Class20/24

Lecture Notes of EE720

20 August 2020

1 Stream Ciphers and applications

In the previous lecture we saw the idea of Vernam pad as encryption of a plaintext stream using a key stream. When the key stream is randomly chosen for each message then it is called One Time Pad (OTP). Shannon showed that OTP was perfectly secure in the sense that whatever information was a priori known about plaintext remains same even after getting the ciphertext stream. Hence this in a sense satisfies the third condition of a TOWF $y = F(k, x)$, knowing x and, y computation of k is difficult. Here difficult means each alphabet of the key stream is uniformly random.

OTP is very difficult to use practically because

1. Key stream as long as plaintext has to be exchanged securely a priori.
2. Every plaintext requires freshly generated random keysteam.

But if we can relax the security from the ideal notion of Perfect Secrecy then practical options emerge. This led to the idea of stream cipher.

1.1 Stream cipher

Stream cipher was described in the previous class as a finite state machine defined by a map $F : X \to X$ on the set of states and output stream generated by a map $f : X \to A$ on the set of alphabets. The initial state is $x(0) = (K, IV)$ where K is a key secretly exchanged and IV is the initializing vector sent along with the ciphertext. The operation of the cipher to encrypt a plaintext stream $P = \{p(i)\}$ is as follows: Assume K is exchanged by Alice and Bob.

1.1.1 Algorithm

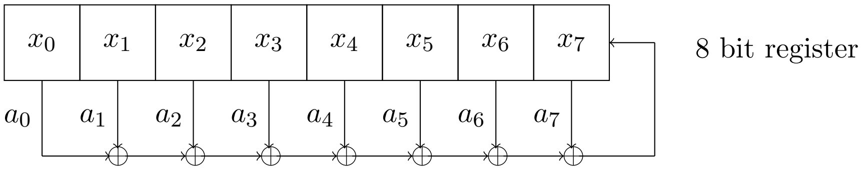
Stream cipher operation.

1. Alice chooses random IV and a time index $k_0 >> 0$.
2. Alice computes the state $x(k_0)$ by iterating the map $x(k+1) = F(x(k))$ starting from $x(0)$ for $k = 0, 1, 2, \ldots, k_0$.
3. Alice computes the output stream $w(j) = f(x(k_0+j))$ for $j = 0, 1, 2, ldots$ as long as the plaintext stream.
4. Alice sends ciphertext stream
$$\{c(j)\} = \{p(j) \oplus w(j)\}$$
appended by IV to Bob.
5. Bob on receiving $\{c(j)\}$ reconstructs $w(j)$ using IV and decrepts as
$$\{p(j)\} = \{c(j) \ominus w(j)\}$$

Homework 1 write under what conditions on F, f, k_0 and IV does the stream cipher give a TOWF.

1.1.2 Feedback Shift Registers

One common way to create stream ciphers is using the concept of Feedback Shift Register. Let $A = \mathbb{F}_2 = \{0, 1\}$ the binary field.



$$x_8 = a_0x_0 + a_1x_1 + \ldots + a_7x_7$$

Such a shift register is called Linear Feedback Shift Register (LFSR). The mechanism is denoted as

$$L[a_0 \quad a_1 \quad a_2, \ldots, a_{n-1}]$$

for a LFSR of length n. The output of the LFSR is the leftmost bit. Consider example of an LFSR of length 6.

x_0	x_1	x_2	x_3	x_4	x_5

The recurrence rule is

$$x_6 = x_1 \oplus x_2 \oplus x_4$$

Consider an Initial Condition (IC) of register $(1, 1, 0, 0, 1, 0)$. The output sequence generated for this IC is

$$1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0..........$$

this repeates after the last bit.

1.1.3 Linear Complexity of a sequence

For sequence $\{x_k\}$, $k = 0, 1, 2, \ldots$ if r is the smallest index such that there exists a recurrence given by:

$$x_{r+i} = \sum_{j=0}^{r-1} a_j x_{r+j}$$

$i = 0, 1, 2, \ldots$, then r is called the *rank* (or *linear complexity*) of the sequence $\{x_k\}$.

Homework 2 Show that the linear complexity is the smallest r such that the Hankel matrix of the sequence at r satisfies

$$\text{rank} H_r = \text{rank} H_{r+1}$$

1.2 Shift Register state mapping

State of the register at time k is the content of the register at k. For example: Initial state is the IC $x(0)$. Consider example

0	1	1	0	1

x(0) = (0 1 1 0 1)

If the recurrence rule is

$$x_5 = x_0 \oplus x_2 \oplus x_4$$

then,

x(1)	1	1	0	1	1
x(2)	1	0	1	1	0
x(3)	0	1	1	0	0

Let the recurrence rule be

$$x_n = \sum_{i=0}^{n-1} a_i x_i \triangleq f(x_0, \ldots, x_{n-1})$$

Then the state update map $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is defined by

$$F : (x_0 \quad x_1, \ldots, x_{n-1}) \longrightarrow (x_1 \quad x_2, \ldots, x_{n-1} \quad f(x_0, x_1, \ldots, x_{n-1}))$$

1.3 Non-linear Shift Register (FSR)

In the state map above if $f(x_0 \quad x_1 \quad ... \quad x_n)$ is not a linear function like in a LFSR then the shift register is called FSR.

Example 1 FSR is given by the map

$$F : (x_0, x_1, x_2, x_3) \to (x_1, x_2, x_3, x_1 x_2 \oplus x_3)$$

With IC: $x(0) = (1 \quad 0 \quad 1 \quad 0)$

$$\begin{array}{rcl} x(1) & = & (0 \quad 1 \quad 0 \quad 0) \\ x(2) & = & (0 \quad 0 \quad 0 \quad 0) \\ x(3) & = & (0, 0, 0, 0) \end{array}$$

$x(3)$ repeats indefinitely. With IC: $x(0) = (0 \quad 1 \quad 1 \quad 1)$

$$\begin{array}{rcl} x(1) & = & (1 \quad 1 \quad 1 \quad 0) \\ x(2) & = & (1 \quad 1 \quad 0 \quad 1) \\ x(3) & = & (1 \quad 0 \quad 1 \quad 1) \\ x(4) & = & (0 \quad 1 \quad 1 \quad 1) \\ x(5) & = & (0 \quad 1 \quad 1 \quad 1) = x(0) \end{array}$$

Which generates the output sequence:

$$0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1$$

For FSR of length n, state set is $X = \mathbb{F}^n$ and the number of all states is $|X| = 2^n$. The state map F is a map of the set of all states X to itself

$$\mathrm{F : X \to X}$$

Question 1 When is F a permutation of X?

$$F \text{ is a permutation } \Leftrightarrow F \text{ is } 1-1$$

$$(x_0 \quad x_1, \ldots, x_{n-1}) \quad \xrightarrow{F} (x_0 \quad x_1, \ldots, f(x_0, \ldots, x_{n-1}))$$

Hence F is $1-1$ if and only if

$$\begin{aligned} & F(x_0, x_1, \ldots, x_{n-1}) && \neq && F(\tilde{x}_0, x_1, \ldots, x_{n-1}) \text{ when } x_0 \neq \tilde{x}_0 \\ \Leftrightarrow \; & f(x_0, x_1, \ldots, x_{n-1}) && \neq && f(\tilde{x}_0, x_1, \ldots, x_{n-1}) \text{ when } x_0 \neq \tilde{x_0} \end{aligned}$$

Since, f is binary valued function and x_i are binary $\forall i \in \{0, 1, \ldots, n-1\}$. On expanding the function f we get

$$\begin{aligned} f(x_0, \ldots, x_{n-1}) \quad = & \\ & c_0 \oplus x_0 f_0(x_1, x_2 \ldots, x_{n-1}) \oplus x_1 f_1(x_0, x_2, \ldots, x_{n-1}), \ldots, \\ & x_{n-1} f_{n-1}(x_0, \ldots, x_{n-2}) \end{aligned}$$

Note that in the functions f_i above (multiplied by x_i) the variable x_i is absent. The above condition for permutation is

$$f(x_0, \ldots, x_{n-1}) = f(\tilde{x_0}, \ldots, x_{n-1}) \oplus 1 \text{ for } \tilde{x_0} = x_0 \oplus 1$$

Show that F is a permutation iff

$$f(x_0, \ldots, x_{n-1}) = x_0 \oplus h(x_1, \ldots, x_{n-1})$$

Note that this is a special form of the function f which has a linear part x_0 and a term defined by a function h which does not have x_0 in its arguments.

Question 2 Which of the following maps are permutations?

$$\begin{aligned} F(x_0 \quad x_1 \quad x_2 \quad x_3) &= (x_1 \quad x_2 \quad x_3 \quad x_0 \oplus x_1 x_2 x_3) \\ F(x_0 \quad x_1 \quad x_2) &= (x_1 \quad x_2 \quad x_1 x_2 \oplus x_0 x_2) \\ F(x_0 \quad x_1 \quad x_2 \quad x_3) &= (x_1 \quad x_2 \quad x_3 \quad x_0 \oplus x_0 x_1 \oplus x_1 x_2) \\ F(x_0 \quad x_1 \quad x_2 \quad x_3) &= (x_1 \quad x_2 \quad x_3 \quad x_0 \oplus x_1 x_2 \oplus x_2 x_3) \end{aligned}$$

1.4 Orbits of a permutation map

A permutation map creates purely closed orbits. Consider

$$F(x_0, x_1, x_2) = (x_1, x_2, x_0 \oplus x_1 \oplus x_1 x_2)$$

From any IC say $x(0) = (1, 0, 1)$

$$(1,0,1) \to (0,1,1) \to (1,1,0) \to (1,0,0) \to (0,0,1) \to (0,1,0)$$

The next state is $(1, 0, 1)$ which is IC. It is an orbit of 6 states out of $2^3 = 8$ total points. The state not in the above orbit is $(1 \quad 1 \quad 1)$. Its otbit is $(1, 1, 1) \to (1, 1, 1)$. It forms an orbit of 1 point, i.e. a fixed point. Similarly, it can be verified that $(0, 0, 0)$ is also a fixed point.

Homework 3 Find all orbit lengths and orbits of permutations in above maps.