INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Cryptography

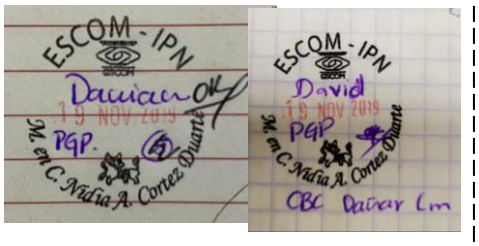
“PGP”

By:

Meza Madrid Raúl Damián & Portilla Martinez Jose David

Professor:
MSc. NIDIA ASUNCIÓN CORTEZ DUARTE

Dec 2019



Introduction

This practice is based on the Pretty Good Privacy program, which allows us to maintain a certain level of confidentiality and / or authentication when sending text messages or even files through online mediums.

We're only focusing in text files, but we're still applying the core concept of the program. Although similar processes, the encryption and decryption process have key differences, first, for the encryption process but only the confidentiality service:

The plain text will be encrypted using the AES Algorithm, this Block Cipher Algorithm uses a 8 o 16 bytes key and must be known by both parties and this is a risk if the key's privacy is not properly handled, so to assure its privacy, the key will only be use in the process and not disclosed to any party, it will be randomly generated but it'll be attached to the encrypted file.

So, the text file is encrypted using the AES Algorithm using a randomly generated key and the encrypted file will be the cipher text and the key… but what about the key? It could be easily read if the file is intercepted, so we need to assure only the person we're sending the message to will be able to read it, so the key must be encrypted. But now, as we only want the receiver to be able to read it, RSA must be used. This algorithm also uses a key, but it needs a pair of keys, each party must have a private key and a public key (further explanation for this in the decryption process). For each process (encryption/decryption) and the scenario (confidentiality/authentication) the key used varies, for this particular case (encryption/confidentiality) the key used must be the public key of the receiver. After doing this (encrypt the message with AES and encrypt the AES key with RSA using the receiver public key), we guarantee confidentiality.

To guarantee authentication is an easiest process, we apply a hash function to the plain text, this gives us a digest, then encrypt the digest with RSA and the encrypted digest is attached to the encrypted file using the sender private key.

The decryption process is very similar, what changes is how the authentication is guaranteed, because in the encryption process, we only made a digest of the plaintext and then encrypted that. So, for the receiver to authenticate the message first he/she must decrypt the message and then the digest, apply the same hash function to the decrypted message and then compare both digests. If both are the same, it means that the message was sent by the desired person, if not it might've been intercepted or fabricated.

Diagrams

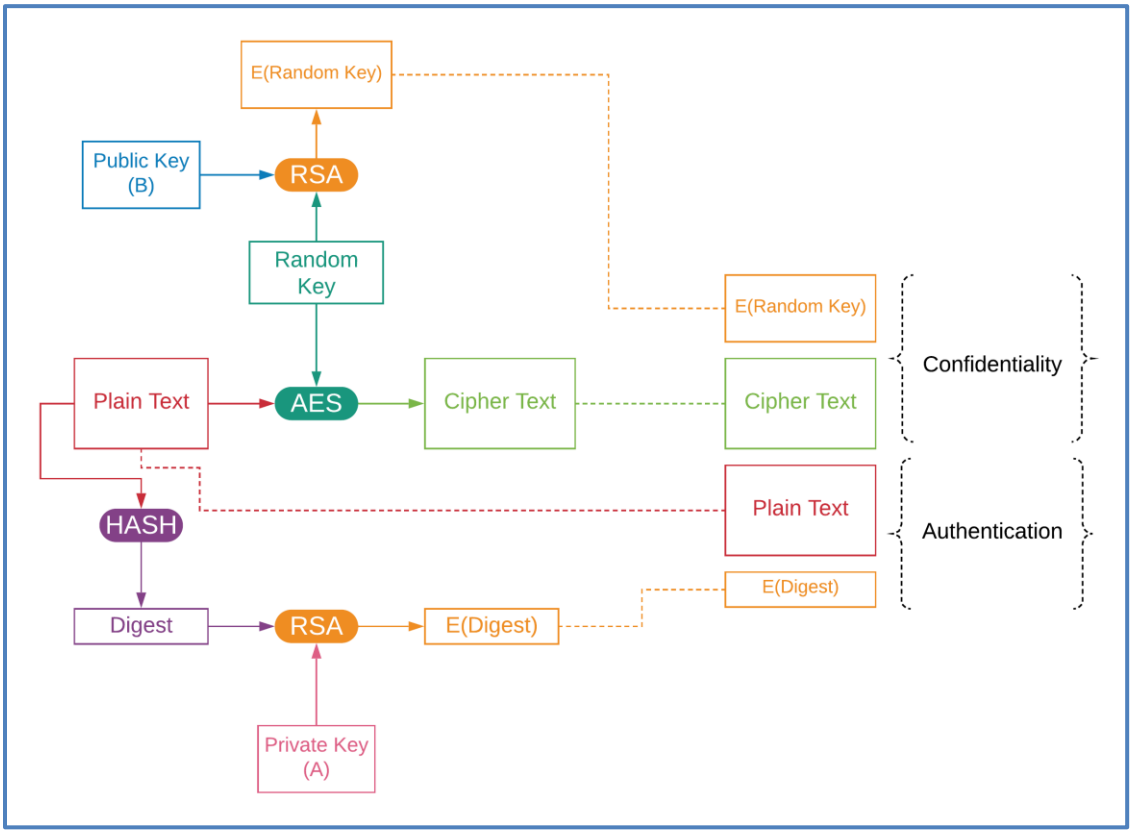


Figure 1: Encryption Process

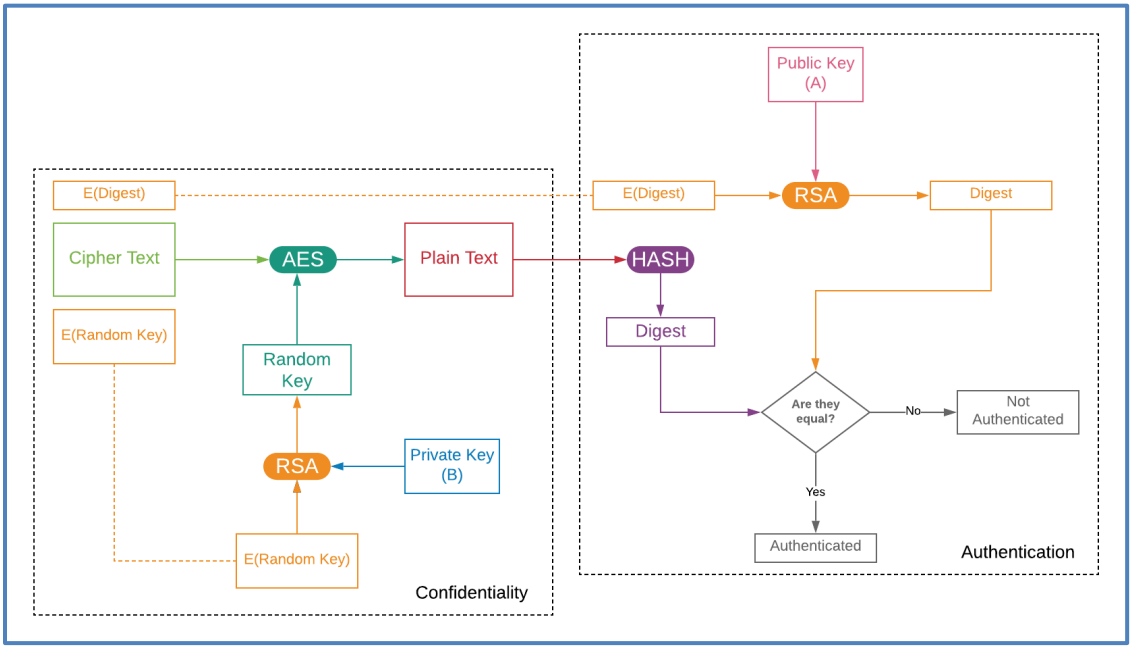


Figure 2: Decryption Process

Conclusions

We came across several problems, specially with how the data was encoded and padded. First, the encoding must be only done when necessary, since at the beginning we used the encoding function because it seemed suitable for what we were trying to do, but no, it caused a lot of troubles when trying to write the data into de the files and we had to debug this issue,

Second, the padding. When working with AES the data must be divided into blocks of 8 or 16 bytes, thus, the complete size of the information must be a multiple of one of those values, so when the information can be divided into block of the same size, that's when padding comes along. When the data has a size not multiple of 8 o 16 bytes (depending of what we choose), it must be padded with anything we want. The solution to this problem was rather simple, we used a library's function that did all the work for us.

The implementation of this program wasn't too difficult, as the professor let us use any library we wanted and although the library's functions weren't perfectly documented, after coding some programs to test what the functions can do and seeing our results, the implementation was easy in most cases.

Something important to mention is the operation mode of AES, as we implemented it with an EAX mode of operation but it failed in some cases (as expected), that was due to what the mode of operation does, so, as the professor requested, we changed to the CBC mode of operation. After that change, we could see how the after modifying a some of the encrypted text, only part of the decrypted text is corrupted.