M Tech I : Introduction to Computer Security

Autumn Semester 2017-18
Lab Assignment 3
Date of Submission :22nd Nov 2017

1. Consider the DES cipher. Write a program that takes sample input plain text (e.g. M=0123456789ABCDEF), and a cipher key K (e.g. K=1334 5779 9BBC DFF1). The program should then compute and output the following
    (a) The round keys (subkey) for each round of the DES cipher.
    (b) Assuming that the initial permutation and the final permutation are also applied, the intermediate ciphertext at the end of the each round of encryption as well as the final ciphetext.

2. Assume that the modern day computer works at the clock rate of 3 GHz. Given this speed, estimate the amount of time necessary to crack a DES encryption by testing all 2^{56} possible keys. Make a similar estimate for a 128-bit AES key. Assume that a machine takes a hundred cycles per brute force against a single 56-bit DES key or 128 bit AES key.

3. Research the literature and enlist at least three different attacks probable against the DES cipher and against the AES cipher, each.

4. TEA family of ciphers are known to be tiny encryption ciphers as are the NIST standards SIMON and SPECK. Write three different programs that implements the TEA, XTEA and XXTEA ciphers. Assume the plaintext is any alphanumeric character string of not more than 300 characters.

5. Just as the TEA family of ciphers are known to be tiny encryption ciphers so are the NIST standards SIMON and SPECK. Write two different programs that implement the SIMON and SPECK ciphers. Assume the plaintext is any alphanumeric character string of not more than 300 characters.

6. Write a program that illustrates the operation of the HMAC routine to compute the MAC.

7. Consider the state array input to the AES cipher and the user input key as shown in Table-1. Using the S-box, Round constant from the Class Handouts, write a program that computes and outputs the ciphertext at the end of the first round of the AES cipher.

    Table-1

State array matrix:	User Input Key matrix:
34, 9a, 35, e4	4b, 4d, bc, 09
43, 5b, 31, 37	34, 92, 35, e4
f8, 34, 9b, 0a	e8, 67, 9a, 0f
a9, 8e, a2, 34	f8, 9f, f2, 73

8. For the AES cipher, write a program that takes as input any arbitrary plaintext and any arbitrary cipher key and outputs the ciphertext at the end of each round in the AES. Your program must store the S-box in the memory and use it as a lookup table as and when required. Call this the speed-optimized version of AES - for every S-box substitution, it is sufficient to merely lookup the S-box, rather than computing each entry at the time of substitution.

9. Write a program that, given a sequence of integers of maximum 6 numbers, outputs a message whether the given sequence is a superincreasing sequence or not.

10. Write a program that implements the Merkle Hellman Public Key Cryptography Algorithm. Your program must first ask the user to input (1) a superincreasing sequence of ONLY a fixed size i.e. of 6 integers viz. $< b_1 \quad b_2 \quad b_3 \quad b_4 \quad b_5 \quad b_6 >$, (2) a system modulus M and check whether $M > (b_1 + b_2 + b_3 + b_4 + b_5 + b_6)$, (3) a random number W such that $1 <= W <= M - 1$, and $gcd(W, M) = 1$. The program must validate all the inputs as per the requirements of the algorithm (e.g. - the program should not accept the input M unless $M > (b_1 + b_2 + b_3 + b_4 + b_5 + b_6)$ and so on....).

Subsequently, the program must output the message "The Private key of the MH Cryptosystem is ______" and "The Public Key of the MH Cryptosystem is______". Next, the program should ask the user for the input plaintext - an alphanumeric string of maximum 10 characters and output the encrypted ciphertext. The program must then decrypt the ciphertext and output the decrypted plaintext message.

11. Write a C/JAVA program to implement RSA Cryptosystem and demonstrate the encryption/decryption of the input plaintext message of upto a maximum of 20 characters. The program then asks the user to input the following and does appropriate validation of the input as per the requirements of the RSA algorithm before proceeding further with encryption: (1) the prime factors p and q (data type long) and the encryption exponent e. Obviously, as part of the design your program must also compute the (a) Euler's Totient function $\phi(n)$, (b) check whether $gcd(e, \phi(n)) = 1$ and (c) compute the decryption key (d,n) such that $d = e^{-1} mod \phi(n)$ (d to be computed using the Extended Euclidean algorithm to find inverse). The program must output the computed ciphertext and then decrypt the ciphertext and print the decrypted plaintext message.

*****

***

*