Assignment 2: Exercises for Topic 2

Important Note. Due date: **June 8, 2020 (11:59am)**. The requested submission consists of the questions with * in the assignments, i.e., you need to submit Exercises 1, 2, and 7, and 7-(d) is a bonus question for 1 mark.

Exercise 1*. LFSR generators.

(a) There 6 LFSRs of degree 5 which generate m-sequences with period 31. We associate a feedback function $f(x_0, \cdots, x_{n-1}) = c_0x_0 + \cdots + c_{n-1}x_{n-1}$ with a polynomial $f(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_1x + c_0, c_i \in \{0, 1\}$. These 6 primitive polynomials are given as follows.

$f(x)$
$x^5 + x^3 + x^2 + x + 1$
$x^5 + x^4 + x^3 + x + 1$
$x^5 + x^2 + 1$
$x^5 + x^4 + x^2 + x + 1$
$x^5 + x^3 + 1$
$x^5 + x^4 + x^3 + x^2 + 1$

You may pick one of these which is not the one in (b), and generate an m-sequence of period 31.

(b) Let

$$\mathbf{a} = \{a_i\} = 00001011010100011101111110010011$$

generated by the primitive polynomial $f(x) = x^5 + x^3 + x^2 + x + 1$. Thus $\mathbf{a}$ is an m-sequence of period 31. Calculate the 0-1 distribution, the run distribution of the sequence, and autocorrelation. What is the initial state of the LFSR which generates this sequence? Sketch the LFSR.

(c) For the LFSR which generates the m-sequence given in (b), if we want to generate a pseudorandom number of 5 bits, how many different numbers the LFSR could generate?

Exercise 2*. Correlation attack. Let a combinatorial generator be given as follows (note that this design is similar as the example given on slide 6 Lecture 9).

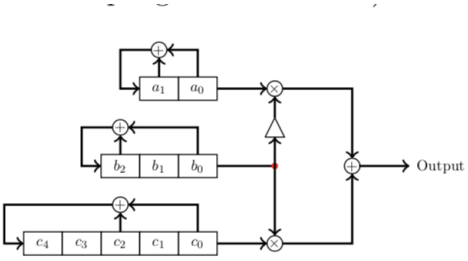


Figure 1: A combinatorial generator

Suppose it is used as a key stream generator and a 10-bit key is loaded as initial states of three LFSRs. An attacker obtains the 40 consecutive bits of the outputs of this generator:

$$\mathbf{s}^{40} = (1001110111110100101001001001111100001101).$$

(a) From the Lecture 9, the output is correlated with the first LFSR and the third LFSR. Let the initial states for the first and third LFSRs be $(a_0, a_1) = (1, 1)$ and $(c_0, c_1, c_2, c_3, c_4) = (1, 1, 1, 1, 1)$. What are their respective outputs of these two LFSRs?

(b) Find the 10-bit key, i.e., the initial state of each LFSR used to generate the key stream bits using the correlation attack.

(c) Compare the complexity of (c) with the exhaustive search.

(Hint. You should write a small program to compute the correlation.)

Exercise 3. Cipher design.

(a) Explain why IV is needed for both stream cipher and block cipher.

(b) Use a credit card with cryptographic protections for the bank transaction as an example to explain some consequence of confidentiality or integrity and authenticity of a transaction is lost when there is no IV and they key is repeated in use for multiple transactions.

Exercise 4. The core nonlinear part of the permutations in AES is the S-box, which is the 8-bit inverse function. Let a simplified AES 4-bit S-box, denoted by $E(x)$ given as follows.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$E(x)$	0	1	9	14	13	11	7	6	15	2	12	5	10	4	3	8

Let $k = 0011$ be the symmetric key shared by two communication entities, say Alice and Bob. (We assume that the right most bit is the least significant bit.) The encryption is defined as follows:

$$c = E(k + m)$$

where the addition is bitwise addition. For example, if the message $m = 0101$, since $k + m = 0011 + 0101 = 0110$, look at the table, for input $x = 0110 = 2 + 2^2 = 6$, then the cipher text is given by

$$c = E(k + m) = E(6) = 7 = 0111.$$

Find the cipher text for the plaintext $m = 110001010$ using CBC encryption.

Exercise 5. SHA

(a) What is the size of the output of SHA1, and each case of SHA2?

(b) What are SHA-3's parameters? What are their respective collision probabilities by a random guess for SHA1, SHA2 and SHA3?

(c) List SHA1, SHA2 and SHA3's respective security levels according to the time-memory trade-off attack.

Exercise 6. Security of public-key systems.

(a) What are their definitions of a one-way function and one-way trapdoor function, respectively? Provide an example to explain those concepts.

(b) Explain how the security of public-key cryptosystems are evaluated.

Exercise 7*. DH protocol under authentic channel. In a small fantasy world computing the discrete logarithm in $GF(p)$ is computational infeasible even for p being an n-bit prime number when $n \approx 6$. So, we could choose the domain parameters $(p, g) = (47, 5)$.

(a) Verify that $p = 47$ is a 6-bit prime number, and $g = 5$ is a primitive element in $GF(p)$.

(b) For three users in the system, say, Alice, Bob and Carol, they use Gen to generate their private keys

$$x_A = 3, x_B = 11, x_C = 7.$$

Compute their respective public-keys for each of them.

(c) Compute shared key for each pair of users using the results in (b).

(d) ** Optional. If you do this correctly, then you will receive one bonus mark.

(1) If one of three users, say user Alice, has got an attack and the attacker retrieved its private-key. Explain how the attacker can impersonate the other users.

(2) Identify a scenario in the real world where the above attack could happen and determine the harmful consequence of the attack.

Exercise 8. Bob will use RSA signature scheme to sign messages. Let $p = 11$, $q = 23$, $n = pq$, hash function $h(x) = 2x \bmod n$.

(a) Generate Bob's private and public key pair.

(b) Bob will sign the message $m = 2$, generate a digital signature for the message $m = 2$. Generate Bob's signature over $m = 2$.

(c) Show the verification process of Bob's signature.

(d) Explain why the timing side-channel attack can be used to recover the secret exponent d when Bob is generating a signature or decrypting a ciphertext.

Hint. We omit the certificate step here. Since $n = 11 \times 23 = 253$, $\phi(n) = (p-1)(q-1) = 10 \times 22$=220. For your convenience, here we list the first ten numbers which are coprime with 220, and their inverses modulo 220.

e	3	7	9	13	17	19	21	23	27	29
$d = e^{-1} \bmod 220$	147	63	49	17	13	139	21	67	163	129

Exercise 9 . Solve the following problems for DSS.

(a) What happens if the random number k used in creating DSS signature is compromised?

(b) DSS specifies that if the signature-generation process results in value of $s = 0$, a new random number k should be generated and the signature should be recomputed? Why?

(c) What happens if the hash function is not secure in DSS (this means that for $a = h(m)$, one can easily find another m' such that $a = h(m')$?

1. (b)

I) 0-1 distribution: Number of zeroes: 15 } difference at most 1
Number of ones: 16 }

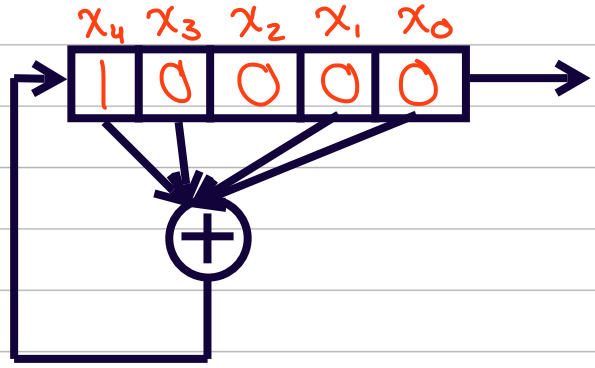
II) Run distribution:

Length	0-run	1-run
1	$2^{5-(1)-2} = 4$	$2^{5-(1)-2} = 4$
2	$2^{5-(2)-2} = 2$	$2^{5-(2)-2} = 2$
$n-2 = 3$	1	1
$n-1 = 4$	1	0
$n = 5$	0	1
Total	$2^{5-2} = 8$	$2^{5-2} = 8$

III) Autocorrelation: $C(\tau=0) = 31 = N$
$C(\tau \neq 0) = K \neq N$

IV) Initial States: first n bits of m-sequence in reverse order
=> x_4 x_3 x_2 x_1 x_0
1 0 0 0 0

V) LFSR Sketch:



(c) An n-bit LFSR should be able to produce at most $2^n - 1$ n-bit pseudorandom numbers. Given $n=5$, this LFSR is capable of producing at most 31 5-bit pseudorandom numbers.

2.(c) correlation attack: $(2^2-1)+(2^5-1)+(2^3-1) = 41$

brute-force attack: $(2^2-1)(2^5-1)(2^3-1) = 651$

7.(a) $47_{10} = \underbrace{101111_2}_{6 \text{ bits}}$: factors = (1, 47)

$GF(47) = \{0, 1, 2, \ldots, 46\}$

g is a primitive element of GF(p) if
- $g^{p-1} = 1$
- $g^r \neq 1 \ \forall \ 1 \leq r < p-1$

① $g^{47-1} = 5^{46} \pmod{47} = 1$

② only need to check $g^r \neq 1$ for $r = 2, 23$ since $g^{46} = g^{2 \times 23}$

$\Rightarrow 5^2 \pmod{47} = 25 \neq 1$

$5^{23} \pmod{47} = 46 \neq 1$

∴ g=5 is a primitive element in GF(p=47)

(b) $pk_A = g^{x_A} = 5^3 \pmod{47} = 31$

$pk_B = g^{x_B} = 5^{11} \pmod{47} = 13$

$pk_C = g^{x_C} = 5^7 \pmod{47} = 11$

(c) AB: $g^{x_A x_B} = 5^{33} \pmod{47} = 35$

AC: $g^{x_A x_C} = 5^{21} \pmod{47} = 15$

BC: $g^{x_B x_C} = 5^{77} \pmod{47} = 29$