CSE413 – Security of Information Systems 2020

PhD Furkan Gözükara, Toros University

https://github.com/FurkanGozukara/Security-of-Information-Systems-CSE413-2020

Lecture 3

Cryptography

Composed from Prof. Audun Jøsang, University of Oslo, Information Security 2018 Lectures

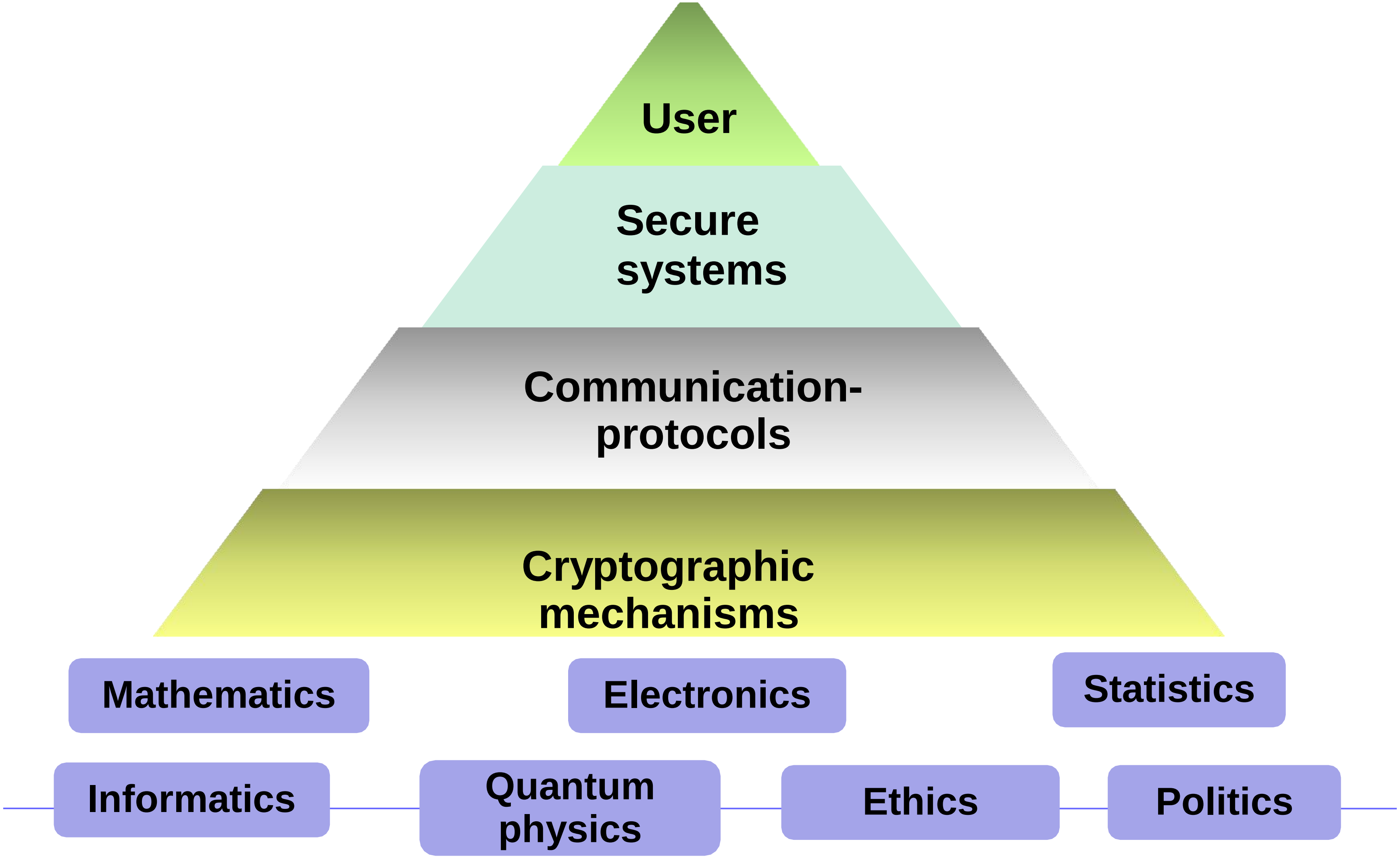
Source : https://www.uio.no/studier/emner/matnat/ifi/INF3510/v18/lectures/

Outline

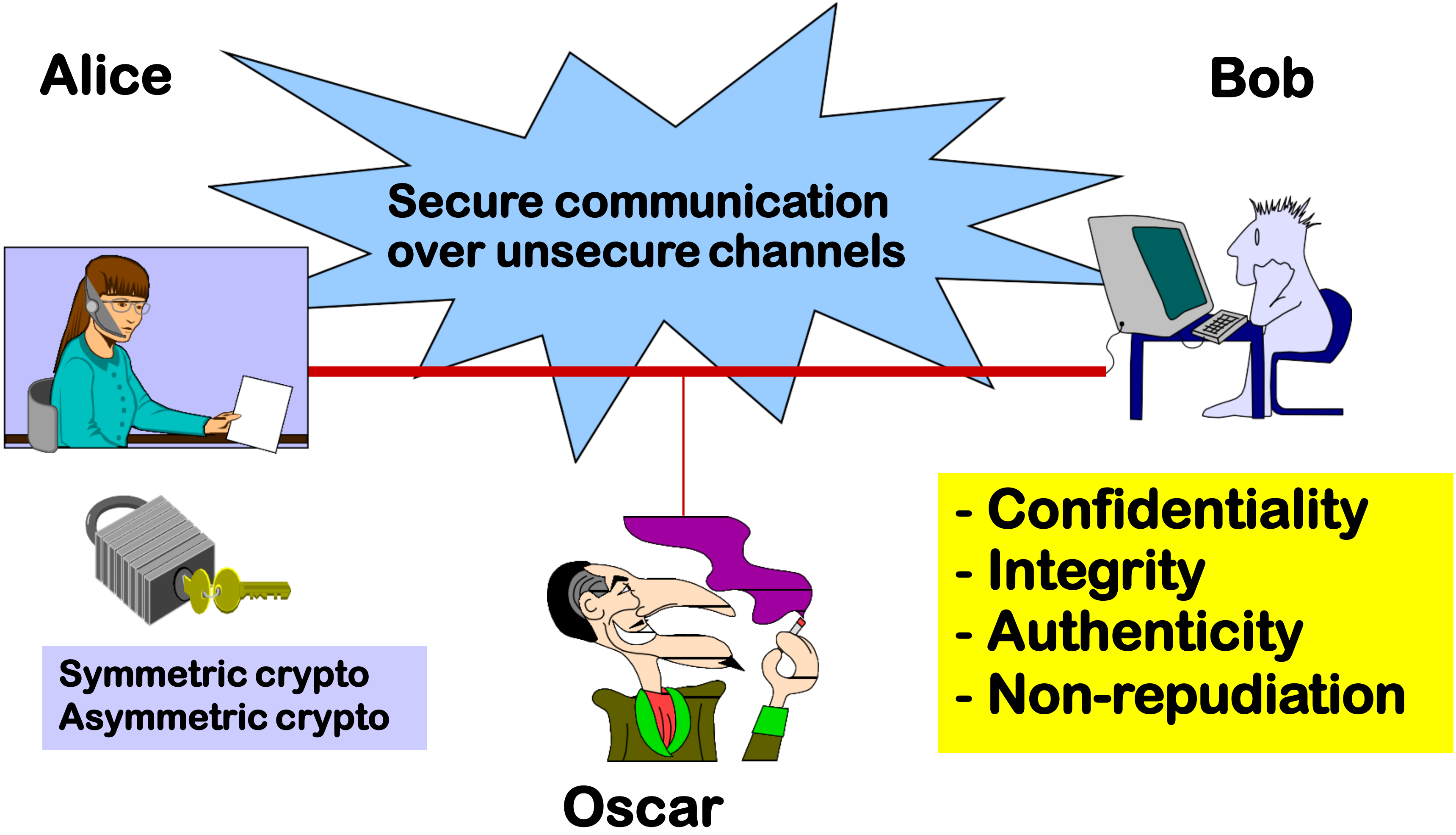
- What is cryptography?
- Brief crypto history.
- Security issues.
- Symmetric cryptography:
  - Stream ciphers.
  - Block ciphers.
  - Hash functions.
- Asymmetric cryptography:
  - Factoring based mechanisms.
  - Discrete Logarithms.
  - Digital signatures.
  - Quantum Resistant Crypto.

Want to learn more?
Look up UNIK 4220

The security pyramid



What is cryptology?



Alice

Bob

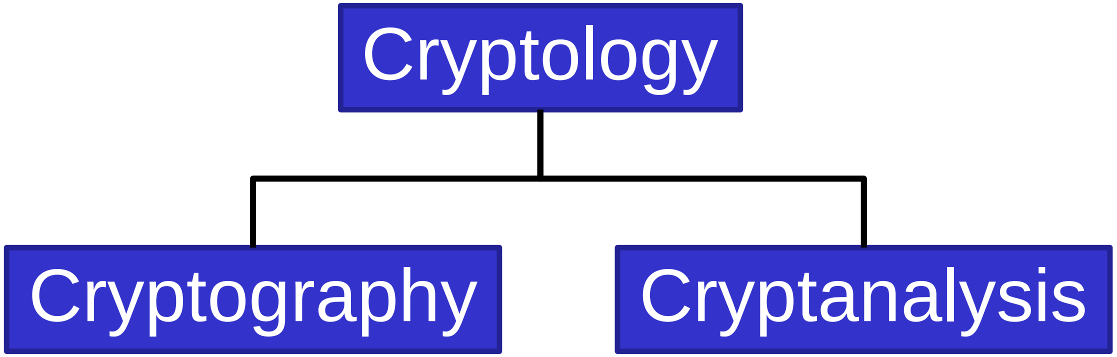
Secure communication over unsecure channels

Oscar

Symmetric crypto
Asymmetric crypto

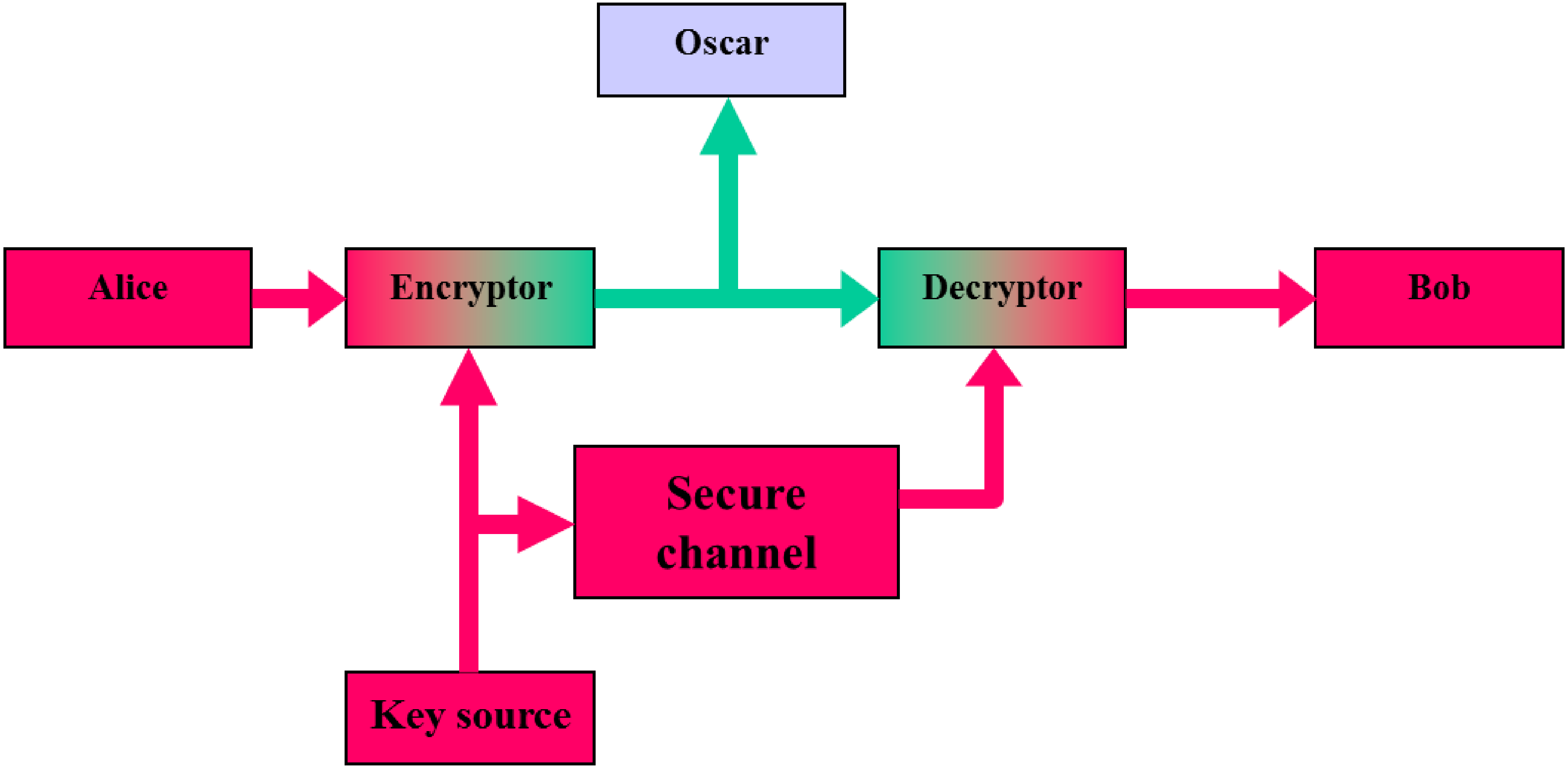
- Confidentiality
- Integrity
- Authenticity
- Non-repudiation

Terminology

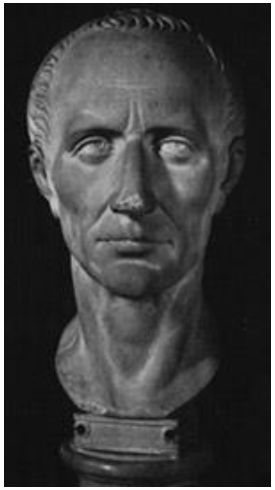


- **Cryptography** is the science of secret writing with the goal of hiding the meaning of a message.
- **Cryptanalysis** is the science and sometimes art of *breaking* cryptosystems.

Model of symmetric cryptosystem



Caesar cipher



Example: Caesar cipher

P = {abcdefghijklmnopqrstuvwxyz}

C = {DEFGHIJKLMNOPQRSTUVWXYZABC}

Plaintext: kryptologi er et spennende fag

Chiphertext: NUBSWRORJL HU HT VSHQQHQGH IDJ

Note: Caesar chipher in this form does not include a variable key, but is an instance of a “shift-cipher” using key *K* = 3.

Numerical encoding of the alphabet

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

p	q	r	s	t	u	v	w	x	y	z	æ	ø	å
14	16	17	18	19	20	21	22	23	24	25	26	27	28

Using this encoding many classical crypto systems can be expressed as algebraic functions over Z_{26} (English alphabet) or Z_{29} (Norwegian alphabet)

Shift cipher

Let **P = C =** $Z_{26.}$ For $0 \leq K \leq 25$, we define

$$E(x, K) = x + K \pmod{26}$$

and

$$D(y, K) = y - K \pmod{26}$$

($x, y \in Z_{26}$)

Question: What is the size of the key space?

Puzzle: ct =

LAHYCXPAJYQHRBWNNMNMOXABNLDANLXVVDWRLJCRXWB

Find the plaintext!

Exhaustive search

```
For[i=0, i<26, i++, Print["Key = ", i, " Plain = ", decrypt[ct,1,i]]]
Key = 0 Plain = LAHYCXPAJYQHRBWNNMNMOXABNLDANLXVVDWRLJCRXWB
Key = 1 Plain = KZGXBWOZIXPGQAVMMLMLNWZAMKCZMKWUUCVQKIBQWVA
Key = 2 Plain = JYFWAVNYHWOFPZULLKLKMVYZLJBYLJVTTBUPJHAPVUZ
Key = 3 Plain = IXEVZUMXGVNEOYTKKJKJLUXYKIAXKIUSSATOIGZOUTY
Key = 4 Plain = HWDUYTLWFUMDNXSJJIJIKTWXJHZWJHTRRZSNHFYNTSX
Key = 5 Plain = GVCTXSKVETLCMWRIIHIHJSVWIGYVIGSQQYRMGEXMSRW
Key = 6 Plain = FUBSWRJUDSKBLVQHHGHGIRUVHFXUHFRPPXQLFDWLRQV
Key = 7 Plain = ETARVQITCRJAKUPGGFGFHQTUGEWTGEQOOWPKECVKQPU
Key = 8 Plain = DSZQUPHSBQIZJTOFFEFEGPSTFDVSFDPNNVOJDBUJPOT
Key = 9 Plain = CRYPTOGRAPHYISNEEDEDFORSECURECOMMUNICATIONS
Key = 10 Plain = BQXOSNFQZOGXHRMDDCDCENQRDBTQDBNLLTMHBZSHNMR
Key = 11 Plain = APWNRMEPYNFWGQLCCBCBDMPQCASPCAMKKSLGAYRGMLQ
Key = 12 Plain = ZOVMQLDOXMEVFPKBBABACLOPBZROBZLJJRKFZXQFLKP
```

- .
-

Substitution cipher - example

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
U	D	M	I	P	Y	Æ	K	O	X	S	N	Å	F	A

p	q	r	s	t	u	v	w	x	y	z	æ	ø	å	
E	R	T	Z	B	Ø	C	Q	G	W	H	L	V	J	

Plaintext: fermatssisteteorem
Ciphertext: YPTÅUBZZOZBPBPATPÅ

What is the size of the key space?

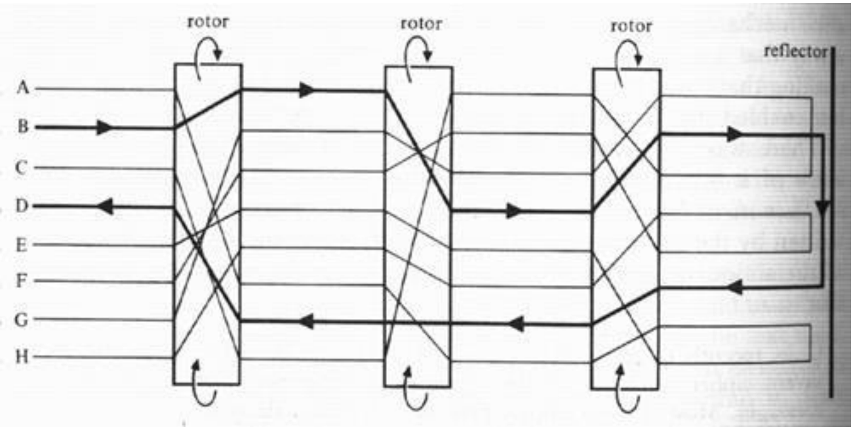
8841761993739701954543616000000 ≈ 2^{103}

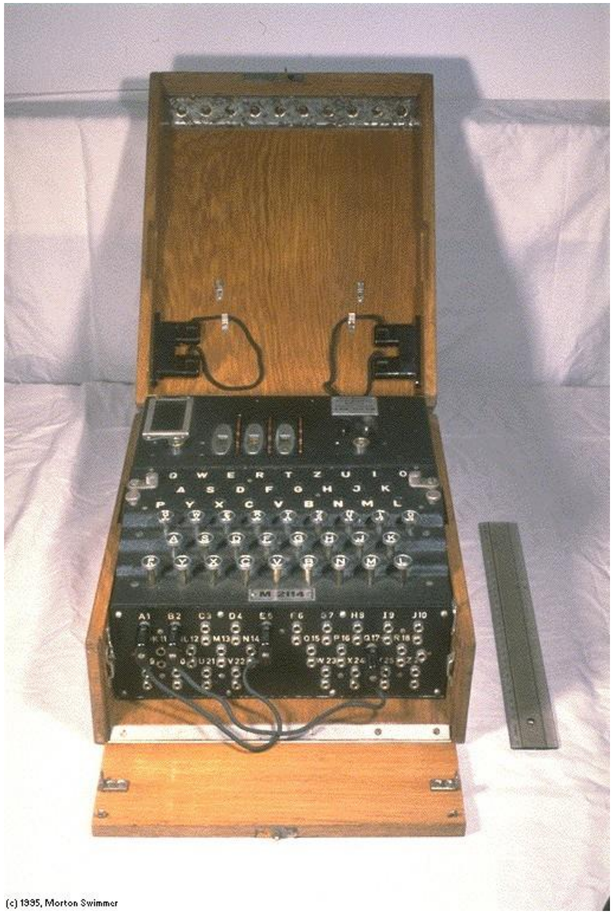
Lessons learned

- A cipher with a small keyspace can easily be attacked by *exhaustive search.*
- A *large keyspace* is necessary for a secure cipher, but it is by itself not sufficient.
- Monoalphabetical substitution ciphers can easily be broken.

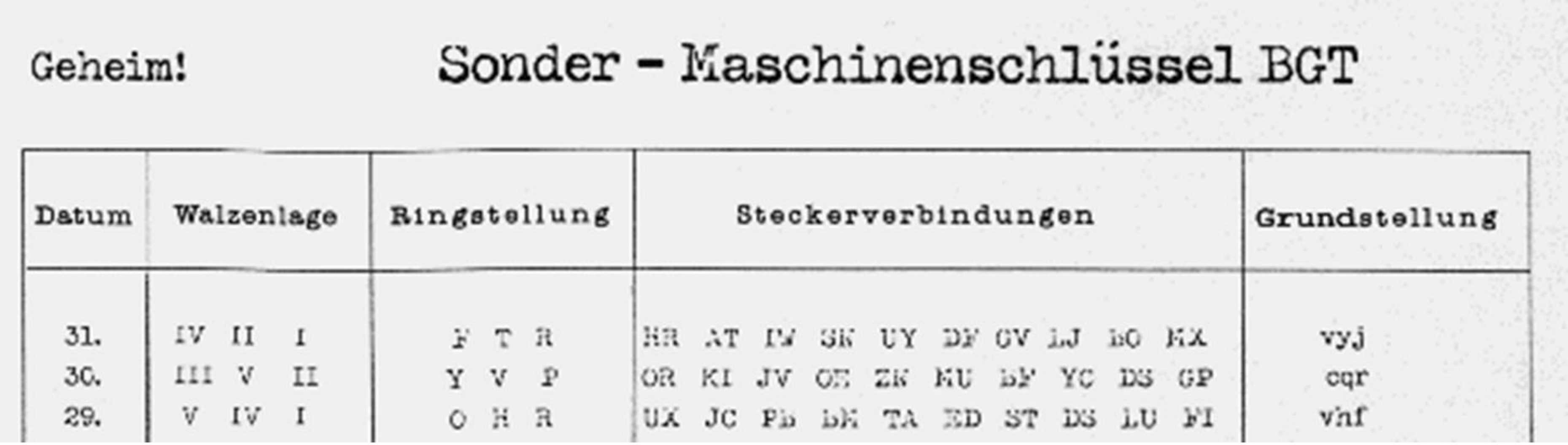
Enigma

- **German WW II crypto machine.**
- **Many different variants.**
- **Polyalphabetical substitution.**
- **Analysed by Polish and English mathematicians.**




(c) 1995, Morton Swimmer

Enigma key list



Geheim!

Sonder - Maschinenschlüssel BGT

Datum	Walzenlage	Ringstellung	Steckerverbindungen	Grundstellung
31.	IV II I	F T R	HR AT IW SK UY DF GV LJ BO MX	vyj
30.	III V II	Y V P	OR KI JV OE ZK MU BF YC DS GP	cqr
29.	V IV I	O H R	UX JC FB BE TA ED ST DS LU FI	vhf

Practical complexity for attacking Enigma

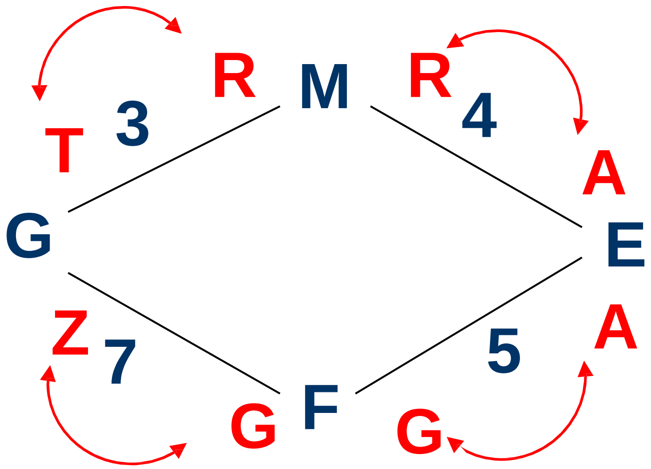
Cryptoanalytical assumptions during WW II:

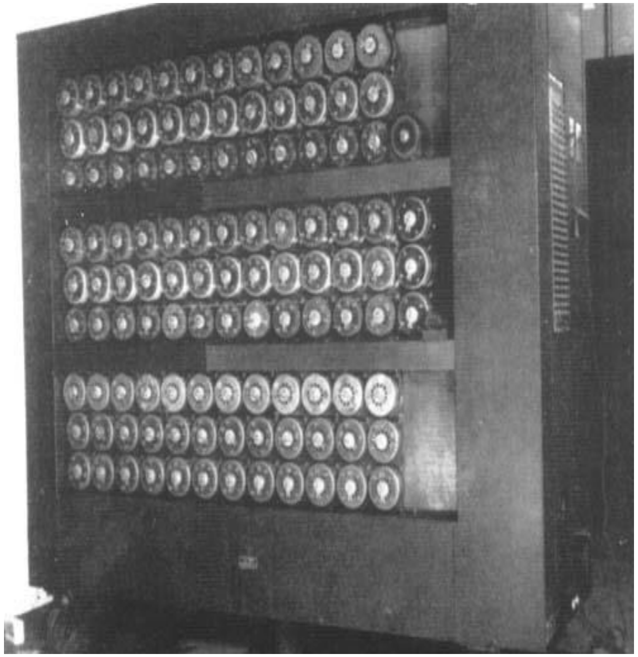
- 3 out of 5 rotors with known wiring.
- 10 stecker couplings.
- Known reflector.

$N = 150\ 738\ 274\ 937\ 250 \cdot 60 \cdot 17\ 576 \cdot 676 =$
107458687327250619360000 (77 bits)

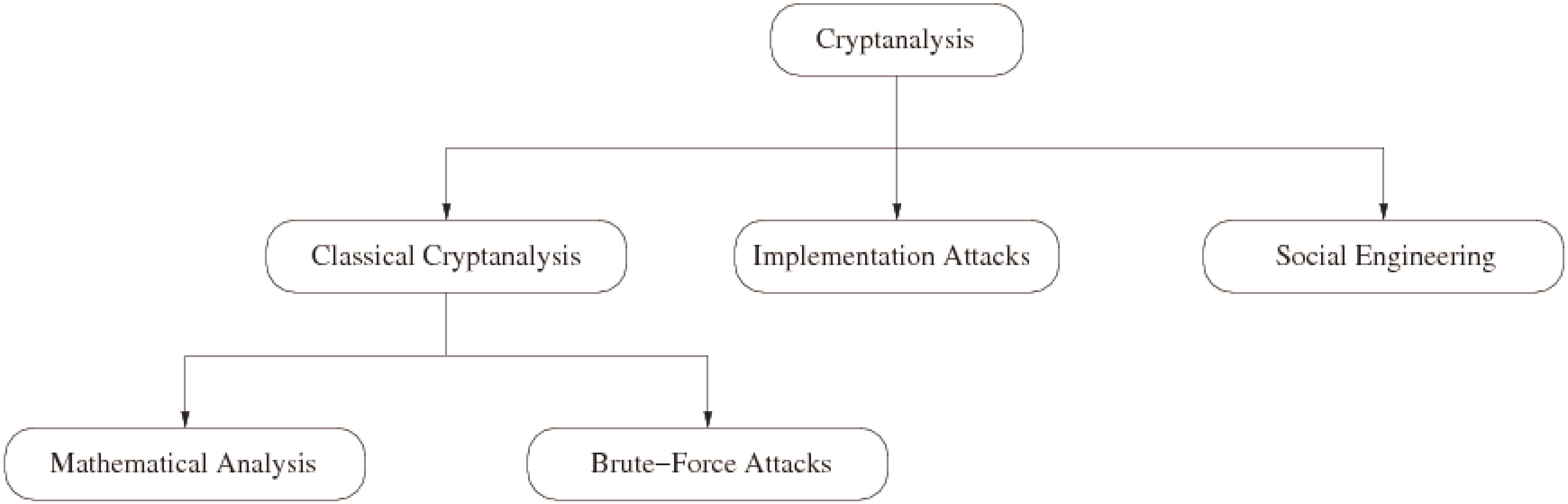
Attacking ENIGMA

Posisjon:	1	2	3	4	5	6	7
Chiffertekst:	J	T	G	E	F	P	G
Crib:	R	O	M	M	E	L	F





Cryptanalysis: Attacking Cryptosystems



- **Classical Attacks:**
  - Mathematical Analysis.
  - Brute-Force Attack.

•**Implementation Attack**: Try to extract the key through reverse engineering or power measurement, e.g., for a banking smart card.

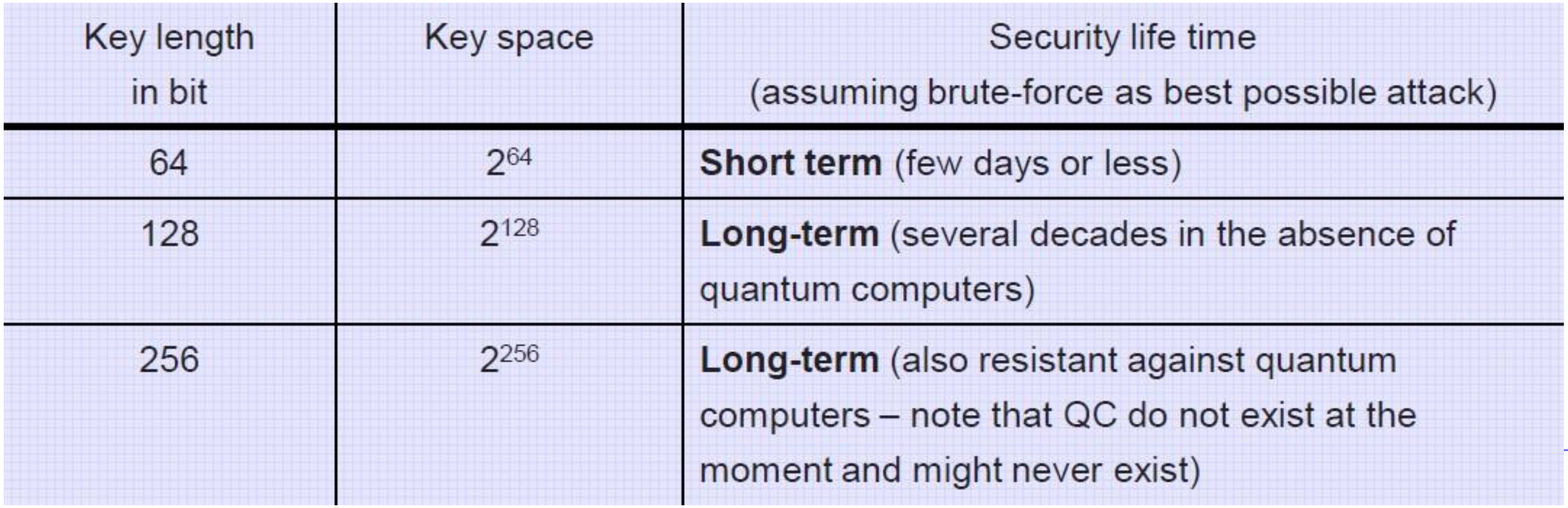
- **Social Engineering**: E.g., trick a user into giving up her password

Brute-Force Attack (or Exhaustive Key Search)

- Treats the cipher as a black box.
- Requires (at least) 1 plaintext-ciphertext pair (x_0, y_0).
- Check all possible keys until condition is fulfilled:

$$\boldsymbol{d_K(y_0) = x_0}$$

- How many keys to we need ?



Key length in bit	Key space	Security life time (assuming brute-force as best possible attack)
64	2^{64}	**Short term** (few days or less)
128	2^{128}	**Long-term** (several decades in the absence of quantum computers)
256	2^{256}	**Long-term** (also resistant against quantum computers – note that QC do not exist at the moment and might never exist)

Attack models:

Known ciphertext.

Known plaintext.

Chosen plaintext (adaptive) .

Chosen ciphertext (adaptive).

What are the goals of the attacker?

- Find the secret plaintext or part of the plaintext.
- Find the encryption key.
- Distinguish the encryption of two different plaintexts.

How clever is the attacker?

Does secure ciphers exist?

- What is a secure cipher?
  - Perfect security.
  - Computational security.
  - Provable security.

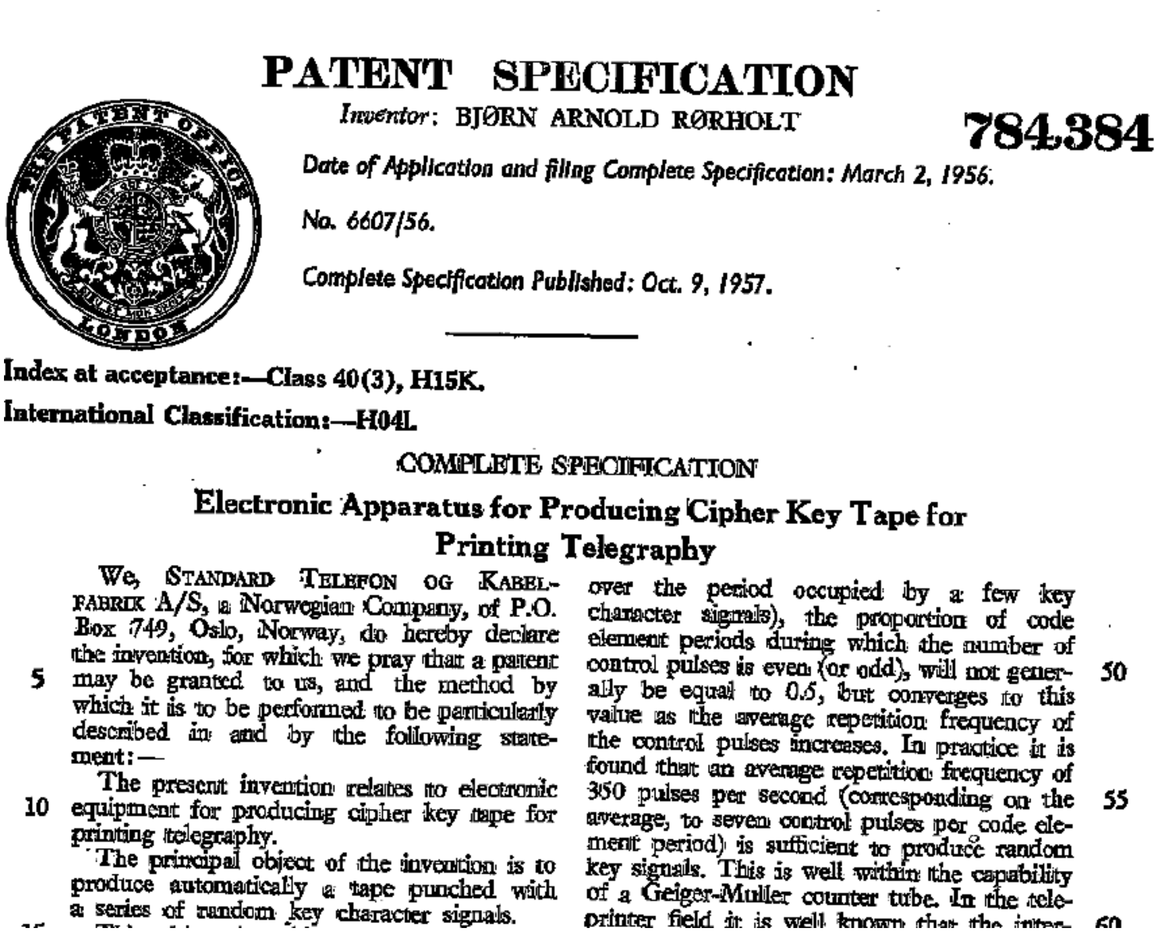
"I'm sorry, we already have a director of security..."

ETCRRM

- Electronic Teleprinter Cryptographic Regenerative Repeater Mixer (ETCRRM).
- Invented by the Norwegian Army Signal Corps in 1950.
- Bjørn Rørholt, Kåre Mesingseth.
- Produced by STK.
- Used for ”Hot-line” between Moskva and Washington.
- About 2000 devices produced.

White House Crypto Room 1960s

Producing key tape for the one-time pad



THE PATENT OFFICE
LONDON

PATENT SPECIFICATION

Inventor: BJØRN ARNOLD RØRHOLT

784,384

Date of Application and filing Complete Specification: March 2, 1956.

No. 6607/56.

Complete Specification Published: Oct. 9, 1957.

Index at acceptance:—Class 40(3), H15K.

International Classification:—H04L.

COMPLETE SPECIFICATION

Electronic Apparatus for Producing Cipher Key Tape for Printing Telegraphy

We, STANDARD TELEFON OG KABELFABRIK A/S, a Norwegian Company, of P.O. Box 749, Oslo, Norway, do hereby declare the invention, for which we pray that a patent may be granted to us, and the method by which it is to be performed to be particularly described in and by the following statement:—

The present invention relates to electronic equipment for producing cipher key tape for printing telegraphy.

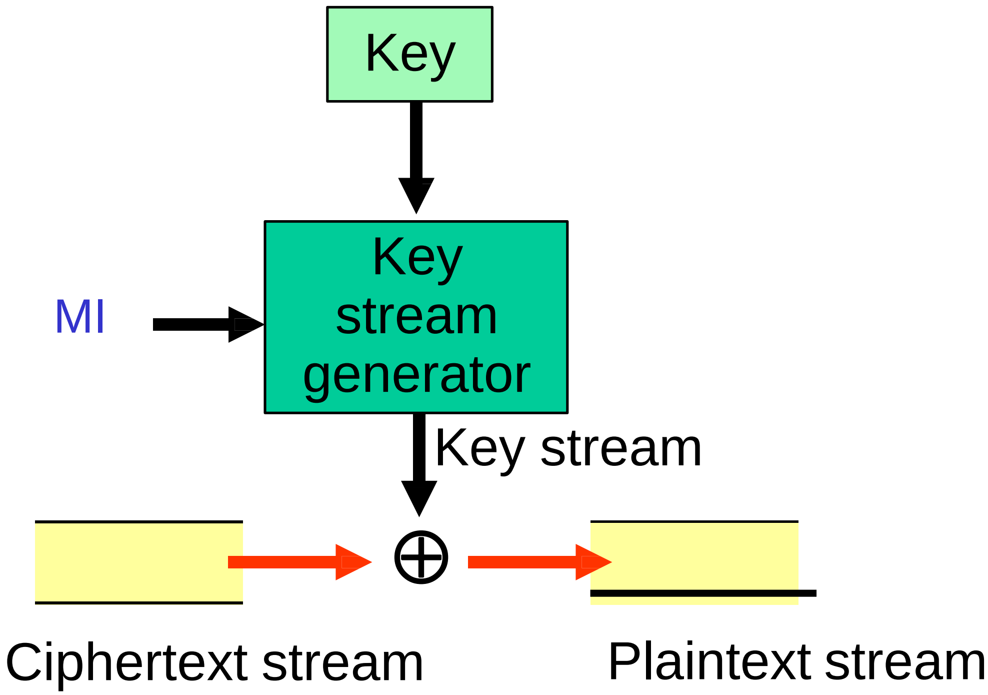
The principal object of the invention is to produce automatically a tape punched with a series of random key character signals.

over the period occupied by a few key character signals), the proportion of code element periods during which the number of control pulses is even (or odd), will not generally be equal to 0.5, but converges to this value as the average repetition frequency of the control pulses increases. In practice it is found that an average repetition frequency of 350 pulses per second (corresponding on the average, to seven control pulses per code element period) is sufficient to produce random key signals. This is well within the capability of a Geiger-Muller counter tube. In the teleprinter field it is well known that the inter-

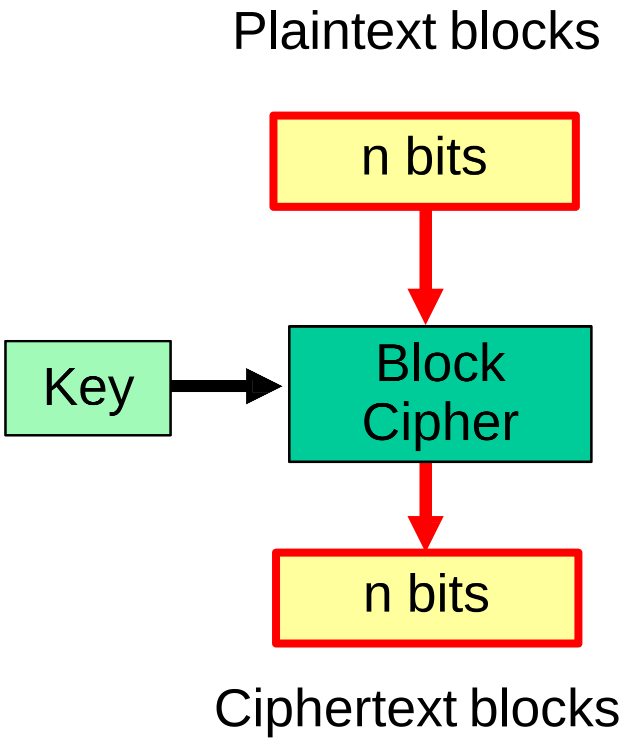
Symmetric encryption

- Is it possible to design secure and practical crypto?

Stream Cipher vs. Block Cipher

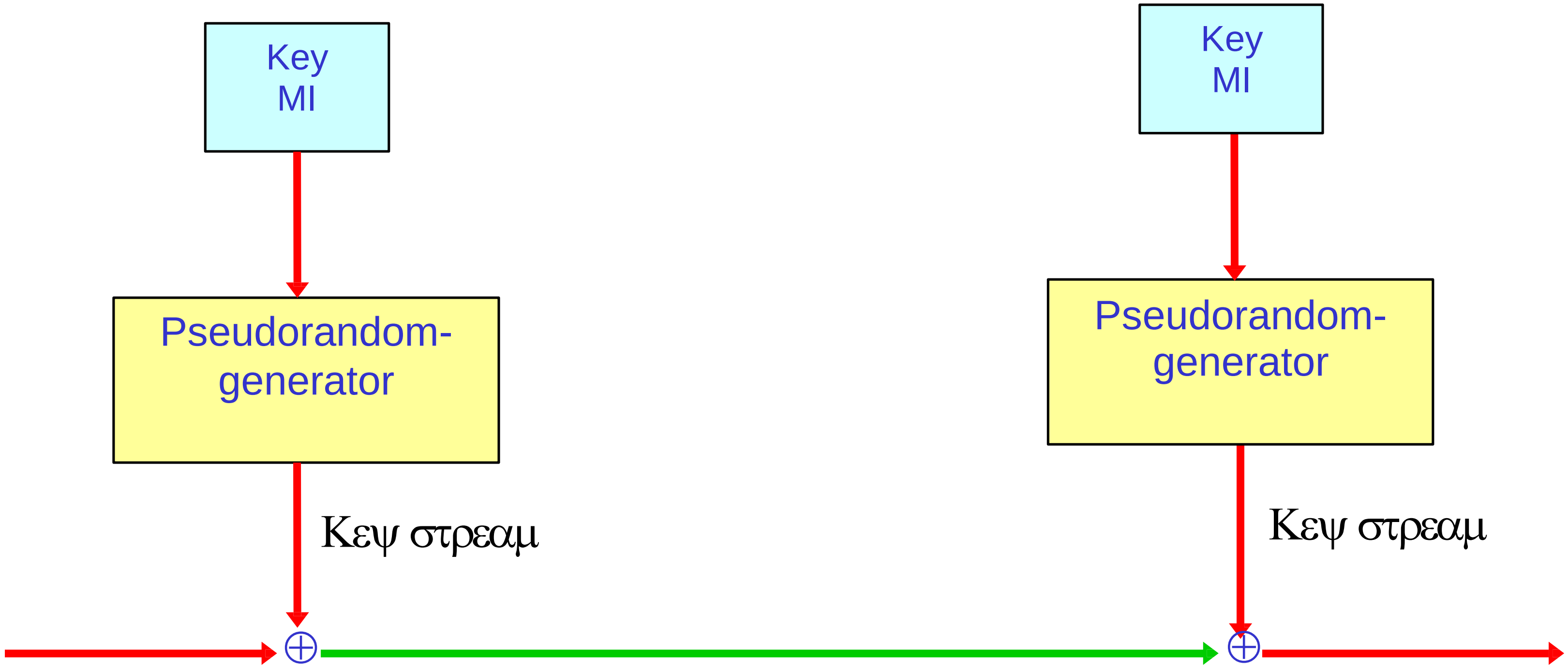


Stream cipher



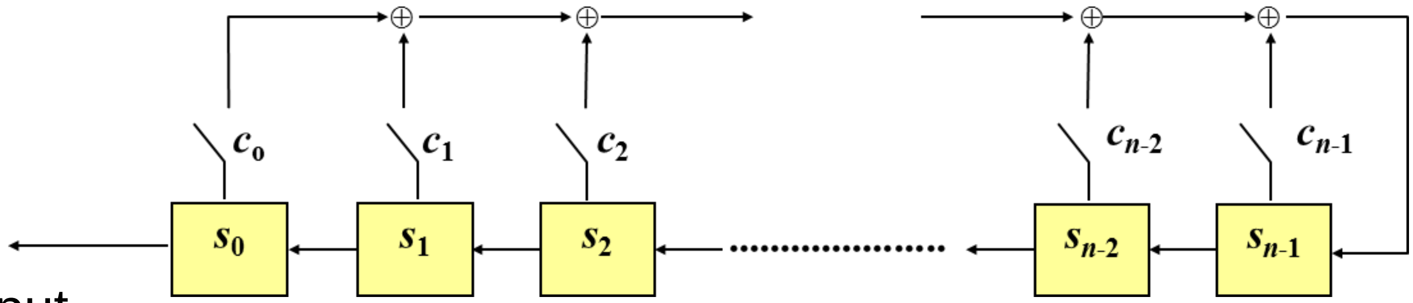
Block cipher

Symmetric stream cipher



LFSR

Linear feedback shift register

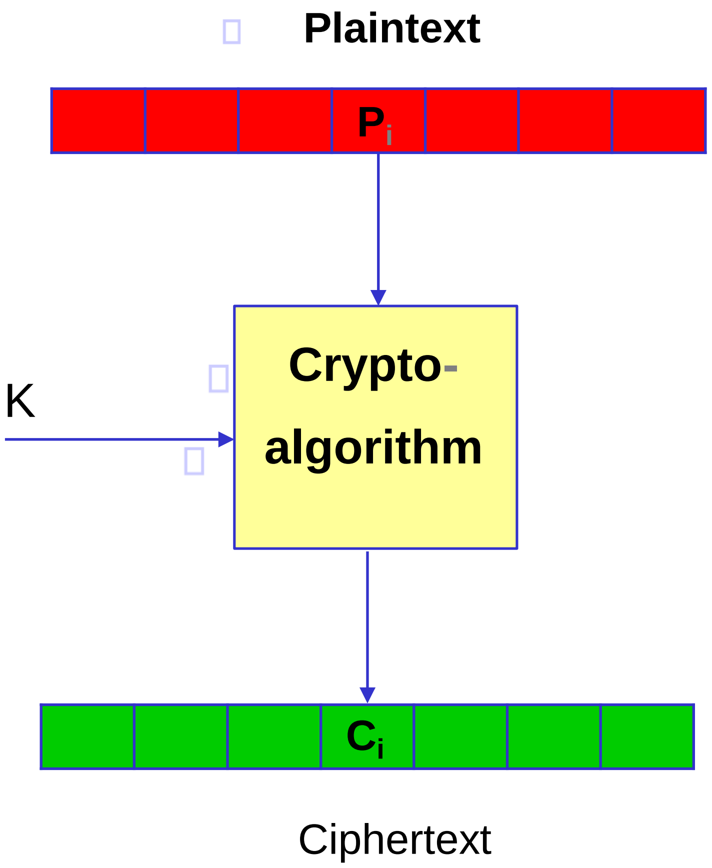


Υσινγ n φλιπ-φλοπσ ωε μαψ γενερατε α βιναρψ σεθυενχε οφ περιοδ 2 ^{n} –1

$$s_{n+i} = c_0\, s_i + c_1\, s_{i+1} + \cdots + c_{n-1}\, s_{i+n-1}$$

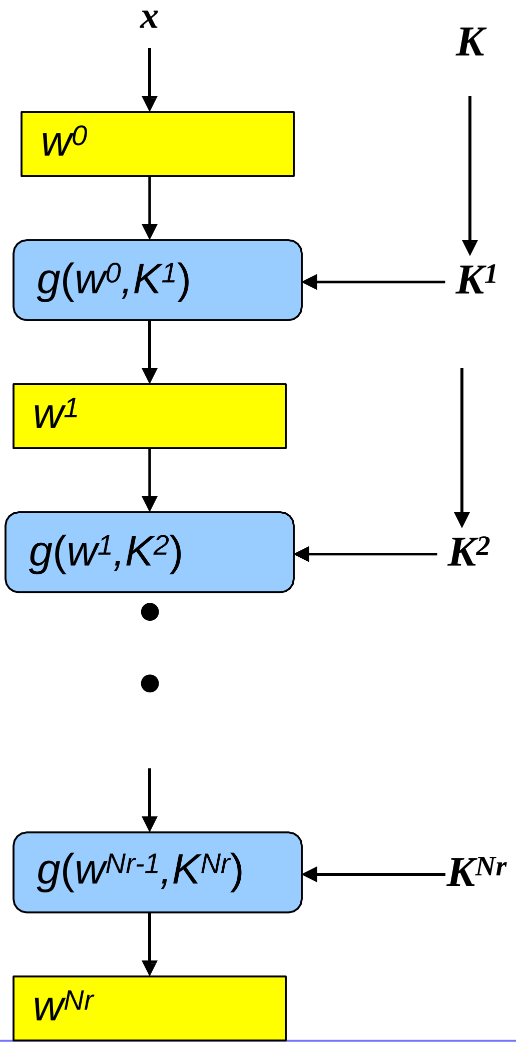
Note: The stream cipher is stateful

Symmetric block cipher



- The algorithm represents a family of permutations of the message space
- Normally designed by iterating a less secure round function
- May be applied in different operational modes
- Must be impossible to derive K based on knowledge of P and C

Itrerated block cipher design



Algorithm:

$w^0 \leftarrow x$

$w^1 \leftarrow g(w^0,K^1)$

$w^2 \leftarrow g(w^1,K^2)$

•

•

$w^{Nr-1} \leftarrow g(w^{Nr-2},K^{Nr-1})$

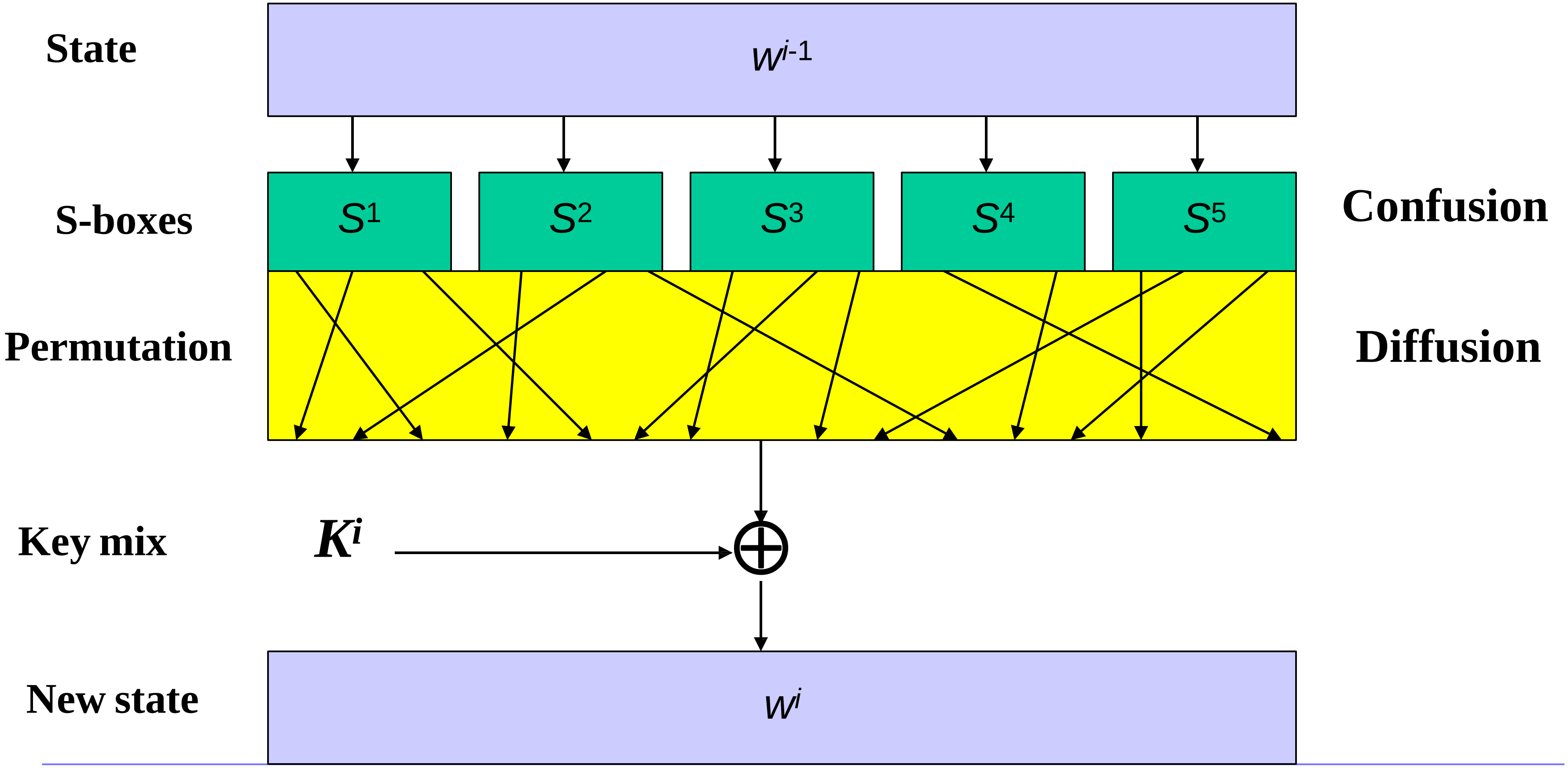
$w^{Nr} \leftarrow g(w^{Nr-1},K^{Nr})$

$y \leftarrow w^{Nr}$

NB! For a fixed *K*, *g* must be injective in order to decrypt *y*

Substitution-Permutation network (SPN):

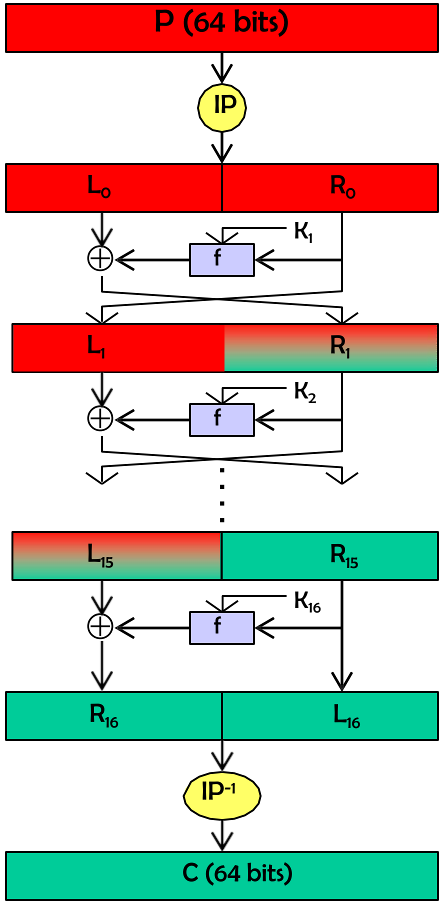
Round function *g* :



Data Encryption Standard

- Published in 1977 by the US National Bureau of Standards for use in unclassified government applications with a 15 year life time.
- 16 round Feistel cipher with 64-bit data blocks, 56-bit keys.
- 56-bit keys were controversial in 1977; today, exhaustive search on 56-bit keys is very feasible.
- Controversial because of classified design criteria, however no loop hole was ever found.

DES architecture



```
DES(P):
$(L_0, R_0) = IP(P)$
FOR i = 1 TO 16
        $L_i = R_{i-1}$
        $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
$C = IP^{-1}(R_{16}, L_{16})$
```

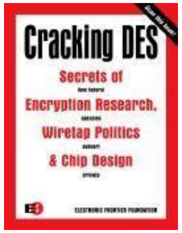
64 bit data block

56 bit key

72.057.594.037.927.936

EFF DES-cracker

- Dedicated ASIC with 24 DES search engines.
- 27 PCBs housing 1800 circuits.
- Can test 92 billion keys per second.
- Cost 250 000 $.
- DES key found July 1998 after 56 hours search.
- Combined effort DES Cracker and 100.000 PCs could test 245 billion keys per second and found key after 22 hours.


Cracking DES

Secrets of

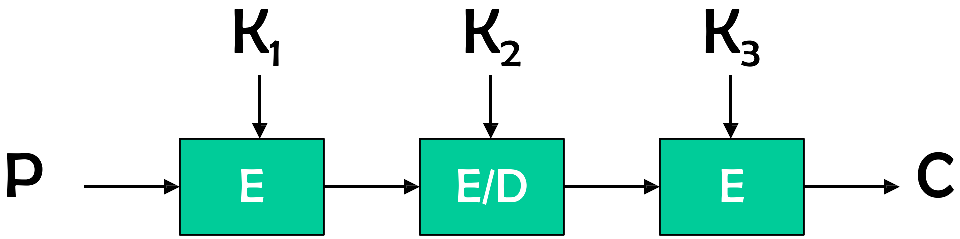
Encryption Research,

Wiretap Politics

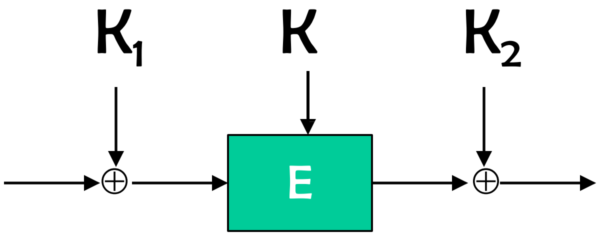
& Chip Design

DES Status

- DES is the “work horse” which over 30 years have inspired cryptographic research and development
- “Outdated by now”!
- Single DES can not be considered as a secure block cipher
- Use 3DES (ANSI 9.52) or DESX



3DES

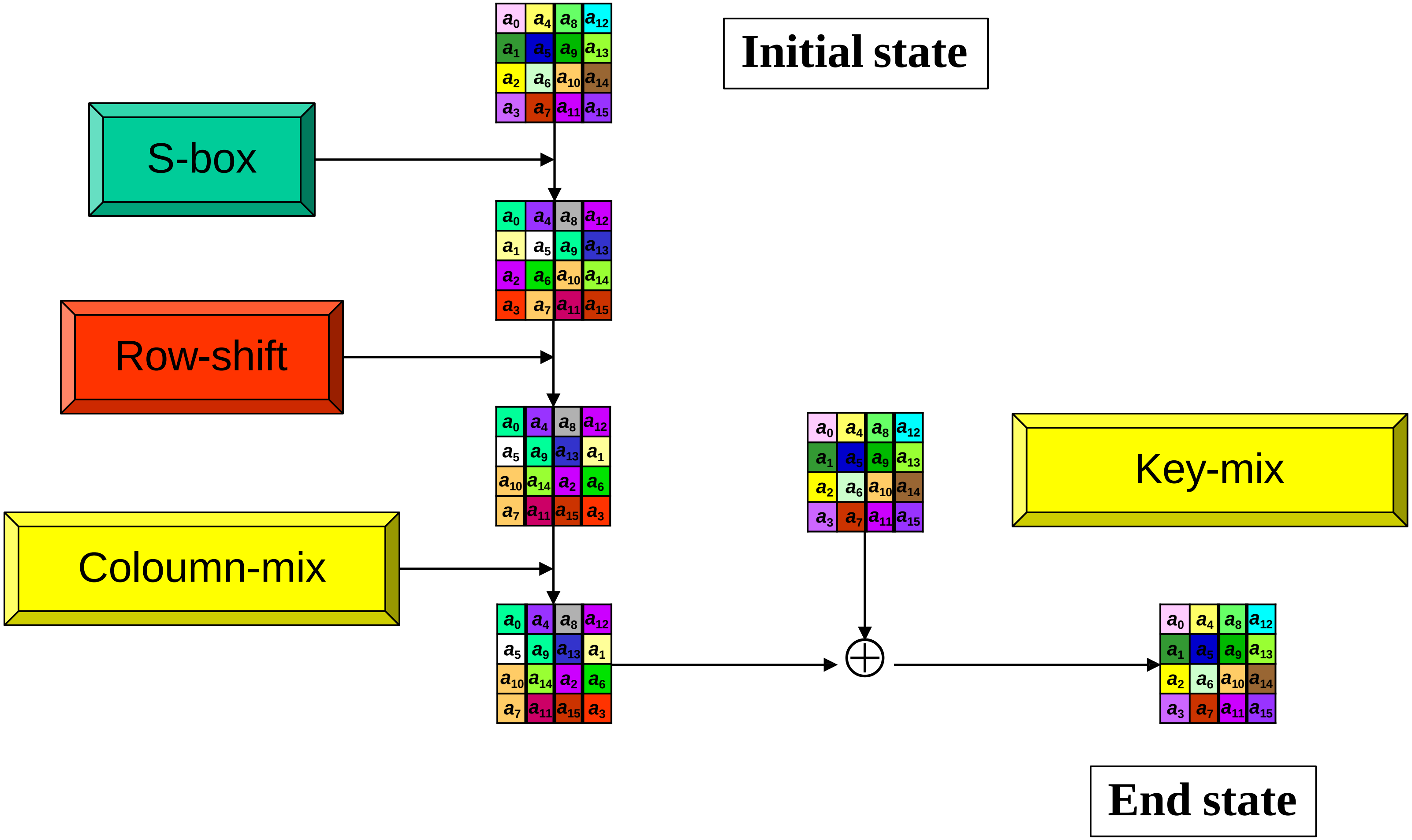


DESX

Advanced Encryption Standard

- Public competition to replace DES: because 56-bit keys and 64-bit data blocks no longer adequate.
- Rijndael nominated as the new Advanced Encryption Standard (AES) in 2001 [FIPS-197].
- Rijndael (pronounce as “Rhine-doll”) designed by Vincent Rijmen and Joan Daemen.
- 128-bit block size (Note error in Harris p. 809).
- 128-bit, 196-bit, and 256-bit key sizes.
- Rijndael is not a Feistel cipher.

Rijndael round function



$$\begin{matrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{matrix}$$

$$\begin{matrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{matrix}$$

$$\begin{matrix} a_0 & a_4 & a_8 & a_{12} \\ a_5 & a_9 & a_{13} & a_1 \\ a_{10} & a_{14} & a_2 & a_6 \\ a_7 & a_{11} & a_{15} & a_3 \end{matrix}$$

$$\begin{matrix} a_0 & a_4 & a_8 & a_{12} \\ a_5 & a_9 & a_{13} & a_1 \\ a_{10} & a_{14} & a_2 & a_6 \\ a_7 & a_{11} & a_{15} & a_3 \end{matrix}$$

$$\begin{matrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{matrix}$$

$$\begin{matrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{matrix}$$

Rijndael encryption

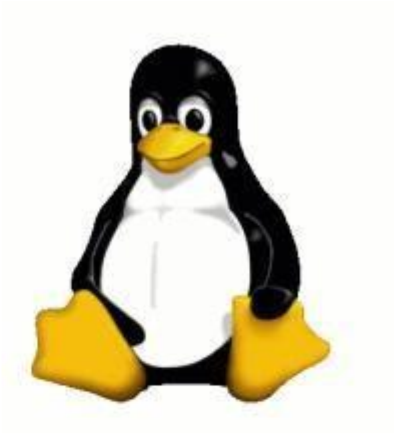
1. Key mix (round key K_0)
2. N_r-1 rounds containing:
    a) Byte substitution
    b) Row shift
    c) Coloumn mix
    d) Key mix (round key K_i)
3. Last round containing:
    a) Byte substitution
    b) Row shift
    c) Key mix (round key K_{Nr})

Key	Rounds
128	10
192	12
256	14

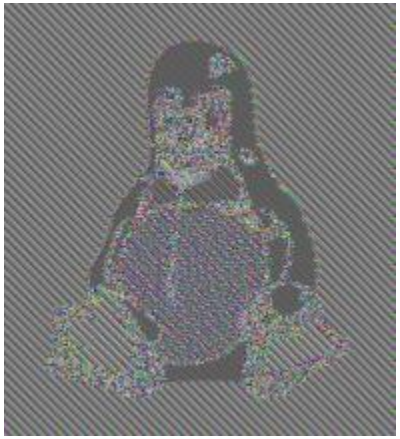
Block Ciphers: Modes of Operation

- Block ciphers can be used in different modes in order to provide different security services.
- Common modes include:
  - **E**lectronic **C**ode **B**ook (ECB)
  - **C**ipher **B**lock **C**haining (CBC)
  - **O**utput **F**eed**b**ack (OFB)
  - **C**ipher **F**eed**b**ack (CFB)
  - **C**oun**t**e**r** Mode (CTR)
  - **G**alois **C**ounter **M**ode (GCM) {Authenticated encryption}

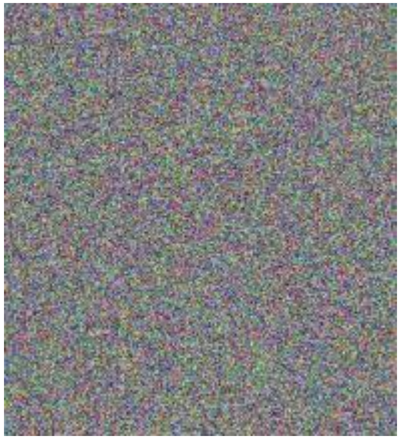
Use a secure mode!



Plaintext



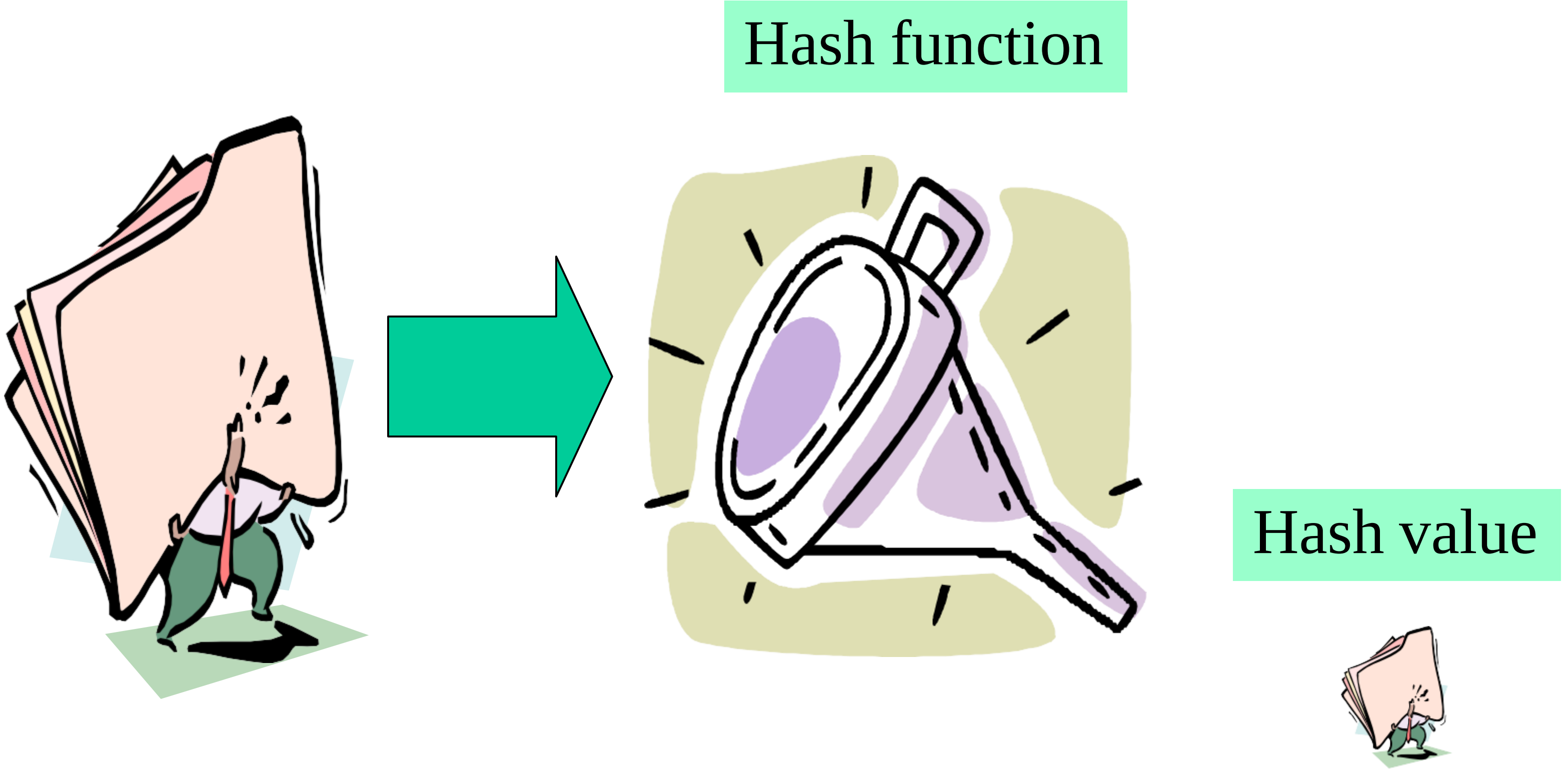
Ciphertext using ECB mode



Ciphertext using secure mode

Integrity Check Functions

Hash functions



Hash function

Hash value

Applications of hash functions

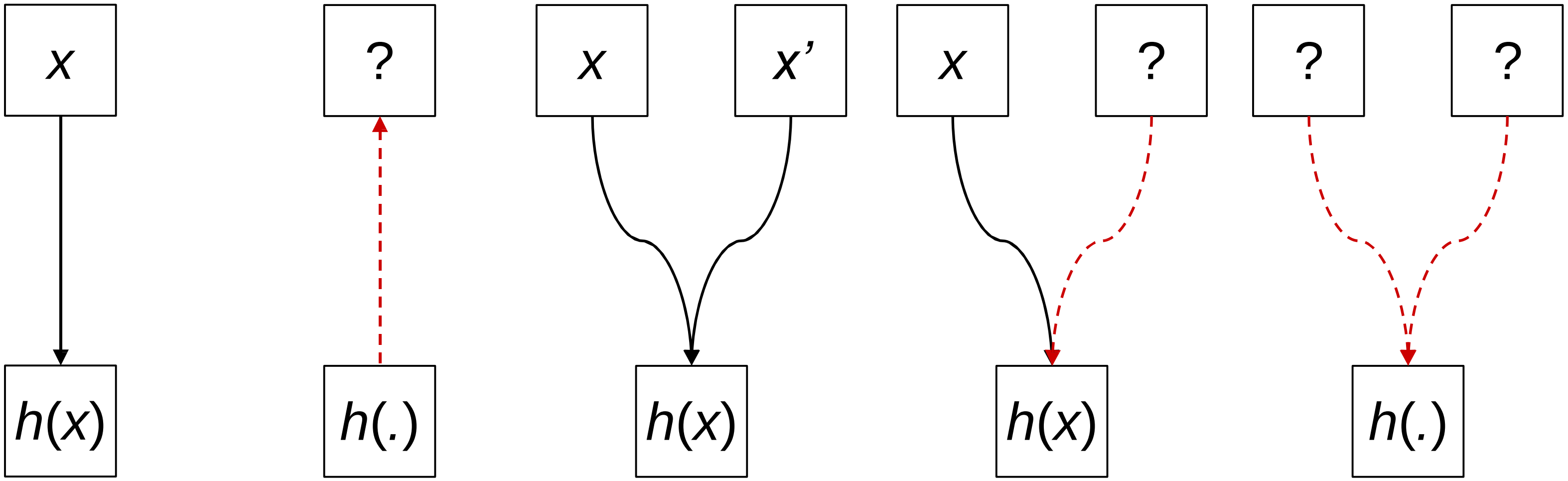
- Protection of password.
- Comparing files.
- Authentication of SW distributions.
- Bitcoin.
- Generation of Message Authentication Codes (MAC).
- Digital signatures.
- Pseudo number generation/Mask generation functions.
- Key derivation.

Hash functions (message digest functions)

Requirements for a one-way hash function h:

1. Ease of computation: given x, it is easy to compute $h(x)$.
2. Compression: h maps inputs x of arbitrary bitlength to outputs $h(x)$ of a fixed bitlength n.
3. One-way: given a value y, it is computationally infeasible to find an input x so that $h(x)=y$.
4. Collision resistance: it is computationally infeasible to find x and x', where $x \neq x'$, with $h(x)=h(x')$ (note: two variants of this property).

Properties of hash functions



Ease of computation

Pre-image resistance

Collision

Weak collision resistance (2nd pre-image resistance)

Strong collision resistance

Frequently used hash functions

- MD5: 128 bit digest. Broken. Often used in Internet protocols but no longer recommended.
- SHA-1 (Secure Hash Algorithm):160 bit digest. Potential attacks exist. Designed to operate with the US Digital Signature Standard (DSA);
- SHA-256, 384, 512 bit digest. Still secure. Replacement for SHA-1.
- RIPEMD-160: 160 bit digest. Still secure. Hash function frequently used by European cryptographic service providers.
- NIST competition for new secure hash algorithm, closed in 2012 with the winner:

A very good read about password hasing

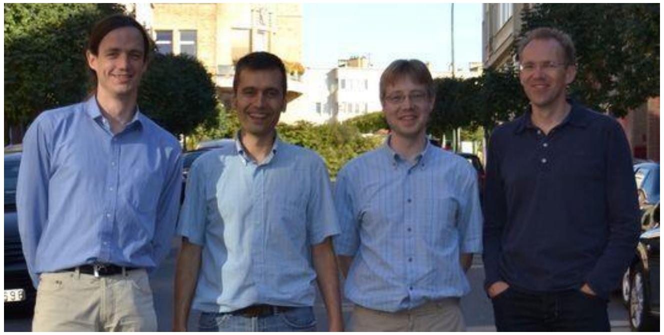
https://www.wordfence.com/learn/how-passwords-work-and-cracking-passwords/

Why MD5 is weaker than higher bits having hashing algorithms?

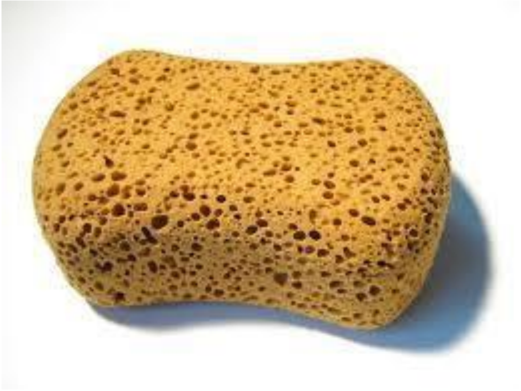
- It is easier to calculate MD5 hash of an input.
  - Therefore, you can guess more MD5 passwords
- Generating collisions is much easier on MD5
  - https://crypto.stackexchange.com/questions/1434/are-there-two-known-strings-which-have-the-same-md5-hash-value
  - https://en.wikipedia.org/wiki/Collision_attack

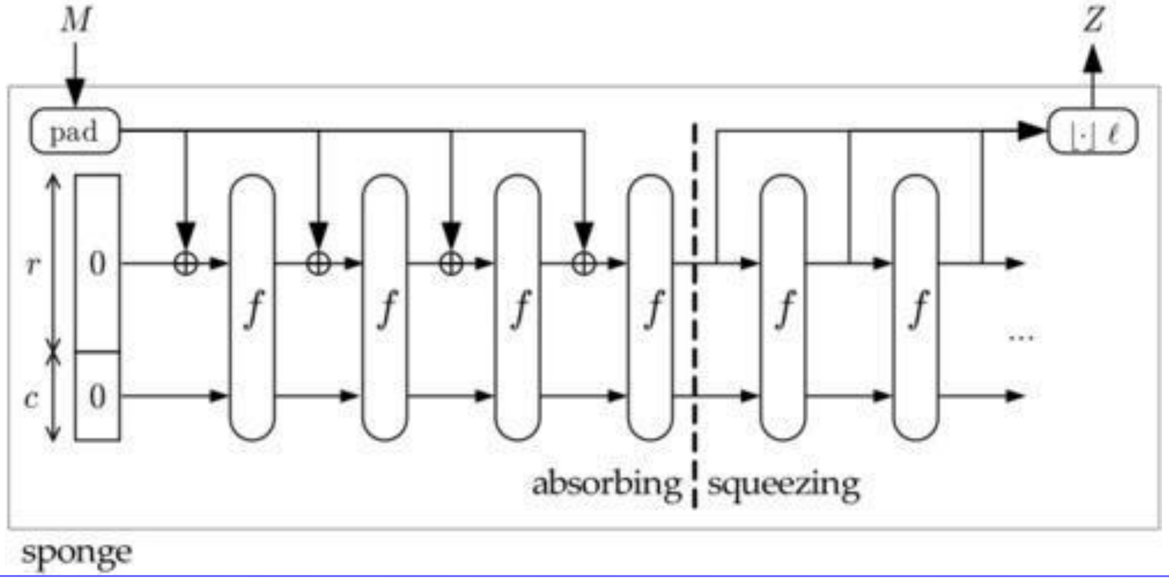
And the winner is?

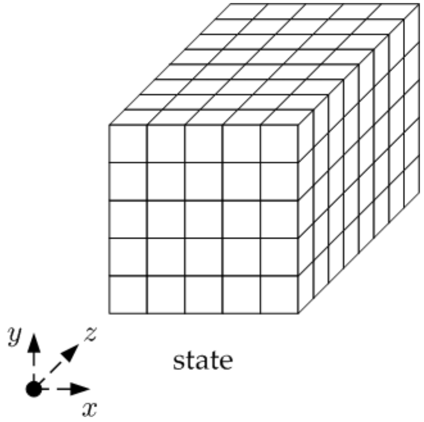
- NIST announced Keccak as the winner of the SHA-3 Cryptographic Hash Algorithm Competition on October 2, 2012, and ended the five-year competition.
- Keccak was designed by a team of cryptographers from Belgium and Italy, they are:
  - Guido Bertoni (Italy) of STMicroelectronics,
  - Joan Daemen (Belgium) of STMicroelectronics,
  - Michaël Peeters (Belgium) of NXP Semiconductors, and
  - Gilles Van Assche (Belgium) of STMicroelectronics.



Keccak and sponge functions







End of lecture