CS 58000\_01 Design, Analysis and Implementation Algorithms (3 cr.)

Assignment As\_02 (Exam 01)

Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This assignment As\_02 is due at 11:59 p.m., Sunday, October 1, 2023. Please submit your assignment to Brightspace (purdue.brightspace.com). No late turn-in is accepted. Please write your name on the first page of your assignment. Your file name should be your last name such as NgP\_As02.docx. Please number your problem-answer clearly such as Problem I.1.a, I.1.b, I.1.c, I.2, …, I.7, Problem II.1, II.2, II.3, II.4. The problems’ answers must be arranged according to the order of the given problem. Please answer your questions using only a Word file (.docx file only). No pdf file will be accepted. Without using a Word file (.docx file) the submitted problems’ answers would not be graded.

The total number of points for this Assignment\_02 (Exam 01) is 150 points.

Problem I [110 points]:

This problem is an exercise using the formalization of the RSA public-key cryptosystem. To solve the problems, you are required to use the following formalization of the RSA public-key cryptosystem.

Given the following formalization of the RSA public-key cryptosystem, each participant creates their public key (n, g) where a is a small prime number, and n is the product of two large primes, p and q. However, the two large primes p and q are secret keys.

1. Select two very large prime numbers p and q. The number of bits needed to represent p and q might be 1024.
2. Compute

n = pq

(n) = (p – 1) (q – 1).

The formula for (n) is owing to the Theorem: The number of elements in is given by Euler’s totient function, which is

where the product is over all primes that divide n, including n if n is prime.

1. Choose a small prime number as an encryption component g, that is relatively prime to (n). That means,

gcd(g, (n) ) = 1, i.e.,

gcd(g, (p-1)(q-1)) = 1.

1. Compute the multiplicative inverse That is,

The inverse exists and is unique.

That is, the decryption component h = g-1 mod (n).

1. Let pkey = (n, g) be the public key, and skey = (p, q, h) be the secret key.

* For any message M mod n, the encryption of M is C = Mg mod n.
* The decryption of C is M = Ch mod n.

End of the formalization of the RSA public-key cryptosystem.

Use the RSA Cryptosystem formalism for solving problem I.

Given g = 59, p = 991 and q = 997.

I.1. [30 pts.] Show that the given values of g, p, and q are prime,

I.1.a Use the Algorithm Sieve (the Sieve of Eratosthenes Method) to check whether p is a prime.

I.1.b Based on Fermat’s Little Theorem, the algorithm in Figure 1.8 can be used to check whether q is a prime. What is the most reasonable set {a1, a2, …, ak} that is used for applying this algorithm?

I.1.c. How do you check that g is a prime? Show the work of how you compute.

[Hint: There is a distinct difference between Algorithm Sieve and Fermat’s Little Theorem for finding the solution. The latter requires to apply “function modexp(x, y, N), Fig 1.4, Chapter 00\_03” to compute modular exponentiation. For the former one, the Algorithm Sieve deletes only the numbers which are divisible by the primes. If you use this approach, use a table to show the elimination process. If you use Fermat’s Little Theorem approach, show all the computations for modular exponentiation.]

I.2.[10 pts.] Compute n = pq and (n) = (p – 1) (q – 1).

I.3.[20 pts.] Given a plaintext **M = 506574**, what is the encryption of M, using

C = Mg mod n.

Show in detail how you derive C, which is the ciphertext of the plaintext M.

I.4.[20 pts.] Compute the multiplicative inverse That is, the decryption component h = g-1 mod (n).

[Hints: Compute a GCD as a Linear Combination. Then, find an inverse Modulo n. In other words, you can apply the extended Euclid algorithm to find the linear combination of g and Then, find a positive inverse of g mod.]

I.5.[10 pts.] From problem I.4, what is your secret key (p, q, h)?

I.6.[20 pts.] What is the decryption of C using M = Ch mod n? Show in detail how you derive M, which is the plaintext M of the ciphertext C.

**I.7 (Bonus)[5 points]:**

What is the message (in terms of the alphabet)?

Problem II[40 points]:

Assume that we define

h1(k) = └ m(k A mod 1) ┘, where m = 13 and A = 0.62,

and

h2(k) = 1 + └ m(k A mod 1) ┘, where m = 11 and A = 0.62,

For the open addressing, consider the following methods

**Linear Probing Linear Probing**

Given an ordinary hash function h: U {0, 1, 2, …, m-1}, the method of *linear probing* uses the hash function

h(k, i) = (h1(k) + i) mod m for i = 0, 1, 2, …, m-1.

**Quadratic Probing**

Uses a hashing function of the form

h(k, i) = (h1(k) + c1i + c2i2 ) mod m,

where h1 is an auxiliary hash function, c1 and c2 0 are auxiliary constants c1 3 c2 = 5,

and i = 0, 1, 2, …, m-1.

**Double hashing**

Uses a hashing function of the form

h(k, i) = (h1(k) + i h2(k) ) mod m,

where h1 and h2 are auxiliary hash functions.

The value of h2(k) must never be zero and should be relatively prime to m for the sequence to include all possible addresses.

Given K = {369, 119, 287, 712, 141, 503, 186, 295, 528, 625} and the size of a table is 13, with indices counting from 0, 1, 2, …, 12. , (a) compute their indices for storing all the keys in the given K in a table with the size 13, including resolving the collision if occurred. Then (b) show the resultant table with 10 given keys for each method applied:

II.1. if linear probing is employed.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

II.2. if quadratic probing is employed.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

II.3. if double hashing is employed.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

II.4. Compare the linear probing, quadratic probing, and double hashing in terms of the number of occurred collisions for the given K.

**Note: If you provide your answer in your handwriting, good handwriting is required.**

**Proper numbering of your answer to each problem is strictly required. The problem’s solution must be orderly given. (10 points off if not)**