

Homework 2 of CS549 Cryptography and Network Security

ASSIGNED: SEPT. 21ST, 2010

DUE DATE: 11:59PM, OCT 23RD, 2010

EXTENSION DATE (WITH 10% DEDUCTION ON GRADING): 11:59PM, OCT 30TH, 2010

FALL 2010, CS DEPARTMENT, IIT

There are 7 questions in HW2. Please type your answer. You can have to upload the PDF file of your solution (for non-programming part) to IIT blackboard system. In addition, you could also print it and give TA the hardcopy, if you are main campus student (you do NOT need to print the programming code).

Keep in mind that you have to upload the electronic file in PDF or Postscript format, and also the zipped file of your programming to the blackboard. Put your name on your solution and name your file as "lastname-HW2.pdf", where lastname is your last name.

For the last question, you also need to demo your code to TA before the deadline.

Notice: We do not accept MS WORD file and an electronic submission is mandatory.

1. (10 points) Assume that there is a block cipher, named XXX, that always encrypts block of $b = 72$ bits using key of $k = 60$ bits. Assume that we know that XXX will be broken even using simple brute-force attacking by guessing the encryption key. To enhance the security level, assume that someone proposes to use 2XXX by use of 2 encryptions of XXX with 2 different independently and randomly chosen keys. Prove that 2XXX does not provide a much stronger security than XXX. Prove in detail why this statement is true (you have to analyze in detail a method attacking 2XXX using time that is not much longer than the brute force attacking on XXX).

2. (10 points) Continue from preceding question. Assume now that someone wants to use 3XXX to enhance the security by using 3 rounds of encryptions of XXX with 3 different and independent keys. Design a method that can attack 3XXX whose time complexity is with order of $2^{2k} = 2^{120}$ instead of naive complexity of 2^{3k} . In your method, how many pairs of plaintexts and ciphertexts do you need? What is the space complexity your method will need? Given this many pairs of plaintexts and ciphertexts, what is the probability that you will find the correct encryption key?

3. (10 points)

This problem is about Hill cipher system. Assume that in a Hill cipher system, the input alphabet is $\{0, 1, 2, \dots, 24, 25\}$. Assume that each time it encrypts m characters. The key K is then a $m \times m$ matrix over Z_{26} . For any input $x = (x_1, x_2, \dots, x_m)^T$, we compute the encryption as $y = K \cdot x \pmod{26}$.

Suppose that Oscar has learned that a plaintext

1, 2, 3, 4, 5, 6, 0, 3, 7

is encrypted by Alice as

16, 7, 3, 11, 6, 25, 23, 2, 24

and also Oscar knows that $m = 3$.

(a) What is the key K used by Alice?

(b) If Oscar intercepts another ciphertext (2, 5, 7) encrypted by Alice using the same key, what is the plaintext corresponding to this ciphertext?

4. (10 points) Prove that RSA algorithm works. In other words, prove that $(M^e)^d = M \pmod{n}$ for any message $M \in [0, n - 1]$. Here $n = p \cdot q$, $e \cdot d = 1 \pmod{\phi(n)}$ and p, q are large prime numbers.

5. (10 points) This question is about RSA again. Assume that Bob uses RSA and selects two "large" prime numbers $p = 101$ and $q = 73$.

How many possible public keys Bob from which Bob can choose?

Assume also that Bob uses a public encryption key $e = 91$. Alice sends Bob a message $M = 2008$. What will be the ciphertext received by Bob? Show the detailed procedure that Bob decrypts the received ciphertext.

6. (10 points) A common way to speed up the RSA decryption is to use the Chinese Remainder Theorem. Suppose that we want to decrypt a ciphertext y and the decryption key is d and the modular $n = p \cdot q$, where p and q are two large prime numbers used by RSA. Let $d_p = d \bmod (p-1)$ and $d_q = d \bmod (q-1)$; and let $M_p = q^{-1} \bmod p$ and $M_q = p^{-1} \bmod q$.
 Let $x_p = y^{d_p} \bmod p$ and $x_q = y^{d_q} \bmod q$. Let $x = M_p \cdot q \cdot x_p + M_q \cdot p \cdot x_q \bmod n$.
 Prove that the computed x is indeed the original plaintext, i.e., $x = y^d \bmod n$.
 Given $p = 1511$ and $q = 2003$ and $y = 152702$, use the above method to decrypt the ciphertext y when the decryption key is $d = 153$.
7. (40 points) Implement the following three methods 1) Miller-Rabin primality test method, 2) Solovay-Strassen Primality testing method 3) AKS primality testing method. Your program should be able to accept a txt file as an input that contains an arbitrarily large integer (the number of bits could be thousands) to be tested whether it is a prime number. The output of the program should be able to say whether the number is a prime number; if it is not, it will produce a prime factor of this number. Here the number to be tested could have arbitrary number of bits (which is given from command or an input file and the value could be around thousands, say from 1000 to 3000 bits).
 Based on this code, write another code that will produce a random prime number of a given number of bits (where the number of bits is given as input to your code). The prime number produced should be output to a file.
 Compare the time used by these three methods to test whether a given number is a prime number. Plot the time used by these three methods to produce a random prime number of 10 bits, 100 bits, 200bits, 300bits, ... and 1000 bits.