## Cryptography application

An application to implement cryptography to encrypt and sign a file to be sent by email using RSA public-private key mechanism.

This is the readme file for fcrypt.py file

text.txt is the file which contains plaintext and is used for the encryption and decryption process.<br />

This name will remain the same since in the decryption part, while verifying the data, it reads input from the plaintext file.<br />

And this filename is not passed as an argument to the command line.<br />

<br />

import pickle- used to serialize /de-serialize a python object structure. It helps to convert the object into a byte stream <br />

during the process of "pickling".And during "unpickling",the inverse opearation takes place.<br />

In this program, pickle is used to dump and load the values of signature,cipher text,aes encrypted key and the IV.<br />

default\_backend-gives the defauly supporting algorithms and the supporting platform specific implementations.<br />

rsa-provides packages to generate the rsa public key<br />

serialization-provides modules to serialize the keys generated<br />

load\_pem\_private\_key, load\_pem\_public\_key- functions to load the private key and public key in the pem format

Cipher, algorithms, modes-provides the algorithms,codes and the backend for the encryption and the decryption process.<br />

padding and hashes- gives the modules needed for the hashing and the padding process during encryption/decryption<br />

<br />

USAGE:<br />

For encryption:<br />

fcrypt.py -e <destination\_public\_key\_file\_name> <sender\_private\_key\_file\_name> <input\_plaintext\_file> <ciphertextfile><br />

where :<br />

<destination\_public\_key\_file\_name>- The file name which contains the public key of the destination<br />

<sender\_private\_key\_file\_name> - The file containing the sender's private key for signing<br />

<input\_plaintext\_file>- The file containing the text which needs to be encrypted<br />

<ciphertextfile>-The file to which the encrypted data has to be written to <br />

<br />

For Decryption:<br />

fcrypt.py -d <destination\_private\_key\_file\_name> <sender\_public\_key\_file\_name> <ciphertextfile> <output\_plaintext\_file<br />

<br />

where :<br />

<destination\_private\_key\_file\_name>- The file name which contains the public key of the destination<br />

<sender\_public\_key\_file\_name> - The file containing the sender's public key used for verification<br />

<ciphertextfile>-The file from which the enrypted data is obtained for decryption<br />

<output\_plaintext\_file>-The file to which the final decrypted data will be written to <br />

<br />

The public-private key pairs for both the sender and destination are store in separate files.<br />

destprivkey.pem- The destination public key<br />

destpubkey.pem- the destination private key in pem format<br />

sendprivkey.pem-the sender’s private key<br />

sendpubkey.pem-the sender’s public key<br />

<br />

To generate the keys:<br />

A file key\_creation.py is used.<br />

pwd:<br />

C:\Users\BHPriyanka\Documents\First Sem\NetworkSecurity\Class\PS2><br />

<br />

py key\_creation.py<br />

<br />

Once the keys are generated, executed for encryption/decryption.<br />

No need to run the above command every time.<br />

Example executin of the file fcrypt.py:<br />

<br />

fcrypt.py -e destpubkey.pem sendprivkey.pem text.txt cipher.txt<br />

<br />

fcrypt.py -d destprivkey.pem sendpubkey.pem cipher.txt final.txt<br />