Stream ciphers

Anne Canteaut

Anne.Canteaut@inria.fr

http://www-rocq.inria.fr/secret/Anne.Canteaut/

Summer School, Šibenik, June 2014

Outline

- Basic principle
- General model for a PRNG
- Generic attacks
- Main families of PRNG

Stream ciphers vs block ciphers [Handbook of crypto]

Block cipher: a family of permutations operating on large blocks (64 or 128 bits), depending on a key.

Stream cipher: an encryption scheme which encrypts individual digits (usually bits or bytes) of a plaintext one at a time, using a transformation which varies with time.

Stream ciphers vs block ciphers [Handbook of crypto]

Block cipher: a family of permutations operating on large blocks (64 or 128 bits), depending on a key.

Stream cipher: an encryption scheme which encrypts individual digits (usually bits or bytes) of a plaintext one at a time, using a transformation which varies with time.

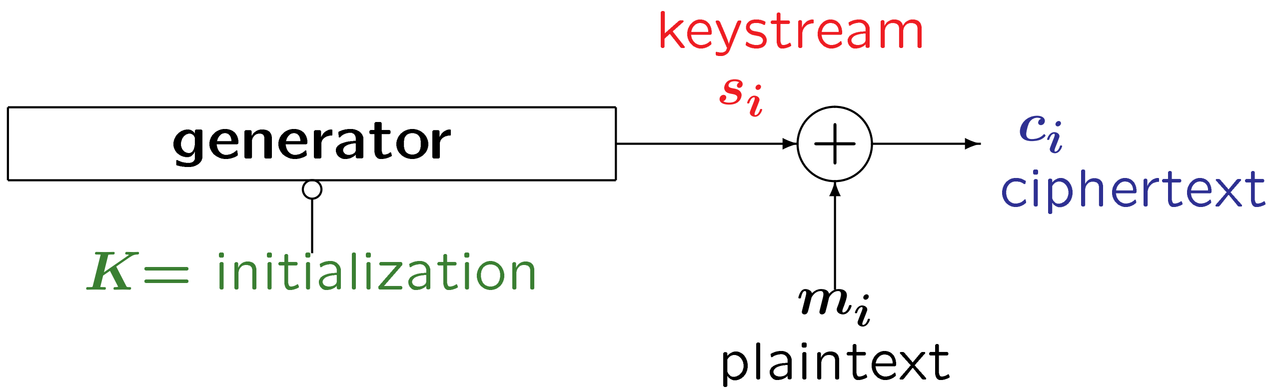
Remark:

- transformations of different nature: stream ciphers operate on variable-length messages while block ciphers operate on fixed-length messages.
- block ciphers must be used with a mode of operation (e.g., CBC).

Some modes of operation build a stream cipher from a block cipher (e.g. AES-CTR)!

Additive synchronous stream ciphers

The keystream is a pseudo-random sequence derived from a (short) secret key.



Advantages of stream ciphers

- no buffering;
- precomputation is possible;
- no padding (important if short packets);
- no error-propagation.

When do we use a stream cipher?

- low-bandwidth communications
- noisy transmissions
- in most applications....

Model for a pseudo-random generator

Pseudo-random generator for additive stream ciphers

Definition. Finite-state automaton which produces in a deterministic way a long sequence s from a (short) seed such that, for an adversary who knows everything except the seed, it is impossible to distinguish s from a random sequence with a significantly lower complexity than an exhaustive search for the seed.

Not to be confused with

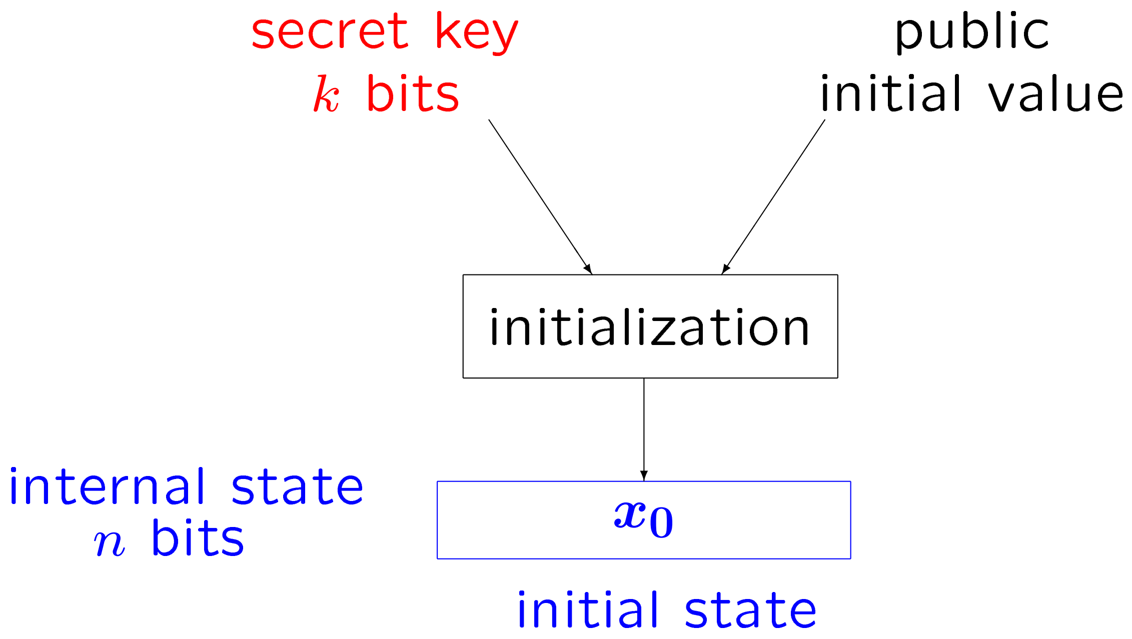
Random number generator:

- Thermal noise from a resistor,
- Observations of some physical events available to the software, e.g., `/dev/random/`

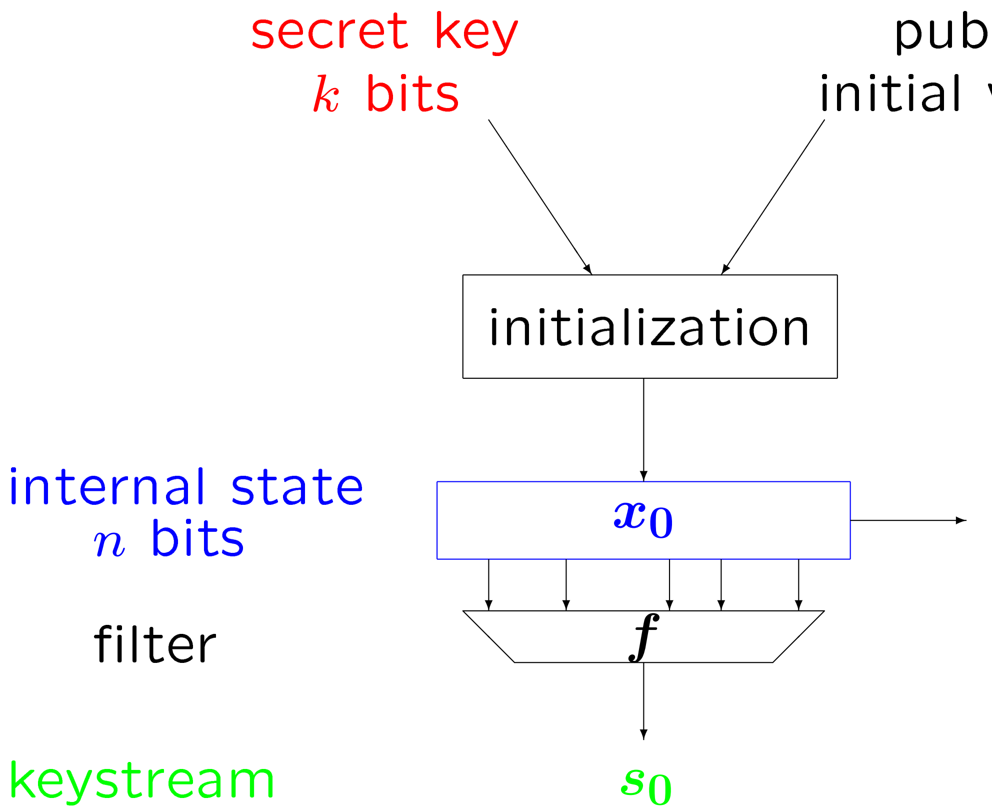
`rand()` in the ANSI/ISO C standard:

```
int rand(void) // RAND_MAX assumed to be 32767
{
      next = next * 1103515245 + 12345;
      return (unsigned int)(next/65536) % 32768;
}
```

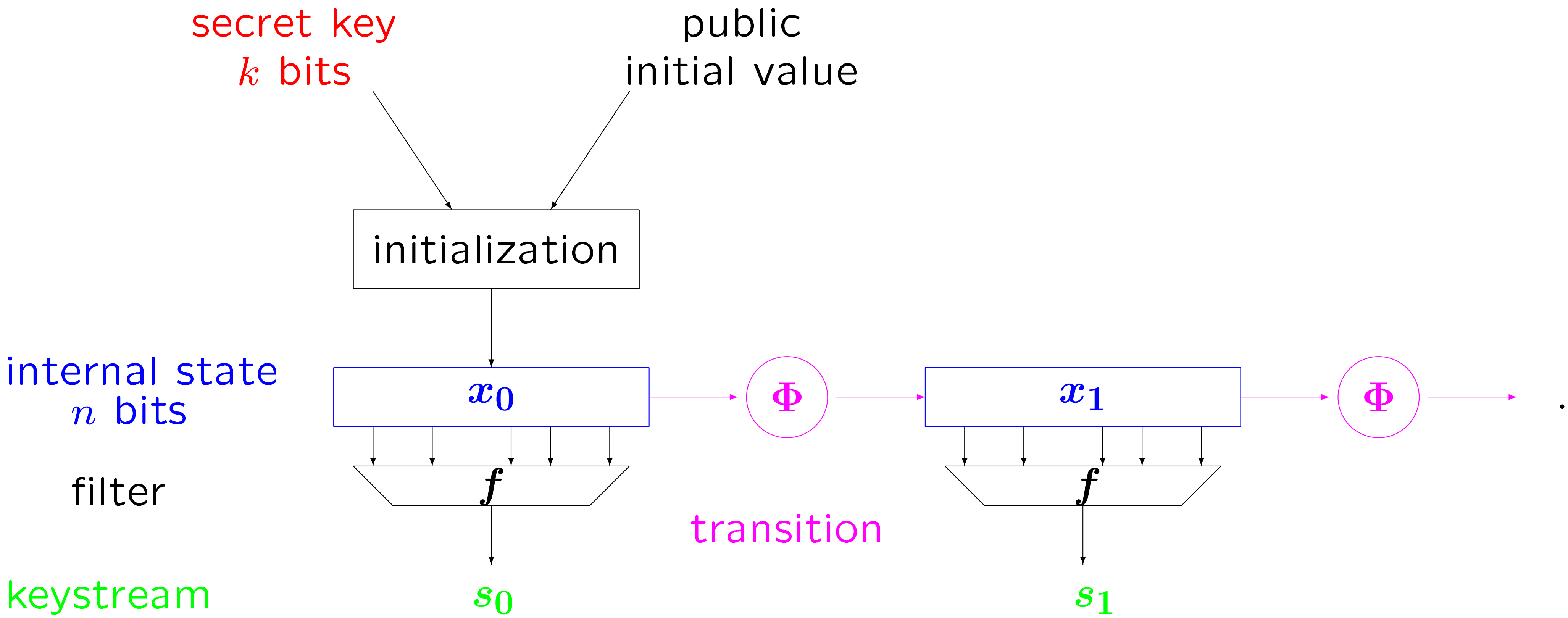
General construction



General construction



General construction



Generic attacks

Different types of attacks

Attacker types:

ciphertext only; known plaintext (or chosen plaintext/ciphertext); related IVs; chosen IV.

Attacker goals:

- Key recovery;
- Initial-state recovery (only for the current IV);
- Next-bit prediction;
- Distinguisher (e.g., for checking whether some eavesdropped ciphertext corresponds to a given plaintext)...

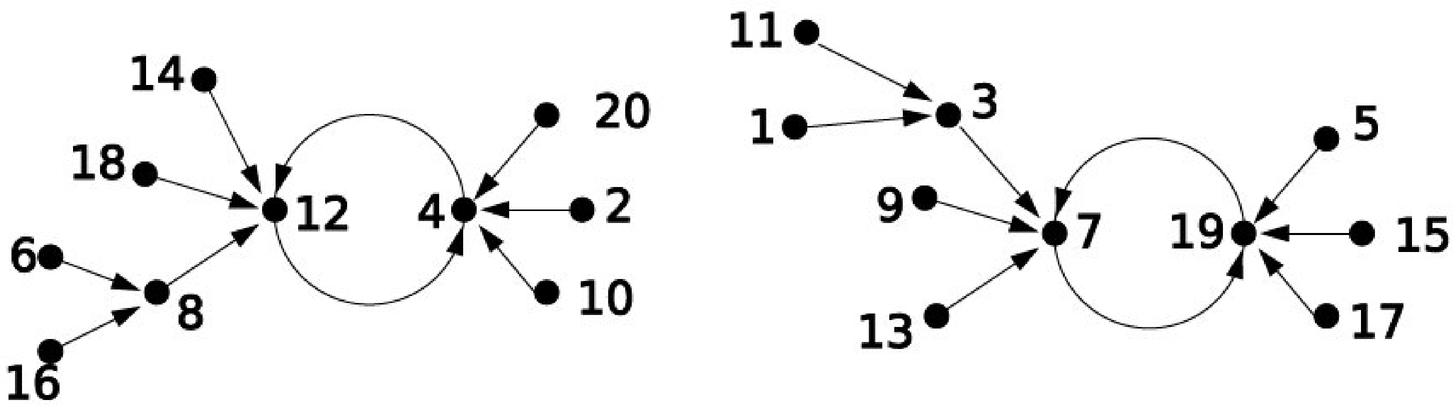
The last two attacks are equivalent [Yao 82].

Period of the sequence of internal states

For any initial x_0, $(x_t)_{t\geq 0} = (\Phi^t(x_0))_{t\geq 0}$ should have a high period.

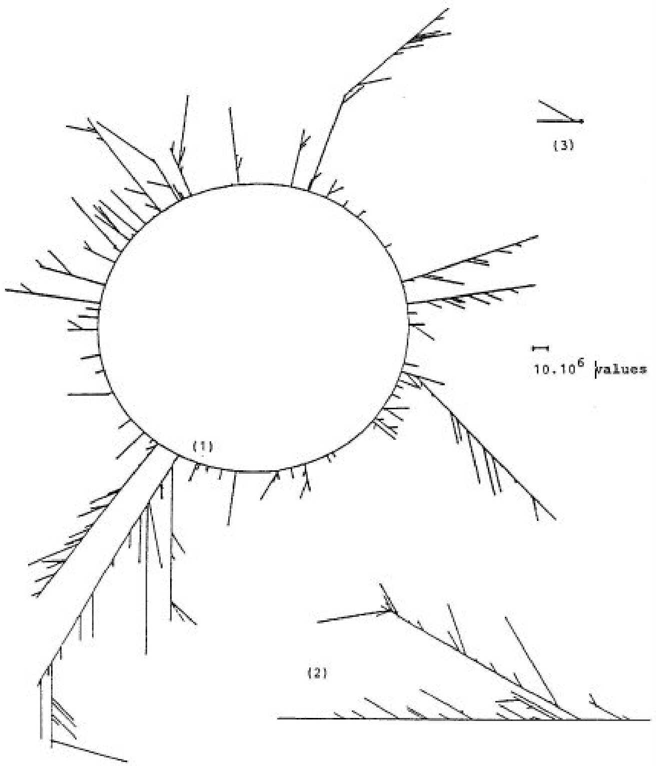
Functional graph of the transition function:

$$\begin{array}{rcl} \Phi : \{1,\ldots,20\} & \longrightarrow & \{1,\ldots,20\} \\ x & \longmapsto & (x-1)^2+2 \bmod 20+1 \end{array}$$



Functional graph of a random mapping [Flajolet Odlyzko 89]

Example from [Quisquater Delescaille 87]:



One "giant component" with length $\mathcal{O}(\sqrt{N})$ where N is the size of the input/output set.

$\mathcal{O}(\sqrt{N})$ points in a cycle (entropy of the state after several iterations).

Choosing the transition function Φ

Two possibilities:

- Choose a random-looking mapping/permutation operating on a large internal state: the period of $(x_t)_{t\geq 0}$ is expected to be close to $2^{\frac{n}{2}}$. Short cycles exist but are unlikely to occur. Eg: RC4.

- Choose a permutation with some known mathematical properties operating on a small internal state: the period of $(x_t)_{t\geq 0}$ can be proved to be close to 2^n. Short cycles are avoided. Eg: counter, LFSR.

Time-memory-data trade-off [Golic 95][Babbage 95]

For a stream cipher:

$$\begin{array}{rccc} F: & \{0,1\}^n & \longrightarrow & \{0,1\}^n \\ & x_0 \text{ (initial state)} & \longmapsto & s_0, s_1, \ldots, s_{n-1} \end{array}$$

Improvement:

If we need to find a preimage for a single y among several ones, the trade-off can involve the amount of data.

If D consecutive bits of the keystream are known, we get $(D - n + 1)$ frames of n bits: $y_t = s_t, s_{t+1} \ldots s_{t+n-1}$ for $0 \leq t \leq D - n$.

$$\text{Time} = D \quad \text{memory} = \text{precomputation} = M = \frac{2^n}{D}.$$

We get an attack with data/time/memory complexity $2^{\frac{n}{2}}$.

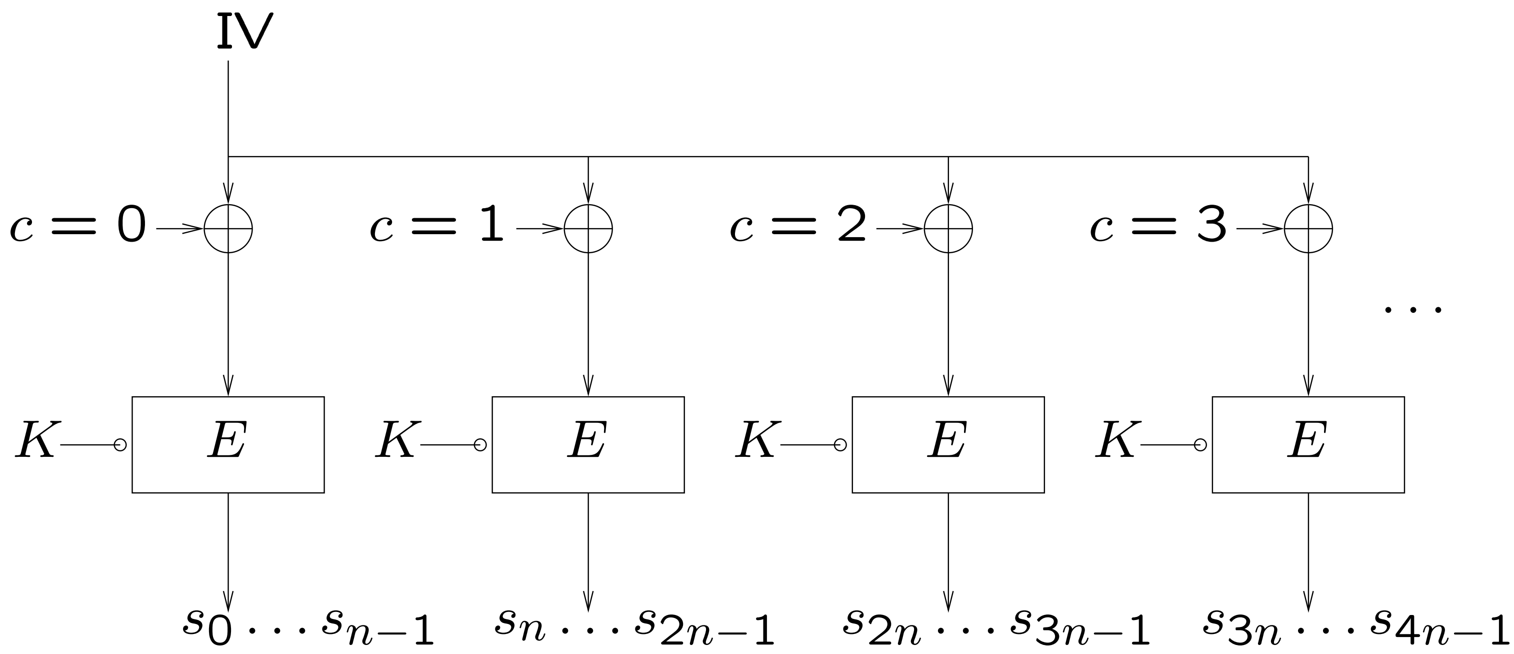
Resisting the main generic attacks

- The internal state must be at least twice larger than the key.
- Either the internal state should be large with a random-looking next-state mapping Φ, or it must be guaranteed that Φ has no short cycles.
- The generator must pass the statistical tests. In particular, the filtering function f must be balanced.
- At least one function among Φ and f must be nonlinear.

Main families of generators

Block-cipher based PRNG

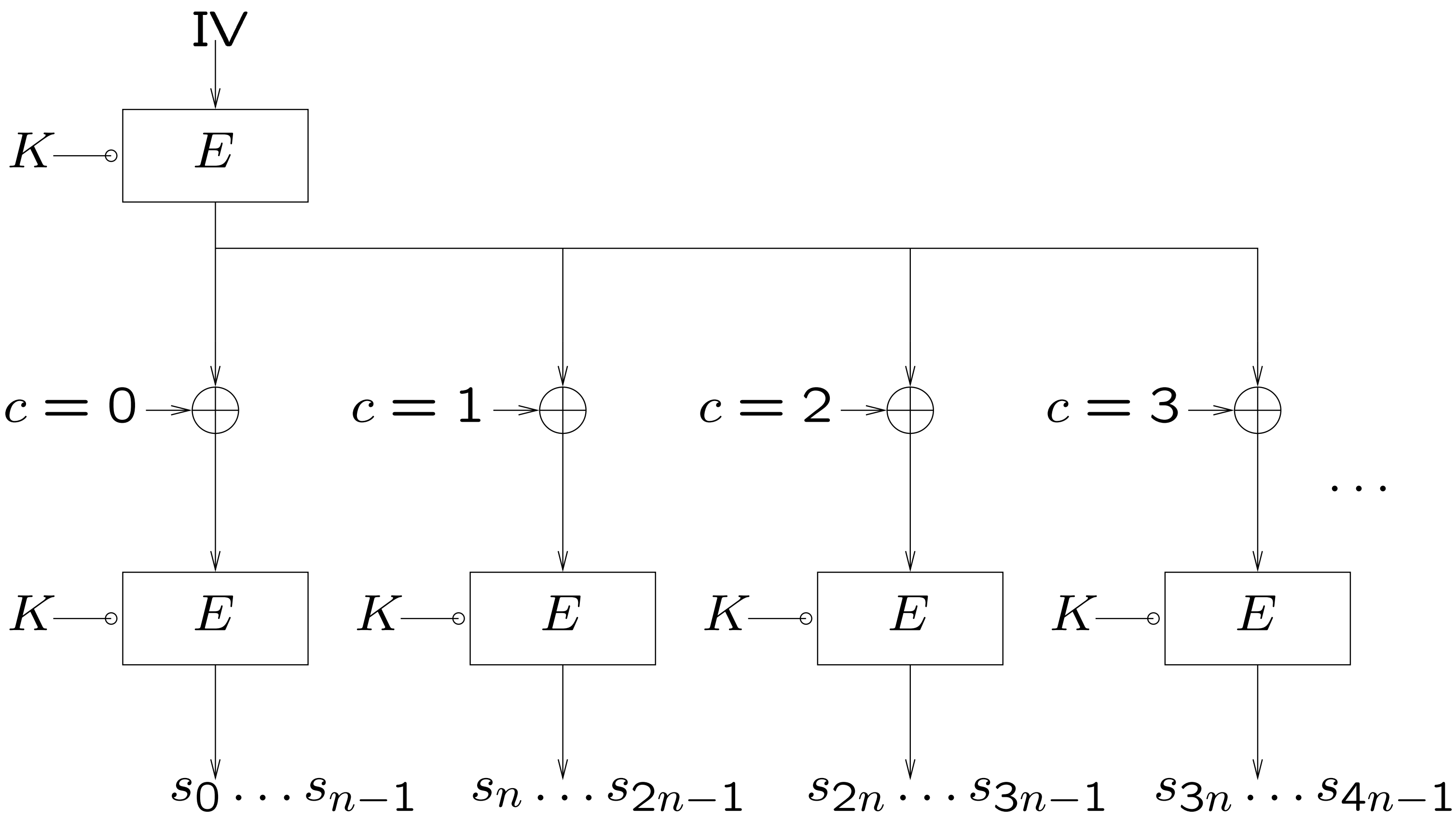
Counter mode (CTR)



Chosen-IV distinguishing attack: if we encrypt $\boldsymbol{m} = (\boldsymbol{m_0}, \boldsymbol{m_1})$ with $(\boldsymbol{K}, \boldsymbol{IV})$ and $\boldsymbol{m'} = (\boldsymbol{m_1}, \boldsymbol{m_2})$ with $(\boldsymbol{K}, \boldsymbol{IV} + \mathbf{1})$, we get two identical ciphertext blocks, namely $\boldsymbol{c_1} = \boldsymbol{c'_0}$.

Block-cipher based PRNG

Modified counter mode (Milenage in UMTS)



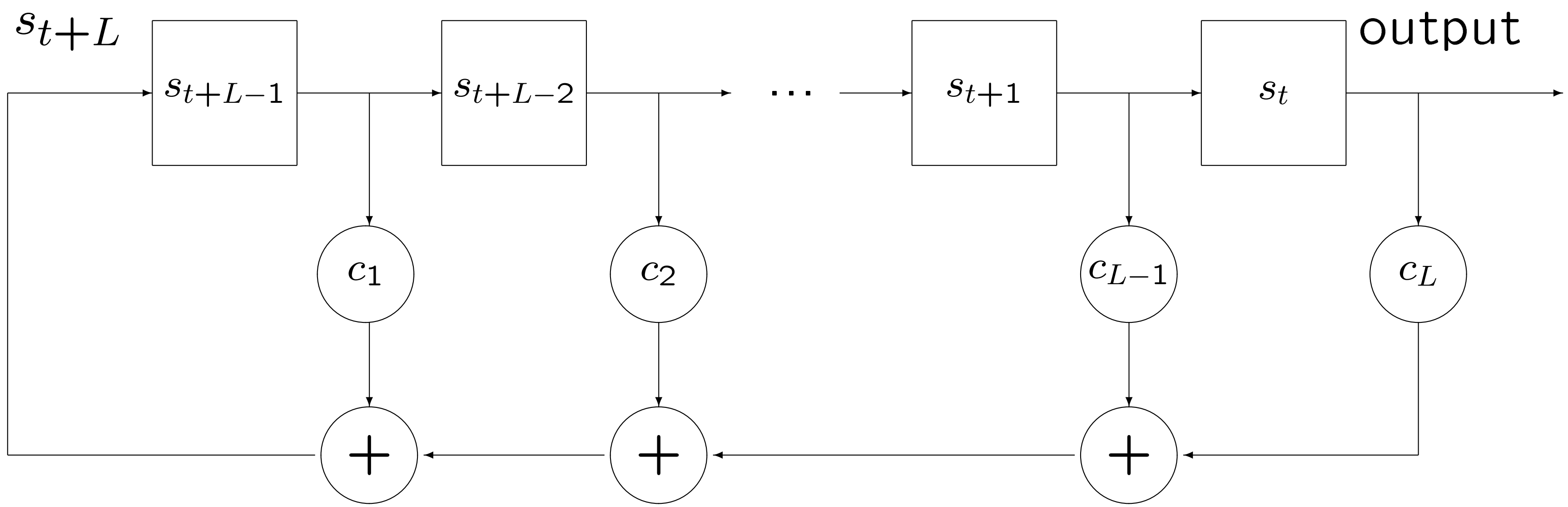
Distinguisher with complexity $\mathcal{O}\left(2^{\frac{n}{2}}\right)$ where n is the block size.

Dedicated PRNG

Typical applications:

- high throughput in software (faster than the AES);
- low-cost hardware (heavely restricted gate count or power).

LFSR



$c_1, \ldots, c_L$ are the binary feedback coefficients of the LFSR.

The binary sequence produced by the LFSR satisfies a linear recurrence relation of degree L:

$$s_{t+L} = \bigoplus_{i=1}^{L} c_i s_{t+L-i}, \quad \forall t \geq 0 \; .$$

Period of the sequence

Any sequence generated by an LFSR of length L is *ultimately periodic*, *i.e.*, the sequence obtained by ignoring a certain number of elements at the beginning is periodic, and its period is at most $2^L - 1$.

Moreover, if $c_L = 1$, the LFSR is non-singular, and it produces periodic sequences.

Feedback polynomial:

$$P(X) = 1 + \sum_{i=1}^{L} c_i X^i \ .$$

Definition. The minimal polynomial of a sequence $(s_t)_{t \geq 0}$ is the feedback polynomial with the lowest possible degree for an LFSR which can generate the sequence.

LFSRs with maximal period

Proposition. The least period of a linear recurring sequence is equal to the order of its minimal polynomial, i.e., the least positive integer e for which $P_0(X)$ divides $X^e + 1$.

Then, a sequence has maximal period $2^{\deg P_0} - 1$ if and only if P_0 is a primitive polynomial.

LFSRs with maximal period

Proposition. The least period of a linear recurring sequence is equal to the order of its minimal polynomial, i.e., the least positive integer e for which $P_0(X)$ divides $X^e + 1$.

Then, a sequence has maximal period $2^{\deg P_0} - 1$ if and only if P_0 is a primitive polynomial.

Similar to a counter:

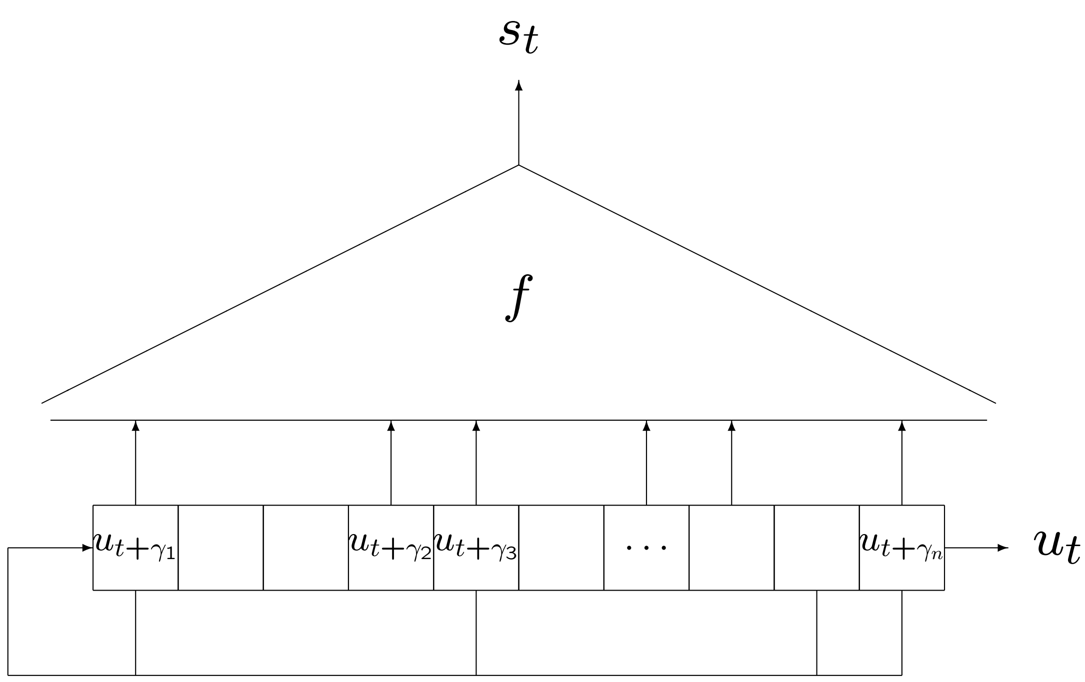
The sequences produced by an LFSR of length L with primitive feedback polynomial P are of the form

$$\{x_0, \alpha x_0, \alpha^2 x_0, \alpha^3 x_0, \ldots.\}, \text{ with } x_0 \in GF(2^L)^*$$

where α is a root of P, i.e.,

$$\{\alpha^i, \alpha^{i+1 \bmod (2^L-1)}, \alpha^{i+2 \bmod (2^L-1)}, \ldots.\}, \text{ with } 0 \leq i \leq 2^L - 2$$

The filter generator



$$s_t = f(u_{t+\gamma_1}, u_{t+\gamma_2}, \dots, u_{t+\gamma_n}), \quad \forall t \geq 0$$

where $(u_t)_{t\geq 0}$ is the LFSR sequence, f is a balanced Boolean function of n variables, $n \leq L$, and $(\gamma_i)_{1\leq i\leq n}$ is a decreasing sequence of nonnegative integers.

Linear complexity of the filter generator

Definition. For a semi-infinite sequence $s = (s_t)_{t \geq 0}$, the linear complexity $\Lambda(s)$ is the smallest integer Λ such that s can be generated by an LFSR of length Λ, and is ∞ if no such LFSR exists.

Lower bound on the linear complexity [Rueppel 86]:

if L is a large prime,

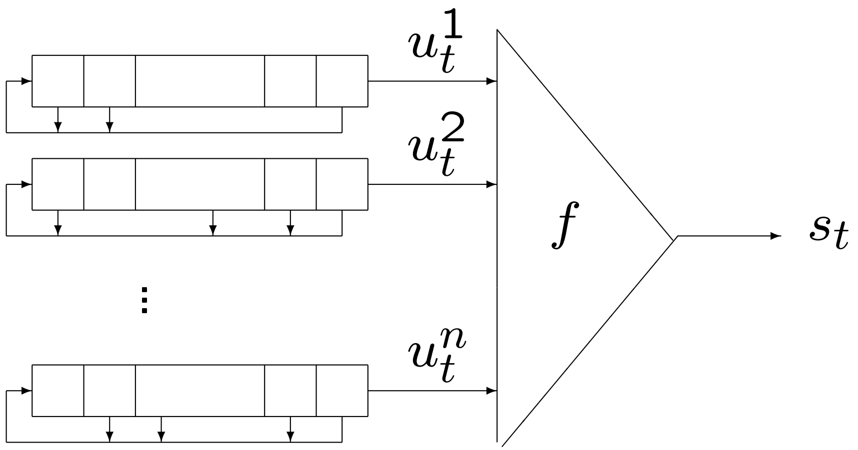
$$\Lambda(\mathrm{s}) \geq \binom{L}{d}$$

for most filtering functions with algebraic degree d.

$\rightarrow$ The degree of f should be as high as possible.

May be vulnerable to algebraic attacks and variants [Courtois Meier 03]

The combination generator



The outputs of n LFSRs are combined by a Boolean function of n variables:

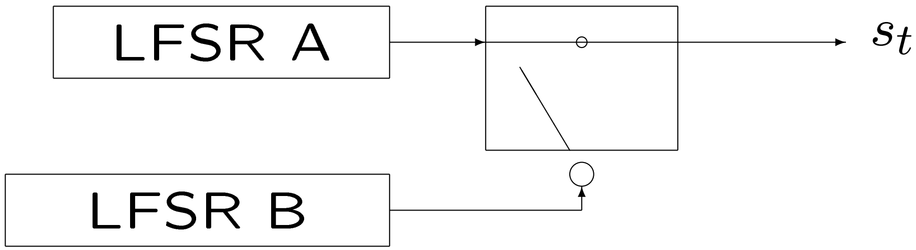
$$s_t = f(u_t^1, u_t^2, \ldots, u_t^n)$$

Correlation attacks [Siegenthaler 85] and many variants [Meier-Staffelbach88]

LFSR with irregular clocking

The generator is composed of one or several LFSRs.
Some LFSR bits decide which LFSR to clock and how often.

The shrinking generator [Coppersmith-Krawczyk-Mansour 93]

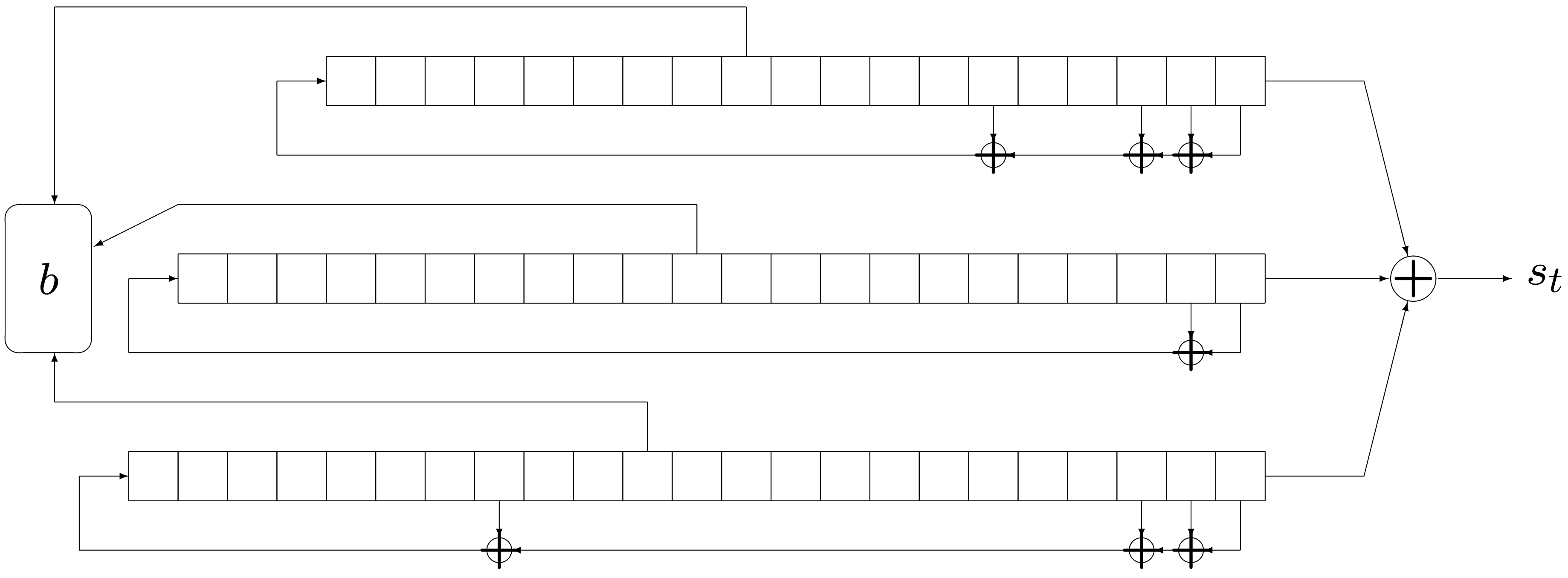


If LFSR B outputs 0, the output bit of LFSR A is discarded.

$$\mathbf{\Lambda(s) \geq L_A 2^{L_B - 2}} .$$

A5/1 (GSM stream cipher)

3 LFSRs of lengths 19, 22 and 23.



$\rightarrow$The 64-bit internal state makes it vulnerable to TMDTO.

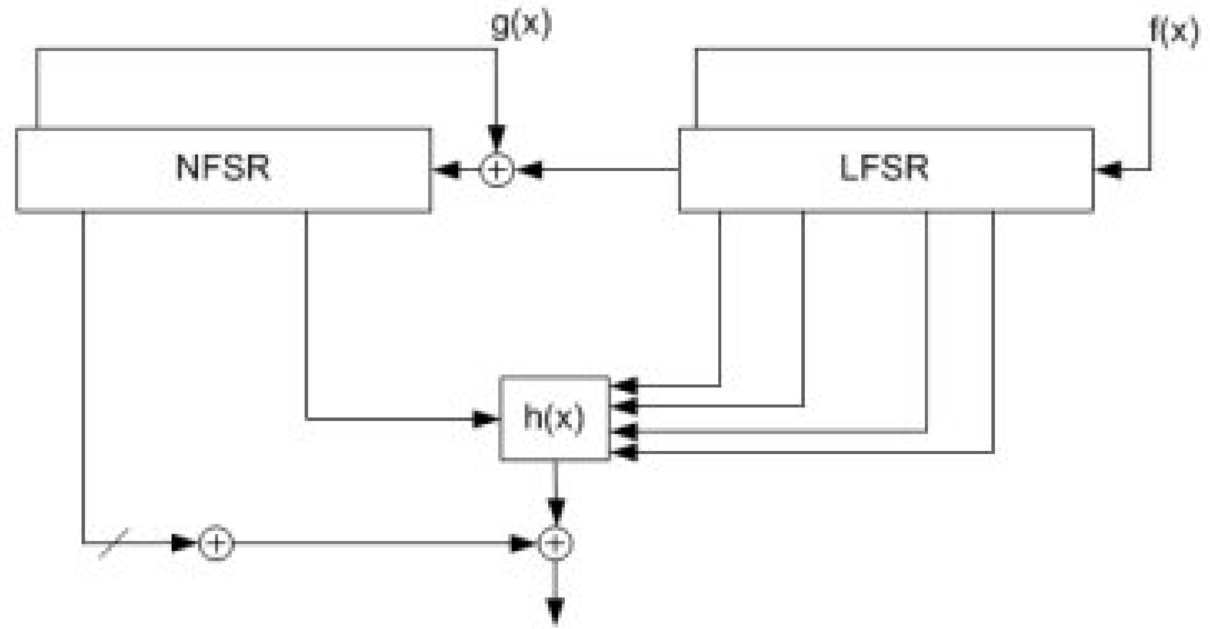
The eSTREAM project (2004-2008)

launched by the European network of excellence ECRYPT

`http://www.ecrypt.eu.org/stream/`

software applications	hardware applications
HC-128	Grain v1
Rabbit	MICKEY 2.0
Salsa20/12	Trivium
SOSEMANUK	

Grain v1 [Hell Johansson Meier]



Conclusions

By default:

use AES-CTR (IV-related attacks, distinguisher of complexity $\mathbf{2^{64}}$).

Otherwise:

- in software: table-driven generators (HC-128, not RC4!) LFSRs over $\boldsymbol{GF(2^{32})}$ (SNOW 2.0...)
- in hardware: LFSR and NLFSR-based designs.