1172/DCP1323 Introduction to Cryptography, Spring 2019 **Due: 2019/4/22 (Mon.)**

Homework 4

Instructor: Prof. Wen-Guey Tseng Scribe: Hung-Yu Chiu

Part 1: Written Problems

1. If we take the linear congruential algorithm with an additive component of 0,

$$X_{n+1} = (aX_n) \bmod m$$

Then it can be shown that if m is prime and if a given value of a produces the maximum period of m-1, then a^k will also produce the maximum period, provided that k is less than m and that k and m-1 are relatively prime. Demonstrate this by using $X_0 = 1$ and m = 31 and producing the sequences for $a^k = 3, 3^2, 3^3$, and 3^4.

2. Alice and Bob agree to communicate privately via email using a scheme based on RC4, but they want to avoid using a new secret key for each transmission. Alice and Bob privately agree on a 128-bit key k. To encrypt a message m, consisting of a string of bits, the following procedure is used.
   (a) Choose a random 64-bit value v
   (b) Generate the ciphertext c = RC4(v || k) $\oplus$ m
   (c) Send the bit string (v || c)
   A. Suppose Alice uses this procedure to send a message m to Bob. Describe how Bob can recover the message m from (v || c) using k.
   B. If an adversary observes several values (v1 || c1), (v2 || c2), c transmitted between Alice and Bob, how can he/she determine when the same key stream has been used to encrypt two messages?
   C. Approximately how many messages can Alice expect to send before the same key stream will be used twice? Use the result from the birthday paradox described in Appendix U.
   D. What does this imply about the lifetime of the key k (i.e., the number of messages that can be encrypted using k)?

3. Suppose you have a true random bit generator where each bit in the generated stream has the same probability of being a 0 or 1 as any other bit in the stream and that the bits are not correlated; that is the bits are generated from identical independent distribution. However, the bit stream is biased. The probability of a 1 is 0.5 + p and the probability of a 0 is 0.5 - p, where $0 < p < 0.5$. A simple conditioning algorithm is as follows: Examine the bit stream as a sequence of non-overlapping pairs. Discard all 00 and 11 pairs. Replace each 01 pair with 0 and each 10 pair with 1.
   A. What is the probability of occurrence of each pair in the original sequence?
   B. What is the probability of occurrence of 0 and 1 in the modified sequence?
   C. What is the expected number of input bits to produce x output bits?

4. In a public-key system using RSA, you intercept the ciphertext C = 20 sent to a user whose public key is e = 13, n = 77. What is the plaintext M?

5. Use the fast exponentiation algorithm to determine 6472 mod 3415. Show the steps involved in the computation.

6. Alice and Bob use the Diffie–Hellman key exchange technique with a common prime q = 157 and a primitive root a = 5.
   A. If Alice has a private key $X_A = 15$, find her public key Y_A.
   B. If Bob has a private key $X_B = 27$, find his public key Y_B.
   C. What is the shared secret key between Alice and Bob?

7. Suppose that Alice and Bob use an ElGamal scheme with a common prime q = 157 and a primitive root a = 5.
   A. If Bob has public key $Y_B = 10$ and Alice chose the random integer k = 3, what is the ciphertext of M = 9 ?
   B. If Alice now chooses a different value of k so that the encoding of M = 9 is C = (25, C_2), what is the integer C_2 ?

8. Consider the elliptic curve $E_7(2,1)$; that is, the curve is defined by $y^2 = x^3 + 2x + 1$ with a modulus of p = 7 . Determine all of the points in $E_7(2, 1)$. Hint: Start by calculating the right-hand side of the equation for all values of x.

9. This problem performs elliptic curve encryption/decryption using the scheme outlined in Section 10.4. The cryptosystem parameters are $E_{11}(1, 7)$ and G = (3, 2). B's private key is $n_B = 7$.
   A. Find B's public key P_B.
   B. A wishes to encrypt the message $P_m = (10, 7)$ and chooses the random value k = 5. Determine the ciphertext C_m.
   C. Show the calculation by which B recovers P_m from C_m.

Part 2: Programming Problem

This programming problem is to practice RSA encoding and decoding. Again, you can use either OpenSSL or Crypto++. Nevertheless, using Crypto++ in Visual Studio is preferred. Please find the related library information and examples on the Internet.

I. Read in the key length in decimal, a public key (e, n) in Hex and a message in ASCII and do encryption as described in the following table. The first entry is for testing and the rest is the problem. We only deal with one-block operation. You need to check whether the message length (in bits) is strictly shorter modulus n's length.

The ASCII message is treated as an integer, for example, “Hi” = “4869” (Hex) = 18537 (decimal). Since we are dealing with very long integer, please use “Integer” class for integer operations.

Key length	(e, n) (in Hex)	Message	Ciphertext (in Hex)
64	(11, ab9df7c82818bab3)	“Alice”	6414587f2f0ddb67
128	(11, cebe9e0617c706c632e64c3405cda5d1)	“Hello World!”	?
256	(11, af195de7988cfaa1dbb18c5862e3853f0e79a12bbfa7aa326a52da97caa60c39	“RSA is public key.”	?

II. Read in key length in decimal, private key (d, n) in Hex and a ciphertext (in Hex) and do decryption as described in the following table. The first entry is for testing and the rest is the problem. Note that the public key is 11 (Hex) or 17 (decimal) if it is needed.

Key length	(d, n) (in Hex)	Ciphertext (in Hex)	Message (in ASCII)
64	(111242af5740d14d, ae20a831558c0d69)	194d5cdc0ec8efbc	“Secret”
128	(974f3eaa763ad0979644dbfaac47867bd87b4c5c8b7fcd72943d0dde4303639, a0c432951d9e7da10fa929ba570bfee52db56fc477e60b742581a35d1723ad6f)	404ea0a1c26fc6562ff17a61849520e0fdf70654c6460b0954918e8447d6cdba	?

III. Submission: you need to upload three files: rsa-encrypt.cpp, rsa-decrypt.cpp and out.txt, where out.txt consists of three lines for the answers in the above problems.

IV. On-site test: Will announce the venue and schedule later. The problem is to use your programs to decrypt some ciphertexts on the spot. You need to pass the onsite test to get the credits of this programming problem.

V. If you want to generate some RSA keys for practice, try the following program segment:

```
// random number generator
AutoSeededRandomPool rng;

InvertibleRSAFunction parameters;

// Generate RSA keys with key_length bits
int key_length = 256;
parameters.GenerateRandomWithKeySize(rng, key_length);

const Integer& n = parameters.GetModulus();
const Integer& p = parameters.GetPrime1();
const Integer& q = parameters.GetPrime2();
const Integer& d = parameters.GetPrivateExponent();
const Integer& e = parameters.GetPublicExponent();
```