

# Chapter 2

## Types of modern encryption

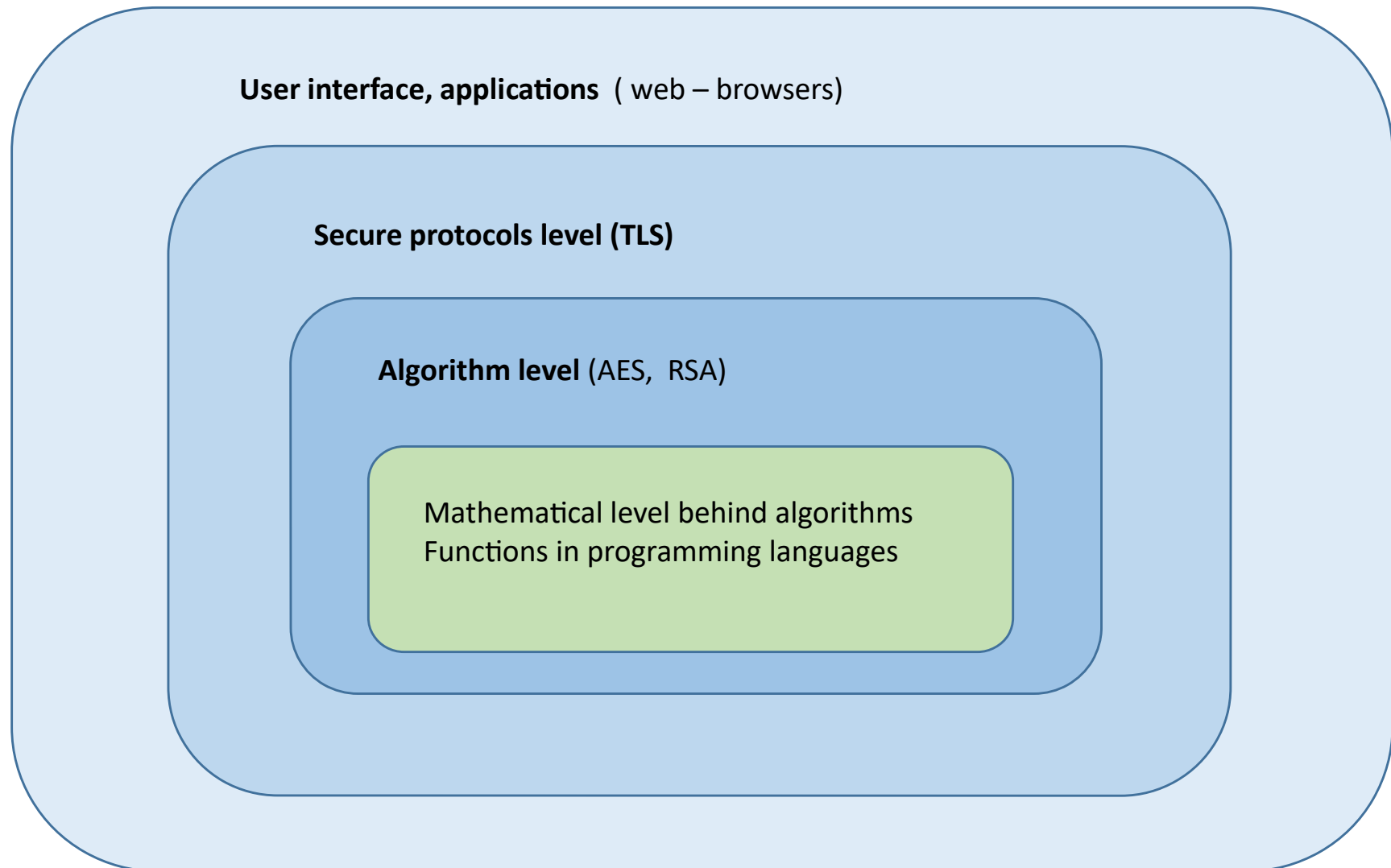
### Secure connections viewed from different levels

- User interface, applications
- Secure protocols level
- Algorithm level
- Level of mathematical structures

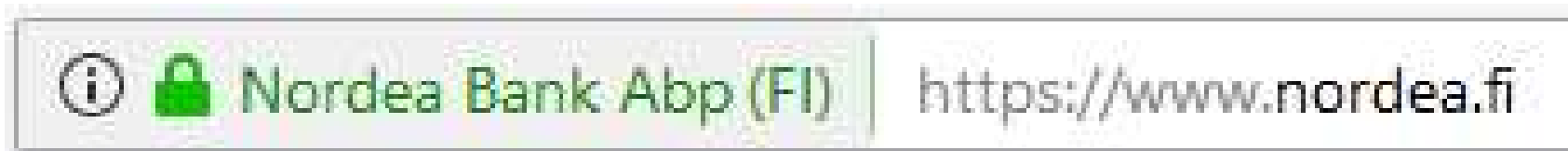
### Block ciphers

### Stream ciphers

# Encryption at different levels



# User level (browsers )



Users know that the connection is secure from the lock of URL row of the browser. If the lock is green, the connection is secure and encrypted and the server is authenticated. Red lock means possible security problems.

The encryption software which provides the secure connection is usually TLS.

# Secure protocols level

In Mozilla Firefox browser clicking the lock reveals following information:

## **Tekniset tiedot**

Yhteys salattu (TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384, 256-bittinen avain, TLS 1.2)

**TLS ver 1.2 is a hybrid encryption software**, which uses several algorithms for different functions. More than 95% of secure connections over Internet use TLS.

The basic functions of a hybrid encryption software are:

1. Authentication
2. Key exchange (agreeing of symmetric key)
3. Encryption of transmitted data
4. Digital signatures

# Algorithmic level

Below there is a list of algorithms used by Nordea Bank's TLS - connection

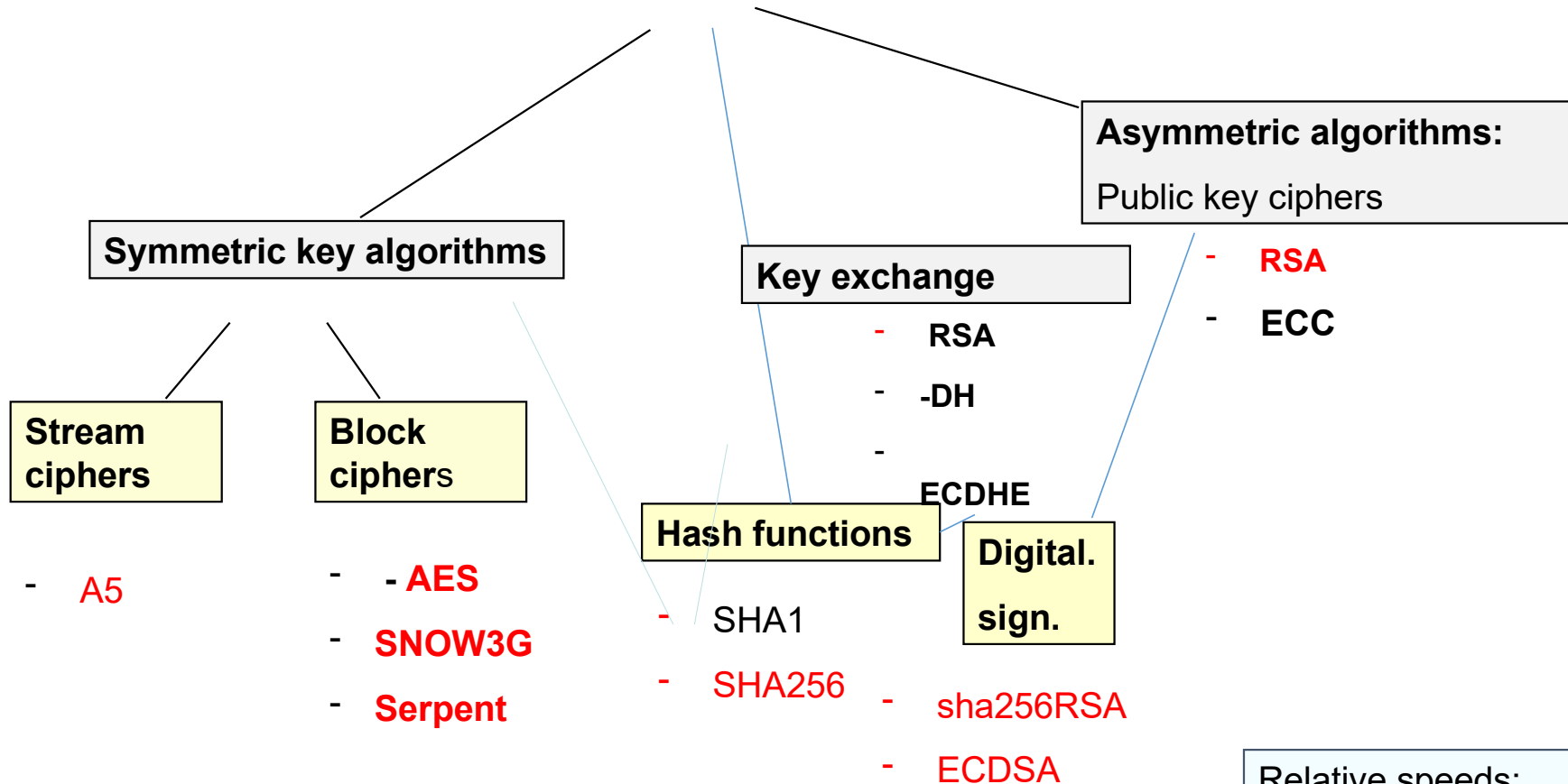
Function	Algorithm name
Server authentication	<b>RSA</b>
Key exchange	<b>ECDHE</b>
Encryption of transmitted data	<b>AES256 GCM-moodi</b>
Digital signature	<b>sha384RSA</b>

# 4. Mathematical level

The algorithms of TLS and other modern software require lots of mathematics

- |   |                                |
|---|--------------------------------|
| 1. Fermat's and Euler's theorems, Extended Euclid's algorithm |                                |
| 2. Random number generation                                   | pseudorandomness               |
| 3. Prime generation, tests                                    | Fermat and Rabin- Miller tests |
| 4. Modular arithmetics  | number sets $Z_n$              |
| 5. Fast exponentiation mod n                                  | Powermod - algorithm           |
| 6. Inverses mod n   | Euclid's extended algorithm    |
| 7. Cyclic group $Z_p^*$                                       | group theory                   |
| 8. Elliptic curves  | group theory                   |

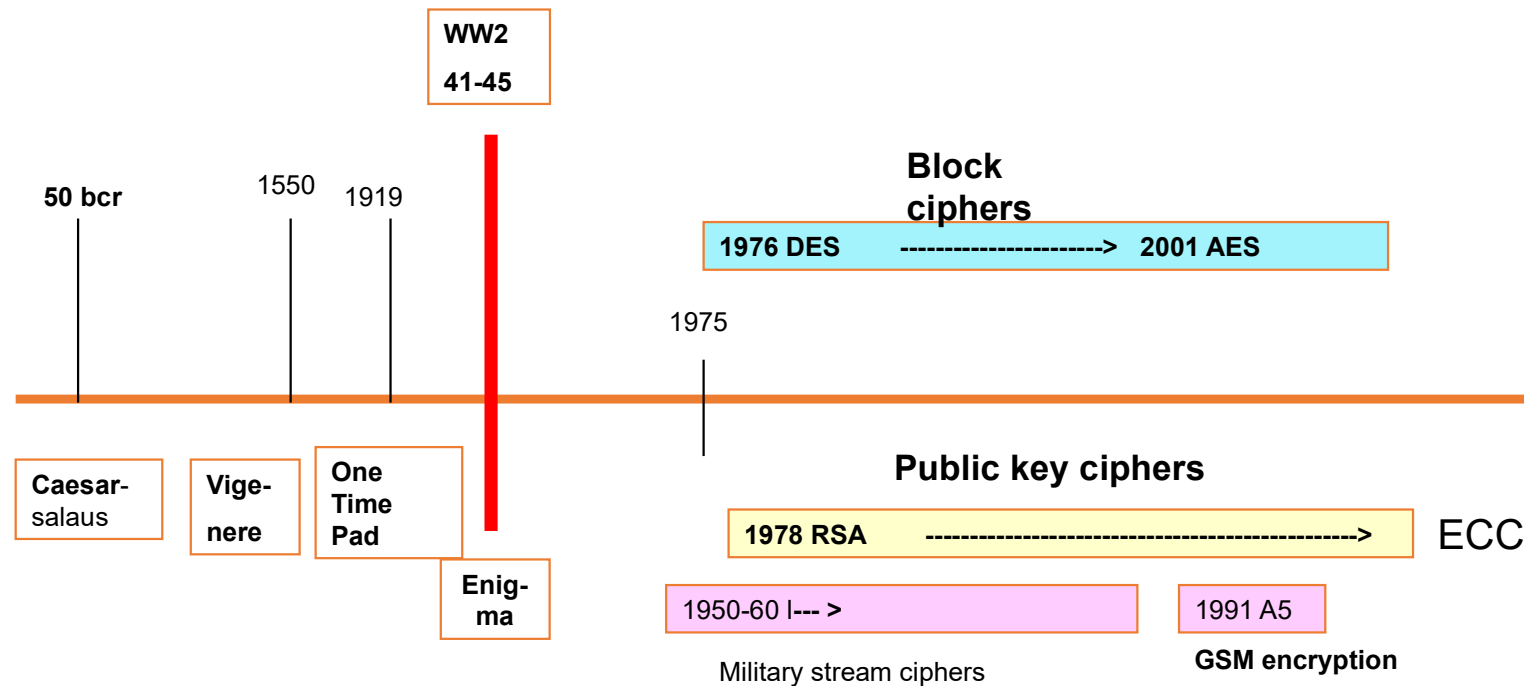
# Algorithm types



Relative speeds:

Hash	3
Stream cipher	2
Block cipher	1
Asymmetric	0.02

# Time line of encryption methods



Algorithms have long lifetimes. RSA has been standard for 45 years in public key cryptography. Block cipher standard has changed only once during 45 years. Reasons for conservatism: 1. Changing algorithms costs money. 2. The security of algorithms are based on mathematical proofs, they have been tested for decennies.



# Symmetric encryption algorithms

## **Types of symmetric algorithms**

### **1. Block ciphers**

- used both in wired and mobile communication
- Diffusion and confusion principles
- Permutation-substitution networks
- Modes of operation
- AES : block cipher standard

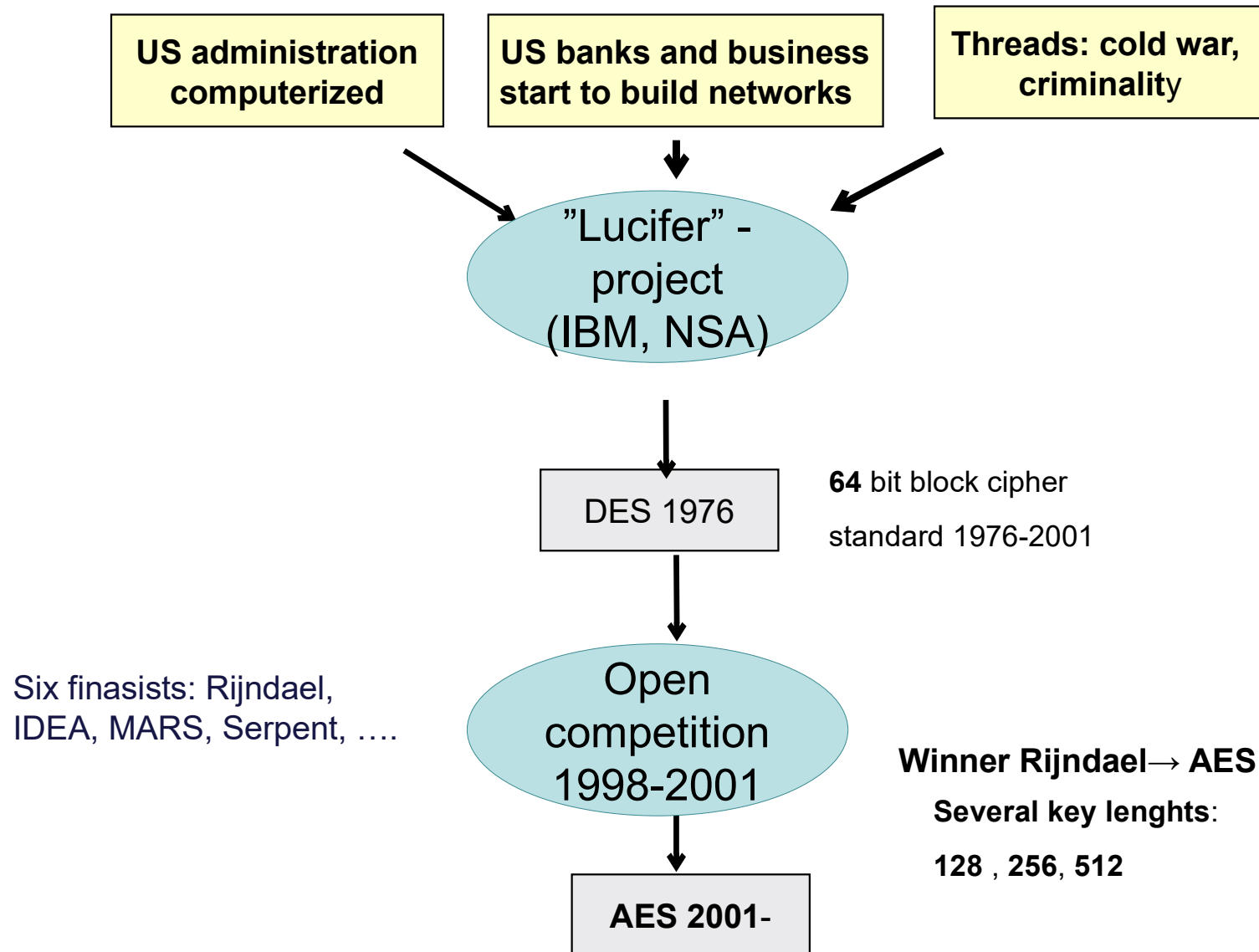
### **2. Synchronic stream ciphers**

- no longer widely used
- GSM (G2)– encryption A5

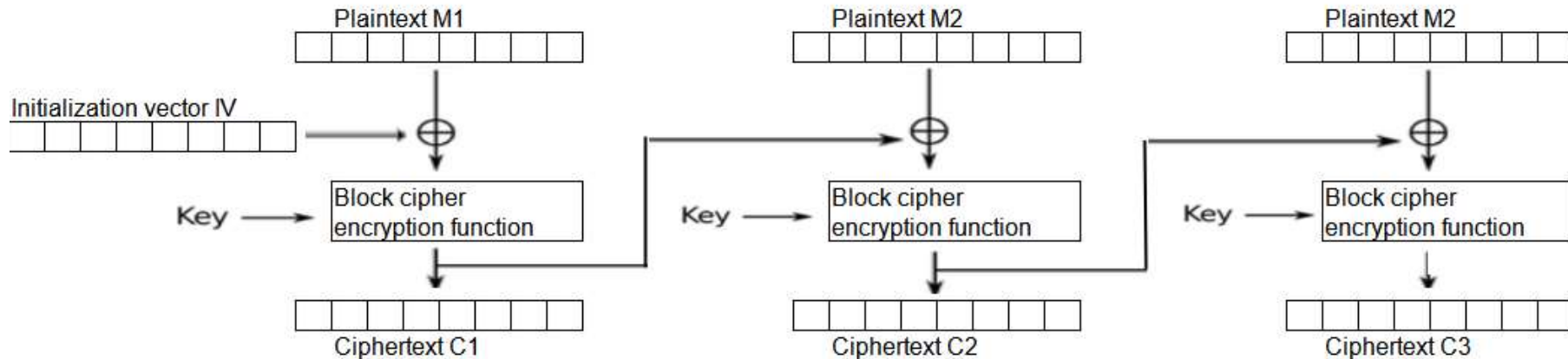
# Block ciphers

- Block ciphers have a central role in secure connections. They encrypt transmitted data reliably and effectively
- Block ciphers provide fast symmetric encryption in which message is encrypted in blocks (usually 128 bits)
- The present block cipher standard is AES (since 2001). It is used both in secure wired connections and in mobile networks

# History of block ciphers



# Diagram shows how block cipher works in CBC mode (CBC = cipher block chaining)



## Principle:

Message  $m$  is divided into 128 bit blocks  $m_1, m_2, m_3, \dots$

Key  $k$  has 128 bits in AES128. Cipher consists of blocks  $c_1, c_2, c_3, \dots$

In CBC mode each output block is taken as 3rd input in calculation of next cipher block.

$$c_n = \text{AES}(m_n, k, c_{n-1})$$

# Requirements for block ciphers

- Block cipher should have both software and hardware (chip) implementations (AES has both)
- Algorithm should be reliable and fast
- It can be applied to both data transfer and files
- Algorithm should be public and transparent (Kerckhoff's principle)
- The key length should be at minimum 128 bits

# Shannon's diffusion and confusion



The operation of block ciphers is based on two principles of Claude Shannon: diffusion and confusion

## **Diffusion: (dependence of cipher and message)**

Changing one bit of message block should change in average half of the output bits. The output bits should equally depend on all input bits.

## **Confusion: (dependence of cipher and key)**

Every bit of cipher should depend on all parts of the key. The change of one bit of the key, should change the output completely, not just part of it.

**The ciphertext should be random and fulfill the properties of pseudorandom bit sequence , which is explained later in this chapter.**

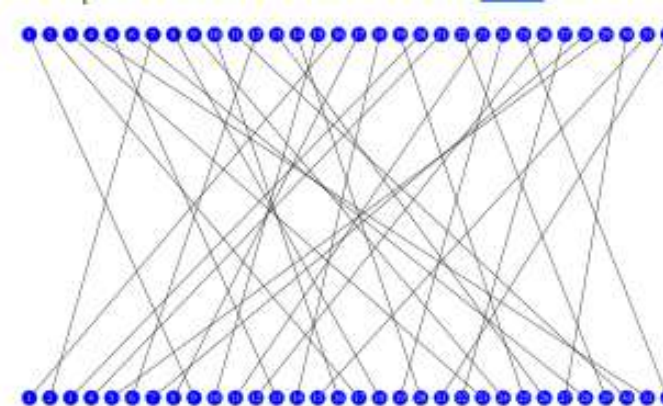
# Permutation – Substitution networks

The principles of diffusion and confusion are usually implemented using a permutation - substitution network.

**Substitution** = replacing certain bit patterns with image patterns, traditionally using substitution tables called SBOX:es. Below is a **table of Sbox nr 6 of DES block cipher**. In the example the bit sequence 011011 is mapped to 1001

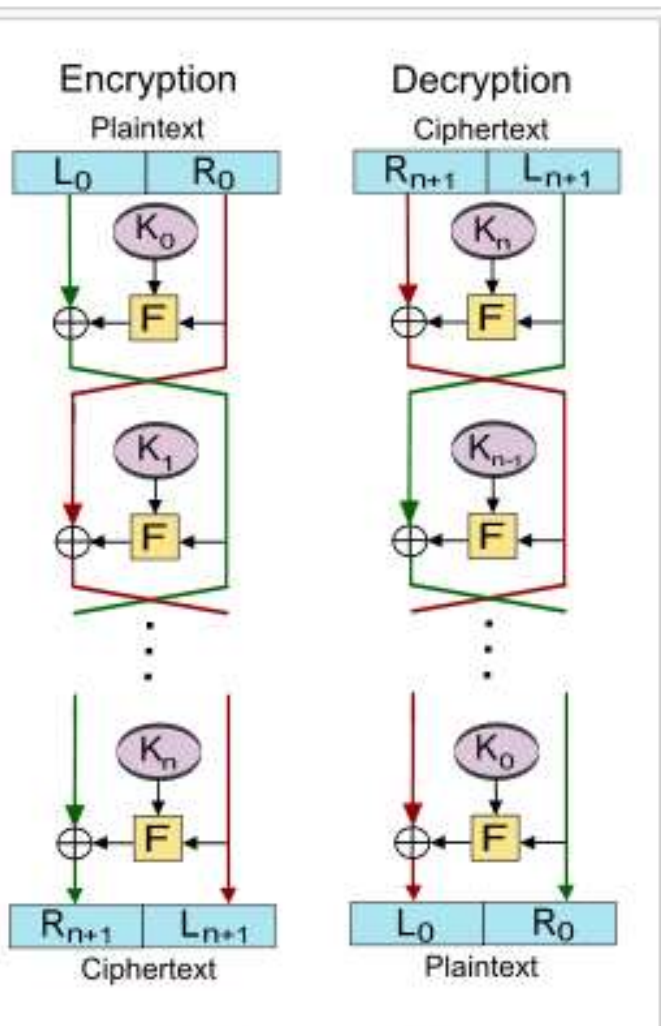
S <sub>5</sub>		Middle 4 bits of input															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

**Permutation** = reordering bits of a block using permutation tables. Figure shows how 64 bits are rearranged in P-box of DES block cipher.



**Rounds:** Permutation- Substitution loop is runned several times, (for example in DES 16 times) in order to remove all regularities in the cipher bit sequence.

# Feistel network



IBM engineer Feistel created a PS network in 1960's using permutations and SBOXES (substitution tables). Feistel network appeared originally in DES block cipher, it was later reused in Blowfish cipher and later in **Kasumi**, which was recently a standard block cipher in 3G and 4G. Today Kasumi is replaced by AES and SNOW3G.

**Feistel algorithm has 16 rounds.** Each round uses a different subkey  $K_i$  derived from the master key  $K$ .

Each round starts with the initial permutation and ends with the final permutation (permutations are called PBOXES).

Between permutations there are eight SBOXES, which obscure the connection between the key and the ciphertext, implementing Shannon's confusion principle.

Many block ciphers, such as DES and Blowfish utilize structures known as *Feistel ciphers*.



# AES encryption standard

In late 1990's NIST organized a competition for a new block cipher standard. There were six finalist, from which the Rijndael algorithm was chosen as the new standard in 2001.

The winner got a new name *Advanced Encryption Standard* AES.

## Advantages of AES

- \* It has many versions AES128, AES196, AES256, AES512. With flexible key lengths it answers the challenges of increasing computing power

- \* The security is based on provable, transparent mathematics

- \* It is generally thought that AES will be used as the standard block cipher for a long time in the foreseeable future.

In this chapter we use in total 12 slides to explain how AES works, not only because AES is the standard block cipher, but also because there is a connection to Finland:

**A Finnish cryptographer Kaisa Nyberg**, emerita professor of Aalto University, has contributed to AES algorithm with a method of using theory of **Galois' Field  $GF(2^8)$**  in several key parts of AES.

In the Moodle there are bonus problems related to AES mathematics. They are not compulsory, but they give bonus points to increase the grade.

# AES handles blocks as tables of 16 bytes

**AES** block size is 128 bits. Blocks are arranged to tables of 16 bytes (1 byte = 8 bits), as in the example below.

$$\begin{bmatrix} 12 & 3F & 95 & 8C \\ 35 & 62 & 87 & D3 \\ FA & 33 & 42 & 7E \\ 85 & 32 & 45 & 99 \end{bmatrix}$$

The elements of the table, also called "AES state", are bytes, which are 8 bit binary numbers. Bytes are usually presented in hexadecimal form.

For example byte 8C is in binary form 10001100 , because according to the table below 8 = 1000 and C = 1100.

hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

## AES128 uses a 128 bit key

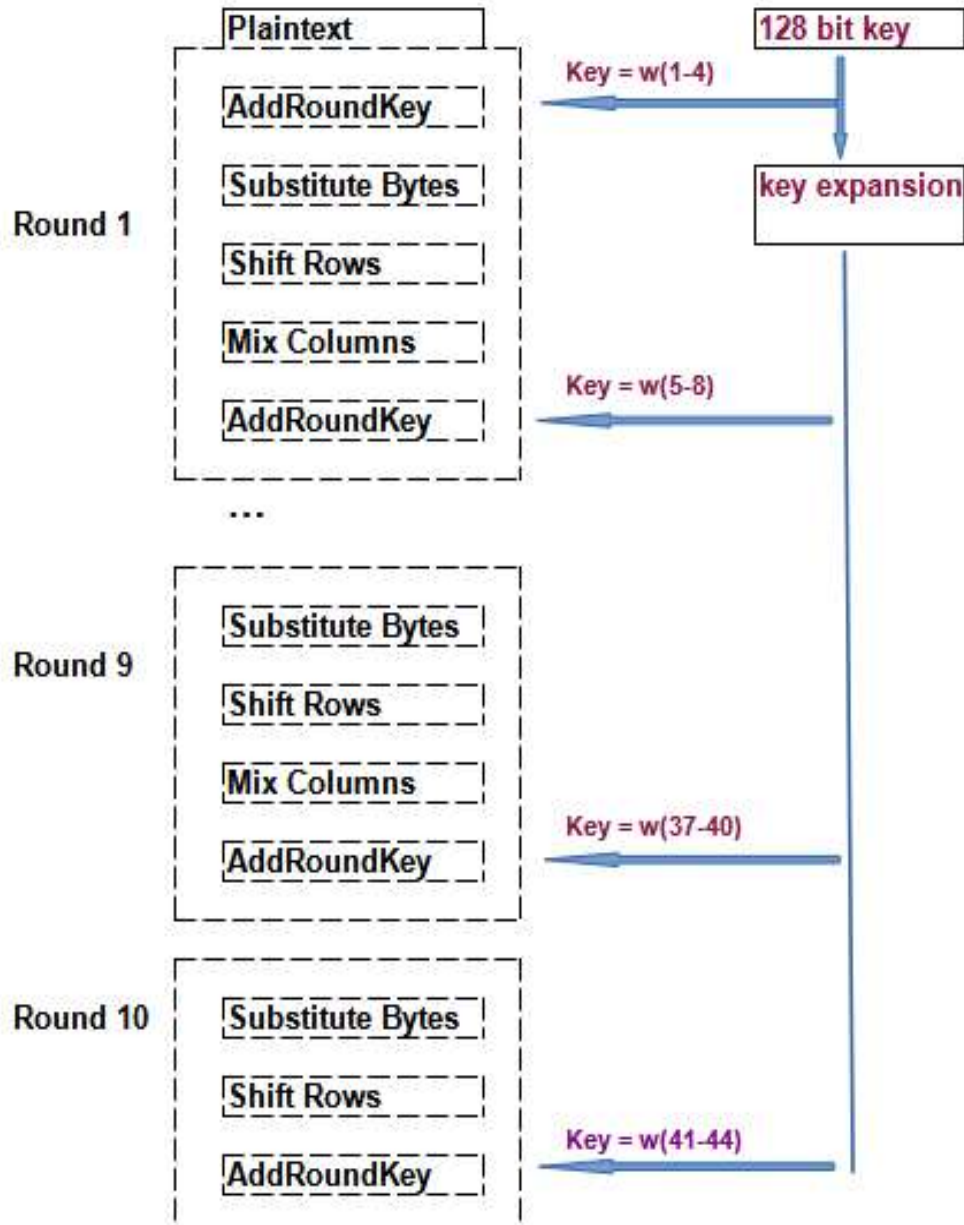
AES128 algorithm has 10 rounds all of which use different "**round keys**" derived from the master key.

Each round consists of several steps, which implement Shannon's diffusion and confusion principles.

Phases of rounds are:

- Substitute bytes (Sbox)
- Shift rows
- Mix columns
- Add round key

The finest part, Sboxes uses Theory of Galois Fields. The algorithm in Sboxes was invented in 1991 by a Finnish cryptographer Kaisa Nyberg.



*Evariste Galois (1811-1832) was a great French mathematician who died at the age of 21*

The key **K** is expanded to 10 Round Keys

## Round Key generation

Each of the ten AES128 rounds has an own round key, which is derived from the 128 bit master key.

**Round Key Generation** is a process, which can be described briefly as follows:

The 128 bit (=16 bytes) master **key K** is expanded to 44 words  $w_1, w_2, \dots, w_{44}$ . (1 word = 4 bytes = 32 bits) usign a specific algorithm. Details of the expansion can be found in internet.

Round1 uses 8 words of the expanded key:  $w_1, \dots, w_4$  in the beginning and  $w_5, \dots, w_8$  in the end of the round.

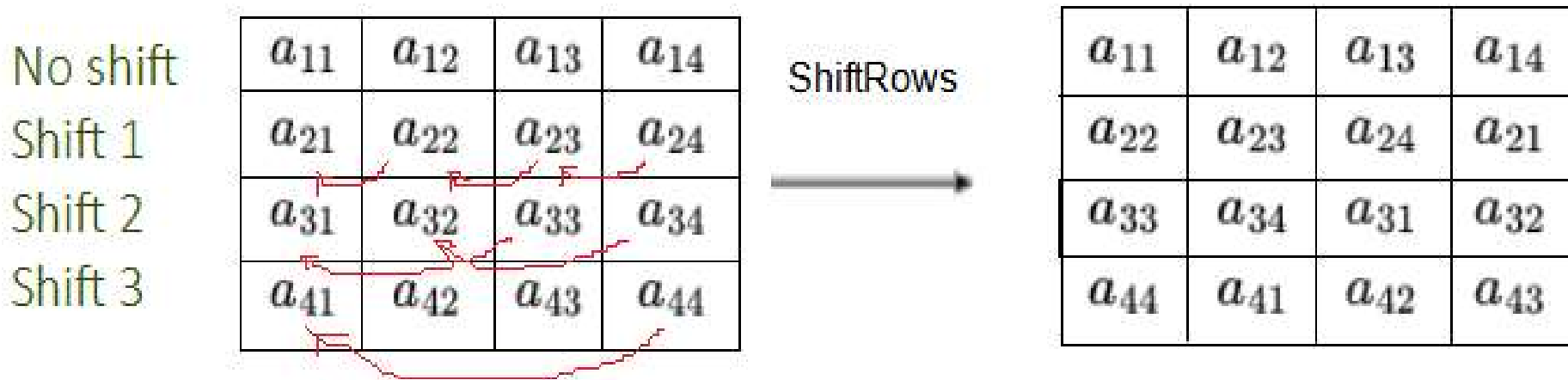
All the other rounds use 4 words each (see the picture in the previous slide)

# Steps of AES rounds

**Notice: All steps must be reversible to make decryption of the cipher possible.**

## ShiftRows step

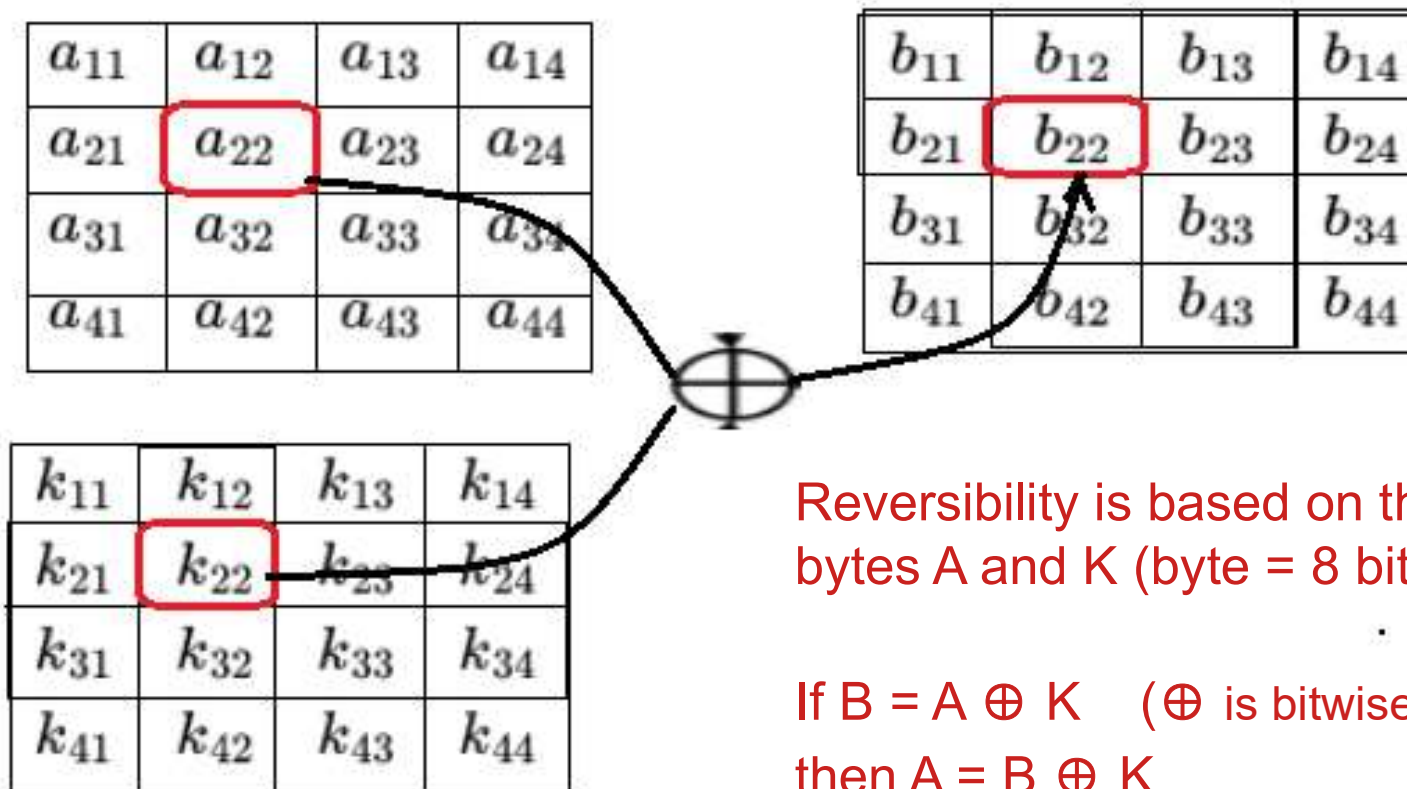
Bytes in each row of the state are shifted cyclically to the left. The amount of shift differs incrementally for each row.



Obviously this step can be reversed by changing the direction of shifts.

## AddRoundKey Step

In AddRoundKey Step each byte of the state is combined with a byte of the Round Key using XOR – addition (symbol  $\oplus$ )



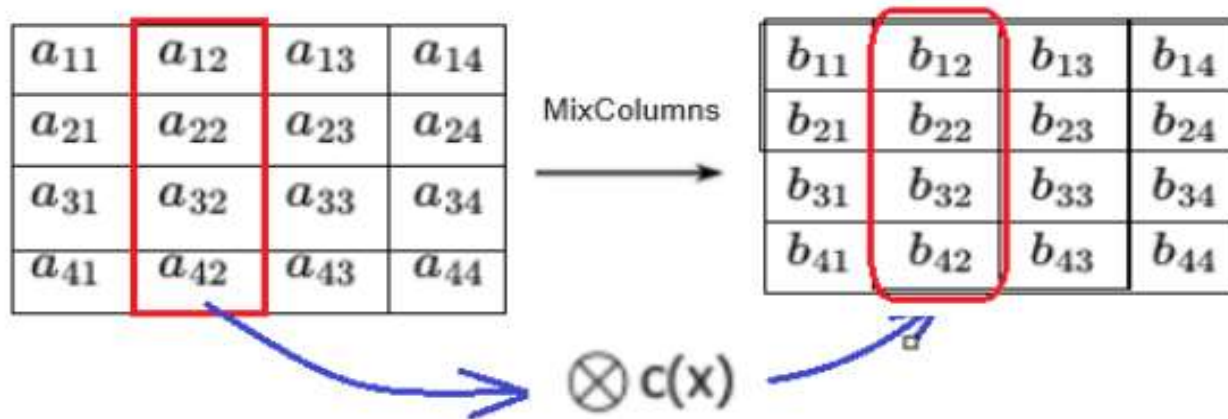
Reversibility is based on the fact that for any bytes A and K (byte = 8 bits) it is true that

If  $B = A \oplus K$  ( $\oplus$  is bitwise XOR),  
then  $A = B \oplus K$

Proof:  $(A \oplus K) \oplus K = A \oplus (K \oplus K) = A$

## MixColumns step

Each column of the state is multiplied with a fixed polynomial  $c(x)$  using Galois' Field multiplication rule, which is explained in later slides.



There is a detailed description in **Appendix1** of the way of using polynomial multiplication in MixColumns step. There is a bonus problem related to Appendix1.

### Reversibility of MixColumns step:

In Galois' Field every polynomial has an inverse polynomial, which can be used to invert the operation, which is needed in AES decryption.

That is: if  $b(x) = c(x)a(x)$ , then  $a(x) = c^{-1}(x)b(x)$

**MixColumns and Substitute Bytes steps use the theory of Galois' Fields.**

Next slides contain the principle with a calculated example of how **multiplication of bytes** can be defined using polynomial presentation of bytes.

### **Presentation of bytes as polynomials in Galois' Field $GF(2^8)$**

An 8 bit binary number  $(b_1b_2b_3b_4b_5b_6b_7b_8)$ , where each  $b_i \in \{0,1\}$  are presented as 7th degree polynomials  $b_1x^7 + b_2x^6 + b_3x^5 + b_4x^4 + b_5x^3 + b_6x^2 + b_7x + b_8$ .

**Example:** Presentation of bytes D4 and 69 as polynomials.

$$D4_{16} = 1101\ 0100 = x^7 + x^6 + x^4 + x^2$$

$$69_{16} = 0110\ 1001 = x^6 + x^5 + x^3 + 1$$

### **Galois' field multiplication rule: steps + calculated example**

**The "Galois' product" of two bytes a and b is calculated with following steps:**

**Step1:** Calculate the product of polynomials using normal polynomial multiplication

**Step2:** Reduce the coefficients of the product to binary numbers using even  $\rightarrow 0$ , odd  $\rightarrow 1$ .

**Step3:** Divide the result of step2 with a specific 8th degree "Field Irreducible polynomial" which in AES is  $x^8 + x^4 + x^3 + x + 1$ . The final result is the polynomial remainder of the division, where the coefficients are reduced to binaries just like in step 2

In the next slides it is shown how product  $D4 \otimes 69$  can be calculated, first using polynomial presentations and then using bit operations.



## Method 1. Calculation of $D4 \otimes 69$ using polynomial presentations

**Step1)** Using standard polynomial multiplication we get

$$(x^7 + x^6 + x^4 + x^2)(x^6 + x^5 + x^3 + 1) = x^{13} + 2x^{12} + x^{11} + 2x^{10} + 2x^9 + x^8 + 3x^7 + x^6 + x^5 + x^4 + x^2$$

**Step2)** Applying rule: even  $\rightarrow 0$ , odd  $\rightarrow 1$  gives a polynomial with binary coefficients

$$x^{13} + x^{11} + x^8 + x^7 + x^6 + x^5 + x^4 + x^2$$

Steps 1 and 2 can be done with a single WolframAlpha. command : `expand (x7 +x6 +x4 +x2)(x6 +x5 +x3 +1) mod 2` , which gives  $x^{13} + x^{11} + x^8 + x^7 + x^6 + x^5 + x^4 + x^2$

**Step3)** The result of step2 is divided with  $x^8 + x^4 + x^3 + x + 1$  in order to calculate the polynomial remainder. Division can be done manually using the division algorithm taught in school. More convenient way is to use calculators like WolframAlpha.

Wolfram Alpha command: **remainder** `(x13 + x11 + x8 + x7 + x6 + x5 + x4 + x2)/(x6 + x5 + x4 + x3 + x)` gives  $-x^6 + x^5 + x^4 - x^3 + 2x^3 + x$ . Reduction with even  $\rightarrow 0$  and odd  $\rightarrow 1$  gives polynomial  $x^6 + x^5 + x^4 + x^3 + x$

The result  $x^6 + x^5 + x^4 + x^3 + x$  corresponds byte  $7A = 0111\ 1010$ . Thus  $D4 \otimes 69 = 7A$

The use of polynomial presentations is not very efficient way of calculating the product of bytes. There is a faster method explained in the next slide, which uses binary operations.



AES SBOX, which is based on Galois' Field multiplication, can be presented in table form.

		Y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
X	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	a	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	b	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	c	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	d	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	e	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	f	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Figure 3.  
AES S-box

A byte can be presented in hexadecimal form as XY, where  $00 \leq XY \leq FF$

In Substitute Bytes step XY is replaced with a hex number found in the row X and in the column Y of the table.

Example: Image of 5D  
Is 4C

**Reversibility:** The hexadecimal numbers in the substitution table are obtained by Galois' Field polynomial multiplication. Galois Field  $GF(2^8)$  forms a group with respect to multiplication, which means that the multiplier polynomial has an inverse. With the inverse polynomial it is possible to invert the substitution table. The inverse table is used in AES decryption.

# Summary of AES

- **AES** encryption algorithm consists of rounds: AES128 has 10 rounds, AES256 has 14 rounds.
- Each round include several steps which produce necessary confusion and diffusion.
- All steps are reversible. MixColumn and Substitute Bytes steps use Galois' Field multiplication in a way developed by a Finnish cryptographer Kaisa Nyberg.
- Galois theory is well established and widely known. This makes AES a transparent and provably secure algorithm with no hidden backdoors. (Predecessor DES was anything but transparent and there were rumors of backdoors around it)

# Modes of operation

## 1) ECB : (Electronic Codebook mode)

Message blocks are encrypted independently => Identical message blocks have identical cipher blocks. Therefore ECB mode is not secure and not used in block ciphers

## 2) CBC : Cipher Block Chaining mode (used in TLS until 2017)

Each cipher block functions as extra input of encryption of next block. The ciphers of first block affects all remaining blocks.

The following line from 2017 describing algorithms of Nordea Bank's TLS connections

Tekniset tiedot

Yhteys salattu (TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA, 256-bittinen avain, TLS 1.2)

## 3) GCM : Galois Counter Mode (today's standard in TLS)

In 2023 Nordea Bank's TLS uses AES in GCM mode

Tekniset tiedot

Yhteys salattu (TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384, 256-bittinen avain, TLS 1.2)

# About AES\_CBC and AES\_GCM modes

One can still find TLS servers using AES\_CBC .

Tekniset tiedot

Yhteys salattu (TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA, 256-bittinen avain, TLS 1.2)

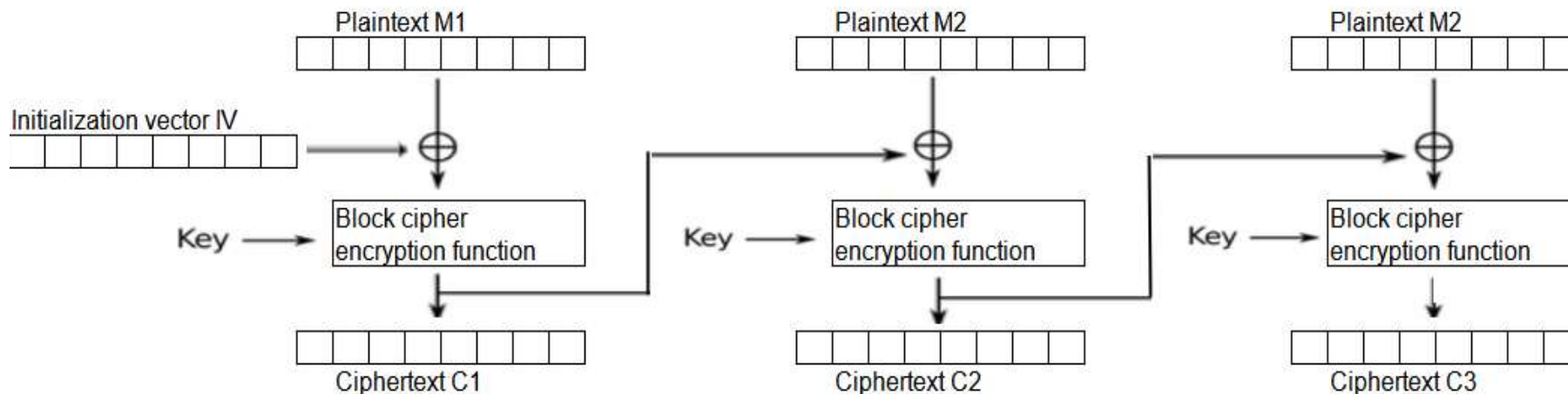


Diagram of CBC mode

Data is encrypted in 128 bit blocks. Inputs of encryption are the message block, key, and the previous cipherblock (or IV). Every cipherblocks affects all future cipher blocks, which makes the encryption stronger against attacks and makes manipulations of some part of the cipher during transmission impossible..



## GCM – mode ("Galois Counter Mode")

GCM has replaced CBC mode in net banks, because it provides stronger protection of integrity and authentication of sender.

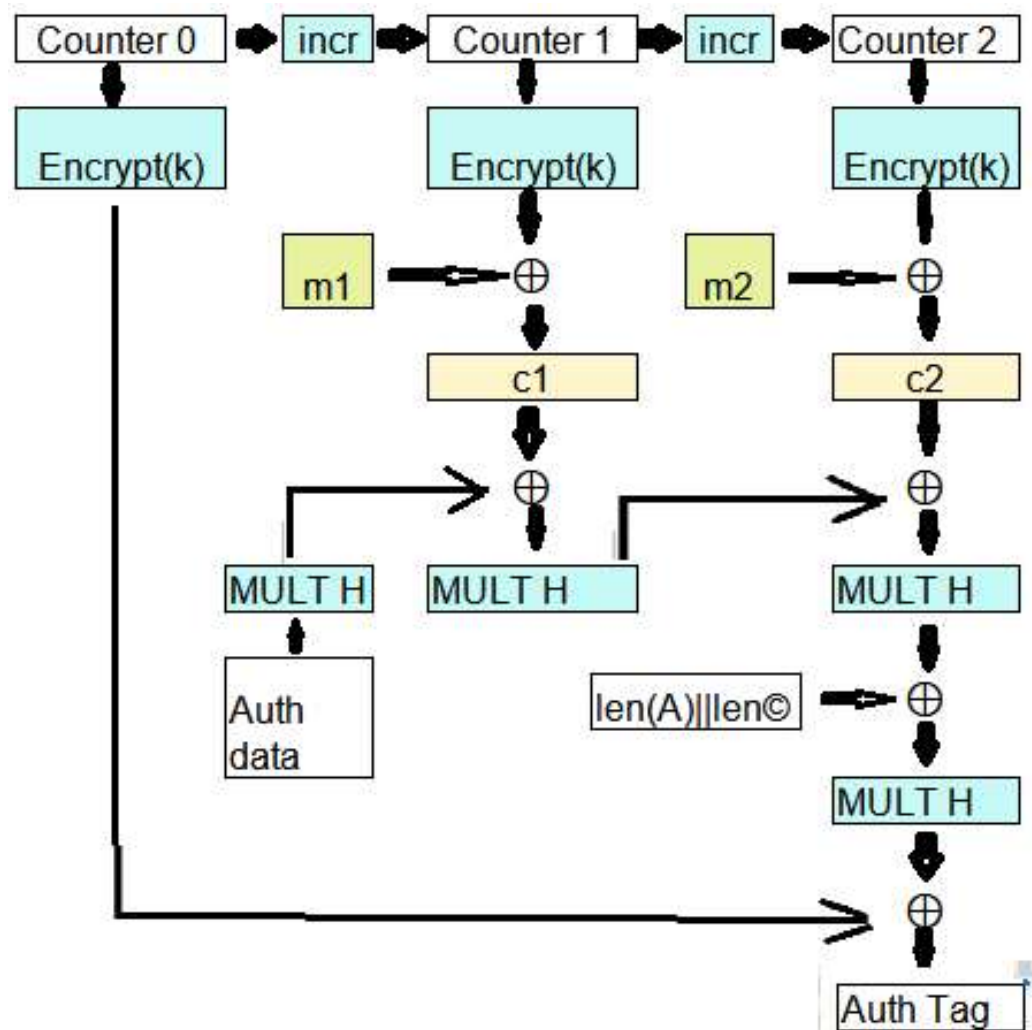
Tekniset tiedot

Yhteys salattu (TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384, 256-bittinen avain, TLS 1.2)

In GCM- mode the counter value (1,2,...) is encrypted with symmetric key K. The output (128 bits) is XOR:ed with the first message block m1 to produce the first cipher block c1.

Authentication data (senders personal data + previous cipher blocks) is collected to a 384 bit hash value using Galois Field multiplication rule.

The goals of this mode are to make attacks impossible and secure the integrity and authenticity of the sender



# Finnish governments instructions for block ciphers (2021)

National security classification	IV	III	II	I
Block ciphers allowed	AES 128 Serpent(128) ChaCha20	AES 196 Serpent(196)	AES 256 Serpent(256)	none *)

\*) Top secret documents (class 1) are allowed only in paper form.  
That is why class1 has no recommendation for a block cipher

**In most cases the government uses AES encryption**

Source: Cyber security center of Finland



# Is AES post-quantum secure?

Quantum computers will be "game changers" in cryptography.

According to today's knowledge AES128 will no longer be secure in post-quantum world, but AES256 and AES512 still provide adequate security. \*)

\*) There is a quantum algorithm called Grover's algorithm, which can break AES128 key in a short time using 128 qubit quantum computers (such computers don't exist yet)

However Grover's algorithm is not able to crack the AES256 key of 256 bit length. The security of AES256 remains at the level that corresponds AES128 in the present pre-quantum world.

Next topics:

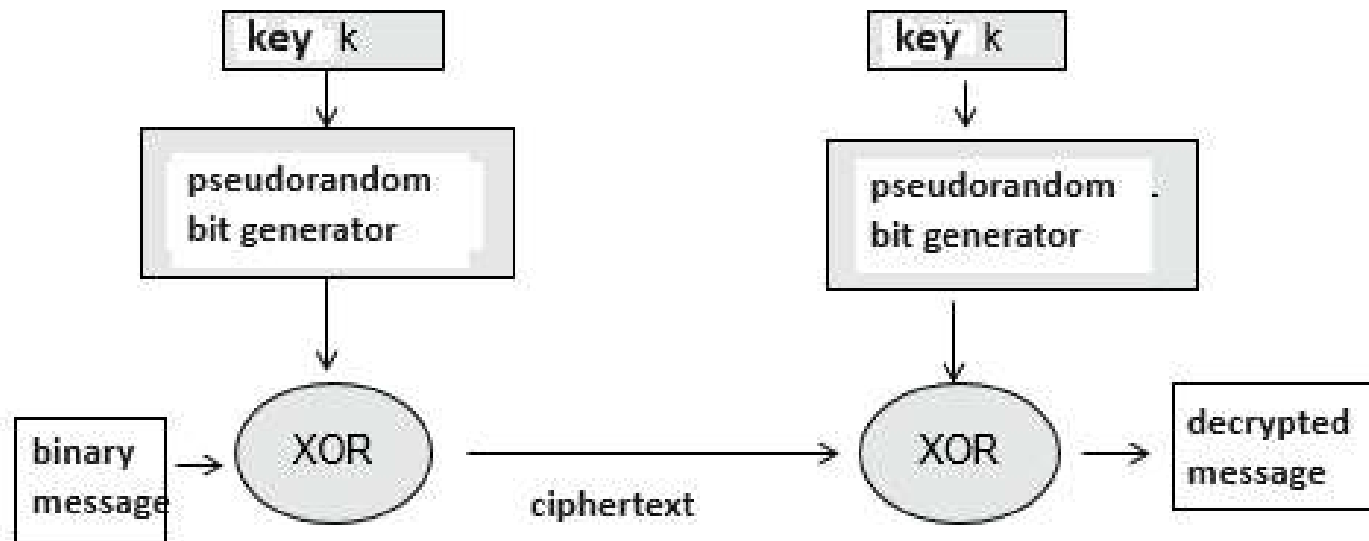
Synchronous stream ciphers

Pseudorandom number generation

LFSR registers

Mobile network encryption

# Synchronous stream ciphers



The **encryption algorithm A5** of GSM phones is a synchronous stream cipher. It imitates the idea of Vernam's version of One Time Pad, in which key is an one time random bit sequence of same length as the message.

GSM phones have a micro circuit which produces a pseudorandom bit sequence, which is added with XOR to the binary message.

GSM call would be safer, if the key sequence would be truly random, which it is not.: Key sequence created by GSM phone looks random, but it is deterministic.

GSM calls are not secure, encryption of GSM calls can be broken. GSM network has been replaced by 3G, 4G and 5G networks, which are more secure.

# Definition of pseudorandomness

**PN- sequence** (PN = "pseudo noise") or pseudorandom bit sequence is a key -based, deterministic bit sequence, which satisfies three properties defined by **Samuel W Colomb** in 1950's..

## PN sequence properties:

**G1.** Sequence must have 50% ones 50% zeros

**G2.** Relative frequencies of blocks repeating same bit (0 or 1)

010	or	101	single bit block
0110		1001	two identical bits block
01110		10001	three identical bits block
011110		100001	four identical bits block
have ratios $1/2$ , $1/4$ , $1/8$ , $1/16$			

**G3.** When a sequence is rotated and the rotated sequence is compared with the original, the number of incidences and disincidences should be approximately equal. This property is measured by so called autocorrelation coefficient, which should be close to zero for all rotations of sequence, which means that the sequence does not have internal periodicity.

## **Additional features required from cryptographic random number generators**

4. All PRNG generators have a period. (They start to produce same numbers when period is full).

**Period should be as large as possible**

5. Random number generation should be **very fast**

6. It should be impossible to calculate next or previous random numbers knowing one of the numbers.

# "Cryptographically secure random number generation"

Secure communication needs "**cryptographically secure**" **pseudorandom number generators (CSPRNG)** for example in creating symmetric keys for AES.  
(Most pseudorandom number generators are not cryptographically secure).

## Some CSPRNG methods

**1. Entropy source method.** Unpredictable data stream is collected mainly from computers operating system (key inputs, time stamps, interruptions, sensors, ....). This data is hashed to create random numbers.

**2. "Stretching entropy".** Data obtained from 1st method is limited, if lots of random numbers are needed at once. One can extend the quantity of random numbers using for example AES in counter mode. Each output is a random number..

**Random key generation is critical issue for safe communication.** A non-safe random number generator has been deliberately used for eavesdropping communication.

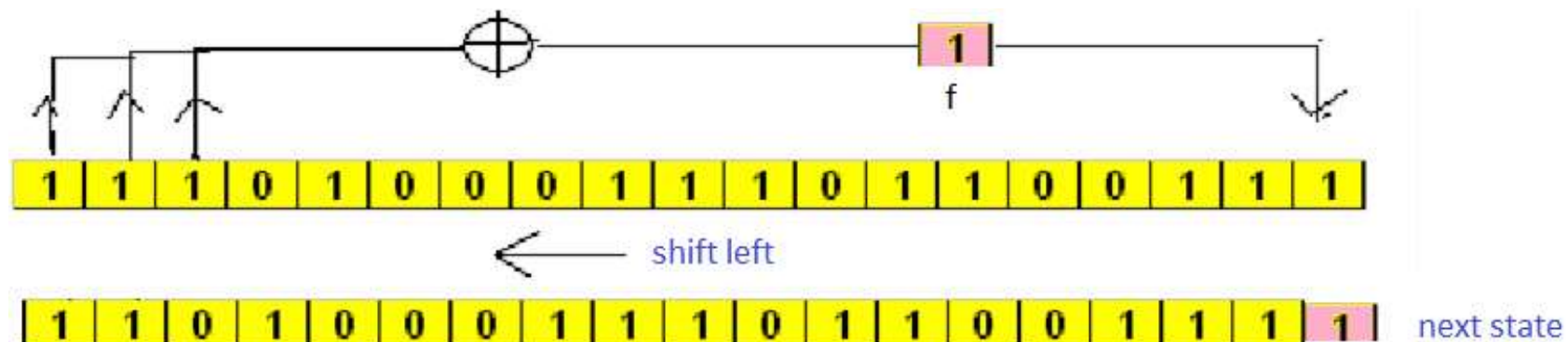
*"The Guardian and The New York Times have reported in 2013 that the National Security Agency (NSA) inserted a backdoor into a pseudorandom number generator (PRNG) of NIST SP 800-90A which allows the NSA to readily decrypt material that was encrypted with the aid of Dual EC DRBG. Both papers report[30][31] that, as independent security experts long suspected,[32] the NSA has been introducing weaknesses into CSPRNG standard 800-90; this being confirmed for the first time by one of the top secret documents leaked to the Guardian by Edward Snowden."* (Wikipedia)

# LFSR - linear feedback shift register

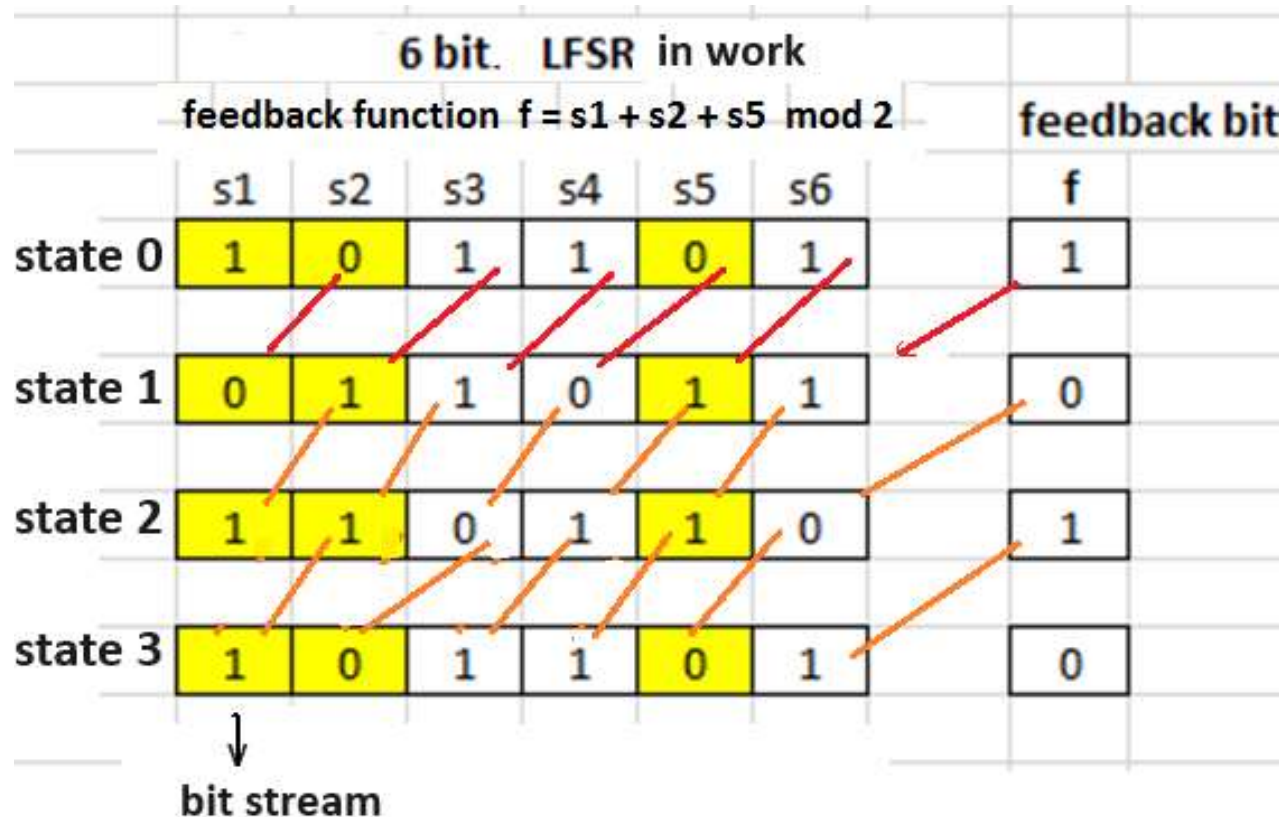
Is a microcircuit traditionally used in RAND function of pocket calculators and later in GSM phones

\* LFSR is n bit register which can be used to produce random numbers. During every clock pulse the bits of the registers are shifted one step to the left. The new rightmost bit, called the feedback bit, is calculated using XOR addition from specific bit values just before the shift

In the picture feedback bit is calculated from first three bits. In the example feedback bit  $f = 1 + 1 + 1 \bmod 2$  which is equal to 1.



# Example of LFSR in operation



Feedback bit is calculated using formula

$$f = s_1 + s_2 + s_5 \pmod 2$$

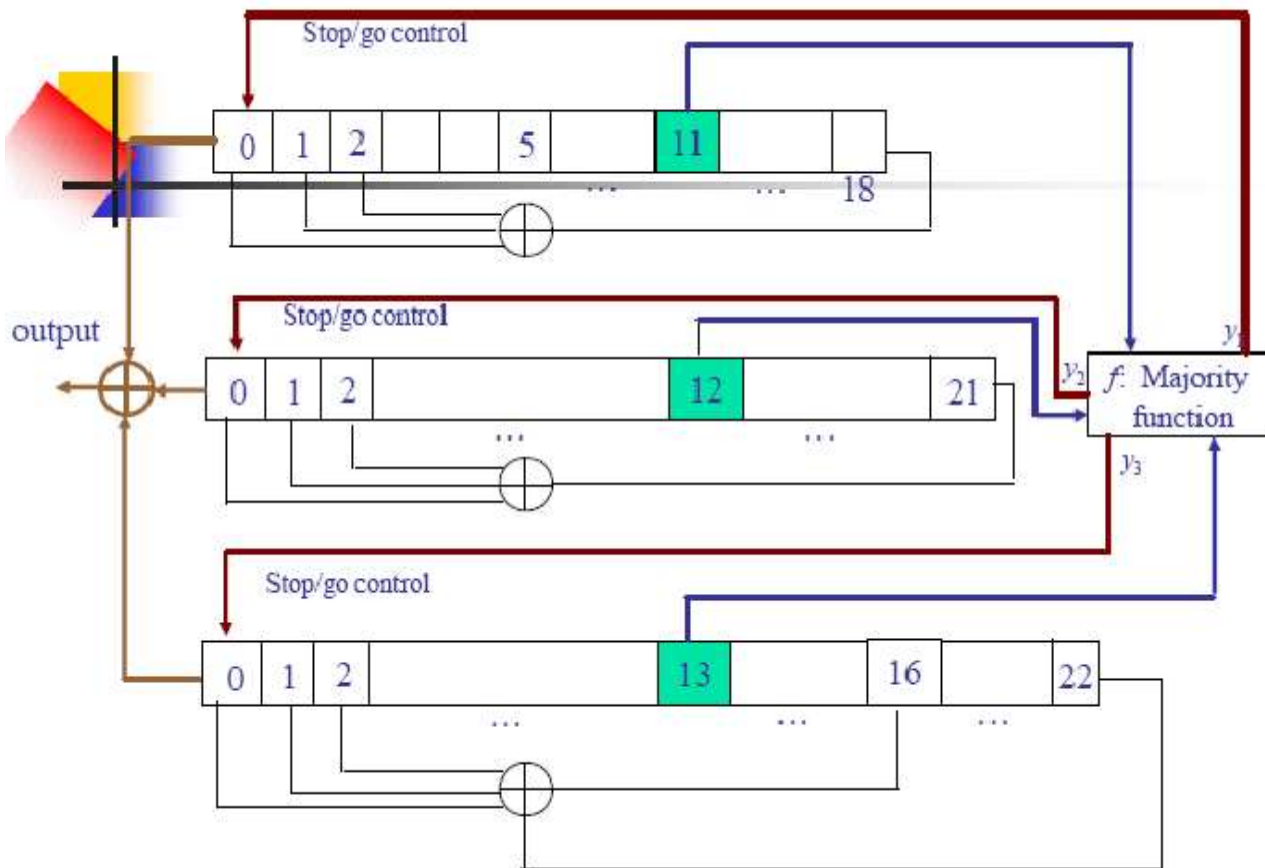
before bits are shifted to left.

LFSR registers leftmost s1 bits form a sequence, which fulfills the three properties of definition of pseudorandomness.

**LFSR registers were used earlier in GSM phones. Today some 3G, 4G and 5G networks use SNOW3G encryption, which also uses LFSR registers.**



# GSM phones A5 PRNG-generator



**GSM – phone has 3 LFSR-  
registers** of lengths 19, 22 and  
23 bits

**Initial state is the 64 bit  
symmetric key.**  $19 + 22 + 23 = 64$

**At each clock pulse circuit  
generates one pseudo random  
bit, which is a XOR-sum of zero  
bits.**

**To increase security a majority  
function  $f$  is added to the circuit.**  
It is calculated from bits marked  
green in the picture, which of the  
three registers are shifted during  
one clock pulse and which  
remain as they are.

Majority function: If all marked green bits have identical values, all registers move forwards. In other cases those two move, which have identical values .

# A5 bit stream fulfills pseudorandomness postulates G1, G2, G3

0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,  
1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1,  
1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0,  
1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1,  
1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0,  
1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1,  
1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0,  
0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1,  
1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0,  
1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1,  
0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1,  
1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0,  
1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0,  
1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1,  
1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0,

**GSM encryption:** The key stream and digitalized voice are added using XOR to produce the encrypted signal

Operators server has a similar PRNG and same symmetric initial state  
Operator decrypts signal using XOR.  
Signal is encrypted only in air, not in the cable.

## GSM security

1. Key in A5 encryption is 64 bits, which is not secure.
2. Key sequence is periodic, effective period is  $4/3 \cdot 2^{23}$
3. GSM signal is encrypted only between the mast and the phone (not in cables)
4. GSM is not secure against professional hackers

# Authentication and key agreement in GSM

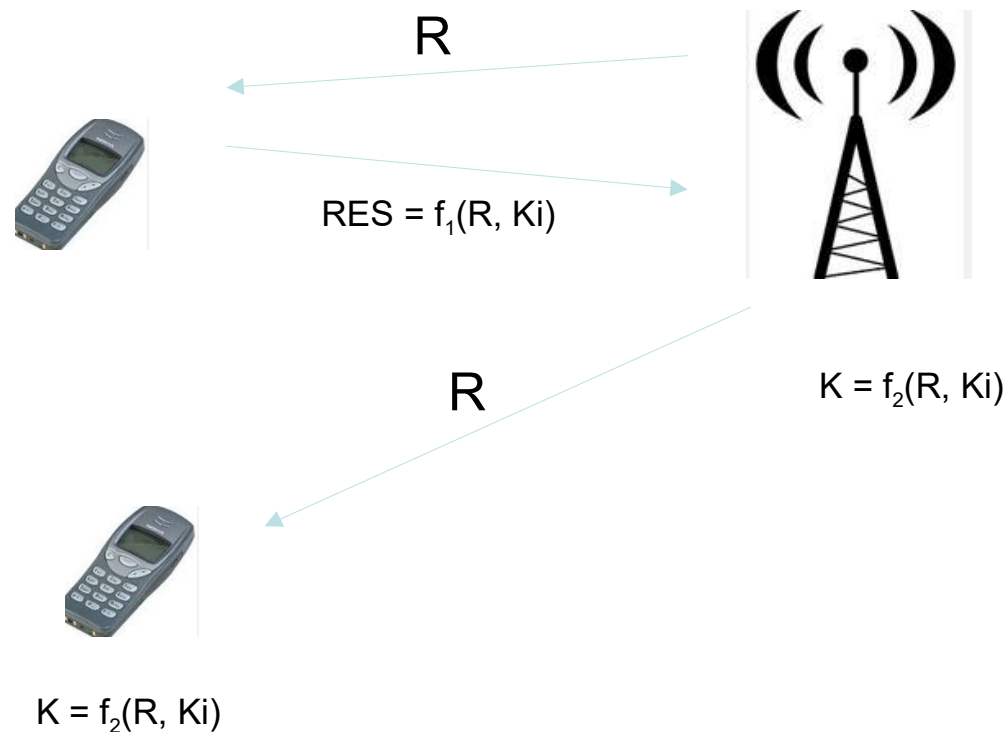
**Authentication A3. Operator sends to the phone a random challenge number R.**

Phone calculates and sends a response RES from R and SIM-key.

Operator calculates RES in the same way. If there is a match, the phone is authenticated.

**Key agreement A8.** Operator sends the phone a random R. The phone calculates a 64 bit key K from R and SIM-key.

The key K is the initial state for the three LFSR -registers



# 3G, 4G, 5G network encryption

**3G and 4G** use block ciphers: AES128 or SNOW3G

- key length = 128 bits
- two way authentication is used: both the phone and the mast authenticate themselves in the beginning of communication

**5G encryption** uses 256 bit versions of AES and SNOW3G

*SNOW3G is a Swedish encryption algorithm created in Lund university.  
It uses LFSR register technology*