M Tech I : Introduction to Computer Security

Autumn Semester 2017-18

Lab Assignment 2

Date of Submission : 25th OCT 2017

1. Write a program that computes $a^x mod\, n$ with a, x and n input by the user such that $a \nless 1000$, $x \nless 100$ and $n \nless 2048$.

2. Write a program that takes as input an index for the Galois field and outputs order of the field. IT then prints all the elements in that field.

3. Write a program to compute two prime factors of a given input integer n. Your program MAY use the following steps (brute force) viz. *(1) While n is divisible by 2, print 2 and divide n by 2. (2) At this stage having executed step 1, n ought to be odd. Now start a loop from i = 3 to square root of n. While i divides n, print i and divide n by i, increment i by 2 and continue. (3) If n is a prime number and is greater than 2, then n will not become 1 by above two steps. So print n if it is greater than 2.*. Your program must give the option to the user to input the number n to be either a two-byte integer (data type int in C) OR a a 4-byte integer (data type long or $uint32_t$ in C) OR an 8-byte integer ($uint64_t$). Comment on your observations whether all of these versions work efficiently (or work at all!!)on your system or not.

4. Write a program to compute the Euler's Totient function $\phi(n)$ of the given input number n. Note that if the input number is a prime p, the Euler's Totient function is (p-1). However, if the input number is not a prime, it must find two of the prime factors of the given input number and compute $\phi(n)$. Alternatively, the other method (brute force) is based on computing all the number x of all the numbers that are relatively prime to the input number n and output that number x as the result of $\phi(n)$.

5. Write a program that implements the Extended Euclidean Algorithm to find inverse of a given number in the Galois field. The operation of the algorithm was illustrated in the class and the psuedocode is given in the class handouts with numerous illustrations in the handout.

6. Write a program that given an index of Galois field and an number, outputs the order of that input number. Your program must validate the number inputted before computing the order of the same.

7. Imagine an application to illustrate the use of the Fermat's little theorem (different from the one illustrated in the class) and implement the same in the form of a program.

8. Write a program that takes an input and outputs whether the group indexed by the input number is a cyclic group or not. IF the group is a cyclic group, your program must also output the number of generators in the group and each of those.

9. Write a program to illustrate all the properties of generators of an input field Z_n^*

10. Write a program that given an index of a multiplicative group, computes the quadratic residues set of that field.

*****

***

*