CON101 **1 Nov, 2021**

Assignment 10: Digital Signature

Due Date: 10 Nov 2021

Problem Statement:

In this assignment, we are going to implement a working algorithm for digital signatures. This assignment has been divided into two parts:

1) For the first part, write a program to generate the public and private keys for given values of p, q and e. Take the values of:

- p = 43
- q = 47
- e = 155

[Where p and q are the starting prime integers and e is part of the public key]

Output:

- Euler's phi function φ
- Public Key (n, e)
- Private Key (n, d)

2) For the next section, consider two users, Alice (Sender) and Bob (Receiver) who are communicating with each other. To ensure the authenticity of her messages, Alice creates digital signatures and sends them along with her messages to Bob. Implement the following steps to be performed by Alice and Bob (Use the public and private keys generated in the first section):

- Alice (Sender) uses hash function SHA (256 bit) to find the hash value of message m i.e., h(m) [**Hint**: Use the hashlib library to get the hash message. You may use SHA-2 or any other hash function as per your convenience (including if you want to write your own)]

- Alice applies the RSA decryption key d_A of Alice on h(m). The output of this step is the Digital Signature (s) of Alice. i.e., s = D(h(m), d_A). [**Hint**: Break up your message into 32-bit integers and apply the RSA encryption function on each of the integers in the message. (You may use library functions for it). The output signature s should be a single string containing the encryption of each integer of digest separated by a '.']

- Alice sends message m and signature s to Bob.

- Bob computes the hash value of the received message m i.e., h(m).

- Bob applies RSA encryption key e_A of Alice on the signature (s) received from Alice i.e. E (s, e_A) = h (let).

- Bob compares h(m) and h. If both are equal, then Bob assured that the message received is correct and unaltered and coming from Alice.

Take message m = “2101-CON101 INTRODUCTION TO COMP.SC. amp; ENG”

Output:

- The digest of the hash value h(m) generated
- The message signature s
- The encryption h generated by Bob. [**Note**: It should be equal to h(m)]

Programming language: **Python / C / C++**

Be careful of overflows while performing large computations. For example, if you break up the hash message into 32-bit unsigned integers, you might require 64-bit long unsigned integers to perform the computations required. In C, in case of unsigned overflow, it may result in a 0.

Submission Instructions:

- Submit the code for the above problem
- Submit a pdf file explaining very briefly the steps performed to obtain the public and private keys and those used to encrypt and decrypt the messages.
- Include both the files in a single zip file and follow the naming convention “<Entry_Number>_Assignment10”.zip while submitting on Turnitin.