**CIS 3362 Homework #3**

**Number Theory, RSA**

**Check WebCourses for the due date**

**Note: This is a group assignment. Each pair of students should turn in one write-up in class on the due date. Both students' names should clearly be at the top of the front page of the assignment. Students may also turn in the assignment by themselves if they do not wish to have a partner.**

**1)** Determine the following values:

a) Φ(7402752) b) Φ(11165) c) Φ(123456789) d) Φ(69500000) e) Φ(10967535067)

**2)** Without the aid of any computing device, show how one can use Euler's Theorem to determine the remainder when 615604 is divided by 1925.

**3)** A primitive root, α, of a prime, p, is a value such that when you calculate the remainders of α, α2, α3, α4 , ... , αp-1, when divided by p, each number from the set {1, 2, 3, ..., p-1} shows up exactly once. A fast way to check if a random integer x a primitive root of a prime p is to calculate each unique prime factor qi of p – 1, and calculate x(p-1)/qi mod p for each qi. If none of these are equal to 1, then x is a primitive root.

One reason for the difficulty of the discrete logarithm problem is that the sequence of numbers generated by a primitive root (α, α2, α3, α4 , ... , αp-1 mod p) doesn’t seem to have any discernible pattern. If something doesn’t seem to have a discernible pattern, it’s a good candidate for a pseudo-random number generator.

For this question, we’ll test one idea utilizing primitive roots to see if we can create an algorithm that passes the random bits tests mentioned in class.

Generate 20,000 random bits and store these in a file (whatever format you would like is fine) using the following algorithm:

1) Pick a random prime number, p, with p > 106.

2) Randomly pick integers until you identify a generator for this prime number, call this g.

3) Use a time function on your system to return a value (in C this is seconds past 1/1/1970) and then take this value mod 100. Call this s.

4) Calculate x = gs mod p.

5) Generate the 20,000 bits (gx mod p) mod 2, (gx+1 mod p) mod 2, …, (gx+19999 mod p) mod 2. Basically, look at the portion of the sequence generated that starts at exponent x and for each of the next 20000 terms, look at whether or not it’s even or odd.

Now, check to see if this passes the random bits tests described on the next page:

a) The number of 0s and 1s is in the range [9654, 10346].

b) If we break the 20000 bits into 5000 blocks of 4 bits and interpret these values from 0 – 15, letting ni (0 ≤ i < 16) represent the number of blocks with value i, and let

The test is passed iff 1.03 < X < 57.4

c) Split the 20,000 strings into maximal runs of 0s and 1s. (For example, the string 00111010111110 contains the maximal runs 00, 111, 0, 1, 0, 11111, 0, which is 4 runs of length 1, 1 run of length 2, 1 run of length 3 and 1 run of length 5.) The number of runs of each length should be in the following ranges:

|  |  |
| --- | --- |
| **Length of Run** | **Required Interval** |
| 1 | 2267 - 2733 |
| 2 | 1079 – 1421 |
| 3 | 502 – 748 |
| 4 | 223 – 402 |
| 5 | 90 – 223 |
| 6 or more | 90 – 223 |

**4)** Two separate RSA keys both use the same value of n = 1397197. In particular, in one of the sets of keys, e = 898909 and in the other set of keys, e = 884255. It is known that the same message M has been encrypted using the public keys above yielding the ciphertexts 1106427 and 422068, respectively. Determine integers x and y such that 1106427x + 422068y = 1. Consequently, determine the original value of M without ever finding  or either value of d. (Hint: Remember what it means to raise a value to a negative exponent – first raise it to the -1 power, and then raise that result to the corresponding positive power. Furthermore, remember that raising a value to the -1 power means finding its modular inverse.) **Please show each step of your work. If you use one/edit one of the programs shown in class or write your own code, please include that in your write-up.**

**5)** Alice and Bob want to set up a new secret key using Diffie-Hellman. Their public key values are p = 12346123784612378461291 and g = 97997897867.

Alice sends Bob the following value: 9583793135283340325422.

Bob’s secret number is 12341234123432431355555.

What does Bob send to Alice, and what is the secret key value?

**6)** Try to break this message. Numbers of interest are: 926002123823977 and 3156478917144031.

299895574003382 2781819096562037 2870170500708305 600200268479477