Introduction to Cryptography, Spring 2024

Homework 4

Due: 4/19/2024 (Friday)

Notes:

(1) For Part A, submit a "hardcopy" right after the class on the due day.
(2) TAs will run plagiarism check on your submitted programs. Write your own code and do not copy from others or anywhere.

Part A: Written Problems

1. Compute the period of the linear congruential generator $X_{n+1} = 3X_n + 2 \bmod 23$ with various initial value X_0

2. Compute the 16 output bits of the LFSR $B_2X^2 + B_0X^4$ with initial values $B_3B_2B_1B_0 =$ 1010 and 1011. What are their periods?

3. Suppose you have an entropy source that produces independent bits, where bit 1 is generated with probability 0.5+p and bit 0 is generated with probability 0.5-p, where 0<p<0.5. Consider the conditioning algorithm that examines the output 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 occurrences of each pair in the original sequence?
   b. What is the distribution of occurrences of bits 0 and 1 in the modified sequence?
   c. What is the expected number of input bits in order to generate an output bit

4. Consider the RSA encryption system. Let $n = 29 \times 43 = 1247$ and $e = 17$.

   a. What is the private key d?
   b. What is the plaintext of ciphertext C=1123?

5. Consider RSA encryption with n=136127. Assume that Alice has key pair $PU_A = (17, n)$ and $PR_A = (79663, n)$. Alice knows that Bob uses the same n to set up his key pair and $PU_B = (31, n)$. Alice intercepts a ciphertext C=3761 which is sent to Bob by Carol. Show that Alice can decrypt C without factoring n.

Part B: Programming Problem

1. This programming problem is to practice RSA encoding and decoding using Crypto++. We only deal with one-block operation without padding, that is, plaintext and ciphertext are both less than

the RSA modulus n. You need to check whether the message length (in bits) is strictly shorter modulus n's length.

2. Data format
   a. For encryption: the input is a line:

      enc 64 B14022EEF719F1BB 11 Alice

      where **enc** indicates encryption, **64** (decimal) is the modulus length, **B14022EEF719F1BB** (Hex) is the modulus n, **11** (Hex) is the encryption exponent e, and **Alice** (ASCII) is the message. Note that the message, consisting of all symbols after the 4th parameter till the end of line, may contain spaces, such as, **Alice is my friend**. The ASCII message is treated as an integer, for example, Hi (ASCII) = 4869 (Hex) = 18537 (decimal). The output is a line of the ciphertext in Hex, such as, **73DC304C7BF6A0FD** and has $\lceil 64/4 \rceil$ hex symbols, that is, adding leading zeros for small ciphertext values, such as **000A3B9F2359BBE3**.
   b. For decryption: the input is a line:

      dec 64 9D001E6473DFACF9 16282B21A7866BF5 154C638CD3615216

      which indicates decryption, modulus length (decimal), the modulus (Hex), the decryption exponent (Hex) and the ciphertext (Hex). The output is a line of the plaintext in ASCII, such as, **Secret.**
3. Submission
   a. Submit your program to the online judge system before 12:01pm, 4/19 (Friday)
   b. Your program reads in multiple lines of the above data format from stdin and outputs the results in separate lines of the specified format to stdout.
4. On-site test
   a. Computer room EC324, 5:30-9:30pm, 4/19 (Fri)
   b. Due to computer room constraint, TA's will ask you to sign up your preferred time slot in advance. There are three time slots: 17:30-19:30pm, 18:30-20:30pm and 19:30-21:30pm. You need to finish the test within two hours.
5. 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();
```

1.

t.py - /Users/origamyee/Documents/t.py (3.12.0)

```python
def period(n):
    cnt = 1
    p = -1
    show = [n]
    for i in range(24):
        X_i = 3 * show[-1] + 2
        X_i %= 23
        if X_i in show and p == -1:
            p = cnt
        else:
            show.append(X_i)
            cnt = cnt + 1
    # print(show)
    return p

num = [i for i in range(23)]
for i in range(23):
    print("when X_0 is ", num[i], "the period is ", period(num[i]))
```

```
Python 3.12.0 (v3.12.0:0fb18b02c8,
300.0.29.30)] on darwin
Type "help", "copyright", "credits"
>>>
=================== RESTART: /Users
when X_0 is  0 the period is  11
when X_0 is  1 the period is  11
when X_0 is  2 the period is  11
when X_0 is  3 the period is  11
when X_0 is  4 the period is  11
when X_0 is  5 the period is  11
when X_0 is  6 the period is  11
when X_0 is  7 the period is  11
when X_0 is  8 the period is  11
when X_0 is  9 the period is  11
when X_0 is  10 the period is  11
when X_0 is  11 the period is  11
when X_0 is  12 the period is  11
when X_0 is  13 the period is  11
when X_0 is  14 the period is  11
when X_0 is  15 the period is  11
when X_0 is  16 the period is  11
when X_0 is  17 the period is  11
when X_0 is  18 the period is  11
when X_0 is  19 the period is  11
when X_0 is  20 the period is  11
when X_0 is  21 the period is  11
when X_0 is  22 the period is  1
>>>
```

2. ① $B_3' = B_1 \oplus B_0$ $B_0' = B_1$

$B_2' = B_3$ $F = B_2 \oplus B_0$

$B_1' = B_2$

B_3	B_2	B_1	B_0	F
1	0	1	0	0
0	1	0	1	0
0	0	1	0	0
0	0	0	1	1
1	0	0	0	0
0	1	0	0	1
1	0	1	0	0

period = 6

output = 010100 010100 0101

② $B_3' = B_1 \oplus B_0$ $B_0' = B_1$

$B_2' = B_3$ $F = B_2 \oplus B_0$

$B_1' = B_2$

output = $\underline{110}\ \underline{110}\ \underline{110}\ \underline{110}\ \underline{110}\ 1$

B_3	B_2	B_1	B_0	F
1	0	1	1	1
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1

period = 3

3.(a) 00: $(0.5-p)(0.5-p)$

01: $(0.5-p)(0.5+p)$

10: $(0.5+p)(0.5-p)$

11: $(0.5+p)(0.5+p)$

(b) ① $P(\text{output}=0)$

$= P(1\text{st}=01) + P(2\text{st}=01 \wedge (1\text{st}:00 \vee 11))$

$\cdots P(i\text{th}=01 \wedge (j\text{th}=00 \vee 11\ \forall j<i))\cdots$

$= (0.5-p)(0.5+p) + (0.5-p)(0.5+p) \times$

$[\underbrace{(0.5-p)(0.5-p)+(0.5+p)(0.5+p)}_{2\times(0.5^2+p^2)=0.5+2p^2}] + \cdots$

$= \dfrac{0.25-p^2}{1-(0.5+2p^2)} = \dfrac{0.25-p^2}{0.5-2p^2} = 1/2$

② $P(\text{output}=1)$

$= P(1st=10) + P(2st=10 \wedge (1st: 00 \vee 11))$

$\cdots P(i\text{th}=10 \wedge (j\text{th}=00 \vee 11 \ \forall j<i)) \cdots$

$= (0.5+p)(0.5-p) + (0.5+p)(0.5-p) \times$

$[\underbrace{(0.5-p)(0.5-p) + (0.5+p)(0.5+p)}_{2\times(0.5^2+p^2)=0.5+2p^2}] + \cdots$

$$= \frac{0.25-p^2}{1-(0.5+2p^2)} = \frac{0.25-p^2}{0.5-2p^2} = 1/2$$

3.(c) $P(\text{orignal sequence} = 01 \vee 10)$

$= (0.5-p)(0.5+p) + (0.5+p)(0.5-p)$

$= 0.5 - 2p^2$

$E[\#\text{ of input}] = \sum_{k=1}^{\infty} (1-(0.5-2p^2))^{k-1}(0.5-2p^2)(2k)$

几何分布 $= \dfrac{2}{0.5-2p^2}$

4.(a) $d = e^{-1} \bmod \phi(n)$

① Calculate $\phi(n)$

$n = 29 \times 43 = 1247$

$\phi(n) = 1247 \times (1-\frac{1}{29})(1-\frac{1}{43})$

$= 28 \times 42$

$= 1176$

② Extend gcd.

Find (x, y) s.t. $ex + \phi(n)y = 1$

$\Rightarrow ex \equiv 1 \mod \phi(n) \Rightarrow e^{-1} = x = d$

$17x + 1176y = 1$

i	r_i	q_i	x_i	y_i
-1	17	X	1	0
0	1176		0	1
1	17	0	1	0
2	3	69	-69	1
3	2	5	346	-5
4	1	1	-415	6
5	0	2		

$\Rightarrow 17 \times (-415) + 1176 \times 6 = 1$

$\Rightarrow d = e^{-1} = -415 + 1176 = 761$

4.(b) $M = C^d \mod n$

$= 1123^{761} \mod 1247$

$= 1104$

IDLE Shell 3.12.0

Python 3.12.0 (v3.12.0:0fb18b02c8, Oct 2 2023, 09:45:56) [Clang 13.0.0 (clang-1
300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> print(pow(1123, ~~425~~, 1247))
761
1104
>>>

5. 假設 plainText = M

① 計算 $k\phi(n) = e_{Alice}d_{Alice} - 1$ (by CRT)

② 計算 $e_{Bob}^{-1} \bmod k\phi(n)$，記為 d'_{Bob} $\Rightarrow d'_{Bob}e_{Bob} = k'k\phi(n)+1$

③ $d'_{Bob} \equiv d_{Bob} \bmod \phi(n)$ $\because e_{Bob}d'_{Bob} \bmod \phi(n)$

$= e_{Bob}d_{Bob} \bmod \phi(n)$

$= 1$

④ $C = M^{e_{Bob}} \bmod n$, $C^{d'_{Bob}} \bmod n = M$

$\because C^{d'_{Bob}} \equiv M^{e_{Bob}d'_{Bob}}$

$\equiv M^{k'k\phi(n)+1} \bmod n$

$\equiv M$

① $k\phi(n) = 17 \times 79553 - 1 = 1352400$

② $d'_{Bob} \times 31 = 1352400k' + 1$

i	r_i	q_i	x_i	y_i
-1	1352400		1	0
0	31		0	1
1	25	43625	1	-43625
2	6	1	-1	43626
3	1	4	5	218129
4	0			

$\Rightarrow \underbrace{218129}_{-d'_{Bob}} \times 31 = 1354771 \times \underbrace{5}_{k'} + 1$

$\Rightarrow d'_{Bob} = -218129 + 1352400 = 1134271$

④ $M = C^{d'_{Bob}} \bmod n$

$= 3761^{1134271} \bmod 136127$

$= 37145$

□