Cryptographic algorithms

Prof. Bart Preneel

COSIC

Bart.Preneel(at)esatDOTkuleuven.be

http://homes.esat.kuleuven.be/~preneel

Outline

- 1. Cryptology: concepts and algorithms
  - symmetric algorithms for confidentiality
  - symmetric algorithms for data authentication
  - public-key cryptology
- 2. Cryptology: protocols
  - identification/entity authentication
  - key establishment
- 3. Public-Key Infrastructure principles

Outline (2)

- 4. Networking protocols
  - email, web, IPsec, SSL/TLS
- 5. New developments in cryptology
- 6. How to use cryptography well
- 7. Hash functions

Definitions

	data	**entities**
confidentiality	encryption	anonymity
authentication	data authentication	identification

Confidentiality
Integrity
Availability

Authorisation

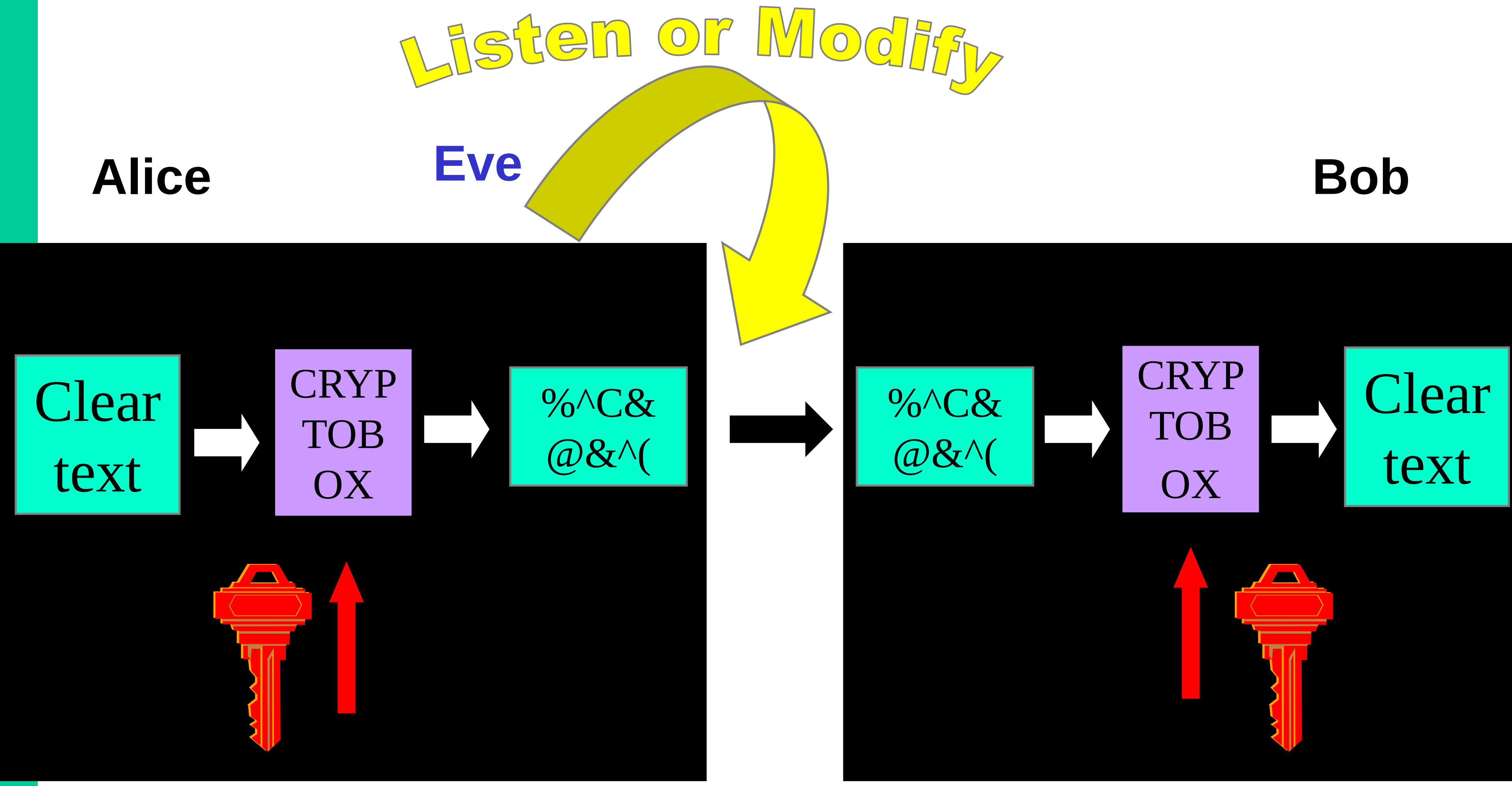
Non-repudiation of origin, receipt

Contract signing

Notarisation and Timestamping

Don’t use the word authentication without defining it

Cryptology: basic principles

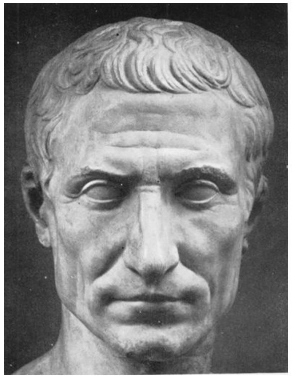


Symmetric cryptology: confidentiality

- old cipher systems:
  - transposition, substitution, rotor machines
- the opponent and her power
- the Vernam scheme
- DES and triple-DES
- AES
- RC4

Old cipher systems (pre 1900)

- Caesar cipher: shift letters over k positions in the alphabet (k is the secret key)



`THIS IS THE CAESAR CIPHER`

`WKLV LV WKH FDHVDU FLSKHU`

- Julius Caesar never changed his key (k=3).

Cryptanalysis example:

TIPGK	RERCP	JZJZJ	WLE	GVCTX	EREPC	WMWMW	JYR
UJQHL	SFSDQ	KAKAK	XMF	HWDUY	FSFQD	XNXNX	KZS
VKRIM	TGTER	LBLBL	YNG	IXEVZ	GTGRE	YOYOY	LAT
WLSJN	UHUFS	MCMCM	ZOH	JYFWA	HUHSF	ZPZPZ	MBU
XDTKO	VOVGT	NDNDN	API	KZGXB	IVITG	AQAQA	NCV
YNULP	WKWHU	OEOEO	BQJ	LAHYC	JWJUH	BRBRB	ODW
ZOVMQ	XKXIV	PFPFP	CRK	MBIZD	KXKVI	CSCSC	PEX
APWNR	YLYJW	QGQGQ	DSL	NCJAE	LYLWJ	DTDTD	QFY
BQXOS	ZMXKX	RHRHR	ETM	ODKBF	MZMXK	EUEUE	RGZ
CRYPT	ANALY	SISIS	FUN	PELCG	NANYL	FVFVF	SHA
DSZQU	BOBMZ	TJTJT	GVO	QFMDH	OBOZM	GWGWG	TIB
ETARV	CPCNA	UKUKU	HWP	RGNEI	PCPAN	HXHXH	UJC
FUBSW	DQDOB	VLVLV	IXQ	SHOFJ	QDQBO	IYIYI	VKD

Plaintext? k = 17

Old cipher systems (pre 1900) (2)

- Substitutions
  - **ABCDEFGHIJKLMNOPQRSTUVWXYZ**
  - **MZNJSOAXFQGYKHLUCTDVWBIPER**

! Easy to break using statistical techniques

- Transpositions

TRANS	**ORI S**
POSIT	**NOTIT**
IONS	**OSANP**

Security

- there are n! different substitutions on an alphabet with n letters
- there are n! different transpositions of n letters
- n=26: **n!=403291461126605635584000000 = 4 . 10^{26} keys**
- trying all possibilities at 1 nanosecond per key requires....

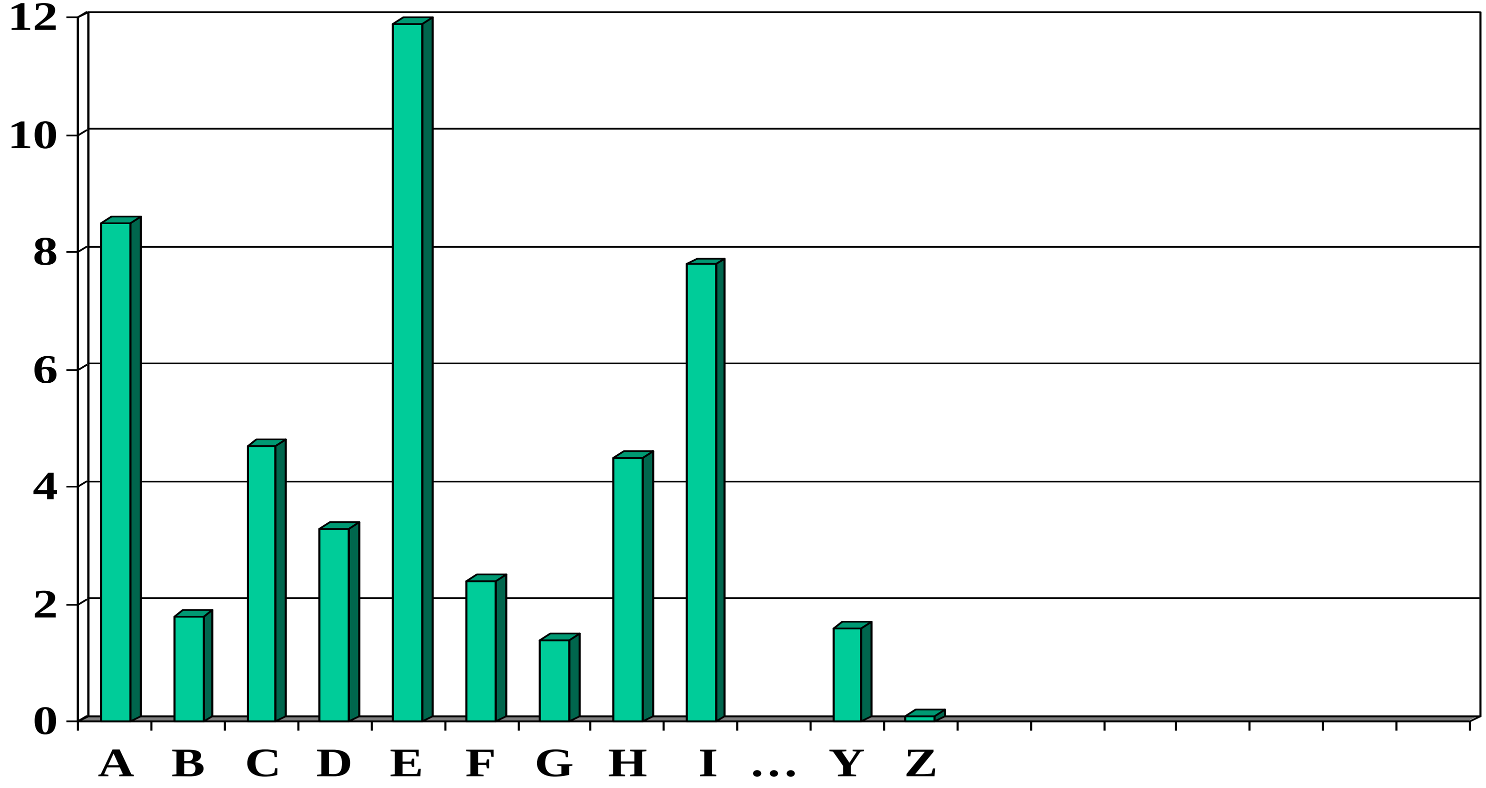
$$4.10^{26} \;/(10^{9} \,.\, 10^{5} \,.\, 4\ 10^{2}) = 10^{10} \text{ years}$$

keys per second

seconds per day

days per year

Letter distributions



Assumptions on Eve (the opponent)

- A scheme is broken if Eve can deduce the key or obtain additional plaintext
- Eve can always try all keys till "meaningful" plaintext appears: a brute force attack
  - solution: large key space
- Eve will try to find shortcut attacks (faster than brute force)
  - history shows that designers are too optimistic about the security of their cryptosystems

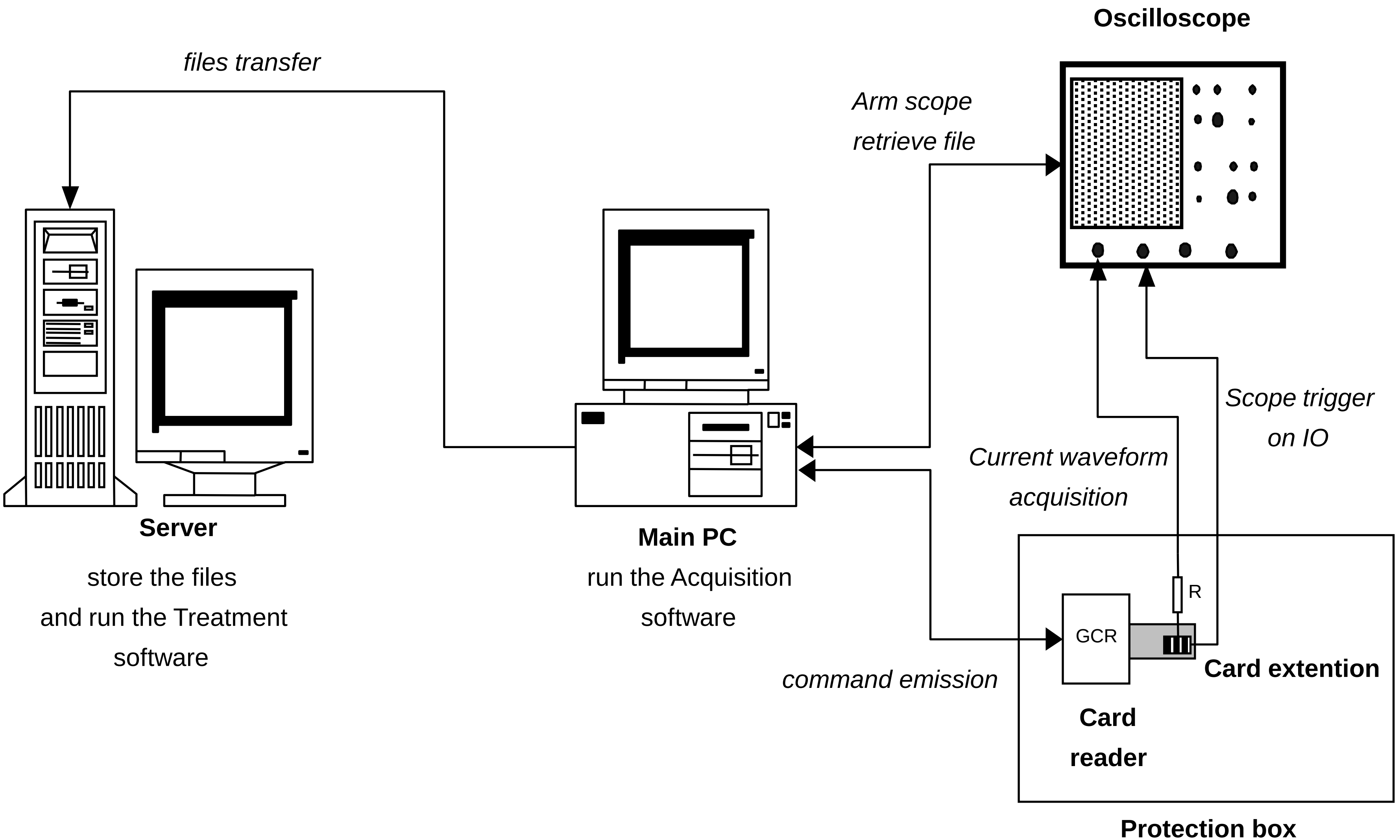
Assumptions on Eve (the opponent)

- Cryptology = cryptography + cryptanalysis
- Eve knows the algorithm, except for the key (Kerckhoffs's principle)
- increasing capability of Eve:
  - knows some information about the plaintext (e.g., in English)
  - knows part of the plaintext
  - can choose (part of) the plaintext and look at the ciphertext
  - can choose (part of) the ciphertext and look at the plaintext

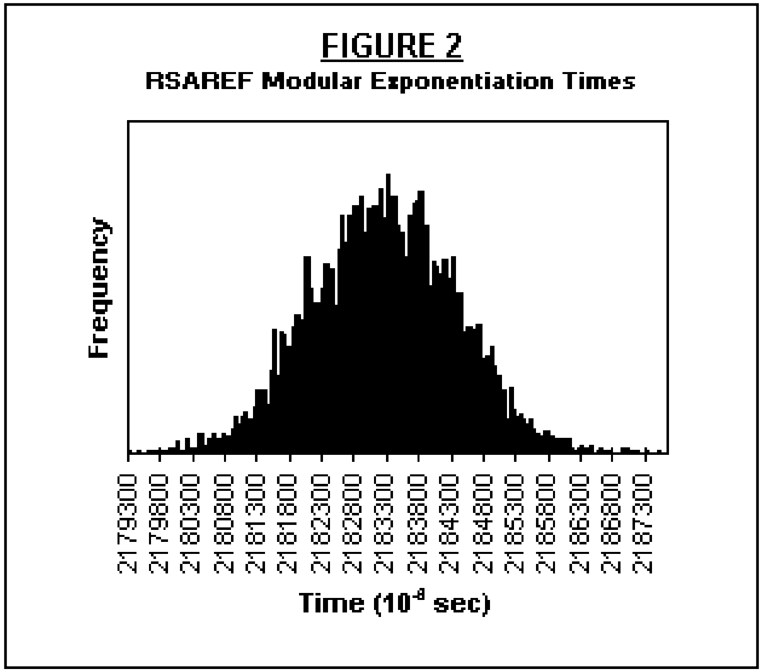
New assumptions on Eve

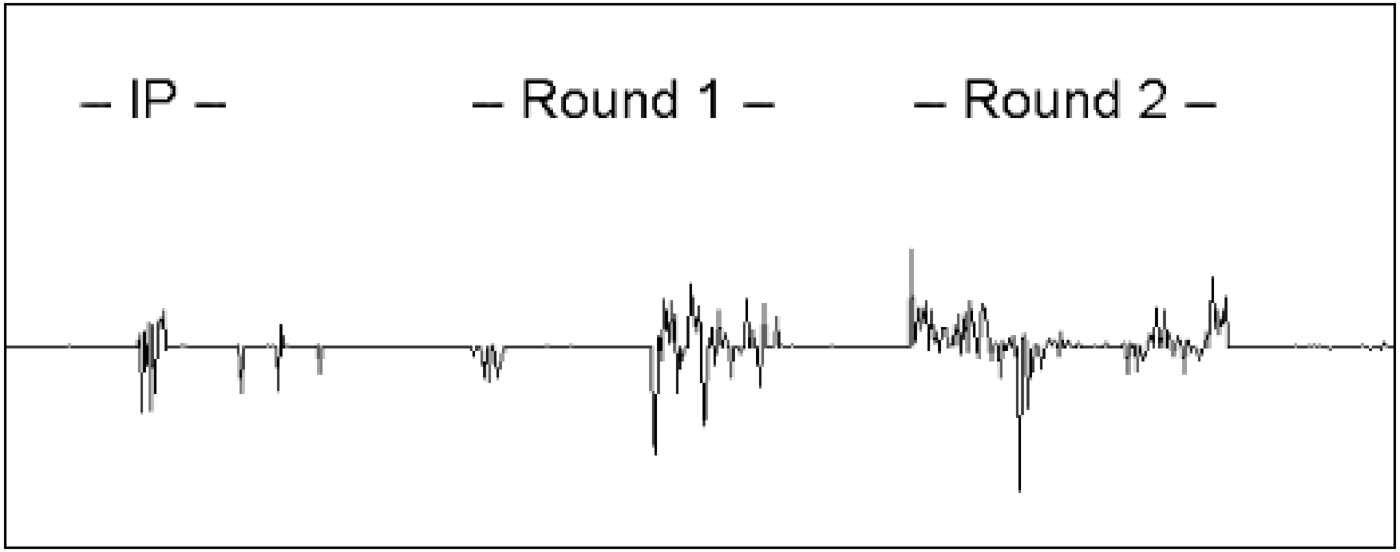
- Eve may have access to side channels
  - timing attacks
  - simple power analysis
  - differential power analysis
  - acoustic attacks
  - electromagnetic interference
- Eve may launch (semi-)invasive attacks
  - differential fault analysis
  - probing of memory or bus

Side channel analysis

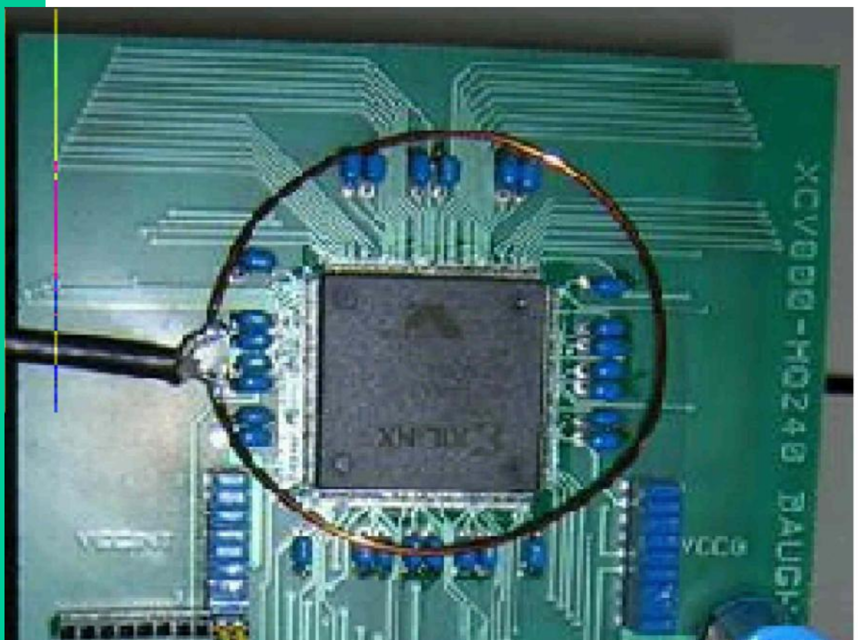


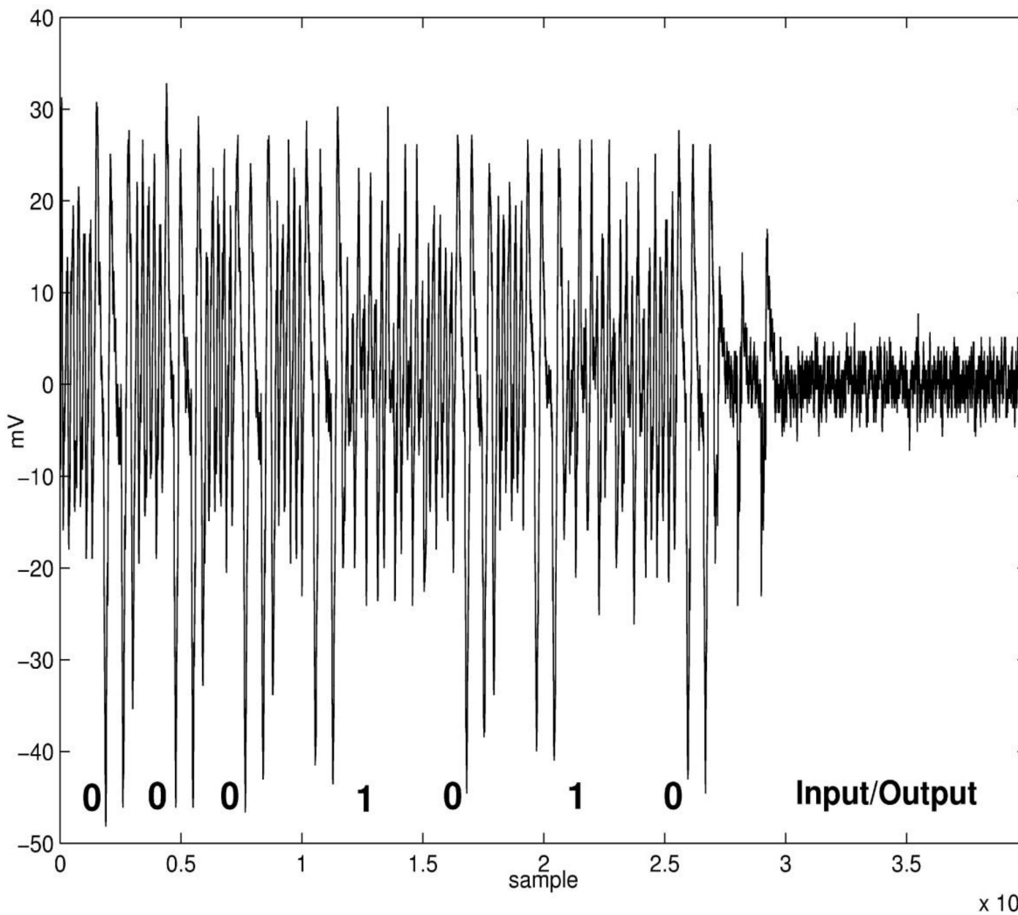
Timing attacks and power analysis



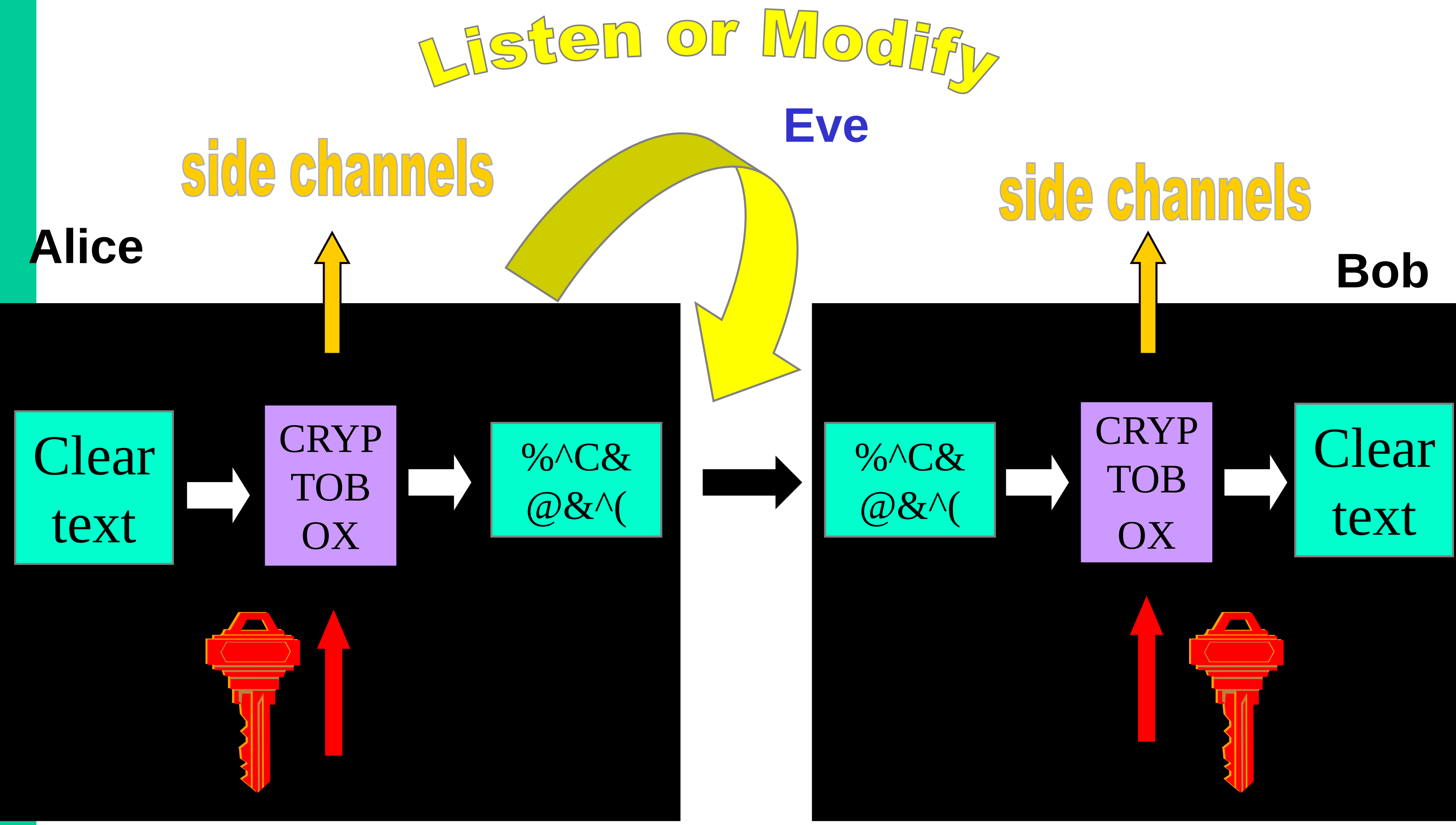


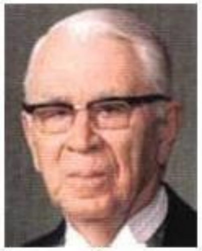
Side channel analysis: EMA





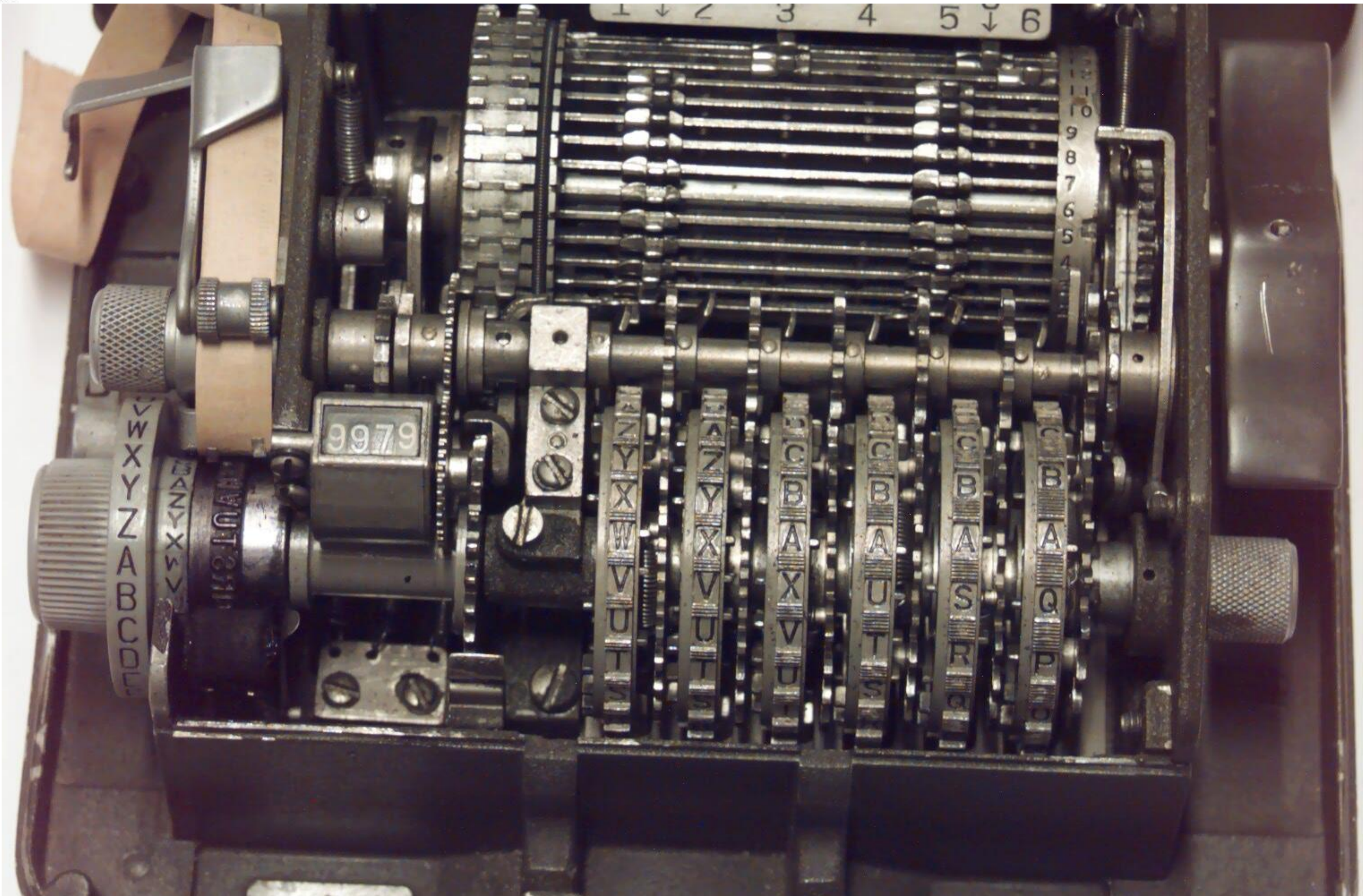
Cryptology + side channels





Boris Hagelin

Mechanical: Hagelin C38



Problem: what is this?

- Cryptogram [=14 January 1961 11.00 h]
- **<AHQNE XVAZW IQFFR JENFV OUXBD LQWDB BXFRZ NJVYB QVGOZ KFYQV GEDBE HGMPS GAZJK RDJQC VJTEB XNZZH MEVGS ANLLB DQCGF PWCVR UOMWW LOGSO ZWVVV LDQNI YTZAA OIJDR UEAAV RWYXH PAWSV CHTYN HSUIY PKFPZ OSEAW SUZMY QDYEL FUVOA WLSSD ZVKPU ZSHKK PALWB SHXRR MLQOK AHQNE 11205 141100>**

The answer

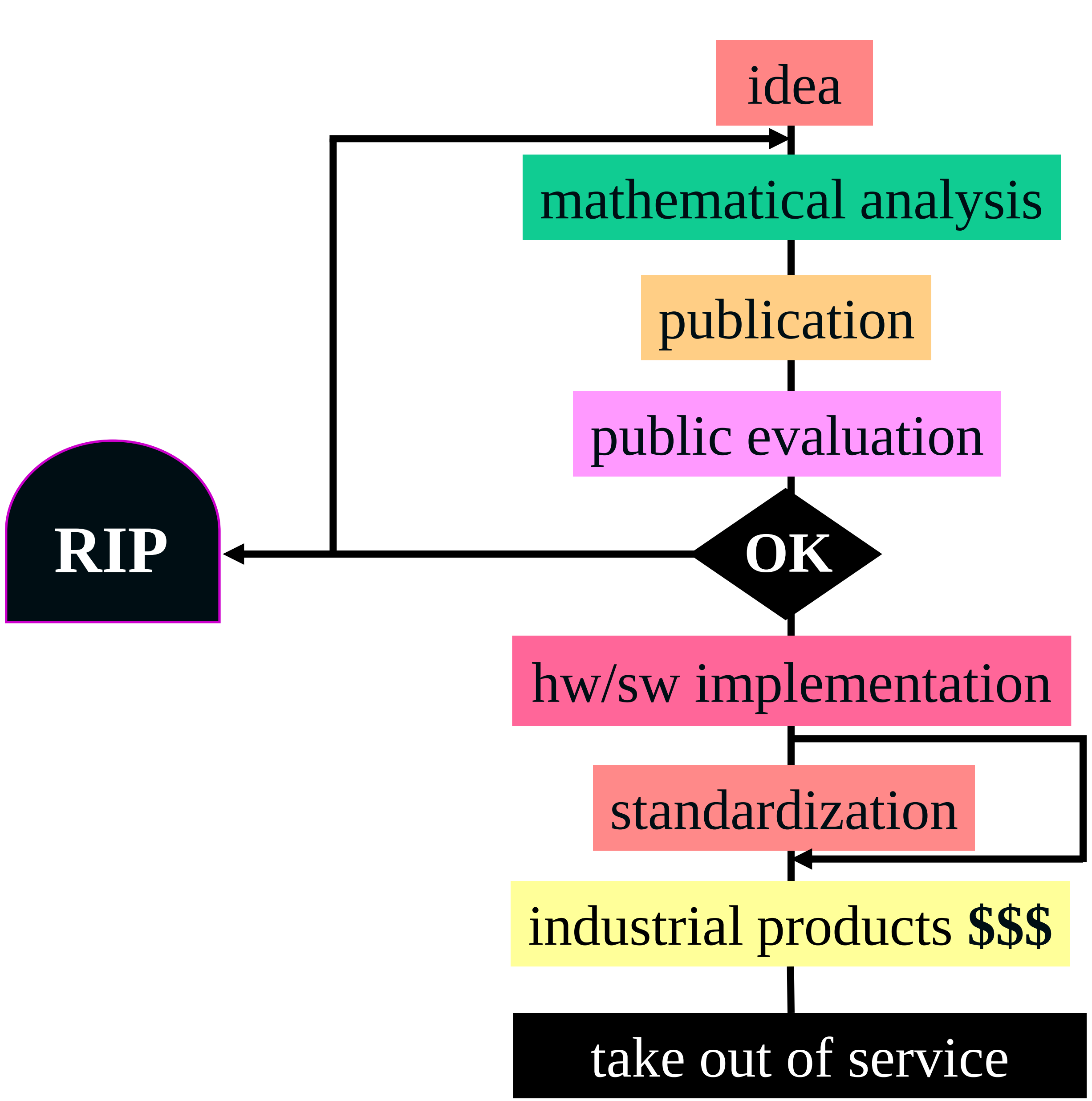
- Plaintext [=14 January 1961 11.00 h]
- **DOFGD** **VISWA WVISW JOSEP HWXXW TERTI OWMIS SIONW BOMBO KOWVO IRWTE LEXWC EWSUJ ETWAM BABEL GEWXX WJULE SWXXW BISEC TWTRE SECVX XWRWV WMWPR INTEX WXXWP RIMOW RIENW ENVOY EWRUS URWWX XWPOU VEZWR EGLER WXXWS ECUND OWREP RENDR EWDUR GENCE WPLAN WBRAZ ZAWWC**

The answer (in readable form)

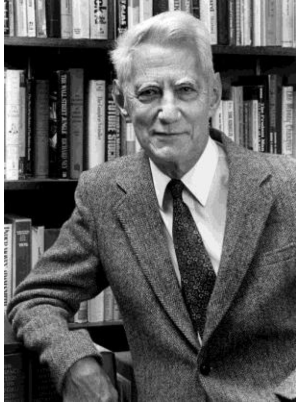
- Plaintext [=14 January 1961 11.00 h]
- **TRESECV. R V M PRINTEX. PRIMO RIEN ENVOYE RUSUR. POUVEZ REGLER. SECUNDO REPRENDRE DURGENCE PLAN BRAZZA VIS A VIS JOSEP H. TERTIO MISSION BOMBOKO VOIR TELEX CE SUJET AMBABELGE. JULES.**

The Rotor machines (WW II)

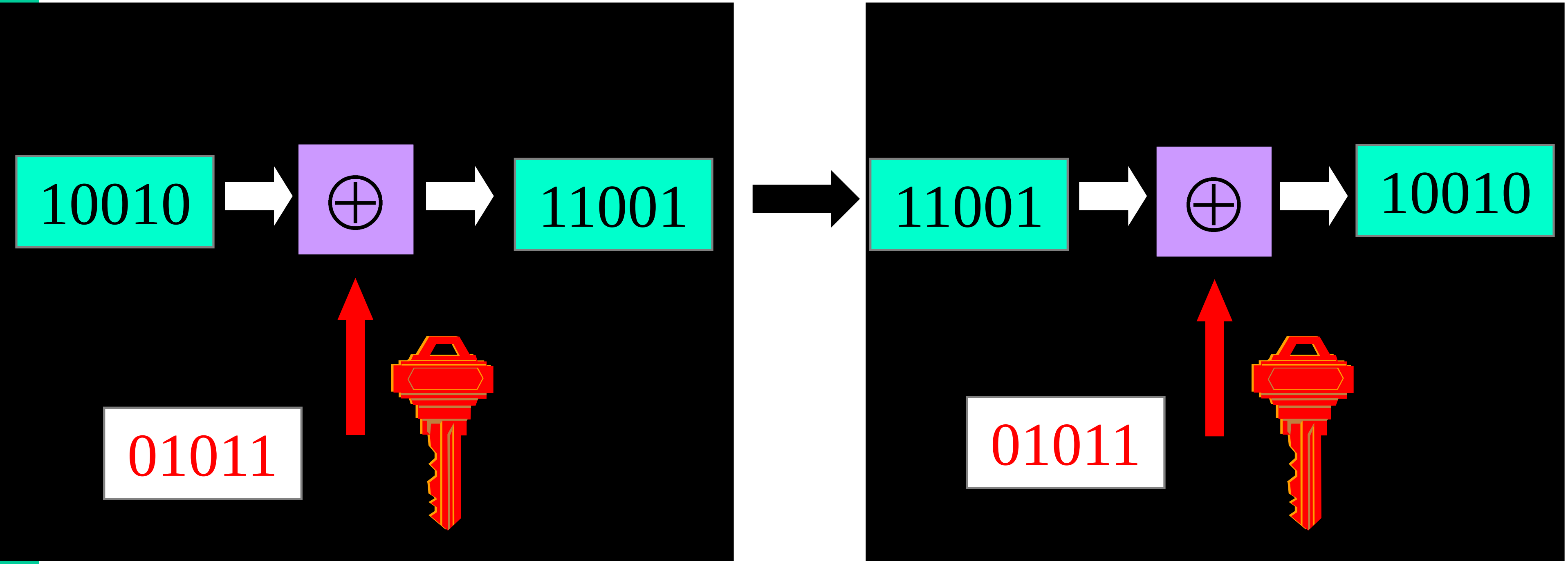
Life cycle of a cryptographic algorithm



Vernam scheme or one-time pad (1917)



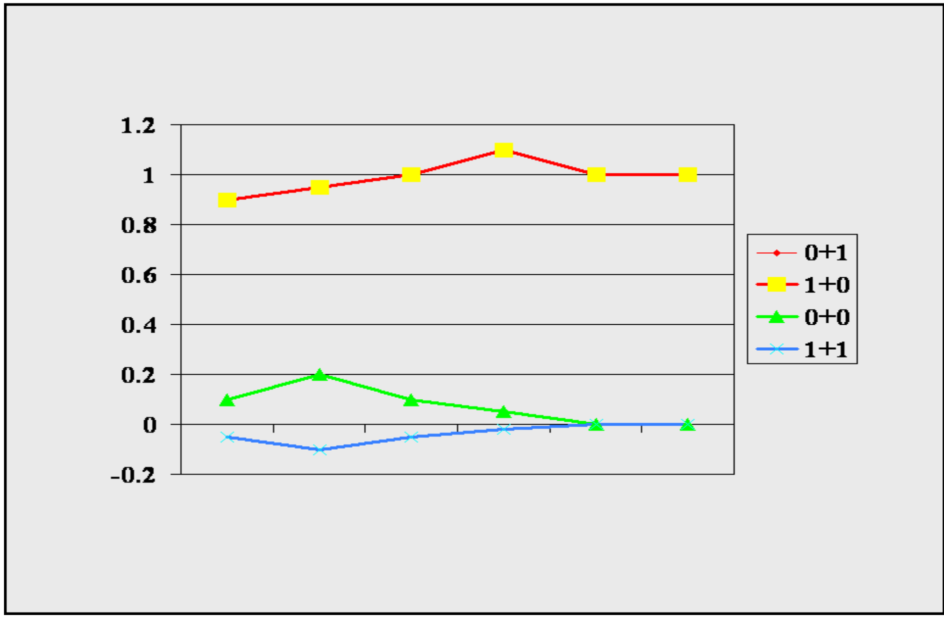
- key is random string, as long as the plaintext
- provably secure, but impractical



Vernam scheme

- 0 + 1 = 1
- 1 + 0 = 1
- 0 + 0 = 0
- 1 + 1 = 0

- identical mathematical symbols can result in different electrical signals

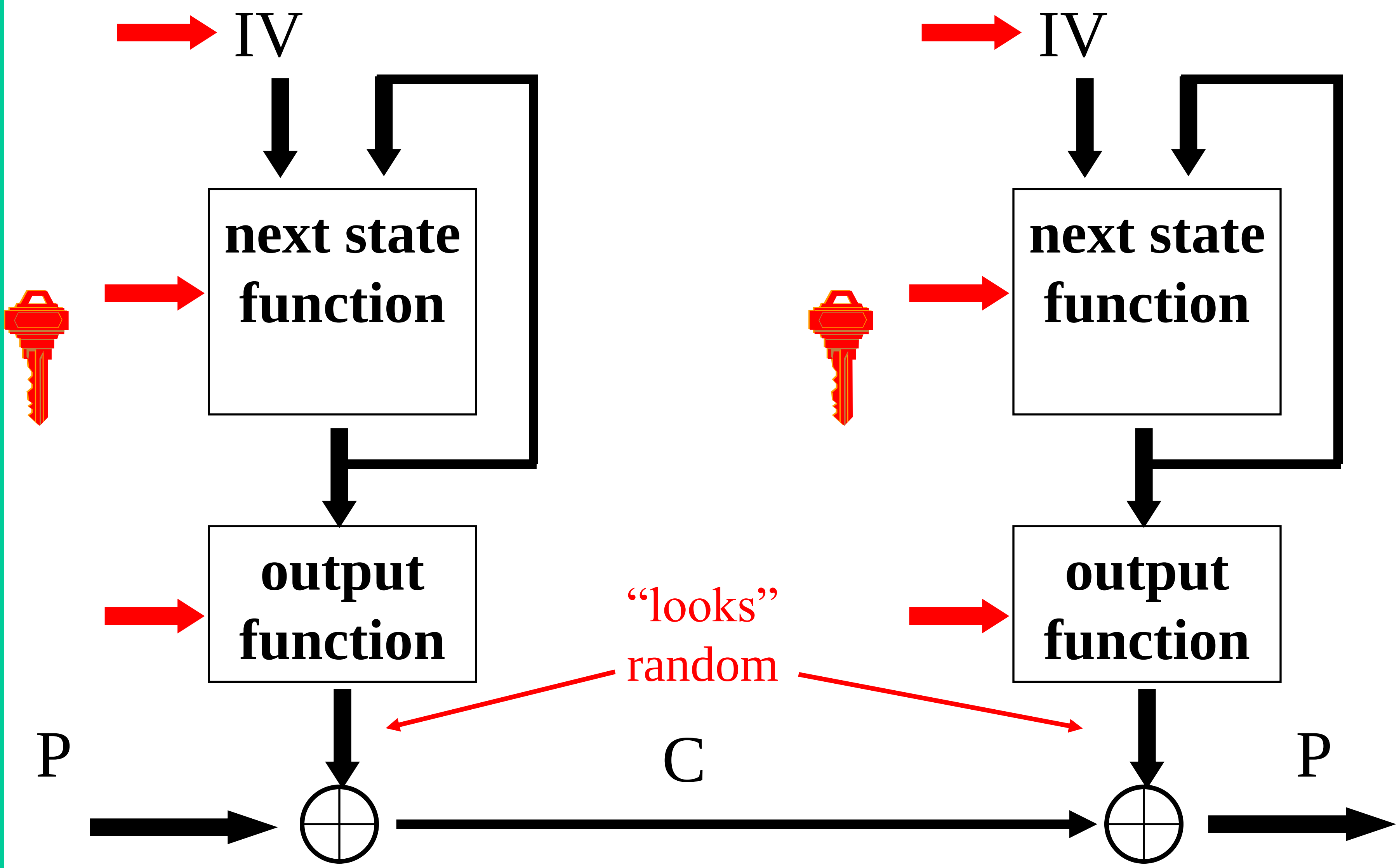


Three approaches in cryptography

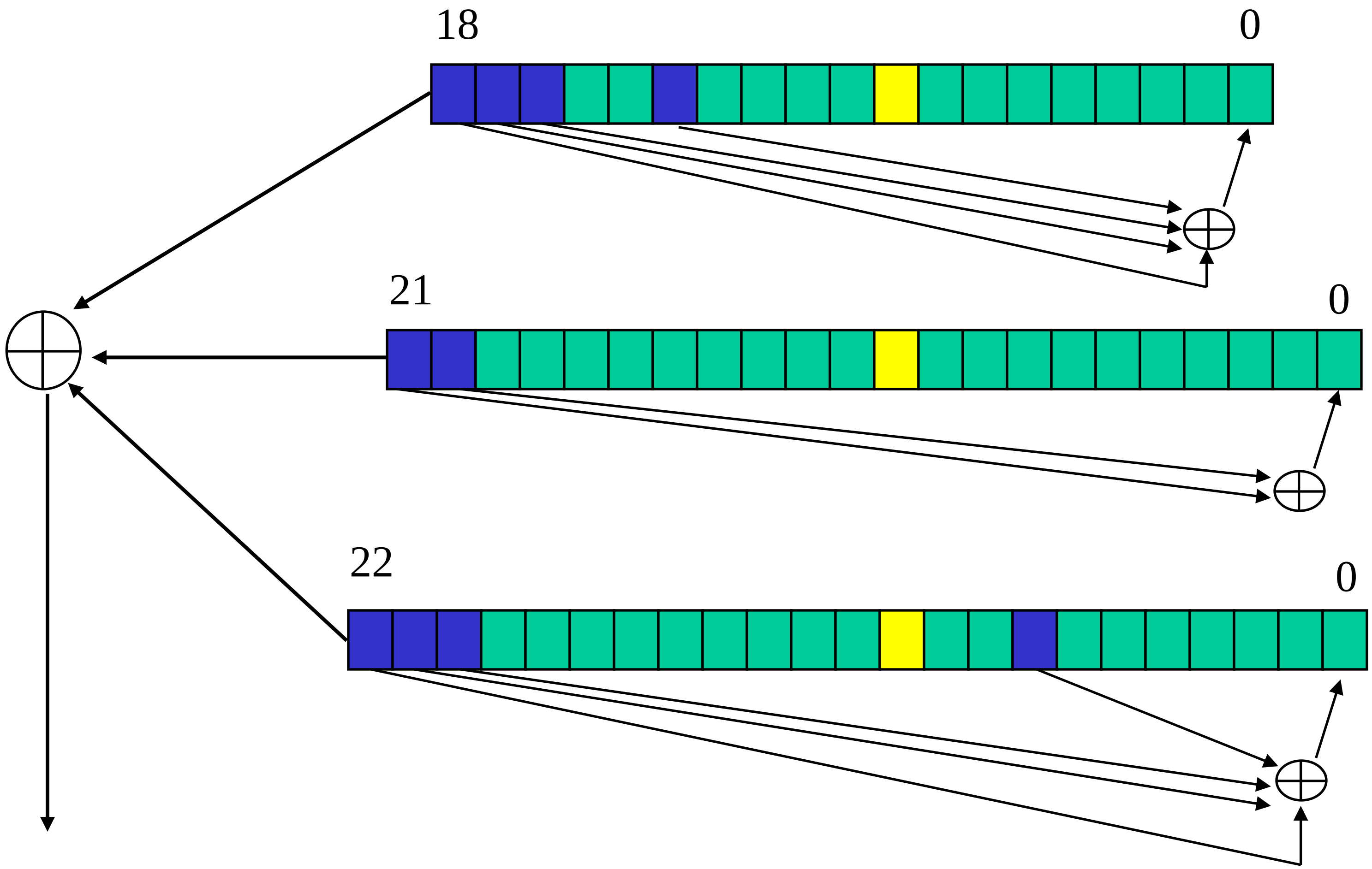
- **information theoretic** security
  - ciphertext only
  - part of ciphertext only
  - noisy version of ciphertext
- **system-based** or practical security
  - also known as “prayer theoretic” security
- **complexity theoretic** security:

  model of computation, definition, proof
  - variant: quantum cryptography

Model of a practical stream cipher



A5/1 stream cipher (GSM)



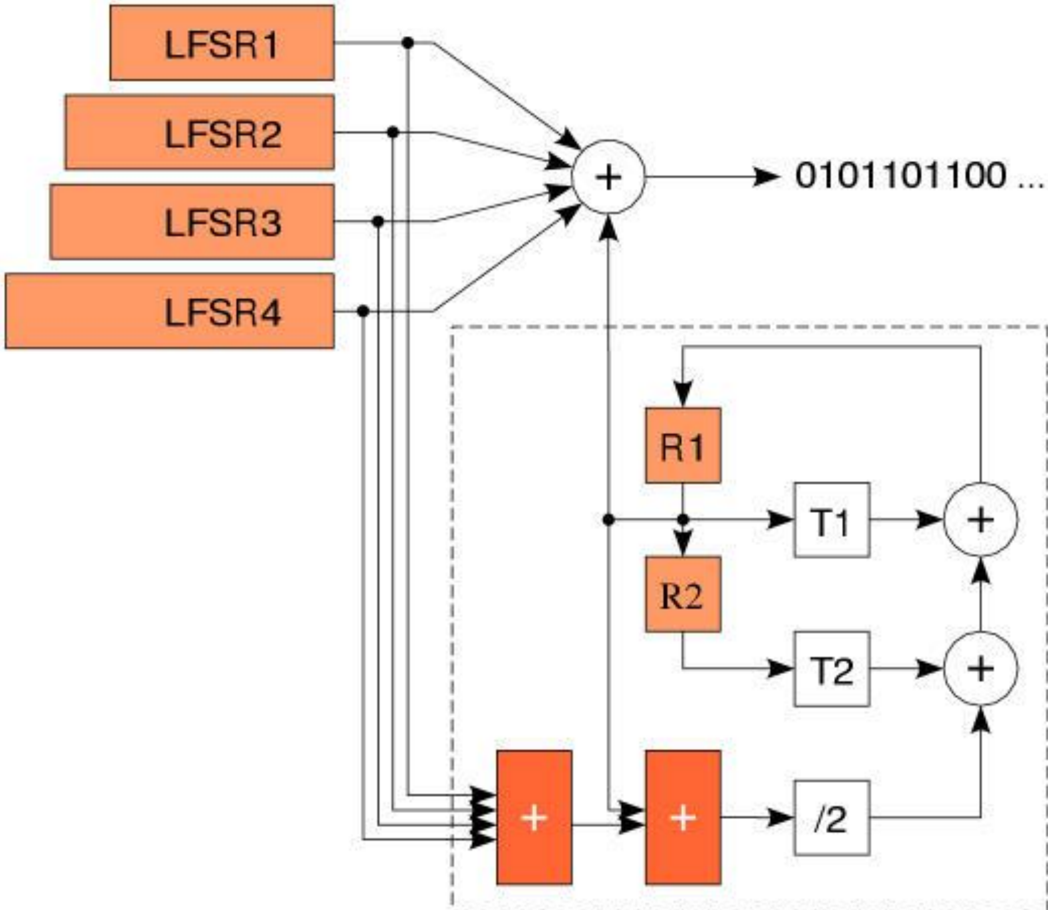
Clock control: registers agreeing with majority are clocked (2 or 3)

A5/1 stream cipher (GSM)

A5/1 attacks

- exhaustive key search: 2^{64} (or rather 2^{54})
  - Hardware 10K$ < 1 minute ciphertext only
- search 2 smallest registers: 2^{45} steps
- [BWS00] 1 minute on a PC
  - 2 seconds of known plaintext
  - 2^{48} precomputation, 146 GB storage
- [BB05]: 10 minutes on a PC,
  - 3-4 minutes of ciphertext only

Bluetooth stream cipher



brute force: 2^{128} steps

[Lu+05] 24 known bits of 2^{24} frames, 2^{38} computations, 2^{33} memory

A simple cipher: RC4 (1987)

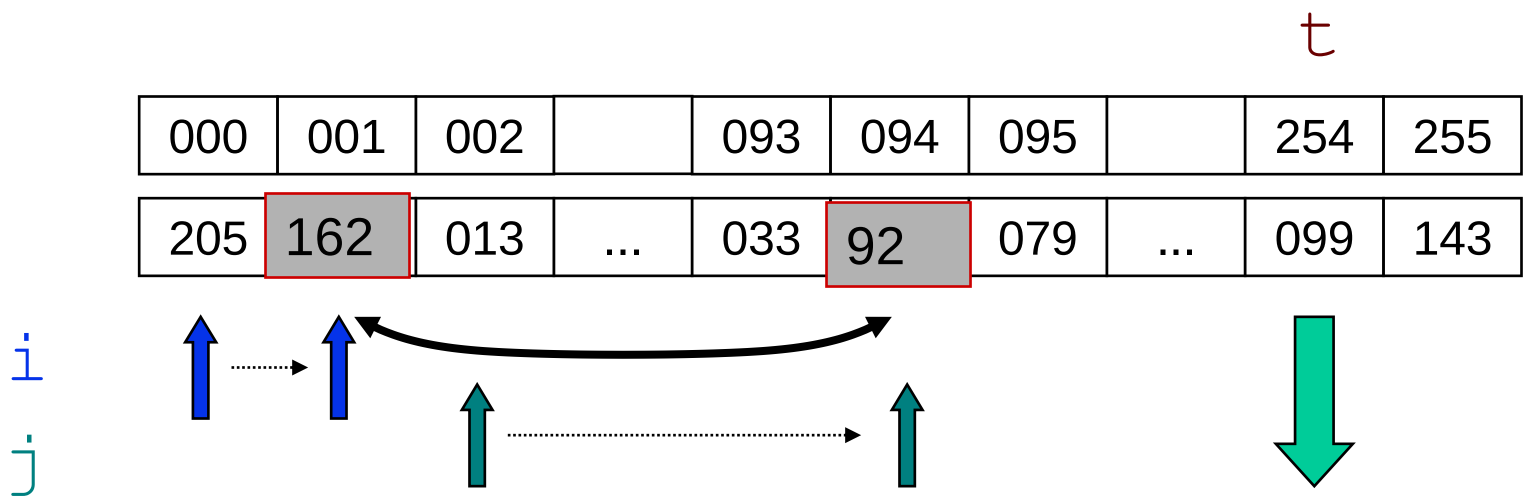
- designed by Ron Rivest (MIT)
- leaked in 1994
- S[0..255]: secret table derived from user key K

```
for i=0 to 255 S[i]:=i
j:=0
for i=0 to 255
    j:=(j + S[i] + K[i]) mod 256
    swap S[i] and S[j]
i:=0,  j:=0
```

A simple cipher: RC4 (1987)

Generate key stream which is added to plaintext

```
i:=i+1
j:=(j + S[i]) mod 256
swap S[i] and S[j]
t:=(S[i] + S[j]) mod 256
output S[t]
```



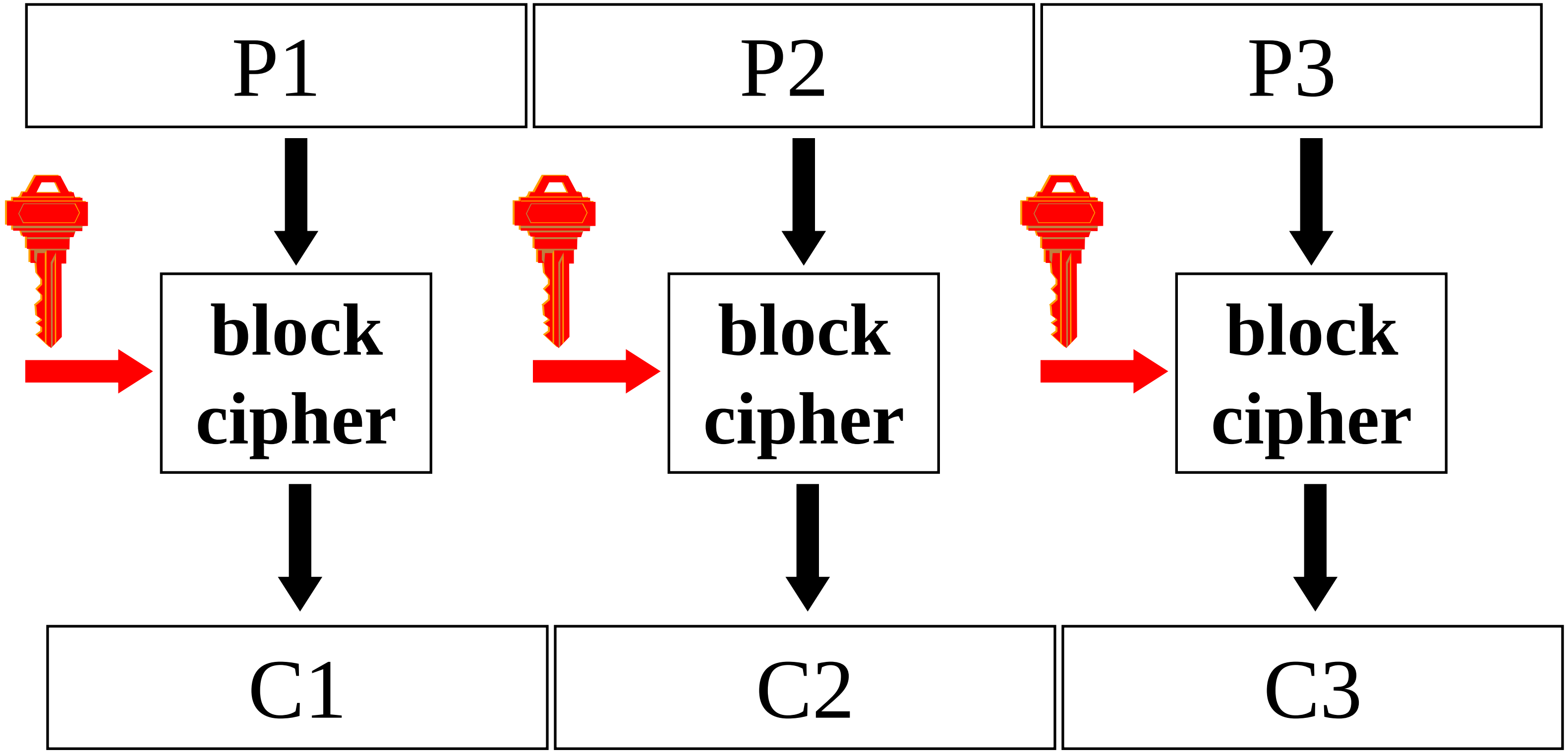
RC4: weaknesses

- often used with 40-bit key
  - US export restrictions until Q4/2000
- best known general shortcut attack: 2^{300}
- weak keys and key setup (shuffle theory)
- some statistical deviations
  - e.g., 2nd output byte is biased
  - solution: drop first 256 bytes of output
- problem with resynchronization modes (WEP)

Block cipher

- large table: list n-bit ciphertext for each n-bit plaintext
  - if n is large: very secure (codebook)
  - but for an n-bit block: 2^n values
  - impractical if $n \geq 32$
- alternative n = 64 or 128
  - simplify the implementation
  - repeat many simple operations

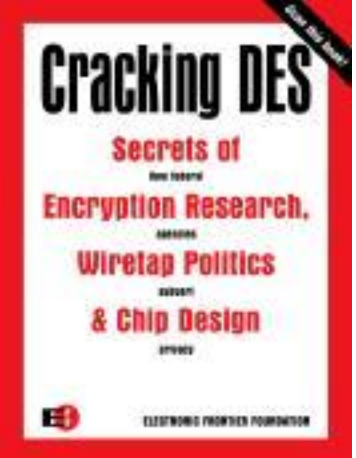
Block cipher (2)



- larger data units: 64…128 bits
- memoryless
- repeat simple operation (round) many times

Data Encryption Standard (1977)

- encrypts 64 plaintext bits under control of a 56-bit key
- 16 iterations of a relatively simple mapping
- FIPS: US government standard for sensitive but unclassified data
- worldwide de facto standard since early 80ies
- surrounded by controversy



Security of DES (56 bit key)

- PC: trying 1 DES key: 15 ns
- Trying all keys on 250 PCs:
  1 month: $2^{26} \times 2^{16} \times 2^{5} \times 2^{8} = 2^{55}$
- M. Wiener's design (1993):
  1,000,000 $ machine: 3 hours
  (in 2009: 10 seconds)

EFF Deep Crack (July 1998)

250,000 $ machine: 50 hours…

DES: security (ct'd)

- Moore's "law": speed of computers doubles every 18 months
  - key lengths need to grow in time
- Use new algorithms with longer keys
  - adding 1 key bits doubles the work for the attacker
- Key length recommendations in 2009
  - < 64 bits: insecure
  - 80 bits: 3-5 years
  - 100 bits: 20-25 years

Federal Register, July 24, 2004

DEPARTMENT OF COMMERCE

National Institute of Standards and Technology
[Docket No. 040602169– 4169– 01]

Announcing Proposed Withdrawal of Federal Information Processing Standard (FIPS) for the Data Encryption Standard (DES) and Request for Comments

AGENCY: National Institute of Standards and Technology (NIST), Commerce.

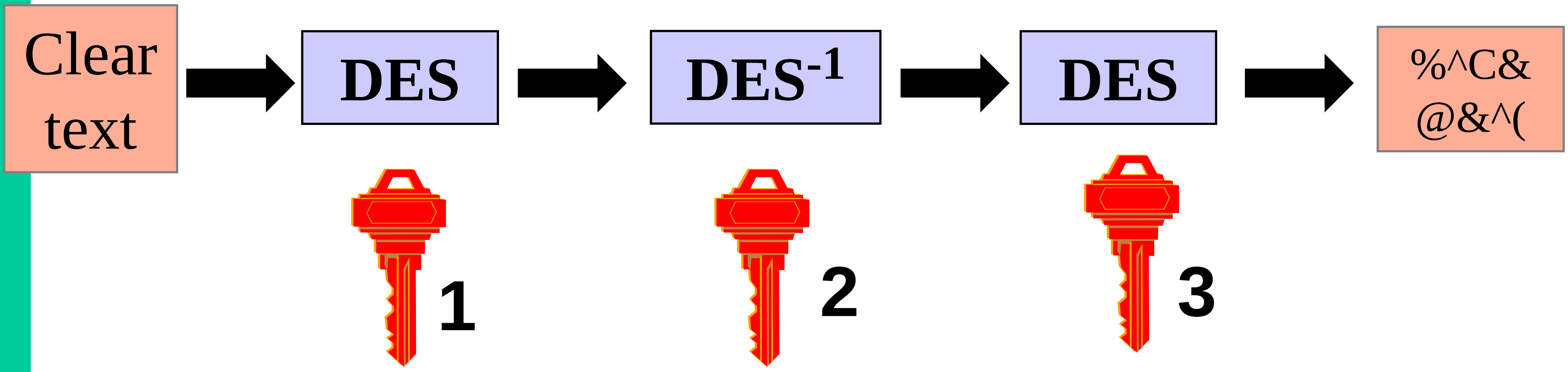
ACTION: Notice; request for comments.

- **SUMMARY:** The Data Encryption Standard (DES), currently specified in Federal Information Processing Standard (FIPS) 46–3, was evaluated pursuant to its scheduled review. At the conclusion of this review, **NIST determined that the strength of the DES algorithm is no longer sufficient to adequately protect Federal government information**. As a result, NIST proposes to withdraw FIPS 46–3, and the associated FIPS 74 and FIPS 81. Future use of DES by Federal agencies is to be permitted only as a component function of the Triple Data Encryption Algorithm (TDEA).

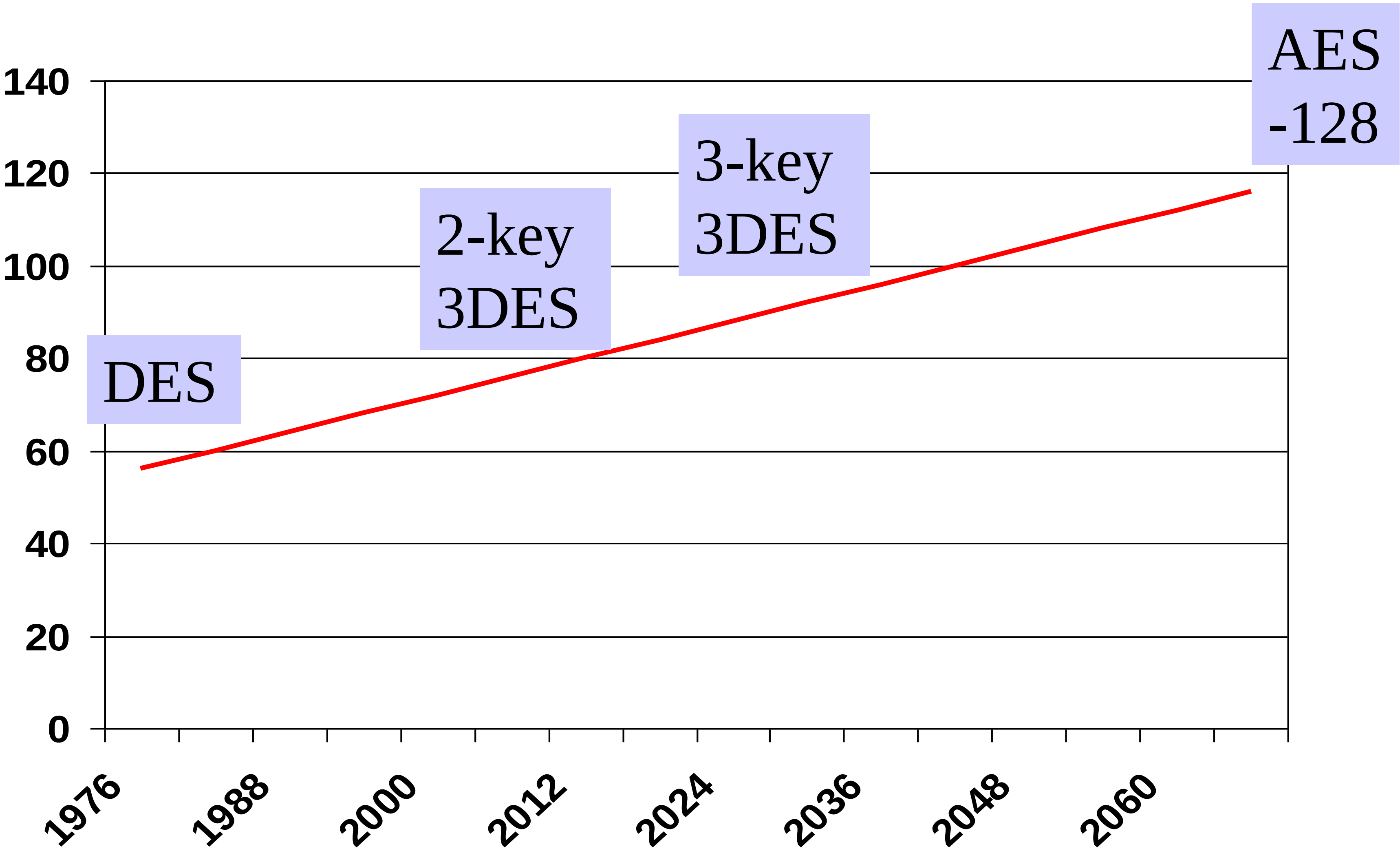
3-DES: NIST Spec. Pub. 800-67

(May 2004)

- two-key triple DES: until 2009
- three-key triple DES: until 2030



Symmetric Key Lengths and Moore's "law"



Moore's "law": speed of computers doubles every 18 months

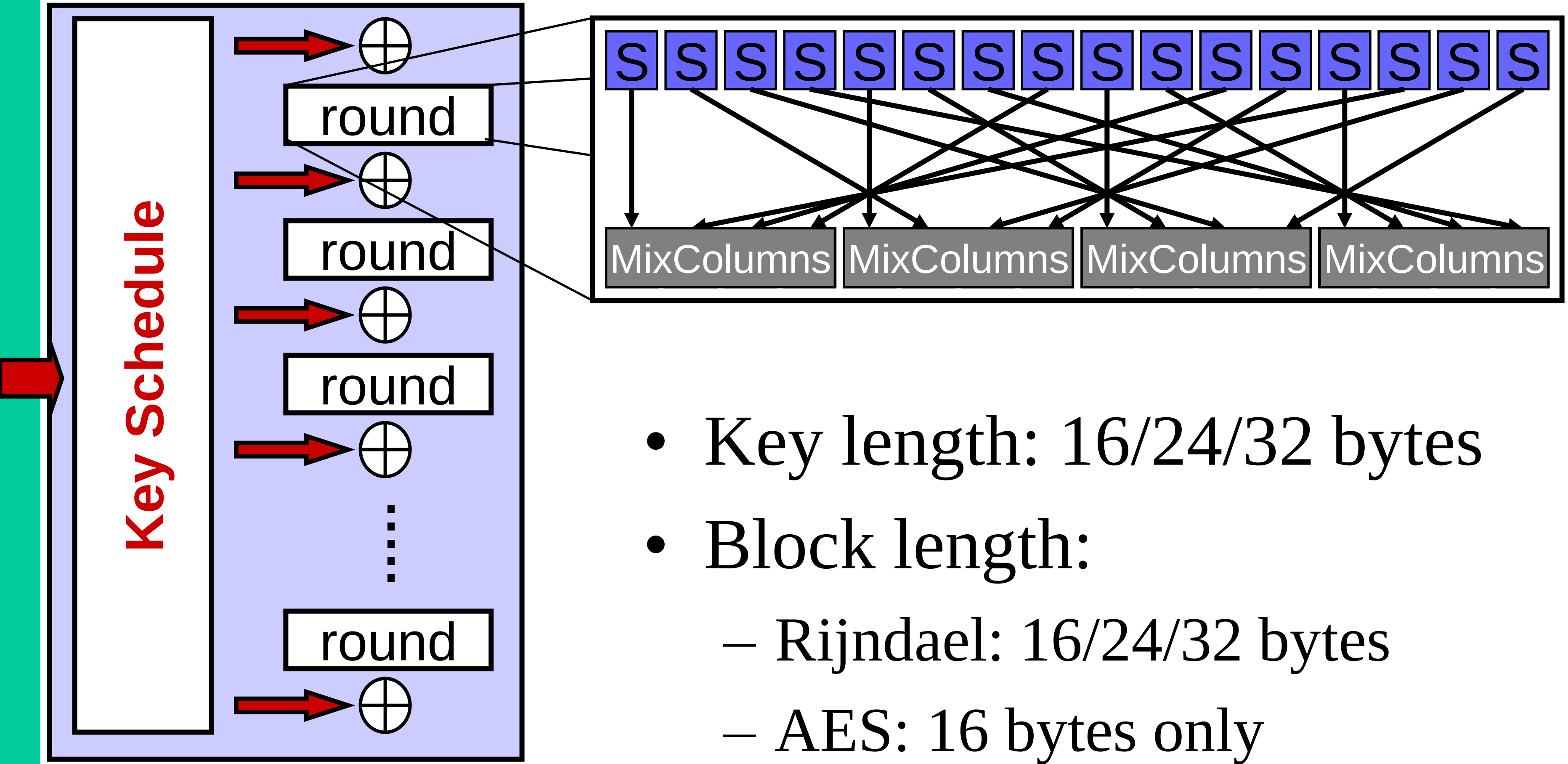
AES (Advanced Encryption Standard)

- open competition launched by US government (Sept. '97) to replace DES
- 22 contenders including IBM, RSA, Deutsche Telekom

- 128-bit block cipher with key of 128/192/256 bits
- as strong as triple-DES, but more efficient
- royalty-free

A machine that cracks a DES key in 1 second would take 149 trillion years to crack a 128-bit key

AES: Rijndael



- Key length: 16/24/32 bytes
- Block length:
  - Rijndael: 16/24/32 bytes
  - AES: 16 bytes only

AES Status

- FIPS 197 published on November 6, 2001, effective May 26, 2002
- mandatory for sensitive US govt. information
- fast adoption in the market (thousands of products)
  - Jan. 09 > 976 AES product certifications by NIST
  - standardization: ISO, IETF, IEEE 802.11,…
- slower adoption in financial sector
- mid 2003: AES-128 also for classified information and AES-192/-256 for *secret* and *top secret* information!
- Intel will provide AES instruction from 2009

Symmetric cryptology: data authentication

- the problem
- hash functions without a key
  - MDC: Manipulation Detection Codes
- hash functions with a secret key
  - MAC: Message Authentication Codes

Data authentication: the problem

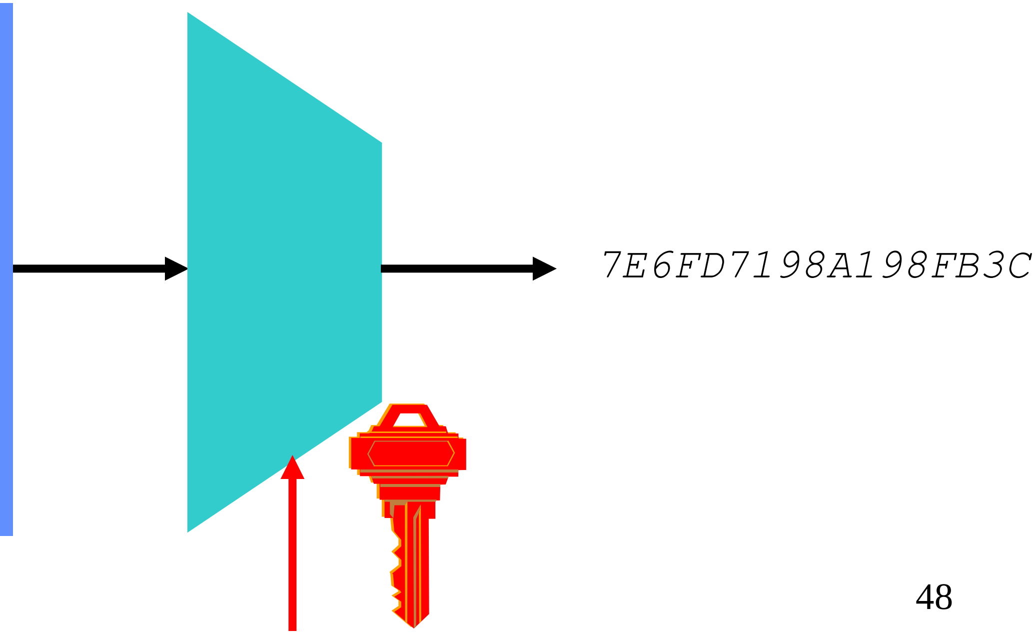
- encryption provides confidentiality:
  - prevents Eve from learning information on the cleartext/plaintext
  - but does not protect against modifications (active eavesdropping)
- Bob wants to know:
  - the **source** of the information (data origin)
  - that the information has not been **modified**
  - (optionally) **timeliness** and **sequence**
- data authentication is typically more complex than data confidentiality

Data authentication: MAC algorithms

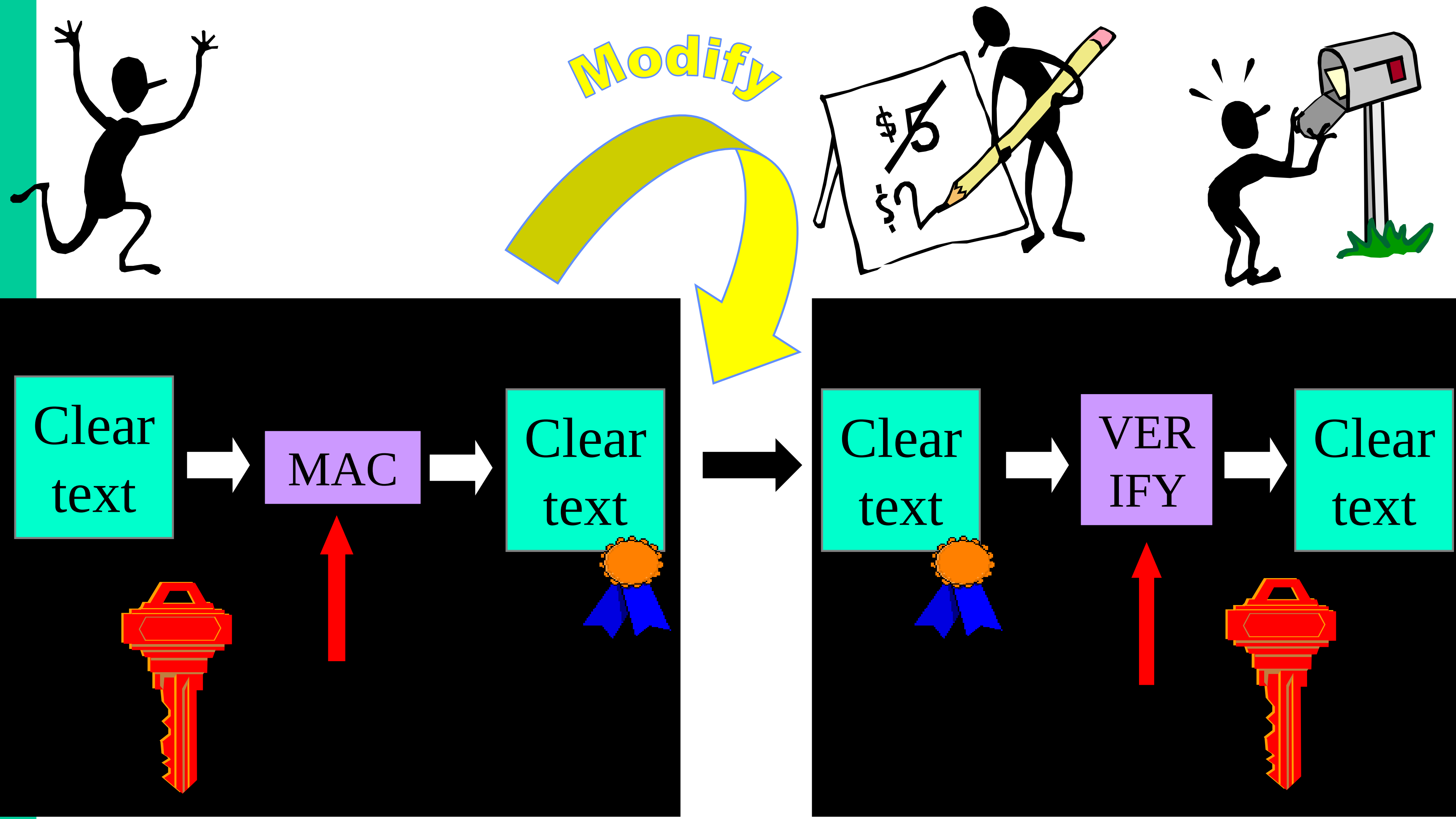
- Replace protection of authenticty of (long) message by protection of secrecy of (short) key
- Add MAC to the plaintext

- CBC-MAC
- HMAC

This is an input to a MAC algorithm. The input is a very long string, that is reduced by the hash function to a string of fixed length. There are additional security conditions: it should be very hard for someone who does not know the secret key to compute the hash function on a new input.



MAC algorithms



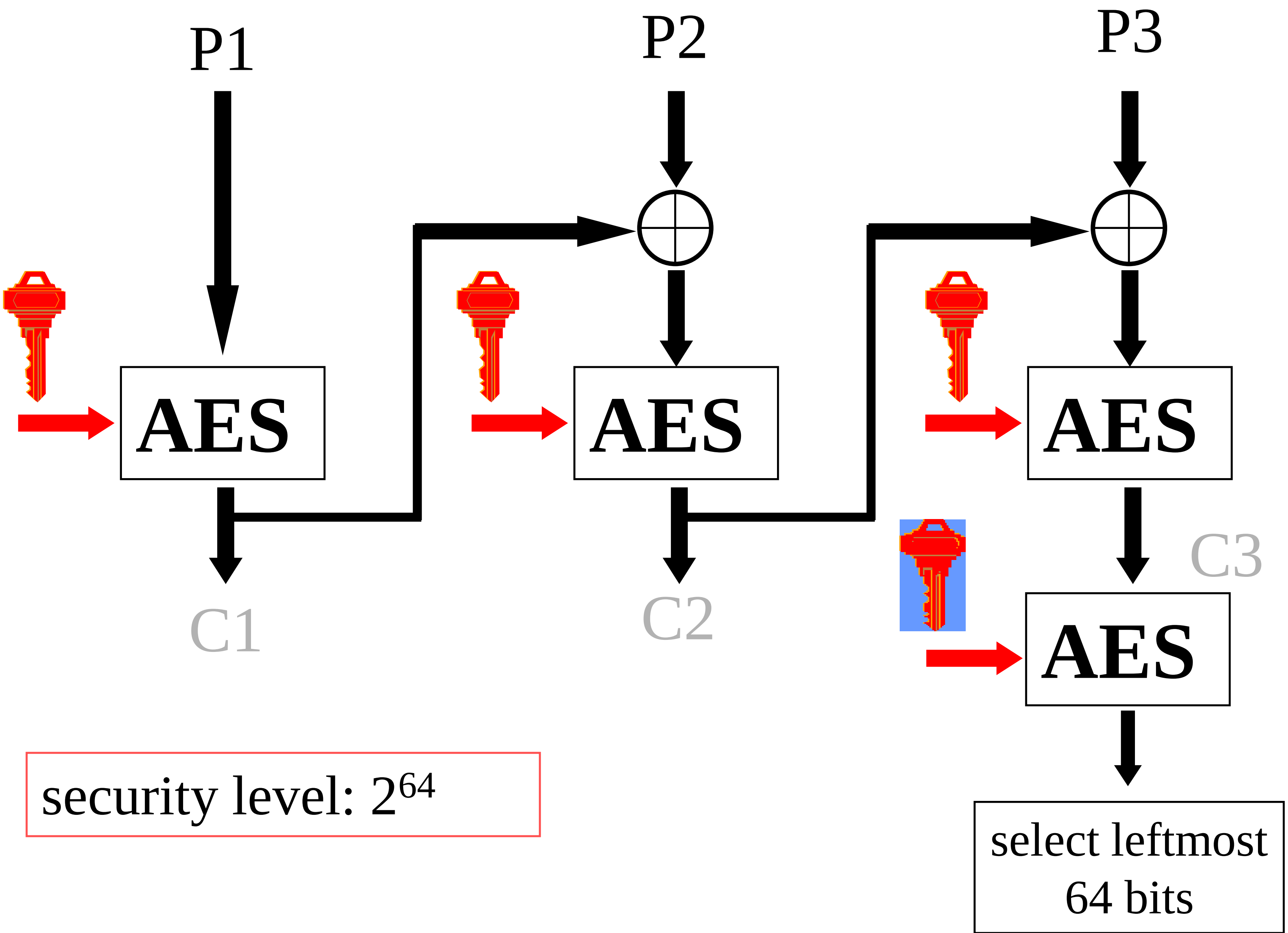
Data authentication: MAC algorithms

- typical MAC lengths: 32..96 bits
  - Forgery attacks: 2^m steps with m the MAC length in bits
- typical key lengths: (56)..112..160 bits
  - Exhaustive key search: 2^k steps with k the key length in bits
- birthday attacks: security level smaller than expected

MAC algorithms

- Banking: CBC-MAC based on triple-DES
- Internet: HMAC and CBC-MAC based on AES
- information theoretic secure MAC algorithms (authentication codes):
  - highly efficient
  - rather long keys
  - part of the key refreshed per message

CBC-MAC based on AES



security level: 2^{64}

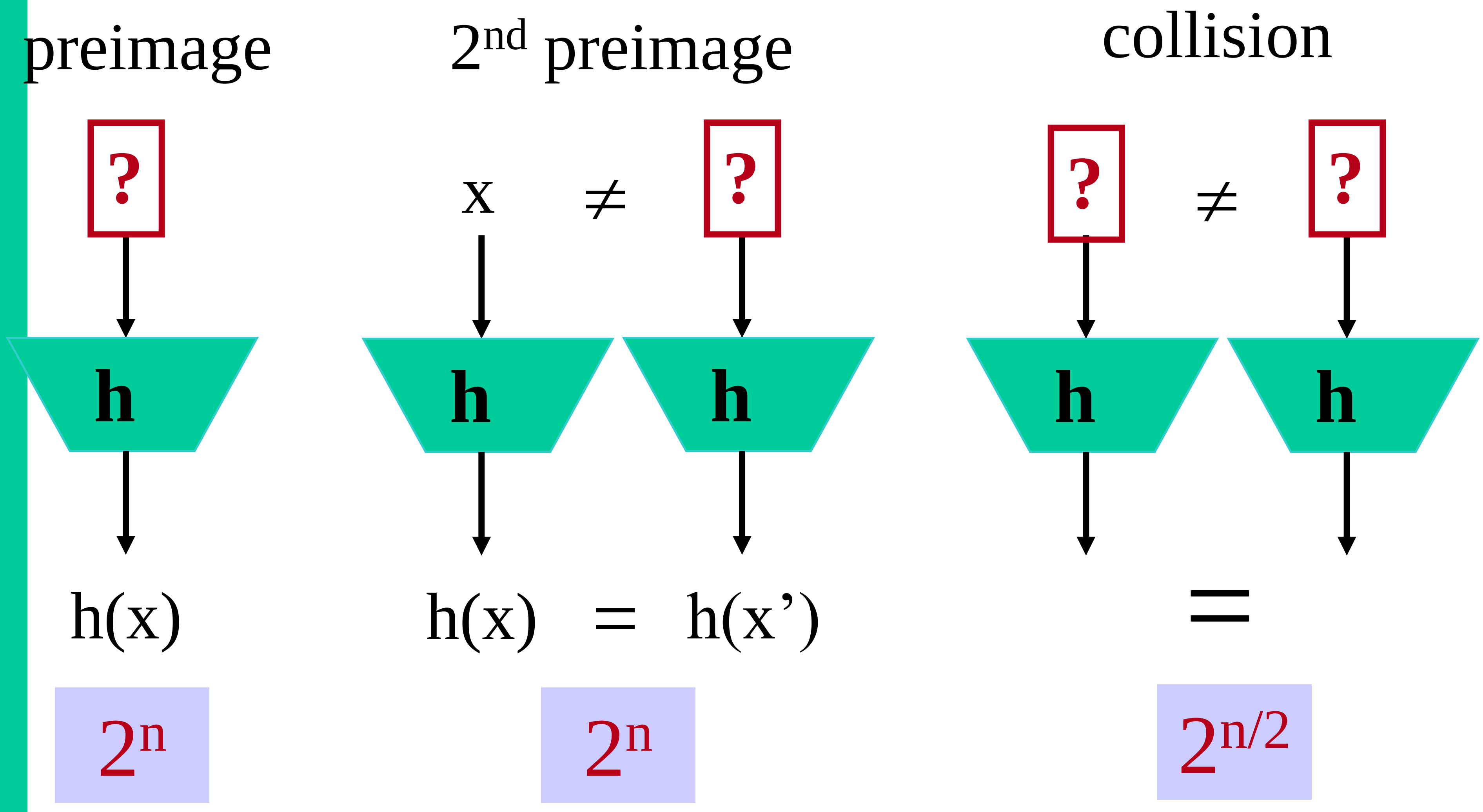
Data authentication: MDC

- MDC (manipulation detection code)
- Protect short hash value rather than long text

- (MD5)
- (SHA-1), SHA-256, SHA-512
- RIPEMD-160

This is an input to a cryptographic hash function. The input is a very long string, that is reduced by the hash function to a string of fixed length. There are additional security conditions: it should be very hard to find an input hashing to a given value (a preimage) or to find two colliding inputs (a collision).

MDC Security requirements (n-bit result)



Data authentication: MDC

- n-bit result
- preimage resistance: for given y, hard to find input x such that h(x) = y (2^n operations)
- 2^{nd} preimage resistance: hard to find x'≠ x such that h(x') = h(x) (2^n operations)
- Collision resistance: hard to find (x,x')
with x' ≠ x such that h(x') = h(x)
($2^{n/2}$ operations)

MD5 and SHA-1

- SHA-1:
  - (2nd) preimage 2^{160} steps
  - collisions 2^{80} steps

60 M$ for 1 year

Shortcut: Aug. 2007: 2^{60} steps

- MD5
  - (2nd) preimage 2^{128} steps
  - collisions 2^{64} steps

15 K$ for 1 month

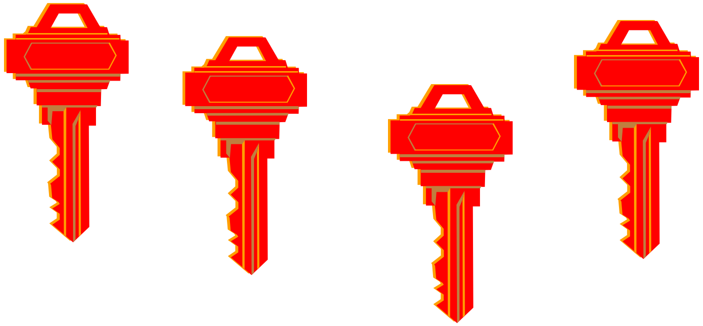
Shortcut: Aug. 2004: 2^{39} steps (today: seconds)

Public-key cryptology

- the problem
- public-key encryption
- digital signatures
- an example: RSA
- advantages of public-key cryptology

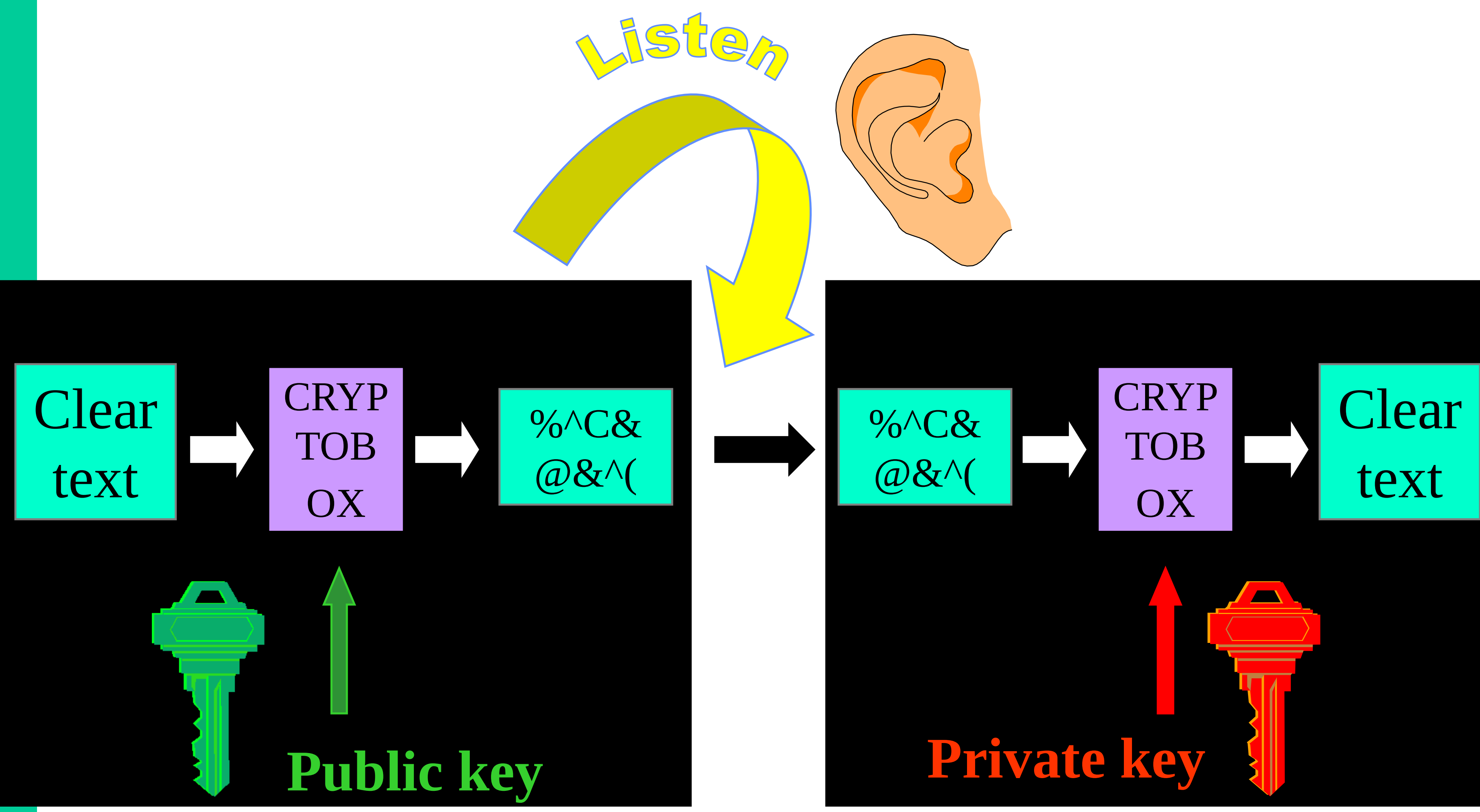
Limitation of symmetric cryptology

- Reduce security of information to security of keys

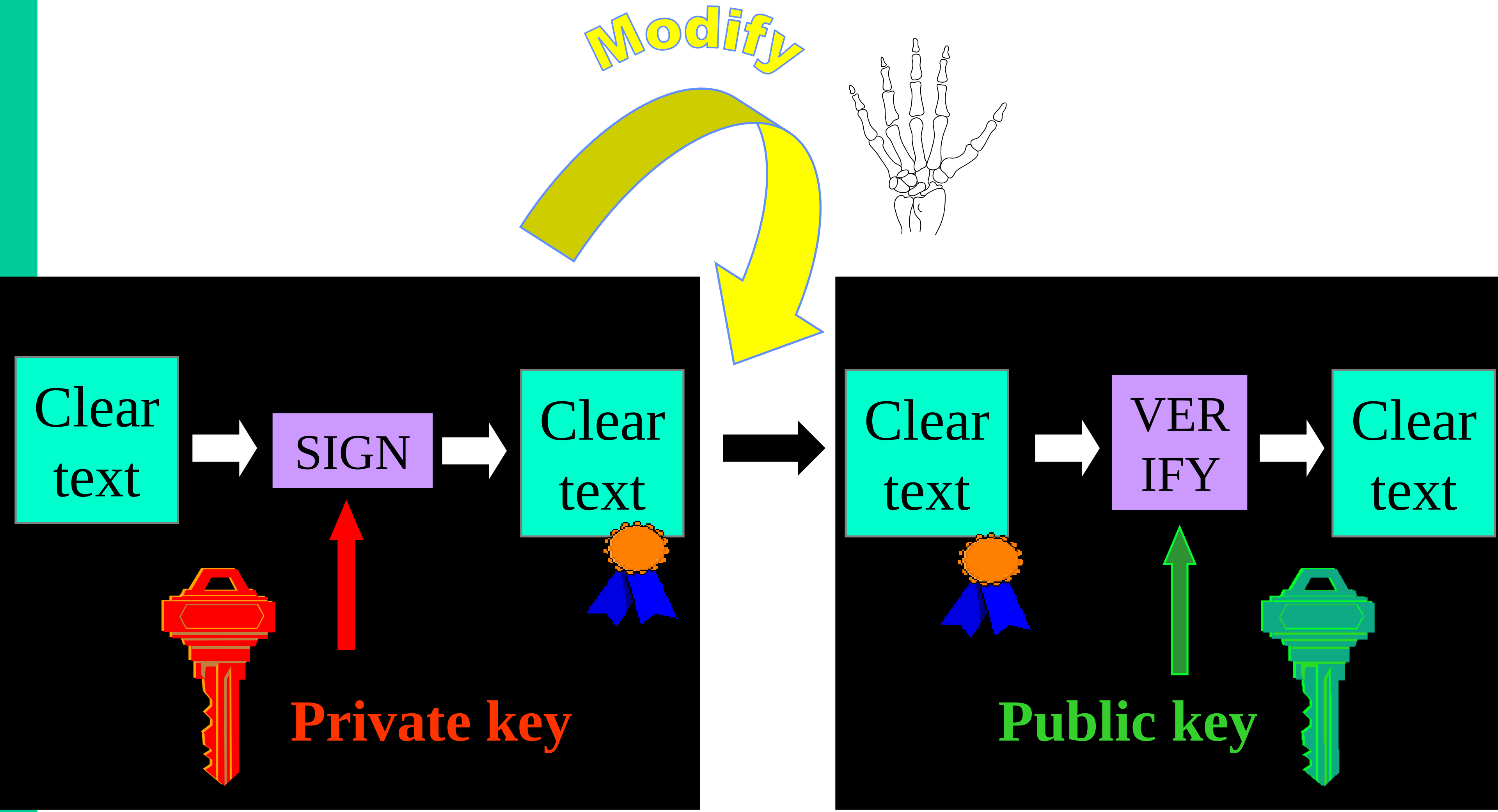


- But: how to establish these secret keys?
  - Cumbersome and expensive
  - Or risky: all keys in 1 place
- Do we really need to establish secret keys?

Public key cryptology: encryption



Public key cryptology: digital signature



A public-key distribution protocol: Diffie-Hellman

- Before: Alice and Bob have never met and share no secrets; they know a public system parameter α

generate x | α^{x} → | *generate* y

compute α^{x} | ← α^{y} | *compute* α^{y}

compute $k=(\alpha^{y})^{x}$ | | *compute* $k=(\alpha^{x})^{y}$

- After: Alice and Bob share a short term key k
  – Eve cannot compute $\boldsymbol{k}$: in several mathematical structures it is hard to derive x from α^{x} (this is known as the discrete logarithm problem)

RSA ('78)

- Choose 2 "large" prime numbers p and q
- modulus n = p.q
- compute $\lambda(n) = \mathrm{lcm}(p-1,q-1)$
- choose e relatively prime w.r.t. $\lambda(n)$
- compute $d = e^{-1} \bmod \lambda(n)$

The security of RSA is based on the "fact" that it is easy to generate two large primes, but that it is hard to factor their product

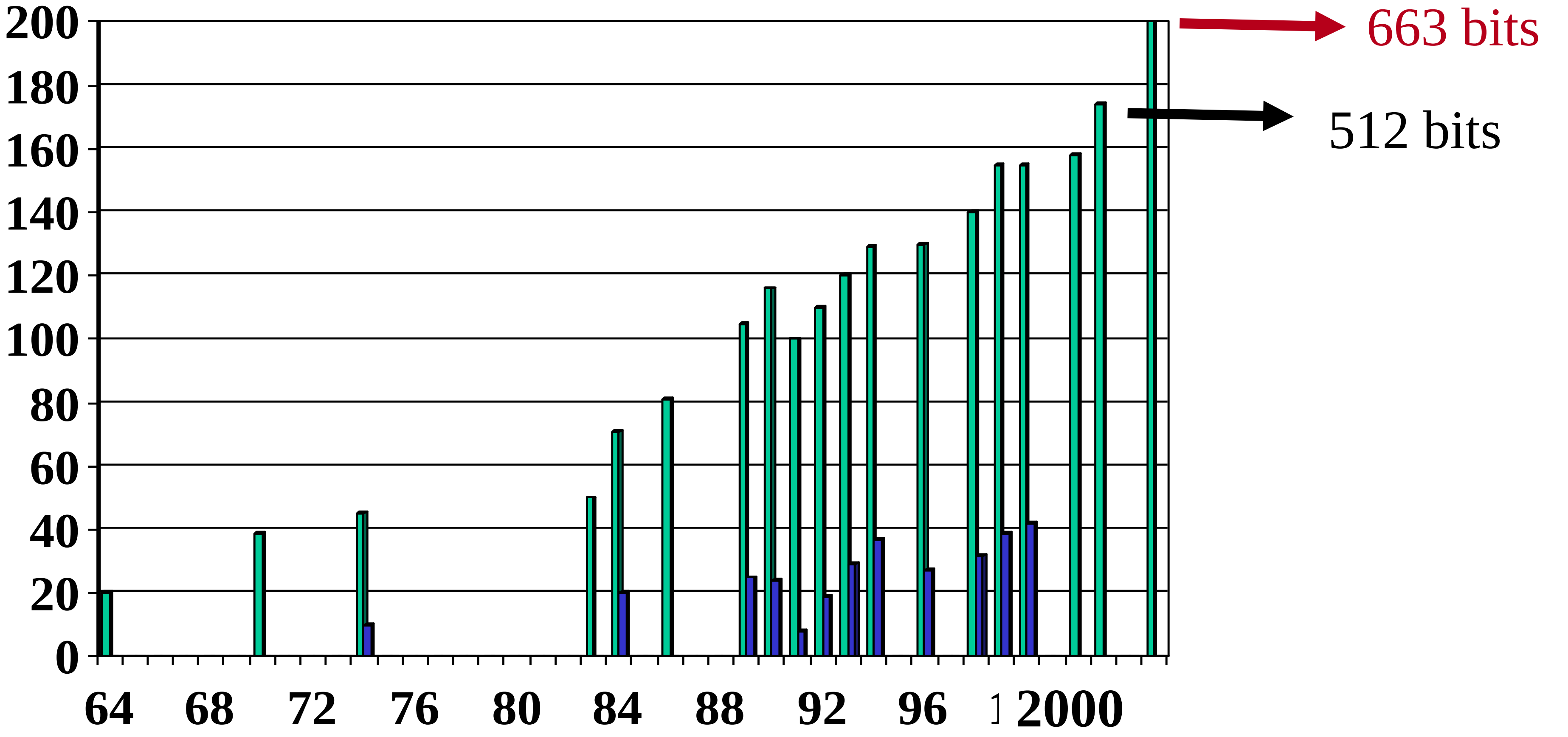
- public key = (e,n)
- private key = d of (p,q)

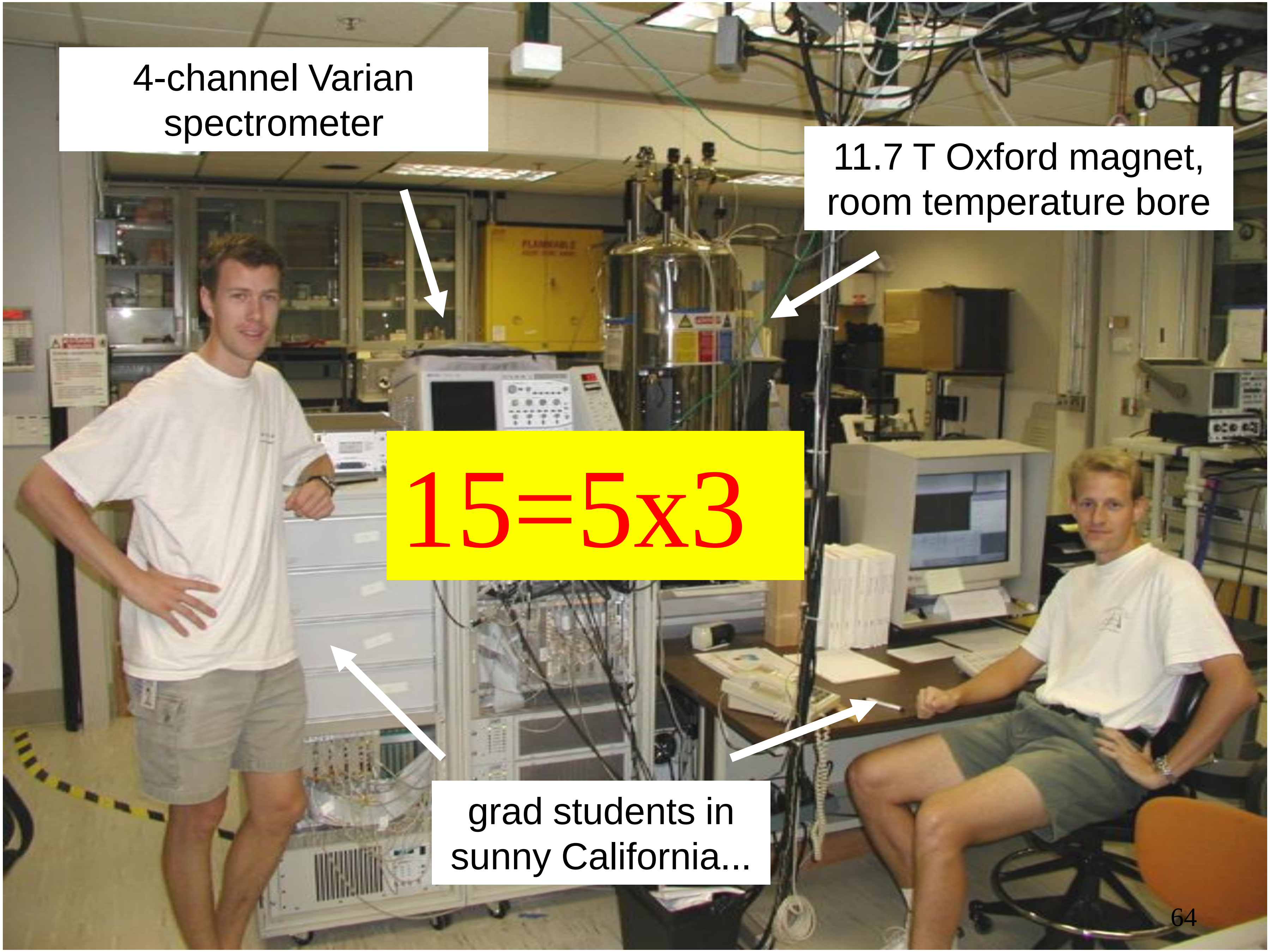
- encryption: $c = m^e \bmod n$
- decryption: $m = c^d \bmod n$

try to factor 2419

Factorisation records

1 digit ~3.3 bits




4-channel Varian spectrometer
11.7 T Oxford magnet, room temperature bore
15=5x3
grad students in sunny California...

Advantages of public key cryptology

- Reduce protection of information to protection of authenticity of public keys
- Confidentiality without establishing secret keys
  - extremely useful in an open environment
- Data authentication without shared secret keys: digital signature
  - sender and receiver have different capability
  - third party can resolve dispute between sender and receiver

Disadvantages of public key cryptology

- Calculations in software or hardware two to three orders of magnitude slower than symmetric algorithms
- Longer keys: 1024 bits rather than 56…128 bits
- What if factoring is easy?

Crypto software libraries

C/C#

- Botan
- Cryptlib
- Crypto
- Libgcrypt
- Miracl
- OpenSSL

- BouncyCastle (BC#)

Java

- SunJCA/JCE
- BouncyCastle (BC)
- CryptixCrypto
- FlexiProvider
- GNU Crypto
- IAIK
- RSA JSafe

Reading material

- B. Preneel, Modern cryptology: an introduction.
  - This text corresponds more or less to the second half of these slides
  - It covers in more detail how block ciphers are used in practice, and explains how DES works.
  - It does not cover identification, key management and application to network security.

Selected books on cryptology

- D. Stinson, Cryptography: Theory and Practice, CRC Press, 3rd Ed., 2005. Solid introduction, but only for the mathematically inclined.
- A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997. The bible of modern cryptography. Thorough and complete reference work – not suited as a first text book. Freely available at http://www.cacr.math.uwaterloo.ca/hac
- N. Smart, Cryptography, An Introduction: 3rd Ed., 2008. Solid and up to date but on the mathematical side. Freely available at http://www.cs.bris.ac.uk/~nigel/Crypto_Book/
- B. Schneier, Applied Cryptography, Wiley, 1996. Widely popular and very accessible – make sure you get the errata.
- Other authors: Johannes Buchmann, Serge Vaudenay

Books on network security and more

- W. Stallings, *Network and Internetwork Security: Priniples and Practice*, Prentice Hall, 2004. Solid background on network security. Explains basic concepts of cryptography. Tends to confuse terminology for decrypting and signing with RSA.
- Nagand Doraswamy, Dan Harkins, IPsec - *The New Security Standard for the Internet, Intranets, and Virtual Private Networks*, Prentice Hall, 1999. A well written overview of the IPsec protocol (but now outdated).
- W. Diffie, S. Landau, *Privacy on the line. The politics of wiretapping and encryption*, MIT Press, 2007. The best book so far on the intricate politics of the field.

More information: some links

- IACR (International Association for Cryptologic Research): www.iacr.org
- IETF web site: www.ietf.org
- Cryptography faq: www.faqs.org/faqs/cryptography-faq
- Counterpane links: www.counterpane.com/hotlist.html
- Digicrime (www.digicrime.org) - not serious but informative and entertaining