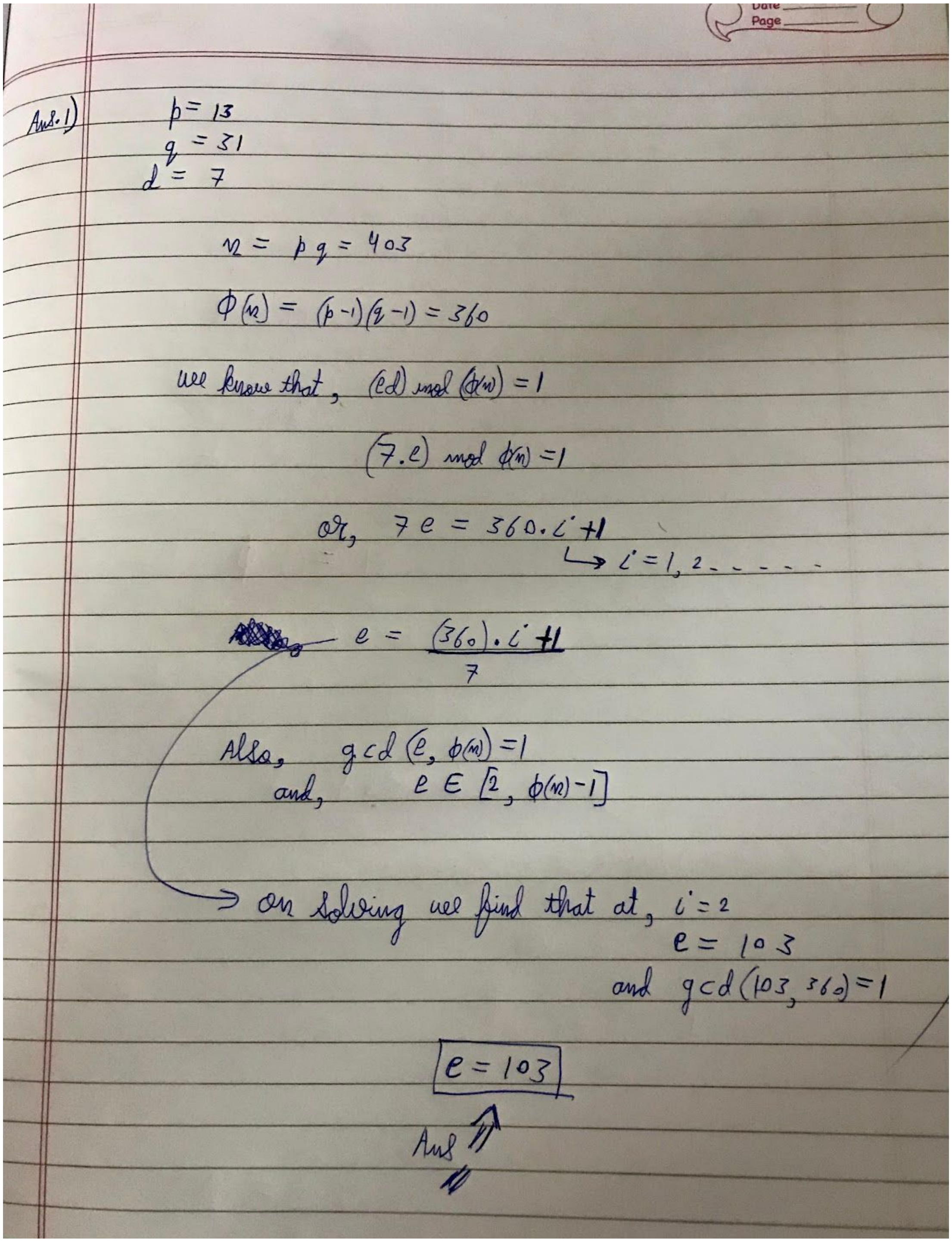
Assignment- 2

Harshit Rai
2017152

Part- I

Question- 1



Date
Page

Ans.1) $p = 13$

$q = 31$

$d = 7$

$n = pq = 403$

$\phi(n) = (p-1)(q-1) = 360$

we know that, $(ed) \bmod (\phi(n)) = 1$

$(7.e) \bmod \phi(n) = 1$

or, $7e = 360.i + 1$

$\hookrightarrow i = 1, 2 \ldots$

$e = \frac{(360).i+1}{7}$

Also, $gcd(e, \phi(n)) = 1$

and, $e \in [2, \phi(n)-1]$

→ on solving we find that at, $i = 2$

$e = 103$

and $gcd(103, 360) = 1$

$\boxed{e = 103}$

Ans

Question- 2

A) Requirements to break a cryptosystem:

- A machine with very high **computing power**.
- **Availability** of the cyphertext and its corresponding plaintext.
- A **big dataset** to make an efficient deciphering tool.
- **Large storage devices** to store the plaintexts and the cyphertexts.

B) Measures/rules/parameters to evaluate the strength of a cryptosystem are:

- How strong is the cryptosystem's **"one-way trapdoor function."** It is hard to do the reverse engineering when given only the private keys, and you have to guess the correct corresponding public keys.
- The total time required to perform the fastest known attack on a specified processor should be large. That is, **"Attack Time"** should be big.
- There must not be an attack faster than the **brute force attack**.
- The number of rounds also plays a role. Because more rounds lead to greater **"confusion" and "diffusion."**
- Time taken for reverse engineering should increase significantly with the increase in the **length of the keys**.

C) Base assumption that all secure cryptosystems hold are:

- The **"one-way trapdoor function"** should be strong. Give the plaintext, we can easily encrypt it, but the reverse process must be very hard. It must take many years to decipher it correctly.
- **Semantic Security**- The adversary must not gain any information regarding the plain text when the corresponding ciphertext is given. That is, the cyphertext does not leak any information about the plaintext.
- **Kerckhoff's Principle**- The system must be secure even if everything except the private key is known to the adversary.

D) Security through obscurity is not a good idea for a security system because everything is lost as soon as its secrets are revealed/breached. That's why it was not considered a good idea, even in the pre-quantum world. But on the other hand, provable security might lose their relevance in the quantum world. Using security through obscurity as an additional layer and the provable security algorithms can result in a robust encryption scheme. Relying solely on security through obscurity can cause a catastrophic disaster in the future.

Question- 3

In location-based access control, access to various data/objects is done based on the location of the user/subject. In terms of computer security, location can be measured in terms of the user’s IP address or GPS coordinates. Whereas in the case of situation-aware access control, the access to various data/objects is done based on the situation, condition, or environment of the user/subject. For example, in a classroom, only a user with the teacher’s role during the “class time” can create a group discussion.

LBAC can be achieved by enabling the GPS of the mobile phone, which will track the location of the user. And based on their locations, various roles can be assigned. For example, a company’s portal can only be accessed fully within the company and not from any other location. This can be done easily using LBAC by maintaining the GPS coordinates of the user.

SAAC can be achieved in a mobile phone in case of an emergency like for a terminal patient, their body activities can be measured, and details can be measured by the phone. If the value rises or reached some particular threshold point, then according to the severity of the situation, different roles can be assigned to the user like he can directly contact the doctor or can be given much more powerful features that were nor given in his usual state or condition.

Part- II

Question- 1

C) We have selected ‘p’ and ‘q’ to be prime numbers such that ‘p’<‘q.’ Now we have to find the value of ‘e’ in the range [2,phi_n-1] such that ‘e’ and ‘phi_n’ are relatively primes or co-primes to each other. Numbers ‘a’ and ‘b’ are co-primes if and only if GCD(a,b)=1.

If we find the next prime number greater than ‘q,’ then this new prime number and ‘phi_n’ will always be co-primes. This is proved in the images given below.

Ans. 1) (c) We are given that, $p < q$ and $p > 3 \Rightarrow q > 3$

So, $p, q > 3$

$n = pq$

$$\phi(n) = (p-1)\cdot(q-1)$$

If we choose, e = next-prime $(q) \Rightarrow$ It will always be co-prime to $\phi(n)$ or GCD $(e, \phi(n)) = 1$. Because $e > (p-1)$ and $e > (q-1)$.

So both $(p-1)$ and $(q-1)$ can not be a factor of e. Also e can not have any factor other that 1 and e because e is a prime number.

Therefore, e will be relatively prime with $(p-1)$ and $(q-1)$

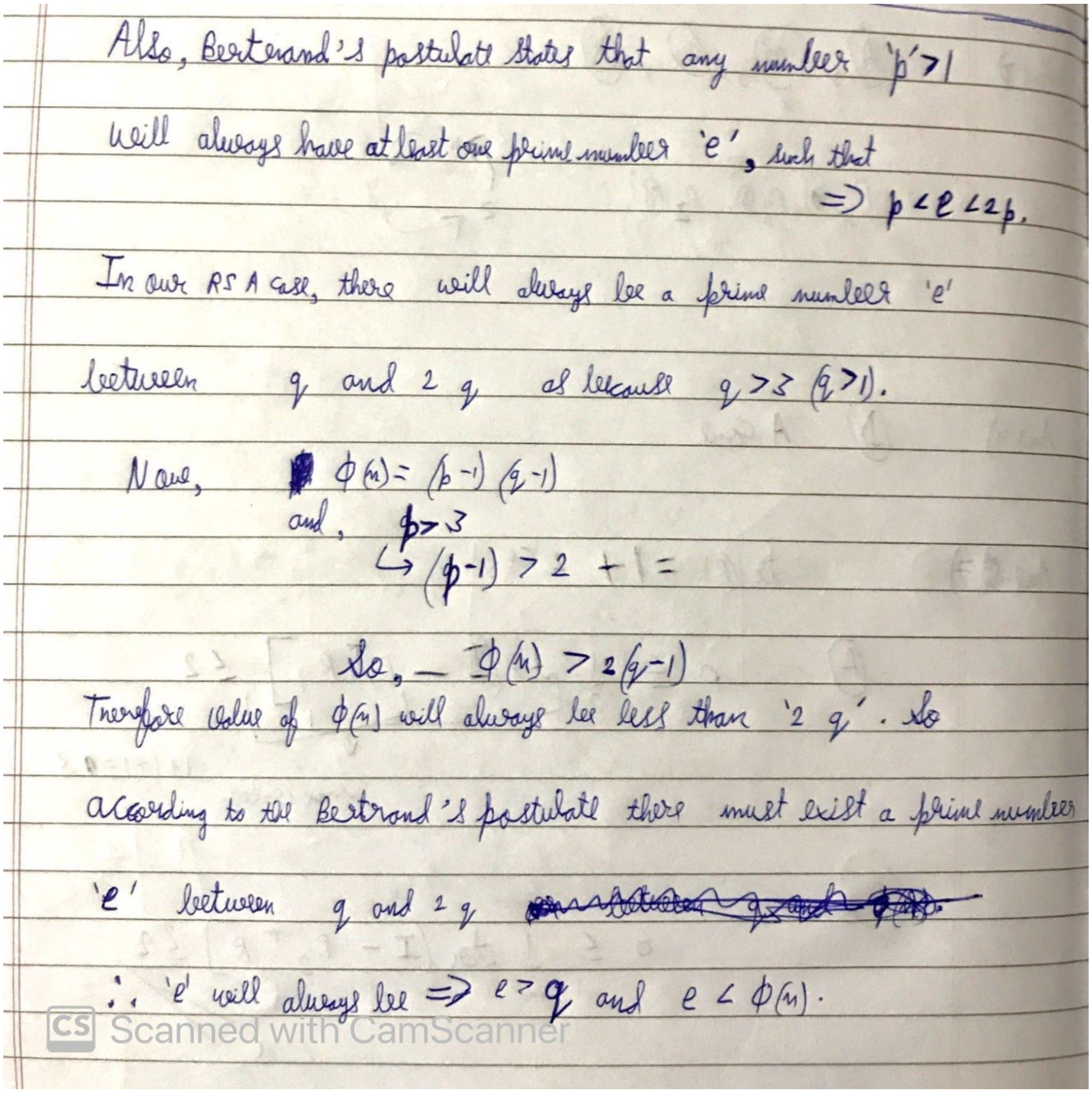
or, e is relatively prime with $\phi(n)$.

↳ $GCD(e, \phi(n)) = 1$.

And due to Bertrand's postulate, $e < \phi(n)$ ~~and~~.

Therefore will get a co-prime number e with $\phi(n)$ when

e = next-prime (q).



Also, Berterand's postulate states that any number 'p' > 1 will always have atleast one prime number 'e', such that

$\Rightarrow p < e < 2p.$

In our RSA case, there will always be a prime number 'e' between q and 2 q as because $q > 3$ ($q > 1$).

Now, $\phi(n) = (p-1)(q-1)$

and, $p > 3$

$\hookrightarrow (p-1) > 2 + 1 =$

So, — $\phi(n) > 2(q-1)$

Therefore value of $\phi(n)$ will always be less than '2 q'. So according to the Bertrand's postulate there must exist a prime number 'e' between q and 2 q ~~between q and $\phi(n)$~~

$\therefore$ 'e' will always be $\Rightarrow e > q$ and $e < \phi(n)$.

Question- 2

C) **Comparison between Dependent RSA and vanilla RSA:**

- The Dependent RSA is more secure and difficult to break as compared to the vanilla RSA because of the fact that dependent RSA requires additional keys for encryption and decryption.
- Dependent RSA will be computationally more expensive as compared to vanilla RSA because of more number of keys in dependent RSA (C1 and C2).
- As RSA is not completely semantically secure but its variation, dependent RSA is semantically secure.
- The decryption of dependent RSA uses a random integer 'k' in the residue class of Zn*.

why Dependent RSA works:

- Like is vanila RSA, the process of encryption involves -> C = (m^e) mod(n).
- In DRSA, we just replace C by C1 and m by (k+1). Here ‘k’ is a random integer in the residue class of Zn*. It is a set of integers from 1 to N-1 (both includind) which are relatively prime to N.
- So, C1 = [(k+1)^e] mod(n).
- As ‘p’ is a prime factor of ’n’. Therefore all the integer from 1 to ‘p-1’ (both including) will be relatively prime to ‘n’.
- Also, C2 = m * [(k^e) mod(n)]. Using the 2 equation, the decryption is done.
- Encryption of vanila RSA is done by -> m = (c^d) mod(n). In DRSA, we just replace C by C1 and m by (k+1). So, k = [(C1^d) mod(n)] - 1.
- And, m = C2 / [(k^e) mod(n)]. No, change has been made on this equation, we simply rearrange the equation in the decryption process.
- Using these equations, we can decrypt the message successfully.
- As we know that vanilla RSA works, therefore DRSA also works because we have proved it using vanila RSA equations.

References

1. CRYPTOGRAPHIC ALGORITHM METRICS
2. https://iopscience.iop.org/article/10.1088/1742-6596/1542/1/012024/pdf