CS4614 – Lab assessment

In your job as a cryptanalyst, you have been provided with a number of encrypted files retrieved during a Garda investigation. Multiple layers of encryption have probably been used, but the key is hidden somewhere in the files. However, only some of the files seem to be relevant: some other have been fabricated with the purpose of misleading the investigation.

Instructions

1. The directory `aes/` contains some files (`[0-9].aes`) encrypted using the AES cryptosystem, a 128-bit key and the Electronic CodeBook mode of operation. Each file has also been signed using a private key, and the signature files are also present (`[0-9].aes.gpg`). As a first step, import the public key used to sign the AES files (`Key ID: FC2593EA`), as seen in Lab 5. The key is included in the assessment Zip file. (`pubkey.asc`).
2. After having imported the public key, verify each of the signatures (as seen in Lab 5). It appears that only one file has been correctly signed. Which one?
3. Once you have found the correct file, you will need to decrypt it. But what is the key? During the investigation some older files were found. In particular, one that was encrypted using an insecure cryptosystem, the Vigenère cipher. Find the key to decrypt the Vigenère-encrypted file (`johnny-v.txt`) as seen in Lab 1, using the following web tool:
   https://www.cryptool.org/en/cto/vigenerebreak
4. Once you have retrieved the key, try to use it to decrypt a second file also encrypted using the Vigenère cipher: `key-v.txt`.
5. The contents of this file are actually the AES key (in hexadecimal format) for the AES-128 file you have identified before! However, some bits are missing: you can recognize them from the character '`X`'. Using your preferred programming language, write a script to brute force the missing bits and recover the plaintext.
   Hint: *Several libraries are available that implement AES. For python, for example pycrypto is available. However, you can also build a script around OpenSSL (as seen in Lab 3). For help with the OpnSSL command, you can refer to the provided OpenSSL cheatsheet.*
   Note: *A recent version of OpenSSL such as 1.1.X, is recommended.*
6. Finally, open the decrypted file and check its contents. Then, use the SHA-256 hash function to obtain the file hash digest (as seen in Lab 3).

Submit your results here.