**Objectives**: By the end of this practical exercise, the students should be able to:

* Apply the classes and methods of the relevant pycryptodome packages to generate and verify digital signatures
* Export and Import relevant digital keys from file stores for digital signing or verification process.

Requirements:

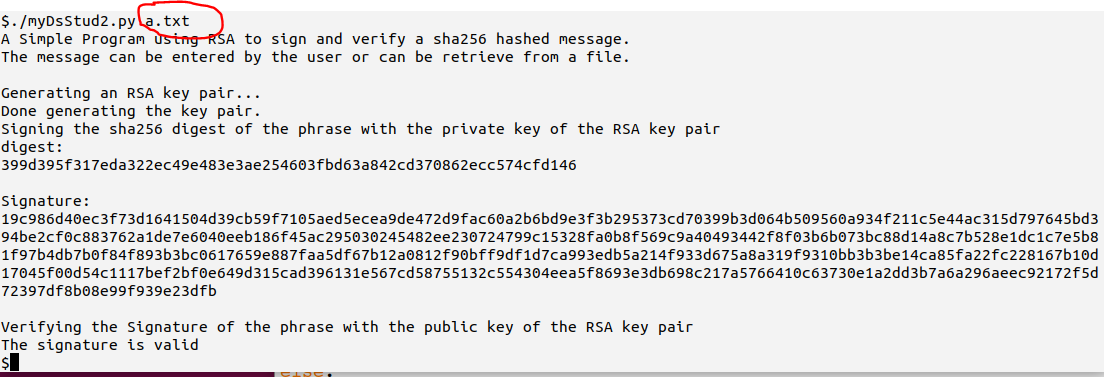
1. Extend from the sample program given (myDsStud.py) to generate a Signature and Verify it from a string entered from the command line or with a text file. Ensure that: -

* Sender can sign a message or a file and the recipient can verify a message or a file.

Sample run session 1, using the myDsStud2.py to sign on short message:



Modify myDsStud.py 🡪 myDsStud2.py (which will support file input)



*Note: Since your program will generate a new private / public key pair on each run, the program will perform the signature generation and verification together.*

1. Based on the given sample Python source, **myDskeysStud.py**. Explore the key pair export features to **export a passphrase protected key pair** to a **binary file containing the private key**, and its **public key to a text file** (base64 encoded).
2. Modify myDsStud2.py into **myDsStud2File.py** to (1) generate a new key pair, (2) store the keys as file (3) read the keys from file, (4) use the keys to sign or verify messages.
   1. Implement a RSA key pair generation program in Python and **store the generated keys** in the file **privatekey.der** and **publickey.pem** accordingly.
   2. Modify your solution to **sign** on a message (or a file) using the private key which is imported from the file **privatekey.der**.
   3. Modify your solution to **verify** a signed message (or a file) using the public key which is imported from the file **publickey.pem**.
3. Digital Signatures with DSA (Optional)

Repeat your exercise 1 with DSA key pair rather than RSA key pair. Your may refer to myDs\_DSA\_stud.py sample source to complete this task.

~End of the Practical~