialization Vector (16 bytes generated randomly), the encryption of the message using AES-128-CBC and the HMAC code. The last three elements where stored in BASE64, between delimiters of the begin and the end of each one using a similar system to what PEM uses. As what it is used as public key scheme is Hashed ElGamal, to encrypt the message and for generate the HMAC, the common secret key is hashed using SHA-256, using the first 16 bytes to encrypt the message and the last 16 for the HMAC code generation.

In case that someone tries to encrypt a message for a user that his or her public key is not in the same directory or not follows the naming format that *gen_key.py* uses, it raises the following error:

```
*** ERROR: missing file "charles_pubkey.pem" ***
```

For this case, this would be the output of the script if someone tries to encrypt a message for charles and his public key is not in the same directory than the script or it has a different name than *charles_pubkey.pem*.

4 Decryption

For the decryption, I developed the script ***decrypt.py***. With the next command, the decryption is performed:

```
$ python3 decrypt.py ciphertext.pem alice
```

After the execution, a file called *deciphered.txt* is generated and saved in the same directory that the script is, containing the message in plain text. First of all, what this script does is regenerate the common secret key using the ephemeral public key, contained in the *ciphertext.pem* and the private key of alice, which should be in the same directory as the script. Once the common secret key is regenerated, it is hashed to obtain the two 16-bytes sub-keys, necessaries to decrypt the message and to generate the HMAC code and verify if alice is the correct receiver.

In case that the receiver specified as second parameter in the script execution is not the one that the cipher message was for, it raises the following error:

```
*** ERROR: Wrong TAG. Please specify the correct receiver. ***
```

Another error that can appear if the private key of the receiver, specified as second parameter, is not in the same directory or not follows the naming format that *gen_key.py* uses.

```
*** ERROR: missing file "charles_pkey.pem" ***
```

For this case, this would be the output of the script if charles tries to decrypt the message with his private key and it is not in the same directory as the script or it has a different name than *charles_pkey.pem*.