UNIK4250 Security in Distributed Systems
UNIK University Graduate Center
Spring 2012

Lecture 2

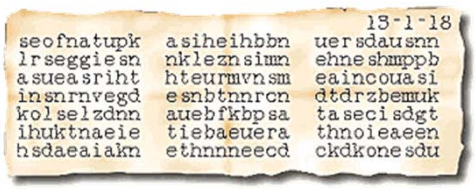
Symmetrical Encryption and Message Confidentiality

Leif Nilsen
Ed. 1.0

UNIK

Outline

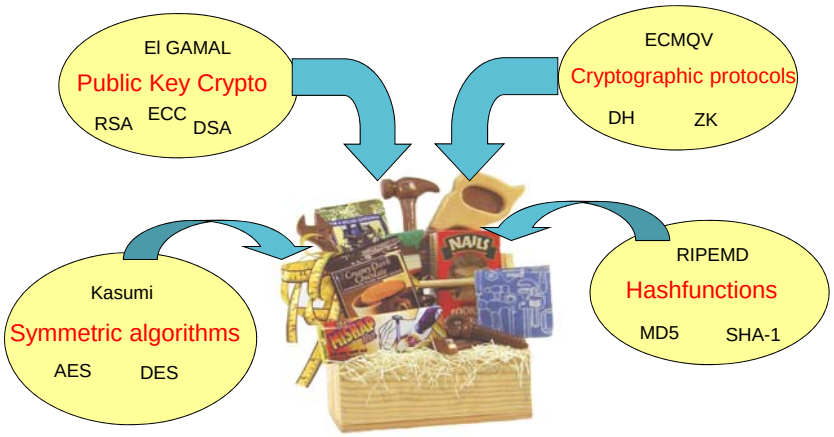
- What is cryptography?
- Where is cryptography used?
- Cipher types
- Classic symmetric ciphers
- Ciphers and security



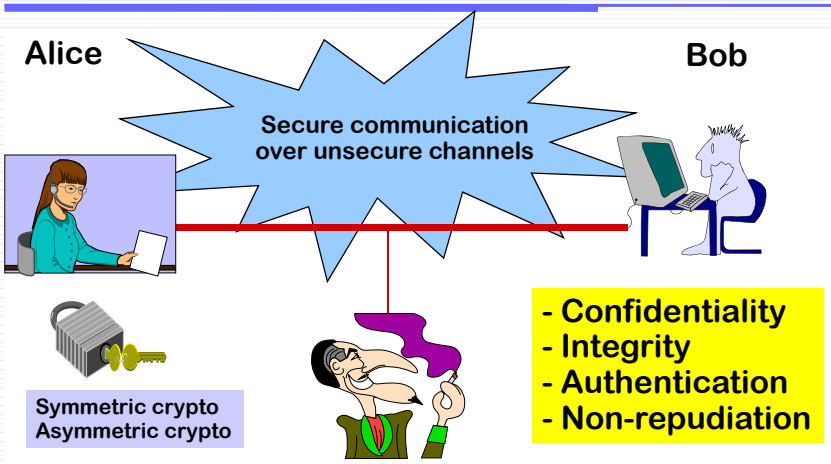
What is cryptography?

- Cryptography is part of the field of study known as **cryptology**.
- Cryptology includes
  - cryptography:
    - derived from the Greek, means 'hidden writing'.
    - the study of methods for secret writing: for transforming messages into an unintelligible form, and for recovering them, using some secret knowledge.
  - cryptanalysis:
    - analysis of cryptographic systems, inputs and outputs to derive confidential information, usually without using the secret knowledge.

The cryptographic toolbox



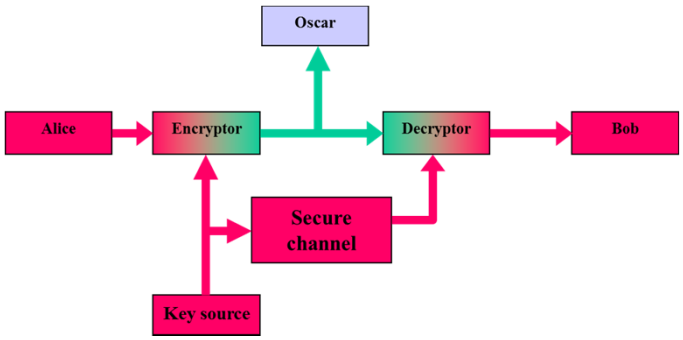
What is cryptography?



Some Basic Terminology

- **plaintext** - original message
- **ciphertext** - coded message
- **cipher** - algorithm for transforming plaintext to ciphertext
- **key** - info used in cipher known only to sender/receiver
- **encipher (encrypt)** - converting plaintext to ciphertext
- **decipher (decrypt)** - recovering ciphertext from plaintext
- **cryptography** - study of encryption principles/methods
- **cryptanalysis (codebreaking)** - study of principles/ methods of deciphering ciphertext *without* knowing key
- **cryptology** - field of both cryptography and cryptanalysis

Model of symmetric cryptosystem



What is NOT cryptography?

- **Steganography:** – used to hide the existence of a message
  - Hide the information within a document or image, so that the presence of the message is not detected
- **Steganographic techniques include**
  - Using invisible ink (try writing in lemon juice)
  - Microdots
  - Character arrangement and selection
  - Hiding information, e.g. in graphics and sound files
- **Steganographic techniques do not use a secret key**

When is cryptography used?

- If you require
  - **Confidentiality:**
    - so that your data is not made available to anyone who shouldn't have access.
    - That is, protection against snoops or eavesdroppers
  - **Integrity:**
    - So you know that the message content is correct, and has not been altered, either deliberately or accidentally
  - **Authentication:**
    - So you can be sure that the message is from the place or sender it claims to be from
- Cryptography can provide these security services.

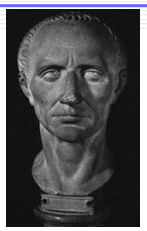
When is cryptography used?

- Some example situations:
  - **Historically**, the military and spy agencies were the main users of cryptology
    - Situation: transmitting messages over insecure channels
  - **Now**, it is used in many other areas, especially in electronic information processing and communications technologies:
    - Banking: your financial transactions, such as EFTPOS
    - Communications: your mobile phone conversations
    - Info stored in databases: hospitals, universities, etc.
- Cryptography can be used to protect information in storage or during transmission

When is cryptography used?

- Cryptographic mechanisms such as ciphers and hash functions can provide data integrity services.
- If a message is altered, the changes to a message or data file can be detected using:
  - manipulation detection codes (MDC)
    - based on (unkeyed) hash functions
  - message authentication codes (MAC)
    - based on keyed hash functions (such as HMac), or
    - Block ciphers used in suitable modes.

Historical ciphers



Example 1 : Caesar chipher

$\mathcal{P}$ = {abcdefghijklmnopqrstuvwxyz}
$\mathcal{C}$ = {DEFGHIJKLMNOPQRSTUVWXYZABC}

Plaintext: kryptologi er et spennende fag
Chiphertext: NUBSWRORJL HU HW VSHQQHQGH IDJ

Note: Caesar chiper may be seen as a general shift cipher with K = 3.

$c = e_k(p) = p + k \pmod{26}$, i.e. 26 possible keys

A general monoalphabetic substitution cipher has 26! = 403291461126605635584000000 keys (88 bits)
Easily broken using statistics of the underlying language

Historical ciphers

Example 2 : Vigenére (1523-1596) chipher

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
k →	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
o →	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
p →	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
r →	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
t →	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y →	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

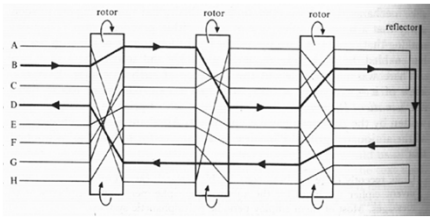
Key: **kryptokry**
Plaintext: **OLAOGKARI**
Chiphertext: **ycydzykig**

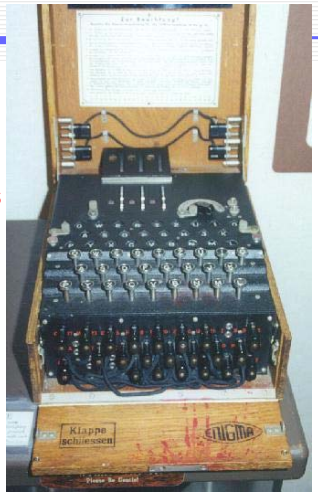
Easily broken using statistics of the underlying language

Historical ciphers

Example 3 : Enigma

- German encryption device used under WW2
- Many variants
- Used by Norwegian security police in 1950s
- Broken by Polish and English mathematicians (not easily)

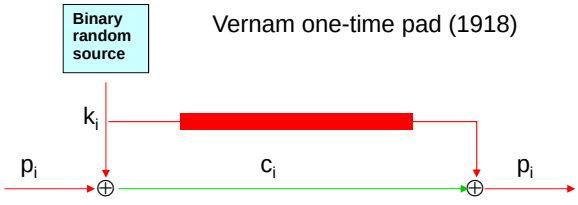


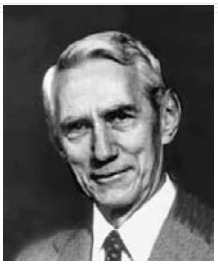


Is there a 'perfect' cipher?

- What is a secure cryptosystem?
- Is it possible to design secure crypto?

A perfect secure cipher





$c_i = p_i \oplus k_i$
$p_i = c_i \oplus k_i = p_i \oplus k_i \oplus k_i = p_i$

Provides perfect security if and only if the key is random, of the same length as the message and is only used once! Proved by Claude E. Shannon i 1949.

The Norwegian Contribution

- Electronic Teleprinter Cryptographic Regenerative Repeater Mixer (ETCRRM)
- Invented by the Norwegian Army Signal Corps in 1950
- Bjørn Rørholt, Kåre Mesingseth
- Produced by STK
- Used for ”Hot-line” between Moskva and Washington
- About 2000 devices produced

White House Crypto Room 1960s

Attack models

- Ciphertext only
- Known plaintext
- Chosen plaintext (adaptive)
- Chosen ciphertext (adaptive)

The goal of an opponent is to find the secret key or some unknown plantext

How clever is the attacker?

Notions of security

Unconditional security

Means that there are no restrictions on the amount of operations Oscar is allowed to do in order to break the system. The system cannot be broken, even with infinite computational resources.

Computational security

Means that the best known algorithms for breaking the systems require a huge number of computations (time complexity) or huge amount of data (memory complexity). It is practical impossible for Oscar to break the system. It is difficult to prove that a system is secure in this model.

Provable security

Means that breaking the system can be proved equivalent to solve a difficult problem (factoring, discrete logarithm)

Goals of an opponent

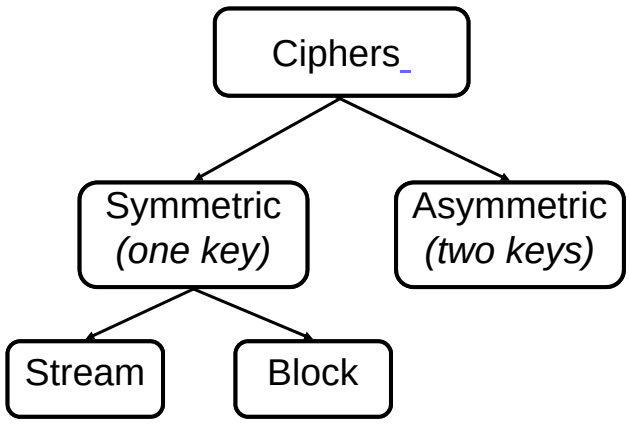
- Finding the secure key (total break)
- Determine an unknown plaintext
- Determine a few bits in the unknown plaintext
- Indistinguishability of two plaintexts

Kerckhoff's principles

- The system should be, if not theoretically unbreakable, unbreakable in practice.
- The design of a system should not require secrecy and compromise of the system should not inconvenience the correspondents (Kerckhoffs' principle).
- The key should be rememberable without notes and should be easily changeable
- The cryptograms should be transmittable by telegraph
- The apparatus or documents should be portable and operable by a single person
- The system should be easy, neither requiring knowledge of a long list of rules nor involving mental strain

Diagram

Taxonomy of modern ciphers



Notation for Cipher Operations

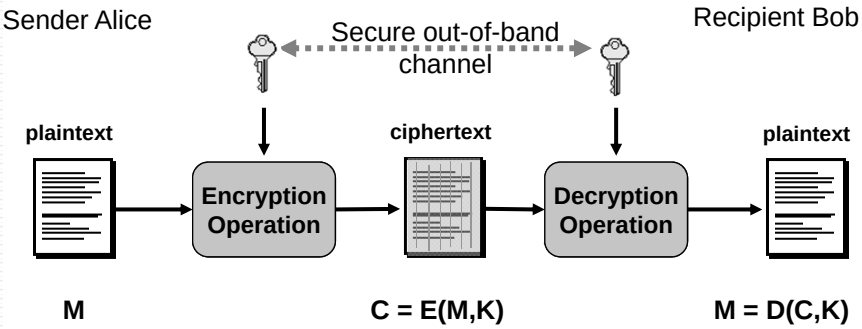
- Notation
  - Plaintext message: M
  - Encryption: E
  - Cryptographic Key: K
  - Ciphertext: C
  - Decryption: D
- Basic Operations
  - Encryption: C = E(K,M)
  - Decryption: M = D(K,M)

Symmetric ciphers

- Encryption and decryption keys are the same (or one can easily be deduced from the other)
- The encryption and decryption algorithms are usually made public
- The cryptographic key K
  - must be kept secret
  - is used for both encryption and decryption, so has to be distributed or stored securely
- Two types of symmetric ciphers:
  1. Stream ciphers
  2. Block ciphers

Diagram

Symmetric ciphers: Operation



Requirements

- two requirements for secure use of symmetric encryption:
  - a strong encryption algorithm
  - a secret key known only to sender / receiver
- mathematically have:

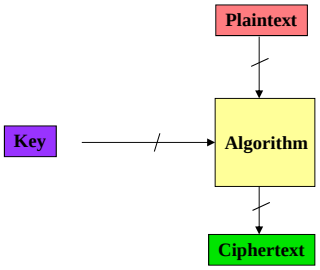
  C = E(M, K)

  M = D(C, K) = D(E(M,K),K)
- assume encryption algorithm is known
- implies a secure channel to distribute key

Cryptography

- can characterize cryptographic system by:
  - type of encryption operations used
    - substitution
    - transposition
    - product
  - number of keys used
    - single-key or private
    - two-key or public
  - way in which plaintext is processed
    - block
    - stream

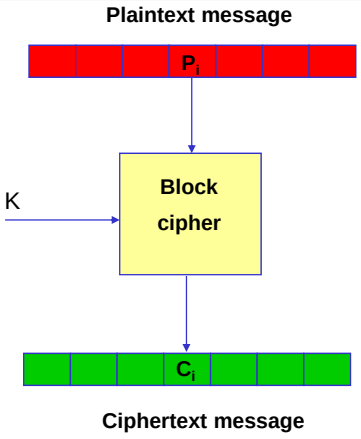
Blockcipher model



Electronic Code Book (ECB-modus)

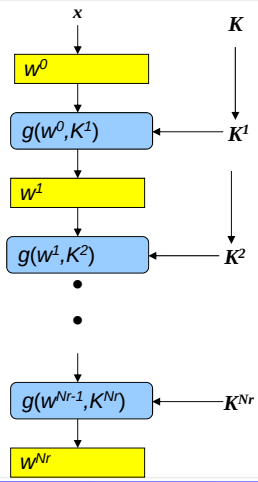
For a fixed key the algorithm spesifices a permutation on the (large) set of all possible blocks!
An ideal block cipher should «look» like a set om random permutations on the set of n-bit blocks.
Must be impossible to recover key from known P and C.

Symmetric blockchipher



- The algorithm represents a family of permutations that should look a set of random permutations!
- Normally designed by iterating a weaker round function
- Can be used in several different modes of operation
- Must be impossible to deduce K from known P and C

Iterated blockcipher



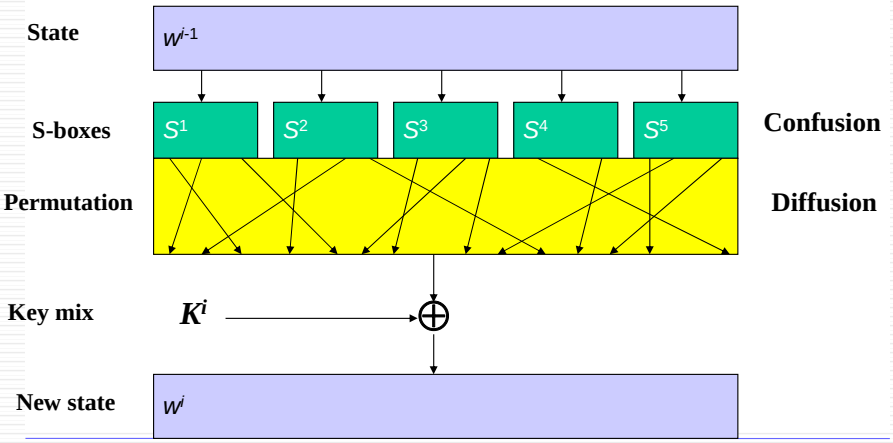
Algorithm:

$w^0 \leftarrow x$
$w^1 \leftarrow g(w^0,K^1)$
$w^2 \leftarrow g(w^1,K^2)$
•
•
$w^{Nr-1} \leftarrow g(w^{Nr-2},K^{Nr-1})$
$w^{Nr} \leftarrow g(w^{Nr-1},K^{Nr})$
$y \leftarrow w^{Nr}$

NB! For fixed value of K, *g* must be invertible in order to decrypt *y*

Substitution-Permutation network (SPN)

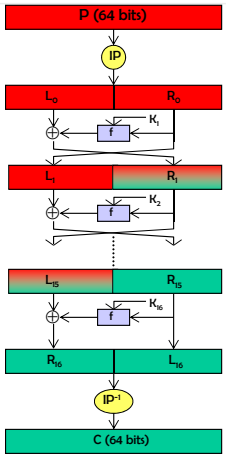
Round function *g*:



Data Encryption Standard

- FIPS PUB 46, published as US Federal Standard in 1977
- Developed by IBM team lead by Horst Feistel
- Leading symmetric block cipher for 25 years
- On 19 May 2005, FIPS 46-3 was officially withdrawn
- 64 bit block size
- 56 bit key

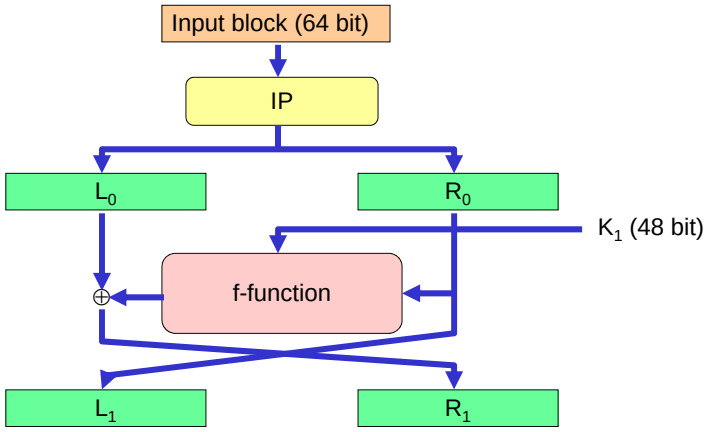
DES architecture



DES(P):
$(L_0, R_0) = IP(P)$
FOR i = 1 TO 16
$L_i = R_{i-1}$
$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
$C = IP^{-1}(R_{16}, L_{16})$

64 bit datablocks
56 bit key
72.057.594.037.927.936

DES (Data Encryption Standard)

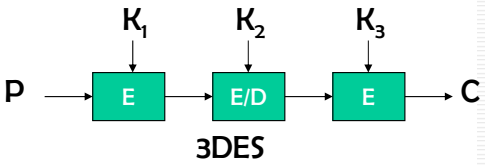


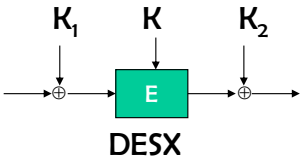
EFF DES Cracker

- Dedicated circuit with 24 DES search engines
- 27 PCBs totalling 1536 chips
- Can test 88 billion keys per second
- Cost 250.000 $
- DES key found July 1998 after 56 hours search
- New project using DES Cracker and100.000 PCs could test 245 billion keys per second and found key after 22 hours!

DES status

- DES is the “work horse” which for 30 years has inspired cryptographic research and development
- “Expired”!
- Single DES can not be considered to be a secure cipher!
- Use 3DES (ANSI 9.52) or DESX





From DES to AES

- A replacement for DES became necessary
  - 56 bit is to short
  - 64 bit block size could be questioned?
- Growing need for good confidentiality services
  - Internet, e-commerce
  - Mobility
- 3-DES is inefficient
  - Short time solution to extend the life time of installed technology
  - 48 round are needed to encrypt 64 bit
- 1997 NIST initiated a project to develop AES
- Open and international process

Schedule

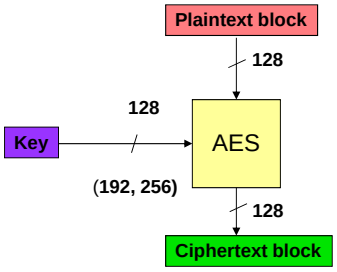
- 2. Jan. 1997 - NIST announces the AES project
- 15. April 1997 - Initial AES Workshop
  - Draft requirements for algorithm and nominations
- 12. Sept. 1997 - NIST calls for submission
  - Deadline 15. June 1998
- Aug. 1998 - The First AES Candidate Conference
  - 15 candidates accepted
- Mars 1999 - The Second AES Candidate Conference
- 9. Aug. 1999 - Presentation of 5 finalists
- April 2000 - The Third AES Candidate Conference
- 2. Oct. 2000 - Presentation of the winner

Evaluation criteria

- Security
  - Mathematical foundation
  - Cryptanalysis
- Cost
  - Effective with regards to time and memory
- Algorithm and implementation
  - Flexible over different platforms
  - Suitable for HW and SW
  - Design simplicity
- Other issues
  - Access to implementations, IPR

AES parameters

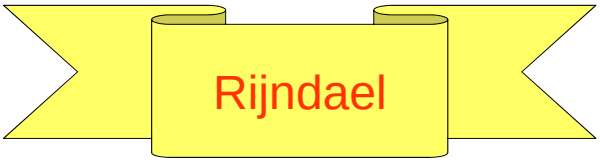
- Symmetric blockchipher algorithm
- 128 bits blocks in and out
- Adjustable key size 128, 192 or 256 bits



Electronic Code Book
(ECB-modus)

Candidates round 2

- MARS - IBM
- RC6 - Ron Rivest, RSA
- RIJNDAEL - Joan Daemen, Vincent Rijemen
- SERPENT - L. Knudsen, E. Biham, R. Anderson
- TWOFISH - B. Schneier, Counterpane

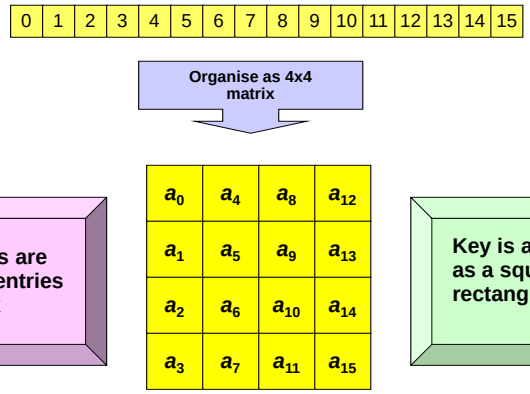


FIPS-PUB 197

Federal Information
Processing Standards Publication 197
November 26, 2001
Specification for the
ADVANCED ENCRYPTION STANDARD (AES)

Rijndael

128 bit data block is organised as 16 octets



All operations are done on the entries of this matrix

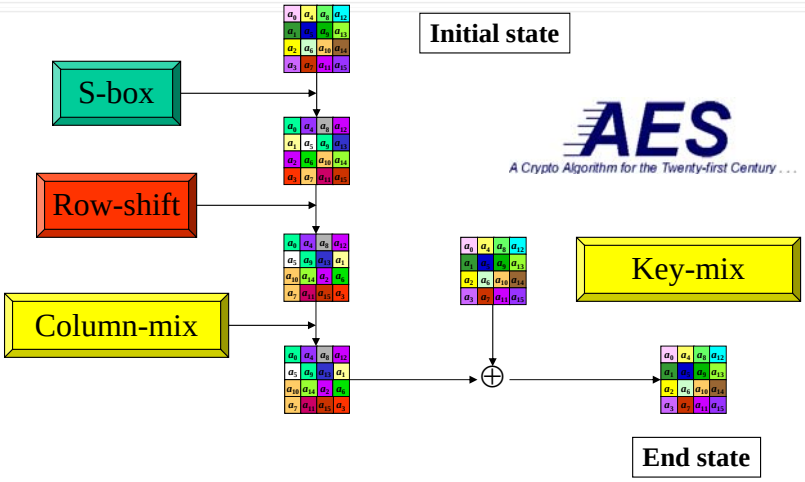
Key is also organised as a square or rectangle (k > 128)

Rijndael operations

- **Rijndael uses four invertible operations:**

1. Byte substitution (S-box)
2. Row shift (rotate the rows in the matrix)
3. Column mix (linear mix of a column word)
4. Key mix

AES (Advanced Encryption Standard)



Rijndael encryption

1. Key mix(round key K_0)
2. N_r-1 rounds consisting of:
   a) Byte substitution
   b) Row shift
   c) Column mix
   d) Keymix (round key K_i)
3. Last round consisting of:
   a) Byte substitution
   b) Row shift
   c) Keymix (round key K_{Nr})

Key	Rounds
128	10
192	12
256	14

Rijndael security

- None complementary properties, no weak keys
- Security totally dependent of the S-box
- Best theoretical attack known has complexity 2^{120} and works for 8 rounds
- No known attacks against full Rijndael
- Critizised for “low security margin”
- “Related key attack” for AES-192 og AES-256, complexity $2^{99.5}$
- “Side channel attacks”, timing attack against cache look up

Rijndael implementation

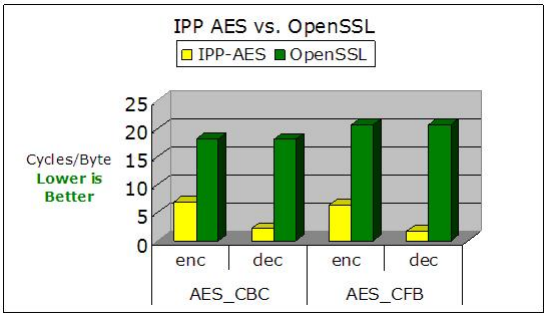
- Effective implementations for all platforms!
- On 32 bits architectures the complete round function can be compted using 16 table look ups (8 bit in - 32 out) and 16 xor
  Based on 4 precomputed tables (4Kb)
- Roundkeys can be generated once, or “on the fly”
- Direct implementation of the inverse algorithm looks a bit more complicated, but optimizations exist
- AES-128 at around 18 cycles/byte = 110 MB/s @ 2GHz

Implementation updates

- 2000: Aoki and Lipmaa report 14.8 cycles/byte on Pentium II
- . . .
- 2007: Matsui and Nakajima report 9.2 cycles/byte for AES-CTR on Core 2
  - Assuming data is processed in 2 KB blocks
  - Compatibility with existing implementations via an extra input/output transform
- 2008: Bernstein-Schwabe report 10.57 cycles/byte for AES-CTR on Core 2
- 2009: Käsper-Schwabe report 7.59 cycles/byte for AES-CTR
- on Core 2

2010

- Intel® AES instructions are a new set of instructions available on the 32nm Intel® microarchitecture (formerly codenamed Westmere-EP). These instructions enable fast and secure data encryption and decryption, using the Advanced Encryption Standard (AES) which is defined by FIPS Publication number 197 and widely used today in secure commerce, database and full disk encryption.



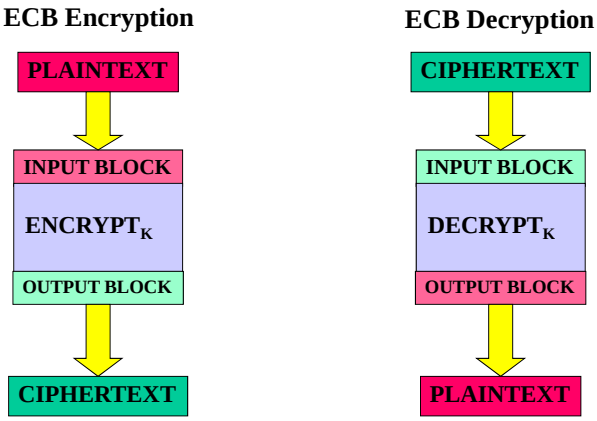
AES Status

- AES approved as US federal standard 27.11.2001
- ISO-standard: ISO/IEC 18033
- Used as crypto engine in MILENAGE algorihtm suite for authentication and key derivation in UMTS.
- Approved for classified information, but requires certified implementation.

Modes of operation

- How should a block cipher be used in practical applications?
  - ECB
  - CBC
  - OFB
  - CFB
  - CTR
  - GCM

Electronic Code Book (ECB)

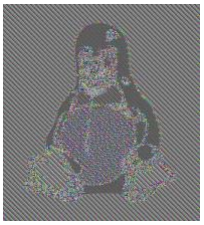


ECB mode properties

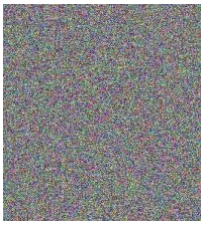
- Identical plaintext blocks (for same key) results in identical ciphertext blocks.
- Block independence: Data blocks are encrypted/decrypted independent of other blocks. Shuffling of ciphertext blocks result in corresponding shuffling of plaintext blocks.
- Error expansion: One or more errors in one ciphertext block only affect that block. For most algorithms the decrypted block will be completely corrupted (approx 50% of recovered bits incorrect).
- Allows for pipelined implementation.

Use a safe mode

Plaintext

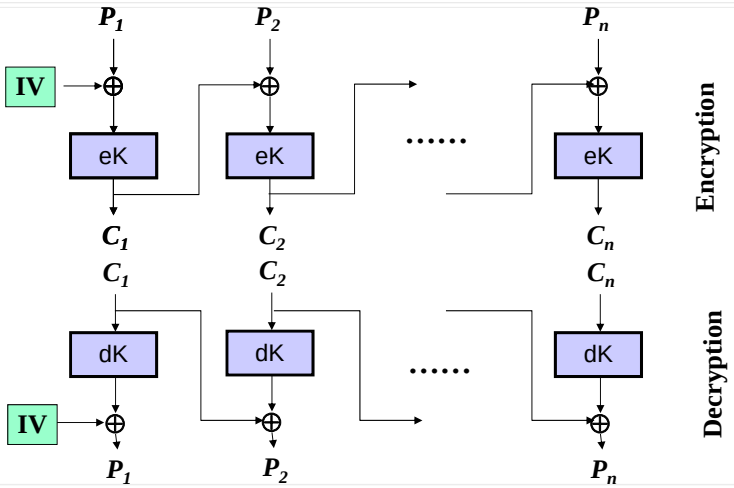


Ciphertext using ECB mode



Ciphertext using cecure mode

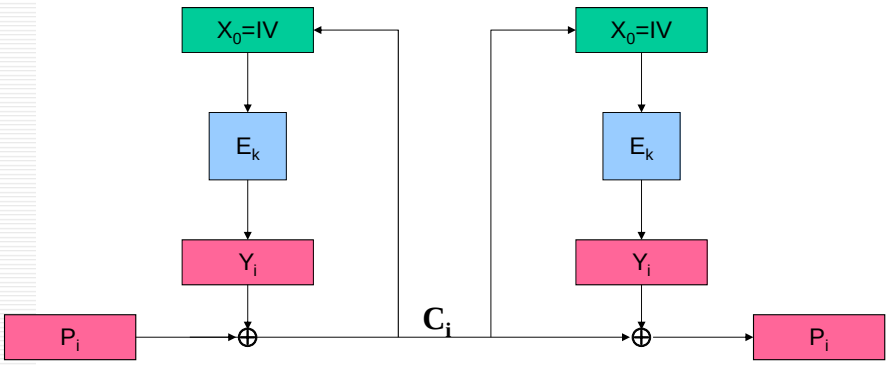
Cipher Block Chaining (CBC)



CBC mode properties

- Identical plaintexts gives identical chiphertexts for same K and IV. Change in K, IV or first plaintext block will result in different ciphertexts.
- Chaining makes ciphertext block C_i dependent of plaintext block P_i and all previous plaintext blocks. Correct decryption requires correct receipt of corresponding and previous ciphertext block.
- One bit error in position *j* of C_i results in completely random result in decrypted P_i and additional bit error in position *j* of P_{i+1}.
- The mode provides self synchronisation if block limits are maintained. Error in C_i, but correct receipt C_{i+1} and C_{i+2}, will result in correct decryption of C_{i+2}.
- Cannot be “pipelined”!

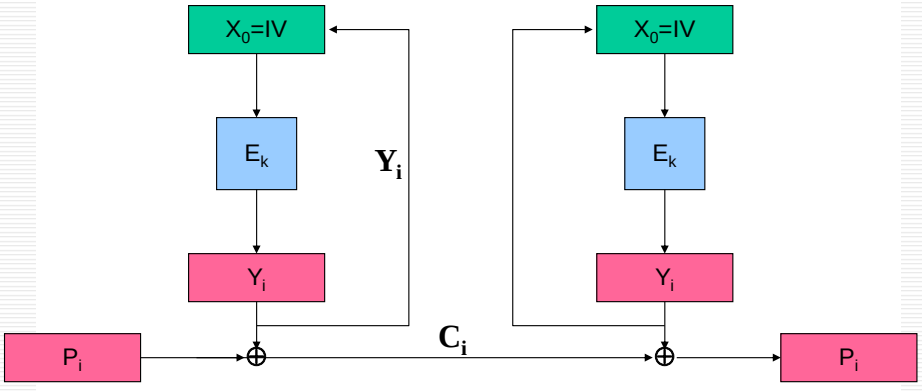
Cipher Feedback Mode (CFB)



CFB mode properties

- Identical plaintexts gives identical chiphertexts for same K and IV. Change in K or IV will result in different ciphertexts.
- Chaining makes ciphertext block C_i dependent of plaintext block P_i and all previous plaintext blocks. Correct decryption requires correct receipt of corresponding and previous ciphertext block.
- One bit error in position j of C_i results in completely random result in decrypted P_{i+1} and additional bit error in position j of P_i.
- Many implementations use 1-bit or 8-bit feedback. Resync when all errors have been shifted out of in-register.
- Shorter feedback results in slower performance.

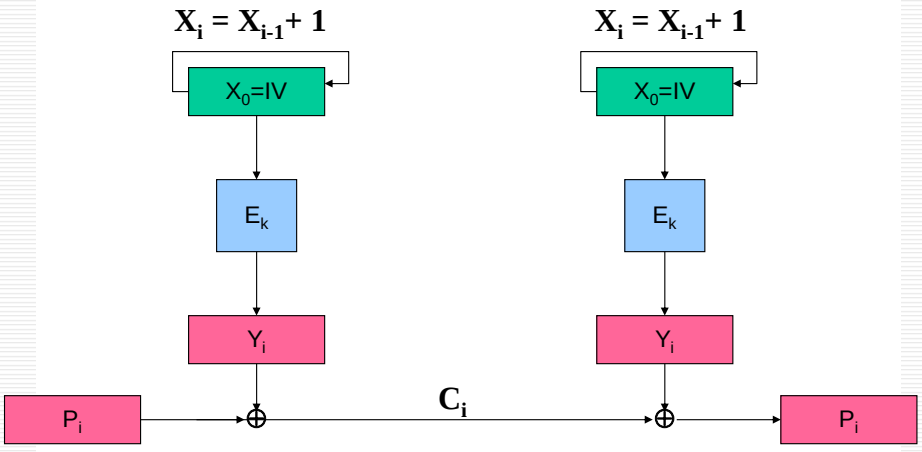
Output Feedback Mode (OFB)



OFB mode properties

- Identical plaintexts gives identical chiphertexts for same K and IV. Change in K or IV will result in different ciphertexts.
- Key stream is independent of plaintext.
- Error in ciphertext bit will result in error (complement) in corresponding plaintext bit.
- Will recover from bit errors, but requires new synchronization after bit loss.
- Shorter feedback than full block size will give reduced speed, but key stream may be generated off-line.

Counter Mode (CTR)



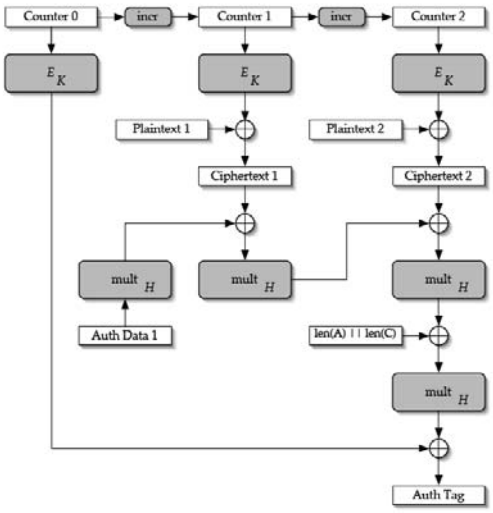
CTR mode properties

- Equal blocks results in the equal ciphertext block for the same counter start value.
- Key stream is independent of plaintext.
- Can control cycle period (avoid short cycles)
- Can be used for “random access”. Not necessary to decrypt blocks in the received order.

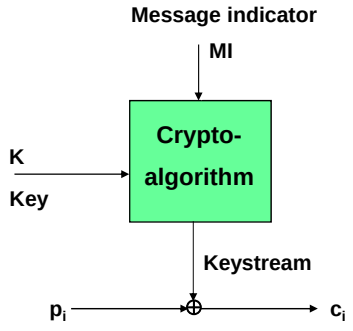
Galois Counter Mode (GCM)

- Mode of operation that combines encryption and authentication (authenticated encryption)
- Specified in NIST Spes. Pub 800-38D
- Primary designed for use with AES, but can be used with other block cipher as well
- Encryption using AES-CTR
- Authentication using ”Uniform hashing”, called GMAC
- Suitable for IPSEC and TLS
- Included in NSA Suite-B
- GMAC ”evaluates” the message as a polynomial over GF(2^{128}) der $f = 1 + x + x^2 + x^7 + x^{128}$

GCM

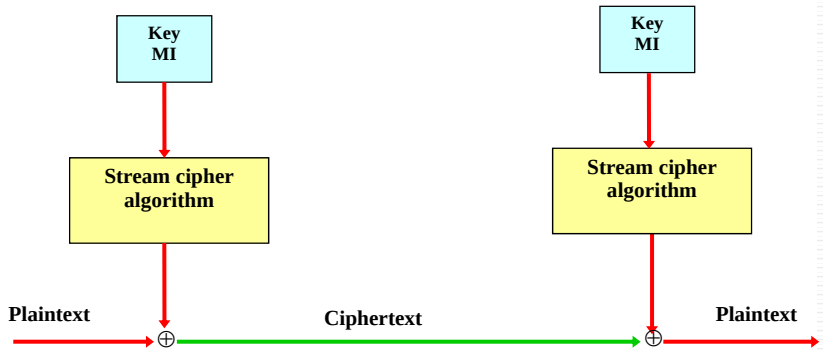


Symmetric stream cipher

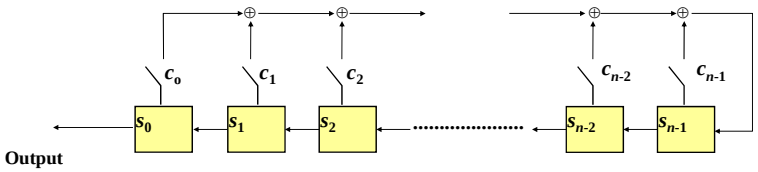


- The cryptoalgorithm generates a random keystream that is xored to the plaintext
- Receiver decrypts by adding the same keystream
- Blockcipher can be used in “stream modes”
- Dedicated algorithms: NSK, RC4, A5/1 (GSM), SNOW3G

Symmetric stream cipher



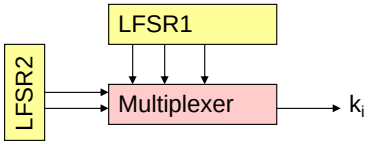
LFSR



Using n cells we can generate a «random» sequence with period 2^n -1

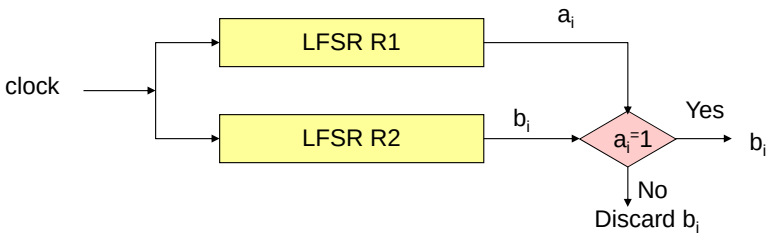
From LFSR to stream ciphers

- Non-linear combining
  - Output from several LFSRs used as input to a non-linear function
- Non-linear filtrering
  - Tap contents from several cells in an LFSR to a non-linear function
- Clock-controlled generator
  - Let one LFSR clock another that is used to generate the keystream
- Multiplexing



"Shrinking Generator"

- Coppersmith, Krawczyk og Mansour, 1993



RC4

- a proprietary cipher owned by RSA DSI
- another Ron Rivest design, simple but effective
- variable key size, byte-oriented stream cipher
- widely used (web SSL/TLS, wireless WEP/WPA)
- key forms random permutation of all 8-bit values
- uses that permutation to scramble input info processed a byte at a time

RC4 Key Schedule

- starts with an array S of numbers: 0..255
- use key to well and truly shuffle
- S forms **internal state** of the cipher

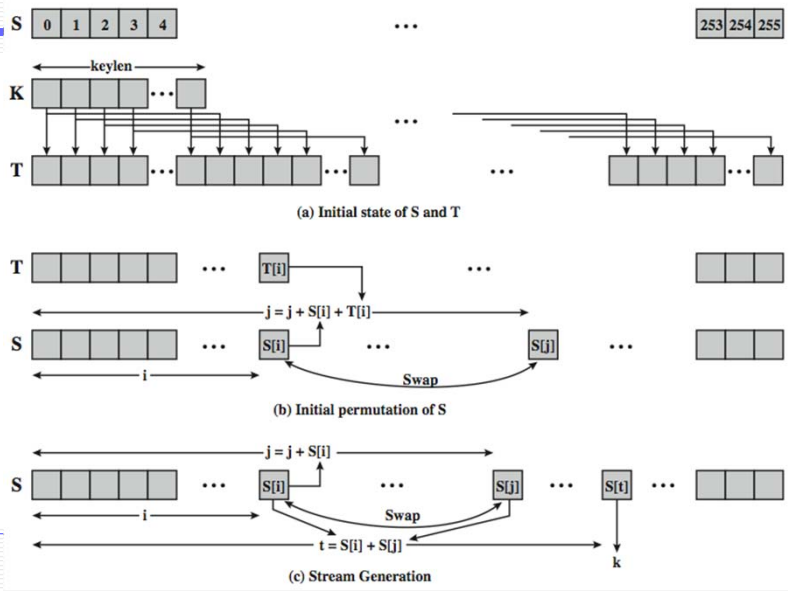
```
for i = 0 to 255 do
   S[i] = i
   T[i] = K[i mod keylen])
j = 0
for i = 0 to 255 do
   j = (j + S[i] + T[i]) (mod 256)
   swap (S[i], S[j])
```

RC4 Encryption

- encryption continues shuffling array values
- sum of shuffled pair selects "stream key" value from permutation
- XOR S[t] with next byte of message to en/decrypt

```
i = j = 0
for each message byte Mi
   i = (i + 1) (mod 256)
   j = (j + S[i]) (mod 256)
   swap(S[i], S[j])
   t = (S[i] + S[j]) (mod 256)
   Ci = Mi XOR S[t]
```

RC4 Overview



RC4 Security

- claimed secure against known attacks
  - have some analyses, none practical
- result is very non-linear
- since RC4 is a stream cipher, must **never reuse a key**
- have a concern with WEP, but due to key handling rather than RC4 itself

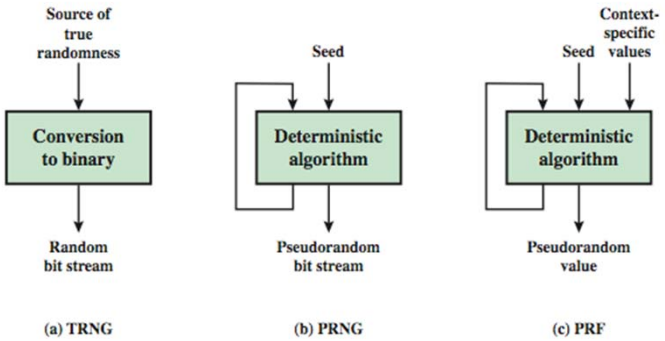
Random Numbers

- many uses of **random numbers** in cryptography
  - nonces in authentication protocols to prevent replay
  - session keys
  - public key generation
  - keystream for a one-time pad
- in all cases its critical that these values be
  - statistically random, uniform distribution, independent
  - unpredictability of future values from previous values
- true random numbers provide this
- care needed with generated random numbers

Pseudorandom Number Generators (PRNGs)

- often use deterministic algorithmic techniques to create “random numbers”
  - although are not truly random
  - can pass many tests of “randomness”
- known as “pseudorandom numbers”
- created by “Pseudorandom Number Generators (PRNGs)”

Random & Pseudorandom Number Generators



Random binary sequences

- What is a random sequence?
- Which sequence are random?
  a) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  b) 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1
  c) 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0
  d) 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0
  e) 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0

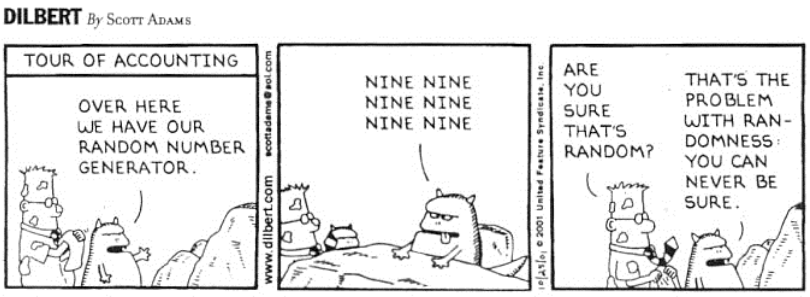
- *A "true" random source should generate each sequence of a given length* n *with uniform probability1/2*^{n}
- *What is typical for random sequences?*

FIPS PUB 140-2 Statistiske tester

- Generate 20 000 continous bits from the generator
  - Monobit test (9725 < #1 < 10 275)
  - Poker test (Kji-kvadrat test on 4 bits blocks)
  - Runs test (all runs of 0 and1 shall fulfill)
    Length: 1 - Interval: 2315 - 2685
    Length: 2 - Interval: 1114 - 1386
    Length : 3 - Interval: 527 - 723
    Length : 4 - Interval: 240 - 384
    Length : 5 - Interval: 103 - 209
    Length : 6+ - Intervll: 103 - 209
  - Long runs test (no run equal or longer than 26)

- D. Knuth, The Art of Computer Programming - Vol. 2 standard reference for random testing. (NB! Not cryptographic)

What is random?



Summary

- have considered:
  - Classical cipher techniques and terminology
  - Security aspects
  - Symmetric cryptography
  - Block ciphers
  - DES and AES
  - Modes of operation
  - Stream ciphers
  - RC4
  - Random numbers