ПОТОЧНЫЕ ШИФРЫ

А.Е.Жуков
Ассоциация РусКрипто

ПОТОЧНЫЕ ШИФРЫ

по материалам открытой печати

А.Е.Жуков
Ассоциация РусКрипто

Определение поточного шифра

Поточный шифр (*stream cipher*) задается следующими элементами:

- $\mathcal{P}$ – конечный алфавит, элементы которого являются знаками *открытого текста* (*plaintext*). Открытый текст представляется в виде конечной последовательности $x_1, x_2, \ldots, x_n$, где $x_t \in \mathcal{P}$.
- $\mathcal{C}$ – конечный алфавит, элементы которого являются знаками *шифртекста* (*ciphertext*). Шифртекст представляется в виде конечной последовательности $y_1, y_2, \ldots, y_n$ той же длины, что и соответствующий открытый текст, при этом $y_t \in \mathcal{C}$.
- $\mathcal{Z}$ – конечный алфавит, элементы которого являются знаками ключевого потока $k_1, k_2, \ldots, k_n, k_t \in \mathcal{Z}$.
- $\mathcal{K}$ – множество *секретных ключей* всей системы; мощность множества $\mathcal{K}$, как правило, весьма велика, но в то же время $|\mathcal{K}| << |\mathcal{Z}|^n$.

– *k – секретный ключ системы* , $k \in \mathcal{K}$, с помощью которого определяется ключевой поток $k_1, k_2, \ldots$

Определение поточного шифра

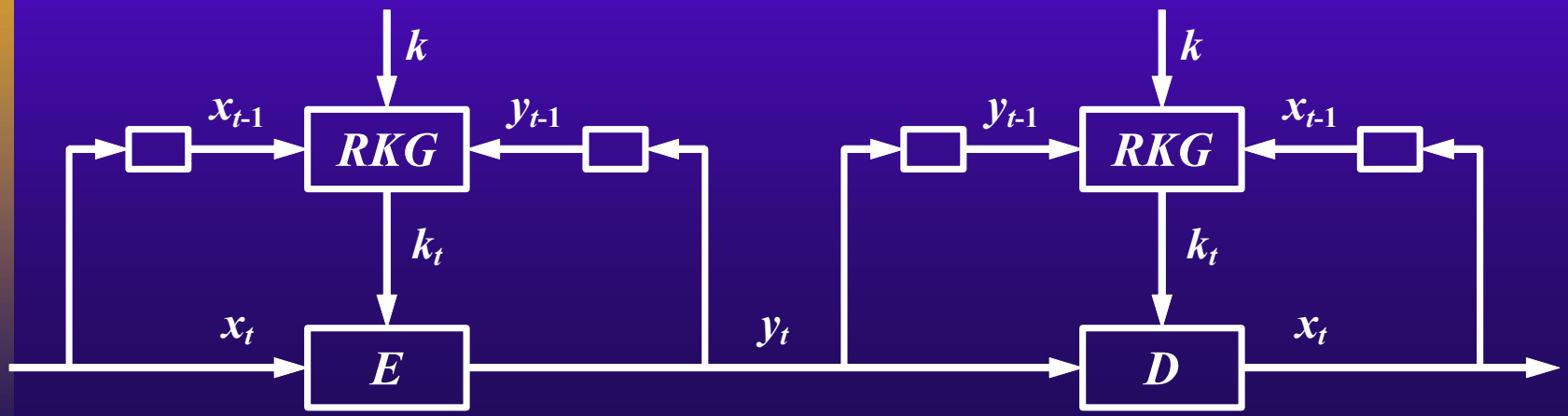
- $RKG = \{f_1, f_2, \ldots\}$ – *генератор ключевого потока* (*running key generator*, *RKG*) – семейство функций $f_1, f_2, \ldots$, таких, что для всех $t \geq 1$ функция $f_t : K \times \mathcal{P}^{t-1} \mapsto Z$, тогда знак ключевого потока

$$k_t = f_t(k, x_1, \ldots, x_{t-1}).$$

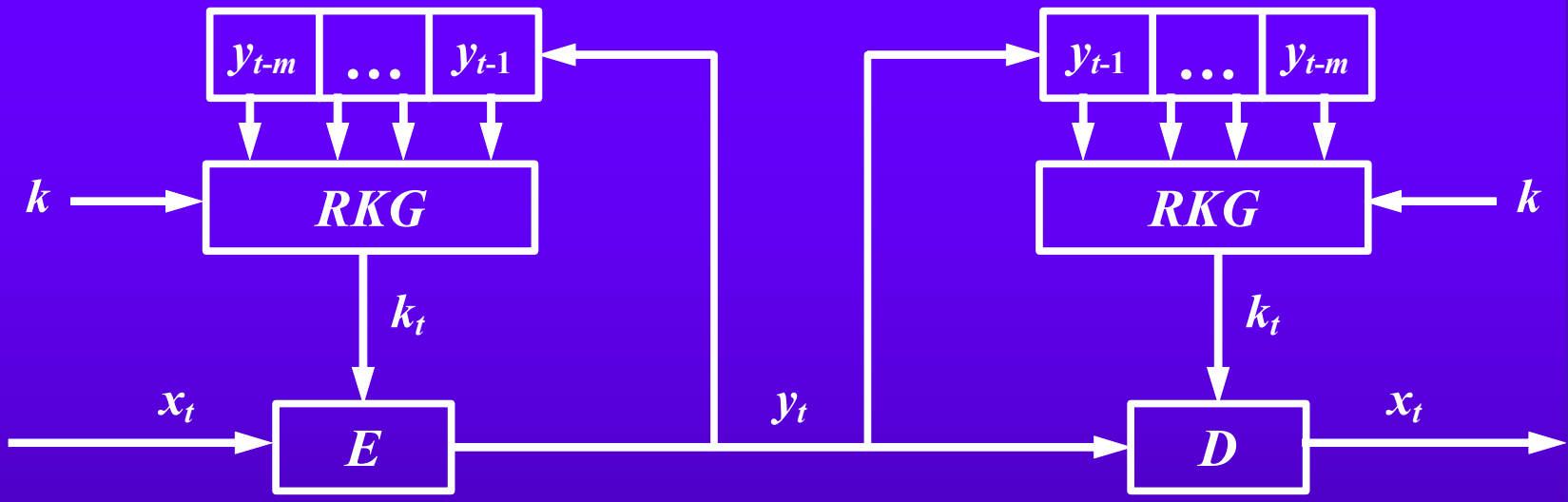
– $E(k_t, .)$ и $D(k_t, .)$ – семейство функций шифрования и расшифрования в общем случае зависящих от значения k_t

$$E(k_t, .) : \mathcal{P} \mapsto C; \qquad D(k_t, .) : C \mapsto \mathcal{P},$$

при этом $y_t = E(k_t, x_t)$; $x_t = D(k_t, y_t)$, т.е. функции E и D при фиксированном k_t являются взаимнообратными



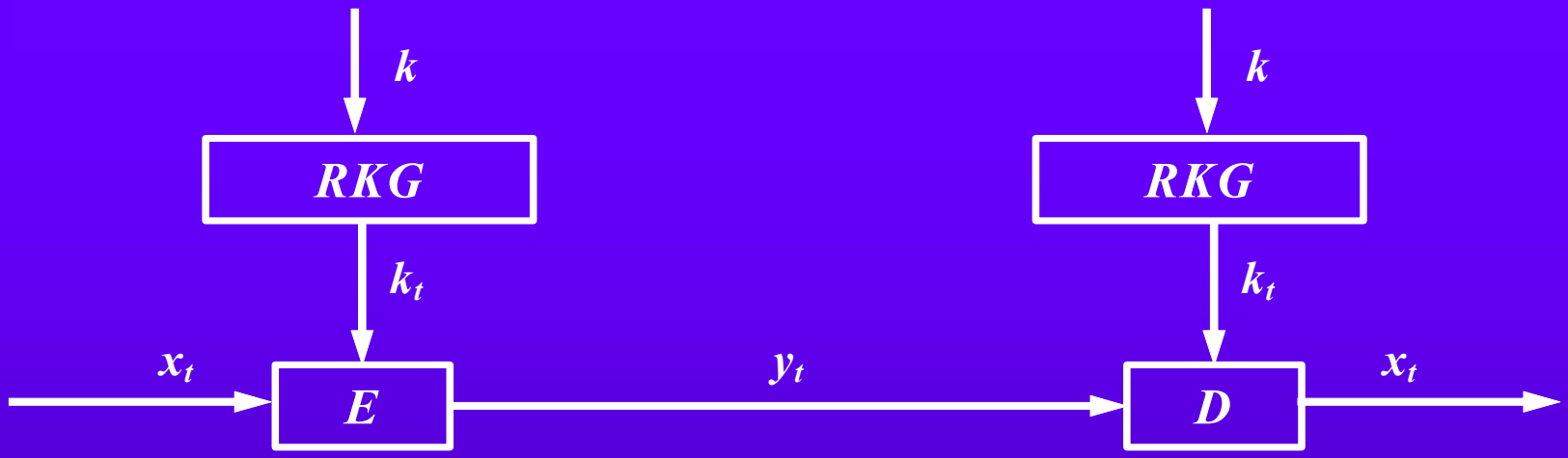
Самосинхронизирующийся поточный шифр



• **Самосинхронизация криптосистемы: если отдельные знаки шифртекста были удалены, изменены или вставлены, через *m* тактов работы (т.е. после получения *m* неискаженных знаков) криптосистема приходит в корректное состояние.**

• **Ограниченное распространение ошибки: если знаки шифртекста были удалены, изменены или вставлены, то не более чем *m* следующих знаков будут расшифрованы неверно.**

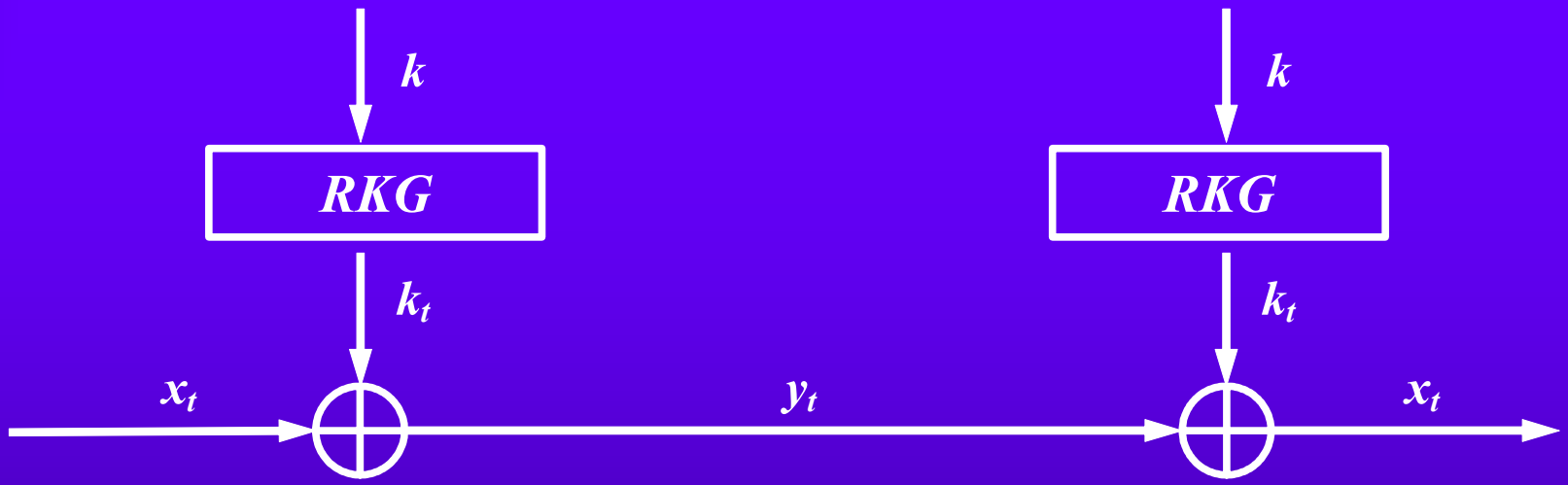
Синхронный поточный шифр



• **Обязятельная синхронизация криптосистем на передающем и приемном конце.**

• **Нераспространение ошибки: если знаки шифртекста были изменены (но не удалены или вставлены), то все следующие знаки будут верно расшифрованы.**

Аддитивный синхронный поточный шифр



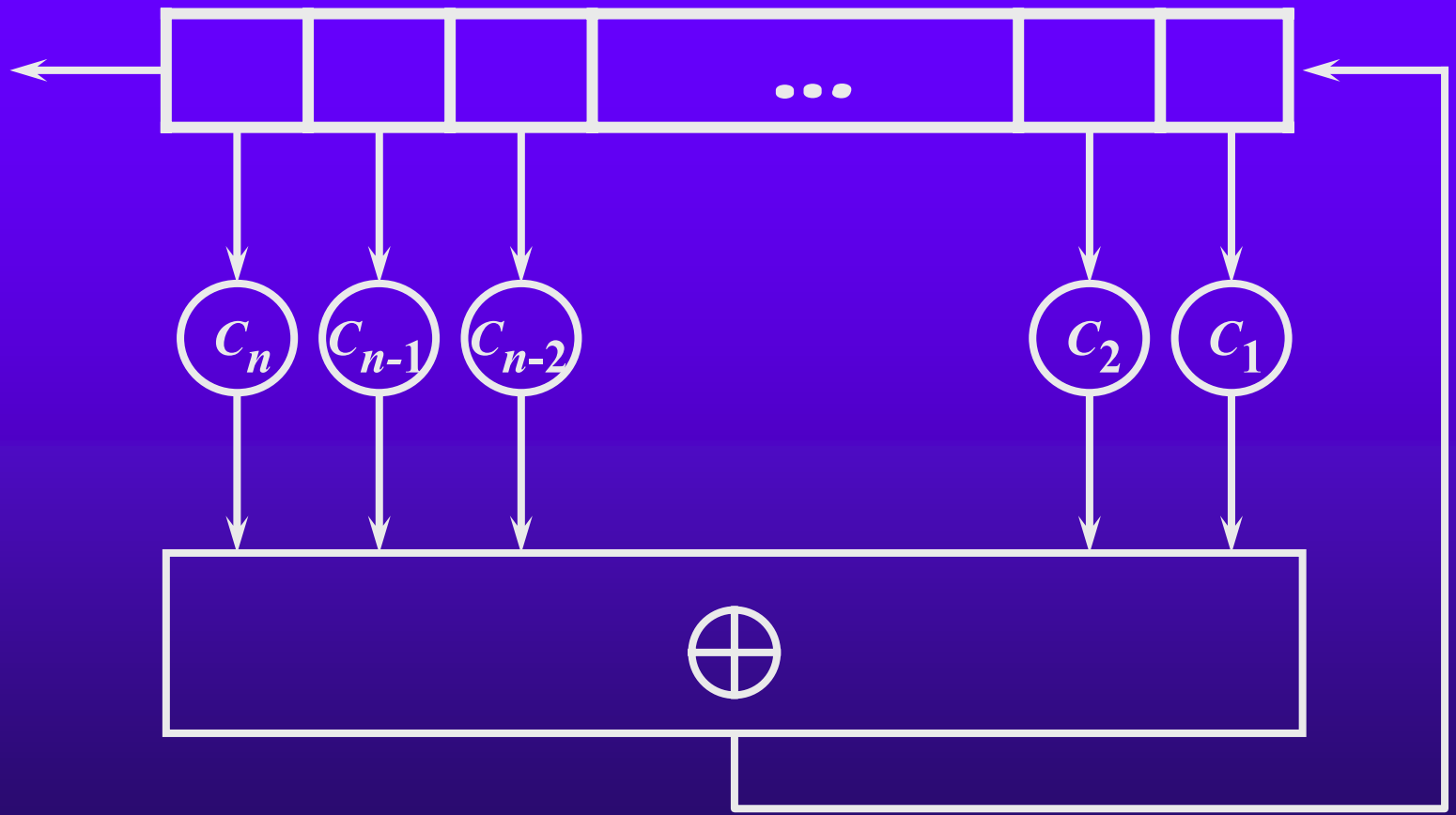
Наиболее распространенный тип поточного шифра (шифр гаммирования)

RKG – Running Key Generator

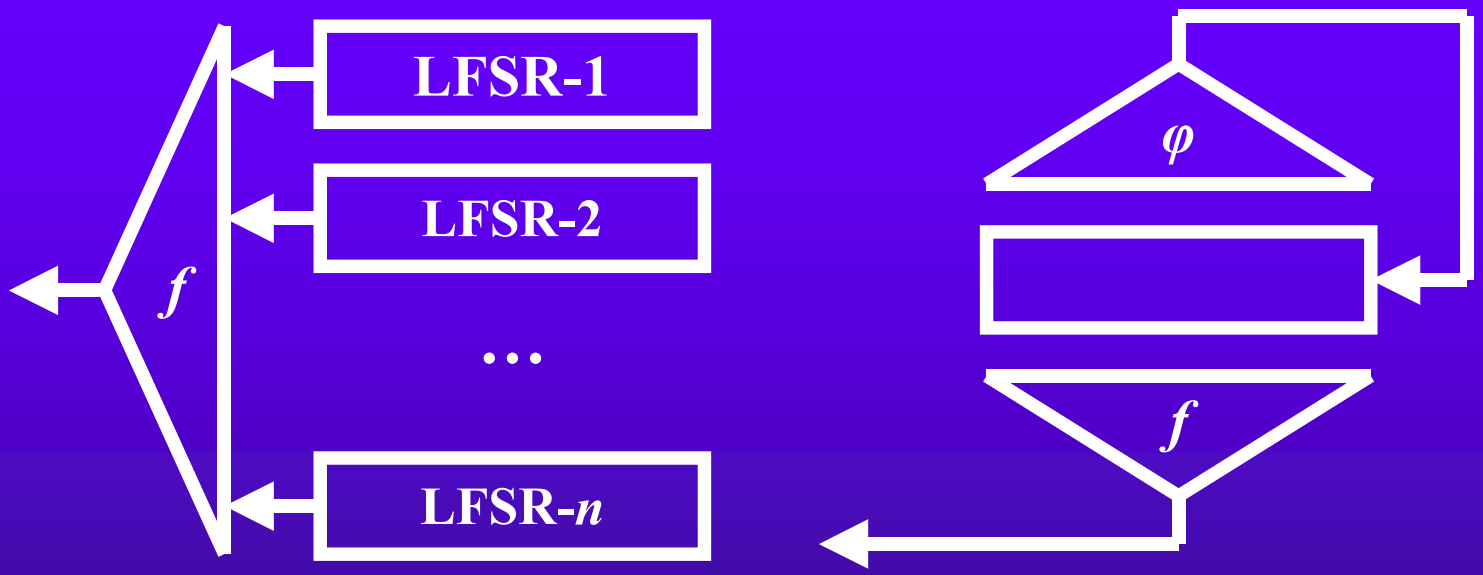
Рассматривая *RKG* как автономный автомат, Руппель выделял в конструкции типичного *RKG* *управляющую часть* (*driving part*) и *комбинационную часть* (*combining part*). Управляющая часть задает переход из одного внутреннего состояния автономного автомата в другое и обеспечивает большой период и хорошие статистические свойства выходной последовательности.

Комбинационная часть непосредственно принимает участие в выработке знаков выходной последовательности и отвечает за ее сложность и непредсказуемость, не разрушая при этом хороших свойств, обеспечиваемых управляющей частью.

Линейные регистры сдвига с обратной связью (LFSR)



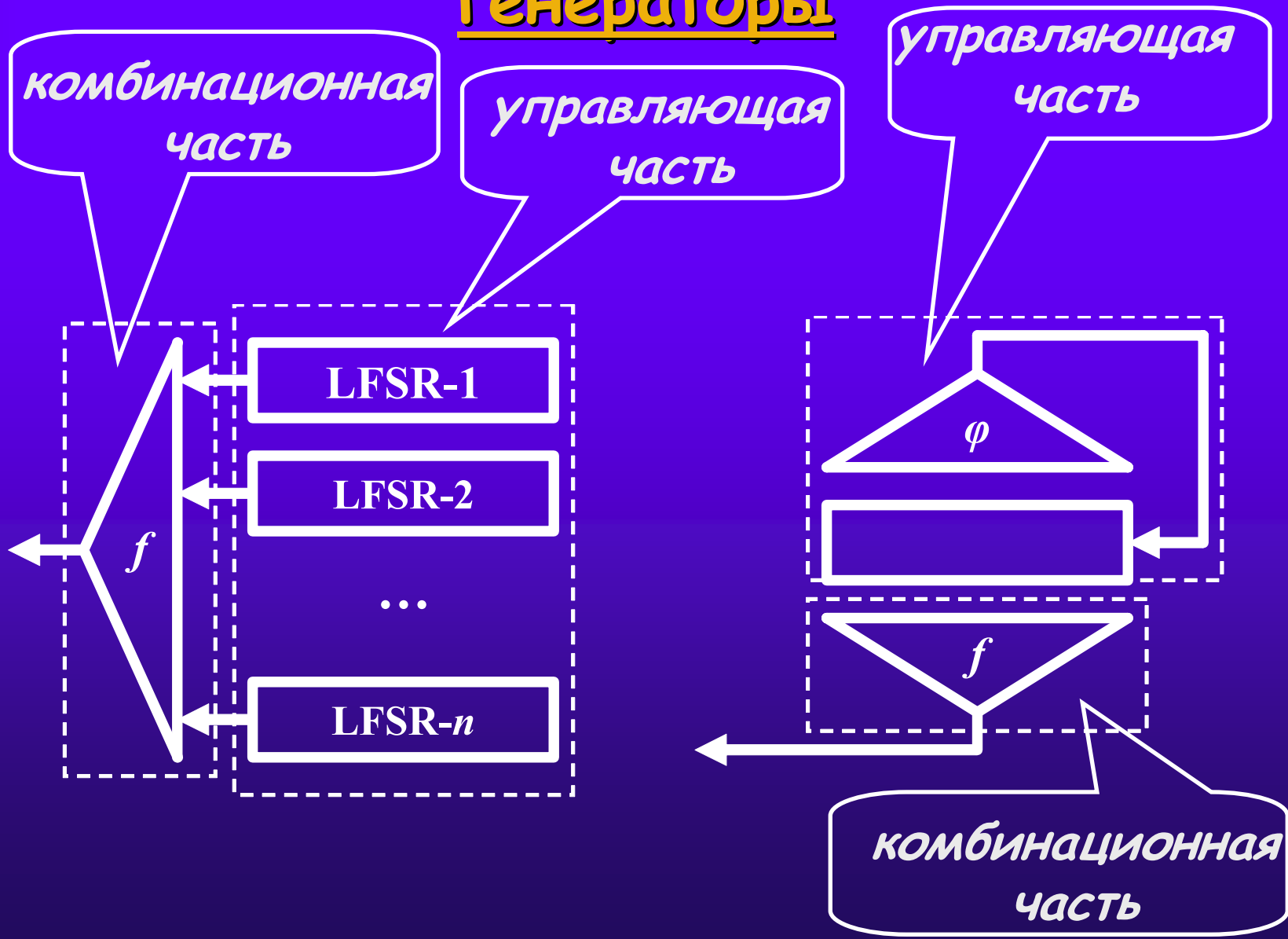
Комбинирующие и фильтрующие генераторы



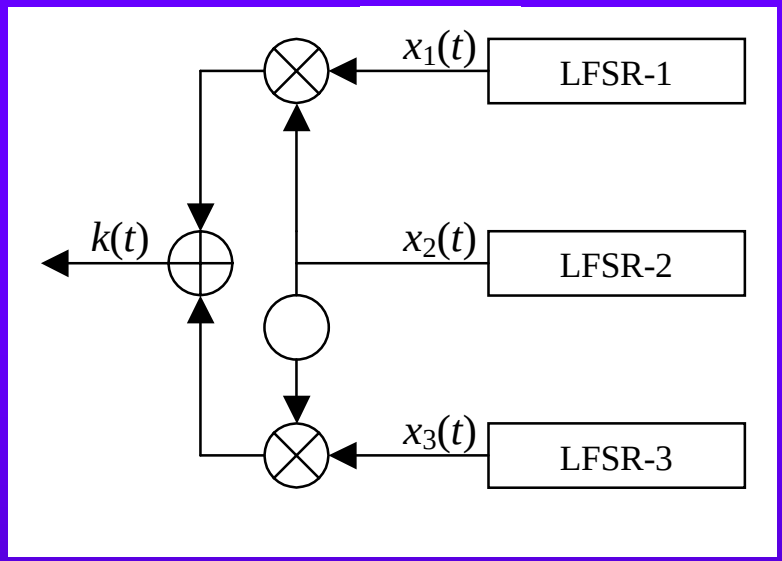
Комбинирующий генератор

Фильтрующий генератор

Комбинирующие и фильтрующие генераторы

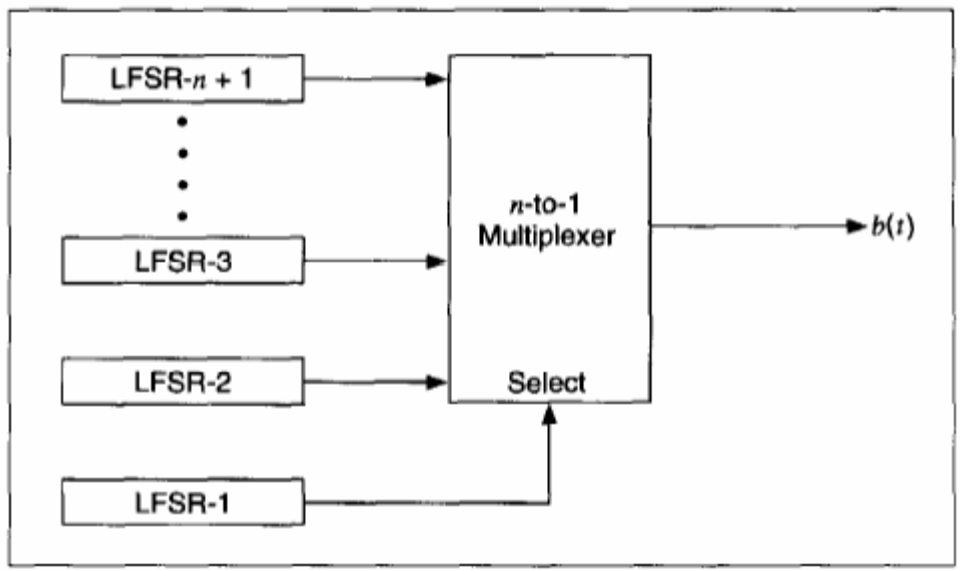


Генератор Геффе (Geffe) - 1973



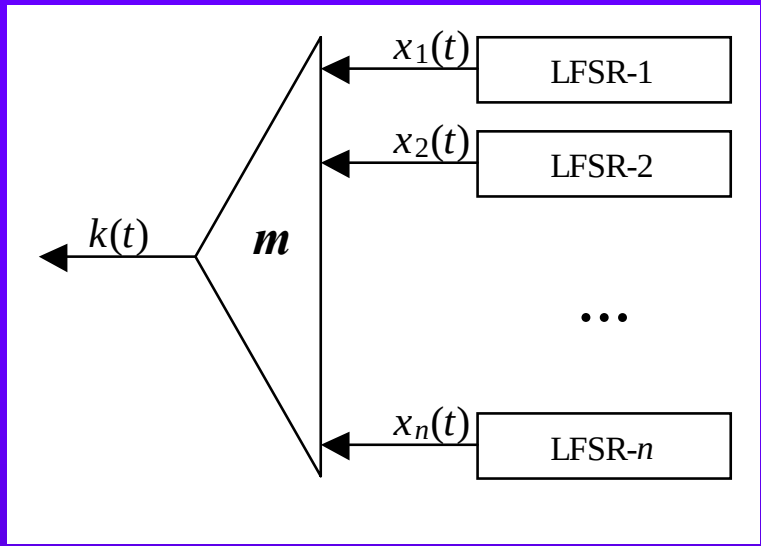
$$T_k = \text{НОК}(T_1, T_2, T_3) = (2^{L_1}-1)(2^{L_2}-1)(2^{L_3}-1)$$

$$\Lambda(k) = L_1 L_2 + L_2 L_3 + L_3$$



Generalized Geffe generator.

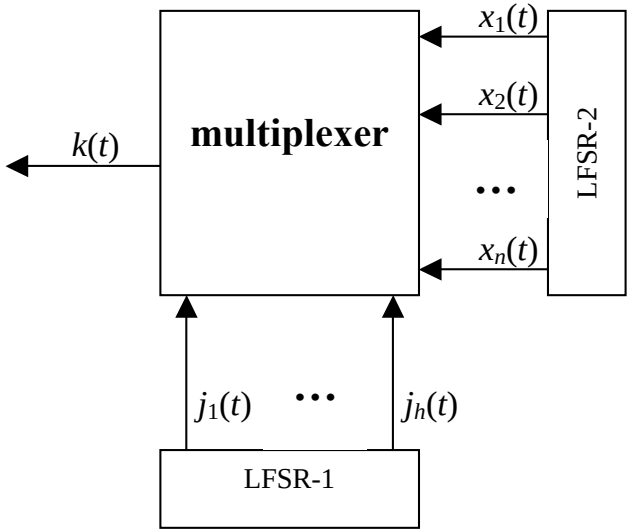
Пороговый (threshold) генератор (1984)



$$T_k = \text{НОК}(T_1, T_2, T_3) = (2^{L_1}-1)(2^{L_2}-1)(2^{L_3}-1)$$

$$\Lambda(k) = L_1 L_2 + L_1 L_3 + L_2 L_3$$

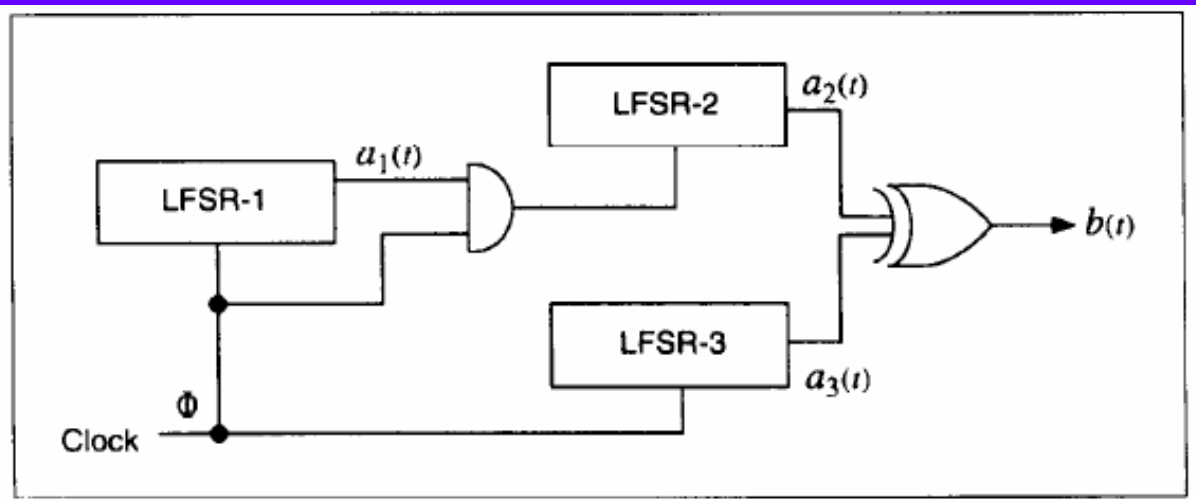
Генератор Дженнингса (Jennings) или мультиплексорный генератор (1982)



$$T_k = (2^{L_1}-1)(2^{L_2}-1)$$

$$\Lambda(\vec{z}) \le L_2\left(1 + \sum_{i=1}^{h} \binom{L_1}{i}\right)$$

Генераторы с неравномерным движением (Stop-and-go generators)



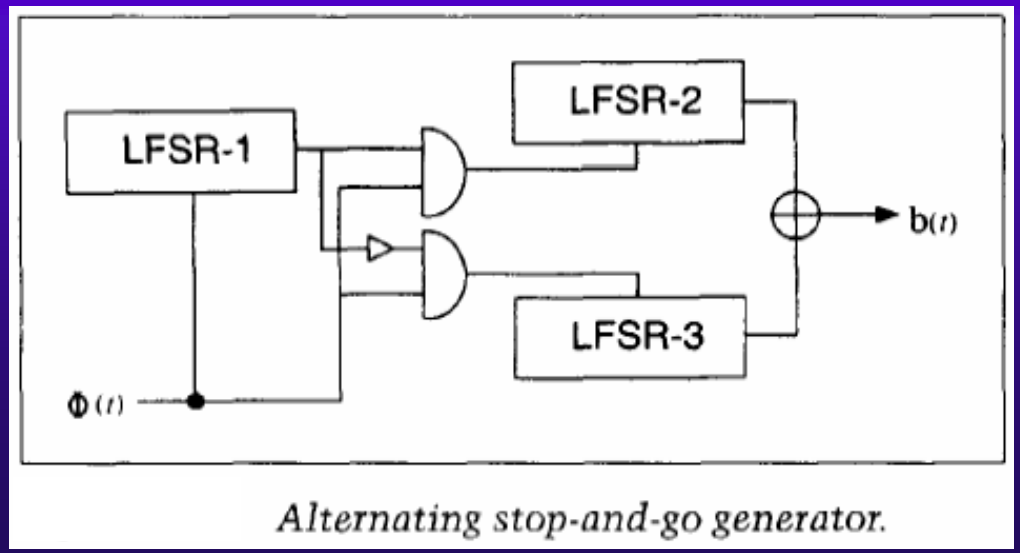
Beth-Piper stop-and-go generator.

1984

1987

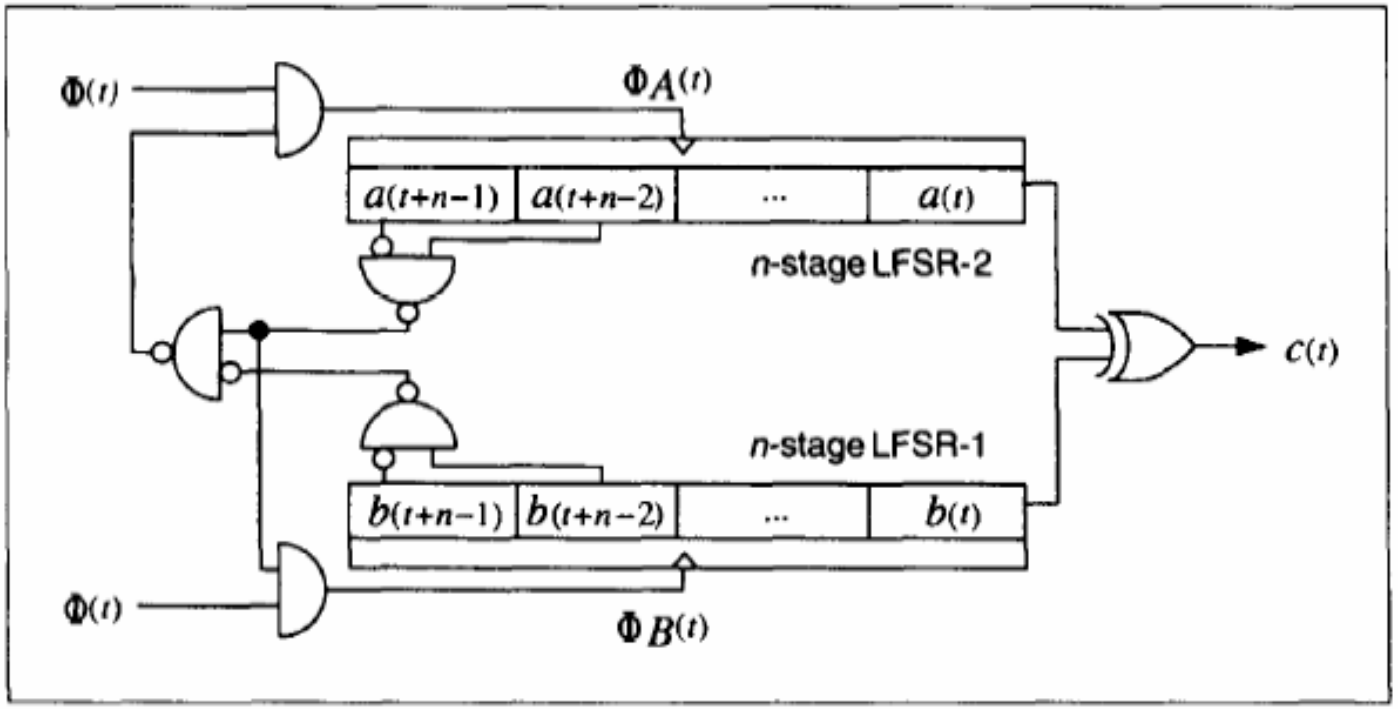
$$T_k = 2^{L_1}(2^{L_2}-1)(2^{L_3}-1)$$

$$2^{L_1-1}(L_2+L_3) \leqslant \leqslant \Lambda(k) \leqslant \leqslant 2^{L_1}(L_2+L_3)$$



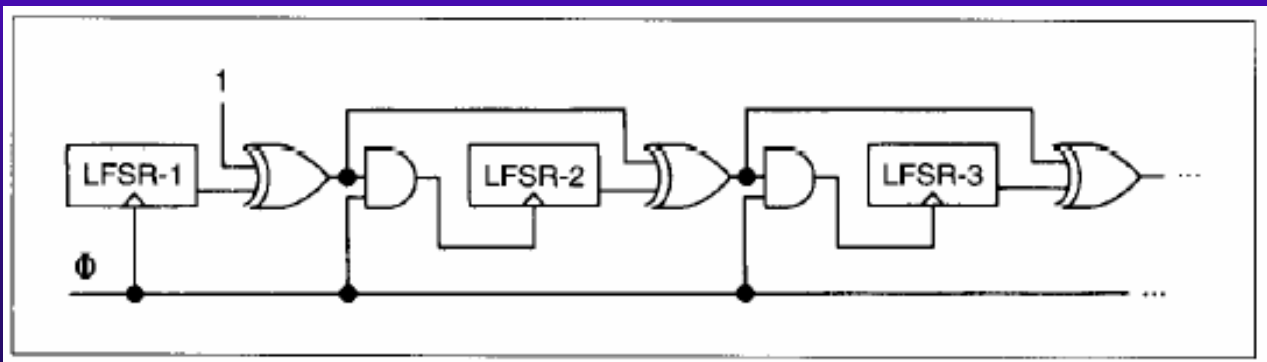
Alternating stop-and-go generator.

1989



Bilateral stop-and-go generator.

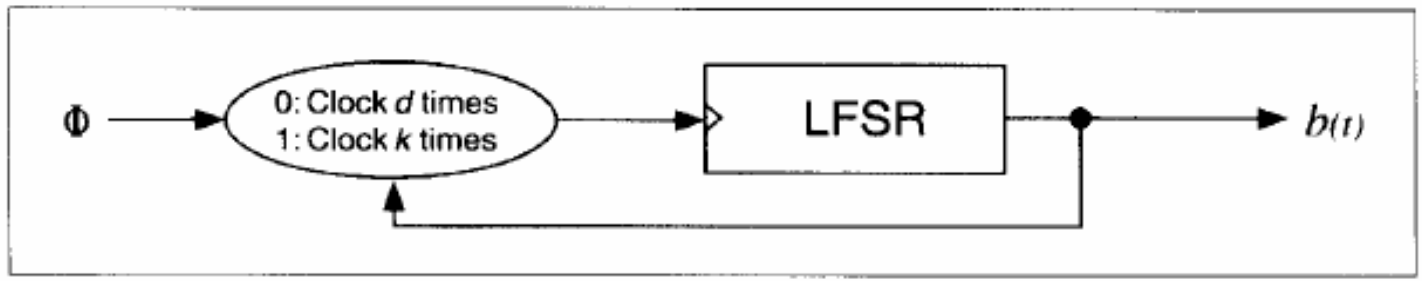
1988



Gollmann cascade.

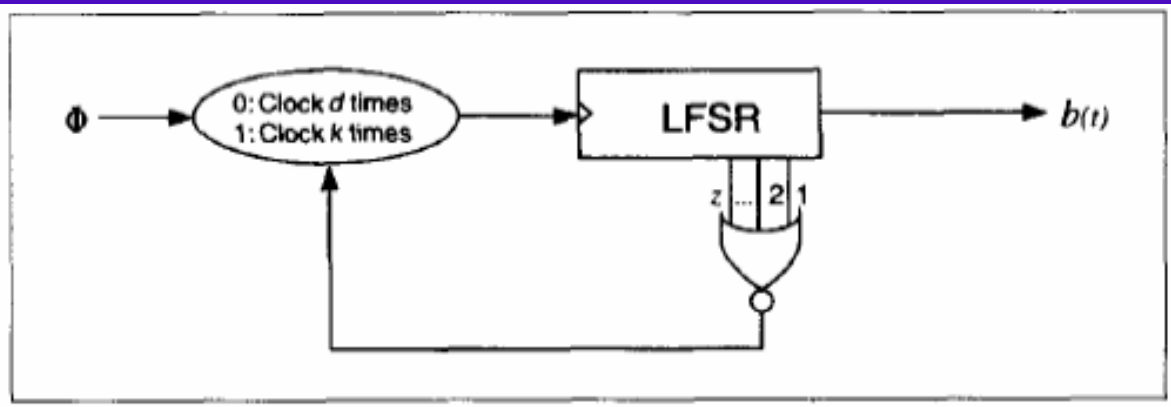
Самопрореживающиеся генераторы (self-decimated generators)

1987



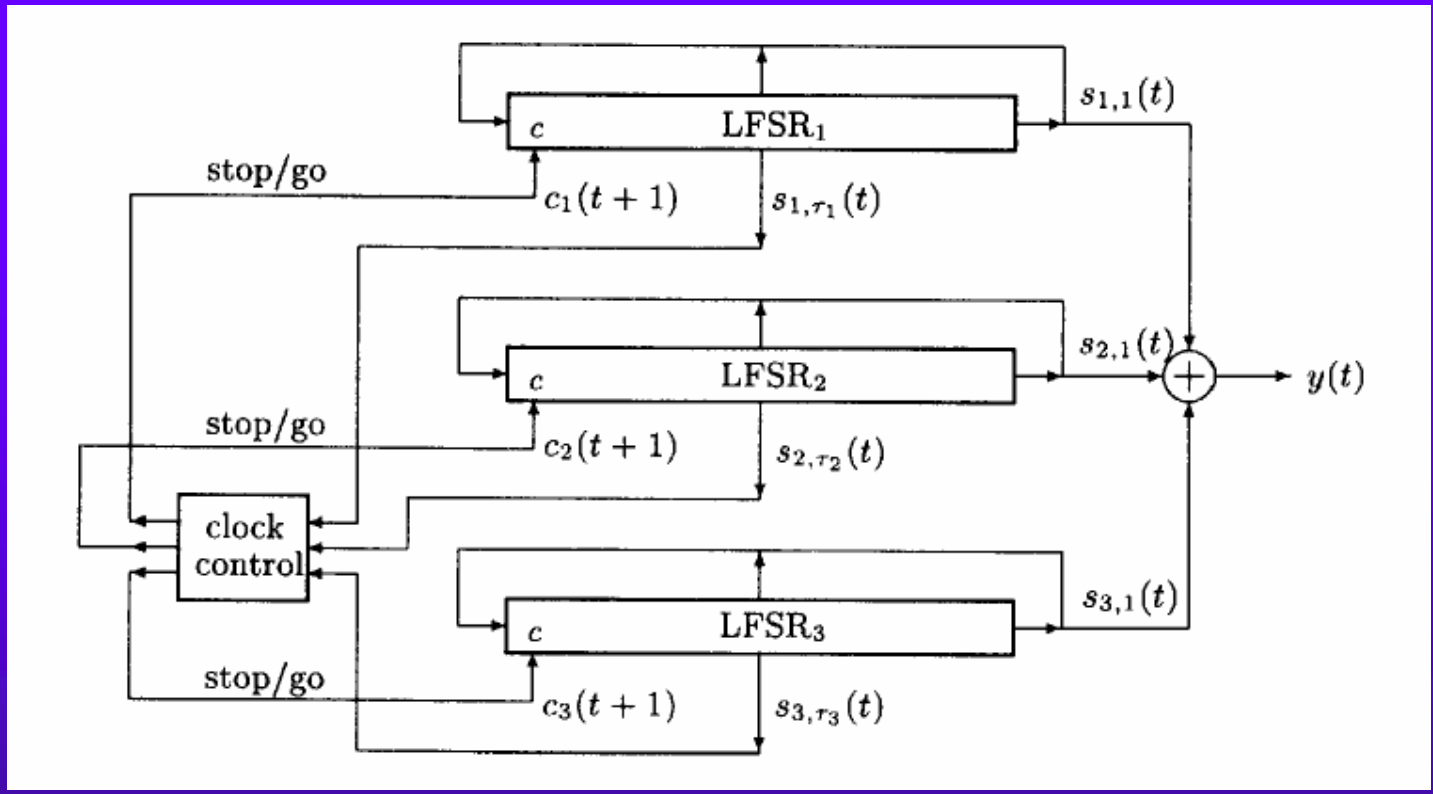
Rueppel's self-decimated generator.

1988



Chambers's and Gollmann's self-decimated generator.

A5 GSM (Group Special Mobile)-1988



A5 состоит из трех LFSR. Длины регистров равны, соответственно, 19, 22 и 23. Все многочлены обратных связей разреженны. Выходом является сумма по модулю 2 выходов всех трех LFSR. A5 использует следующую схему движения: каждый регистр сдвигается по значению своего среднего бита, сложенного по модулю 2 с инверсной мажоритарной функцией от значений средних бит всех трех регистров. Как правило сдвигаются два LFSR из трех.

RC4

Ron Rivest (1987)

RSA Data Security, Inc.

Начальная загрузка:

$S_0 = 0, S_1 = 1, \ldots, S_{255} = 255$

for $i = 0$ *to* 255:

$j = (j + S_i + K_i) \bmod 256$

swap S_i *and* S_j

Выработка ключевого потока:

$i = (i+1) \bmod 256$

$j = (j + S_i) \bmod 256$

swap S_i *and* S_j

$t = (S_i + S_j) \bmod 256$

$K = S_t$

NESSIE project (2000-2003)

NESSIE - New European Schemes for Signature, Integrity and Encryption

http://www.cryptonessie.org.

There are 26 symmetric algorithms:

- 17 block ciphers, are divided as follows:
  - six 64-bit block ciphers: CS-Cipher, Hierocrypt-L1, IDEA, Khazad, MISTY1, Nimbus;
  - seven 128-bit block ciphers: Anubis, Camellia, Grand Cru, Hierocrypt-3, Noekeon, Q, SC2000;
  - one 160-bit block cipher: Shacal;
  - three block ciphers with a variable block length: NUSH (64, 128, and 256 bits), RC6 (at least 128 bits), and SAFER++ (64 and 128 bits).

NESSIE project

- *6 synchronous stream ciphers*:
  BMGL, Leviathan, LILI-128, SNOW, SOBER-t16, SOBER-t32.

- 2 MAC algorithms: Two-Track-MAC and UMAC;
- 1 collision-resistant hash function: Whirlpool.

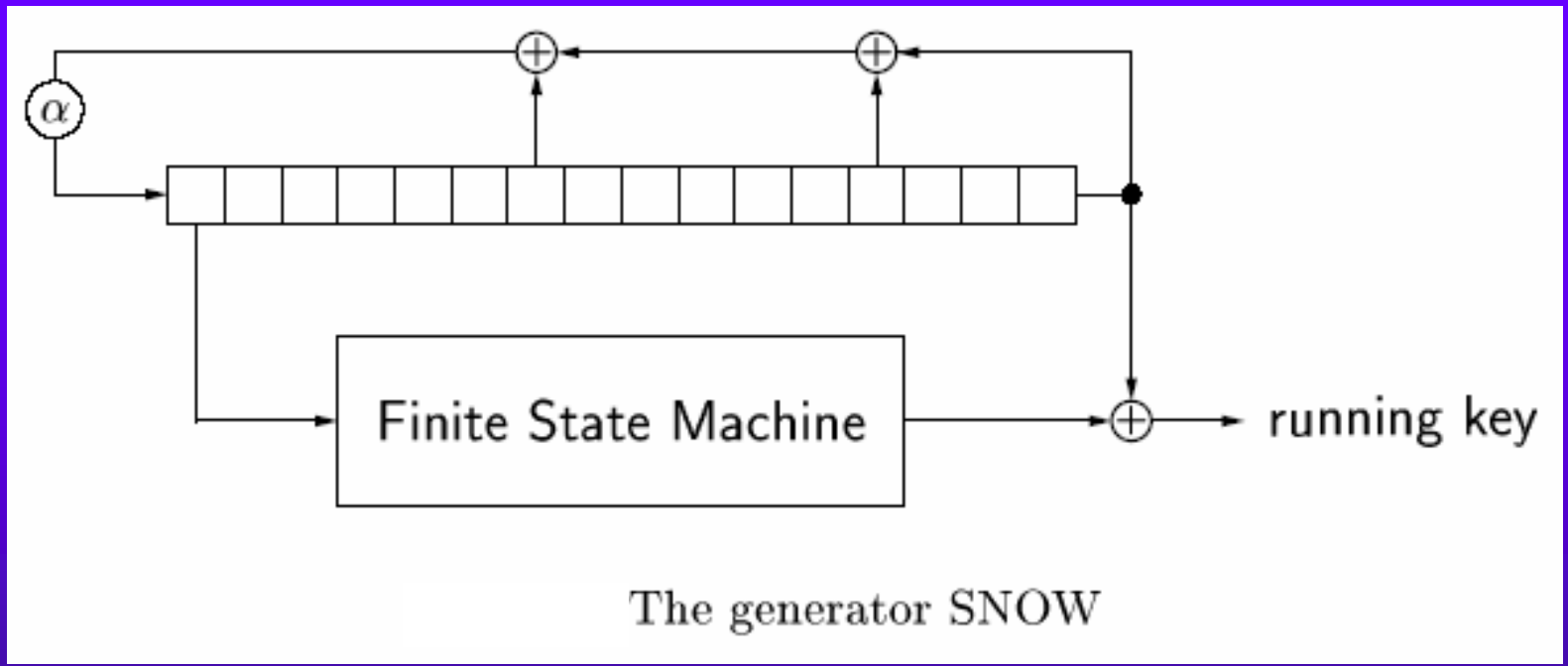
13 asymmetric algorithms have been submitted:

- 5 asymmetric encryption schemes: ACE Encrypt, ECIES, EPOC, PSEC, and RSAOAEP (both EPOC and PSEC have three variants);
- 7 digital signature algorithms: ACE Sign, ECDSA, ESIGN, FLASH, QUARTZ, RSA-PSS, SFLASH;
- 1 identification scheme: GPS.

SNOW

(Patrik Ekdahl, Thomas Johansson)

(Lund University, Sweden)

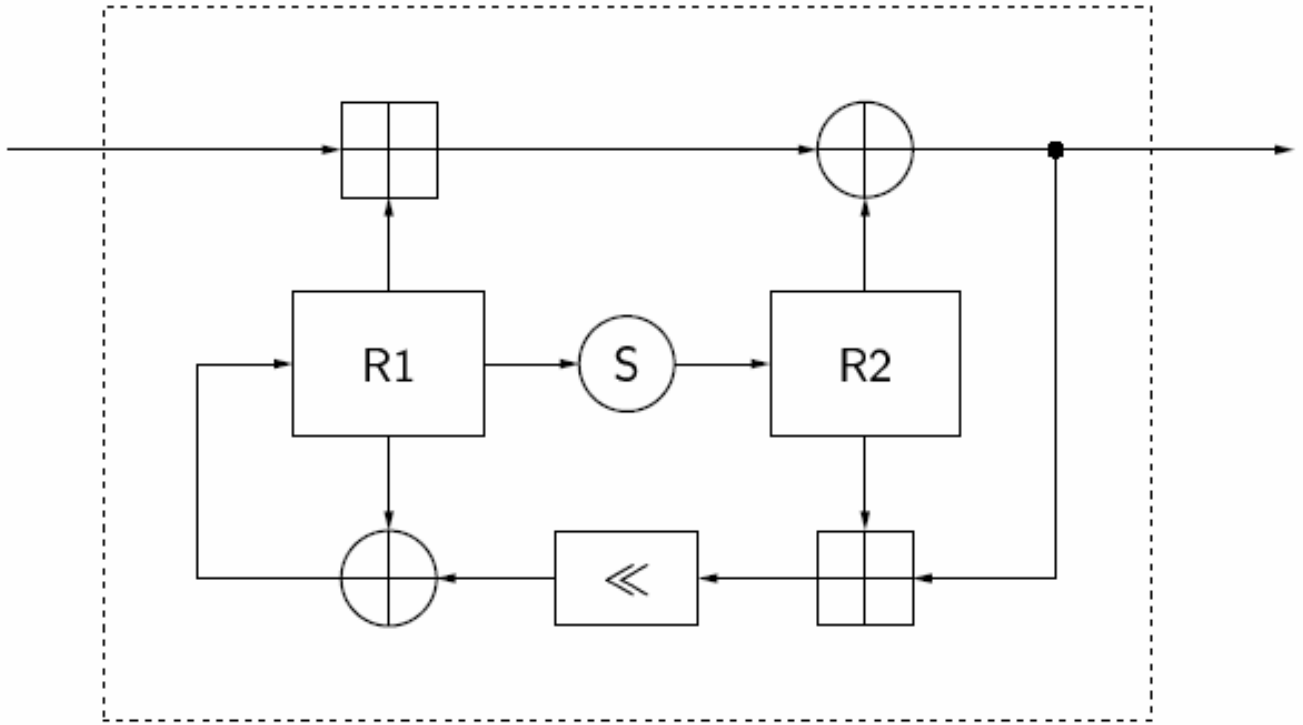


The generator SNOW

Оперирует 32-битными словами, которые рассматриваются как элементы поля GF(2^{32}) или кольца Z_2^{32}. Поле GF(2^{32}), используемое в алгоритме, строится по неприводимому многочлену $\pi(x)$ степени 32 над полем GF(2): $\pi(x) = x^{32}+x^{31}+x^{22}+x^{17}+x^{12}+x^3+1$

Характеристическим многочленом LFSR является примитивный многочлен над полем GF(2^{32}): $P(x) = x^{16}+x^{13}+x^7+\alpha$, где через α обозначен примитивный элемент поля – корень многочлена $\pi(x)$.

Конечный автомат



= addition mod 2^{32}

= bitwise XOR

≪ = cyclic shift 7 steps left

S = S-box

The Finite State Machine

S-блок (S-box)

$$F: (\mathbb{Z}_2)^n \rightarrow (\mathbb{Z}_2)^m$$

$f_1 (x_1, x_2, \ldots, x_n)$

$f_2 (x_1, x_2, \ldots, x_n)$

$\ldots$

$f_m (x_1, x_2, \ldots, x_n)$

$(x_2\, x_1\, x_0)$	$(y_2\, y_1\, y_0)$
0 0 0	0 1 1
0 0 1	1 0 0
0 1 0	1 0 1
0 1 1	0 0 1
1 0 0	1 1 1
1 0 1	0 1 0
1 1 0	0 0 0

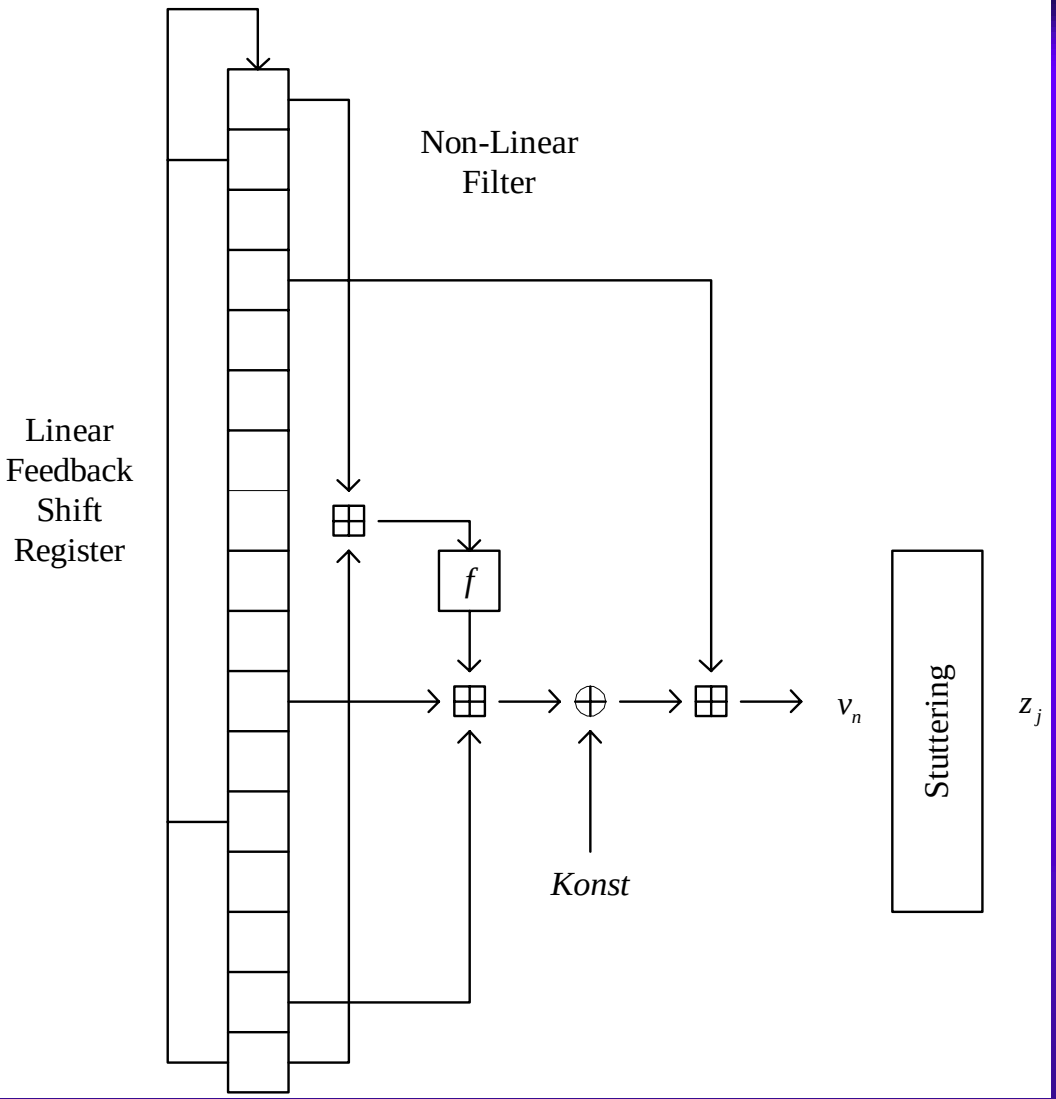
Скорость программной реализации

Неоптимизированная программа на языке C показала следующие скорости работы на разных платформах:

Operating system/ Compiler	*Processor/Speed*		
	Pentium II 400MHz	Pentium III 500MHz	UltraSPARC 400Mhz
Linux/gcc	380	610	-
WinNT/Microsoft C++	430	520	-
Solaris 2.7/gcc	450	-	430

Encryption speed (Mbits/s) on various platforms.

Время инициализации алгоритма равняется 7 μs на Pentium III 500 MHz и может быть снижено до 4 μs.



SOBER

(Greg Rose, Philip Hawkes)

(Qualcomm International, Australia).

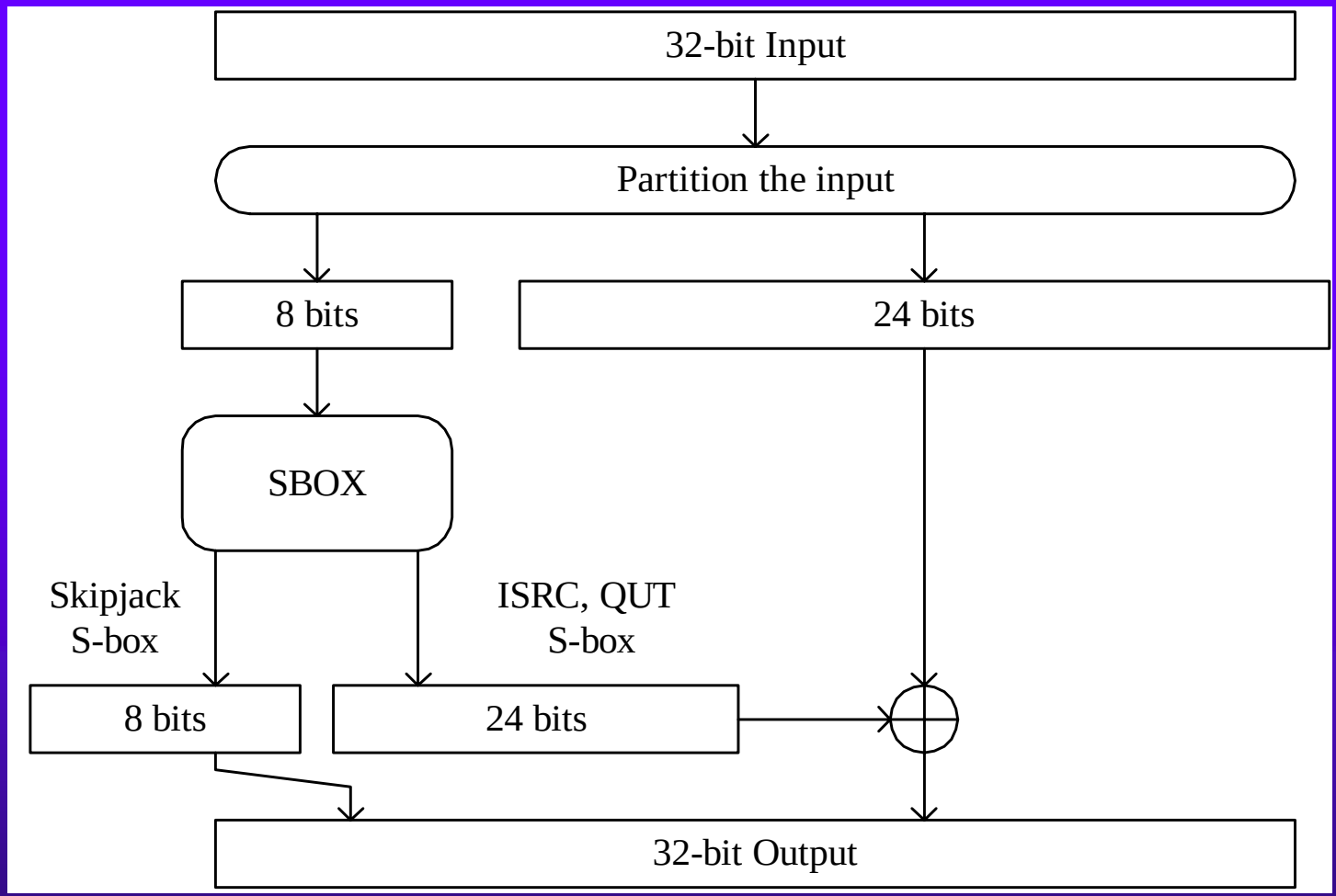
В качестве порождающего многочлена для поля $GF(2^{32})$ выбран неприводимый многочлен

$$x^{32} + (x^{24} + x^{16} + x^8 + 1)(x^6 + x^5 + x^2 + 1).$$

Линейный регистр сдвига вырабатывает последовательность:

$$s_{t+17} = s_{t+15} \oplus s_{t+4} \oplus \alpha\, s_t, \text{ где } \alpha = 0\text{xC2DB2AA3}\ .$$

Нелинейная функция *f*



На вход SBOX подается 8 бит. Первые 8 старших бит выхода совпадают с 8 битами выхода S-box алгоритма Skipjack. 24 младших бит выхода формирует 8×24 S-box ISRC.

NESSIE project

http://www.cryptonessie.org.

- **Security Evaluation.**

  For each algorithm, a short description is followed by a discussion of the provable security (which security properties are proved under which assumptions) and of the concrete security reduction.

- **Performance Evaluation.**

  A set of four test platforms has been defined on which each candidate may be tested. These platforms are smart cards,32-bit PCs, 64-bit machines, and Field Programmable Gate Arrays (FPGAs).

NESSIE project (2000-2003)

NESSIE - New European Schemes for Signature, Integrity and Encryption

http://www.cryptonessie.org.

- Another conclusion from the NESSIE project is that there is a clear need for a very fast and highly secure stream cipher (as the submitted candidates do not seem to satisfy the requirements).

ECRYPT (2004 - 2008)

eSTREAM - ECRYPT Stream Cipher project

- Algorithm designers are invited to submit new stream cipher proposals to the ECRYPT Stream Cipher project; more information and detailed submission requirements are provided below:

http://www.ecrypt.eu.org

eSTREAM

- ECRYPT Stream Cipher project

(2004 - 2008)

- 14-15.10.2004 – SASC - The State of the Art of Stream Ciphers (Bruges)
- 01.11.2004 – объявление о начале конкурса eSTREAM
- 29.04.2005 – Конец подачи документов на конкурс eSTREAM
- 26-27.05.2005 – SKEW - Symmetric Key Encryption Workshop
- 2-3.02.2006 – SASC 2006 (Leuven) – конец 1-й фазы конкурса
- 01.07.2006 – Начало 2-й фазы конкурса
- 31.01-1.02.2007 – SASC 2007 (Bochum)
- 01.2008 – Итоговый отчет ECRYPT Stream Cipher project

Call for Stream Cipher Primitives

- PROFILE 1. Stream ciphers for software applications with high throughput requirements. A key length of 128 bits must be accommodated. An IV length of at least one of 64 or 128 bits must be accommodated.

- PROFILE 2. Stream ciphers for hardware applications with restricted resources such as limited storage, gate count, or power consumption. A key length of 80 bits must be accommodated. An IV length of at least one of 32 or 64 bits must be accommodated.

Performance Testing

We will test all submissions on the following platforms:

- 32-bit: Intel Pentium4 and PentiumM; AMD Athlon XP and Sempron.
- 64-bit: Alpha; PA-RISC; SPARC; AMD Athlon 64.

Performance Testing

- On x86 platforms, the submissions will be compiled with the GNU, Intel, and Microsoft C Compilers, under various compiler options. On the UNIX machines, the testing framework will use the Compaq C compiler, the HP ANSI C Compiler, or the Sun WorkShop Compiler, in addition to the GNU compiler. The fastest implementation (which produces correct test vectors) will be used when comparing the ciphers.

Performance Testing

We will test all submissions with respect to

- time to encrypt >4 Kbytes measured by encrypting a long stream with calls to the function CRYPT_encrypt_blocks(),
- set-up and time to encrypt 40, 576, and 1500 bytes with calls to the function ECRYPT_encrypt_packet().

Алгоритмы, представленные на конкурс eSTREAM

Профиль 1

- ABC
- CryptMT/ Fubuki
- DICING
- DRAGON
- Frogbit A
- HC-256
- Mir-1
- Py
- Salsa20
- SOSEMANUK

Профиль 1,2

- F-FCSR
- Hermes8
- LEX
- MAG
- NLS A
- Phelix A
- Polar Bear
- POMARANCH
- Rabbit
- SSS A
- TRBDK3 YAEA
- Yamb

Профиль 2

- Achterbahn
- DECIM
- EDON-80
- Grain
- MICKEY
- MICKEY-128
- MOSQUITO
- SFINKS A
- Trivium
- TSC-3
- VEST A
- WG
- ZK-Crypt

Скорость программной реализации поточных шифров профиля 1 с 256-битным ключом

Участники 3-й фазы по профилю 1:

- CryptMT
- Dragon
- HC
- LEX
- NLS
- Rabbit
- Salsa20
- SOSEMANUK
- AES (OFB)

Исследование алгоритмов профиля 1

- 2000MHz (one of two CPU cores) AMD Athlon 64 X2
- 533MHz (one of two CPUs) Motorola PowerPC G4 7410
- 1300MHz Intel Pentium M
- 900MHz AMD Athlon
- 440MHz (one of two CPUs) HP 9000/785 J5000
- 3000MHz Intel Pentium 4
- 1000MHz (one of two CPUs) Intel Pentium III
- 900MHz Sun UltraSPARC III
- 2800MHz (one of two CPUs) Intel Pentium 4
- 1900MHz Intel Pentium 4
- 400MHz DEC Alpha EV5.6 21164A
- 133MHz Intel Pentium

Исследование алгоритмов профиля 1

- Результаты тестирования приведены для Intel Pentium 4 3000MHz

Шифрование большого объема данных

Алгоритм	Внутр. состояние (в битах)	Скорость (тактов на байт)
Py	4196	3.7
Py6	1124	3.8
HC-256	8396	5.0
Phelix	132	5.6
SOSEMANUK	452	5.7
Dragon	284	12.3
Salsa20	64	13.9
DICING	4396	14.7
Crypt MT	3020	16.1
YAMB	424	16.5
AES (OFB)	260	33.1

Установка ключа и шифрование 40-байтного пакета

Алгоритм	Внутр. состояние (в битах)	Скорость (тактов на байт)
Phelix	132	33.5
Salsa20	64	44.9
AES (OFB)	260	51.6
SOSEMANUK	452	64.2
Dragon	284	88.4
Py6	1124	117.3
Py	4196	320.6
DICING	4396	416.2
Crypt MT	3020	1066.9
YAMB	424	1227.6
HC-256	8396	2429.0

CryptMT (v.3)

Makoto Matsumoto, Hagita Mariko,
Takuji Nishimura, Mutsuo Saito

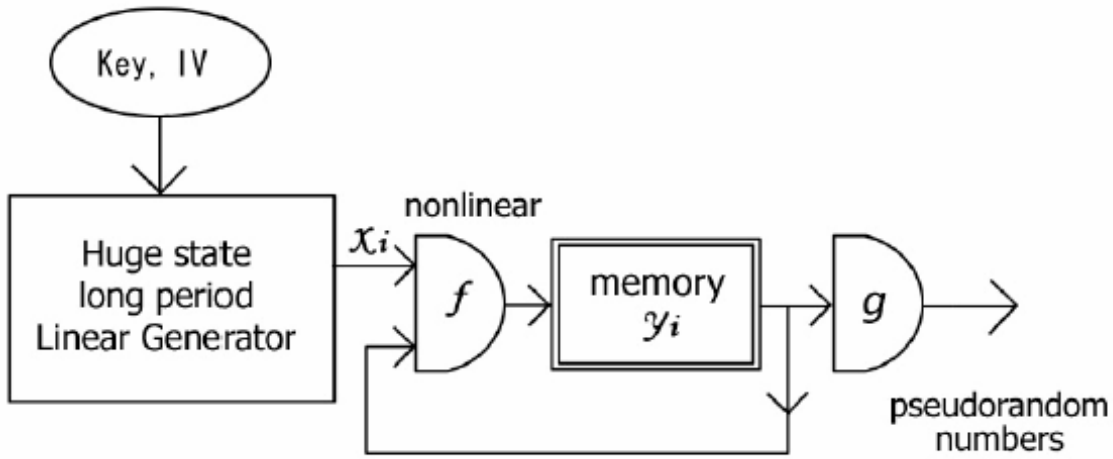


FIGURE 1. Combined generator = linear generator + filter with memory.

CryptMT (v.3)

Makoto Matsumoto, Hagita Mariko,
Takuji Nishimura, Mutsuo Saito

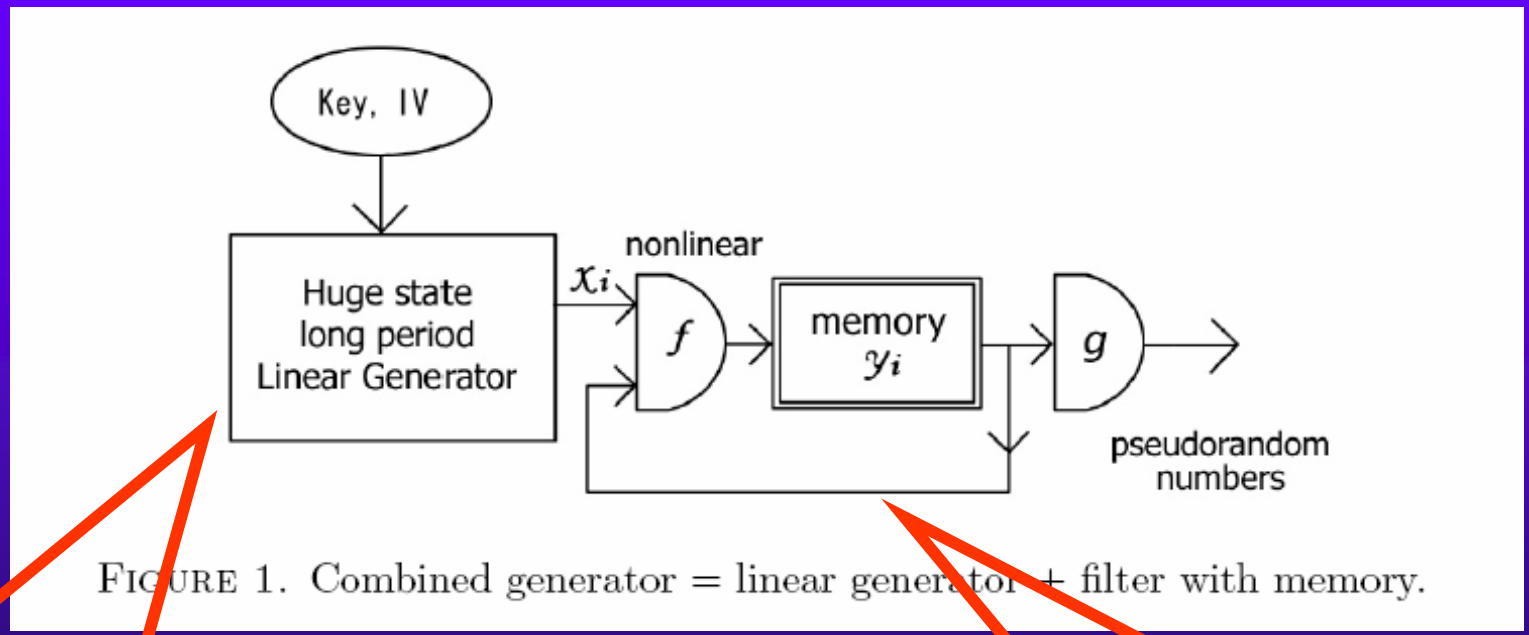


FIGURE 1. Combined generator = linear generator + filter with memory.

Simple Fast Mersenne Twister (SFMT)

Фильтрующая функция

CryptMT (v.3)

Makoto Matsumoto, Hagita Mariko,
Takuji Nishimura, Mutsuo Saito

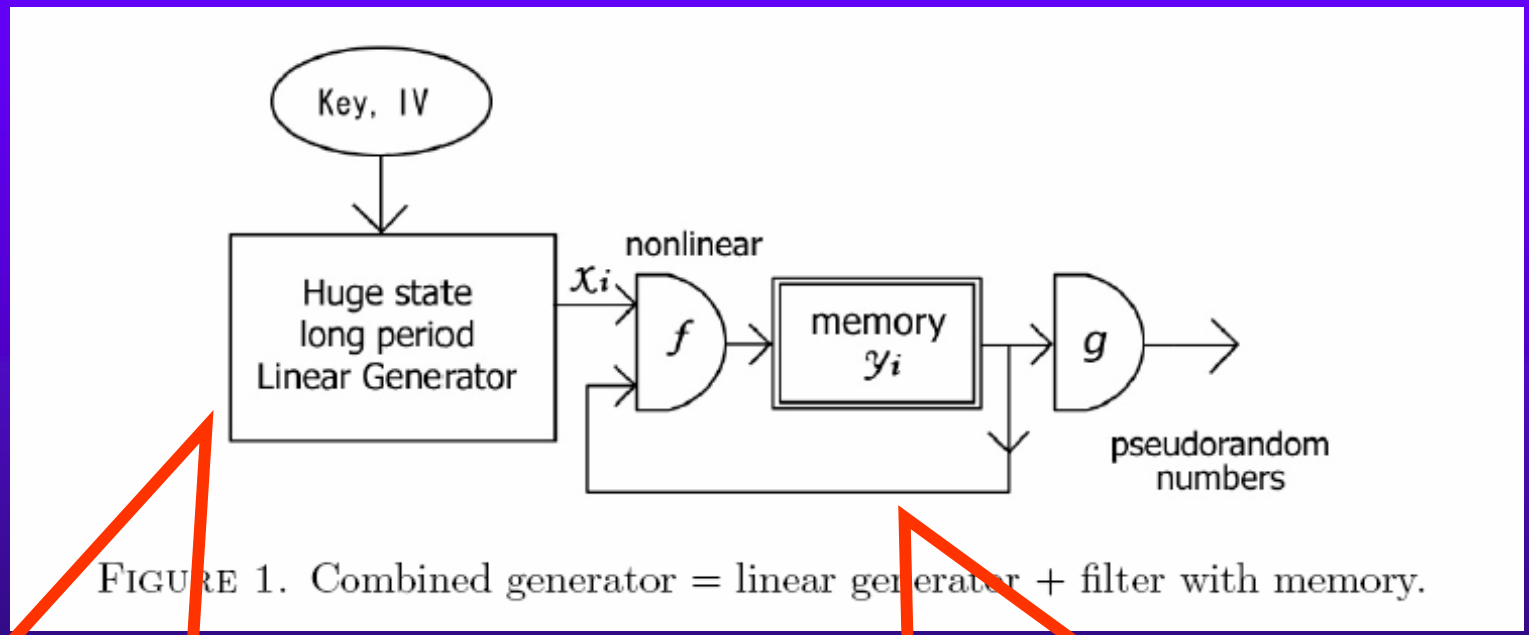


FIGURE 1. Combined generator = linear generator + filter with memory.

Управляющая часть

Комбинационная часть

CryptMT (v.3)

- Simple Fast Mersenne Twister (SFMT)

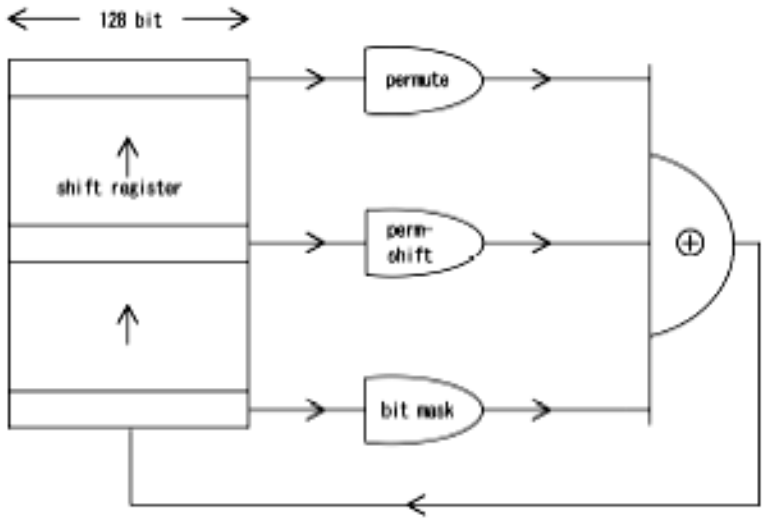


FIGURE 2. The mother generator: Simple Fast Mersenne Twister.
permute: $\mathbf{y} \mapsto \mathbf{y}[0][3][2][1]$.
perm-shift: $\mathbf{y} \mapsto \mathbf{y}[2][0][3][1] \oplus (\mathbf{y} >>_{64} 3)$.
bit-mask: ffdfafdf f5dabfff ffdbffff ef7bffff

$$x_i \in \mathbb{Z}_2^{128}$$

$$N = 156$$

$$M = 108$$

$$S = 3$$

$$x_{N+j} = (x_{N+j-1} \,\&\, MASK) \oplus (x_{M+j} >>_{64} S) \oplus$$
$$\oplus\, x_{M+j}[2][0][3][1] \oplus x_j[0][3][2][1]$$

CryptMT (v.3)

- Фильтрующая функция

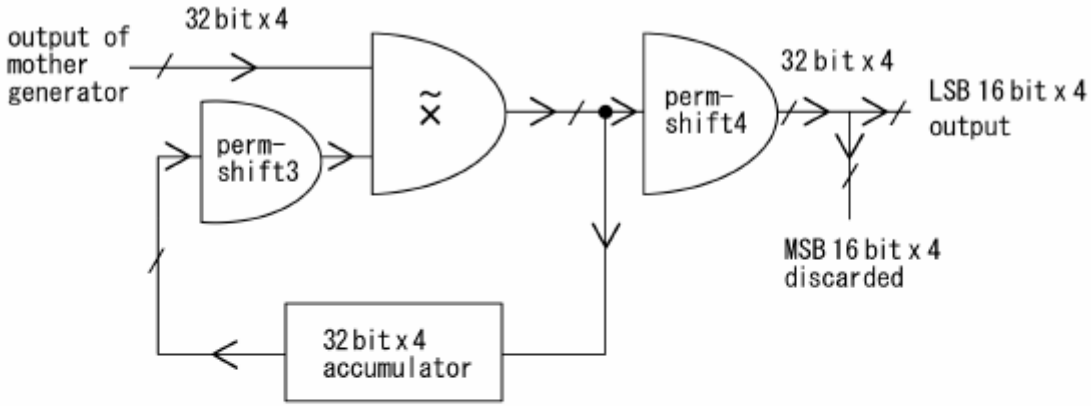


FIGURE 3. Filter of CryptMT Ver.3.
perm-shift3: $\mathbf{y} \mapsto \mathbf{y} \oplus (\mathbf{y}[0][3][2][1] >>_{32} 1)$.
perm-shift4: $\mathbf{y} \mapsto \mathbf{y} \oplus (\mathbf{y} >>_{32} 16)$.
$\tilde{\times}$: multiplication of 33-bit odd integers.

$$x \mathbin{\tilde{\times}} y := 2xy + x + y \pmod{2^{32}}$$

Скорость аппаратной реализации поточных шифров профиля 2 с 256-битным ключом

Участники 3-й фазы по профилю 2:

- DECIM
- Edon80
- F-FCSR
- Grain
- MICKEY
- Moustique
- Pomaranch
- Trivium
- AES

Аппаратная реализация поточных шифров профиля 2

Алгоритм	Тактовая частота [MHz]	Скорость [битов за такт]	Скорость [Gbit/s]	Эффект. в сравнении с AES
Trivium	312	64	18.57	68.30
ZK-Crypt	203	32	6.06	22.61
Grain	300	16	4.48	19.79
VEST	286	16	4.26	5.74
MICKEY	308	1	0.29	1.85
SFINKS+	167	8	1.24	1.82
Moustique	265	3	0.74	1.27
Achterbahn	250	2	0.47	1.08
AES (OFB)	182	3.12	0.53	1.00

КОНЕЦ ФИЛЬМА

Вопросы?