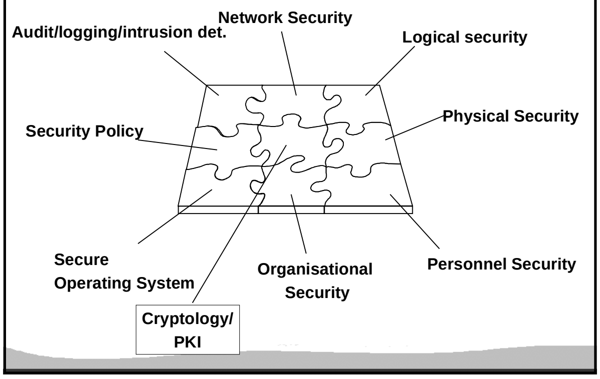
An Introduction to Symmetric Cryptology

Prof. Bart Preneel
Katholieke Universiteit Leuven, Belgium
Bart.Preneel@esat.kuleuven.ac.be
http://www.esat.kuleuven.ac.be/~preneel

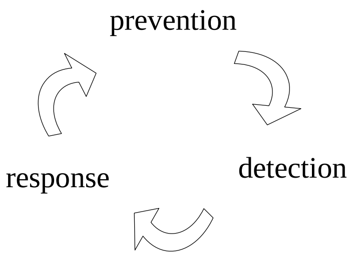
Outline

- concepts
- algorithms
  - symmetric algorithms for confidentiality
  - symmetric algorithms for data authentication
- how hard is it to invert a one-way function?

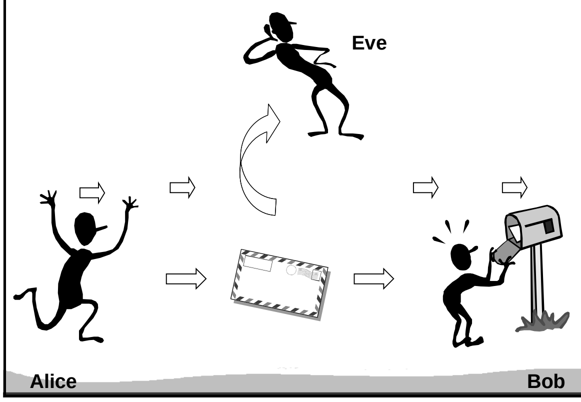
Information security: a puzzle



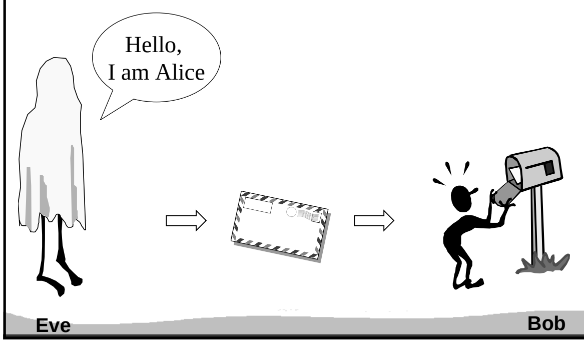
Process approach to security



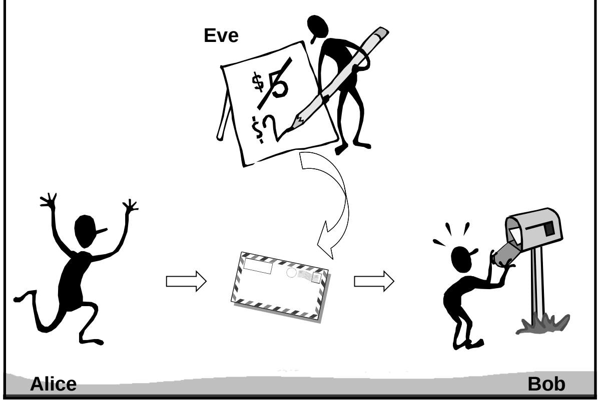
Data confidentiality



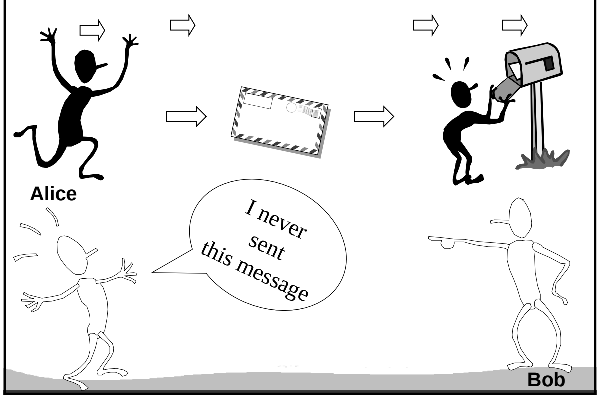
Entity authentication



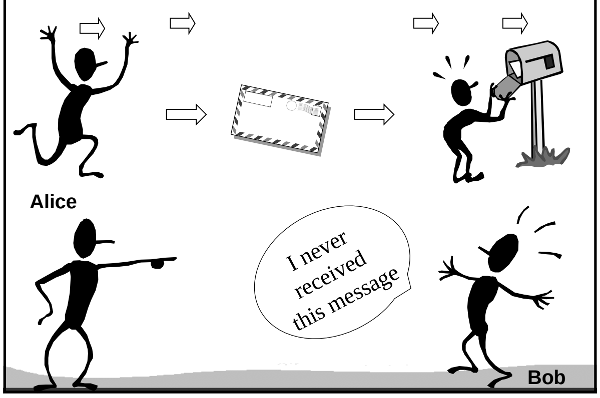
Data authentication



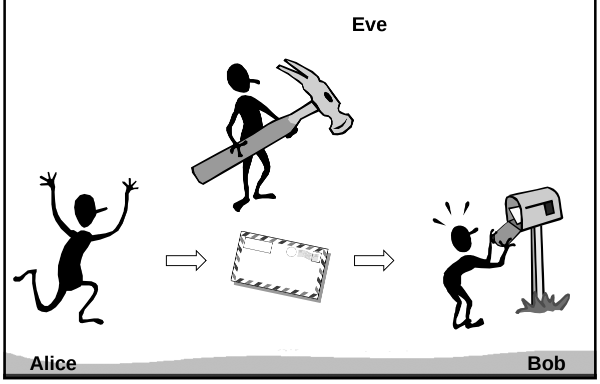
Non-repudiation (origin)



Non-repudiation (receipt)



Denial of service



Definitions

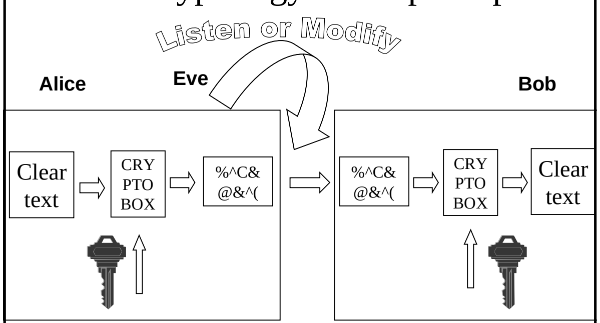
	data	entities
confidentiality	encryption	anonymity
authentication	data authentication	identification

Non-repudiation of origin, receipt

Contract signing

Notarisation and Timestamping

Cryptology: basic principles



Symmetric cryptology: confidentiality

- old cipher systems:
  - transposition, substitution, rotor machines
- the opponent and her power
- the Vernam scheme
- A5/1, Bluetooth, RC4
- DES and triple-DES
- AES

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:

```
HJAEG JAWFW FNGQW JMKMJ
IKBFH KBXGX GOHRX KNLNK
JLCGI LCYHY HPISY LOMOL
KMDHJ MDZIZ IQJTZ MPNPM
LNEIK NEAJA JRKUA NQOQN
MOFGL OFBKB KSLVB ORPRO
NPGHM PGCLC LTMWC PSQSP
OQHLN QHDMD MUNXD QTRTQ
PRIMO RIENE NVOYE RUSUR
QSJNP SJFOF OWPZF SVTVS
RTKOQ TKGPG PXQAG TWUWT
```

Old cipher systems (pre-1900) (2)

- Substitutions
  - ABCDEFGHIJKLMNOPQRSTUVWXYZ
  - MZNJSOAXFQGYKHLUCTDVWBIPER
- Transpositions

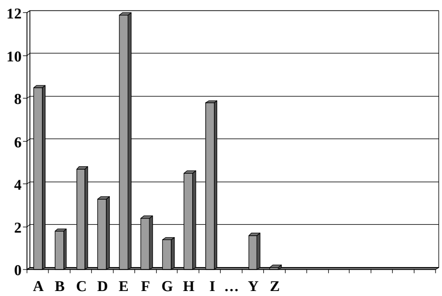
```
TRANS      ORI S
POSIT      NOTIT
IONS       OSANP
```

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....

Letter distributions

Easy to break simple substitution using statistical techniques



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 plaintxt (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

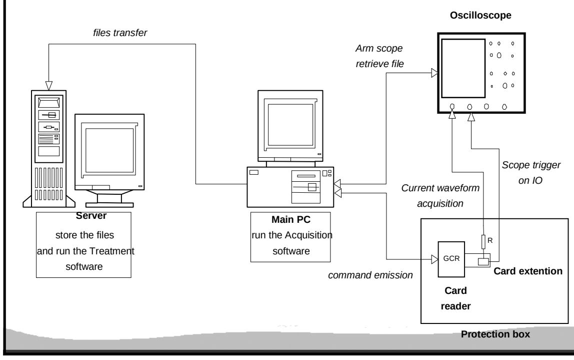
Assumptions on Eve (the opponent)

- A scheme is broken if Eve can deduce the key or obtain additional plaintext
- Eve can always try all possible 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

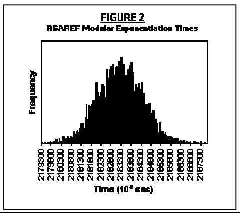
New assumptions on Eve

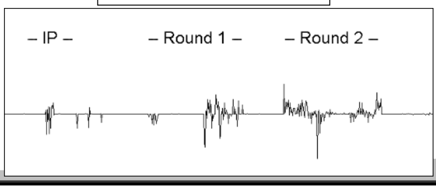
- Eve may have access to side channels
  - timing attacks
  - simple power analysis
  - differential power analysis
  - differential fault analysis
  - electromagnetic interference

Side channel analysis

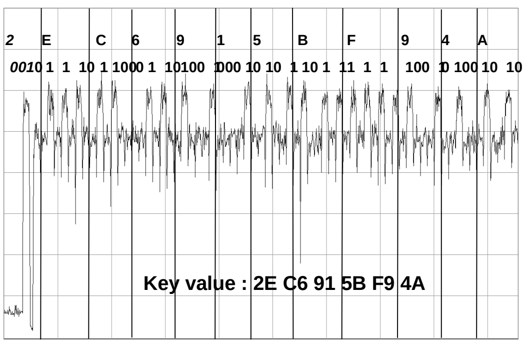


Timing attacks and power analysis

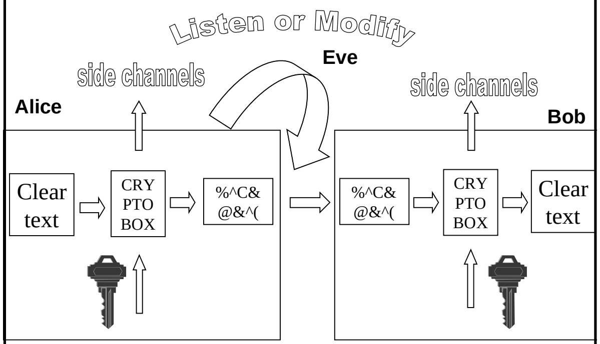




A simple attack on RSA (courtesy: Gemplus)

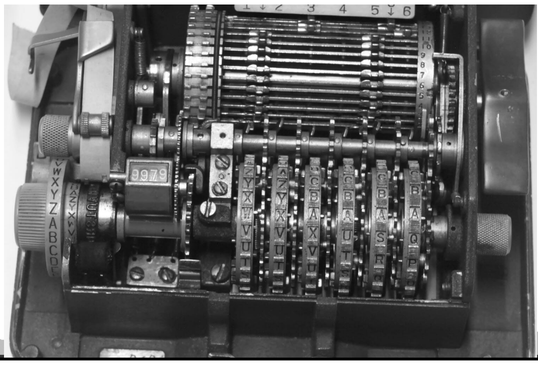


Cryptology + side channels



The Rotor machines (WW II)

Mechanical: Hagelin C38



Congolese history 101

- Independence of Congo: 30 June 1960
- first president: Kasa Vubu
- first prime minister: Patrice Lumumba
- Tshombé (Katanga)
- Belgium: government, king, industry (UM)
- United Nations, Dag Hammerskjöld
- USA
- USSR

Congolese history 101 (2)

- 5 September 1960: L fired
- 10 October 1960: L arrested
- 17 January 1961: L transported to Katanga and executed
- US Congress (Church report, 1975)
  - No US involvement
- Belgian Parliament: investigation
  - May 2nd 2000-October 31 2001

"historians refuse to decipher cryptograms, as this may reveal compromising information"

Problem (17-09-01)

- 15 telexes of 12/1960 - 2/1961
- Minaf - Rusur: 4 telexes in OTPL
- Minaf -Brazzaville and Minaf -E'ville:
  - 11 telexes in "Printex"
  - for 5 (part of) the cleartext is known
  - for 1 incorrect cleartext is available
  - a few "real keys" were known
- "please decrypt within 3 weeks"

Example (1) #14

- Brazza 28b (stamp: 15-2-1961)
  - Jacques to Nicolas
- Cryptogram 11150 [30x5=150]:
- 11150 HSMEO TDUYB ZJQZI VVRHP
  ELHIL FXUKQ MNAFF ZPWSE DOXPX
  NFPPA RNMXS RZPUG LBZAI MXNFC
  ZZSHR XVTZI DZABT LPEET CNHFV
  RSNUF CJTQI HUKYM XZWBG HTLMO
  SWLOH EVJLF NOFYV ROSYC WXDTE
  WVEXE ACKPT HSMEO 11150

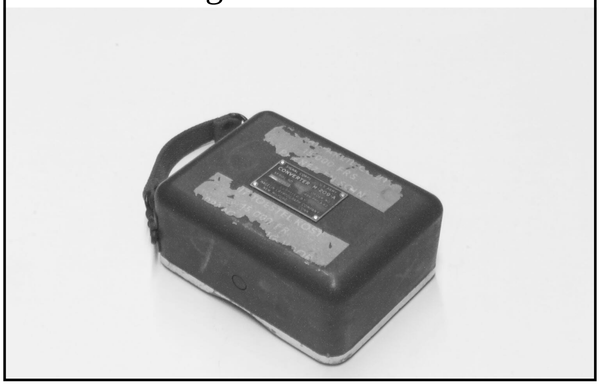
Example (2) #14

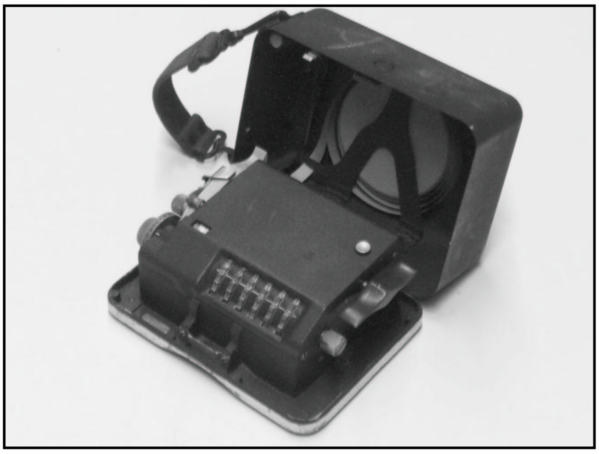
- Brazza 28b (stamp: 15-2-1961)
  - Jacques to Nicolas
- Cleartext:
- CONTINUE INTRIGUES INQUIETANTES
  TANT LEO QU EVILLE JACQUES
  BISSECT VOUS PRIE VOUS INFORMER
  DISCR?TEMENT MISSION EXACTE
  CONFIEE HUBERT STOP INTERESSEYX

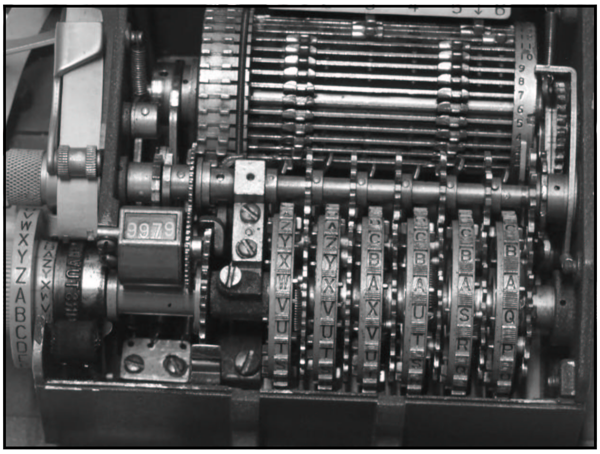
Problem: what is this? #5

- 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>

Hagelin C-38 = M-209







How does it work (C-38)

- 6 pins form a 6-bit word
- when a rotor pin encouters a lug, the bar is moved to the left and it shifts the plaintext over one position (non-linear)
- the total number of active bars is k
- the ciphertext is computed as 25-p+k
  - = involution

How to identify the right variant?

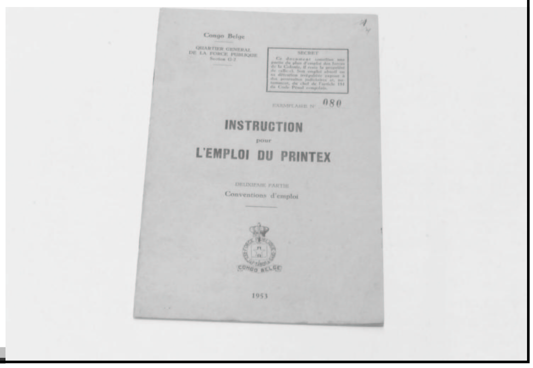
- 5 characters for false key suggest C-35 or C-36 with 5 rotors
- cryptanalysis was tried but failed

- rotors provide 5-bit address
- weights: 10-8-4-2-1
- very easy to go back from displacement to input address

How to identify the right variant?

- there was some particular behaviour for plaintext/ciphertext pairs with distances 26-25-23-21-19-17

How to use this?



Encryption (1): set up main key

- 131 pins on rotors
- drum: 2 lugs on 27 bars

- once every 2-3 months

Encryption (2): cleartext #11

- **<TRES SECRET CONTACT PRIS CE JOUR AVEC MANKOVKA ET RUDNICKI COUSIN DE MANKOWSKI STOP ACCORD PRINCIPE AIDE SEMBLE ACQUIS STOP SUBORDONNE CEPENDANT A EXAMEN SITUATION A EVILLE PAR RUDNICKI STOP AI SENTIMENT CE DEPLACEMENT PAS OPPORTUN STOP N'ETANT QUE INTERMEDIAIRE JE VOUS DEMANDE SI ACCORD CE VOYAGE STOP DEMANDE REPONSE URGENTE INTERESSE ATTENDANT ICI STOP RAPPELLE DISCRETION NECESSAIRE STOP JULES>**

Encrypt (3): prepare cleartext

- **REPON SEWUR GENTE WINTE RESSE WATTE NDANT WICIW XXWRA PPELL EWDIS CRETI OWNEC ESSAI REWXX WJULE SWBIS ECTWX XWTRE SECWC ONTAC TKWPI SWCEW JOURW AVECW MANKO VVKAW ETKWX UDNIC KIWCO USINW DEWMA NKOVV SKIWX XWACC ORDWP RINCI PEKWA IDEWS EMBLE WACQU ISWXX WSUBO RDONN EWCEP ENDDA NTWWA WEXAM ENWSI TULTI ONKWA WEVIL LEWPA RWRUD NICKI WXXWA IWSEN TIMEN TWCEW DEPLA CEMEN TWPAS WOPPO RTUNK WXXWN WETAN NTWQU EWINT ERMED IAIRE WJEWV OUSWD EMAND EKWSI WACCO RDWCE WVOYA GEWXX WDEMA NDEWK**

Encryption (4): choose starting positions of rotors

- choose 5 random letters: EXATF
- real key = starting position rotors (session key)
- encrypt with Playfair [1854]

G	X	L	N	S
K	H	T	W	O
Q	D	E	F	A
M	V	I	C	R
Z	B	P	U	J

yields false key: DLEOE (encrypted session key)

Encryption (4a)

- cleartext EX

G	X	L	N	S
K	H	T	W	O
Q	D	E	F	A
M	V	I	C	R
Z	B	P	U	J

- ciphertext DL

Encryption (4b)

- cleartext AT

G	X	L	N	S
K	H	T	W	O
Q	D	E	F	A
M	V	I	C	R
Z	B	P	U	J

- ciphertext EO

Encryption (4c)

- cleartext F

G	X	L	N	S
K	H	T	W	O
Q	D	E	F	A
M	V	I	C	R
Z	B	P	U	J

- ciphertext E

Encryption (4d)

- cleartext AO

G	X	L	N	S
K	H	T	W	O
Q	D	E	F	A
M	V	I	C	R
Z	B	P	U	J

- ciphertext OS

Encryption (4e): alternative (1958)

- agree beforehand on a session key of 5 random letters: EXATF
- set rotors to this position
- encrypt the letters AAAAA
- the false key (encrypted session key) is the corresponding ciphertext

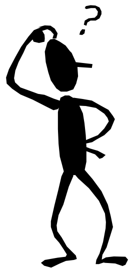
Encryption (5): use Printex

```
• <DLEOE EPEUZ DJWEX HBAAJ TNWRJ AQUCM
  VJPVI VPWHQ UGIQW THNEO THBXA BVSJE JIOBQ
  ZMEQH QTNQG WQIUU RFXLF SSTDD QLLTY TPCIF
  ZNPJN HIMSJ WAUFO RPKFX MHQIM TURPS SKELV
  AUVQY SMICQ RFAHD YOZKD KXGJY KDYJM HCLSO
  CHX e e e   CHWBP PUVUN LEONF OEYMO FBBMS
  OSNTV EBLFQ QKCXZ FDYOQ YBSIE HLUAR MNTQW
  LSMRT BQNAQ VPLOG EIZUH SYDYJ AQLAJ MGUHA
  NNTCF SSYBM AFJHM TRMQQ AQVQE FHBBZ BBJLN
  HQKNV XJXHJ VWAPA YVITU ZMXAG ZSPVF XGWQJ
  YZNTL OSPHP FTFLS EPLDB VQLUZ BORAJ LLOFE
  MYWUN DLFOG ELVKF ZYDSO HPHZQ YFABT ASDWL
  DLEDE 11400 021800>
```

Decryption

- set up main key in Printex (rotors and drum)
- determine manually real key from the false key
- set rotors of Printex in starting position
- decrypt
- clean up the cleartext (BISECT, XX, KW, ...)

How to decrypt without knowing the key?

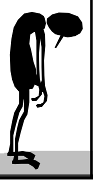


cryptanalysis

- determine main key based on known ciphertexts (and plaintexts)
- determine starting position of the rotor
- decrypt

Determine main key

- 26+25+23+21+19+17 = 131 pins (10^{40})
- 22 positions for lugs on 27 bars (10^{36}), but effectively only 27 bits
- exhaustive search:
  - transform every atom of the earth (10^{50}) to a supercomputer
  - trying all keys takes 3 billion years….



Determine main key (2)

- Need a better idea

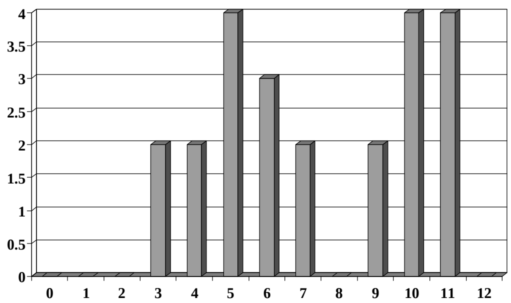
Ciphertext only attack

- attack needs about 2000-3000 ciphertexts + statistics on the plaintext
  - we had only ciphertexts of length < 370 available
  - the relation between the rotor positions was unknown (use of session keys)

Known plaintext attack [Morris 78]

- need 75-100 plaintext/ciphertext characters
- based on the fact that the number of lugs for the rotors is of the form:
  - 12-10-8-4-2-1
- idea: divide and conquer:
  - guess first the pins on the rotor with most active lugs
  - subtract the effect of this rotor
  - more complex: partial guess and forward/backward

Values of pins: historgram of average difference between plaintext and ciphertext for rotor 2 (23 pins)



Progress

- lugs and rotor pins recovered for message #14 (22 September)
- an “easy” test confirmed that a different key was used for earlier messages (23 Sept.)
- cryptanalysis attempts yielded only partial results (29 September)

- … what if the same key had been used anyway?

Why not try the key of #14?

- Just try exhaustively the 26x25x23x21x19x17~110 million starting positions of the rotors
- takes 5-15 minutes on a 1 GHz PC
  - identify correct solution from number of spaces (W) and BISECT (or BISSECT or BISOCT)

- extra trick: beginning position of rotor 6 is equal to that of rotor 5 (weakness in use)

It worked!!!

- October 1st: all plaintexts decrypted at 3:30am

- Why did displacements 1-3 and 7 occur?
  - many more errors than expected

Real key -> false key (Oct. 06)

- known plaintext pairs
  - IQ -> ME, CA -> FJ, LF -> NE, OP -> TJ
  - EU -> FP, SZ -> GJ, QT -> EK, CL -> IN
- find secret square (some keys wrong!)
- can now decrypt in a few microseconds

G	X	L	N	S
K	H	T	W	O
Q	D	E	F	A
M	V	I	C	R
Z	B	P	U	J

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

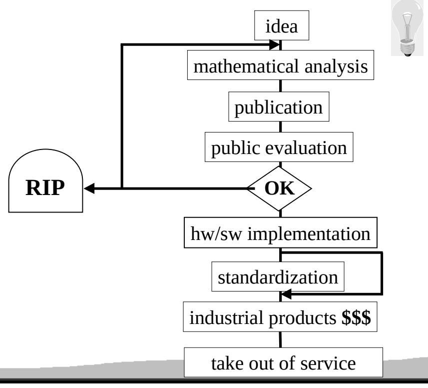
- Plaintext [=14 January 1961 11.00 h]
- **DOFGD VISWA WVISW JOSEP HWXXW TERTI OWMIS SIONW BOMBO KOWVO IRWTE LEXWC EWSUJ ETWAM BABEL GEWXX WJULE SWXXW BISEC TWTRE SECVX XWRWV WMWPR INTEX WXXWP RIMOW RIENW ENVOY EWRUS URWWX XWPOU VEZWR EGLER WXXWS ECUND OWREP RENDR EWDUR GENCE WPLAN WBRAZ ZAWWC**

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.**

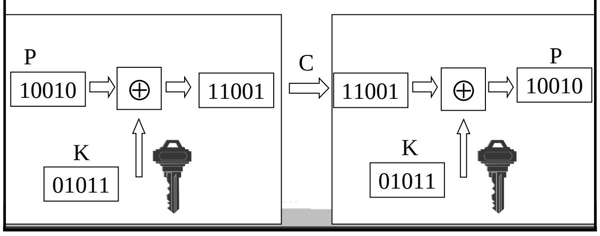
Resume urgently plan Brazzaville w.r.t. P. Lumumba

Life cycle of a cryptographic algorithm



Vernam scheme (1917) + Shannon (1948)

- key is random string, as long as the plaintext



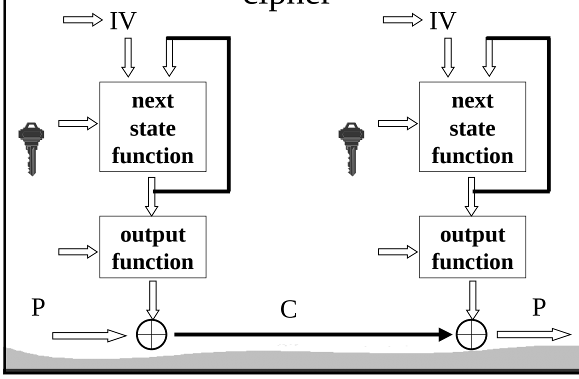
Vernam scheme

- perfect secrecy: ciphertext gives opponent no additional information on the plaintext or H(P|C)=H(P)
- impractical: key is as long as the plaintext
- but this is optimal: for perfect secrecy H(K) ≥ H(P)

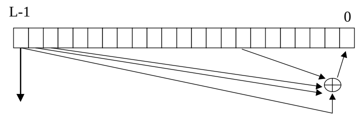
Three approaches in cryptography

- **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



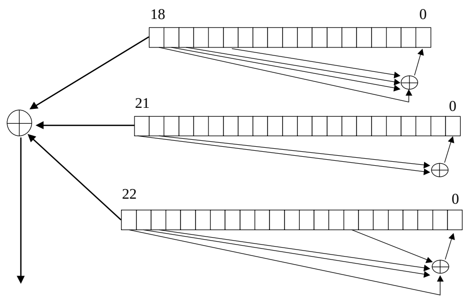
LFSR based stream cipher



+ good randomness properties
+ mathematical theory
+ compact in hardware

- too linear: easy to predict after 2L output bits

A5/1 stream cipher (GSM)

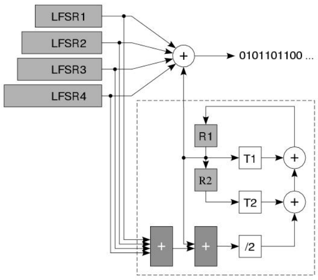


Clock control: registers agreeing with majority are clocked (2 or 3)

A5/1 stream cipher (GSM)

A5/1 attacks
- exhaustive key search: 2^{64} (or rather 2^{54})
- search 2 smallest registers: 2^{45} steps
- [BWS00] 2 seconds of plaintext: 1 minute on a PC
  - 2^{48} precomputation, 146 GB storage

Bluetooth stream cipher



- best known shortcut attack: 2^{70} rather than 2^{128}

Cryptanalysis of stream ciphers

- exhaustive key search (key of *k* bits)
  - 2^k encryptions, about *k* known plaintext bits
- time-memory trade-off (memory of *m* bits)
  - 2^t short output sequences
  - 2^{m-t} precomputation and memory
- linear complexity
- divide and conquer
- fast correlation attacks (decoding problem)

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]
```

t

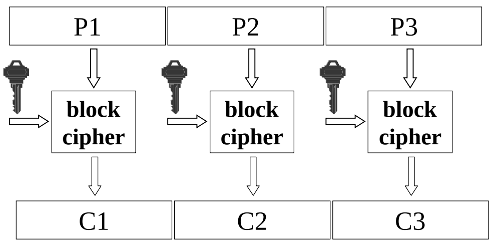
000	001	002		093	094	095		254	255
205	162	013	...	033	92	079	...	099	143

i
j

RC4: weaknesses

- often used with 40-bit key
  - US export restrictions until Q4/2000
- best known general shortcut attack: 2^{600}
- 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



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

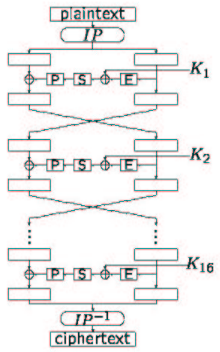
Cryptanalysis of block ciphers

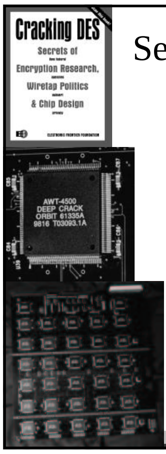
- exhaustive key search (key of k bits)
  - 2^k encryptions, k/n known plaintexts
- code book attack (block of n bits)
  - collect 2^n encryptions
  - with k/n chosen plaintexts : 2^k memory and time
- time-memory trade-off:
  - k/n chosen plaintexts
  - 2^k encryptions (precomputation)
  - on-line: $2^{2k/3}$ encryptions and $2^{2k/3}$ memory
- shortcut attacks: dc, lc,…..

DES properties

- design: IBM + NSA (1977)
- 64-bit block cipher with a 56-bit key
- 16 iterations of a relatively simple mapping
- optimized for mid 1970ies hardware
- FIPS 41: US government standard for sensitive but unclassified data
- worldwide de facto standard since early 80ies
- surrounded by controversy: key length

Data Encryption Standard





Security of DES (56-bit key)

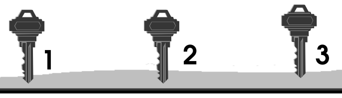
- PC: trying 1 DES key: 0.25 µs
- Trying all keys on 4000 PCs:
  1 month: $2^{22} \times 2^{16} \times 2^{5} \times 2^{12} = 2^{55}$
- M. Wiener's estimate (1993):
  1,000,000 $ machine: 35 minutes

EFF Deep Crack (July 1999)
250,000 $ machine: 50 hours…

Solution to DES key length

- Moore's "law": speed of computers doubles every 18 months
  - Conclusion: key lengths need to grow in time
- Use new algorithms with longer keys
- Or replace DES by triple-DES (168-bit key):

Clear text ⇨ **DES** ⇨ **DES^{-1}** ⇨ **DES** ⇨ %^C& @&^(



AES (Advanced Encryption Standard)

- Open competition launched by US government ('97)
- 21 contenders, 15 in first round, 5 finalists
- decision October 2, 2000
- 128-bit block cipher with long key (128/192/256 bits)
- five finalists:
  - MARS (IBM, US)
  - RC6 (RSA Inc, US)
  - Rijndael (KULeuven/PWI, BE)
  - Serpent (DK/IL/UK)
  - Twofish (Counterpane, US)

And the winner is...Rijndael

- Joan Daemen (pronounced Yo'-ahn Dah'-mun)
- Vincent Rijmen (pronounced Rye'-mun).

Joan Daemen

PhD in COSIC in 1995

now at Proton World International

Vincent Rijmen

PhD in COSIC in 1997

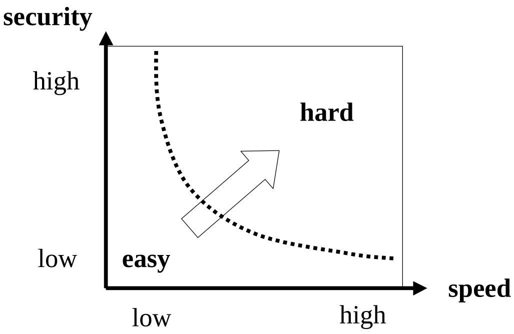
now at Cryptomathic

AES properties

- Rijndael: design by V. Rijmen (COSIC) and J. Daemen (Proton World, ex-COSIC)
- 128-bit block cipher with a 128/192/256-bit key
- 10/12/14 iterations of a relatively simple mapping
- optimized for software for 8/16/32/64-bit machines, also suitable for hardware

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

Design trade-off



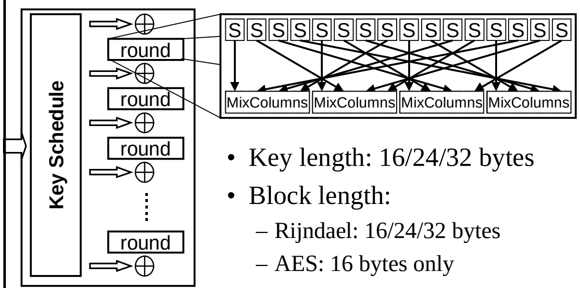
O’Connor versus Massey

- Luke O’Connor
  “most ciphers are secure after sufficiently many rounds”
- James L. Massey
  “most ciphers are too slow after sufficiently many rounds”

Rijndael

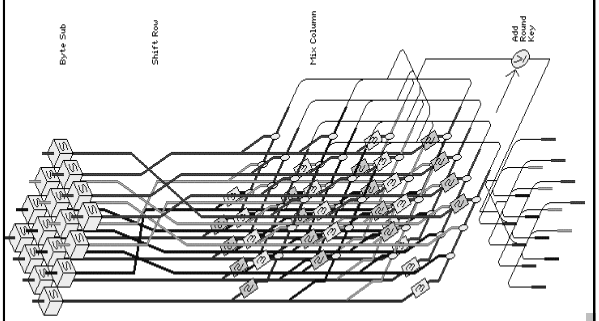
- history: Shark (1996) and Square (1997)
- security and efficiency through
  – simplicity
  – symmetry
  – modularity
- MDS codes for optimal diffusion
- efficient on many platforms, including smart cards
- easier to protect against side channel attacks

Rijndael: a key iterated block cipher



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

Rijndael: 1 round



AES: hardware performance

	Gb/s	MHz	kgates	Bits/ kgates
lookup	1.82	100	173	0.11
Local-1	0.12	100	5.7	0.21
Local-2	0.3	131	5.4	0.42
	2.6	224	21	0.55
	0.8	137	8.8	0.66
Global	7.5	32	256	0.92

AES/Rijndael: 1 round

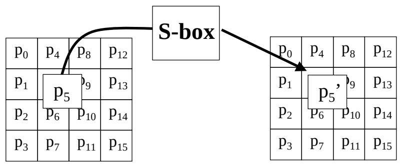
p_0	p_4	p_8	p_{12}
p_1	p_5	p_9	p_{13}
p_2	p_6	p_{10}	p_{14}
p_3	p_7	p_{11}	p_{15}

state: 16 bytes = 128 bits

1 round consists of 4 operations

SubBytes
ShiftRows
MixColumn
AddRoundKey

Rijndael round: SubBytes



256 byte table

mapping x^{-1} over $GF(2^8)$, plus some affine transformation over GF(2)

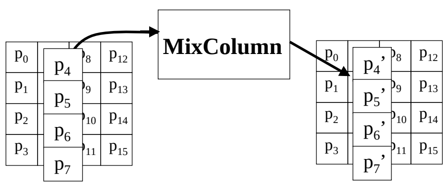
Rijndael round: ShiftRows

p_0	p_4	p_8	p_{12}
p_1	p_5	p_9	p_{13}
p_2	p_6	p_{10}	p_{14}
p_3	p_7	p_{11}	p_{15}

⇒

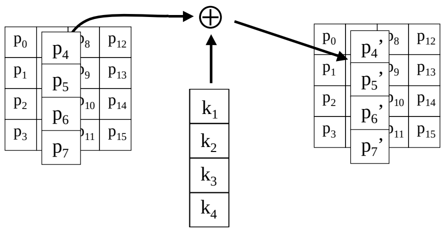
p_0	p_4	p_8	p_{12}
p_{13}	p_1	p_5	p_9
p_{10}	p_{14}	p_2	p_6
p_7	p_{11}	p_{15}	p_3

Rijndael round: MixColumn

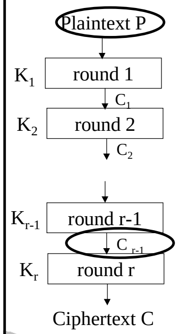


$$\begin{bmatrix} p_4' \\ p_5' \\ p_6' \\ p_7' \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot \begin{bmatrix} p_4 \\ p_5 \\ p_6 \\ p_7 \end{bmatrix}$$

Rijndael round: AddRoundKey



Linear cryptanalysis [Matsui93]

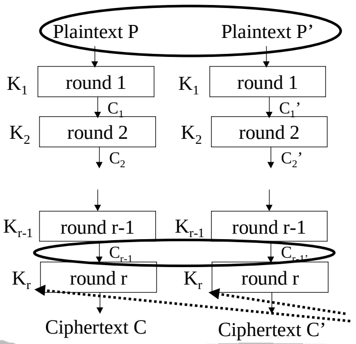


for a non-perfect cipher,
there exist values δ_0, δ_{r-1} s.t.
$P.\delta_0 \oplus C_{r-1}.\delta_{r-1} = 0$
with probability $p \neq 1/2$

or $P.\delta_0 \oplus \text{round}^{-1}(K_r,C).\delta_{r-1} = 0$
with probability $p \neq 1/2$

this leaks information on K_r

Differential cryptanalysis [Biham Shamir90]



choose the difference P, P'

try to predict the difference C_{r-1}, C_{r-1}'

this leaks information on K_r

Linear and differential cryptanalysis

- hard to find good linear or differential attacks
  - it is even harder to prove that it is impossible to find good linear or differential attacks
  - for some ciphers, this proof exists
- there exist many optimizations and generalizations
  - it is even harder to show that none of these work for a particular cipher
- analysis requires some heuristics
- DES: linear analysis needs 2^{43} known texts and differential analysis needs 2^{47} chosen texts

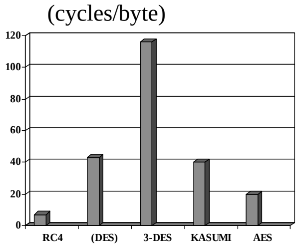
Rijndael design strategy

- simple and elegant
- no integer arithmetic
- wide trail strategy:
  - strong resistance against linear and differential attacks
  - over 4 rounds, sum of number of "active" input and output bytes equals 25
- diffusion based on (8,4) MDS code with minimum distance 5

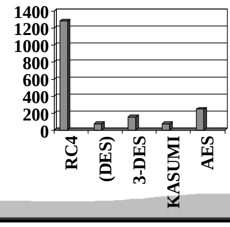
  [p1 p2 p3 p4 | p1' p2' p3' p4']

Performance reference data (Pentium)

encryption speed (cycles/byte)



key setup (cycles)



Recent "attacks" on Rijndael

- affine equivalence between bits of S-boxes
- algebraic structure in the S-boxes leads to simple quadratic equations
- simple overall structure leads to embedding in larger block cipher BES

- more research is needed...

AES Status

- FIPS 197 published on 6 December 2001
- Revised FIPS on modes of operation
- Rijndael has more options than AES

- fast adoption in the market (early 2003)
  – 51 products are FIPS 197 validated
  – > 100 products in the market
  – standardization: ISO, IETF, …
- slower adoption in financial sector

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

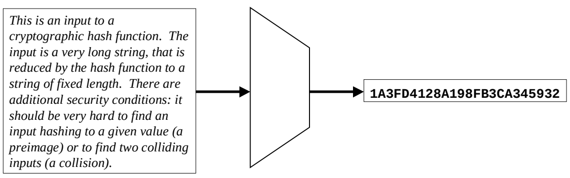
Data authentication: the problem

- 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: MDC

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

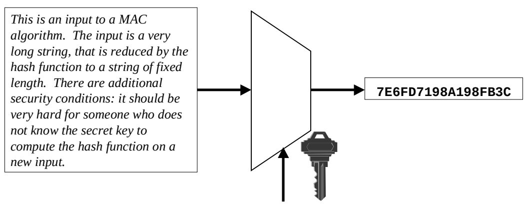
- (MD5)
- SHA-1
- SHA-256, -512
- RIPEMD-160



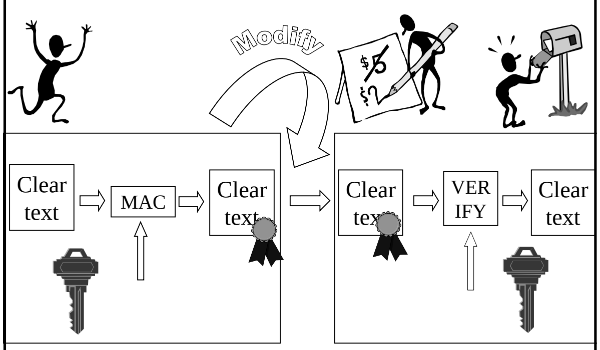
Data authentication: MAC

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

- CBC-MAC
- HMAC



MAC algorithms



One-way function: definition

- f (x) is a one-way function: $\{0,1\}^n \rightarrow \{0,1\}^n$
- easy to compute, but hard to invert
- f(x) has (ε,t) preimage security iff
  - choose x uniformly in $\{0,1\}^n$
  - let M be an adversary that on input f(x) needs time ≤ t and outputs M(f(x)) in $\{0,1\}^n$
  - Prob{f(M(f(x))) = f(x) < ε}, where the probability is taken over x and over all the random choices of M
- t/ε should be large

How to invert a one-way function?

- exhaustive search
  - $\Theta(e\ 2^n)$ steps, $\Theta(n)$ bits memory
  - recovering preimage for one out of s instances: $\Theta(e\ 2^n/s)$ steps, $\Theta(sn)$ bits memory
- tabulation
  - $\Theta(e\ 2^n)$ steps and $\Theta(n\ 2^n)$ memory (precomputation)
  - solve 1 instance: 1 table lookup
- time-memory trade-off:
  - $\Theta(e\ 2^n)$ steps and $\Theta(n\ 2^{2n/3})$ memory (precomputation)
  - solve 1 instance: $\Theta(e\ 2^{2n/3})$ steps
- problem: how to compare attacks with different processing time and memory?

How to find collisions for a function?

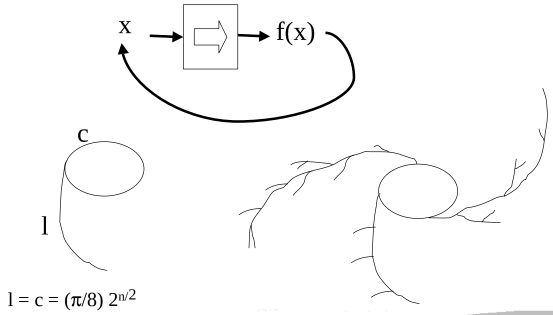
- collision = two different inputs x and x' to f for which f(x)=f(x')?
- requires $\Theta(e\ 2^{n/2})$ steps, $\Theta(n\ 2^{n/2})$ memory
- birthday paradox
  - given a set with S elements
  - choose r elements at random (with replacements) with r « S
  - the probability p that there are at least 2 equal elements is 1 - exp (- r(r-1)/2S)

How to find collisions for a function? (2)

- Numerical:
  - S large, r = √S, p = 0.39
  - S = 365, r = 23, p = 0.50
- surprising or paradoxical that finding collisions is much easier than inverting a function

Time-memory trade-off (1) [Hellman80]

- Consider the functional graph of f

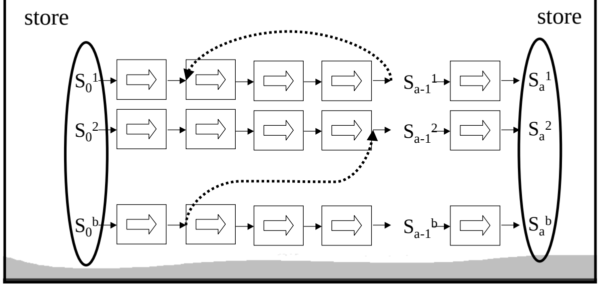


$l = c = (\pi/8)\ 2^{n/2}$

Time-memory trade-off (2)

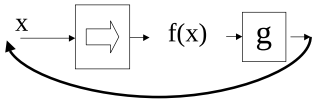
- Choose b different starting points and iterate for a steps

! problem: collisions: m t « 2^n



Time-memory trade-off (3)

Use c different variants of f by introducing the function g



- result:
  - precomputation: a . b . c
  - memory: b . c
  - on-line inverting of one value: a .c
- good choice: a = b = c = $2^{n/3}$
  - success probability 0.55

Time-memory trade-off (4)

- success probability = 1 - exp (- a D/ 2^n)

  with D the expected number of different points

  $D = (2^n / b) . G(a . b^2 / 2^n)$

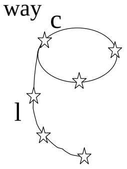
  $G(y) = \int_0^y (1-\exp(-x))/x \, dx$

  for 2^n »1, b » 1, ab « 2^n

- optimization: use distinguished points to reduce memory accesses

How to find collisions for a function - part 2 distinguished points [Pollard78][Quisquater89]

- define “dinstinguished” point, say a point that ends with d zero bits
- start from a distinguished point d and iterate f
- store the distinguished points along the way



if you find a collision in the distinguished points, “trace back” from the distinguished points before the collision

$\Theta(e\, 2^{n/2} + e\, 2^{d+1})$ steps
$\Theta(n\, 2^{n/2-d})$ memory

$l = c = (\pi/8)\, 2^{n/2}$

Time-memory trade-off (5)
with distinguished points

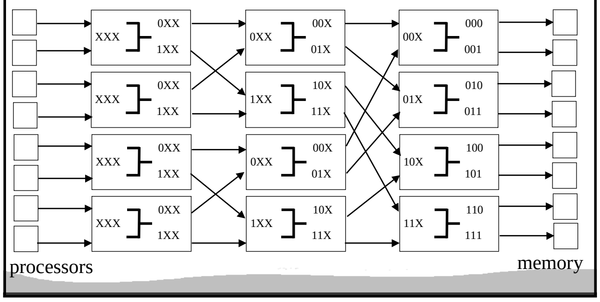
- precomputation: start chains in distinguished points until a new distinguished point is reached (or a certain bound is exceeded)
- recovery: iterate until a distinguished point is reached
- advantage: reduced memory access - only required to store and look up distinguished points; this makes the attack much cheaper

Full cost measure [Wiener02]

- full cost of hardware = product of number of components with the duration of their use
- motivation: hardware = ALUs, memory chips, wires, switching elements
- question: if an algorithm requires $\Theta(2^n)$ steps and $\Theta(2^n)$ memory, what is the full cost: $\Theta(2^{2n})$ or $\Theta(2^n)$ or $\Theta(2^{3n/2})$?
- answer: it depends on inherent parallelism and memory access rate
  - for 1 processor with $\Theta(2^n)$ steps and 1 big memory of size $\Theta(2^n)$, full cost is $\Theta(2^{2n})$
  - for $\Theta(2^{n/2})$ processors with $\Theta(2^{n/2})$ steps and 1 big memory of size $\Theta(2^n)$, full cost is $\Theta(2^{3n/2})$

Full cost of connecting many processors to a large memory

- easy case: wiring cost to connect q processors to q blocks of memory equals $\Theta(q^{3/2})$



Full cost of connecting many processors to a large memory (2)

- cost of wires
  - for $q = 8 = 2^3$: $4 + 8 = 12$
  - for $q = 16 = 2^4$: $8 + 16 + 32 = 56$
  - for $q = 2^t$: $2^{t-1}(2^{t-1}-1) = \Theta(q^2)$
- more than half of the cost is between 2 last stages: q wires of length q/2
- 2D packing reduces length of wires to $\Theta(q^{1/2})$
- total volume is $\Theta(q^{3/2})$ (need in fact 3D packing)
- this can also shown to be optimal

Full cost of connecting many processors to a large memory (3): general case

- r = memory access rate per processor (# bits requested every unit of time)
- p = number of processors
- m = number of memory elements

- The total number of components to allow each of p processors uniformly random access to m memory elements at a memory access rate of r equals $\Theta(p + m + (pr)^{3/2})$

Full cost of connecting many processors to a large memory (4): general case

- For an algorithm where p processors access a memory of size m at rate r, and the total number of steps is T, the full cost is equal to $F=\Theta((T/p)(p + m + (pr)^{3/2}))$

- $F = \Theta(T)$ iff $p = \Omega(m)$ and $r = O(p^{-1/3})$
  - processors may access small individual memory at high rate
- If r is high and m is independent of p, then $F=\Theta(T\ r\ m^{1/3})$, with $p=\Theta(m^{2/3}/r)$
- Be careful in practice with the constants!

Full cost of inverting a one-way function (1)

- exhaustive search $F = \Theta(e\ 2^n)$

- tabulation: $F = \Theta(e\ n\ 2^{2n})$
- but if we are recovering $s = \Theta(2^n)$ preimages using tabulation
- $r = \Theta(n/e)$ (high); $T = \Theta(e\ 2^n)$;
- $F = \Theta(T\ r\ m^{1/3}) = \Theta((n\ 2^n)^{4/3})$ with $p = \Theta(e\ 2^{2n/3}/n^{1/3})$
- Full cost per key: $\Theta(2^{n/3}\ n^{4/3})$

Full cost of inverting a one-way function (2)

- time-memory trade-off with c=a or $b = 2^n/a^2$
- precomputation
  - $m = \Theta(abn) = \Theta(n\ 2^n/a)$
  - $r = \Theta(n/(ae))$ $T = \Theta(e\ 2^n)$
  - $F = \Theta(T/p).\ \Theta(p + m + (pr)^{3/2})$ with $p_{max} = \Theta(2^n/a)$
  - $F = \Theta(ne\ 2^n)$ with $a = \Omega(n^{1/4}\ 2^{n/4}/e^{3/4})$
- key recovery
  - memory $m = \Theta(abn) = \Theta(n\ 2^n/a)$
  - $r = \Theta(n/e)$ $T = \Theta(e\ a^2)$
  - F per key $= \Theta(2^{n/3}n^{4/3}\ a^{5/3})$, $p = \Omega(e\ 2^{2n/3}/(n^{1/3}\ a^{2/3}))$

Full cost of inverting a one-way function (3)

- precomputation and key recovery each have a full cost of $F = \Theta(ne\ 2^n)$
- but need to work on many problems: $p \leq \Theta(a)$
- precomputation does NOT reduce the full cost to find a single key
- total number of keys that can be found for the cost of exhaustive search is $s = \Theta(2^{n/4} e^{9/4} / n^{3/4})$; the full cost per key decreases from $\Theta(e 2^n)$ to $\Theta(e\ 2^{3n/4})$
- variant with distinguished points: $s = \Theta(2^{3n/5} e^{6/5} / n^{2/5})$ and full cost per key decreases to $\Theta(e\ 2^{2n/5})$
- table lookup: $s = \Theta(2^n)$ and cost per key $\Theta(e\ 2^{n/3})$

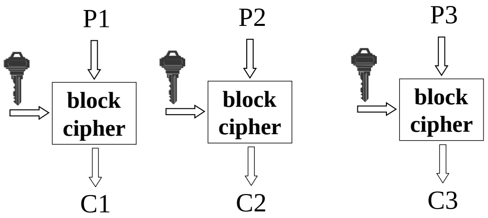
Full cost of collision search

- $T = \Theta(e\ 2^{n/2})$, $m = \Theta(n\ 2^{n/2})$, $r = \Theta(n/e)$ (high)
- $F = \Theta(2^{2n/3}\ n^{4/3})$ with $p = \Theta(e\ 2^{n/3} / n^{1/3})$
- Pollard rho with distinguished points $F = \Theta(e\ n\ 2^{n/2})$
- cost drops further for multiple collisions

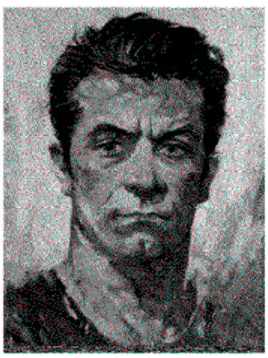
Full cost (summary)

- full cost of an algorithm that requires $\Theta(2^n)$ steps and $\Theta(2^n)$ memory
  - if no parallelism possible: $\Theta(2^{2n})$
  - if arbitrary parallelism: between $\Theta(2^n)$ and $\Theta(2^{4n/3})$ depending on the memory access rate
- For an algorithm where p processors access a memory of size m at rate r, and the total number of steps is T, the full cost is equal to $F=\Theta((T/p)(p + m + (pr)^{3/2}))$
- In practice, constants are important!
- M. Wiener, The full cost of cryptanalytic attacks, J. Cryptology, to appear

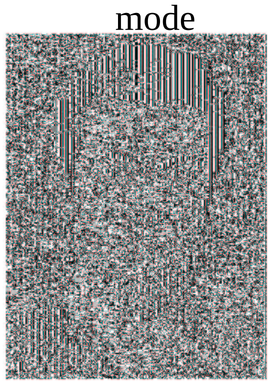
How NOT to use a block cipher: ECB mode



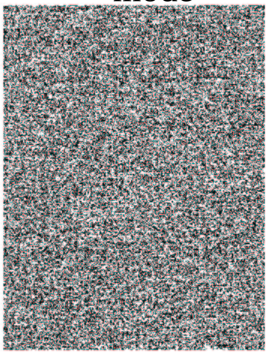
An example plaintext



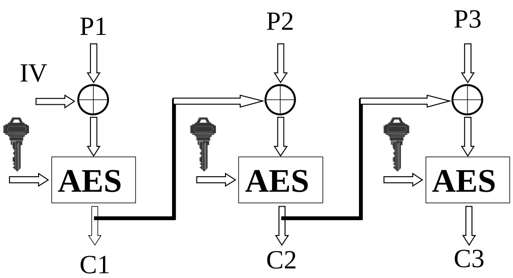
Encrypted with AES in ECB mode



Encrypted with AES in CBC mode

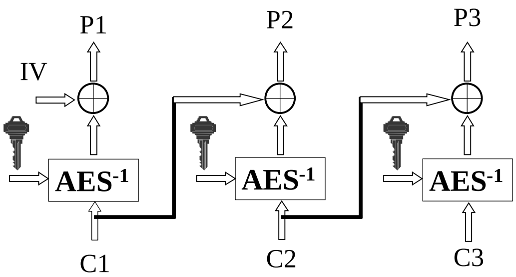


How to use a block cipher: CBC mode



need random IV

CBC mode decryption



Secure encryption

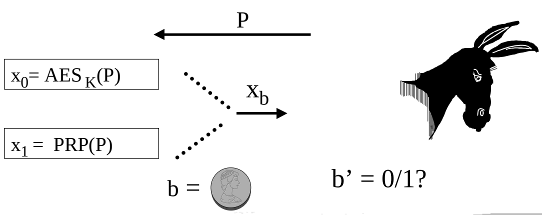
- What is a secure block cipher anyway?
- What is secure encryption anyway?

- Definition of security
  - security assumption
  - security goal
  - capability of opponent

Security assumption: the block cipher is a pseudo-random permutation

- It is hard to distinguish a block cipher from a random permutation
- Advantage of a distinguisher

  $Adv_{AES/PRP} = Pr[b'=1|b=1] – Pr[b'=1|b=0]$



Security goal: "encryption"

- semantic security: adversary with limited computing power cannot gain any extra information on the plaintext by observing the ciphertext
- indistinguishability (real or random) [IND-ROR]: adversary with limited computing power cannot distinguish the encryption of a plaintext P from a random string of the same length

- IND-ROR ⇒ semantic security

Indistinguishability: IND-ROR

- Advantage of a distinguisher
  $Adv_{ENC} = Pr[b'=1|b=1] – Pr[b'=1|b=0]$

P

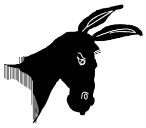
$x_0 = C = E_{CBC\ K}(P)$

x_b

x_1 = random string of size |C|

b = 

b' = 0/1?



Capability of opponent

- ciphertext only
- known plaintext
- chosen plaintext
- adaptive chosen plaintext
- adaptive chosen ciphertext

[Bellare+97] CBC is IND-ROR secure against chosen plaintext attack

- consider the block cipher AES with a block length of *n* bits; denote the advantage to distinguish it from a pseudo-random permutation with Adv_{AES}
- consider an adversary who can ask q **chosen plaintext** queries to a CBC encryption

$Adv_{ENC/CBC} \leq 2\, Adv_{AES} + (q^2/2)2^{-n} + (q^2-q)2^{-n}$

reduction is tight as long as $q^2/2 \ll 2^n$ or $q \ll 2^{n/2}$

[Bellare+97] CBC security

- matching lower bound:
  - collision $C_i = C_j$ implies $C_{i-1} \oplus P_i = C_{j-1} \oplus P_j$
  - collision expected after $q = 2^{n/2}$ blocks
- CBC is very easy to distinguish with **chosen ciphertext** attack:
  - decrypting C || C || C yields P' || P || P

The birthday paradox

- Given a set with S elements
- Choose q elements at random (with replacements) with q « S
- The probability p that there are at least 2 equal elements is 1 - exp (- q(q-1)/2S)
- S large, $q = \sqrt{S}$, p = 0.39
- S = 365, q = 23, p = 0.50

Some books on cryptology

- B. Schneier, Applied Cryptography, Wiley, 1996. Widely popular and very accessible – make sure you get the errata.
- D. Stinson, Cryptography: Theory and Practice, CRC Press, 1995. Solid introduction, but only for the mathematically inclined.
  - 2nd edition, part 1 available in 2002.
- 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. All chapters can be downloaded for free at http://www.cacr.math.uwaterloo.ca/hac

Books on network security and more

- W. Stallings, *Network and Internetwork Security: Priniples and Practice*, Prentice Hall, 1998. 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.
- W. Diffie, S. Landau, *Privacy on the line. The politics of wiretapping and encryption*, MIT Press, 1998. 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
- links: Ron Rivest, David Wagner, Counterpane www.counterpane.com/hotlist.html
- Digicrime (www.digicrime.org) - not serious but informative and entertaining