Project 3:
CS6035

Jingying Liu
jliu929@gatech.edu

1 TASK 2

From task 2 we can see it is easy to guess simple passwords even when we know hashed passwords cannot be reversed. Hence, we need to ask people to make passwords more complicated and harder to guess. By simply implementing periodical force update of secure passwords, the network security will be greatly improved. Multi-factor authentication is also a good idea to improve security.

2 TASK 3

Proof-of-work (PoW) is the most popular and widely used consensus mechanism because of bitcoin. Proof-of-Stake (PoS), which is still under development, has the potential to replace PoW [1]. Unlike PoW where everyone can become a miner, not everyone can join in PoS except those who owns the currency [2]. Thus, the biggest advantage of PoS over PoW is that it saves greatly on costs or computing resources because there is no need to solve the puzzles like in PoW. The disadvantage, however, includes that PoS will not be able to generate the rewards for miners (i.e. bitcoin) as in PoW. Another disadvantage of PoS is that small numbers of people own most stakes because higher monetary value means higher stake [1].

3 TASK 4

The following steps were taken to compute the private key:

1) Firstly, get the p and q from N. This one is computationally hard and is the key reason why RSA is hard to attack. For smaller numbers as in this task, python can easily find the factorization.

2) Then the question really becomes how to get d from p, q and e. Here p and q will help us get to module (p-1)*(q-1). Then key question becomes how to get the modular inverse.
3) From Wikipedia [3], the best way to solve modular inverse is by extended Euclidean algorithm [4], which leveraged the recursive computation of successive quotients and remainders.

After all, three steps we will be able to find the private key d.

4 TASK 5

The public in this task is vulnerable because of the lack of randomness. We know RSA is hard to break because it is hard for the computer to easily break a 2,048 digits N into two 1,024 prime factors p and q. However, it is super easy to compute greater common divisor (GCD).

The lack of randomness because of the malfunctioning random number generators [5] will potentially have some Ns share the same p. Then once we have all the public keys available, it is easy to loop through all of them in pair and calculate the GCD p that is not equal to 1. Once we have p, q is simply N/p. Then by leveraging the steps illustrated in task 4, we will be easily getting to the private key.

5 TASK 6

The broadcast RSA can work because the small e makes it easy to reverse-engineer m without knowing the public key d. The vulnerability comes from the fact that we can easily leverage the Chinese Remainder Theorem [6] to get the m by only using encrypted messages with the corresponding public key (N,e).

Here are the steps used to recover the m:

1) Using Chinese Remainder Theorem step by step to get pow(m,3). I have followed the easy math tutorial example step-by-step to set up the variables. [7]

2) Calculate the log(m**3,3) using self-generated function. The python built-in function gave me wrong answer because of the large number input. I searched online and found the binary search algorithm can really help me in getting the answer quickly.

After taking the log using self-built function, we can get the m asked in this task.

6 REFERENCES

[1] Ali, A., Latif, S., Qadir, J., Kanhere, S., Singh, J. and Crowcroft, J., 2019. Blockchain and the future of the internet: A comprehensive review. *arXiv preprint arXiv:1904.00733*.

[2] Seang, S. and Torre, D., 2018. Proof of Work and Proof of Stake consensus protocols: a blockchain application for local complementary currencies. *France: Universite Cote d'Azur-GREDEG-CNRS. Str*, 3(4).

[3] https://en.wikipedia.org/wiki/Modular_multiplicative_inverse

[4] https://en.wikipedia.org/wiki/Extended_Euclidean_algorithm#Iterative_method_2

[5] Heninger, N., Durumeric, Z., Wustrow, E., & Halderman, J. A. (2012). Mining your Ps and Qs: Detection of widespread weak keys in network devices. In *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)* (pp. 205-220).

[6] https://en.wikipedia.org/wiki/Chinese_remainder_theorem

[7] Chinese Remainder Theorem (2019) https://www.youtube.com/watch?v=zIFehsBHB8o