INSTITUTO POLITÉCNICO NACIONAL

Escuela Superior de Cómputo
(ESCOM)

ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

Room-control

Profa: Reyna Elia Melara Abarca

Benítez Morales Manuel Emilio
Camacho Soto Kevyn
Hernández Pacheco César Iván
Lozano Rivera Oscar
Ramírez Gaytán Omar

2CM9

Required cryptographic services and cryptographic primitives

Confidentiality

Block Cipher: AES 192 bits, CBC operation mode and PKCS5Padding.

Necessary because it must guarantee only sender and receiver know the content, if an adversary tries to obtain the message, it mustn't be able to be processed or read.

Integrity

Hash function: SHA-256.

To assure all information comes complete to receiver, the right state of message is a way to confirm there were not a successful trying to get the content by an adversary.

Not repudiate

Public-key cipher: RSA with PKCS8 encode for private key and X509 encode to public key; pair of key are of 2048 bits.

It is necessary because the person who sends the message must confirm that he is sending. For example when we talk about a legal procedure we need to confirm the person who is giving the license.

Internal process description

A sender use the desktop application 1 to cipher a plaintext -any extension- this is done internally divided in:

1. ***Charge process:*** A file is charged to system, its name is saved and content is read by bytes.
2. ***Integrity process:*** With the SHA-256 is created hash function from plaintext file, this is part of digital signature, function is gotten from a unique digest.
3. ***Not-repudiate process:*** To ensure protection of generated hash, encrypting with private key RSA is created the digital signature.
4. ***Confidentiality process:*** Key and initialization vector of AES are ciphered with RSA algorithm, this action to protect them. On the other hand, plaintext is ciphered with AES algorithm, creating the ciphertext.
5. ***XML file writing process:*** AES key, AES initialization vector, digital signature, RSA public key and original file name are written at XML file according to wodgame tags (See Notes section).
6. ***Compression process:*** To make easy the message transfer, both of XML and cipher file are compressed into .zip file at an specific path (See user manual).

A receiver use the desktop application 2 to decipher a message -which is received as .zip file- this is done internally divided in:

1. ***Charge process:*** An specific .zip file (See user manual) is charged to system.
2. ***Decompression process:*** To make easy the message reading, the .zip file with XML and cipher file is decompressed (See user manual).
3. ***XML file reading process:*** The XML tags are read to variables to be able to get AES key, AES initialization vector, digital signature, RSA public key and original file name at XML file according to wodgame tags (See Notes section).
4. ***Validating process:*** The generated plain text file is read to make the hash function, if this hash is the same than hash deciphering digital signature, file is valid.
5. ***Writing process:*** From the previous step, if file is valid, then process will finished, else, generated file will be deleted.

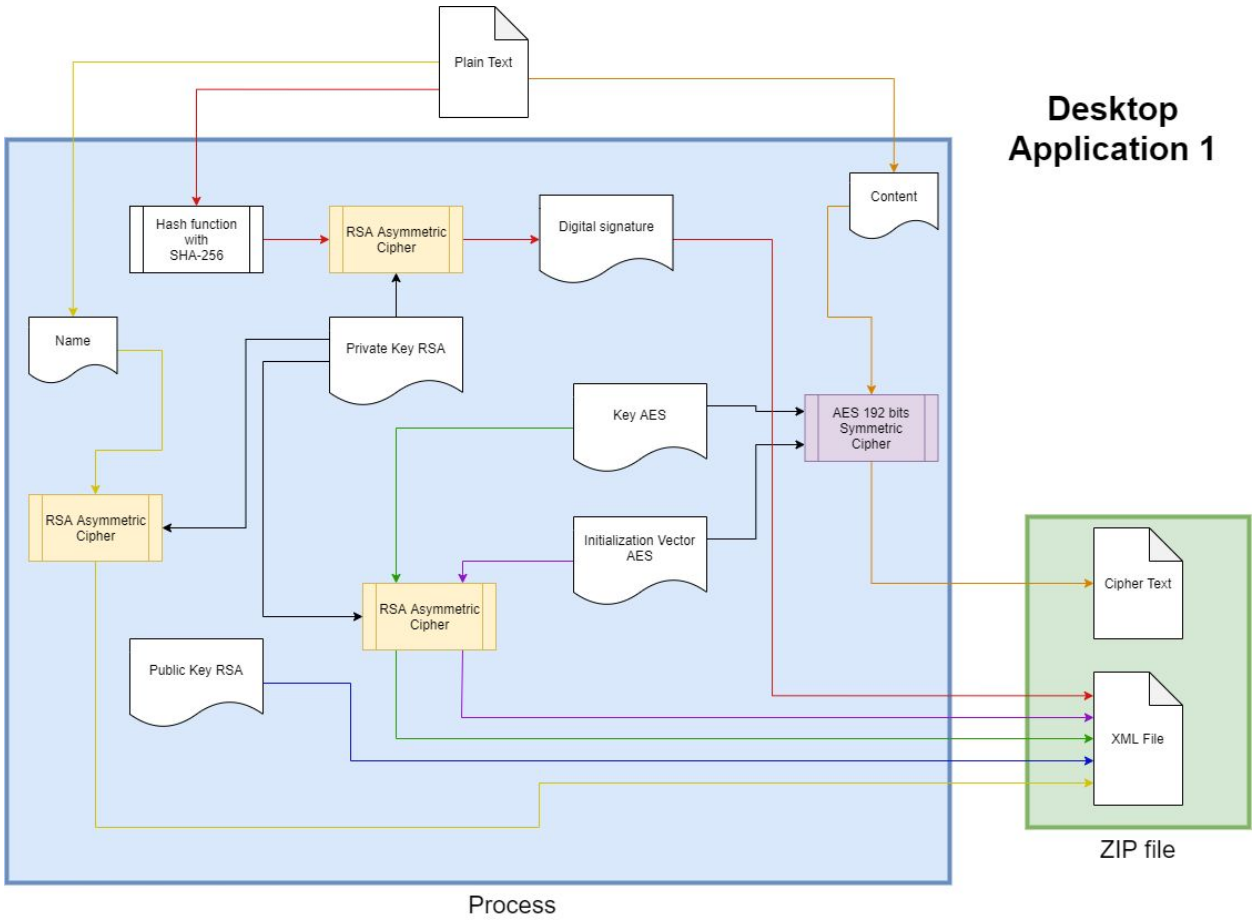


FIGURE 1. Cipher process structure.

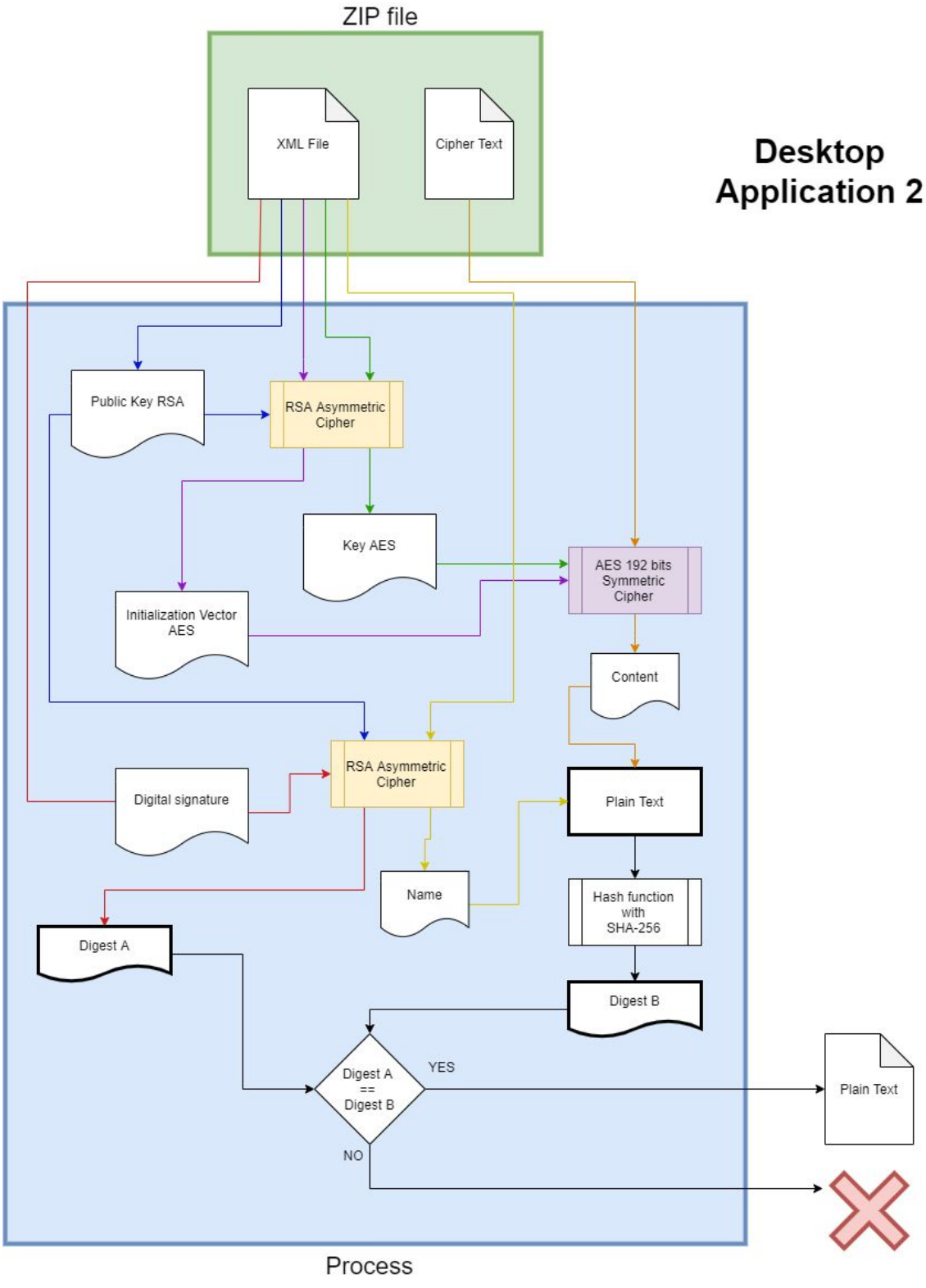


FIGURE 2. Decipher process structure.

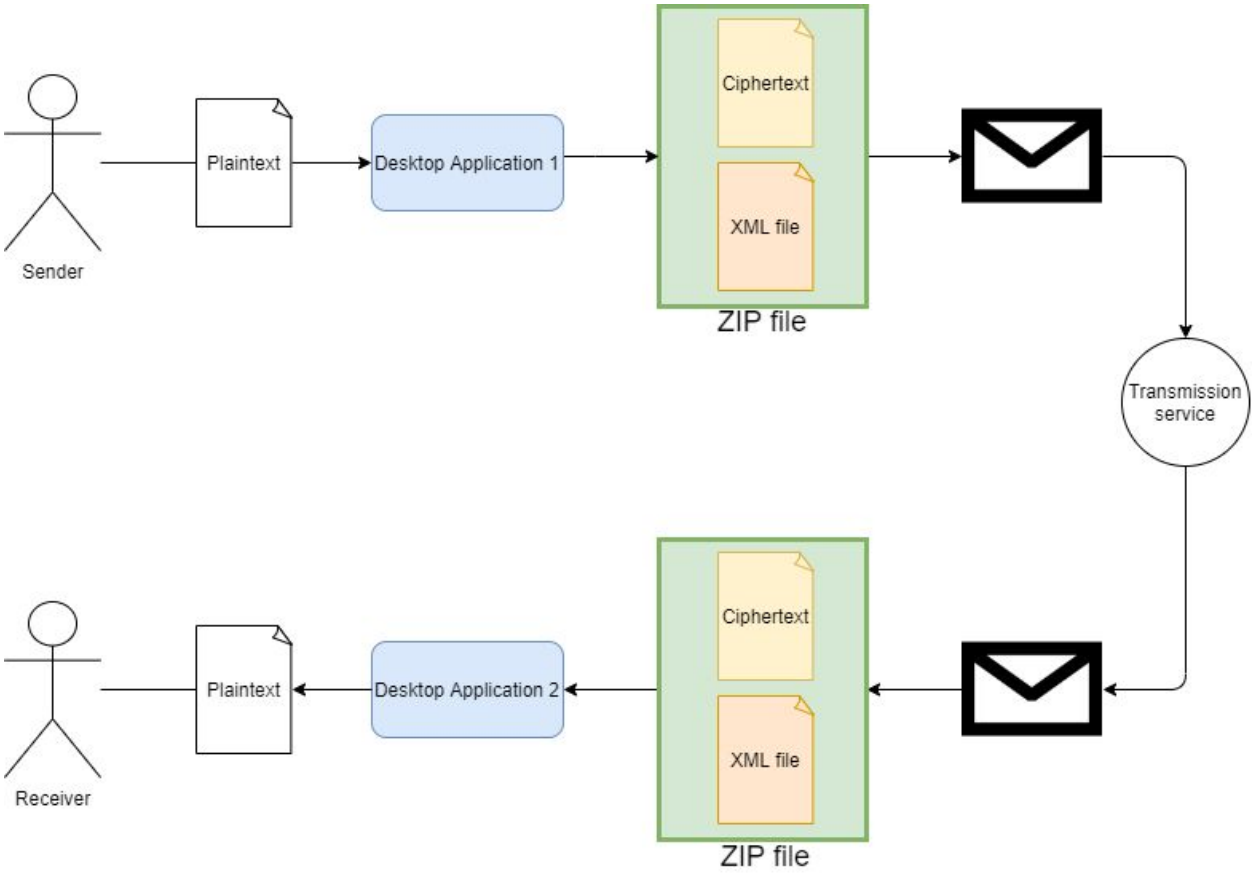


FIGURE 3. Working diagram.

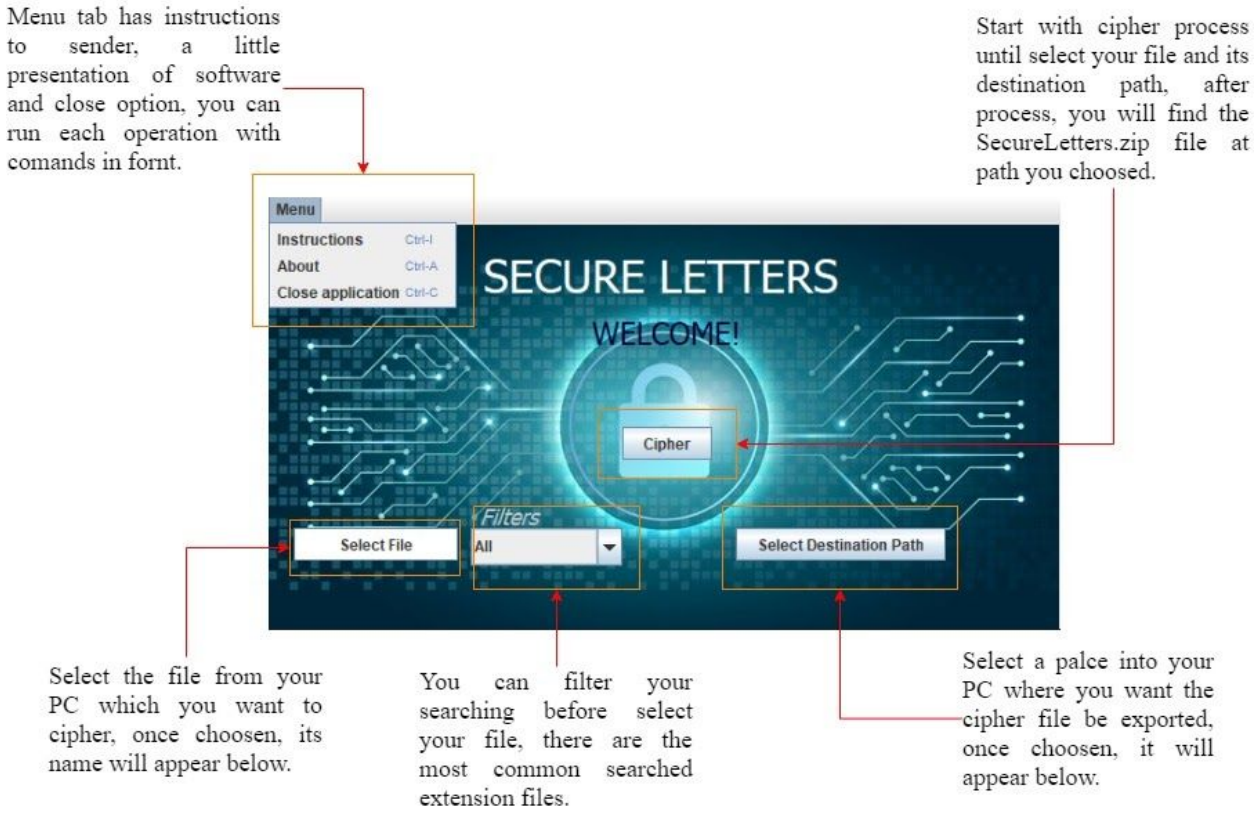
Notes

- All file names and tags into XML are word games, adversary could not to know or get confused about what is written but they have different meaning into program.
- Into XML file is located an extra tag called “information” which contents the name -something.format- of output file to write in decipher process.
- XML tags:
  - contramaestre: AES key.
  - hybrid: AES initialization vector.
  - unsigned: digital signature.
  - private: RSA public key.
  - information-value: cipher file original name.

User manual

Sender instructions

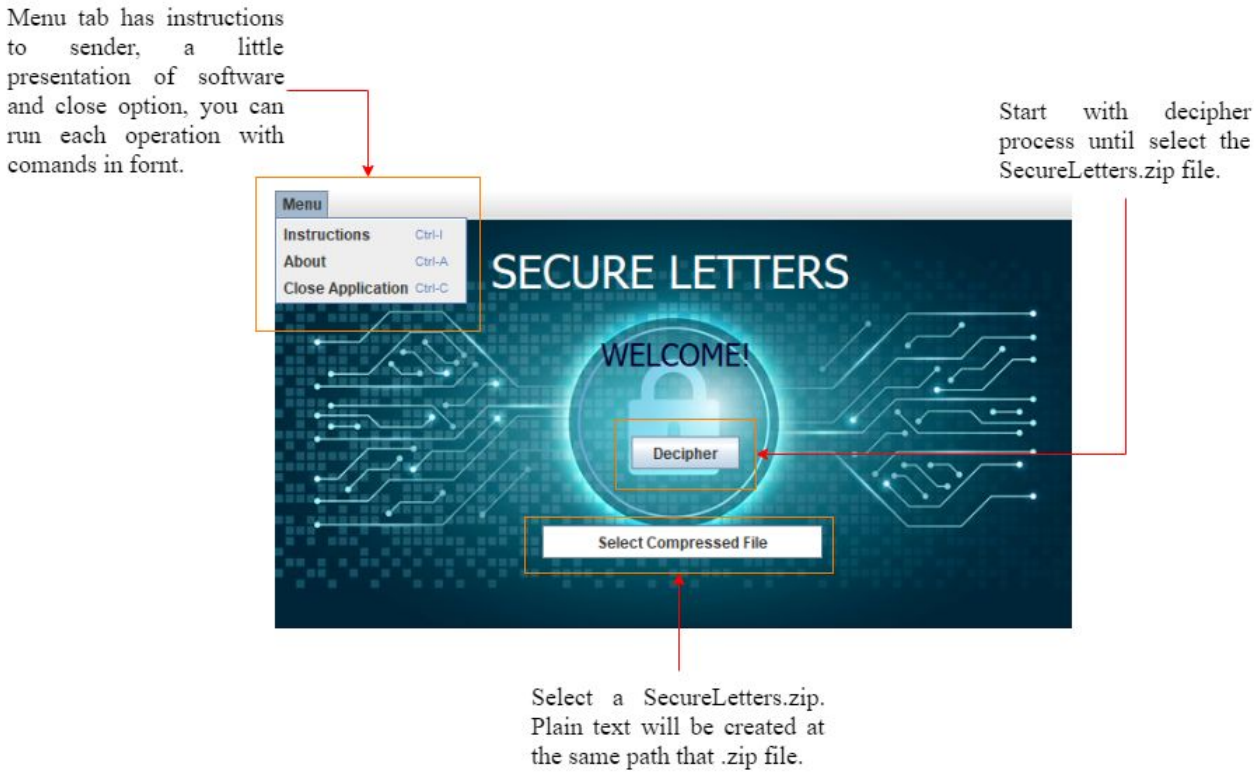
By running software to cipher, it will be showed the next pane:



Into destination path you will be able to see the *SecureLetters.zip* file, which contents the XML file and cipher text file, their names are *TheXML.xml* and *Exception.exec* respectively. Program will notify you if process was correct or there were something that not work .

Receiver instructions

By running software to decipher, it will be showed the next pane:



You charge a *SecureLetters.zip* file into system, it contains the two necessary files to make the original message, make sure it is a valid .zip file, it doesn't matter if it is a different name before extension, **it must be a compressed zip file with the specific .xml and .exec**, if it is not a file with this characteristics, program will not work or will show an error message.

.

References

[1] C. Paar and J. Pelzl, Understanding Cryptography, 2nd ed. 2010.
[2] L. R. Knudsen and M. B. Robshaw, The block cipher companion, 1st ed. 2011.
[3] “How to USe Java Cryptography API Securely”. Java. 2 de oct. 2017. [online]. Available https://www.youtube.com/watch?v=3HIdaSgxMlo&fbclid=IwAR39UjhcfVzYDC37chmVM-PG58b-5r2WcZlc-cUGKTu8ofcv3PqYy9_O5fc