CS411-507 CRYPTOGRAPHY HOMEWORK 2

Name: Elif Cemre Durgut

ID: 26493

Note: See the the .jpynb file in the zipped folder for my code. Alternatively, here is Google Colab link:

<https://colab.research.google.com/drive/1XAQPRpKm5cxuy_DdHivTKSvXyt2tl4Uu?usp=sharing>

**1)**

**1.a)** Values given by the server was n: 983. I calculated the number of integers from 0 to 983 such that gcd(num, 983) = 1. The answer is 982. This means it is prime number.

**1.b)** I tried 5 as generator and my code generated all the numbers from 0 to 983, so 5 is a valid generator.

**1.c)** I tried 2 and it worked.

**2)** The e and c values are relatively prime (coprime), because of that we can calculate phi(n) as (p-1)\*(q-1). As given in the question d = e^(-1) mod φ(n), I used modinv() function to calculate d. Then I converted integer d to binary and I applied left-to-right binary modular exponentiation method to calculate m = c^d mod n. Finally I decoded it, the answer is "Answer to the ultimate question of life, the universe, and everything is not 42. it is 192"

**3)** The question says two of the messages are corrupted. So first, I tried to decrypt each of them using their own nonces ciphertext[:8]. Message of the ciphertext2 was meaningful. This means ciphertext2[:8] is the nonce for all the messages. Then I tried to decrypt 1 and 3 by trial and error cropping the first bytes of the ciphertexts. I discarded first 6 bytes of ciphertext1 and 5 bytes of ciphertext3. Answers:

message1 = “I love deadlines. I love the whooshing noise they make as they go by”

message2 = “Our knowledge can only be finite, while our ignorance must necessarily be infinite”

message3 = “Any unwillingness to learn mathematics today can greatly restrict your possibilities tomorrow”

**4)**

**4.a)** gcd(a, n) = 1 so there is only one solution. ax ≡ b mod n 🡪 x ≡ b\*a^-1 mod n. I calculated a^-1 mod(n) using modinv function. Then I multiplied by n and took the modulus.

Answer: 56884393062303769019751445983612369117060043083722821988604

**4.b)** gcd(a, n) = 3, and 3 does not divide b, so there is no solution.

**4.c)** gcd(a, n) =3, and 3 divides b, so there should be 3 solutions. I divided all the parts by 3. And I repeated the same operations as in 4.a) and found answer1. To find other 2 answers, I added n to answer1. However, when I tested the mod equation, my founding is not equal to b and I could not find where the problem is.

Answer1: 9609279374756105288427021898499890361717105145551739027963

Answer2: 110042907140942997509799652683766709621865315673440026493694

Answer3: 210476534907129889731172283469033528882013526201328313959425

Answers:

**5)** I used FindPeriod function and entered the C array values as

**5.1)** C[0] = C[2] = C[5] = 1

and the period is always 31 = 2^5 – 1. So, it gives the maximum period.

Answer: Yes

**5.2)** C[0] = C[2] = C[3] = C[5] = 1

and the period changes in each run, it isn’t stable, and I’ve never seen it was 31. It does not give the maximum period.

Answer: No

**6)** I used Berlekamp-Massey algorithm given in the lfsr.py and it returned me the same linear complexity (31) and the shortest lfsr for all the binary sequences. However, the linear complexity should be half of the length of the sequence (31 < (84/2)). Also, I checked the number of 0’s and 1’s and none of them were equal. Therefore, they are not unpredictable.

**7)** I treated the known plain text “Erkay Savas” as seed and applied Berlekamp-Massey algorithm on it. It gave x^35 + x^33 + x^27 + x^25 + x^24 + x^18 + x^17 + x^16 + x^13 + x^2 + x^1. In this case, I need to try 2^35 different states to find the state where the “Erkay Savas” has started which is not feasible.