COSIC

KU LEUVEN

Symmetric algorithms for confidentiality

Prof. Bart Preneel
COSIC – KU Leuven - Belgium
Firstname.Lastname(at)esat.kuleuven.be
http://homes.esat.kuleuven.be/~preneel

October 2021

1

1

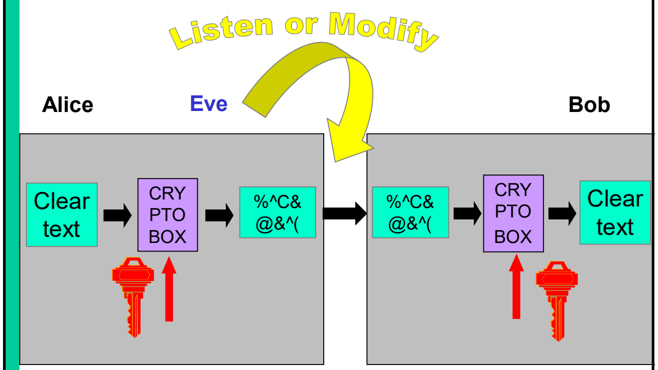
Learning goals

- The difference between block ciphers and stream ciphers
- How to choose the parameters for symmetric ciphers
- How some widely used stream ciphers and block ciphers work
- Understand the modes of operation of a block cipher and their strengths and weaknesses
- How to analyze a new mode of operation

2

2

Cryptology: basic principles



3

3

Symmetric cryptology: stream ciphers and block ciphers

- stream ciphers: A5/1, RC4
- block ciphers: DES, 2-DES, 3-DES and AES
- modes of operation
- security of the modes of operation

4

4

Stream ciphers

Block ciphers: well-established standards (DES and AES)

↕

Stream ciphers: some standards but no clear winners

5

5

Stream ciphers and block ciphers: history

- 1882 one-time pad
- 1930s Hagelin
- 1930s Enigma
- 1960s LFSRs

- 1987 RC4
- 1988 A5/1, A5/2
- 1992 SEAL
- 199? CRYPTO-1
- 1998 GEA-1, GEA-2, Panama
- 1999 E0, SNOW
- 2001 MUGI
- 2004 Trivium
- 2006 SNOW 3G
- 2008 Chacha20
- 2010 ZUC
- 2019 SNOW V (5G)

2004–2008 **estream**

Block ciphers

- 1977 **DES**
- 1978 Triple-DES
- 1987 FEAL
- 1990 Khufu/Khafre/IDEA
- 1999 KASUMI
- 2000 **AES**
- 2007 Present

6

6

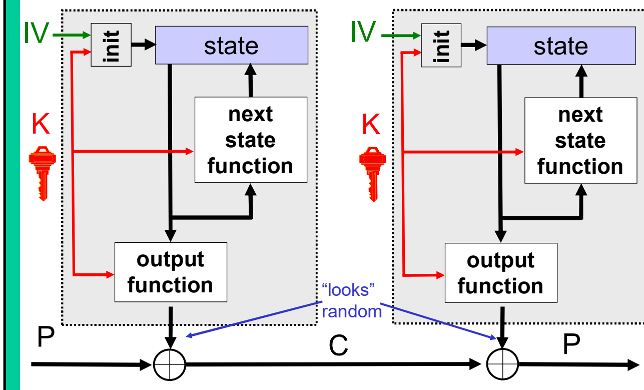
Stream ciphers: definitions

- Main characteristics:
  - internal state: finite state machine
  - operate on small words (bits, bytes, 32-bit words)
- Focus here only on Synchronous Stream Cipher (SSC)
  - Self-Synchronizing Stream Cipher (SSSC) are legacy solution

7

7

Synchronous Stream Cipher (SSC)



8

SSC: Specific properties

- Recipient needs to be synchronized with sender
- No error-propagation
  - excellent for wireless communications
- Key stream independent of data
  - key stream can be precomputed
  - particular model for cryptanalysis: attacker is not able to influence the state

9

9

SSC: Avoid repeating key stream

- For a fixed key K and initial value IV, the stream cipher output is a deterministic function of the state.
- A repetition of the state (for a given K, IV) leads to a repetition of the key stream and plaintext recovery (transmission in depth or the "Venona problem" of one time pad with reused key)
  - hence state needs to be large and next state function needs to guarantee a long period
  - IV can be used to generate a different key stream for every packet in a packet-oriented communication setting
  - old stream ciphers defined without IV are problematic in such a setting (example: RC4)

10

10

Advantages of stream ciphers

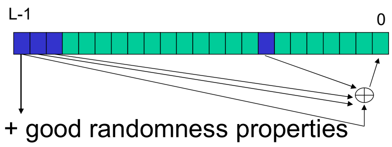
- Software:
  - up to five times faster than AES
  - useful in high throughput applications
- Hardware
  - use less area than AES
    - < 2 kGate compared to > 3 kGate for AES
  - higher throughput/area possible
  - may be useful in restricted environments: wireless video, medical devices

11

11

Types of SSC

- LFSR based stream ciphers



+ good randomness properties
+ mathematical theory
+ maximum period (2^L-1)
+ compact in hardware
- too linear: easy to predict

12

12

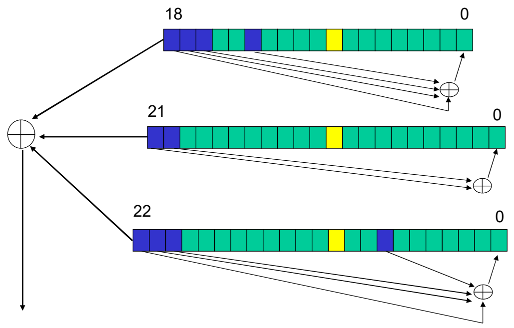
Types of SSC (2)

- Destroy linearity of LFSRs
  - with irregular clocking
    - e.g. A5/1 (GSM)
  - with additional memory that is updated with a nonlinear function of the LFSR state bits
    - e.g. E0 (Bluetooth), SNOW 3G, ZUC

13

13

A5/1 stream cipher (GSM)



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

14

14

A5/1 stream cipher (GSM)

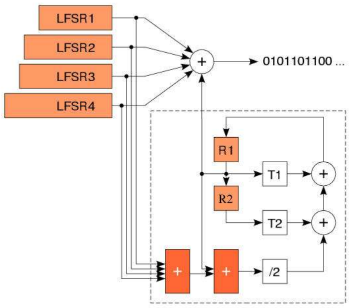
A5/1 attacks

- exhaustive key search: 2^{64} (or rather 2^{54})
- search 2 smallest registers: 2^{41} values – a few steps to verify a guess
- [BB05]: 10 minutes on a PC
  - 3-4 minutes of ciphertext only
- [Nohl-Paget'09]: "rainbow tables"
  - seconds with a few frames of ciphertext only

15

15

Bluetooth stream cipher (E0)



- brute force: 2^{128} steps
- [Lu+05] 24 known bits of 2^{24} frames, 2^{38} computations, 2^{33} memory

16

16

Alternatives to LFSR-based stream ciphers

- Designs based on random shuffles
  - RC4, HC-128
- Stream ciphers based on block ciphers (cf. modes of operation)
  - less efficient

17

17

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
```

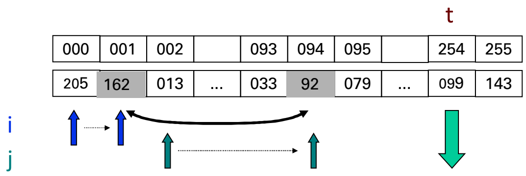
18

18

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]



19

19

RC4: weaknesses

- was often used with 40-bit key
  - US export restrictions until Q4/2000
- best known general shortcut attack: 2^{241} [Maximov-Khovratovich'09]
- weak keys and key setup (shuffle theory)
- large statistical deviations
  - bias of output bytes (sometimes very large)
  - can recover 220 out of 256 bytes of plaintexts after sending the same message 1 billion times (WPA/TLS)
- problem with resynchronization modes (WEP)

20

20

Cryptanalysis of RC4 in TLS and WPA

http://www.isg.rhul.ac.uk/tls/
[AlFardan-Bernstein-Paterson-Poettering-Schuldt'13]

- recover 220 out of 256 bytes of plaintexts after sending the same message 1 billion times
- some bytes can be recovered after "only" 16 million transmissions
- extensions can find more bytes

Related: Full Plaintext Recovery Attack on Broadcast RC4 [Isobe-Ohigashi-Watanabe-Morii '13]

21

Cryptanalysis of stream ciphers

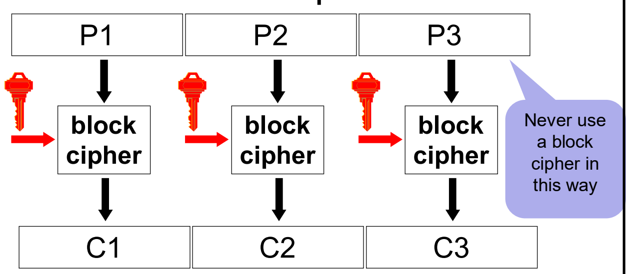
- exhaustive key search (key of k bits)
  - 2^k encryptions, about k known plaintext bits
- time-memory trade-off (memory of m bits)
  - 2^{m-t} precomputation and memory
  - can recover state (and in most cases the key) after observing 2^t short output sequences

(based on the birthday paradox – see later)

22

22

Block cipher



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

23

23

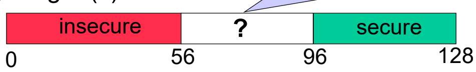
Block ciphers

Block length (n) — cf. discussion of CBC mode

- 64 bits: too small for many applications today
- 128 or 256 bits is ok

56 bits: 0.03 second with M$ 1
80 bits: 6 days with M$ 1
128 bits: 0.5 billion years with B$ 10

Key length (k)



DES: n=64; k=56
3-DES: n=64; k=112 or 168
IDEA: n=64; k=128
KASUMI: n=64; k=128 in 3G, 64 in 2G)
AES: n=128; k =128 or 192 or 256

24

24

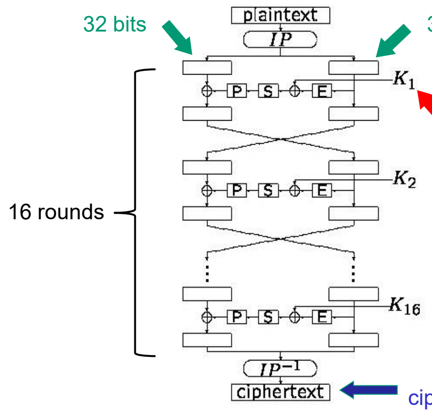
Data Encryption Standard (1977)

- block length n = 64 bits, key length k = 56 bits
- 16 iterations of a relatively simple mapping
- efficient in mid 1970s hardware
- FIPS: US government standard for sensitive but unclassified data
- worldwide de facto standard 1980-2004
- surrounded by controversy
- after almost 40 years, no practical shortcut attack
  - best one requires 2^{42} known plaintexts

25

25

Data Encryption Standard (DES)

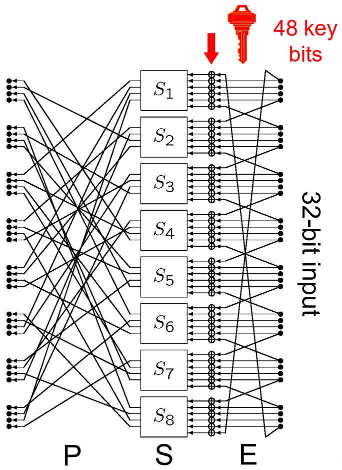


26

26

The DES round function

maps 32 to 32 bits

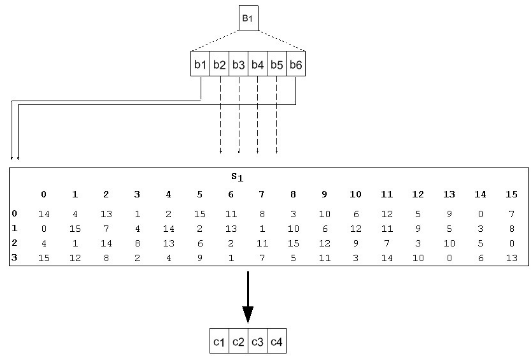


27

27

DES S-box 1

S-boxes are the only nonlinear part of the DES

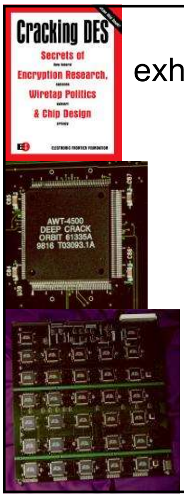


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Source: http://www.gungfu.de

28

28



DES is weak against exhaustive key search (56-bit key)

- PC: trying 1 DES key: 1 ns thus finding a key on 64 PCs requires 8 days: $2^{30} \times 2^{16} \times 2^{6} \times 2^{3} = 2^{55}$
- M. Wiener's ASIC design (1993): 1,500,000 \$ machine: 3.5 hours (in 2023: 0.015 second)
- EFF Deep Crack ('98): FPGA
  - 250,000 \$ machine: 50 h
- COPACABANA FPGA ('07):
  - 10,000 \$: 6.4 days

29

29

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.

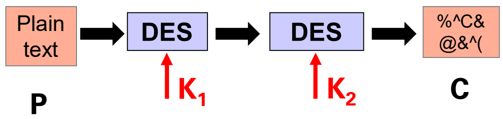
ACTION: Notice; request for comments.

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).

30

Double DES (or 2-DES)

- key length: 112 bits
- exhaustive search: 2^{112} encryptions
  - with US $10 billion search machine 60,000 years

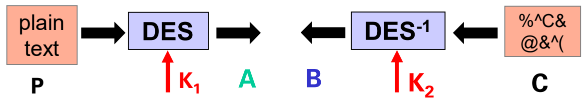


- Meet-in-the-middle attack (next slide): 2^{57} encryptions but unrealistic storage requirement: 2^{60} bytes
- Improved meet-in-the-middle attack: 2^{72} encryptions but modest storage cost [Wiener-van Oorschot'94]

31

31

Shortcut Attack on 2-DES (meet-in-the middle) (1)



- step 1: collect 2 pairs (P,C), (P'C')
- step 2: compute and store **A** = E_{K1}(P) for all 2^{56} values of K_1 in a huge table of size 2^{56} x 15 bytes or 2^{60} bytes = 960 Petabyte
- step 3: compute **B** = D_{K2}(C) for all 2^{56} values of K_2 and check for each **B** whether it is in the table (a match)
- step 4: if (K_1,K_2) generates a match, test it with a second pair (P',C') just to be sure

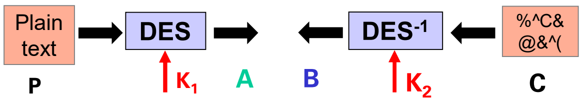
Value for K_1	E_{K1}(P)
000…000	A_1
000…001	A_2
000…011	A_3
…	…
111…111	$A_{2^{56}}$

Cost of step 2 + 3: 2^{57} encryptions and ≅ 2^{60} bytes storage

32

32

Shortcut Attack on 2-DES (meet-in-the middle) (2)



- if (**K1**,**K2**) is the correct key, both (P,C) and (P',C') will give a match
- if (**K1**,**K2**) is incorrect, will there be a match in step 3?
- table contains 2^{56} 64-bit values, so 1 64-bit value in 256 will appear

Value for K_1	E_{K1}(P)
000…000	A_1
000…001	A_2
000…011	A_3
…	…
111…111	$A_{2^{56}}$

- step 3 generates 2^{56} 64-bit values **B**, so we expect $2^{56}/256 = 2^{48}$ matches that go to step 4
- the probability that a wrong (K1,K2) survives step 4 is 2^{-64}
- with high probability only the correct key survives step 4
- total cost of step 4: $2.(1 + 2^{48})$ encryptions ≪ step 2 + step 3

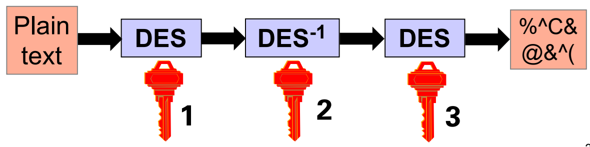
total cost: 2^{57} encryptions and ≅ 2^{60} bytes storage

33

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

(May 2004)

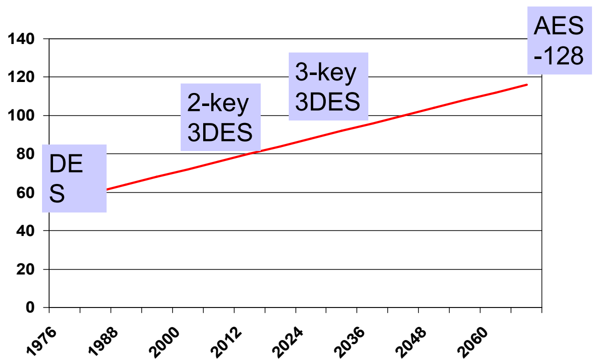
- single DES abandoned (56 bit)
- double DES not good enough (72-bit security level)
- 2-key triple DES (K1=K3): until 2009 (80-bit security level)
- 3-key triple DES: until 2030 (100-bit security level)



34

34

Symmetric Key Lengths and Moore's "law"



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

35

35

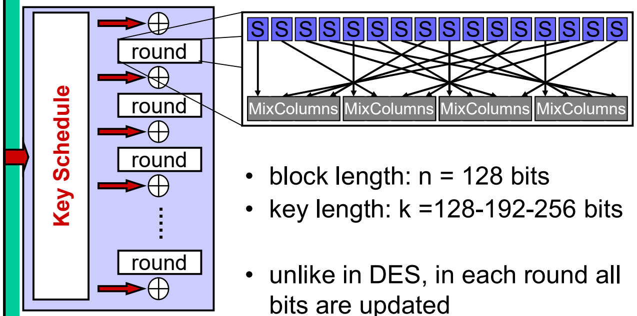
AES (Advanced Encryption Standard)

- open competition launched by US government (Sept. '97) to replace DES
- requirements
  - block length n = 128
  - key length k = 128-192-256
  - as strong as triple-DES, but more efficient
  - winner should be royalty-free
- 22 contenders including IBM, RSA, Deutsche Telekom

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

36

AES: Rijndael



- block length: n = 128 bits
- key length: k =128-192-256 bits
- unlike in DES, in each round all bits are updated

37

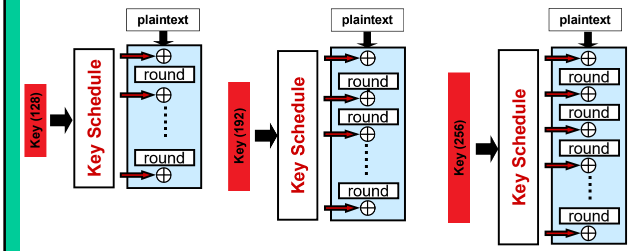
37

AES variants (2001)

- AES-128
- 10 rounds
- sensitive/classified

- AES-192
- 12 rounds
- top secret

- AES-256
- 14 rounds
- top secret



Light weight key schedule, in particular for the 256-bit version

38

38

AES (2001)

- FIPS 197 published Dec. 2001 after 4-year open competition
  - other standards: ISO, IETF, IEEE 802.11,…
- fast adoption in the market
  - except for financial sector
  - NIST validation list: > 6100 implementations
    - http://csrc.nist.gov/groups/STM/cavp/documents/aes/aesval.html
- [NSA 2003:] AES-128 also for secret information and AES-192/-256 for top secret information!
- [NSA 2015:] preference for AES-256 (quantum computers)
- security:
  - algebraic attacks of [Courtois+02] not effective
  - side channel attacks: cache attacks on unprotected implementations

[Shamir '07] AES may well be the last block cipher

39

39

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

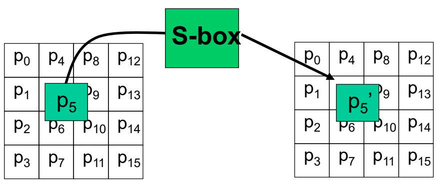
1 round consists of 4 operations

SubBytes
ShiftRows
MixColumns
AddRoundKey

40

40

Rijndael round: SubBytes



256 byte table (8-bit to 8-bit)

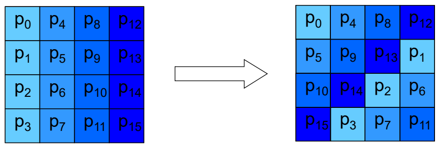
mapping x^{-1} over GF(2^8), plus some affine transformation over GF(2)

only nonlinear part of AES

41

41

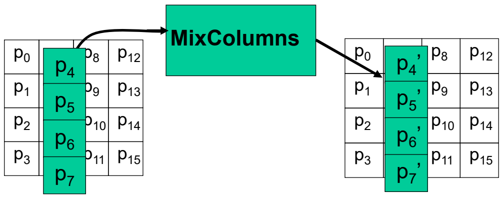
Rijndael round: ShiftRows



42

42

Rijndael round: MixColumns

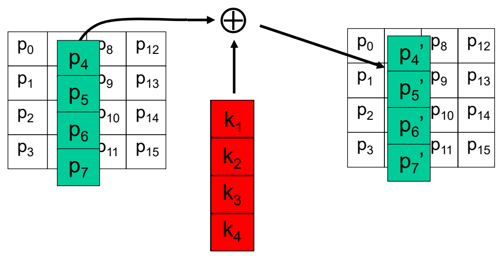


$$\begin{pmatrix} p_4' \\ p_5' \\ p_6' \\ p_7' \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} p_4 \\ p_5 \\ p_6 \\ p_7 \end{pmatrix}$$

43

43

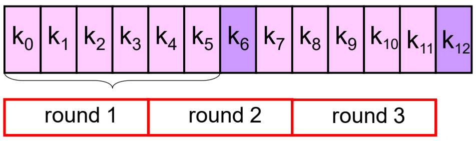
Rijndael round: AddRoundKey



44

44

Rijndael key schedule



- $k_{6n} = k_{6n-6} + f(k_{6n-1})$
- $k_i \; = k_{i-6} + k_{i-1}$

k_i = 32 bits or 4 bytes

45

45

AES implementations: efficient/compact

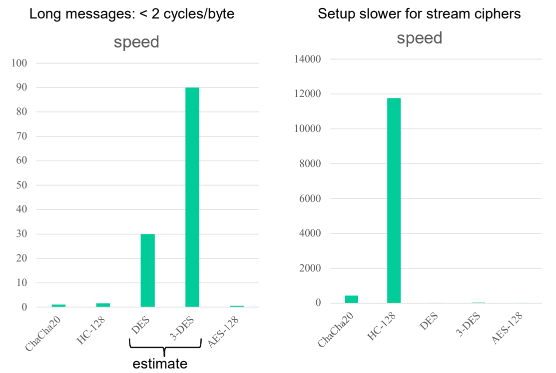
- HW: 43 Gbit/s in 130 nm CMOS ['05]
- Intel/AMD/ARM: new AES instruction in high end processors after 2010: 0.64-1.27 cycles/byte ['09-'10]
- SW: 7.6 cycles/byte on Core 2 bitsliced [Käsper-Schwabe'09]
- HW: most compact: about 3000 gates

46

46

Software performance on a 4-core 3 GHz processor

(KabyLake (906e9); 2017 Intel Xeon E3-1220 v6)
source: https://bench.cr.yp.to/index.html



47

47

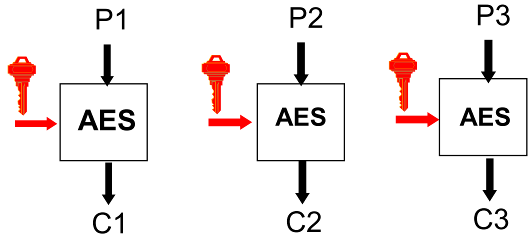
Cryptanalysis of block ciphers

- exhaustive key search (key of k bits)
  - 2^k encryptions, k/n known plaintexts
- tabulation attack (block of n bits):
  - collect large number of plaintext/ciphertext pairs
- code book attack (block of n bits)
  - encrypt 1 plaintext under all 2^k keys and store the result in a very large table
  - one key can now be found in constant time (table look-up)
- time-memory trade-off
  - k/n chosen plaintexts
  - 2^k encryptions (precomputation) and $2^{2k/3}$ encryptions memory
  - on-line: $2^{2k/3}$ encryptions
- differential cryptanalysis
- linear cryptanalysis
- ….

48

48

How NOT to use a block cipher: ECB mode (Electronic Code Book)



49

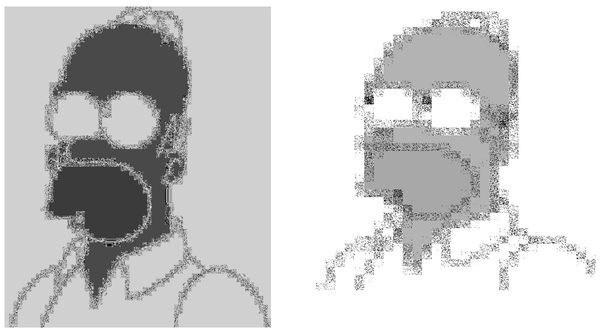
49

An example plaintext

50

50

Encrypted with substitution (left) and transposition cipher (right)

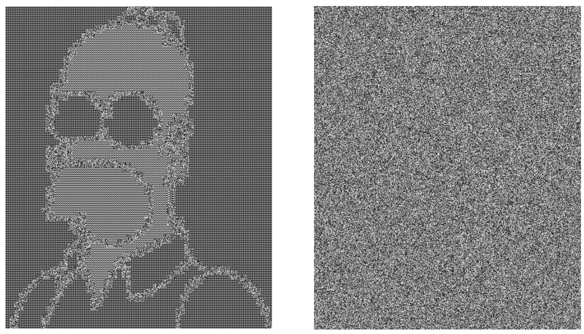


51

51

Encrypted with AES in ECB (left) and CBC mode (right)

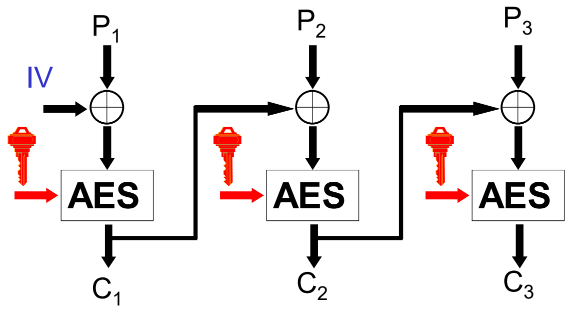
ECB CBC



52

How to use a block cipher: CBC mode

Cipher Block Chaining $C_i = E_K(P_i \oplus C_{i-1})$



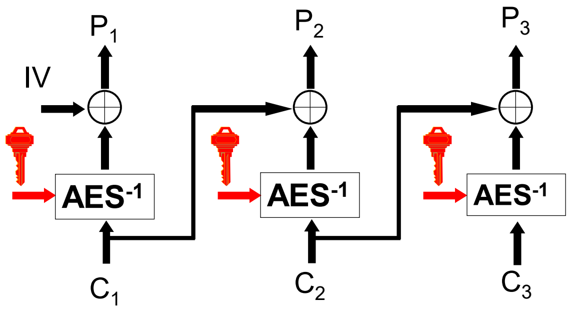
need random and secret IV

53

53

CBC mode decryption

$P_i = D_K(C_i) \oplus C_{i-1}$



54

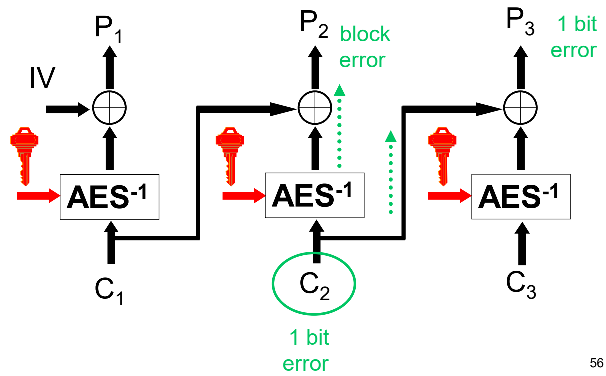
54

CBC properties

- propagation from left to right
- random and secret IV: hides repetitions in the beginning of the plaintext
- encryption only from left to right, but decryption with random access
- need integral number of blocks (n bits)
- decryption with limited error propagation

CBC mode decryption

$P_i = D_K(C_i) \oplus C_{i-1}$



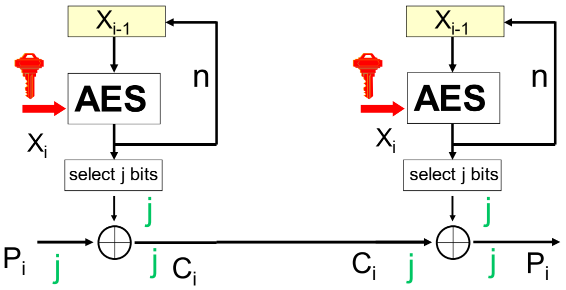
CBC is “secure” against chosen plaintext attack (informal)

[Bellare et al. 97]

If AES is a “secure” n-bit block cipher, then AES in CBC mode with a random and secret IV is an encryption algorithm “secure” against chosen plaintext attacks provided that you encrypt at most r blocks with $r \ll 2^{n/2}$

Output Feedback Mode (OFB)

$X_i = E_K(X_{i-1})$, $C_i = P_i \oplus$ leftmost j bits of (X_i)



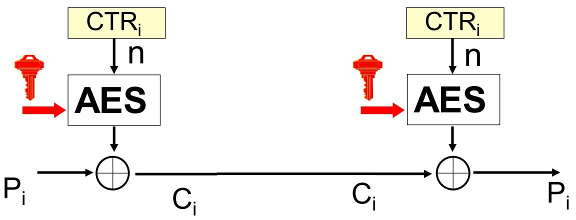
state initialized with random IV, or X_0 = IV, $j \leq n$

OFB: properties

- no linking between subsequent blocks
- different IV necessary; otherwise insecure
- uses only encryption
- if j <n: more effort per bit – mostly j = n
- key stream independent of plaintext: can be pre-computed
- no error propagation: errors are only copied

CounTer Mode (CTR)

$C_i = P_i \oplus E_K(CTR_i)$, CTR_i ++



state initialized with random IV, or CTR_0 = IV, typically 32 rightmost bits of IV equal to 0

CTR: properties

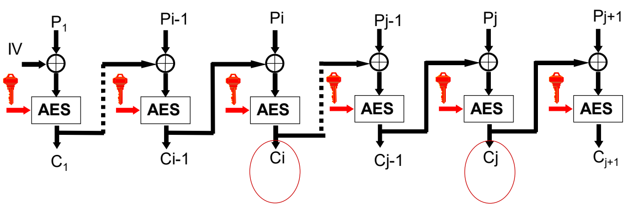
- similar properties as OFB
- but random access on decryption
- but better suited for hardware:
  - **parellelism**: one can process multiple counter values at the same time
  - **pipelining:** no need to know the ciphertext block corresponding to the current plaintext block to start processing the next plaintext block
- risk: what if counters are (accidentally) reset to same value? (“Venona problem”)

Overview modes: when to use

- ECB: never
- CBC: legacy – to be replaced by authenticated encryption because of chosen ciphertext attacks (see later)
- OFB, CTR: no error propagation (e.g. wireless); CTR if high speed necessary (hardware)
- (not discussed) CFB: if synchronization is important (but slow)

Limits of 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 r $=2^{n/2}$ blocks



The birthday paradox (1)

- 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 (a collision) is

  $\approx 1 - \exp(-r(r-1)/2S)$

  - S large, $r = \sqrt{S}$, p = 0.39
  - S = 365, r = 23, p = 0.50 (exact)

The birthday paradox (2) - proof

$$q = 1\text{-}p = 1 \, . \, \overbrace{((S-1)/S) \, . \, ((S-2)/S) \, ... \, ((S-(r-1))/S)}^{\text{r terms}}$$

or $q = \Pi_{k=1}^{r-1} (S-k/S)$

$\ln q = \Sigma_{k=1}^{r-1} \ln(1-k/S) \cong \Sigma_{k=1}^{r-1} -k/S = -r(r-1)/2S$

Taylor: if x « 1: $\ln(1-x) \cong x$

summation: $\Sigma_{k=1}^{r-1} k = r(r-1)/2$

hence $p = 1 - q \cong 1 - \exp(-r(r-1)/2S)$

Intermezzo: Gauss’s formula

- $G_{r-1} = \Sigma_{k=1}^{r-1} k = ?$
- $G_{r-1} = 1 + 2 + .. + r\text{-}2 + r\text{-}1$
- $G_{r-1} = r\text{-}1 + r\text{-}2 + .. + 2 + 1$
- $2G_{r-1} = r + r + .. + r + r$
- $2G_{r-1} = r\,(r\text{-}1)$
- $G_{r-1} = r\,(r\text{-}1)/2$

The birthday paradox (3) – without proof

- Given a set with S elements, in which we choose r elements at random (with replacements) with r « S
- The number of collisions follows a Poisson distribution with $\lambda = r(r-1)/2S$
  - the expected number of collisions is equal to λ
  - the probability to have c collision is $e^{-\lambda} \lambda^c / c!$

67

67

The birthday paradox: CBC (4)

- the ciphertext blocks C_i are random n-bit strings or $S = 2^n$
- if we collect $r = \sqrt{2^n} = 2^{n/2}$ ciphertext blocks, we will have a high probability that there exist two identical ciphertext blocks, that is, there exist indices i and j such that $C_i = C_j$
- this leaks information on the plaintext (see above)

68

68

The birthday paradox: CBC (5)

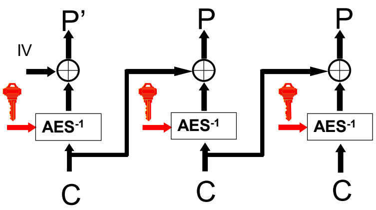
- for DES, n = 64: leakage after 2^{32} 64-bit blocks or 32 Gbyte
- for AES, n = 128: leakage only after 2^{64} 128-bit blocks or 256 Exabyte
- Example: DES with an encryption speed of 1 Gbit/s,
  - one expect the first collision after 4.5 minutes
  - after 19.5 hours, one has obtained 2^{40} ciphertext blocks; the expected number of collisions is then $(2^{40})^2 / 2^{65} = 2^{15}$
  - !! warning (frequent mistake): after 19.5 hours, the number of collision is NOT $2^{40} / 2^{32} = 2^8$
- Solution: change key quickly or use larger block length

69

69

CBC: insecure against chosen ciphertext attack

- CBC is very easy to distinguish with **chosen ciphertext** attack:
  - decrypting C || C || C yields P' || P || P

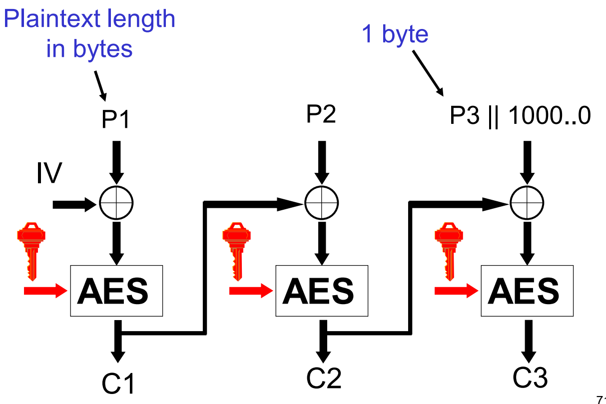


- The weaknesses of CBC decryption can be exploited through padding oracle attacks, hence today CBC is not longer recommended (idea is explained on next slides)

70

70

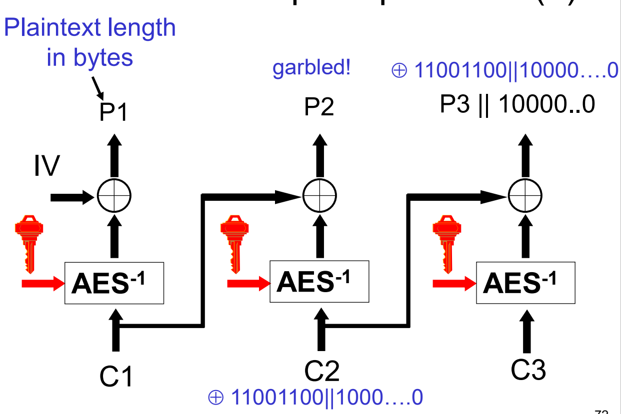
CBC with incomplete plaintext (1)



71

71

CBC with incomplete plaintext (2)



72

72

CBC with incomplete plaintext (3)

Plaintext length in bytes

garbled! $\oplus$ 11001100||10000….0

P1 P2 P3 || 10000..0

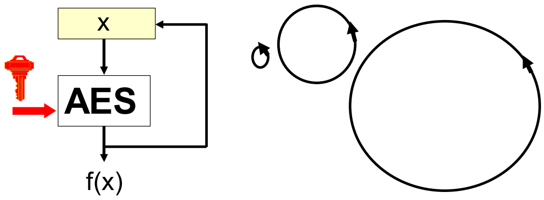
- If the first 8 bits of P3 are equal to 11001100 then after the modification P3' will be equal to 0
- The decryption will then produce an error message because the plaintext length field is incorrect
- Conclusion: information on 1 byte of P3 can be obtained using on average 128 chosen ciphertexts
- Protection: a careful implementation of random padding (no error messages) or authenticate the ciphertext (authenticated encryption)

73

73

Security of the OFB mode

Consider the functional graph of the f(x) = $AES_K(x)$ for a fixed key



repetition of key stream is dangerous (think of the Vernam scheme), but fortunately the expected cycle length is 2^{n-1}

the probability that a random point lies on a cycle shorter than c is c / 2^n

the OFB key stream does not have a repetition after $2^{n/2}$ blocks, so it deviates from random and leaks some information on the plaintext blocks!

74

74

Security of OFB mode (2)

- select a random point (IV)
- what is the probability p_c that the cycle on which IV lies has length exactly c?

  $p_c = (1/2^n) \,.\, (1-1/2^n)^{c-1}$
- what is the probability p that the cycle on which IV lies is $\leq$ c?

  $p = \Sigma_{k=1}^{c}\ p_k \cong c/2^n$
- here we have used the sum of a geometric expression

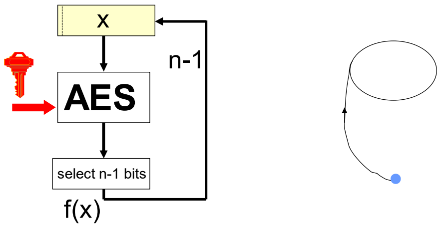
  $\Sigma_{k=0}^{c-1}\ q^k = (1-q^c)/(1-q)$

75

75

Security of a variant of the OFB mode (1)

Consider the functional graph of the function f(x) = leftmost n-1 bits of $AES_K(x)$ for a fixed key



starting in one point, we expect to hit a cycle (by the birthday paradox) after approximately $2^{n/2}$ values

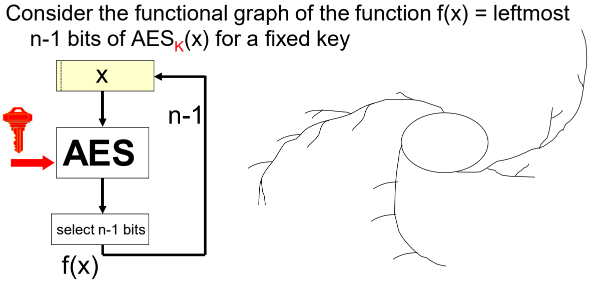
this means that the key stream repeats after only $2^{n/2}$ blocks!!!

76

76

Security of a variant of the OFB mode (2)

Consider the functional graph of the function f(x) = leftmost n-1 bits of $AES_K(x)$ for a fixed key

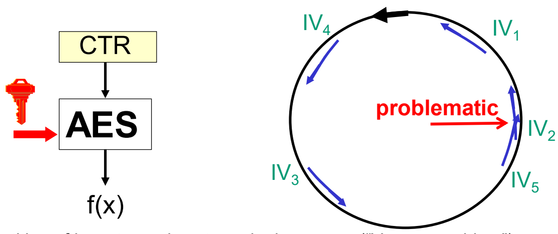


the complete functional graph looks as above (hairy ball)

77

77

Security of the CTR mode with random IV



repetition of key stream is extremely dangerous ("Venona problem")

the cycle length is 2^n

the CTR key stream does not have a repetition after $2^{n/2}$ blocks, so it deviates from random and leaks some information on the plaintext blocks (but this is not as bad as in the case of CBC mode)

if `many' random IVs (starting value for the CTR) are used and in total $2^{n/2}$ ciphertext outputs are generated, a collision is expected (and one gets a long batch of repeating key stream, which is worse than in the case of CBC!)

78

78

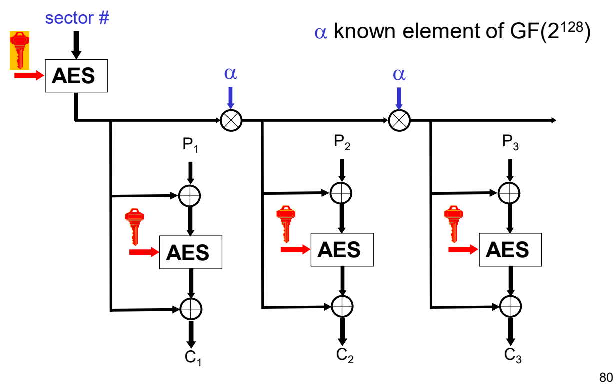
Hard disk encryption

- Large fixed-size blocks (1 sector = 512 bytes)
- No redundancy possible: size of ciphertext should be equal to size of plaintext
- Hence encryption cannot be randomized

79

79

XEX (XOR-Encrypt-XOR) mode: disk encryption



80

80

Block cipher versus stream cipher

block cipher

- standards (DES, 3-DES, AES)
- blocks of 64-128-256 bits
- needs a mode (has no internal memory)
- flexible building block – can also be used as a stream cipher
- easier to make secure since many operations per bit (memory inside the function)

stream cipher

- lack of widely accepted standards
- operates at bit, byte or word level (1-8-32)
- often simpler hence very low cost (in hardware/power) or very high speed

81

81

Secret versus publicly known algorithm

publicly known algorithm

- open and independent evaluation
- standardization
- availability of (certified) implementations

secret algorithm

- additional security...? (maybe against side channel)
- only acceptable if there is a budget for independent evaluation and re-evaluation

82

82

Exercises (1)

1. Show that in DES the encryption implementation can be reused for decryption (if the order of the round keys are reversed).
2. Birthdays. Consider a classroom with 32 students.
   - Compute the probability that at least one student has his/her birthday on January 1.
   - Compute the probability that exactly two students have their birthdays on December 31.
   - Compute the probability that at least two students have the same birthday (you can use an approximation)

(you can assume that the birthdays are equally distributed over the year and that there are no leap years)

83

83

Exercises (2)

3. Try to find a meet-in-the middle shortcut attack on 3-DES with 2 keys and 3 keys. Are these attacks practical today?
4. Find the period of the CTR mode. When does the CTR mode start leaking information on the plaintext?
5. A 64-bit block cipher is used in CBC mode with a speed of 2 Gigabit/s (2 10^9 bits/s)
   - How long does it take before information starts to leak on the plaintext?
   - How many collisions do you expect after 1 hour?

84

84