Introducere in criptografie

Marios Choudary

What is cryptography ?

- Greek:
  - crypto: hide, make secret (RO: a ascunde)
  - Γραφω [Grafo]: to write (RO: a scrie)
- In older cryptographic systems:
  - “the *art* of secret writing”
- In modern cryptography:
  - “the *science* of secret writing”

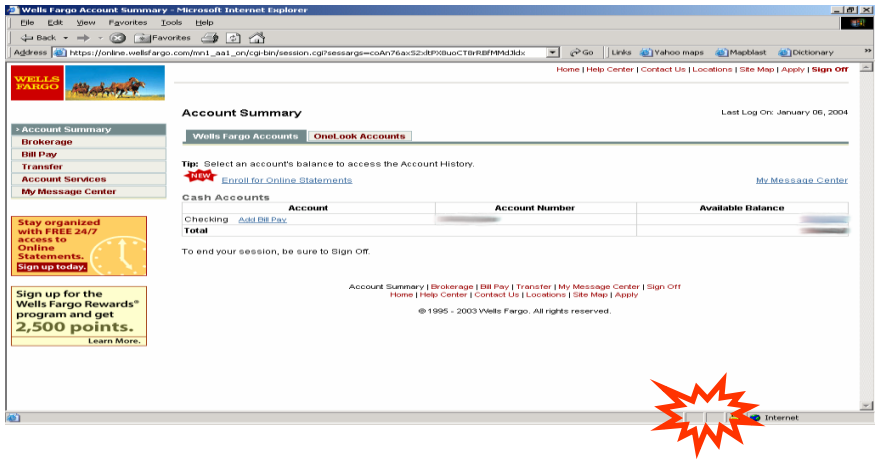
What is cryptography ?

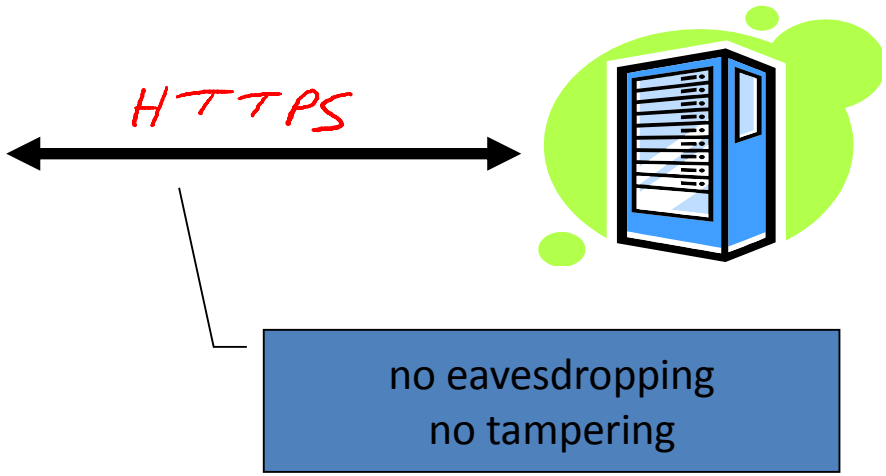
- Cryptography, cryptology, cryptanalysis
  - Some consider cryptology = cryptography + cryptanalysis
  - Cryptanalysis : art/science of analysing (often breaking) the security of cryptographic systmes
  - Modern cryptographic algorithms often have a strong cryptanalysis performed

Exemple

APLICATII ALE CRYPTOGRAFIEI

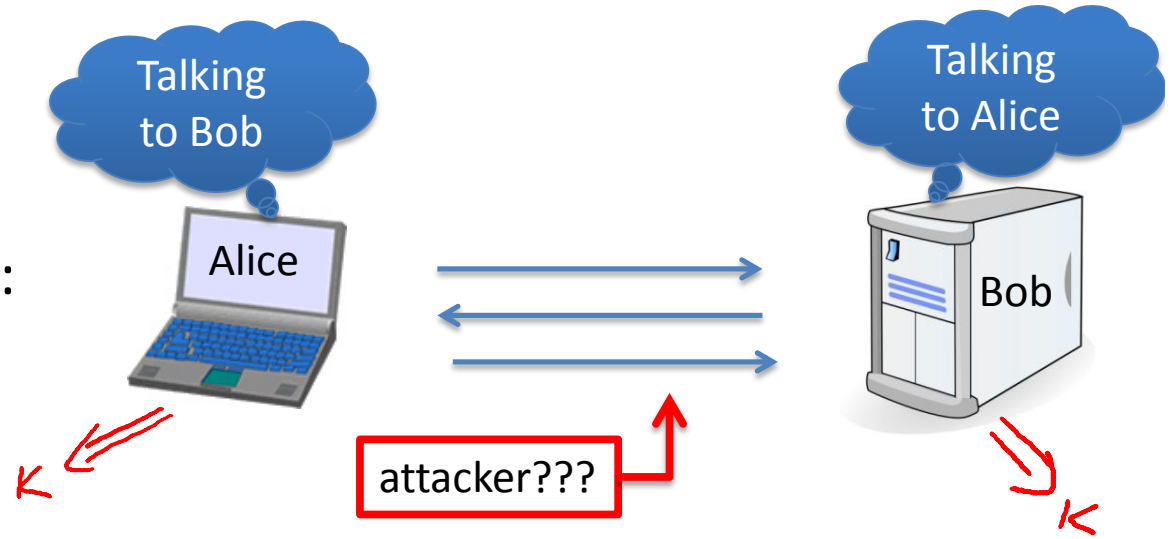
Secure communication





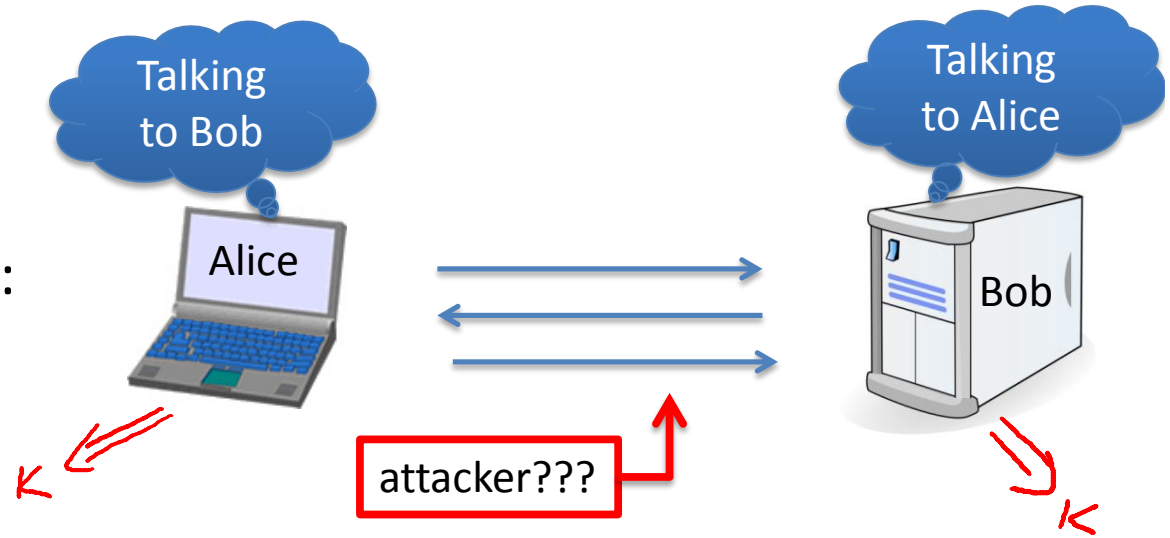
Crypto core

Secret key establishment:

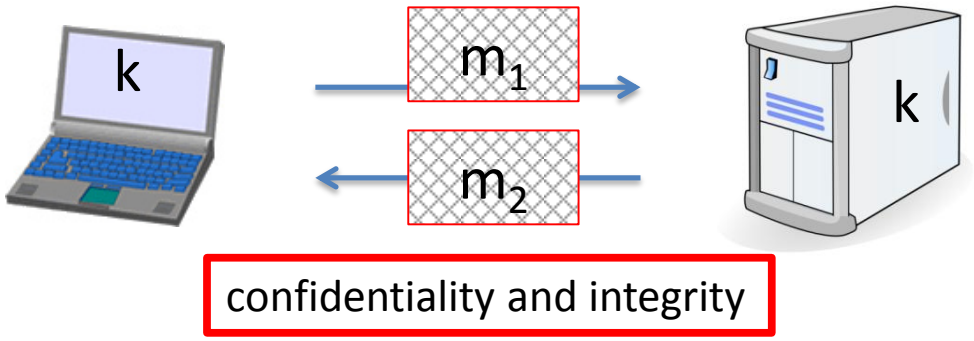


Crypto core

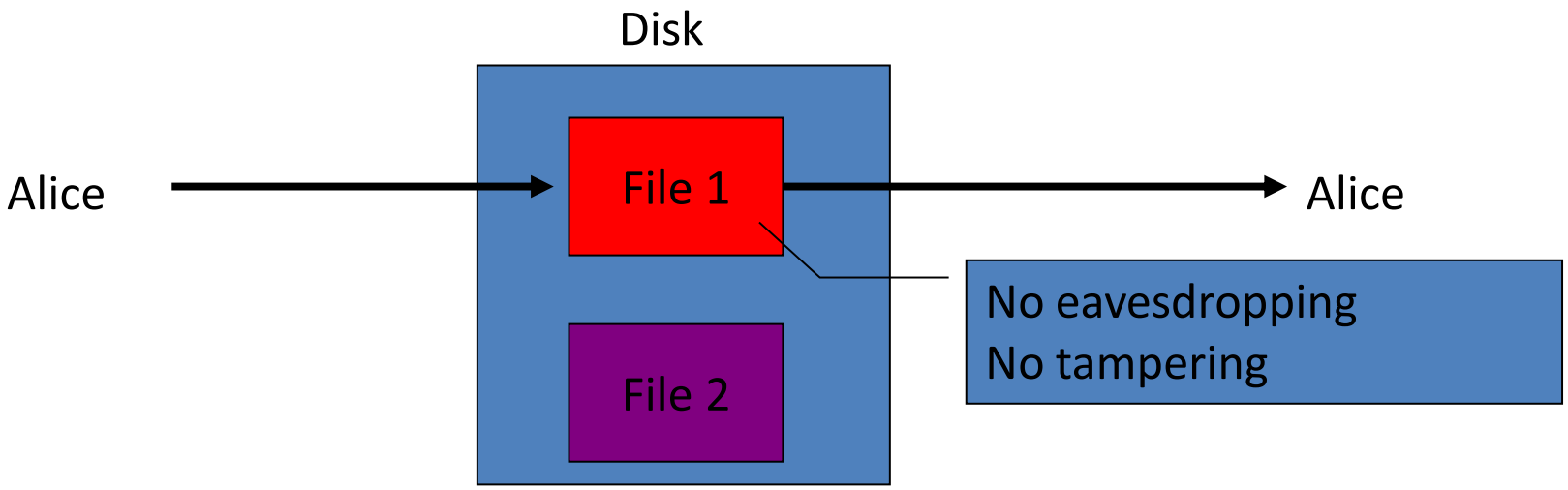
Secret key establishment:



Secure communication:



Protected files on disk



Analogous to secure communication:

Alice today sends a message to Alice tomorrow

EMV online authorisation

K K

D={Amount, Country, Date, UN, …}

REQ={UN,ATC,IAD,…}, AUTH REQ=MAC$_K$(D, ATC, IAD)

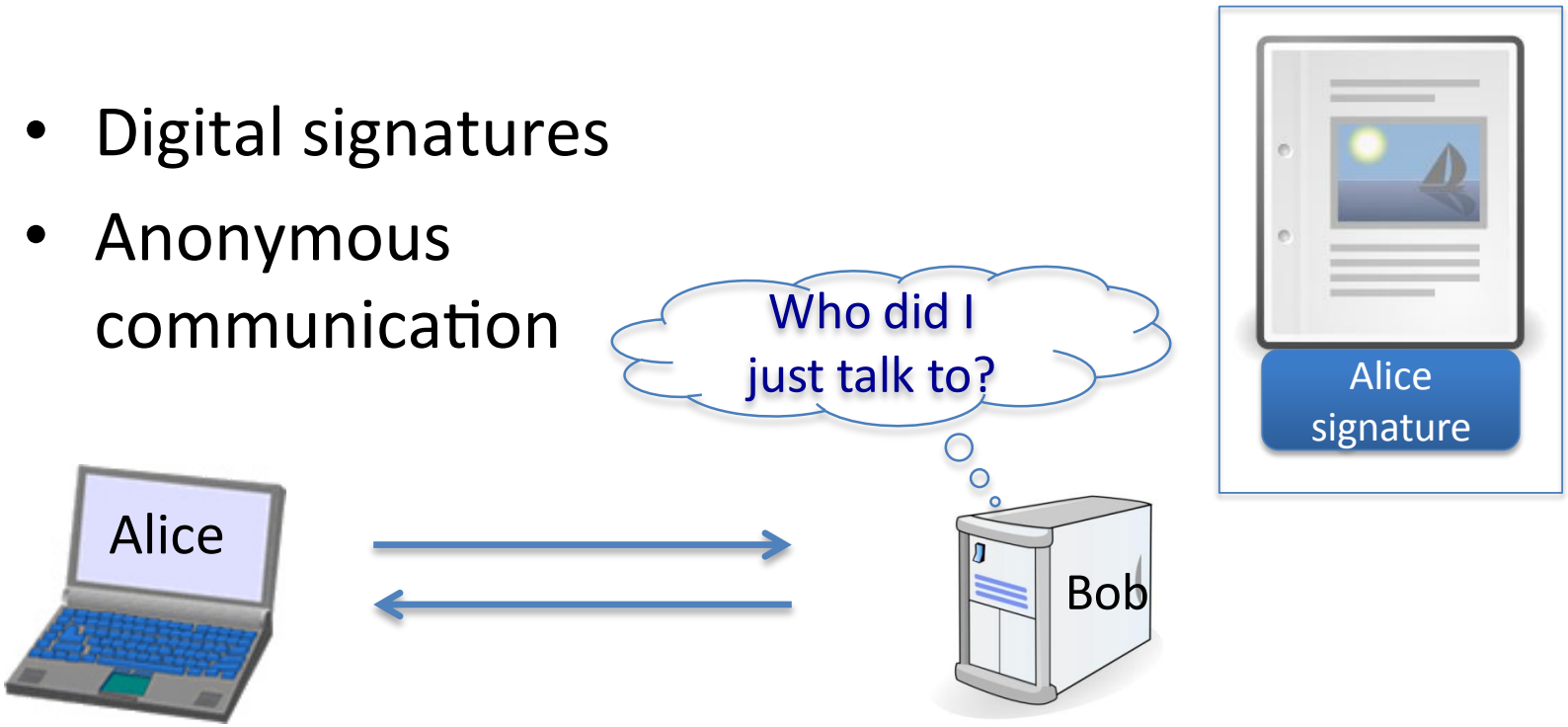
RESP={OK/BAD}, AUTH RESP=MAC$_K$(RESP, AUTH REQ,…)

Fancier applications

- Digital signatures

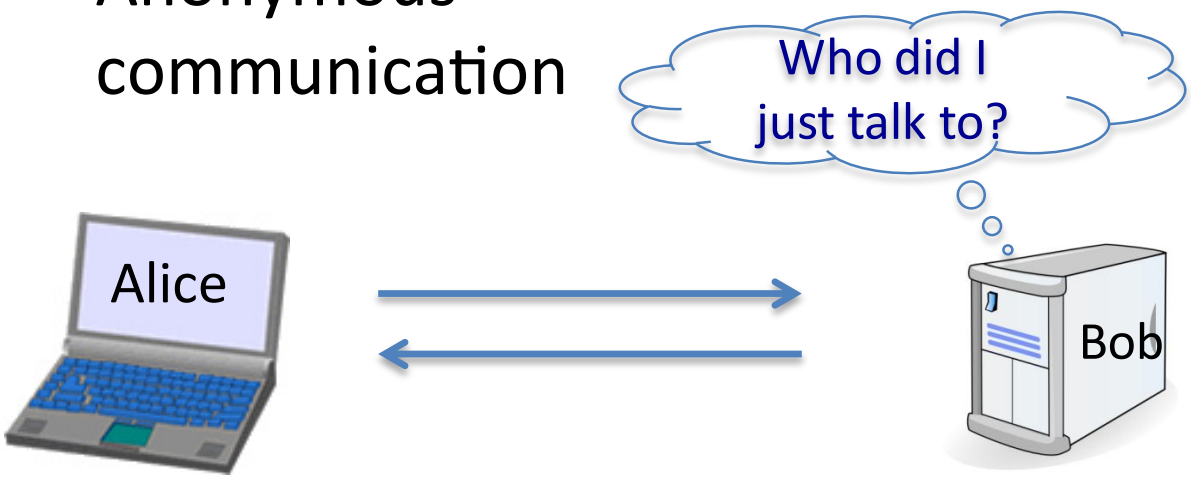
Fancier applications

- Digital signatures
- Anonymous communication



Fancier applications

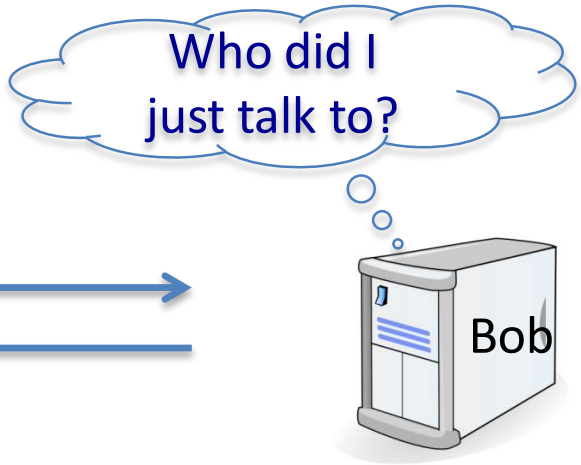
- Digital signatures
- Anonymous communication



- Digital cash (bitcoin)

Fancier applications

- Digital signatures
- Anonymous communication



- Digital cash (bitcoin)
- E-voting, etc.

Things to remember

Cryptography is:

- A tremendous tool
- The basis for many security mechanisms

Cryptography is not:

- The solution to all security problems
- Reliable unless implemented and used properly
- Something you should try to invent yourself
    - many many examples of broken ad-hoc designs

Vedeti “The code breakers” de David Kahn

EXEMPLE ISTORICE

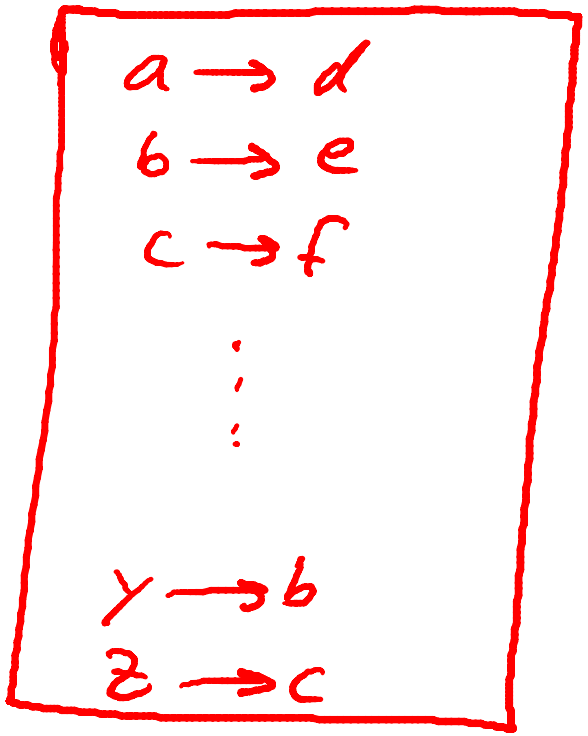
Atbash: criptografie din istoria lui Israel

- Ieremia 25, 17-26: “Și am luat cupa din mâna Domnului și am dat să bea tuturor neamurilor la care m-a trimis Domnul: [...] iar regele **Șișacului** va bea după ei.”
- Șișac (ebr: Sheshakh) = Babilon cu litere schimbate; prima cu ultima, a doua cu penultima, etc.

More popular: Caesar cipher

shift by 3:

(or K)

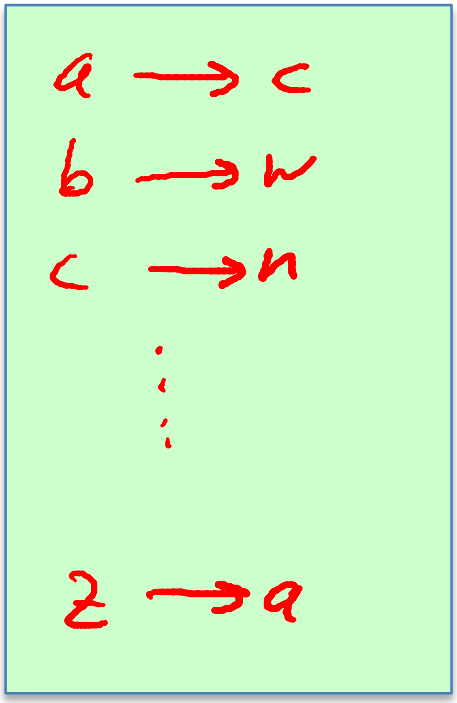


Enc(k, c) = c + k mod 26

Dec(k, c) = c - k mod 26

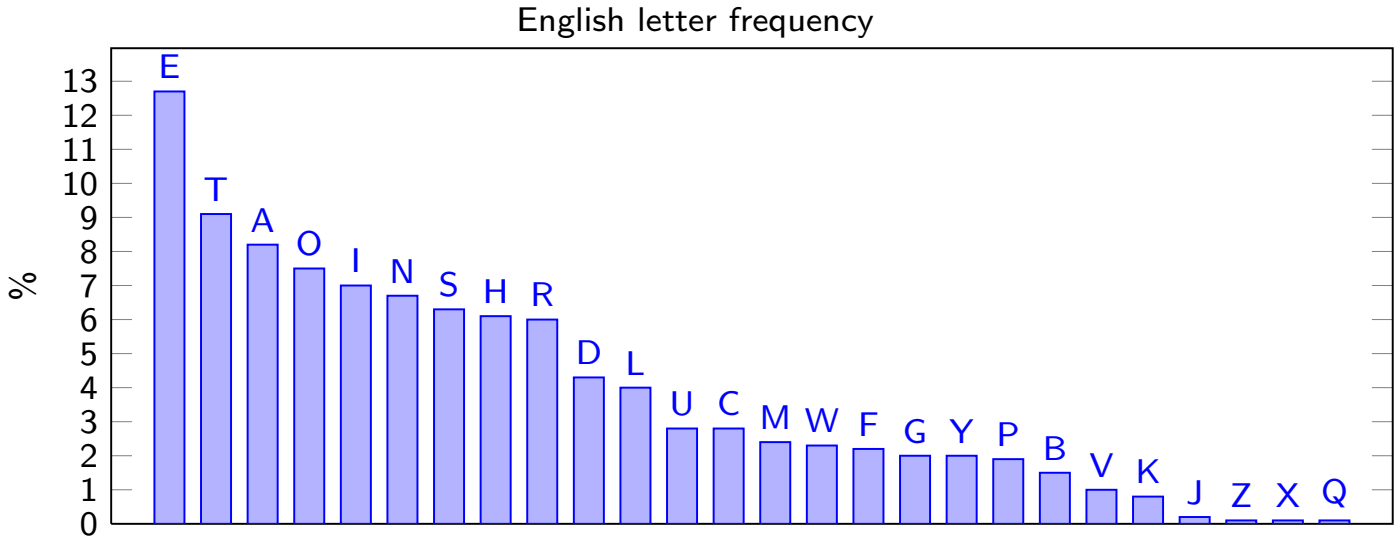
Substitution cipher

k :=



Key size: $26! \approx 2^{88}$

Statistical properties of plain text



The most common letters in English:
E, T, A, O, I, N, S, H, R, D, L, U, C, M, W, F, G, Y, P, B, V, K, J, ...

The most common digrams in English:
TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, ...

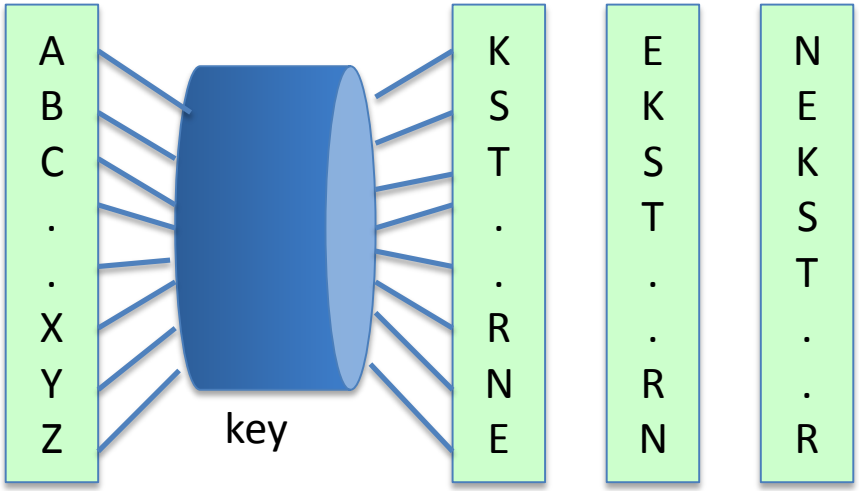
The most common trigrams in English:
THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, ...

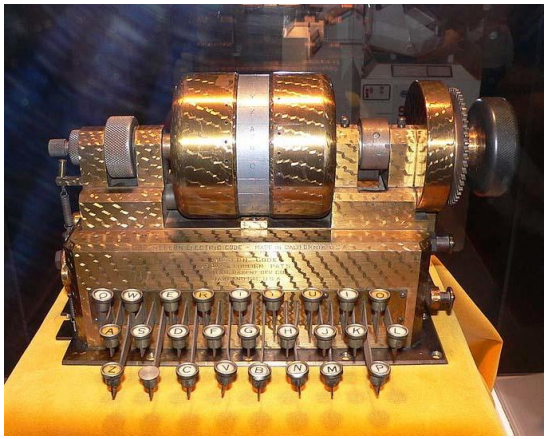
English text is highly redundant: very roughly 1 bit/letter entropy.

Monoalphabetic substitution ciphers allow simple ciphertext-only attacks based on digram or trigram statistics (for messages of at least few hundred characters).

Rotor machines

Early example: the Hebern machine (single rotor)

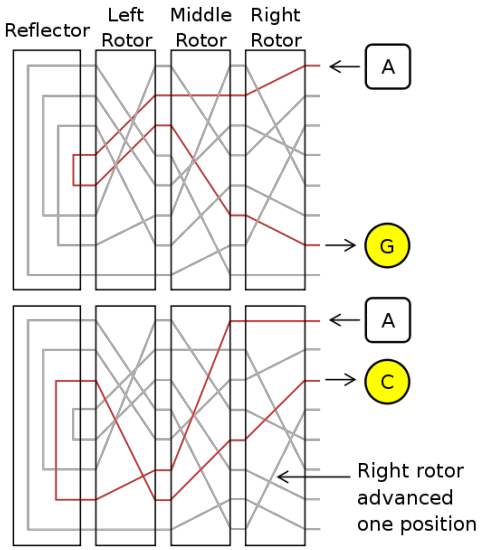




Dan Boneh

Rotor machines

Most famous: the Enigma (3-5 rotors)



keys = $26^4 = 2^{18}$ (actually 2^{36} due to plugboard)

Bletchley Park

- Britain's codebreaking centre during WW II
- Alan Turing's and other mathematicians break first versions of Enigma
- 'Bomb': computer specialised in breaking such encryptions
- 'Colossus': first semi-programmable electronic computer
- Great help against Germany and Japan - very important factor in determining the end of the war

https://www.bletchleypark.org.uk/

MODERN CRYPTOGRAPHY

What is a secure cipher ?

Possible security requirements:

attempt #1: **attacker cannot recover secret key**

What is a secure cipher ?

Possible security requirements:

attempt #1: **attacker cannot recover secret key**

$E(k, m) = m$ would be secure

What is a secure cipher ?

Possible security requirements:

attempt #1: **attacker cannot recover secret key**

$E(k, m) = m$ would be secure

attempt #2: **attacker cannot recover all of plaintext**

What is a secure cipher ?

Possible security requirements:

attempt #1: **attacker cannot recover secret key**

$E(k, m) = m$ would be secure

attempt #2: **attacker cannot recover all of plaintext**

$E(k, m_0 \| m_1) = m_0 \| k \oplus m_1$ would be secure

What is a secure cipher ?

Possible security requirements:

attempt #1: **attacker cannot recover secret key**

$E(k, m) = m$ would be secure

attempt #2: **attacker cannot recover all of plaintext**

$E(k, m_0 \| m_1) = m_0 \| k \oplus m_1$ would be secure

Shannon's idea:

CT should reveal no "info" about PT

Approach to modern cryptography

- Precisely specify threat model
- Propose a construction
- Prove that breaking construction under threat mode will solve an underlying hard problem

Symmetric vs asymmetric cryptography

- Symmetric cryptography
  - All parties have the same private key
  - E.g. stream ciphers (Salsa20), block ciphers (AES)
  - Used for bulk encryption
- Asymmetric cryptography
  - Uses private AND public keys
  - Users can publish their public keys without affecting security of private keys
  - Used for key exchange and authentication

STREAM CIPHERS

One time pad (Vernam, 1917)

$C := E(K, m) = K \oplus m$

$D(K, C) = K \oplus C$

msg: 0 1 1 0 1 1 1

key: 1 0 1 1 0 1 0 $\oplus$

CT:

Lemma: OTP has perfect secrecy.

The bad news …

Thm: perfect secrecy $\Rightarrow$ $|\mathcal{K}| \geq |\mathcal{M}|$

i.e. perfect secrecy $\Rightarrow$ key-len $\geq$ msg-len

$\Longrightarrow$ hard to use in practice !!

Stream ciphers: making OTP practical

idea: replace "random" key by "pseudorandom" key

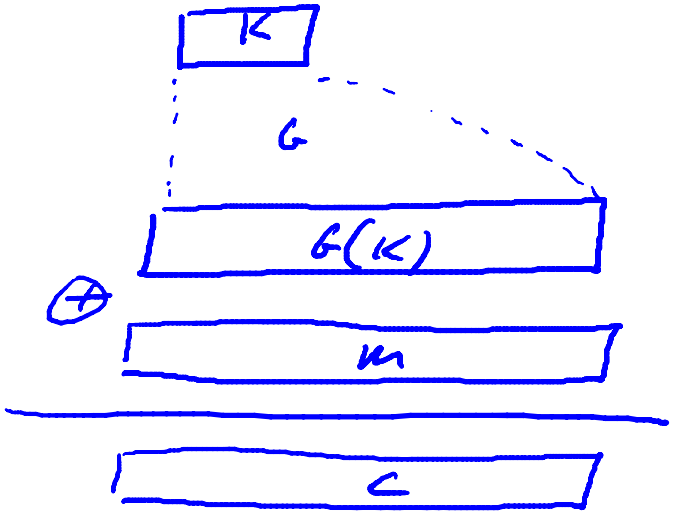
PRG is a function $G: \underbrace{\{0,1\}^s}_{\text{seed space}} \rightarrow \{0,1\}^n$ $\quad n \gg s$

(eff. computable by a deterministic algorithm)

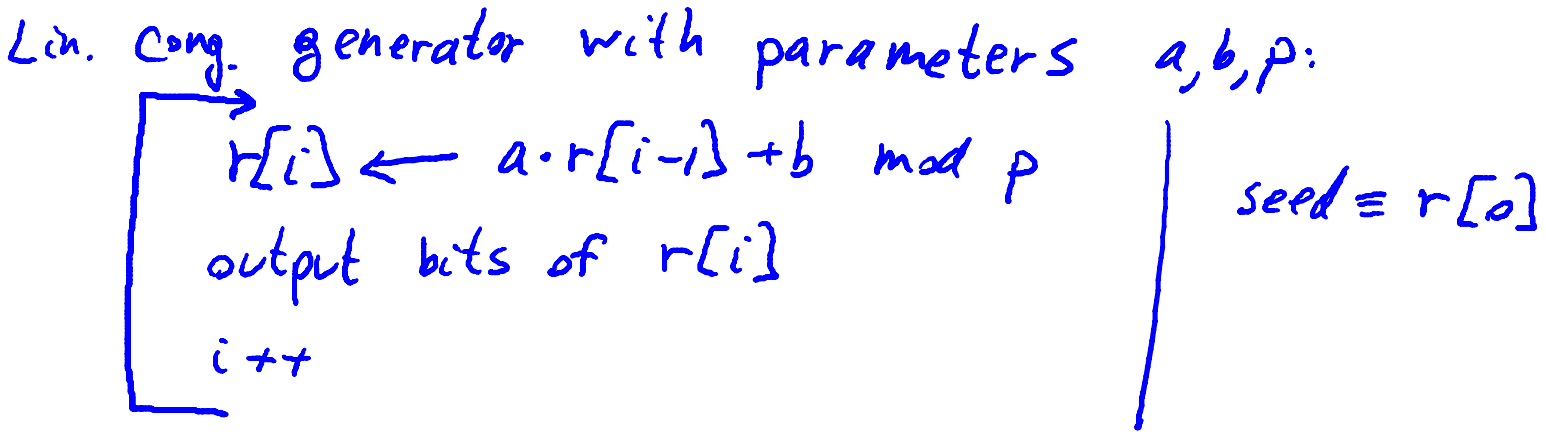
Stream ciphers: making OTP practical

$C := E(k,m) = m \oplus G(k)$

$D(k,c) = c \oplus G(k)$



Weak PRGs (do not use for crypto)



glibc random():

$r[i] \leftarrow (r[i-3] + r[i-31]) \ \% \ 2^{32}$

output $r[i] >> 1$

never use random() for crypto !!

(e.g. Kerberos V4)

Never use one-time pad (or generated stream) more than once

$C_1 \leftarrow m_1 \oplus PRG(k)$

$C_2 \leftarrow m_2 \oplus PRG(k)$

Eavesdropper does:

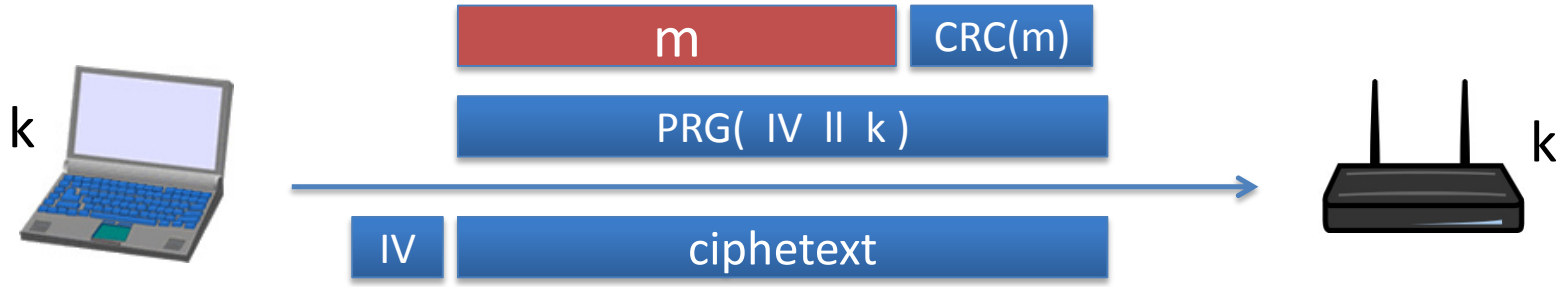
$C_1 \oplus C_2 \rightarrow$ 

Enough redundancy in English and ASCII encoding that:

$m_1 \oplus m_2 \rightarrow m_1, m_2$

Real world examples

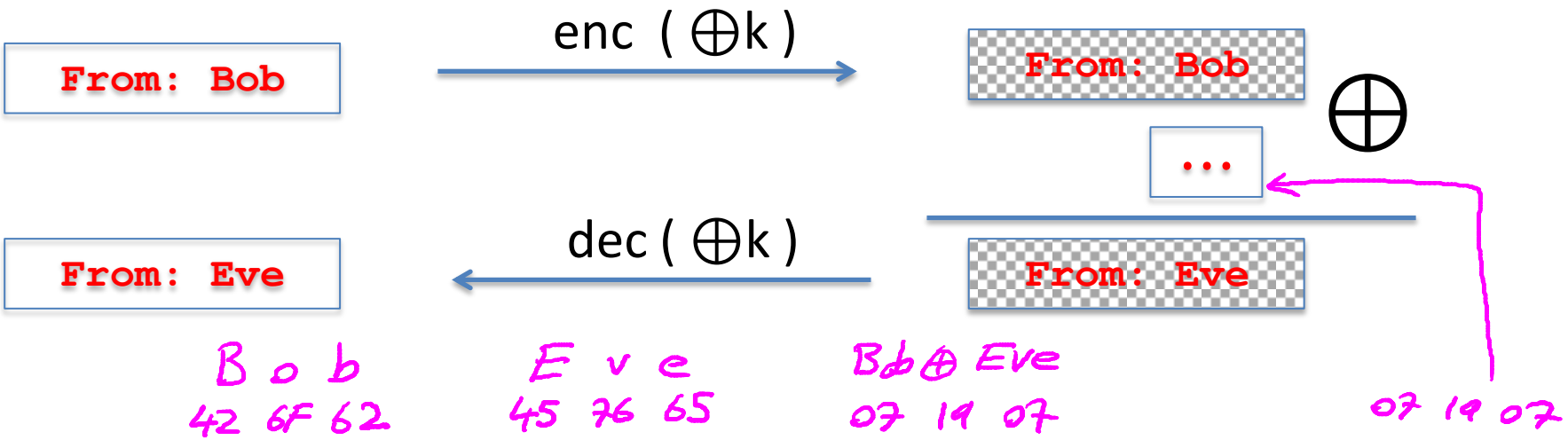
802.11b WEP:



Length of IV: 24 bits

- Repeated IV after $2^{24} \approx$ 16M frames
- On some 802.11 cards: IV resets to 0 after power cycle

Problem 2: OTP is malleable (no integrity)



Modifications to ciphertext are undetected and have predictable impact on plaintext

Modern stream ciphers: eStream

PRG: $\{0,1\}^s \times R \longrightarrow \{0,1\}^n$

seed

nonce

Nonce: a non-repeating value for a given key.

$$E(k, m ; r) = m \oplus PRG(k ; r)$$

The pair (k,r) is never used more than once.

Performance:

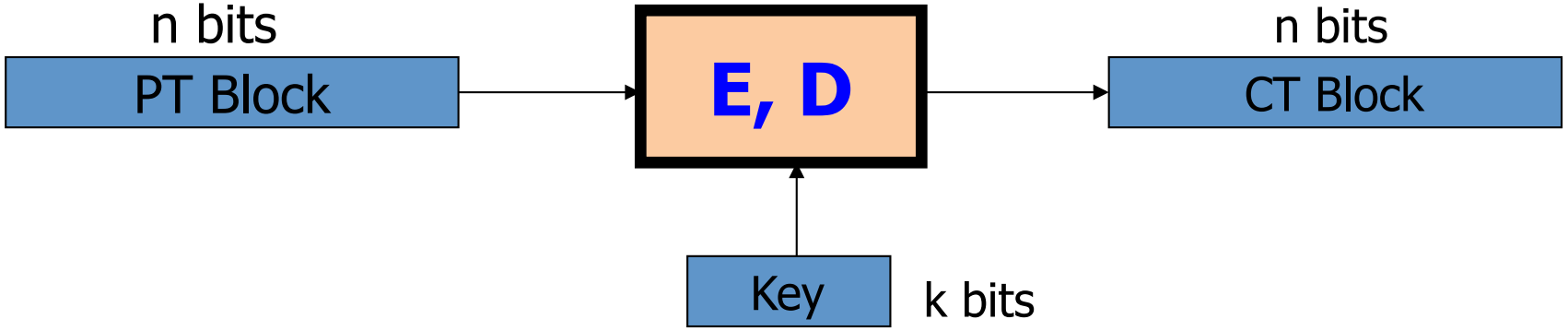
Crypto++ 5.6.0 [Wei Dai]

AMD Opteron, 2.2 GHz (Linux)

	PRG	Speed (MB/sec)
	RC4	126
eStream	Salsa20/12	643
eStream	Sosemanuk	727

BLOCK CIPHERS

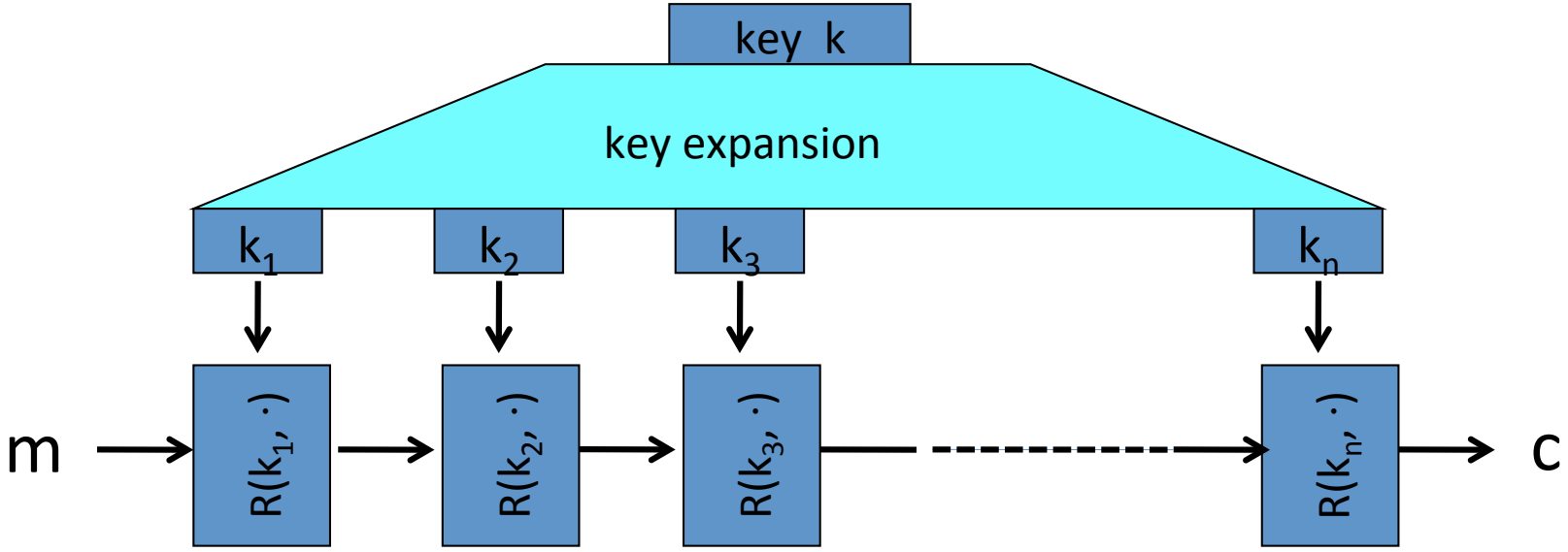
Block ciphers: crypto work horse



Canonical examples:

1. 3DES: n= 64 bits, k = 168 bits
2. AES: n=128 bits, k = 128, 192, 256 bits

Block Ciphers Built by Iteration



R(k,m) is called a round function

for 3DES (n=48), for AES-128 (n=10)

DES history

- Early 1970s: Horst Feistel designs Lucifer at IBM

  key-len = 128 bits ; block-len = 128 bits

- 1973: NBS asks for block cipher proposals.
  IBM submits variant of Lucifer.
- 1976: NBS adopts DES as a federal standard

  key-len = 56 bits ; block-len = 64 bits

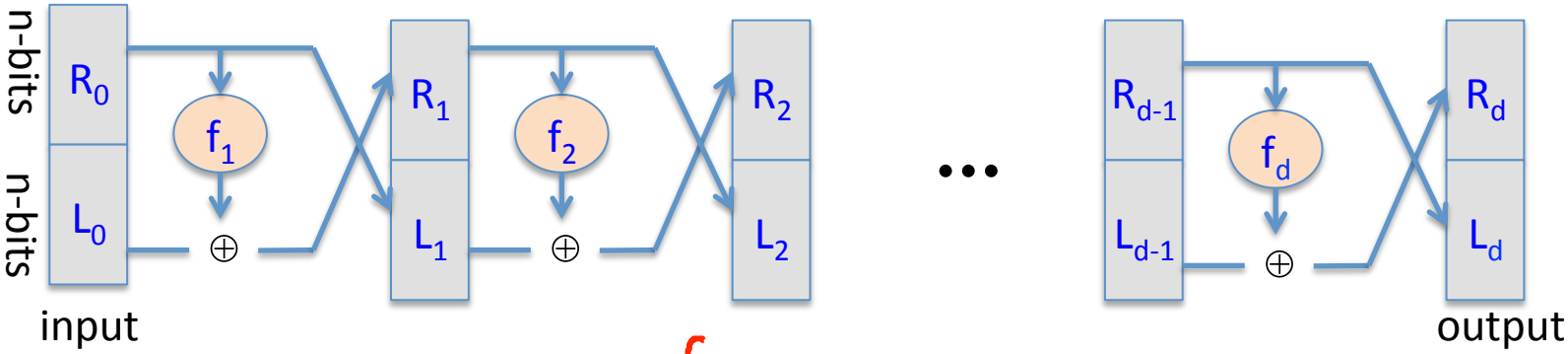
- 1997: DES broken by exhaustive search
- 2000: NIST adopts Rijndael as AES to replace DES

Widely deployed in banking (ACH) and commerce

DES: core idea – Feistel Network

Given functions $f_1, \ldots, f_d$: $\{0,1\}^n \longrightarrow \{0,1\}^n$

Goal: build invertible function $F: \{0,1\}^{2n} \longrightarrow \{0,1\}^{2n}$



In symbols:

$$\begin{cases} R_i = f_i(R_{i-1}) \oplus L_{i-1} \\ L_i = R_{i-1} \end{cases}$$

The S-boxes

$S_i: \{0,1\}^6 \longrightarrow \{0,1\}^4$

S_5		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

DES challenge

msg = "The unknown messages is: XXXX … "

CT = c_1 c_2 c_3 c_4

Goal: find $k \in \{0,1\}^{56}$ s.t. $DES(k, m_i) = c_i$ for i=1,2,3

1997: Internet search -- **3 months**

1998: EFF machine (deep crack) -- **3 days** (250K $)

1999: combined search -- **22 hours**

2006: COPACOBANA (120 FPGAs) **-- 7 days** (10K $)

⇒ 56-bit ciphers should not be used !! (128-bit key ⇒ 2^{72} days)

Strengthening DES against ex. search

Method 1: **Triple-DES**

- Let $E : K \times M \longrightarrow M$ be a block cipher
- Define **3E**: $K^3 \times M \longrightarrow M$ as

$$\mathbf{3E}\big((k_1,k_2,k_3), m\big) = E(k_1, D(k_2, E(k_3, m)))$$

$k_1 = k_2 = k_3 \Rightarrow$ single DES

For 3DES: key-size = 3×56 = 168 bits. 3×slower than DES.

(simple attack in time $\approx 2^{118}$)

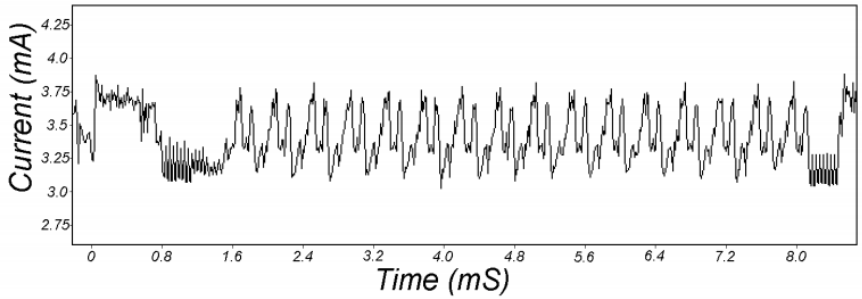
Attacks on the implementation

1. Side channel attacks:

We'll talk in detail in ~2 lectures

   – Measure **time** to do enc/dec, measure **power** for enc/dec

smartcard



[Kocher, Jaffe, Jun, 1998]

2. Fault attacks:

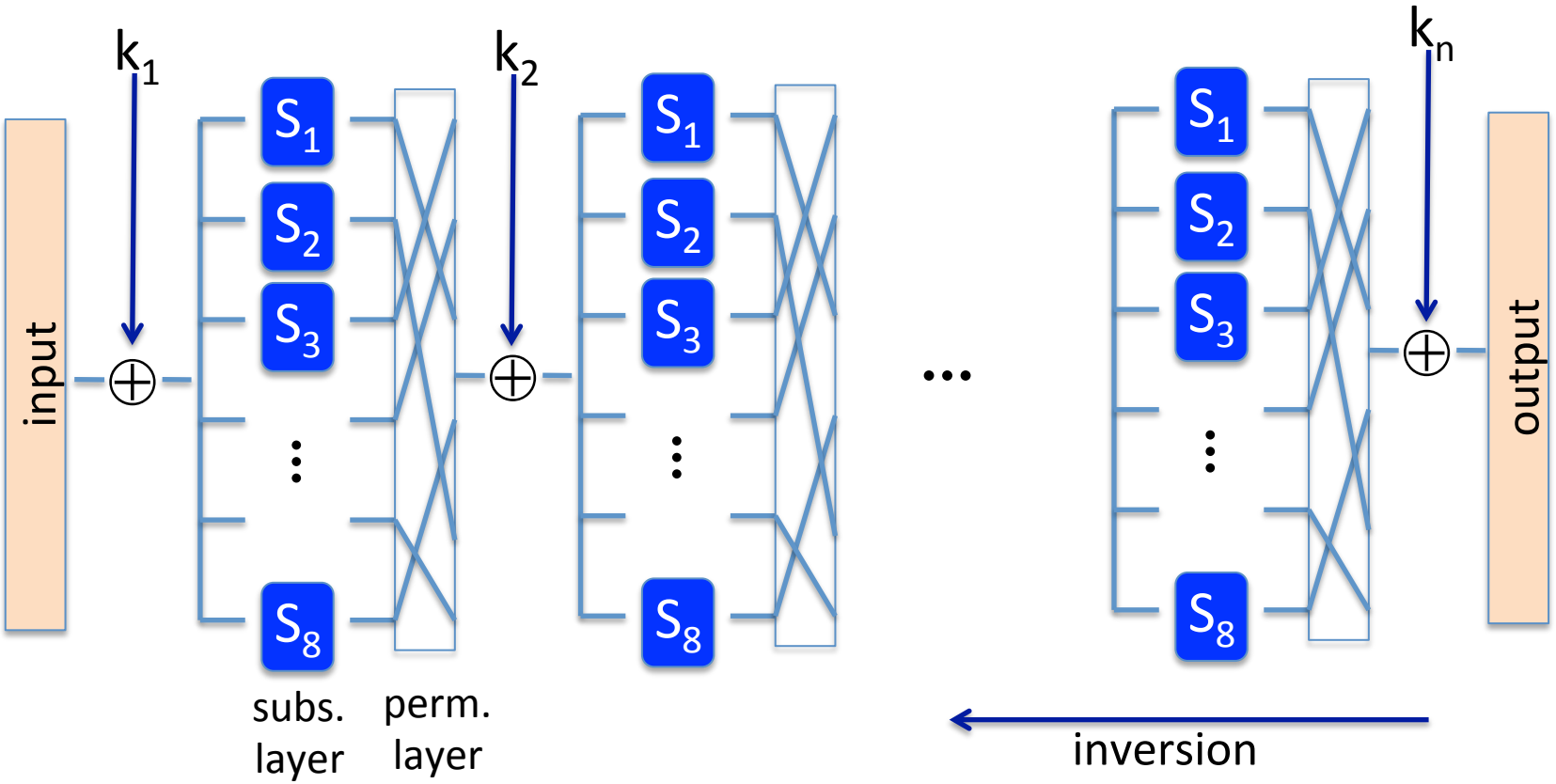
   – Computing errors in the last round expose the secret key k

AES history

- 1997: NIST publishes request for proposal
- 1998: 15 submissions. Five claimed attacks.
- 1999: NIST chooses 5 finalists (Rijndael, Serpent, Twofish, RC6, MARS)
- 2000: NIST chooses Rijndael as AES (designed in Belgium)

Key sizes: 128, 192, 256 bits. Block size: 128 bits

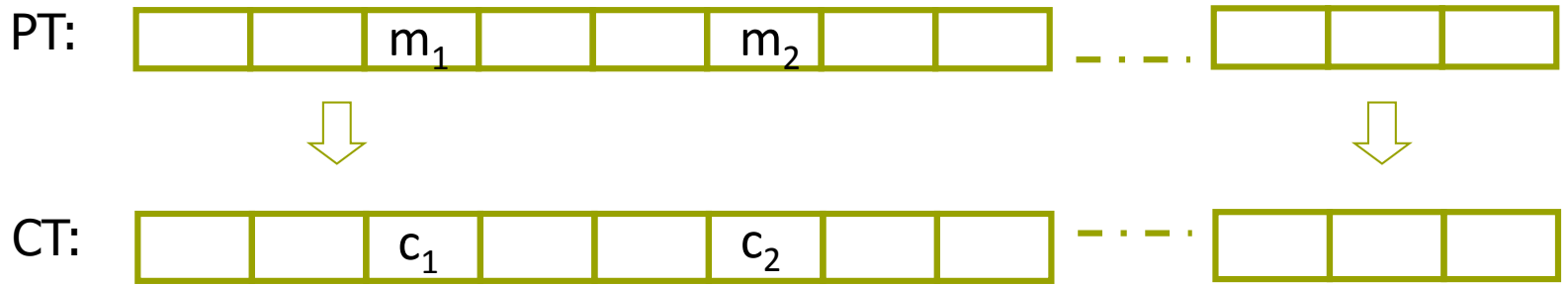
AES is a Subs-Perm network (not Feistel)



MODES OF OPERATION

ECB: do NOT use

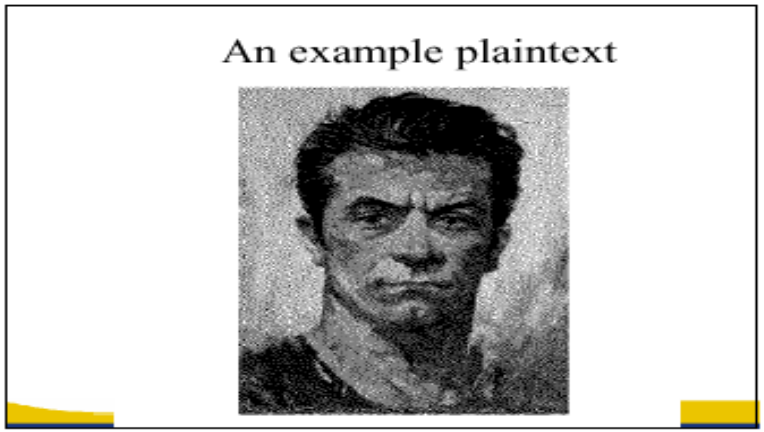
Electronic Code Book (ECB):

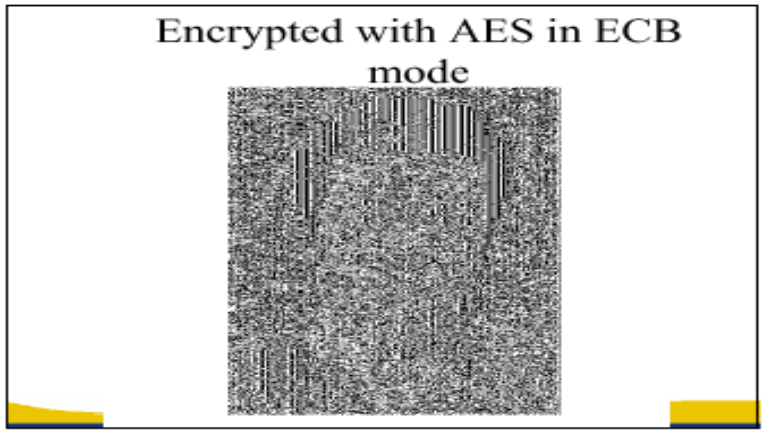


<u>Problem</u>:

– if $m_1=m_2$ then $c_1=c_2$

In pictures



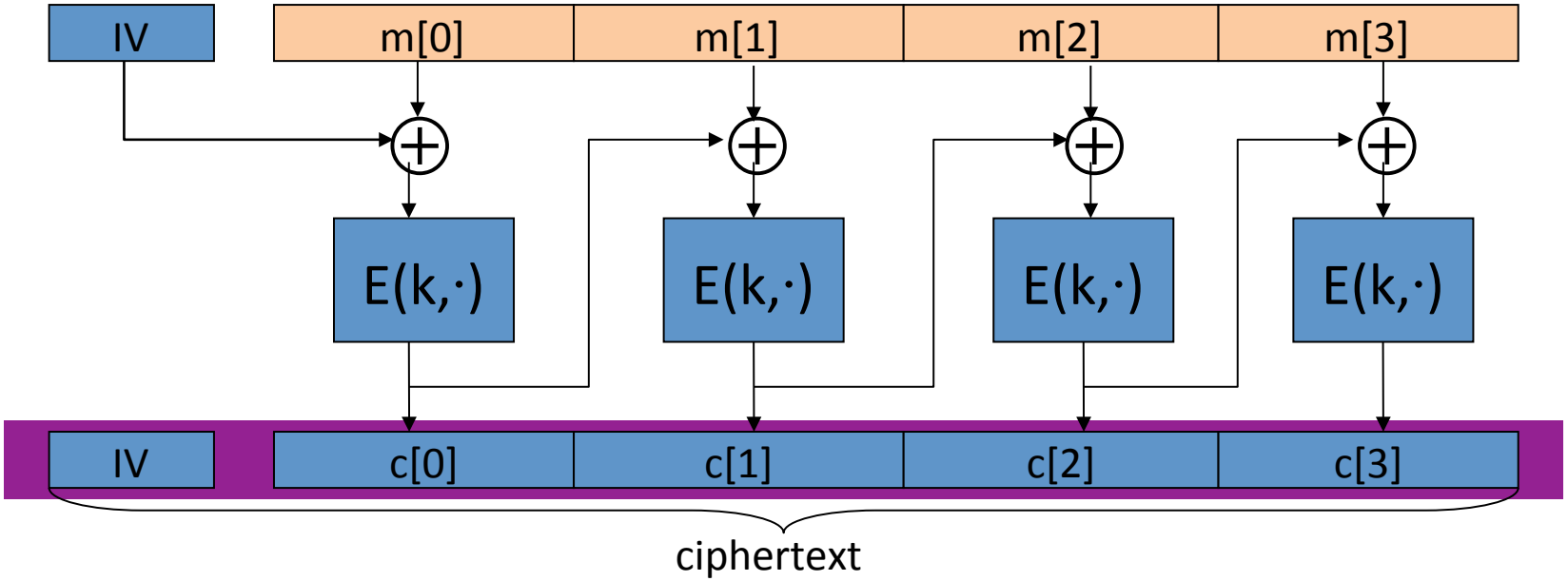


(courtesy B. Preneel)

Good construction 1: CBC

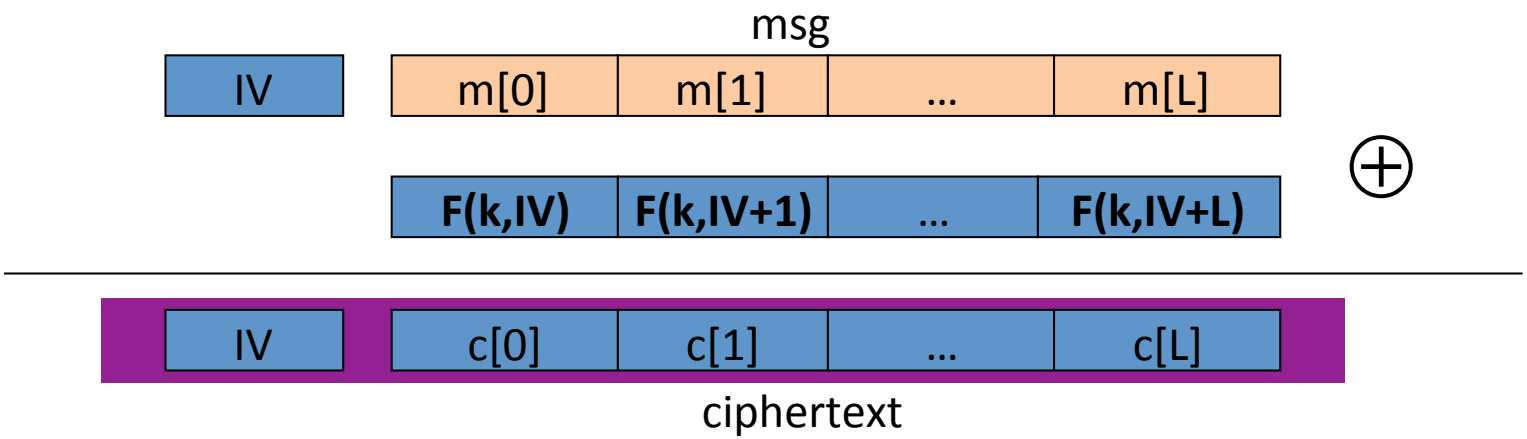
$E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$

$IV \in \{0,1\}^n$



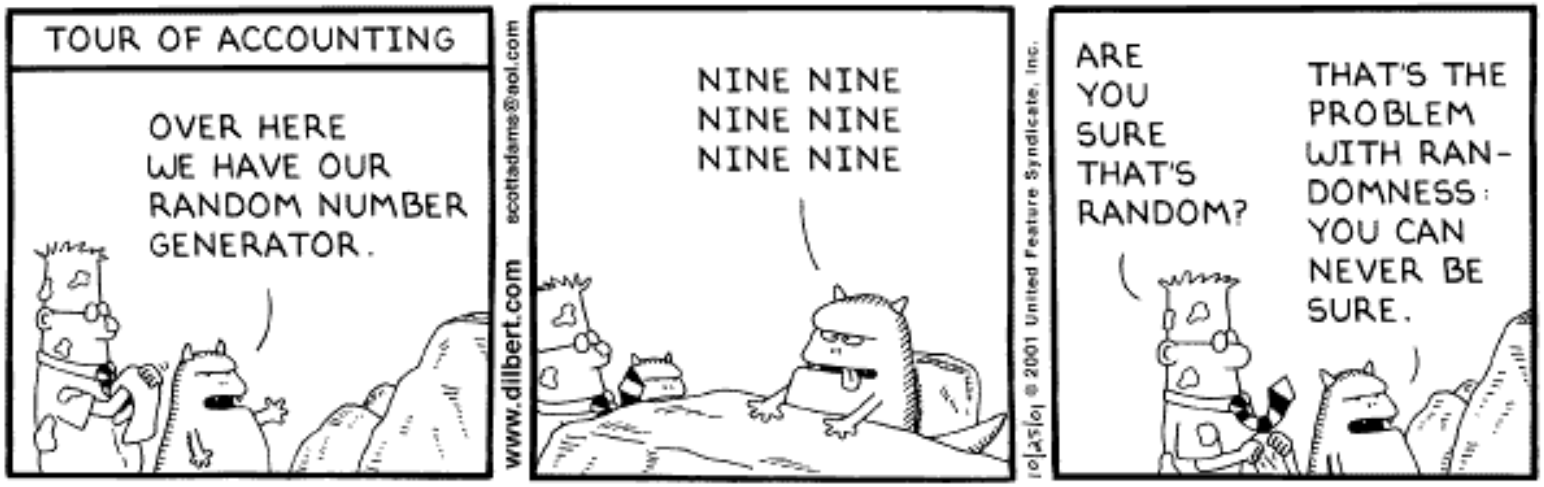
Good construction 2: CTR

E(k,m): choose a random $IV \in \{0,1\}^n$ and do:

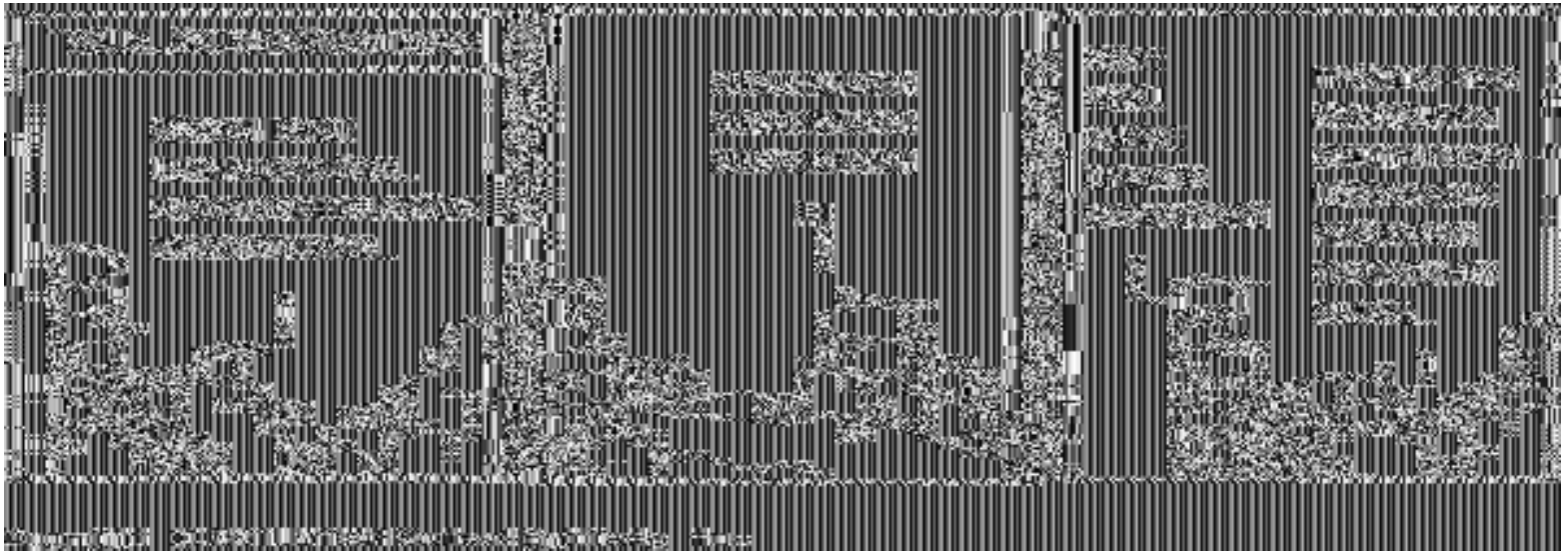


note: parallelizable (unlike CBC)

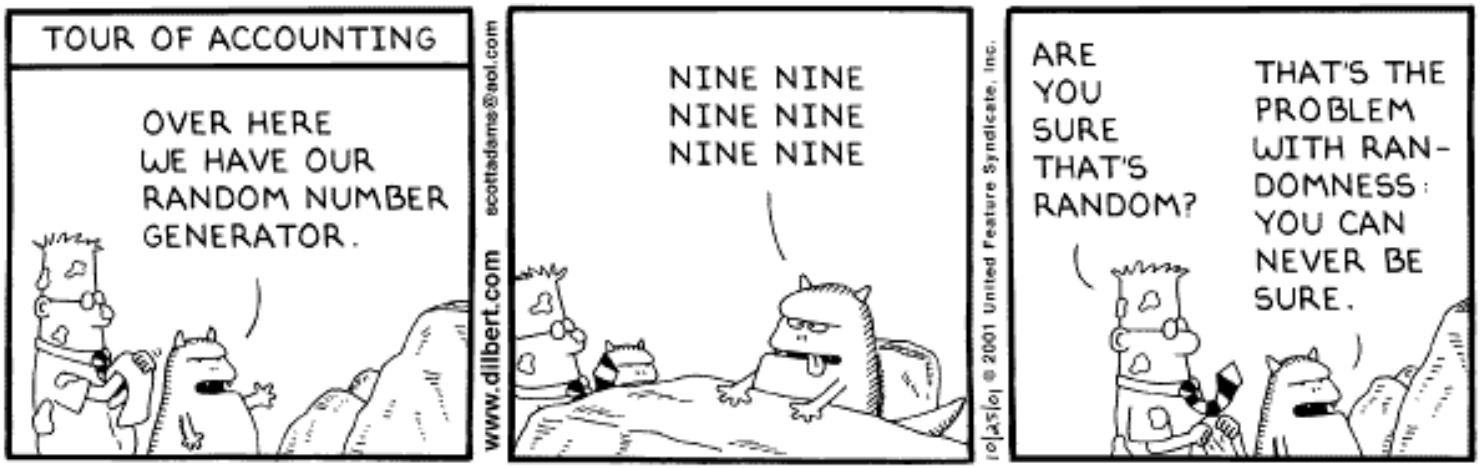
Plain-text bitmap:



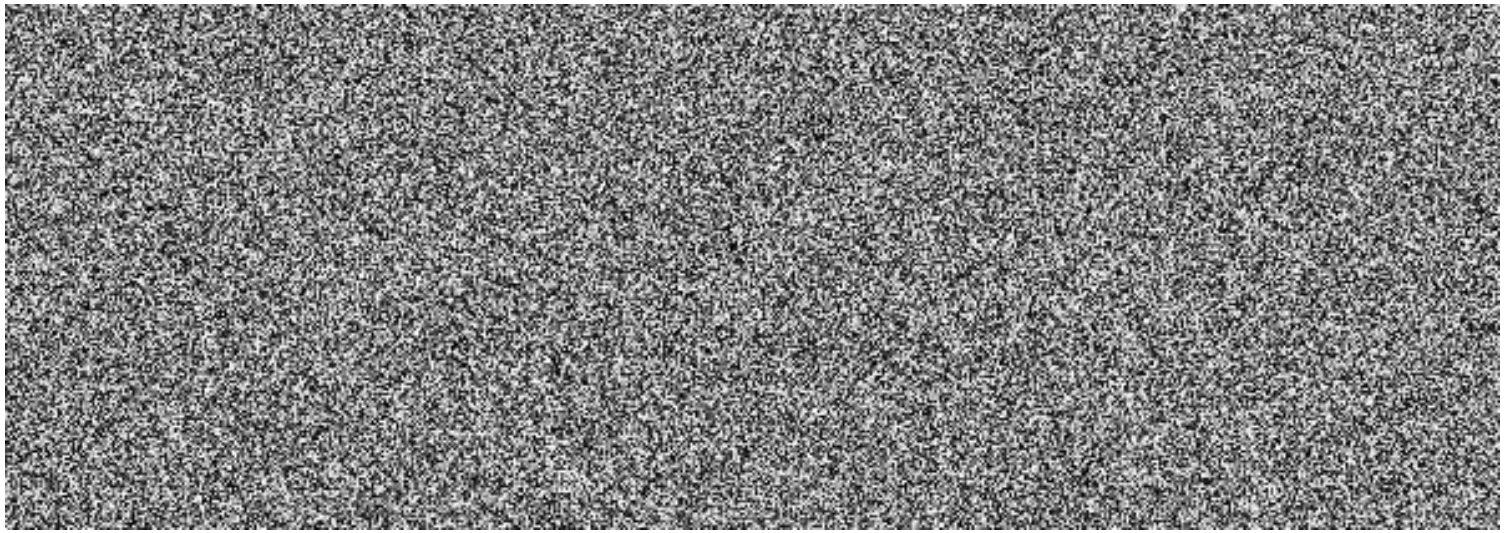
DES-ECB encrypted:



Plain-text bitmap:



DES-CBC encrypted:



Resurse utile

- Cursul de la Stanford (Dan Boneh): https://www.coursera.org/learn/crypto/
- Jonathan Katz, Yehuda Lindell: Introduction to Modern Cryptography CRC Press — 2008 (1st Ed.), 2015 (2nd Ed.)

Resurse utile

- Conferinte
  - IEEE Security & Privacy
  - Usenix Security
  - Crypto, Eurocrypt, Asiacrypt
  - ACM CCS, ESORICS
  - CHES, CARDIS
- https://eprint.iacr.org/
- https://scholar.google.co.uk/