Introduction to Cryptology

Lecture-08

Secret-Key Ciphers: Block Ciphers:
Design Principles and Contemporary Standards

03.05.2023, v54

Block Ciphers

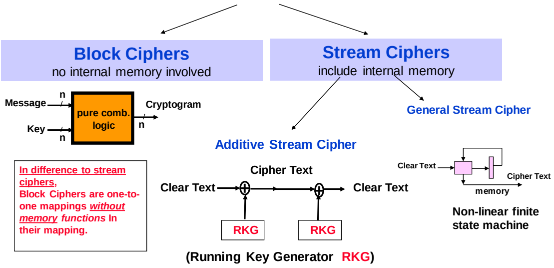
Design Fundamentals and Standards

Outlines

- **Basics of Block Ciphers**
- **Modern Block Ciphers since 1976**
- **Some Contemporary Standards**

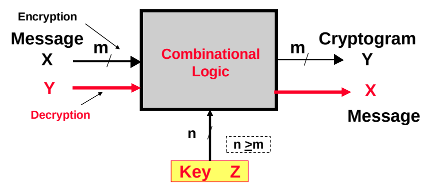
Designing block ciphers is a very challenging task.
A nice introductory short course:
EIDMA minicourse MC-CIC-7 Design and Analysis of Block Ciphers
by Prof. James L. Massey:
http://www.win.tue.nl/wsk/eidma/courses/minicourses/massey/MC-CIC-7.html

Two Major Secret Key Cipher Classes

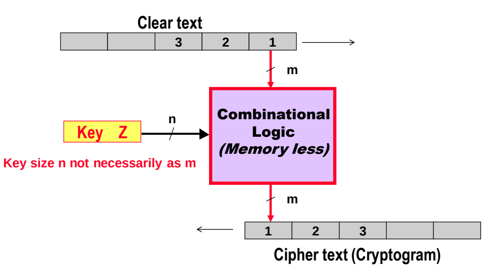


Block-Ciphers

A block cipher Is like a code-book: for each input clear text, there is a corresponding unique cipher text for a certain key and vice versa

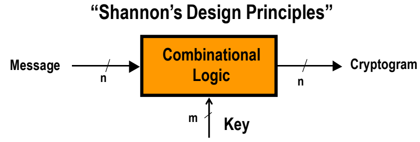


Block-Ciphers on Data Streams



Key Concepts for Block-Ciphers

"Shannon's Design Principles"



Basic Design Requirements: (Shannon ***diffusion*** and ***confusion*** principles)

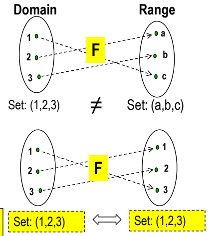
1. If key is not known, there no identifiable relation between message, key and cryptogram
2. Any minor changes in message or key, result in tremendous changes in the cryptogram
3. No leakage of any part of the key or the message bits to the cryptogram
4. Exhaustive search attack should be infeasible
   (example: key size = 100 bits, then $\leq 2^{100}$ search cycles are necessary).

*To compete with today's computation technology, **more than 2^{80} search cycles are required.***
That is, key size for acceptable security level should be at least 80 bits

Targeted Crypto-Mappings

Bijective Invertible Function/Mapping :
lls a mapping which is:
One-to-One AND Onto at the same time

- A function **F** has an inverse function $\mathbf{F^{-1}}$, *if and only if* F is one to one



Permutation:
A **bijective function** from a set to itself is also called a **permutation** *(Domain = Range)*

Crypto-Mappings use permutations to keep data size unchanged.
(domain space is the same as the range pace)

Number of permutations: **3!** = 3 x (3-1) x (3-2) = 6

6 possible permutations
123 to 123
123 to 132
123 to 231
123 to 213
123 to 312
123 to 321

In general: |F| = Number of n to n permutations = n!

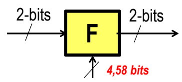
Example of all invertible 2-to-2 bit mappings

$n = 2^2 = 4$ possible input combinations

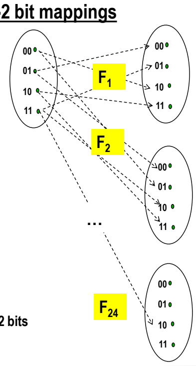
No. Possible invertible mappings = $n! = 2^2\,! = 4! = 24$

No. of selection bits = $\log_2 24$ = 4,58 bits

Memory Implementation:



implementation complexity in bits = $2x2^2$ x 24 = 192 bits = $2^{7,58}$ bits

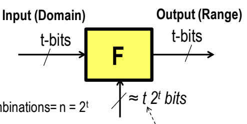


General Permutations Bounds

Engineering approach:

Mapping as a hardware-block:

A mapping function from **t-bits to t-bits** keeping the input space=output space
(Data size unchanged!)



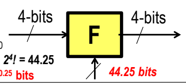
- Domain size = number of possible input combinations= $n = 2^t$
- Number of all possible "F" mappings = $2^{t\,2^t}$
- Number of all possible **invertible F mappings** $S_{max} = 2^t\,!$

Stirling's approximation $S_{max} = 2^t! \approx [\,2^t/e\,]^{2^t}$ or $S_{max} \approx 2^{(t-1.45)\,2^t}$

- Number of bits needed **to select all S_{max} mappings = $\log_2 S_{max} = \log_2 2^t\,! \approx t\,2^t$ bits**

Example: *t=4*, 4-bit to 4-bit mapping

Number of all possible mappings = $2^{4x16} = 2^{64}$ mappings
Number of possible invertible mappings = $2^4\,!$ = 20.922.789.888.000
 # bits required to select all possible invertible mappings = $\log_2 2^4!$ = 44.25
!!! Im plementation omplexity is very high !!! = 4 2^4 x $2^{44.25}$ = $2^{50.25}$ bits

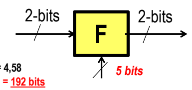


Sample bounds for implementing all possible block ciphers

Complexity growth: Engineering approach:

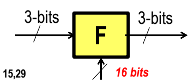
Example: ***t=2*, 2-bit to 2-bit mapping**

Number of all possible mappings = $2^{2x4} = 2^8$ mappings
Number of possible invertible mappings = $2^2\,!$ = 24
 # bits required to select all possible invertible mappings = $\log_2 2^2!$ = 4,58
Memory implementation complexity: $2^{4.58}$ $(2x2^2)$ = $2^{7.58}$ bits = 8 x 24 = 192 bits



Example: ***t=3*, 3-bit to 3-bit mapping**

Number of all possible mappings = $2^{3x8} = 2^{24}$ mappings
Number of possible invertible mappings = $2^3\,!$ = 40 320
 # bits required to select all possible invertible mappings = $\log_2 2^3!$ = 15,29
Memory implementation complexity: $2^{15.29}$ $(3x2^3)$ = $2^{19,87}$ bits

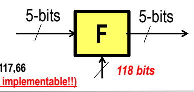


Example: ***t=4*, 4-bit to 4-bit mapping**

Memory implementation complexity:) = $2^{50.24}$ bits (see former page)

Example: ***t=5*, 5-bit to 5-bit mapping**

Number of all possible mappings = $2^{5x32} = 2^{160}$ mappings
Number of possible invertible mappings = $2^5\,!$ = 2,63 10^{35}
 # bits required to select all possible invertible mappings = $\log_2 2^5!$ = 117,66
Memory implementation complexity: $2^{117.66}$ $(5x2^5)$ = $2^{124,98}$ bits (not implementable!!)



Implementing all 2-to-2 bit Block-Crypto-Mappings

Implementation complexity in bits:

Example: **a** full **bijective function** from 2 bits to 2 bits

Truth table F_1	
00	10
01	00
10	11
11	01

Needs a ROM of 2×2^2 bits

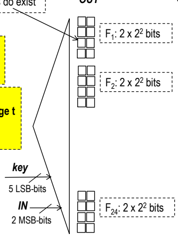
$2^2\,!$ = 24 such mappings do exist

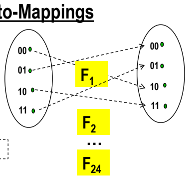
Total combinational bit complexity for a perfect cipher = $(\,2^t!\,) x\,t\,2^t$ bits for any t

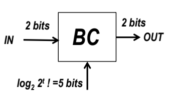
Approximate total combinational bit complexity for perfect block cipher s for large t
$\approx (\,2^{t\,2^t}\,)$ x $t\,2^t$ bits for large t

Selector needs $\log_2 24$ = 4,58 = 5 bits

Select one out of 24



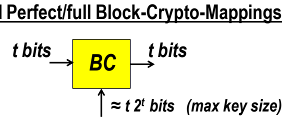




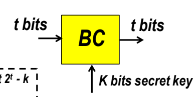
In general:
Memory complexity = **|F| x t 2^t** bits

memory complexity |F| x 2 2^t = 24 x 8 = 192 Bits

General Perfect/full Block-Crypto-Mappings (BC)



Practically usable secret-key cipher form:



|BC| = Number of all existing t-to-t bit ciphers ≈ $2^{t\,2^t - k}$ with k-bit secret-key size

Example : *number of existing ciphers of* ***64 to-64*** *bits with* ***64 bits key size*** ≈ ***$2^{2^{70}}$ ciphers***

a small fraction of them is known !!!
The lecture would discuss few of the most known standard ciphers

Widespread Standard Block Ciphers

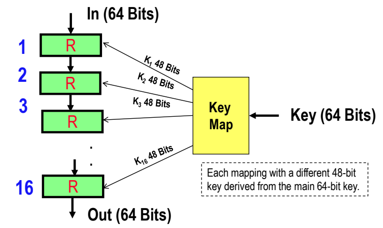
- **DES :** **D**ata **E**ncryption **S**tandard, IBM (NIST) 1976 (USA)
- **IDEA** (J. Massey and Lai) 1990 (Europe)
- **FEAL** NTT 1989 (Japan)
- **KASUMI** 1999 UMTS/3GPP (Mitsubishi Japan)
  to replace the broken *GSM A5 (Secret Stream Cipher)!*

. AES Advanced Encryption Standard (NIST):
New international standard **Rijndael** **Belgium (Oct. 2000)**

Many other ciphers were proposed ….

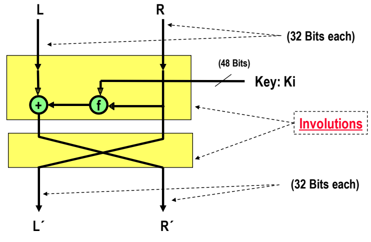
The First Standard Block-Cipher
The DES (Data Encryption Standard) Cipher (1976)

A mapping function **R** is repeated 16 times, each time with a different key



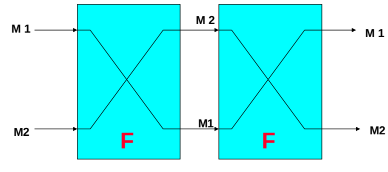
DES Feistel Round Structure R (Lucifer Late 1960)

(Horst Feistel, IBM) (1915 in Berlin; † 1990 USA)
(Migration 1934/1944 US citizen) Bsc MIT, Msc Harvard

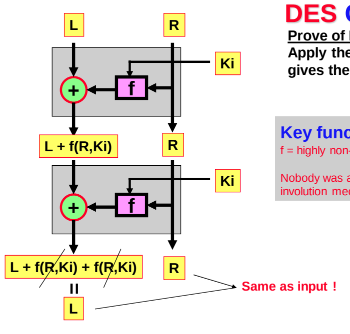


Involution

An involution: is a function that is self-inverting



F x F = 1 ⇒ $F = F^{-1}$



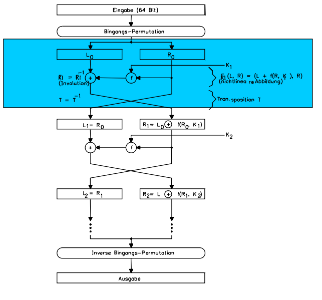
DES Core Involution

Prove of Involution:
Apply the same function two times gives the original input!

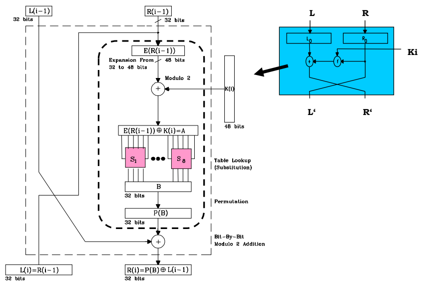
Key function is : L + f(R,Ki)
f = highly non-linear function !

Nobody was able since 1976 to break this involution mechanism in DES !

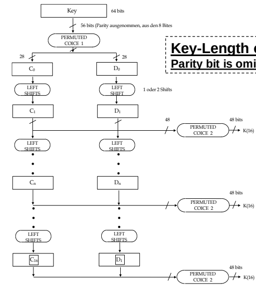
DES Cipher



DES Cipher- Core



DES Key Generation

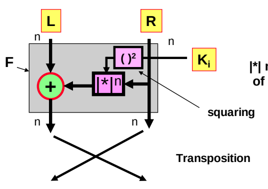


Key-Length only 56 bits!

Parity bit is omitted from each byte!

Simplified Example:

Given the following two mapping functions, **F** and transposition are to be used as a round operation in a block cipher.

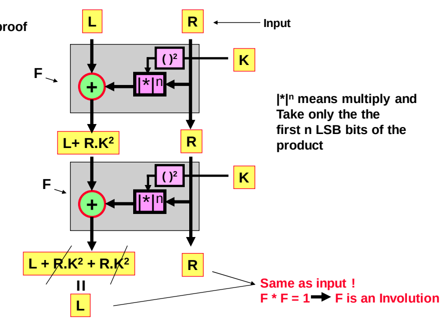


|*| means that only the first n- LSB bits of the product are taken

1. **Prove that the given function is an involution**
2. **Compute the cipher text Y = R1, L1 for an input R=9, L=11 using two rounds with the keys K_1=2, K_2=3. Take n=4 bits.**
3. **Decipher the cryptogram Y**

Solution :

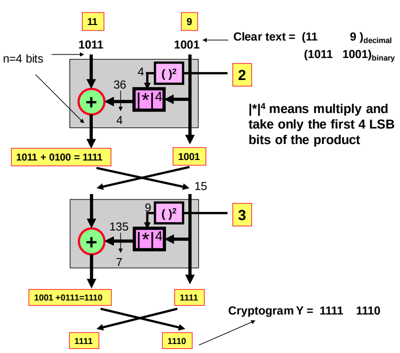
1. Involution proof



$|*|^n$ means multiply and Take only the the first n LSB bits of the product

Solution :

2. Encryption



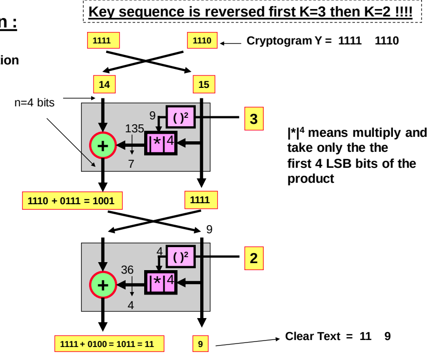
$|*|^4$ means multiply and take only the first 4 LSB bits of the product

Cryptogram Y = 1111 1110

Solution :

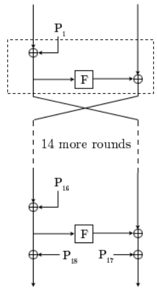
Key sequence is reversed first K=3 then K=2 !!!!

3. Decryption

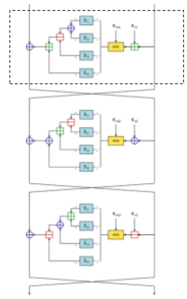


$|*|^4$ means multiply and take only the the first 4 LSB bits of the product

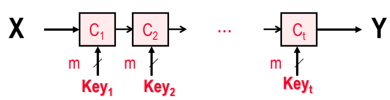
Blowfish Feistel Based Core Involution
1994
Bruce Schneier



CAST Feistel based Core Involution
1996
Carlisle Adams
& Stafford Tavares

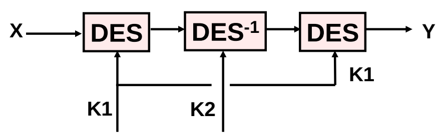


Increasing Security Strength by:
Cascading t-Ciphers
Or repeating the same cipher t-times



If the same Cipher with key length of m bits is cascaded t times.
The resulting key strength <u>is not</u> = t · m bits, But = (t-1) m bits
(due to meet-in-the middle attack technique!)

Triple-DES is a De Facto Standard in many Systems !!



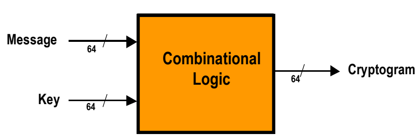
- 3 times cascaded DES results with a key strength of only 2 x DES key= 2x 56=112 bits
- IF K1=K2, the result is a single DES.

DES is *still* not broken !!
There is no proof that DES can not be broken !!

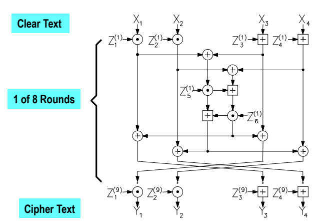
IDEA Cipher

1990 by Massey & Lai (ETH Zürich)

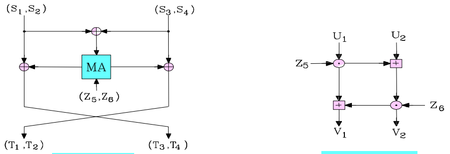
Swiss-Telecom



IDEA Round structure
Based on group operations



IDEA Involution



Involution

MA Function

Uses two group operations:

Multiplicative group : ⊙ Multiplication modulo $(2^{16} + 1)$

Additive group : ⊞ Addition modulo 2^{16}

Elements form a group under multiplication as the modulus is a prime number and every non-zero element has an inverse.

$(2^{16} + 1)$ is prime !

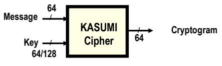
KASUMI

3rd Generation Mobile Cipher is

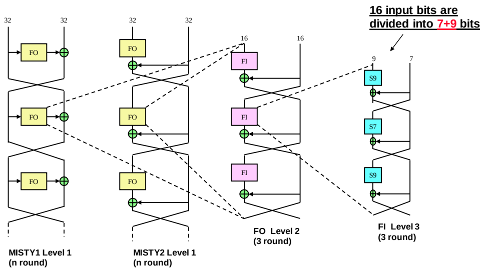
KASUMI Cipher to replace A5/1,2,..

UMTS/3GPP Standard (March 2000)

Original Cipher: Mitsubishi's " MISTY" 1997



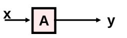
Recursive Structure of MISTY



16 input bits are divided into (7 + 9) bits resulting with two mappings: Mappings S7 and S9

Table of S7 over GF (2^7)

27, 50, 51, 90, 59, 16, 23, 84, 91, 26,114,115,107, 44,102, 73,
31, 36, 19,108, 55, 46, 63, 74, 93, 15, 64, 86, 37, 81, 28, 4,
11, 70, 32, 13,123, 53, 68, 66, 43, 30, 65, 20, 75,121, 21,111,
14, 85, 9, 54,116, 12,103, 83, 40, 10,126, 56, 2, 7, 96, 41,
25, 18,101, 47, 48, 57, 8,104, 95,120, 42, 76,100, 69,117, 61,
89, 72, 3, 87,124, 79, 98, 60, 29, 33, 94, 39,106,112, 77, 58,
1,109,110, 99, 24,119, 35, 5, 38,118, 0, 49, 45,122,127, 97,
80, 34, 17, 6, 71, 22, 82, 78,113, 62,105, 67, 52, 92, 88,125

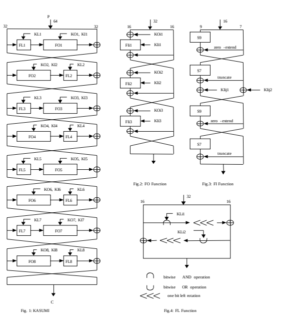


Mappings S7 and S9 generated by:

$Y = A(x^i)$ in GF(2^7) and GF(2^9)

Table of S9 over GF (2^9)

451,203,339,415,483,233,251, 53,385,185,279,491,307, 9, 45,211,
199,330, 55,126,235,356,403,472,163,286, 85, 44, 29,418,355,280,
331,338,466, 15, 43, 48,314,229,273,312,398, 99,227,200,500, 27,
1,157,248,416,365,499, 28,326,125,209,130,490,387,301,244,414,
467,221,482,296,480,236, 89,145, 17,303, 38,220,176,396,271,503,
231,364,182,249,216,337,257,332,259,184,340,299,430, 23,113, 12,
71, 88,127,420,308,297,132,349,413,434,419, 72,124, 81,458, 35,
317,423,357, 59, 66,218,402,206,193,107,159,497,300,388,250,406,
481,361,381, 49,384,266,148,474,390,318,284, 96,373,463,103,281,
101,104,153,336, 8, 7,380,183, 36, 25,222,295,219,228,425, 82,
265,144,412,449, 40,435,309,362,374,223,485,392,197,366,478,433,
195,479, 54,238,494,240,147, 73,154,438,105,129,293, 11, 94,180,
329,455,372, 62,315,439,142,454,174, 16,149,495, 78,242,509,133,
253,246,160,367,131,138,342,155,316,263,359,152,464,489, 3,510,
189,290,137,210,399, 18, 51,106,322,237,368,283,226,335,344,305,
327, 93,275,461,121,353,421,377,158,436,204, 34,306, 26,232, 4,
391,493,407, 57,447,471, 39,395,198,156,208,334,108, 52,498,110,
202, 37,186,401,254, 19,262, 47,429,370,475,192,267,470,245,492,
269,118,276,427,117,268,484,345, 84,287, 75,196,446,247, 41,164,
14,496,119, 77,378,134,139,179,369,191,270,260,151,347,352,360,
215,187,102,462,252,146,453,111, 22, 74,161,313,175,241,400, 10,
426,323,379, 86,397,358,212,507,333,404,410,135,504,291,167,440,
321, 60,505,320, 42,341,282,417,408,213,294,431, 97,302,343,476,
114,394,170,150,277,239, 69,123,141,325, 83, 95,376,178, 46, 32,
469, 63,457,487,428, 68, 56, 20,177,363,171,181, 90,386,456,468,
24,375,100,207,109,256,409,304,346, 5,288,443,445,224, 79,214,
319,452,298, 21, 6,255,411,166, 67,136, 80,351,488,289,115,382,
188,194,201,371,393,501,116,460,486,424,405, 31, 65, 13,442, 50,
61,465,128,168, 87,441,354,328,217,261, 98,122, 33,511,274,264,
448,169,285,432,422,205,243, 92,258, 91,473,324,502,173,165, 58,
459,310,383, 70,225, 30,477,230,311,506,389,140,143, 64,437,190,
120, 0,172,272,350,292, 2,444,162,234,112,508,278,348, 76,450



Structure of KASUMI

A slightly modified version of MISTY

AES

Advanced Encryption Standard

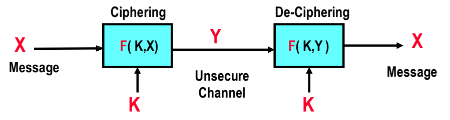
International Standard competition managed by NIST:
US National Institute of Science and Technology
1998-2001

Proposed 2001/2002 for 3G Mobile Authentication Functions

History

- A Standard Managed by the US National Institute of Science and Technology NIST
  - DES (1976) is an aging standard
  - Triple-DES: considered by NIST as a defacto-standard
- AES: Advanced Encryption Standard: finalized in 2001
  - Goal: define the Federal Information Processing Standard (FIPS)
  - AES candidate algorithms have to be:
    - Symmetric-key ciphers supporting 128, 192, and 256 bit keys
    - Royalty-Free and unclassified (i.e. public domain)
    - Available for worldwide export

Basic Secret Key Cipher Requirement



Fundamental requirement: Given the function *F*, it's infeasible to recover data *X* from ciphertext *y* without knowing the key *K*

AES Round-3 Finalist Algorithms

(finalized in 2001)

- MARS : IBM (USA)
- RC6 : R. Rivest (MIT), creator of the widely used RC4 (USA)
- Twofish : Counterpane Internet Security, Inc. (USA)
- Serpent : Ross Anderson, Eli Biham and Lars Knudsen (USA)
- Rijndael: Designed by J. Daemen and V. Rijmen (Belgium)

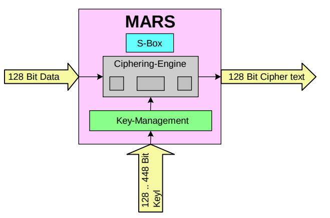
Joan Daemen (of Proton World International)

Vincent Rijmen (of Katholieke Universiteit Leuven).

MARS (IBM)

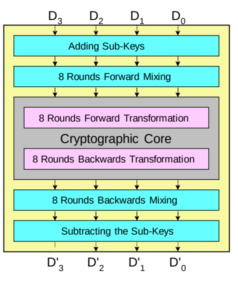
(AES Candidate: a round 3 finalist)

- IBM Candidate for AES
  - Block-Cipher
  - Optimized for 32 bit processors
  - All operations are 32 bits based.

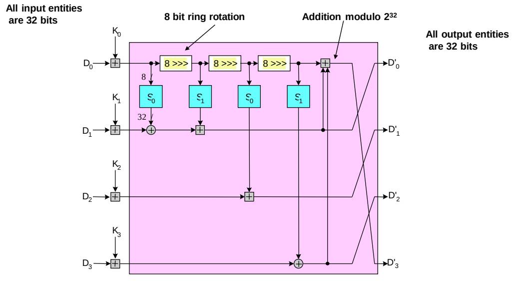


MARS

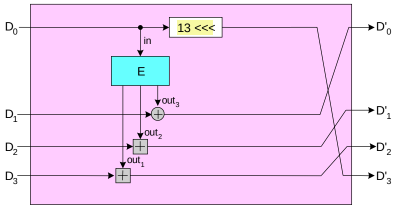
- MARS Ciphering process
  - Forward / Backwards Mixing
  - Forward / Backwards Transformation



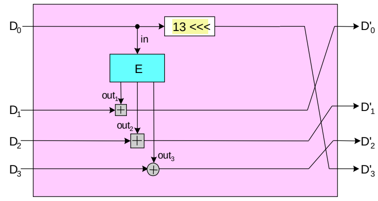
Forward Mixing



Cryptographic Core- Forward Transformation



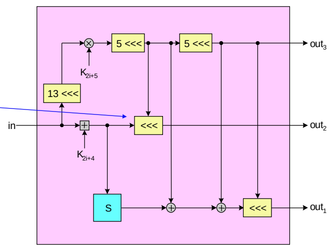
Cryptographic Core- Backward Transformation



Expansion Function E

Operations involve:

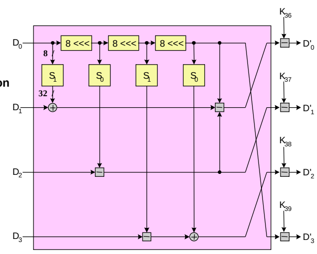
- Multiplications
- EX-OR's
- Additions
- 2 data dependent rotations



Backwards Mixing

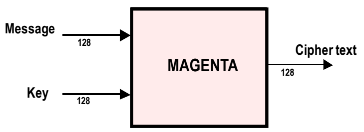
Operations involved:

- **8 Rounds**
- **One S-Box access per byte**
- **Round-dependent subtraction**
- **Word swapping**

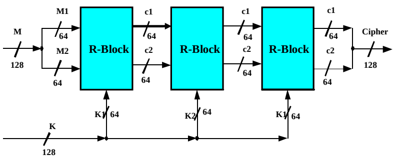


MAGENTA Block-Cipher

(Proposed for AES Competition by German Telecom)

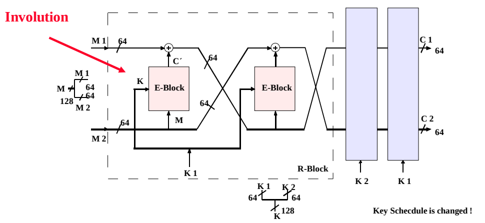


MAGENTA Block-Cipher

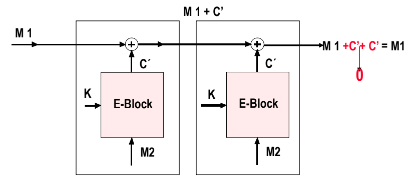


MAGENTA Block-Cipher

R-Block

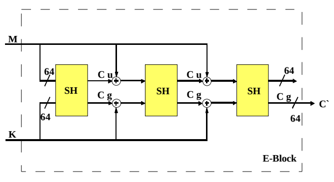


MAGENTA Involution



MAGENTA Block-Cipher

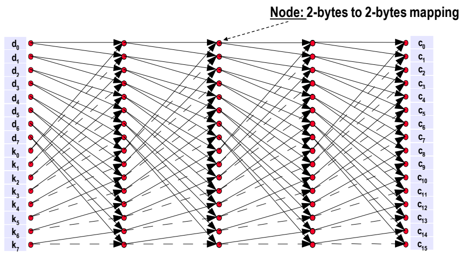
E-Block



MAGENTA Block-Cipher

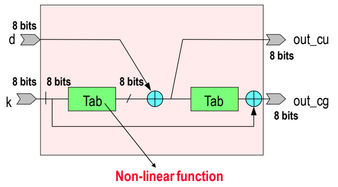
SH : Shuffle function.

(4-stages such that each key or data bit affect each cryptogram bit)



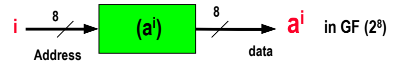
MAGENTA Block-Cipher

The Node Mapping



MAGENTA Block-Cipher

The TAB Mapping



a = primitive element in in GF (2^8)

Generator can be any irreducible polynomial over GF(2)

Exercise: Define a GF(2^8) arithmetic rules, find a primitive element and generate a possible TAB mapping for MAGENTA Cipher.

The AES Competion Winner Algorithm

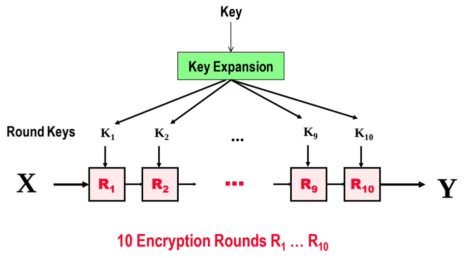
(Decision Oct. 2000)

The Rijndael Block Cipher

- By Joan Daemen (of Proton World International) and Vincent Rijmen (of Katholieke Universiteit Leuven).
- (pronounced "Rhine-doll")
- Allows only 128, 192, and 256-bit key sizes (unlike the other candidates)
- Variable block length of 128, 192, or 256 bits. All nine combinations of key/block length possible.
  - A block is the smallest data size the algorithm will encrypt
- speed improvement over DES

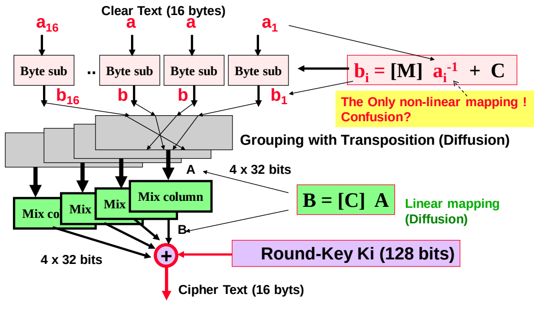
Page : 55

Rijndael: Basic concept



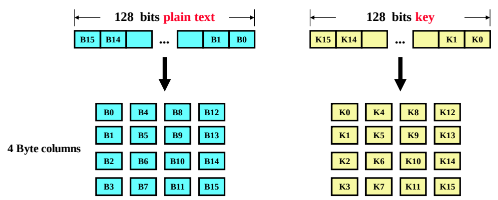
Page : 56

Rijndael AES: Basic Encryption Round Functions



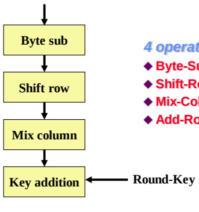
Page : 57

Rijndael: Data Format for 128-Bit Blocks



Page : 58

Rijndael: Encryption Round transformation

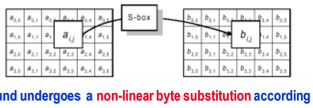


4 operations :

- Byte-Sub: non-linear 8 bit one-to-one mapping
- Shift-Row: Inverible row byte ring-shift
- Mix-Column: Invertible on columns with 4-bytes
- Add-Round-Key: Simply EXOR'ing with round key

Page : 59

Rijndael: ByteSub



Each byte at the input of a round undergoes a non-linear byte substitution according to the following transform:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

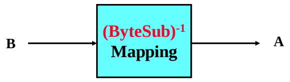
Non-singular matrix (Invertible matrix)

$$\mathbf{b} = [M]\ \mathbf{a}^{-1} + \mathbf{C}$$

Page : 60

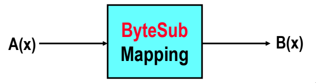
Rijndael: ByteSub and its inverse ByteSub^{-1}

$\mathbf{B} = [M]\ \mathbf{A}^{-1} + \mathbf{C_0}$



$\mathbf{A} = \{\ [M]^{-1}\ [\mathbf{B} + \mathbf{C_0}]\ \}^{-1}$

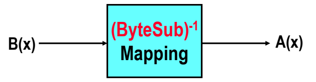
Rijndael: ByteSub and its inverse ByteSub^{-1}



Irreducible with period 51

[BytSub] : $B(x) = M(x) . A(x)^{-1} + C_0(x) \quad \text{mod} \ (1 + x + x^3 + x^4 + x^8)$

$1 + x + x^5 + x^6$

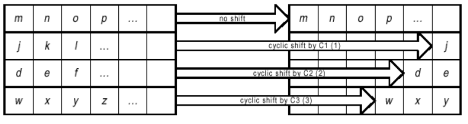


[BytSub]$^{-1}$: $A(x) = M(x)\ [B(x) + C_0(x)]^{-1} \quad \text{mod} \ (1 + x + x^3 + x^4 + x^8)$

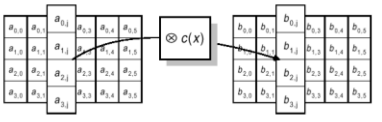
Rijndael: ShiftRow

Nb	C1	C2	C3
4	1	2	3
6	1	2	3
8	1	3	4

Depending on the block length, each "row" of the block is cyclically shifted according to the above table



Rijndael: MixColumn



Each 4 byte column is multiplied by a fixed polynomial

$C(x) = (03) . X^3 + (01) . X^2 + (01)\ X + (02)$

This corresponds to matrix multiplication $b(x) = c(x) \otimes a(x)$:

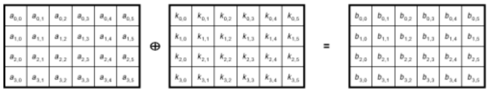
$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

GF(2^8) elements
8 bits each

B = [C] A

Non-singular matrix
(Invertible matrix)
Elements in HEX Format

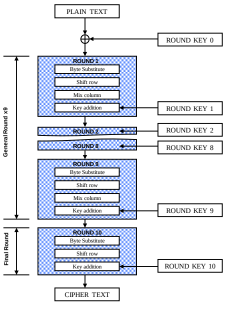
Rijndael: AddRoundKey



Each word is simply EXOR'ed with the expanded round key

Key Expansion algorithm see next

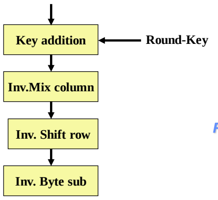
Rijndael Encryption



Inverse Rijndael Cipher

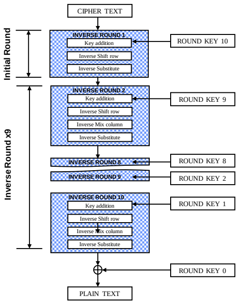
- Except for the non-linear ByteSub step, each part of Rijndael has a straightforward inverse and the operations simply need to be undone in the reverse order.
- However, Rijndael was specially written so that the same code that encrypts a block can also decrypt the same block simply by changing certain tables and polynomials for each layer. The rest of the operation remains identical.

Rijndael: Decryption Round transformation



Reversed sequence !

Rijndael Decryption



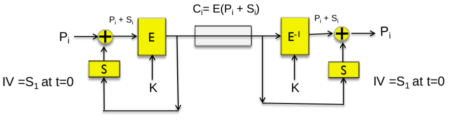
Hardware Implementations

- Rijndael performs very well in software
- Multiple S-Box engines, round-key EXORs, and byte shifts can all be implemented in hardware when speed is required
- Small amount of hardware can vastly speed up 8-bit implementations

Other
Block Ciphers Operation Modes
as
Running Key Generators

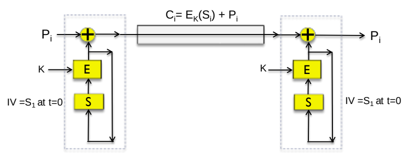
CBC Mode

Cipher Block Chaining Mode



KSG Mode

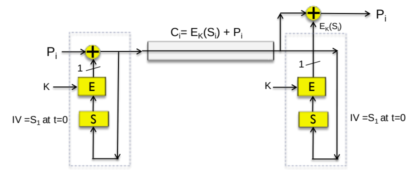
Key Stream Generator Mode



Self-Synchronizing OFB Mode

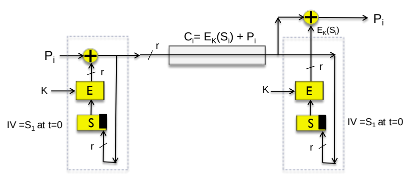
Single-Bit

(OFB: Output Feedback)



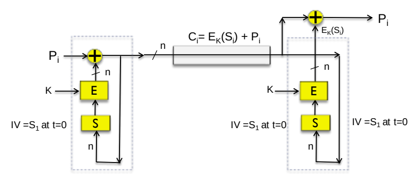
Self-Synchronizing OFB Mode

r-Bits

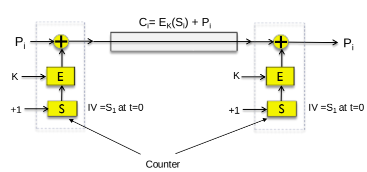


Self-Synchronizing OFB Mode

n-Bits

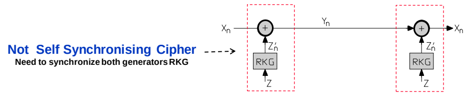


Counter Mode



Block Cipher in Self-Synchronizing Mode

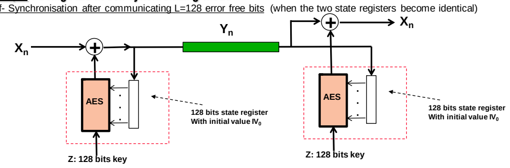
Operation Mode 3: in (Cipher Feedback Mode CFM)



Self Synchronising Running key Generator (Cipher Feedback Mode CFB)

Example : Using AES in self-synchronizing mode

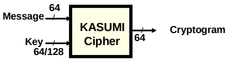
Self- Synchronisation after communicating L=128 error free bits (when the two state registers become identical)



3rd Generation Mobile Cipher is

KASUMI Cipher to replace A5/1,2,..

Used as key-stream generator in CBC mode



KASUMI (RKG) : Using CBC mode for 3rd Generation Mobile System (3GPP)