CS 411-507 Cryptography

Homework #2

Yigit Kaan Tonkaz

29154

Q1

1. When I used getQ1 function to get the values of n and t, I get {'n': 359, 't': 179} from the server. Then I used the phi function because it returns a amount which is the number of coprime numbers to that input number. Therefore, in a way, I have reached the number of elements in that group. I get 358 from phi function and checked that using checkQ1a(amount) which prints Congrats!.
2. To find generators in Z\*359 we need to find positive integers that are less than t and co-prime to t which is 179. So I implement find\_generator function to find the generators list which takes the t number as input and check if it is coprime and less than t. Then I find the number of that is 178. Then I put the answer to the checkQ1b which prints Congrats! as well.
3. I implement a find\_generator2 function which takes n and t as inputs. It checks the current g from 2 to n-1, if it is coprime with n. Then it checks g^t mod n is equal to 1 which is based on g^t ≡ 1 mod n. And after that if both coprimality and correct order satisfies, it returns generator for the specific subgroup which I get 2. And I try with checkQ1c and it gives Congrats!.

Q2

I get the e and c from the server by using getq2 function. Compute m = cd mod n, we need to find d first. I used totient function because p and q are prime numbers, and we can get much smaller number by using totient which also decrease time to complete. So, I used Φ (p) = p – 1 and Φ (q) = q – 1. Then to find d, I used modinv with e and totient n. After finding d, we can find m by using the formula m = cd mod n. Finally, I get the m but its form is int, so we need to decode to it to the string that I used utf-8 which I found on google for decoding solution. Then I check it in checkQ2 and Congrats!. The decoded string is “I think I have 359 unread e-mails. Is that a lot?”

Q3

Q4

The solution exists If gcd(a,n) divides b for ax ≡ b (mod n). First, I calculate gcd(a,n) and used egcd to find x0 and y0. If b is divisable by gcd, we can find solution according to the

x ≡ x\_p (mod n/gcd(a,n)). I implement results function to find solution set. It calculates egcd and checks if b is divisible. Then it founds x\_p for solution or else no solution.

1. [1115636343148004398322135138661008357945126147114770093414826]
2. no solution
3. [1840451085636978827079830514312022149966941191143010614385900, 4573017168579321153146925263568627765759266852067838063135011]
4. [120574576795431477647425259344685590574672051332591719355582, 1692041454071987051397898895041599936536312796085130907016143, 3263508331348542625148372530738514282497953540837670094676704, 4834975208625098198898846166435428628459594285590209282337265]

Q5

Q6

To say it is predictable, we need to know the sequences and check if they display any predictable patterns. I try to use The Berlekamp-Massey algorithm to find lfsr that can generate a the given binary sequence. So I used the given BM function to get L from sequence. Then I check if it is equal to length.

X1-> unpredictable

X2-> unpredictable

X3-> unpredictable

Q7