3 Stream ciphers

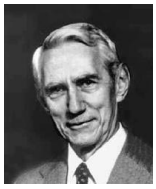
3.1 Syncroneous stream ciphers

- **One Time Pad - salauksen idea jonosalauksen taustalla**

According to Shannon's theory an *unconditionally secure cipher* is a cipher, where the number of possible keys equals to the number of possible messages. This is true for One Time Pad , in which the key is a completely random bit sequence which has the length of the message and the encryption function is bitwise XOR.

As mentioned it is intuitively clear that One Time Pad is perfectly safe, because complete search of key space gives all possible meaningful and meaningless messages of that size.

Claude E. Shannon (1916 - 2001) was an American engineer and mathematician, who got Nobel prize 1940. He is known as the father of mathematical communication theory and information theory



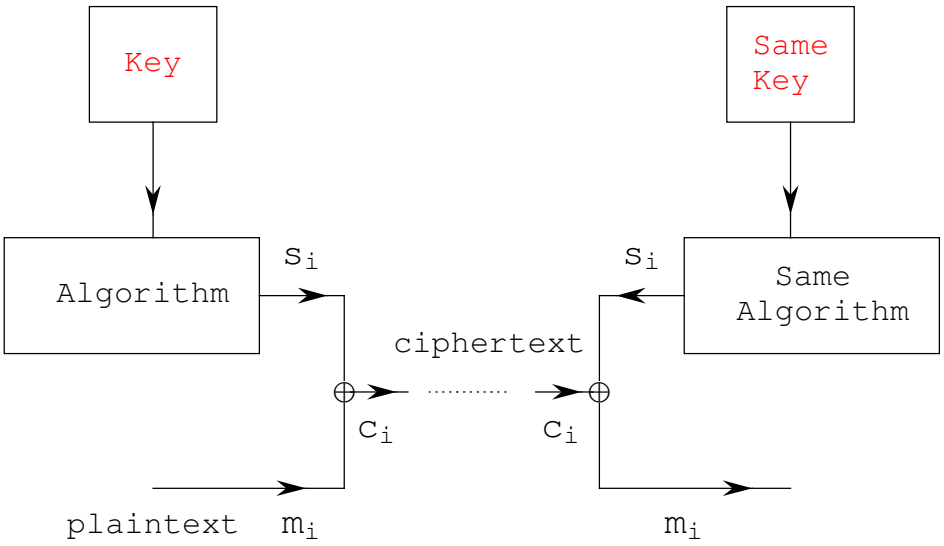
- **Syncroneous stream ciphers**

A syncroneous stream cipher imitates One Time Pad. A problem lies in delivering the random key to the other party. This could be done in hot line between Moscow and Washington, but not in the mass telecommunication.

Random sequences can be produced for example by throwing a coin, but in a computer it is more difficult.

The requirement of complete randomness must be replaced with a requirement of pseudorandomness A pseudorandom bit sequence looks random and fulfills certain criteria, but is actually deterministic.
Both parties generate exactly same key bit sequence, if the initial states are the same. The initial state acts as a symmetric encryption key..

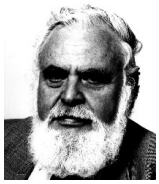
In the picture below is the principle of a syncroneous key cipher. Encryption is bitwise XOR of the bits of the message and the pseudorandom bits . The receiver has an identical generator and the symmetric key produces an identical key stream. The decryption algorithm is identical with the encryption algorithm.



■ 3.2 Pseudorandom number generation

Psudorandomness is defined by professor Solomon W. Golomb.

Solomon W. Golomb (1932 -) is an engineer and mathematician, emeritus professor of Southern California University , researcher of combinatorics, number theory, coding theory and communication theory. His achievements are development of the concept of pseurodandomness and the theory of LFSR registers.



■ Golomb's postulates for pseudorandomness

Definition

A sequence $\{s_i\}_{i\geq 0}$ is called **periodic with period *p*,** if p is the smallest integer with

$$s_{i+p} = s_i \qquad \text{for all } i \geq 0.$$

Definition:

A *run of length k* is a subsequence of $\{s_i\}_{i\geq 0}$ with k identical symbols bordered by other symbols. There are two subtypes:

block of length k $\quad 0\,\overbrace{1\,1\ldots1}^{k}\,0$

gap of length k $\quad 1\,\overbrace{0\,0\ldots0}^{k}\,1$

Definition : Out of phase autocorrelation for *a* sequence with period p, is defined as

$$\mathrm{AC}(k) = \frac{A(k) - D(k)}{p}$$

where $\quad A(k) = |\{\, 0 \le i < p \mid s_i = s_{i+k} \,\}|$

and $\quad D(k) = |\{\, 0 \le i < p \mid s_i \neq s_{i+k} \,\}|$

$A(k)$ and $D(k)$ are the numbers of coincidences during a whole period in sequences obtained from the original sequences $\{s_i\}_{i\ge 0}$ by rotating the sequence k units left

Golomb's randomness postulates

G1: The number of ones and zeros in the sequence is approximately same: $p/2$ if p is odd and $(p \pm 1)/2$ if p is even.

G2: 1/2 of the runs have length 1, 1/4 of them have length 2, 1/8 length 3, ...
Furthermore blocks and gaps of same length are equally probable.

G3: Out-of-phase autocorrelation AC(k) has the same value with different values of k.

■ The cryptographic requirements for syncroneous stream ciphers

- All pseudorandom generators have a *finite number of states*. It returns to its initial state sooner or later. Produced sequence is *periodic*.

- In most generators knowing one state it is easy to calculate the previous and following states. Cryptographically this is unacceptable.

Following requirements must be fulfilled from cryptographically safe generators

Cryptographic requirements for a pseudorandom generator

C1: The period of sequnce $\{s_i\}$ must be very large (2^{50}).

C2: The generation of bits $\{s_i\}$ must be fast and easy

C3: Knowing message - ciphertext pairs should not make it possible to reveal the whole sequence ("known plaintext attack").

3.3 Linear Feedback Shift Registers (LFRS)

Methods of generating pseudorandom numbers are for example. linear congruence generators , Blum Blum Shub-generators and so on.
In cryptography a popular method is also using of Linear Feedback Shift Registers (LFSR) .

LFSR properties

1. LFSR is easy to implement on a chip
2. LFSR produces bit fast
3. LFSR is known mathematically well (Golombs book about LFSR:s)

Principle of LFSR

1. LFSR is a register of n bits $s_0, s_1, \ldots, s_{n-1}$
2. The initial values of bits define the initial state
3. Next state is obtained by
   A) moving bits to the left
   B) calculating the last bit with XOR from chosen bits of LFRS

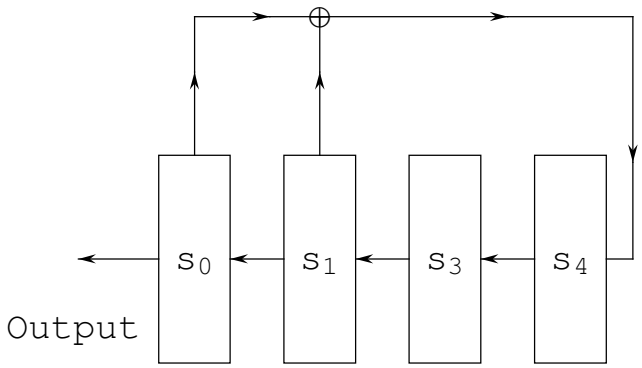
The formula for the last bit is called **feedback function**

$$s_{n-1} = f(s_0, s_1, \ldots, s_{n-1}) = \sum_{k=0}^{n-1} c_k s_k$$

Where coefficients c_k are either 0's or 1's.
Initial state and feedback functions determine later states of LFSR

A LFSR of length n can produce max $2^n - 1$ different states.
(–1 is because from (0,0,0,0) you get nothing else)

Picture of 4 bit LFSR and feedback function $f = s_0 \oplus s_1$



Implementation with *Mathematica* when initial state is (0,1,0,1).

```
{s0, s1, s2, s3} = {0, 1, 0, 1};  (* initialization *)

tbl = {{s0, s1, s2, s3}};   (* store the initial state *)

Do[ {s0, s1, s2, s3} = {s1, s2, s3, Mod[s0 + s1, 2]};
      tbl = Append [tbl, {s0, s1, s2, s3}], {i, 1, 15}];

tbl // MatrixForm    (* print the states *)
```

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

Notice: First and last states are same => LFSR has a period 15.

- **m -sequences (maximal length sequences)**

  Previous LFSR produces a sequence of its states. At the same time it produces a bit sequence, if one takes into the sequnce the s0 - bits of the states.

  Mathematica code for the

```
s = {0, 1, 0, 1};
seq = {};
Do[
  f = Mod[s[[1]] + s[[2]], 2];
  seq = Append[seq, s[[1]]];
  s[[1]] = s[[2]];
  s[[2]] = s[[3]];
  s[[3]] = s[[4]];
  s[[4]] = f,
  {i, 1, 30}];
seq
```

```
{0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0,
 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1}
```

Period of the bit sequence is $2^4 - 1 = 15$.

Definition:
A bit sequenc e produced by a LFSR of length n is said to be m - sequence (or maximal length sequence), if its period is maximal $2^n - 1$.

LFSR does not automatically create m - sequences with period $2^n - 1$. Usually period is smaller

Example of another choice of feedback function, which gives a smaller period

```
{s0, s1, s2, s3} = {0, 1, 0, 1};   (* initialization *)

tbl = {{s0, s1, s2, s3}};   (* store the initial state *)

Do[ {s0, s1, s2, s3} = {s1, s2, s3, Mod[s0 + s2, 2]};
      tbl = Append [tbl, {s0, s1, s2, s3}], {i, 1, 15}];

tbl // MatrixForm    (* print the states *)
```

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Now state 1 = state 7. LFSR has a period 6, much less than maximal $2^4 - 1$.

1. In syncroneous stream ciphers m -sequences are necessary because they fufill the requirement of large periods.

2. The choice of feedback coefficients is crucial for getting m- sequences

3. Samuel W. Golom has written a book " The Theory of LFSR's " in 1950's. According to his theory a n -bit **LFSR produces maximal length sequences if and only if its characteristic polynomial is an irreducible primitive polynomial of the finite field F_{2^n}.** This theory is not explained in detail in this material.

- **LFSR security**

The use of one LFSR does not fulfill the requirement C3 of cryptographic safety.
The initial state of the register could be traced easily.

- **LFSR - combinations**

Using a combination of several LFSRs it is possible to
A) fulfill the cryptographic requirements
B) increase the period

It can be shown:

Using 3 LFSR's with coprime lengths n_1, n_2 and n_3 and calculating the output sequence with XOR from the outputs generated by the individual LFSR's, we get a sequence with period

$$(2^{n1}-1)(2^{n2}-1)(2^{n3}-1)$$

Even this system does not quarantee cryptographic safety

In combining the output streams in addition to XOR also other techniques are used. This may decrease the period, but increase the safety.

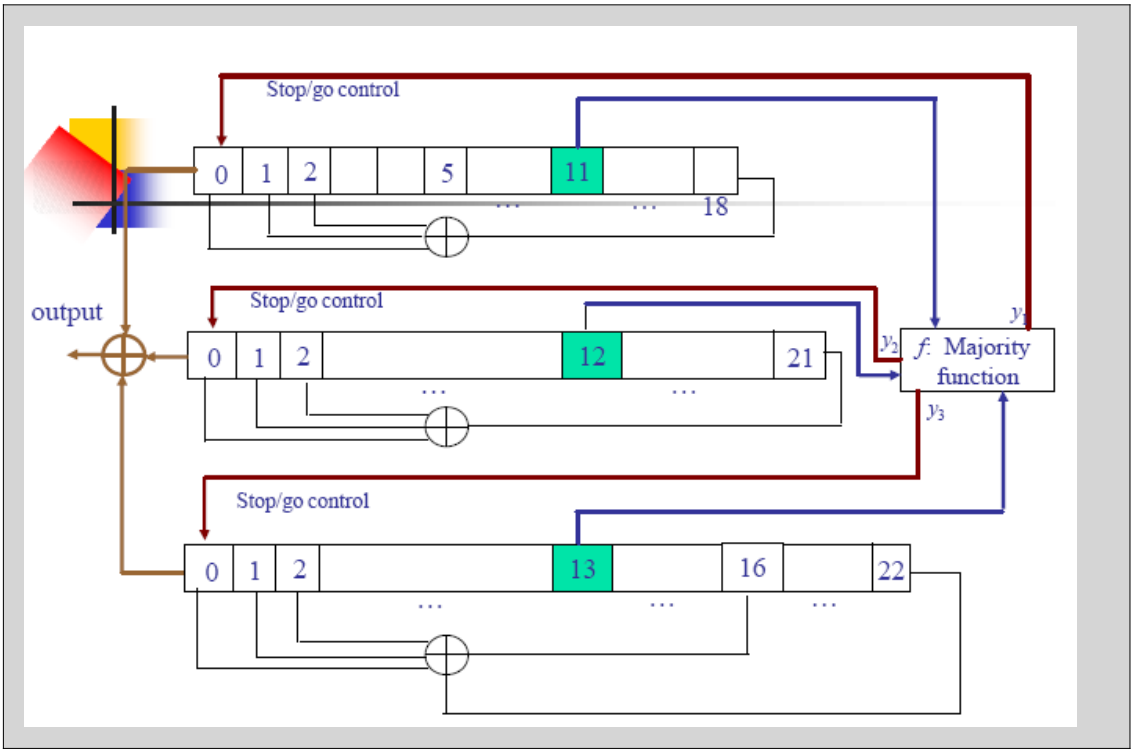
A5 (used in GSM and GPRS) algorithm is a good example of these techniques

■ 3.4 Case : GSM - encryption algorithm A5

A5 was developed 1991 for GSM

There exists several versions
A5/2 (USA) is the weakest
A5/1 is stronger (used in Europe)
A5/3 is the newest, but not widely used

Picture of A5



- **Algorithm**

A5

A5/1 has 3 LFSR - registers A, B and C, with lengths 19, 22 ja 23.

1) **Feedback - funktio** is s0⊕s1⊕s2 in A and B and s0⊕s1⊕s2⊕s16 in C

2) A5 produces one bit during each clock pulse. Output is XOR sum of the leftmost bits.

3) ***Majority function***, which is calculated from bits 11, 12 and 13, define which of the registers step forward during one clock pulse.

4) ***Key*** is the initial state. Its length is 19+22+23 = 64.

5) LFSR output stream k is XOR added with the message stream to produce the cipher

encryption: $\mathbf{c = m \oplus k}$

6) To improve security the LFSR's are initialized after every 114 bit blocks. The new initial state is calculated using encryption key and the block number.
Block numbers start from zero after 3 h 29 min

Majority - function f

1. The arguments are 3 bits (x1, x2,and x3) , one from each LFSR (green bits in picture).
2. It returns 3 bits (y1, y2 and y3), which act as Stop and Go -bits to the three LFSRs.
(If for example y1 = 1, LFSR1 steps forwards, if y1 = 0, it does not move)
3. Function is specified by the followig truth table:

x1	x2	x3	y1	y2	y3
1	1	1	**1**	**1**	**1**
0	0	0	**1**	**1**	**1**
1	1	0	**1**	**1**	**0**
0	0	1	**1**	**1**	**0**
0	1	1	**0**	**1**	**1**
1	0	0	**0**	**1**	**1**
1	0	1	**1**	**0**	**1**
0	1	0	**1**	**0**	**1**

- **A5/1 security**

1. A5 can be broken in few hours with PC

2. Without majority function the period of sequence would be $(2^{19}-1)(2^{22}-1)(2^{23}-1) = 2^{64}$.
Majority function decreases period down to $4/3*2^{23}$ (experimental fact)

3. Collision problem: Different initial states can lead to same bit stream.

4. Majority function is weak against linear cryptoanalysis.

5. Brute force - resistance:
* With Pentium III PC it takes 250 hours.
* Biryukov and Shamir claim to be able to break A5 with PC in less than 1 second

GSM -phone uses 3 algorithms:

A5 = encryption

A3 = authentication

A8 = key agreement protocol

A3 and A8 are explained in more detail later.

Implementation of A5 with *Mathematica* is found at
http://tl.ramk.fi/~jouko.teeriaho/krypto2006/a5.nb