Monsoon 2020
FCS Assignment 2

Part I

1. Given the values of p = 13 ; q = 31 ; d = 7 and find e = ?
(Reference is taken from Supplementary material)

n = (p)(q) = 13 * 31 = 403

Totient(n) = (p-1) (q-1) = (12)(30) = 360

As given d = 7, RSA algo says that, **(e)(d) = 1 mod 360**

This can also be written as, (e) (7) mod 360 = 1

From trying different values for e, the value of e can be **103.**

$$\boxed{\mathbf{e = 103}}$$

2.
   a. Cryptanalysis is the process of studying cryptographic systems to know their weakness or loss of data. Generally, cryptanalysis deals with the mathematics of the system but sometimes also looks at the implementation details as there might be a chance to attack a communication channel. The requirements to break a cryptosystem are, Cracking the passwords or the keys used in the process of encryption. Understanding the process of encryption and decryption will also lead to attacks by outsiders. These attacks are again classified into active attacks and passive attacks.
   Reference:- https://www.sciencedirect.com/topics/computer-science/cryptanalysis

   b. Reference:- https://www.tutorialspoint.com/cryptography/cryptosystems.htm

   Kerchoff proposed six design principles or requirements for a cryptosystem, they are:-
      1. The cryptosystem should be unbreakable practically, if not mathematically.
      2. The cryptosystem has to be secure even if it falls in the hands of an intruder without causing any inconvenience to the users.
      3. The key should be easily communicable, memorable and changeable.
      4. The ciphertext should be transmitted by the unsecure channel.

5. The encryption process and apparatus should be done and maintained by a single person.
6. The system should be easy to use without requiring the knowledge of memorising the long process.

c. Provable security is a level of computer security which can be proved. Mostly, these proofs are mathematical and emphasise on the mathematics of the cryptosystems ignoring the side channel attacks as these side channel attacks cannot be known until the implementation.

If the mathematics of the cryptosystem is as hard as a computationally not solvable problem, this proves the security of the cryptosystem. So, the base assumption is, the cryptosystem algorithm should always be as hard as a computationally unsolvable problem.

Quantum computers are so fast and highly reliable in computations when compared to normal computers. As these computers are fast in computations, these can solve problems easily even if the problems are NP-hard and NP-complete.

d. Even after the post quantum world, the security by obscurity will help to some extent. It is sure that if one knows the obscurity the system is vulnerable again but having security to some extent is better than having no security.

3. Location based access control is based on the location of the device. For example, if we want to access the ERP of IIIT Delhi, we should be in and around the campus, and if we move out of the campus,the site cannot be accessed. Similarly, drones can be flown at any places but not in the surroundings of military areas.

Situation aware access controls can also be explained through an example like students will be allowed to look into books while writing some open book examinations, but are not allowed to see their books for all the exams.

Location based access controls can be implemented in mobile phones using the geographical locations of the mobile. Situations change with the time always. So, situation aware access controls can be implemented based on the time of the mobile phone.

Part II

1. **Assumption:** The length of p,q and m has to be the same for getting the values correct.
   a. I have used gmpy2 extensively and calculated the required output taking the help of supplementary material. The method of calculating 'e' is described in 1(c). 'd' is calculated using the invert() function in gmpy2 library. For getting the ciphertext I used the powmod() method in gmpy2 library.

   b. Decryption is also done using powmod() method.

   c. https://crypto.stackexchange.com/questions/35499/can-d-and-e-be-the-same-number-in-rsa
   From this reference it says that, " e is relatively prime to (p-1) and (q-1) or equivalently relatively prime to lcm of (p-1),(q-1).
   As p and q are considered to be prime, (p-1)(q-1) will always be an even number. Our task is to find 'e' to be coprime to (p-1)(q-1) i.e, gcd=1.
   This can be done by selecting any prime from range 1 to totient_n whose gcd will be 1. Instead of selecting randomly, I am getting the next_prime number from totien_n/2.
   For eg. if totient_n=50, 50/2=25. I will call next_prime function on 25 and it gives 29 as a result and 29 is co-prime to 50.

2. **Assumption:** All p,q and m should be of the same length.
   a. All the steps are done using the supplementary material. The 'k' is selected as a random number in the range of n. Not used any loops and all the constraints are met.

   b. Decryption is also done using the steps from supplementary material. Loops are not used and constraints are met.

   c. Vanilla RSA:-
      Encryption:-
         C = m**e mod n
      Decryption:-
         m = C**d mod n
   Dependent RSA
      Encryption:-
         C1 = (k+1)**e mod n
         C2 = m * [k**e mod n]
      Decryption:-
         k = (C1**d mod n) - 1
         m = C2 / (k**e mod n)

For comparing Dependent RSA with Vanilla RSA, we will consider Vanilla RSA as holds true and tries to prove Dependent RSA.

Consider “C1=(k+1)**e mod n” from encryption and “k=(C1**d mod n)-1” from decryption as one set. As ‘e’ and ‘d’ generation in Dependent RSA is the same as Vanilla RSA, i.e ed mod totient(n) is 1 and d is co-prime to totient(n). From this information, we can say that the above decryption equation holds true with respect to the above encryption equation.

Consider “C2=m*[k**e mod n]” from encryption and “m=C2/(k**e mod n)” from decryption as one set. These two can be easily said as valid by simple equation transformations. From this we can say that, we can obtain the plaintext message ‘m’ from C2.

References for programming assignment:-

- https://readthedocs.org/projects/gmpy2/downloads/pdf/latest/
- https://crypto.stackexchange.com/questions/35499/can-d-and-e-be-the-same-number-in-rsa
- https://en.wikipedia.org/wiki/RSA_(cryptosystem)