Cryptography

18/2/1404 – 90 minutes

SURNAME, NAME: ________________________ _______________

MATRICOLA: _____________________________

PLEASE NOTE: all the steps performed to obtain the required results must be specified in detail and adequately justified.

Answers without justifications are not going to be evaluated.

EXERCISE 1

Compute the bits s_5, s_4, s_3, s_2 of the stream generated by the LFSR whose polynomial is

$$\chi_{\mathrm{L}}(x) = x^2 + x + 1$$

and the first bits are $s_0 = 1, s_1 = 0$.

Fibonacci LFSR

$\chi_L = p_0 + p_1 x + \dots + p_{m-1} x^{m-1} + x^m \Rightarrow p = [1\,1\,1]$

Nota: The figure of the LFSR in appendix may be useful.

The transition state is given by the multiplication $s \cdot L$

Where $s = [s_{m-1}\, s_{m-2} \dots s_1 s_0] = (0\ 1)$

and $L = \begin{pmatrix} p_{m-1} & 1 & \dots & 0 \\ p_{m-2} & 0 & 1 & \dots \\ \vdots & & \ddots & 0 \\ p_1 & & \ddots & 1 \\ p_0 & & & \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$

$(0\ 1)\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \big((0\wedge 1)\oplus(1\wedge 1), (0\wedge 1)\oplus(1\wedge 0)\big) = (0\oplus 1, 0\oplus 0) = (\overset{s_2}{1}\ \overset{s_1}{0})$

$(1\ 0)\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = (\overset{s_3}{1}\ \overset{s_2}{1})$

$(1\ 1)\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = (\overset{s_4}{0}\ \overset{s_3}{1})$

$(0\ 1)\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = (\overset{s_5}{1}\ 0)$

$s_2 = 1$

$s_3 = 1$

$s_4 = 0$

$s_5 = 1$

EXERCISE 2

Let $\mathbf{E} : y^2 = x^3 + 7$ be the elliptic curve defined on $\mathbb{Z}_{11}$.

(a) Check that $P = (2, 2)$ and $Q = (7, 3)$ are in $\mathbf{E}$,

(b) compute the addition $P{+}Q$ on the elliptic curve $\mathbf{E}$,

(c) check that your answer of (b) is a point of $\mathbf{E}$.

a) $P = (2,2) = (x,y)$

$y^2 \equiv x^3 + 7 \bmod 11$

$4 \equiv \underbrace{8+7}_{} \bmod 11$

$15 \equiv 4 \bmod 11$ ✓

$Q = (7,3) = (x,y)$

$9 \equiv 7^3 + 7 \bmod 11$

$9 \equiv 7 \cdot 7^2 + 7 \bmod 11$

$9 \equiv 7 \cdot 5 + 7 \bmod 11$

$9 \equiv 2 + 7$ ✓

b) $P+Q$; $P \neq Q$; $(2,2)+(7,3)$

$(x_1,y_1)+(x_2,y_2) = (x_3,y_3)$ where

$x_3 = \lambda^2 - x_1 - x_2$

$y_3 = -(\lambda x_3 + \nu)$

$\lambda = \dfrac{y_2 - y_1}{x_2 - x_1}$; $\nu = y_1 - \lambda x_1$

$\lambda = 1 \cdot 5^{-1} \bmod 11 \equiv 9 \bmod 11$

$\nu = 2 - 9 \times 2 = 2 - 7 = 6$

$x_3 = 9^2 - 2 - 7 = 4 - 2 - 7 = 6$

$y_3 = -(\lambda x_3 + \nu) = -(9 \times 6 + 6) = -(10+6) \equiv 6$

$\Rightarrow P+Q = (x_3, y_3) = (6,6)$

c) $6^2 \equiv 6^3 + 7 \bmod 11$

$3 \equiv 3 \times 6 + 7 \bmod 11$

$3 \equiv 7 + 7 \bmod 11 \equiv 3$ ✓

solution: $(11, 18)$

EXERCISE 3

In RSA:
user A keys are $\mathsf{sk}_A = (p_A, q_A, d_A) = (5, 11, 23)$ and $\mathsf{pk}_A = (n_A, e_A) = (55, 7)$,
user B keys are $\mathsf{sk}_B = (p_B, q_B, d_B) = (3, 7, 5)$ and $\mathsf{pk}_B = (n_B, e_B) = (21, 5)$.

A wants to send to B the message M. So she send the ciphertext $C = \mathsf{Enc}_{\mathsf{pk}_B}(M)$ and add as her digital signature the pair $(\mathsf{Enc}_{\mathsf{pk}_B}(F), h(M))$, where $F = \mathsf{Enc}_{\mathsf{sk}_A}(h(M))$ and $h(x)$ is a fixed hash function.

Assuming $h(M) = 18$ find the digital signature i.e. the pair $(\mathsf{Enc}_{\mathsf{pk}_B}(F), h(M))$.

A : prime numbers $p_A = 5$; $q_A = 11$

$n_A = 5 \times 11 = 55$

$\Phi_A(n) = (p_A - 1)(q_A - 1) = 40$

public key pk_A is a random integer $\in \{1, \dots \Phi_A(n) - 1\}$

$pk_A = 7$

private key sk such that $pk \cdot sk \equiv 1 \bmod \Phi(n) \Rightarrow 7 \cdot sk \equiv 1 \bmod 40$

$sk \equiv 7^{-1} \bmod 40$

$sk \equiv 23 \bmod 40$

$p_B = 3$, $q_B = 7 \Rightarrow n_B = 21$

$pk_B = 5$

$\Phi(n) = 12$

$5^{-1} \bmod 12 \Rightarrow sk \equiv 5$

Encryption: $C = Enc\_pk_B(M) = M^e \bmod n_B$

$Enc\_sk_A(h(M)) = [h(M)]^{d_A} \bmod n_A = 18^{23} \bmod 55 = 2$

$F = 32$

$Enc\_pk_B(F) = 2^5 \bmod 21 = 11$

$18^{23} \bmod 55$

$55 = 11 \times 5$

$18^{23} = 7^{23} \bmod 11$
$18^{23} = 3^{23} \bmod 5$

$7^{23} = 7^3 \cdot 7^{10} \cdot 7^{10} = \underbrace{7 \times 7^2}_{7 \times 5 = 2} \times 7^{10} \times 7^{10} = 2 \times 7^5 \times 7^5 \times 7^5 \times 7^5 = 2 \times 10 \times 10 \times 10 \times 10$

$= 9 \times 10 \times 10 \times 10$
$= 2 \times 10 \times 10$
$= 9 \times 10 = 2$

$7^5 = 7^3 \times 7^2 = 2 \times 7^2 = 2 \times$

$\boxed{7^{23} \bmod 11 = 2}$

$3^{23} \bmod 5 = 3^3 \cdot 3^{10} \cdot 3^{10} = 4 \times 3 \cdot 3^{10} \cdot 3^{10} = 2 \cdot 3^{10} \cdot 3^{10} = 2 \times 4 \times 4 = 2$

$3^{10} = 3^5 \cdot 3^5$

$3^5 = 3^3 \cdot 3^2 = 2 \cdot 4 = 3$

$3^{10} = 3 \times 3 = 4$

$\boxed{3^{23} \bmod 5 = 2}$

$\begin{cases} x = 2 \bmod 11 \\ x = 2 \bmod 5 \end{cases}$

$N = 55$
$N_1 = 5 \qquad b_1 = 2$
$N_2 = 11 \qquad b_2 = 2$

$x_1 = 5^{-1} \bmod 11$

$11 = 5 \times 2 + 1$
$5 = 1 \times 5 + 0$

$p_0 = 0$
$p_1 = 1$
$p_2 = 0 - 2 \bmod 11 = 9$

$x_2 = 11^{-1} \bmod 5 = 1^{-1} \bmod 5 = 1$

b_i	N_i	x_i	$b_i N_i x_i$
2	5	9	90
2	11	1	22

$90 + 22 = 112$

$112 \equiv 2 \bmod 55$

$\chi_L(x) = x^2 + x + 1$ Compute s_5, s_4, s_3, s_2

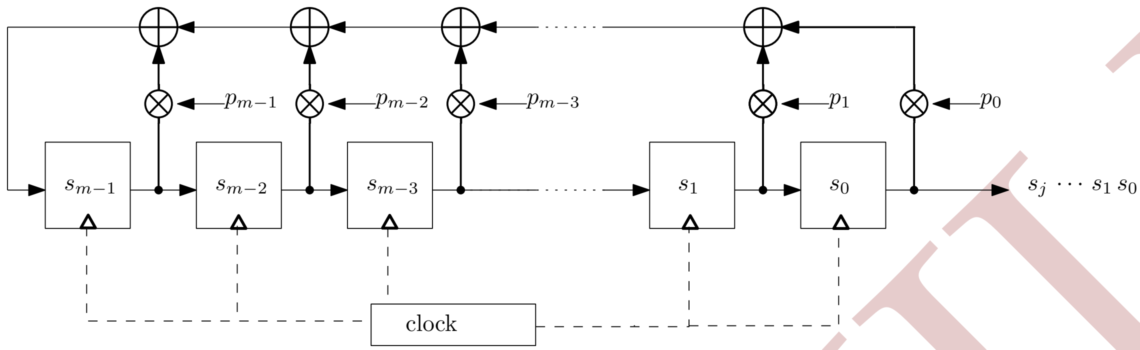
$s_0 = 1$
$s_1 = 0$

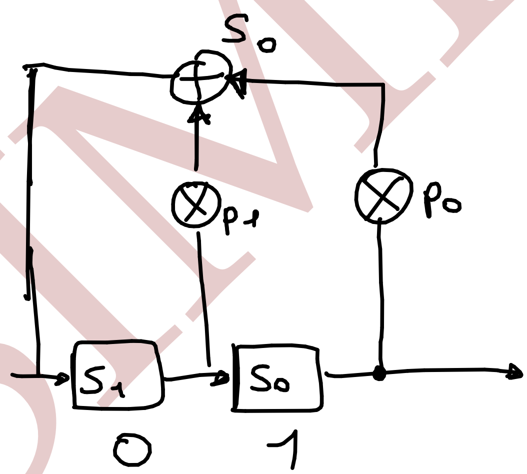
$s_m = \sum_{j=0}^{m-1} s_j p_j$ and $\chi_L(x) = p_0 + p_1 x + \ldots + p_{m-1} x^{m-1} + x^m$

APPENDIX

$p = [000111]$

Fibonacci LFSR





$s_2 = 1$
$s_3 = 1$
$s_4 = 0$
$s_5 = 1$

Exercise 4

Complete the program by adding the missing C code: the program has to send to standard output the formatted SHA256 digest of the content of file whose name is passed as first argument, computed using the OpenSSL library.

```
#include <stdio.h>
/******************************
 * write missing libraries here *
 ******************************/



#define BUFF_SIZE 256

void handleErrors(void){
        ERR_print_errors_fp(stderr);
        abort();
}

int main(int argc, char* argv[]) {
   FILE* f_in;
   unsigned char buff[BUFF_SIZE];
   int i = 0;
   unsigned char digest[                ]; /*use the proper constant */
   SHA256_CTX context;

   f_in = fopen(argv[1], "r");
   if(f_in==NULL)
      handleErrors();

/*****************************************
* write the code to compute the digest with *
* SHA256 of the content of f_in             *
*****************************************/
```

```
    fclose(f_in);

    printf("\n Hash of the file: %s \n\n", argv[1]);
    for(i = 0; i < SHA256_DIGEST_LEN; i++) {
        printf("%.2x", digest[i]);
        if(i+1 != SHA256_DIGEST_LEN)
          printf(":");
    }
    printf("\n");
    return 0;
}
```

Exercise 5

An encryption oracle receives as input a string (named input) and outputs another string, which is the hexadecimal encoding of the result of the encryption with AES256 in ECB mode of the plaintext obtained with the following Python instruction:

```
message = """Here is the msg:{0}- the secret is:{1}""".format( input, secret)
```

where:

- input is the string received as input and
- secret is a secret string, composed of 16 printable characters.

More precisely, the oracle performs a never ending loop as follows:

```
# the encryption key, not to be discovered
from oraclesecrets import key
# the secret to discover
from oraclesecrets import secret

HOST = ''
PORT = 12342

def pad(message):
        if len(message) % 16 != 0:
                message = message + '0'*(16 - len(message)%16 )
        return message

#... socket s : code to create, bind, listening on HOST:PORT...

while 1:
        conn, addr = s.accept()

        input = conn.recv(1024).decode()

        message = """Here is the msg:{0} - and the key:{1}""".format( input, secret)
        message = pad(message)

        cipher = AES.new( key.decode('hex'), AES.MODE_ECB )
        encoded_ciphertext = (cipher.encrypt(message).encode('hex')).encode('hex')

        conn.send(encoded_ciphertext)
        conn.close()
```

Complete the program so that the value of secret is discovered and printed on the standard output (without bruteforcing the whole search space). Note that the objective of the exercise is not to recover the encryption key, which is only known by the server.

```
from pwn import *
import string

ADDRESS = #we do not care the actual address
PORT = 12342
SECRET_LEN = 16

##################################################
secret = ""

Attack : padding oracle ECB

##################################################

print "secret="+secret
```