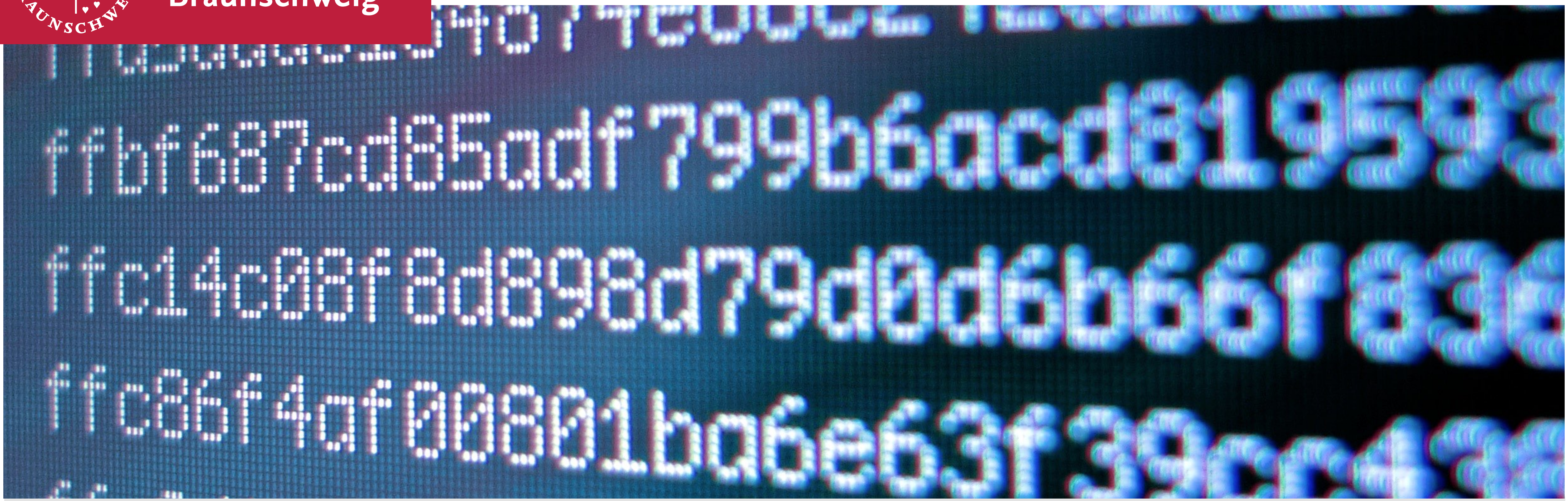
Technische Universität Braunschweig

IAS INSTITUTE FOR APPLICATION SECURITY



Symmetric-Key Cryptography

Vorlesung “Einführung in die IT-Sicherheit”

Prof. Dr. Martin Johns

Overview

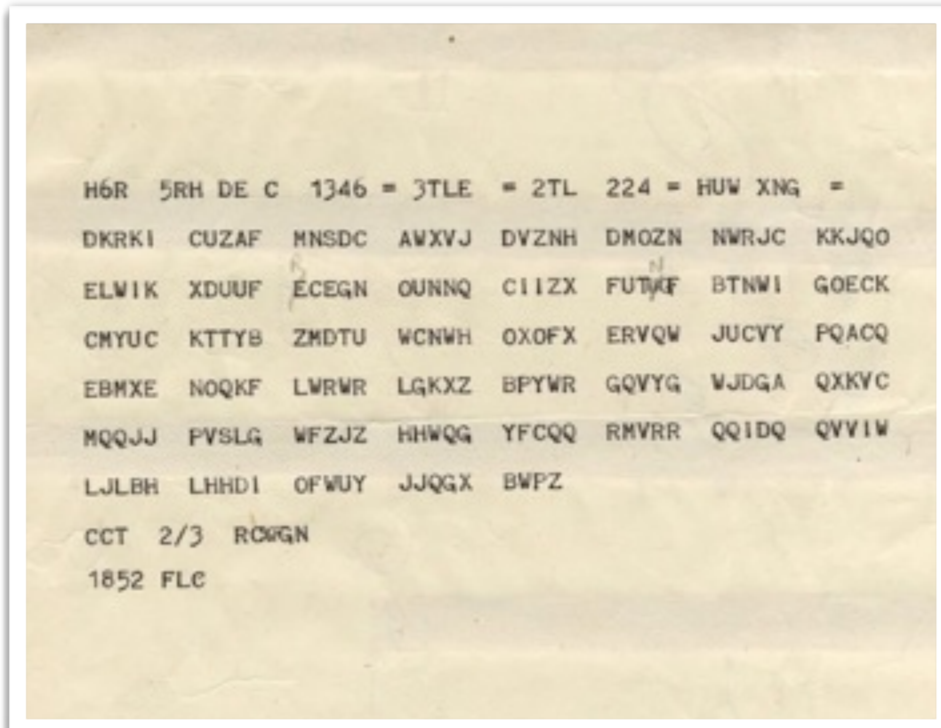
- **Topic of the unit**
  - Symmetric-key Cryptography
- **Parts of the unit**
  - Part #1: Basics of cryptography
  - Part #2: Classic ciphers
  - Part #3: Block and stream ciphers
  - Part #4: Block cipher modes

Cryptography

- **Cryptography** (kryptos: secret; graphein: writing)
  - Art and science of keeping information secure
  - Protection of confidentiality and integrity
- **Cryptanalysis** = study of attacks against cryptography
- **Steganography** (steganos: covered; graphein: writing)
  - Art and science of hiding information
  - Deniability and unobservability of communication

Examples

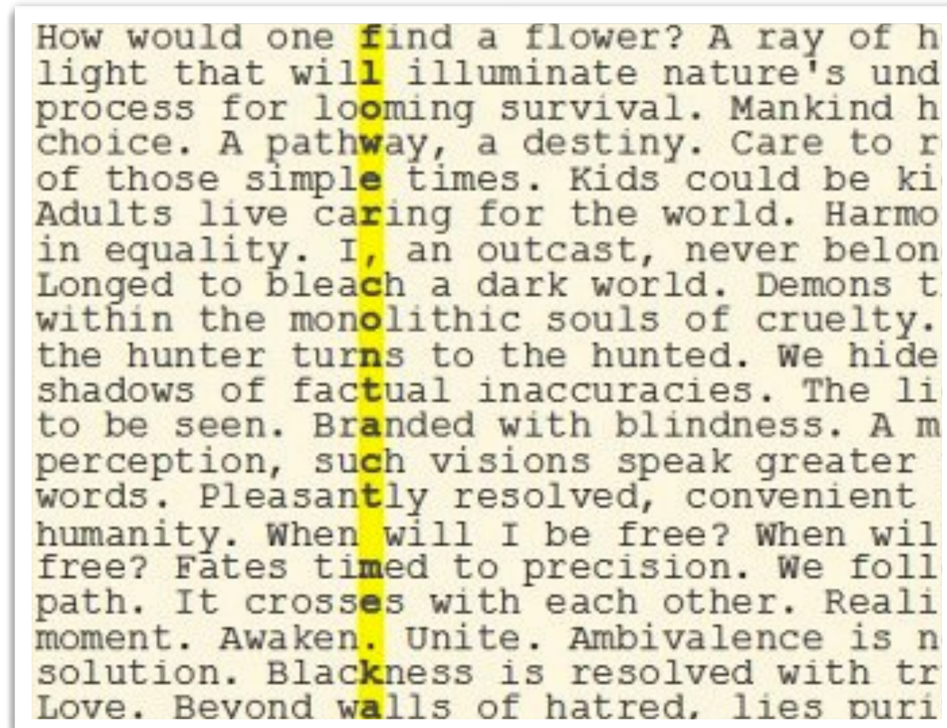
Cryptography



H6R 5RH DE C 1346 = 3TLE = 2TL 224 = HUW XNG =
DKRKI CUZAF MNSDC AWXVJ DVZNH DMOZN NWRJC KKJQO
ELWIK XDUUF ECEGN OUNNQ CIIZX FUTWF BTNWI GOECK
CMYUC KTTYB ZMDTU WCNWH OXOFX ERVQW JUCVY PQACQ
EBMXE NOQKF LWRWR LGKXZ BPYWR GQVYG WJDGA QXKVC
MQQJJ PVSLG WFZJZ HHWQG YFCQQ RMVRR QQIDQ QVVIW
LJLBH LHHDI OFWUY JJQGX BWPZ
CCT 2/3 RCWGN
1852 FLC

Message encrypted using the Enigma during WW2

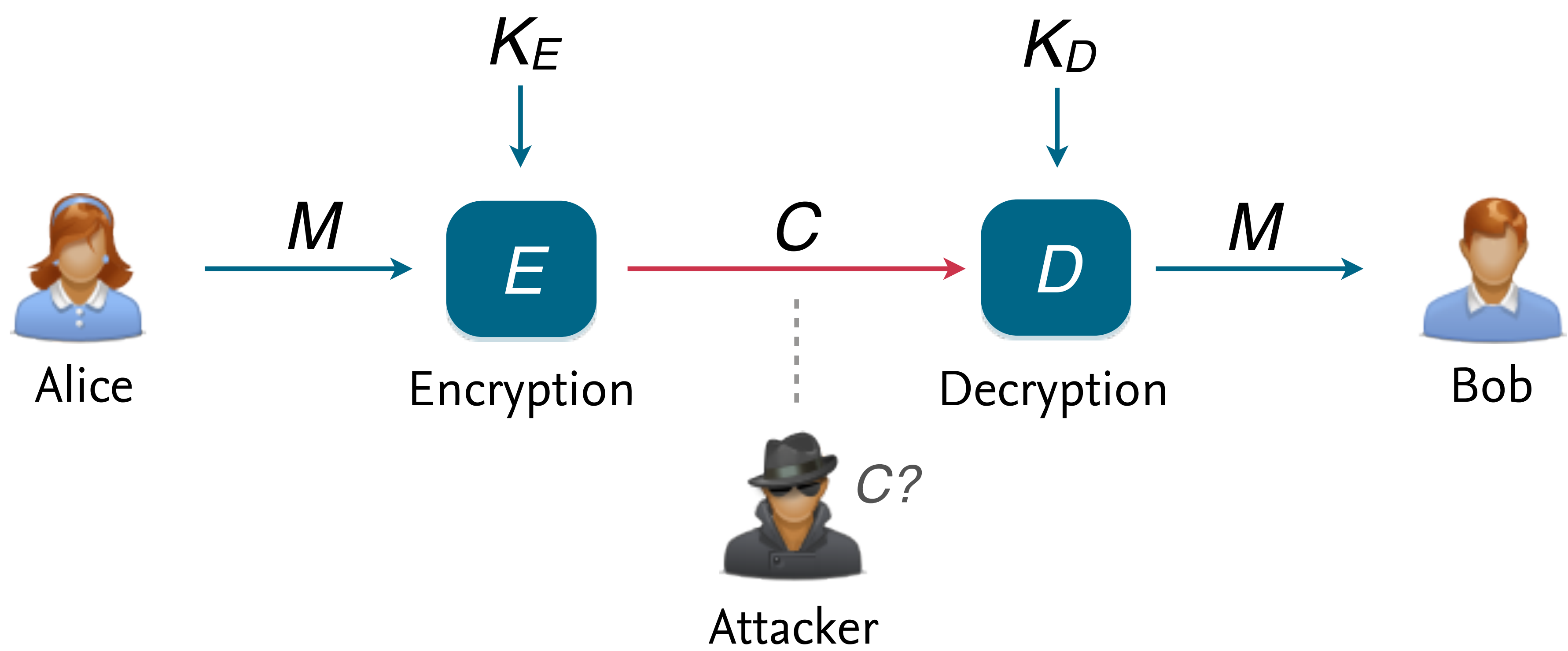
Steganography



How would one find a flower? A ray of h
light that will illuminate nature's und
process for looming survival. Mankind h
choice. A pathway, a destiny. Care to r
of those simple times. Kids could be ki
Adults live caring for the world. Harmo
in equality. I, an outcast, never belon
Longed to bleach a dark world. Demons t
within the monolithic souls of cruelty.
the hunter turns to the hunted. We hide
shadows of factual inaccuracies. The li
to be seen. Branded with blindness. A m
perception, such visions speak greater
words. Pleasantly resolved, convenient
humanity. When will I be free? When wil
free? Fates timed to precision. We foll
path. It crosses with each other. Reali
moment. Awaken. Unite. Ambivalence is n
solution. Blackness is resolved with tr
Love. Beyond walls of hatred, lies puri

Hidden message from the classic series "Heroes"

Cryptosystems



- **Cryptographic system for en/decrypting messages**
  - M = plaintext message C = ciphertext message
  - K_E = encryption key K_D = decryption key

More on Cryptosystems

- **Cipher** (encryption and decryption functions)
  - $E(M, K_E) = C$ and $D(C, K_D) = M$
- **Keyspace**
  - Set of possible values for the keys K_E and K_D
- **Symmetric-key cryptography**
  - $K_E = K_D$ (Sender and receiver use the same key K)
- Public-key cryptography ← next lecture
  - $K_E \neq K_D$ (Public key K_E and private key K_D)

Confusion and Diffusion

- **What makes a cryptographic cipher strong?**
  - That's a difficult question …
- **Confusion property**
  - Complex relation between key and plaintext/ciphertext
  - Getting K from M and C should be hard
- **Diffusion property**
  - Complex relation between plaintext and ciphertext
  - Getting M from C should be hard

Kerckhoffs's Principle

- **Kerckhoffs's Principle**
  - Assume that the cipher is known to the attacker
  - Security should depend on the key only (no obscurity)
- **What makes a cryptographic key strong?**
  - Set of possible keys (keyspace) very large

Testing all keys of a simple cipher (Brute-force attack)

Key size	My Laptop	1 Million Cores
32 bit	6 seconds	0 seconds
64 bit	850 years	7 hours
128 bit	10^{22} years	10^{16} years
256 bit	10^{59} years	10^{52} years

Some Attack Types

- **Classic attack types of cryptanalysis**
  - Ciphertext-only attack
    Attacks involving only ciphertext messages C
  - Known-plaintext and chosen-plaintext attack
    Attacks involving known/chosen plaintext M for C
- **Not so subtle attack types**
  - Brute-force attack
    Guessing of all possible keys K from the keyspace
  - Purchase-key attack
    Attacks involving blackmailing, theft and bribery

Estimating Risk

- **Model of attacker using resources**
  - Computation power, data complexity and time
- **Security level of a cryptosystem**
  - Unconditionally secure (also: absolute security)
    unbreakable with infinite resources and all possible attacks
  - Computationally secure (also: practical security)
    unbreakable with available resources and known attacks
- Risk trade-off: attack costs vs. possible profit

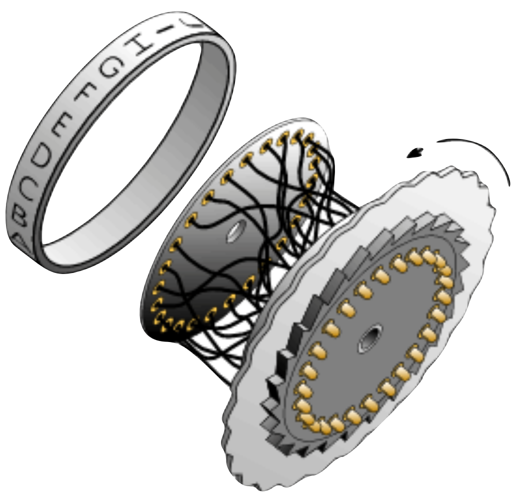
Overview

- **Topic of the unit**
  - Symmetric-key Cryptography
- **Parts of the unit**
  - Part #1: Basics of cryptography
  - Part #2: Classic ciphers
  - Part #3: Block and stream ciphers
  - Part #4: Block cipher modes

Ancient Ciphers

- **Simple substitution ciphers**
  - Rotate alphabet by k characters
  - Examples: Caesar cipher, ROT13
  - Keyspace ridiculously small

- **Monoalphabetic substitution ciphers**
  - Permute characters of alphabet
  - Keyspace significantly larger
  - Character frequencies are preserved



(Fig. from cryptomuseum.com)

Vigenère Cipher

- **Popular polyalphabetic substitution cipher**
  - Also known as “le chiffre indéchiffrable” ;-)
  - Combination of multiple simple substitution ciphers
  - Rotations determined by a word (key)
  - Easy to break: Kasiski and Friedman tests

Message	T	H	I	S	A	T	E	S	T
Running key	K	E	Y	K	E	Y	K	E	Y
	+10	+4	+24	+10	+4	+24	+10	+4	+24
Ciphertext	D	L	G	C	E	R	O	W	R

Vigenère Cipher

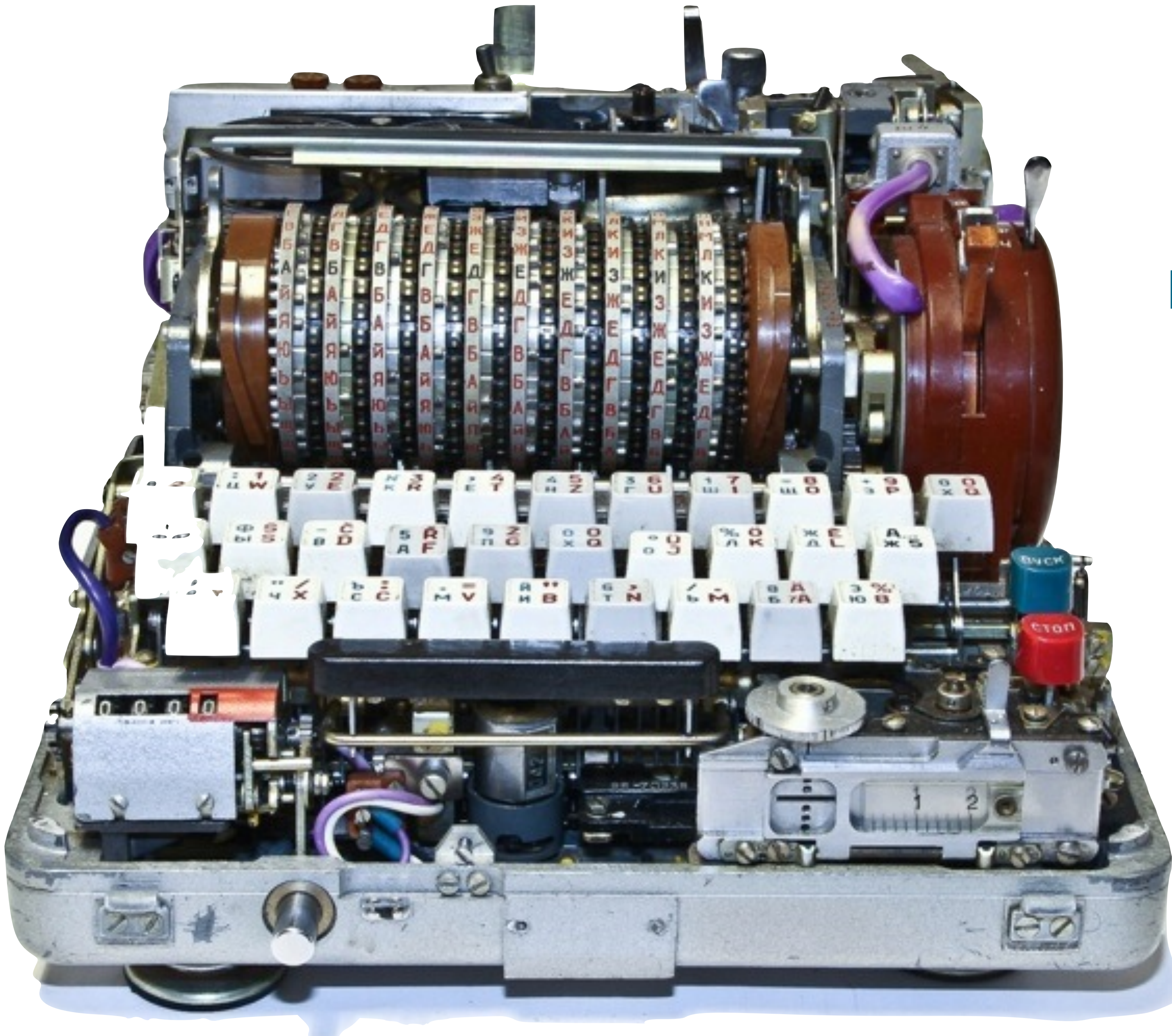
- **Popular polyalphabetic substitution cipher**
  - Also known as “le chiffre indéchiffrable” ;-)
  - Combination of multiple simple substitution ciphers
  - Rotations determined by a word (key)
  - Easy to break: Kasiski and Friedman tests

Message	T	H	I	S	A	T	E	S	T
Running key	K	E	Y	K	E	Y	K	E	Y
	+10	+4	+24	+10	+4	+24	+10	+4	+24
Ciphertext	D	L	G	C	E	R	O	W	R

Period

Polyalphabetic

Rotor Machines



Russian
Rotor Machine
M-125
(Fialka)

(Fig. from cryptomuseum.com)

One-Time Pad

- **XOR ciphers** (Variant of Vigenère Cipher)
  - $E(M, K) = M \oplus K = C$
  - $D(C, K) = C \oplus K = M$

Why does this work?

$$M \oplus (K \oplus K) = M \oplus 0 = M$$

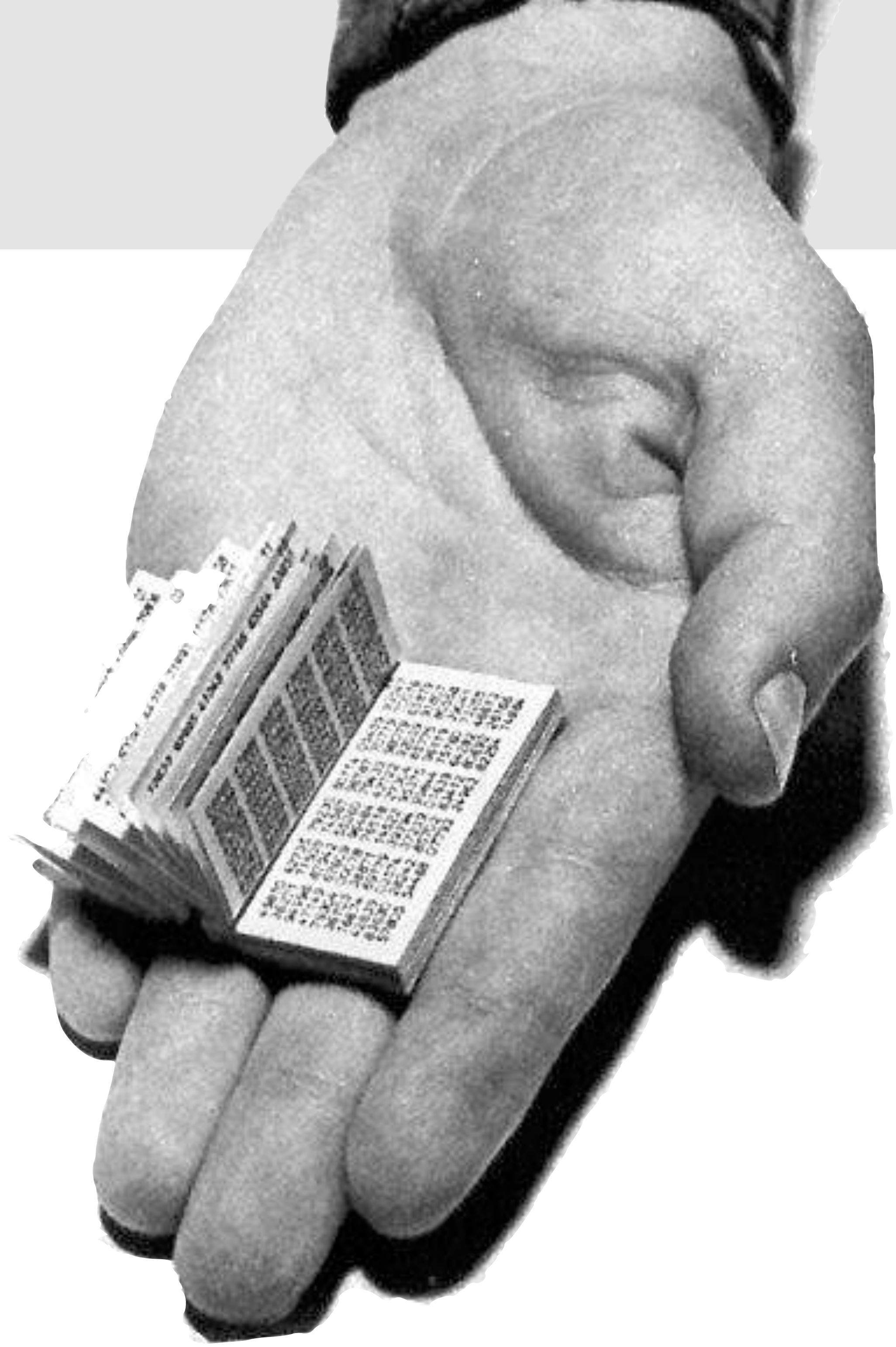
- **One-time pad** = XOR cipher with constraints
  1. Key length equals message length
  2. Key bits are truly random (not pseudo-random)
  3. Key is used only once and destroyed

Security of One-Time Pad

- **Randomness and XOR**
  - Let K be a random bit with $Pr(K = 1) = 0.5$
  - For any bit M holds $Pr(M \oplus K = 1) = 0.5$
- **Effect on security of one-time pad**
  - Given ciphertext $C = M \oplus K$, each plaintext equally likely
    $\Rightarrow$ unconditionally secure (Shannon 1949)
- **Example:** $C = 01$
  - $M = 00$ for $K = 01$ $\quad$ $M = 01$ for $K = 00$
    $M = 10$ for $K = 11$ $\quad$ $M = 11$ for $K = 10$

One-Time Pad in Practice?

- **Intelligence and military services**
  - Regular use during cold war
- **Major problems**
  - Key exchange difficult
  - True randomness required
- **Not very practical today**
  - Inspiration for other methods, e.g. stream ciphers



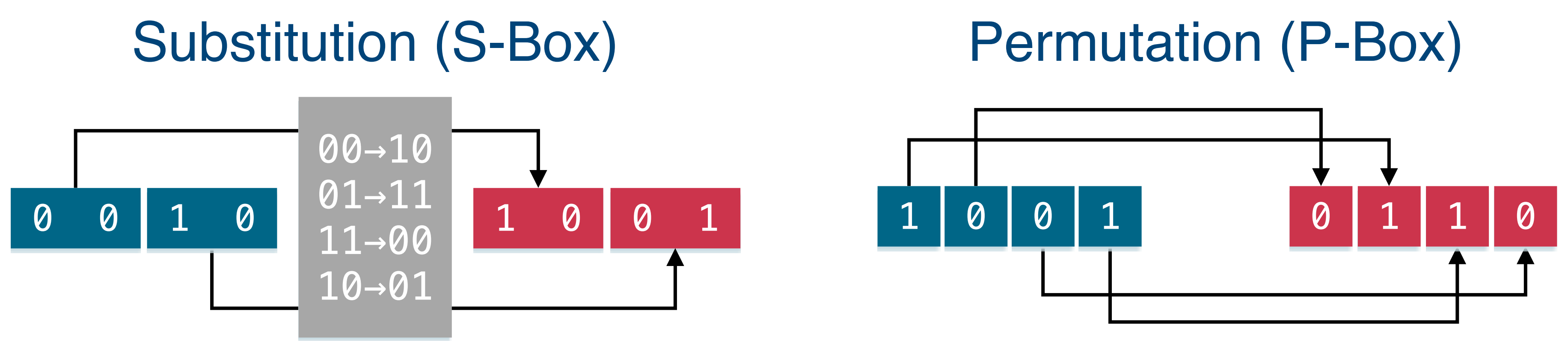
One-time pad as used by the Russian KGB

Overview

- **Topic of the unit**
  - Symmetric-key Cryptography
- **Parts of the unit**
  - Part #1: Basics of cryptography
  - Part #2: Classic ciphers
  - Part #3: Block and stream ciphers
  - Part #4: Block cipher modes

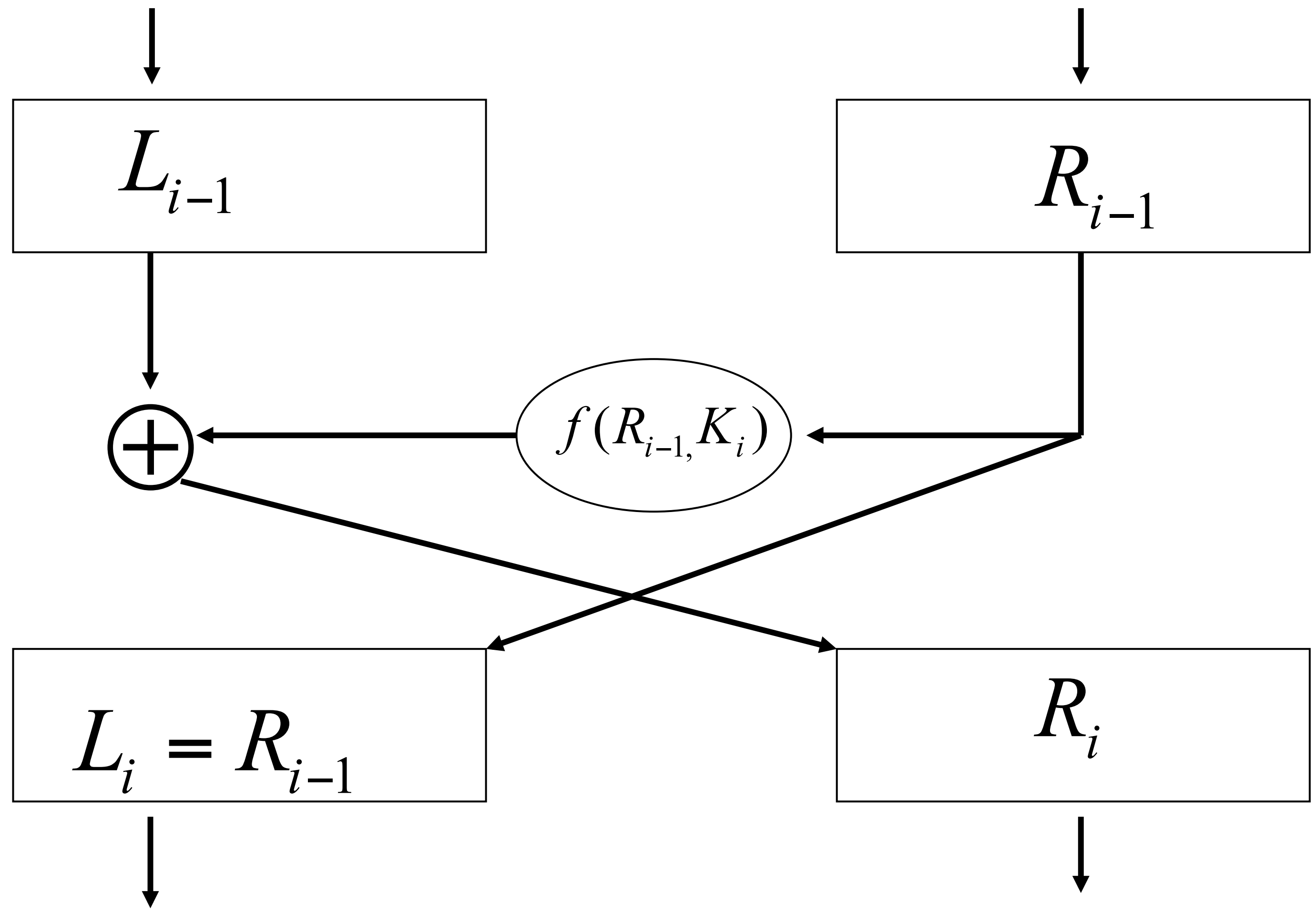
Modern Symmetric Ciphers

- **Modern ciphers**
  - Sophisticated design of substitutions and permutations
  - Often round-based encryption and decryption algorithms
  - Trade-off: security vs. efficiency and portability
- **Common building blocks**



Example: Data Encryption Standard (DES)

- 16 Rounds
- 56 bit key used to generate 48 bit subkeys
- S-Box-based function f(x,y)



Block and Stream Ciphers

- **Block ciphers**
  - Encryption and decryption of fixed-size blocks
  - Examples: AES, Serpent, Twofish, IDEA and DES

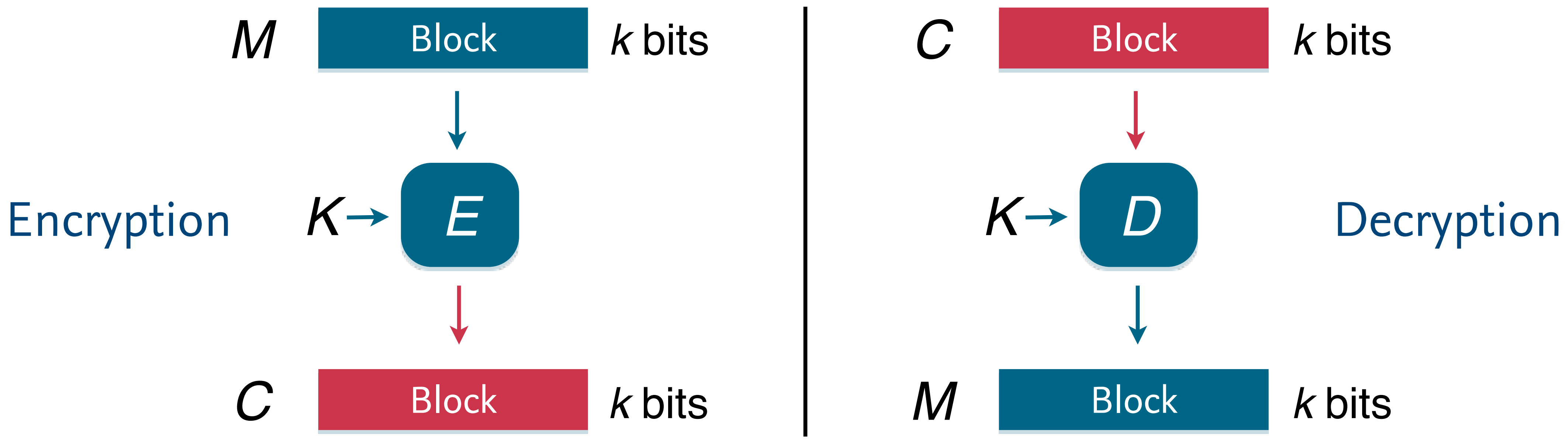
M `0 0 1 0` `1 0 1 0` … → C `1 1 1 0` `0 0 1 1` …

- **Stream ciphers**
  - Encryption and decryption of bit streams
  - Examples: RC4, A5/1, Rabbit and Salsa20

M `0` `0` `1` `0` `1` `0` `1` `0` … → C `1` `1` `1` `0` `0` `0` `1` `1` …

Block Ciphers

- **Block ciphers**
  - Encryption and decryption in blocks (e.g., 64 or 128 bit)
  - Padding of short messages, splitting of long messages
  - Different modes of operations: ECB, CBC, OFB, CTR, ...



Example: **Advanced Encryption Standard (AES)**

- **Standardized block cipher** (also known as Rijndael)
  - Developed by Belgian cryptographs Daemen and Rijmen
  - Winner of public competition by NIST in 2000
  - Secure but also efficient in software and hardware
- **Block cipher with 128 bit**
  - Substitution-permutation network (S-Box & P-Box)
  - Key size: 128, 192 and 256 bits
  - Rounds: 10, 12 and 14 depending on key size
- So far only impractical attacks known against AES

Example: **Advanced Encryption Standard (AES)**

- **Standardized block cipher** (also known as Rijndael)
  - Developed by Belgian cryptographs Daemen and Rijmen
  - Winner of public competition by NIST in 2000
  - Secure but also efficient in software and hardware

- **Block cipher with 128 bit**
  - Substitution-permutation network (S-Box & P-Box)
  - Key size: 128, 192 and 256 bits
  - Rounds: 10, 12 and 14 dependin

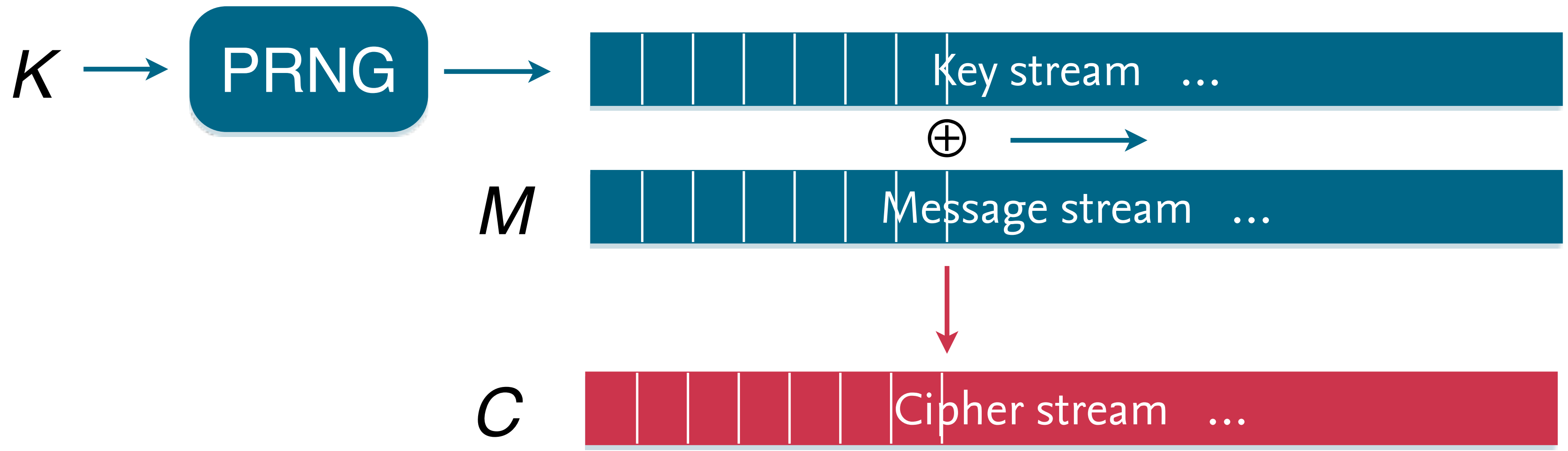
- So far only impractical attacks kno

```
# Python
from Crypto.Cipher import AES
aes = AES.new(key, AES.MODE_CBC, iv)
crypt = aes.encrypt(msg)

aes = AES.new(key, AES.MODE_CBC, iv)
msg = aes.decrypt(crypt)
```

Stream ciphers

- **Stream ciphers**
  - Bit-wise encryption and decryption of data
  - Application of pseudo-random number generator (PRNG)
  - Security usually dependent on quality of PRNG



Example: **RC4 Cipher**

- **Common stream cipher**
  - Developed by Ron Rivest for RSA Security
  - Leaked to the public in 1994 (ARC4 = Alleged RC4)
- **Cipher design**
  - Key size: 40 to 256 bits
  - Key-scheduling algorithm initializing substitution S-box
  - Keystream computed by swapping elements in S-box
- Some known weaknesses, e.g., in WEP implementation

Example: **RC4 Cipher**

- **Common stream cipher**
  - Developed by Ron Rivest for RSA Security
  - Leaked to the public in 1994 (ARC4 = Alleged RC4)
- **Cipher design**
  - Key size: 40 to 256 bits
  - Key-scheduling algorithm initializin… S-box
  - Keystream computed by swapping …
- Some known weaknesses, e.g., in WE…

```
# Python
from Crypto.Cipher import ARC4
arc4 = ARC4.new(key)
crypt = arc4.encrypt(msg)

arc4 = ARC4.new(key)
msg = arc4.decrypt(crypt)
```

Further Ciphers

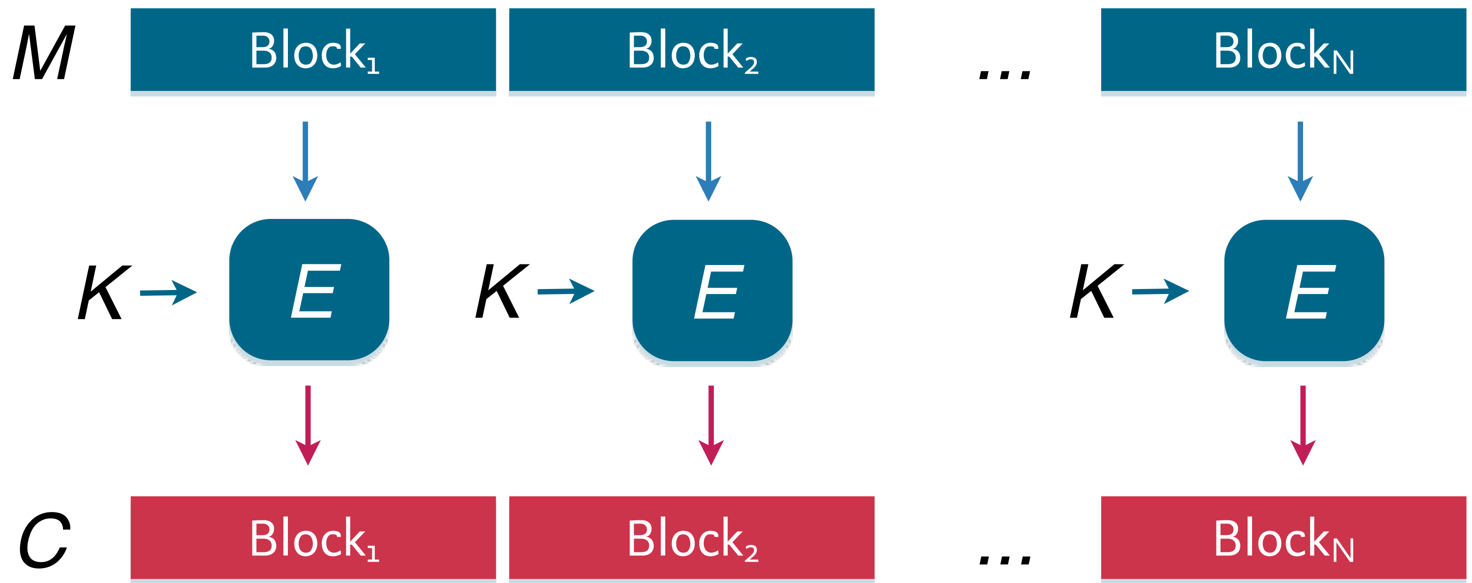
- **Stream ciphers: GSM Standard**
  - A5/1 (1987) and A5/2 (1989)
- **Block ciphers: AES contest finalists**
  - Serpent by R. Anderson, E. Biham and L. Knudsen (1998)
  - Twofish by B. Schneier (1998)
- **Stream ciphers: eSTREAM candidates**
  - Salsa20 by D. J. Bernstein (2004)
  - ...

Overview

- **Topic of the unit**
  - Symmetric-key Cryptography
- **Parts of the unit**
  - Part #1: Basics of cryptography
  - Part #2: Classic ciphers
  - Part #3: Block and stream ciphers
  - Part #4: Block cipher modes

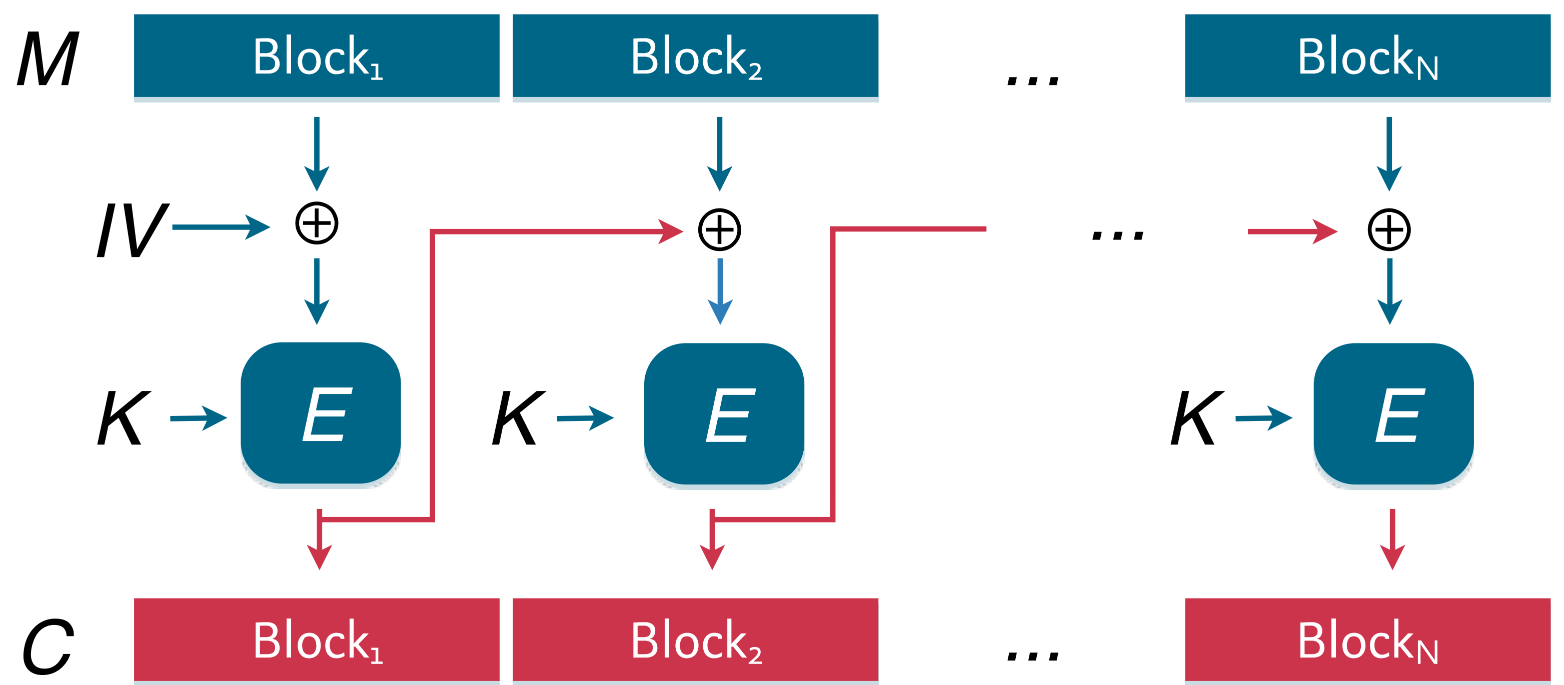
Electronic Code Book

- Mode: **Electronic Code Book (ECB)**
  - Independent encryption and decryption of message blocks
  - Simple and efficient (concurrent) implementation
  - Vulnerable to known-plaintext and replay attacks



Cipher-Block Chaining

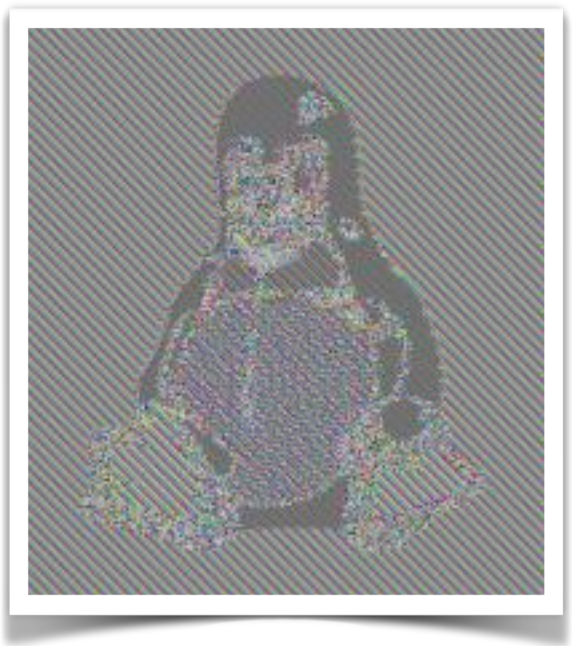
- Mode: **Cipher-Block Chaining (CBC)**
  - Chaining of cipher blocks using XOR operator
  - (Largely) resistant against known-plaintext and replay attacks



ECB vs CBC

Original image

ECB encryption

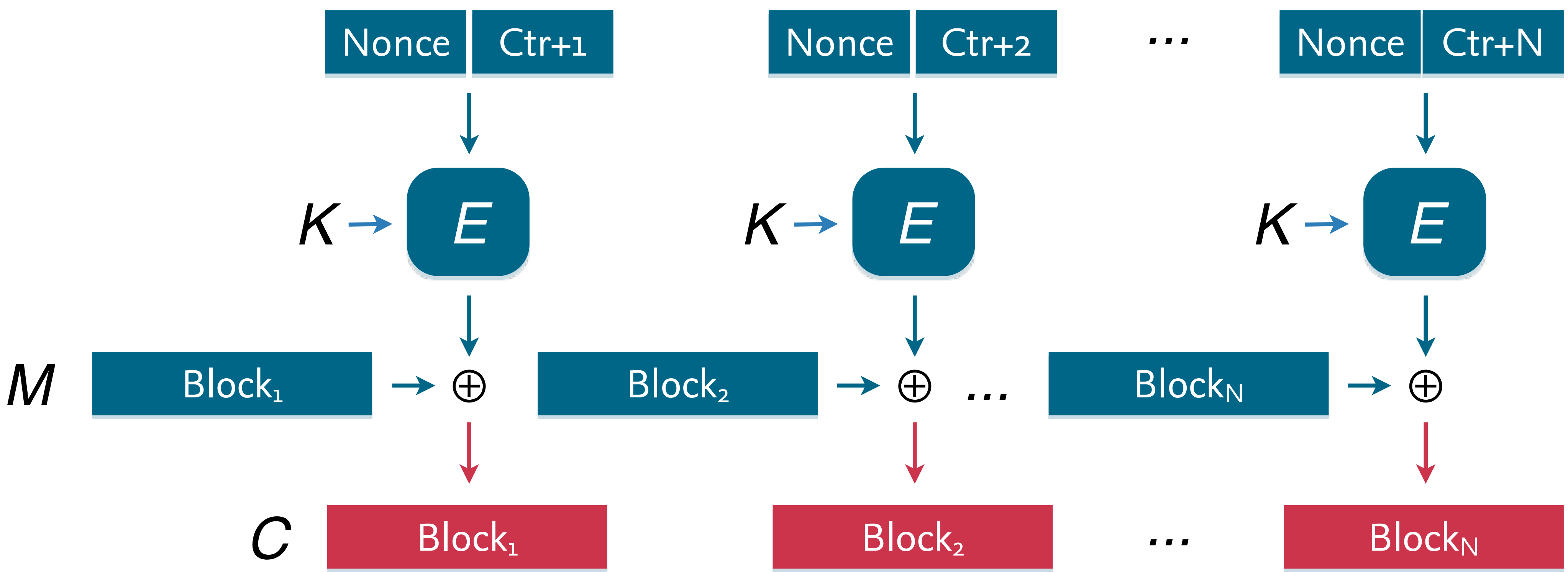


CBC Encryption

Taken from Wikipedia:
Block cipher modes of operation

Counter Mode

- Mode: **Counter Mode (CTR)**
  - Block cipher used as stream cipher
  - Random access to blocks (synchronization) still possible



Summary

Summary

- **Cryptography** (“keeping information secure”)
  - Encryption and decryption of messages
  - Security should depend on key, not algorithm
- **Symmetric-key cryptosystems**
  - Sender and receiver use same key
  - Different cipher types (block and stream cipher)
  - Mode of operation dependent on application