CRYPTOLOGY : CRYPTOGRAPHY + CRYPTANALYSIS

Cryptology = science of secrecy.
How :
encipher a **plaintext** into a **ciphertext** to protect its **secrecy**.
The recipient **deciphers** the ciphertext to recover the plaintext.
A **cryptanalyst** shouldn't complete a successful **cryptanalysis**.
Attacks [6] :

- **known ciphertext** : access only to the ciphertext
- **known plaintexts/ciphertexts** : known pairs (plaintext,ciphertext) ; search for the key
- **chosen plaintext** : known cipher, chosen cleartexts ; search for the key

Short history

J. Stern [8] : 3 ages :

- *craft age* : hieroglyph, bible, ..., renaissance, $\rightarrow$ WW2
- *technical age* : complex cipher machines
- *paradoxical age* : PKC

Evolves through maths' history, computing and cryptanalysis :

- manual
- electro-mechanical
- by computer

Polybius's square

Polybius, Ancient Greece : communication with torches

	1	2	3	4	5
1	a	b	c	d	e
2	f	g	h	ij	k
3	l	m	n	o	p
4	q	r	s	t	u
5	v	w	x	y	z

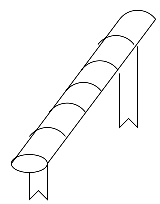
`TEXT` changed in `44,15,53,44`. Characteristics

- encoding letters by numbers
- shorten the alphabet's size

encode a character x over alphabet A in y finite word over B.
Polybius square : $\{a, \ldots, z\} \rightarrow \{1, \ldots, 5\}^2$.

History – ancient Greece

500 BC : *scytale* of Sparta's generals



Secret key : diameter of the stick

History – Caesar

Change each char by a char 3 positions farther
`A` becomes `d`, `B` becomes `e`...
The plaintext `TOUTE LA GAULE` becomes `wrxwh od jdxoh`.

Why enciphering ?

- Yesterday :
  - for strategic purposes
    (the enemy shouldn't be able to read messages)
  - by the church
  - diplomacy
- Today, with our numerical environment
  - confidentiality
  - integrity
  - authentication

Goals of cryptology

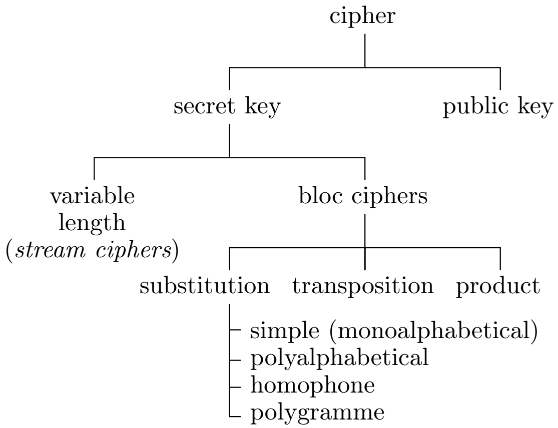
Increasing number of goals :

- *secrecy* : an enemy shouldn't gain access to information
- *authentication* : provides evidence that the message comes from its claimed sender
- *signature* : same as auth but for a third party
- *minimality* : encipher only what is needed.

The tools

- *Information Theory* : perfect cipher
- *Complexity* : most of the ciphers just ensure computational security
- *Computer science* : all make use of algorithms
- *Mathematics* : number theory, probability, statistics, algebra, algebraic geometry,...

Ciphers Classification



Symmetrical ciphers

Made of [1] :

- plaintext alphabet : $\mathcal{A}_{\mathcal{M}}$
- ciphertext alphabet : $\mathcal{A}_{\mathcal{C}}$
- keys alphabet : $\mathcal{A}_{\mathcal{K}}$
- encipher ; application $E : \mathcal{A}_{\mathcal{K}}^{\star} \times \mathcal{A}_{\mathcal{M}}^{\star} \to \mathcal{A}_{\mathcal{C}}^{\star}$;
- decipher ; application $D : \mathcal{A}_{\mathcal{K}}^{\star} \times \mathcal{A}_{\mathcal{C}}^{\star} \to \mathcal{A}_{\mathcal{M}}^{\star}$

E and D are such that $\forall K \in \mathcal{A}_{\mathcal{K}}^{\star}$, $\forall M \in \mathcal{A}_{\mathcal{M}}^{\star}$:

$$D(K, E(K, M)) = M$$

Monoalphabetical ciphers

Monoalphabetical cipher : bijection between letters from $\mathcal{A}_M$ and $\mathcal{A}_C$. If both alphabets are identical : permutation.

Example : Caesar. {a,. . . ,z}≡{A,. . . ,Z} $\equiv \{0, \ldots, 25\} = \mathbb{Z}_{26}$
Caesar cipher is **additive**.
Encipher : $\forall x \in \mathbb{Z}_{26}$, $x \mapsto x + 3 \mod 26$
Decipher : $\forall y \in \mathbb{Z}_{26}$, $y \mapsto y - 3 \mod 26$

Multiplicative cipher

We consider : $x \mapsto t \cdot x \mod 26$ for $t \in \mathbb{N}$.
Acceptable values of t are s.t. $\gcd(t, 26) = 1 \Leftrightarrow t \nmid 26$.
$\varphi(26)$ acceptables values $\{1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25\}$
Other values don't ensure the uniqueness of the deciphering (e.g. 2)

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12
n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25
0	2	4	6	8	10	12	14	16	18	20	22	24

To decipher, we require the existence of t^{-1} modulo 26.
We use the extended Euclidean algorithm which provides Bezout coefficients i.e. $x, y \in \mathbb{N}$ st. $d = \gcd(a, b) = ax + by$.
From Bezout coefficients, one can deduce t^{-1} modulo 26 :

$$\gcd(t, 26) = 1 \Leftrightarrow \exists x, y \in \mathbb{N} : tx + 26y = 1 \Leftrightarrow x \equiv t^{-1} \mod 26$$

Iterative computation

Extended Euclidean(q, r) with $q < r$
    Q$\leftarrow (1, 0)$;
    R$\leftarrow (0, 1)$;
    while $r \neq 0$ **do**
        $t \leftarrow q \mod r$;
        $T \leftarrow Q - \lfloor q/r \rfloor R$;
        $(q, r) \leftarrow (r, t)$;
        $(Q, R) \leftarrow (R, T)$;
    end
    return (q, Q); q : gcd value and Q provides the coeffs.
end

Affines Ciphers

When combining 26 additive ciphers and 12 multiplicative ones, we get **affine** ciphers :
given s and $t \in \mathbb{N}$, encipher with : $x \mapsto (x + s) \cdot t \mod 26$.
The key is the pair (s, t) and the deciphering is done by applying successively the previous methods.

There are 26.12=312 possible affine ciphers. Far from the 26 !=403291461126605635584000000 possible ones.

Extended Euclidean $(11, 26)$

q	r	t	Q	$\lfloor q/r \rfloor$	R	T
11	26	11	$(1, 0)$	0	$(0, 1)$	$(1, 0)$
26	11	4	$(0, 1)$	2	$(1, 0)$	$(-2, 1)$
11	4	3	$(1, 0)$	2	$(-2, 1)$	$(5, -2)$
4	3	1	$(-2, 1)$	1	$(5, -2)$	$(-7, 3)$
3	1	0	$(5, -2)$	3	$(-7, 3)$	$(26, -11)$
1	0		$(-7, 3)$		$(26, -11)$	

pgcd$(11, 26) = 1$ and Bezout's coefficients are $(-7, 3)$.
The mult. inverse of $11 \mod 26 = -7 = 19$.

Ciphers defined by keyword

To get all possible monoalphabetical ciphers by :

- a keyword like, for instance `CRYPTANALYSIS`;
- a key letter like `e`.

Remove multiple occurrences of the same letter in the keyword -here `CRYPTANLSI`- then

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

Cryptanalysis

Shannon : a small proportion of letters provides more information than the remaining 2/3 of the text.

By applying a frequency analysis on the letters then of bigrams, ... in the ciphertext.

Conclusion

Monoalphabetical ciphers aren't robust against a frequency analysis.

We need ciphers for which the statistical distribution of the letters tend to be a uniform one.

1.st attempt : use a crypto transformation which associates a set of distinct letters in the ciphertext to the plaintext letters.

We get what is called **polyalphabetical ciphers**

Solving $ax \equiv b \mod n$

We have used the method for solving the integer equation $ax \equiv b \mod n$. There are two cases :

- $\gcd(a, n) = 1$: $ax \equiv b \mod n \Leftrightarrow x \equiv a^{-1}b \mod n$ with a^{-1} given by the extended Euclidean algorithm.
- $\gcd(a, n) = d \neq 1$ splits into two new cases :
  - $d \nmid b$, the equation has no solution ;
  - $d|b$ $ax \equiv b \mod n \Leftrightarrow da'x \equiv db' \mod dn'$. We divide lhs and rhs by d and we solve $a'x \equiv b' \mod n'$. We get a set of solutions : $\{x = a'^{-1}b' + kn' : 0 \leq k < d\}$.

Vigenère's cipher (1586)

In a **polyalphabetical cipher**, plaintext characters are transformed by means of a key $K = k_0, \ldots, k_{j-1}$ which defines j distinct functions $f_0, \ldots, f_{j-1}$ s.t.

$$\forall i,\ 0 < j \leq n \quad f_{k_l} : \mathcal{A}_M \mapsto \mathcal{A}_C, \forall l,\ 0 \leq l < j$$
$$c_i = f_{k_{i \mod j}}(m_i)$$

Idea : use j distinct monoalphabetical ciphers.

Vigenère's square

```
abcdefghijklmnopqrstuvwxyz abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ NOPQRSTUVWXYZABCDEFGHIJKLM
BCDEFGHIJKLMNOPQRSTUVWXYZA OPQRSTUVWXYZABCDEFGHIJKLMN
CDEFGHIJKLMNOPQRSTUVWXYZAB PQRSTUVWXYZABCDEFGHIJKLMNO
DEFGHIJKLMNOPQRSTUVWXYZABC QRSTUVWXYZABCDEFGHIJKLMNOP
EFGHIJKLMNOPQRSTUVWXYZABCD RSTUVWXYZABCDEFGHIJKLMNOPQ
FGHIJKLMNOPQRSTUVWXYZABCDE STUVWXYZABCDEFGHIJKLMNOPQR
GHIJKLMNOPQRSTUVWXYZABCDEF TUVWXYZABCDEFGHIJKLMNOPQRS
HIJKLMNOPQRSTUVWXYZABCDEFG UVWXYZABCDEFGHIJKLMNOPQRST
IJKLMNOPQRSTUVWXYZABCDEFGH VWXYZABCDEFGHIJKLMNOPQRSTU
JKLMNOPQRSTUVWXYZABCDEFGHI WXYZABCDEFGHIJKLMNOPQRSTUV
KLMNOPQRSTUVWXYZABCDEFGHIJ XYZABCDEFGHIJKLMNOPQRSTUVW
LMNOPQRSTUVWXYZABCDEFGHIJK YZABCDEFGHIJKLMNOPQRSTUVWX
MNOPQRSTUVWXYZABCDEFGHIJKL ZABCDEFGHIJKLMNOPQRSTUVWXY

polyalphabetique KSYSSGTUUTZXVKMZ
VENUSVENUSVENUSV
```

Cryptanalysis...

... becomes more difficult : we tend to a uniform distribution.

But, if we re-arrange the ciphertext in a matrix with as many columns as the key length, all the letters in the same column come from the same monoalphabetical cipher.

Cryptanalysis works as follows :
(1) find the key length
(2) apply the previous methods

2 tests to find the key length : Kasiski and Friedman.

Homophone Ciphers

<u>Goal</u> : smooth the frequency distribution of the letters.
The ciphertext alphabet contains several equivalents for the same plaintext letter.
We thus define a multiple representation substitution.
Thus, letter e from the plaintext, instead of being always enciphered by a 4 could be replaced for instance by 37, 38, 39, ...
These different **cryptographic units** corresponding to the same plaintext character are called **homophones**.

letter	frequency	letter	frequency
a	0,26,27,28,29,30	n	13,68,69,70,71,72
b	1	o	14,73,74,75,76
c	2,31,32,33,34	p	15,77,78
d	3,35,36	q	16
e	4,37,. . . ,54	r	17,79,80,81,82
f	5,55	s	18,83,84,85,86,87
g	6,56	t	19,88,89,90,91,92,93
h	7,57	u	20,94,95,96,97
i	8,58,59,60,61,62	v	21
j	9	w	22
k	10	x	23
l	11,63,64,65,66	y	24,98
m	12,67	z	25

Transposition

Implements a permutation of the plaintext letters $\mathcal{A}_C = A_M$.

$$\forall i, \quad 0 \leq i < 0 \quad \begin{array}{l} f : \mathcal{A}_M \to \mathcal{A}_M \\ \eta : \mathbb{Z}_n \to \mathbb{Z}_n \\ c_i = f(m_i) = m_{\eta(i)} \end{array}$$

Simple array transposition

Given a passphrase, we define a numerical key :

T	R	A	N	S	P	O	S	I	T	I	O	N	S	I	M	P	L	E
18	14	1	8	15	12	10	16	3	19	4	11	9	17	5	7	13	6	2

We encipher, «*le chiffrement est l'opération qui consiste à transformer un texte clair, ou libellé, en un autre texte inintelligible appelé texte chiffré ou chiffré*» [5].

18	14	1	8	15	12	10	16	3	19	4	11	9	17	5	7	13	6	2
l	e	c	h	i	f	f	r	e	m	e	n	t	e	s	t	l	o	p
é	r	a	t	i	o	n	q	u	i	c	o	n	s	i	s	t	e	à
t	r	a	n	s	f	o	r	m	e	r	u	n	t	e	x	t	e	c
l	a	i	r	o	u	l	i	b	e	l	l	é	e	n	u	n	a	u
t	r	e	t	e	x	t	e	i	n	i	n	t	e	l	l	i	g	i
b	l	e	a	p	p	e	l	é	t	e	x	t	e	c	h	i	f	f
r	é	o	u	c	r	y	p	t	o	g	r	a	m	m	e			

Vernam cipher (1917)

Is the one-time pad a «perfect» cipher ?

A and B share a true random sequence of n bits : the secret key K.
A enciphers M of n bits in $C = M \oplus K$.
B deciphers C by $M = K \oplus C$.

Example

$M = 0011$, $K = 0101$
$C = 0011 \oplus 0101 = 0110$
$M = K \oplus C$.

Non-reusability : for every new message, we need a new key.

Why a new key ?

... To avoid revealing information on the $\oplus$ of plaintexts.

Eve can sniff $C = \{M\}_K$ and $C' = \{M'\}_K$ and computes :

$$C \oplus C' = (M \oplus K) \oplus (M' \oplus K) = M \oplus M'$$

Given enough ciphertexts, she's able to recover a plaintext by a frequency analysis and with the help of a dictionnary [4].

If we respect the above requirements, Vernam cipher guarantees the condition of perfect secrecy.

Condition (perfect secrecy)

$$Pr(M = m \mid C = c) = Pr(M = m)$$

Intercepting C doesn't reveal any information to the cryptanalyst

Why is it secure ?

Vernam ciphers provides **perfect secrecy**.
We have three classes of information :

- plaintexts M with proba. distribution $Pr(M)/\sum_M Pr(M) = 1$
- ciphertexts C with proba. distribution $Pr(C)/\sum_C Pr(C) = 1$
- keys with proba. distribution $Pr(K)$ s.t. $\sum_K p(K) = 1$

$Pr(M \mid C)$ = proba that M has been sent knowing that C was received (C is the corresponding ciphertext of M). The perfect secrecy condition is defined as

$$Pr(M \mid C) = Pr(M)$$

The interception of the ciphertext does not provide any information to the crypto-analyst.

Conclusion

Perfect secrecy but difficult to achieve

- generate truly random sequences
- store them and share them with the recipients

example of use : «red phone».

Product and iterated ciphers

Improvement : combine substitutions and transpositions

A cipher is **iterated** if the ciphertext is obtained from repeated applications of a round function to the plaintext
At each round, we combine a round key with the plaintext.

Definition
*In an iterated cipher with r rounds, the ciphertext is computed by repeated applications of a **round function** g to the plaintext :*

$$C_i = g(C_{i-1}, K_i) \quad i = 1, \ldots, r$$

C_0 the plaintext, K_i round key and C_r the ciphertext.
Deciphering is achieved by inverting the previous equation. For a fixed K_i, g must be invertible.

Special case, **Feistel ciphers**.

Feistel ciphers

A **Feistel cipher** with block size $2n$ and r rounds is defined by :

$$g : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n \times \{0,1\}^n$$

$$X, Y, Z \mapsto (Y, F(Y, Z) \oplus X)$$

g function of $2n \times m$ bits into $2n$ bits and $\oplus$ denoting the n bit XOR
Operation mode
Given a plaintext $P = (P^L, P^R)$ and r round keys $K_1, \ldots, K_r$, the ciphertext (C^L, C^R) is obtained after r rounds.
Let $C_0^L = P^L$ and $C_0^R = P^R$ and we compute for $i = 1, \ldots, r$

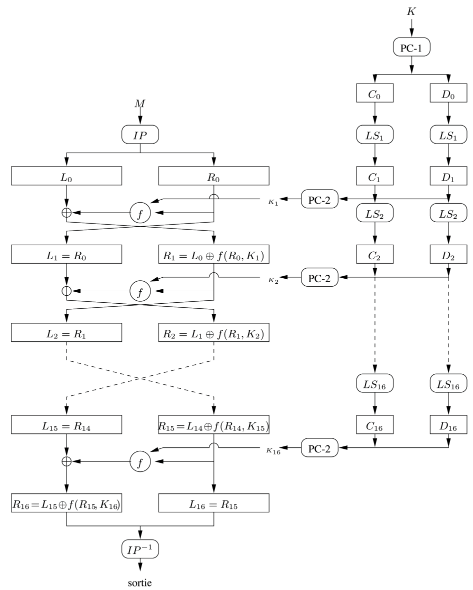
$$(C_i^L, C_i^R) = (C_{i-1}^R, F(C_{i-1}^R, K_i) \oplus C_{i-1}^L)$$

with $C_i = (C_i^L, C_i^R)$ and $C_r^R = C^L$ and $C_r^L = C^R$
The round keys $K_1, \ldots, K_r$, are obtained by a key scheduling algorithm on a master key K.

DES

- NBS launches a competition in 1973
- DES (*Data Encryption Standard*) proposed by IBM in 1975
- adopted in 1977
- security evaluation every 4 years
- replaced by AES or Rijndael [2]
- enciphering example of DES in STINSON's book [9]

DES usage
DES was (is ?) widely used (banks, computer security systems with DES as the main component).
Feistel cipher with special properties.



Operation

DES receives as an input :

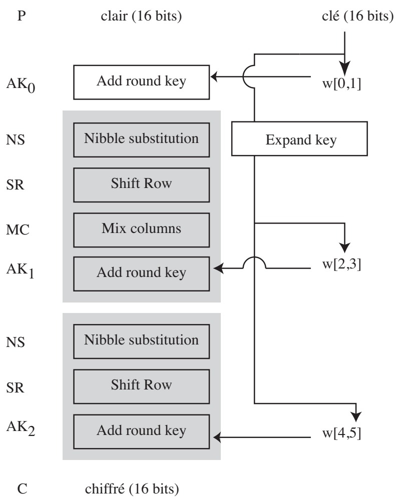
- a message M of 64 bits ;
- a key K of 56 bits.

and outputs a ciphertext of 64 bits.
DES algorithms first applies to M an initial permutation IP which provides M', a perutation of M.
M' is then cut into two 32 bits words :

- L_0 the left part of M'
- R_0 its right part.

DES then applies 16 iterates of function f combining substitutions and transpositions.

AES or Rijndael[3]



Operations in $GF(16) \simeq \mathbb{F}_2[x]/x^4 + x + 1$

- $m(x) = x^4 + x + 1$ is an irreducible of $\mathbb{F}_2$
- elements : nibble $b_0 b_1 b_2 b_3 \overset{\theta}{\leftrightarrow} b_0 x^3 + b_1 x^2 + b_2 x + b_3$
- addition : by adding the coefficients : $(x^3 + x + 1) + (x^2 + 1)$
- multiplication : product of polynomials $\mod m(x)$
- byte encoding : in a quadratic extension $\mathbb{F}_{16}[z]/z^2 + 1$
- beware ! $z^2 + 1$ is not invertible in $GF(16)$
- Reminder : to find the multiplicative inverse of an element : Extended Euclidean on polynomials $(x + 1, m) = (x^3 + x^2 + x)(x + 1) + 1 \cdot m$

State matrix & nibbles

- 1 nibble = 4 bits word (I/O of SAES components)

$$\begin{array}{c|c} b_0 b_1 b_2 b_3 & b_8 b_9 b_{10} b_{11} \\ \hline b_4 b_5 b_6 b_7 & b_{12} b_{13} b_{15} b_{15} \end{array} = \begin{array}{c|c} S_{0,0} & S_{0,1} \\ \hline S_{1,0} & S_{1,1} \end{array}$$

- key representation :

$$\underbrace{k_0 k_1 \ldots k_7}_{w[0]} \quad \underbrace{k_8 \ldots k_{15}}_{w[1]}$$

Inverses in $\mathbb{F}_{16}$

1	0001	1	1	0001	1
2	0010	x	$x^3 + 1$	1001	9
3	0011	$x + 1$	$x^3 + x^2 + x$	1110	e
4	0100	x^2	$x^3 + x^2 + 1$	1101	d
5	0101	$x^2 + 1$	$x^3 + x + 1$	1011	b
6	0110	$x^2 + x$	$x^2 + x + 1$	0111	7
7	0111	$x^2 + x + 1$	$x^2 + x$	0110	6
8	1000	x^3	$x^3 + x^2 + x + 1$	1111	f
9	1001	$x^3 + 1$	x	0010	2
a	1010	$x^3 + x$	$x^3 + x^2$	1100	c

S-box used in Nibble substitution

$i\downarrow$	00	01	10	11
00	9	4	a	b
01	d	1	8	5
10	6	2	0	3
11	c	e	f	7

=

1001	0100	1010	1011
1101	0001	1000	0101
0110	0010	0000	0011
1100	1110	1111	0111

- $\forall$ nibble : $\underbrace{b_0 b_1}_{\text{row}} \underbrace{b_2 b_3}_{\text{column}}$: 00 01 $\overset{S}{\to}$ 01 00
- $$\begin{array}{c|c} 0001 & 0001 \\ \hline 1100 & 1110 \end{array} \overset{S}{\to} \begin{array}{c|c} 0100 & 0100 \\ \hline 1100 & 1111 \end{array} = \begin{array}{c|c} 4 & 4 \\ \hline c & f \end{array}$$

Other transformations

- Shift row : transposition of the nibble bits : $b_0 b_1\, b_2 b_3 \mapsto b_2 b_3\, b_0 b_1$. $\begin{array}{c|c} 4 & 4 \\ \hline c & f \end{array} \mapsto \begin{array}{c|c} 4 & 4 \\ \hline f & c \end{array}$
- Mix columns : modifies the polynomial representation of the state's rows $\begin{array}{c|c} N_i & . \\ \hline N_j & . \end{array}$; we associate $c(z) = N_i z + N_j \in \mathbb{F}_{16}[z]/z^2 + 1$; compute $c(z).(x^2 z + 1) \mod z^2 + 1$.

Example

For $4f \leftrightarrow 0100\ 1111 \mapsto c(z) = x^2 z + x^3 + x^2 + x + 1$: $(x^3 + x^2 + 1)z + (x^3 + x^2) = N_k z + N_\ell \leftrightarrow 1101\ 1100$ because $z^2 = 1, x^4 = x + 1$ and $x^5 = x^2 + x$.

S-box algebraically

1. init the S-box with the nibbles arranged in a 1D array row by row
2. convert each nibble in a polynomial
3. invert each nibbble in $\mathbb{F}_{16}$
4. associate to the inverse its ploynomial in $\mathbb{F}_{16}[y]/y^4 - 1 = N(y)$
5. compute $a(y)N(y) + b(y) \mod y^4 + 1$ with $a = y^3 + y + 1$ et $b = y^3 + 1$

Normally $S(0011) = 1011 \equiv S(3) = b$

Mix columns (matrix)

We work directly on the state :

$$\begin{pmatrix} 1 & x^2 \\ x^2 & 1 \end{pmatrix} \cdot \begin{pmatrix} S_{0,0} & S_{0,1} \\ S_{1,0} & S_{1,1} \end{pmatrix} =_{\mathbb{F}_{16}} \begin{pmatrix} 1 & 4 \\ 4 & 1 \end{pmatrix} \cdot \begin{pmatrix} S_{0,0} & S_{0,1} \\ S_{1,0} & S_{1,1} \end{pmatrix}$$

Example

$$\begin{pmatrix} 1 & x^2 \\ x^2 & 1 \end{pmatrix} \cdot \begin{pmatrix} x^2 & x^2 \\ x^3+x^2+x+1 & x^3+x^2 \end{pmatrix} = \begin{pmatrix} x^3+x^2+1 & 1 \\ x^3+x^2 & x^3+x^2+x+1 \end{pmatrix} = \begin{pmatrix} d & 1 \\ c & f \end{pmatrix}$$

Key scheduling

- initialisation : $w[0] = k_0 \dots k_7 \qquad w[1] = k_8 \dots k_{15}$
- $2 \leq i \leq 5$

$$\begin{cases} w[i] = w[i-2] \oplus \text{RCON}(i/2) \oplus \text{SubNib}(\text{RotNib}(w[i-1])) & i \text{ even} \\ w[i] = w[i-2] \oplus w[i-1] & i \text{ odd} \end{cases}$$

With

- RCON[i]=RC[i]0000
- $\text{RC}[i] = x^{i+2} \in \mathbb{F}_{16}$ ($\text{RC}[1]=x^3 \leftrightarrow 1000$)
- RotNib($N_0 N_1$)=$N_1 N_0$
- SubNib($N_0 N_1$)=$S(N_0)S(N_1)$ where S denotes the S-box

Example

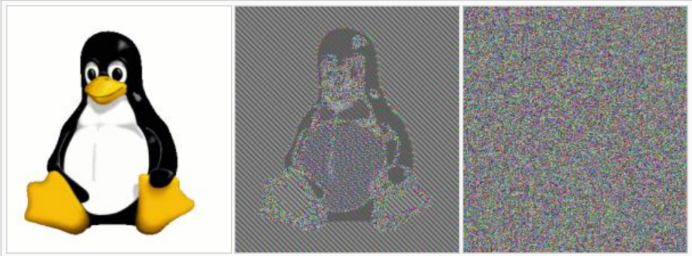
with $w[0]w[1] = 0101\ 1001\ 0111\ 1010$, we have
$w[2] = 1101\ 1100$, $w[3] = 1010\ 0101$, $w[4] = 0110\ 1100$ and
$w[5] = 1100\ 1010$

Why SAES ?

- introduced in [7] for academic purposes
- simpler than AES and can be used by hand
- allows to illustrate cryptanalysis
- has all the features of AES

Block ciphers modes of operation

Modes of operation pictured



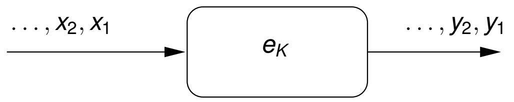
Original image — Encrypted using ECB mode — Modes other than ECB result in pseudo-randomness

The image on the right is how the image might appear encrypted with CBC, CTR or any of the other more secure modes—indistinguishable from random noise. Note that the random appearance of the image on the right does not ensure that the image has been securely encrypted; many kinds of insecure encryption have been developed which would produce output just as "random-looking".

http://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

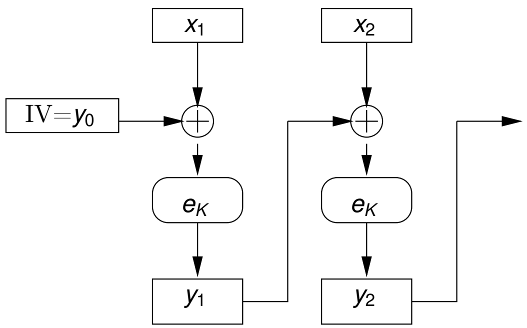
ECB : electronic codebook mode

The one previously used ; given a plaintext, each block x_i is enciphered with the key K, and provides the ciphertext $y_1 y_2 \ldots$

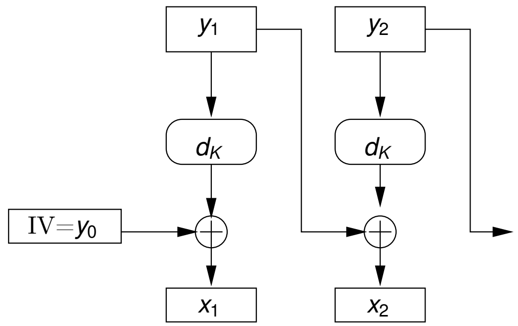


CBC : cipher block chaining mode

Each ciphertext y_i is XORed with next plaintext x_{i+1}



CBC – Deciphering

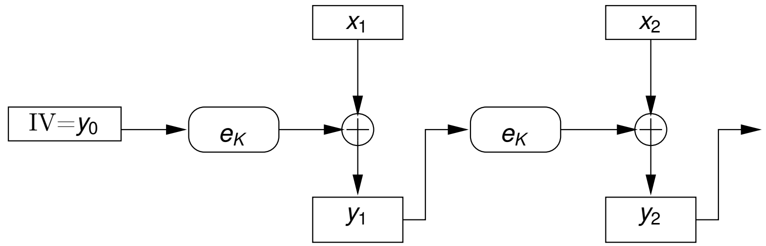


OFB (output feedback mode) and CFB (cipher feedback mode)

Encipher each plaintext block by successive XORing with keys coming from the application of a secret key cipher :

- **OFB** : sequence of keys comes from the repeated enciphering started on an initial value IV. We let z_0=IV and we compute the sequence $z_1 z_2 \ldots$ by $z_i = e_K(z_{i-1})$. The plaintext is then enciphered by $y_i = x_i \oplus z_i$
- **CFB** : We start with y_0=IV and the next key is obtained by enciphering the previous ciphertext $z_i = e_K(y_{i-1})$. Otherwise, everything works like in OFB mode.

CFB enciphering

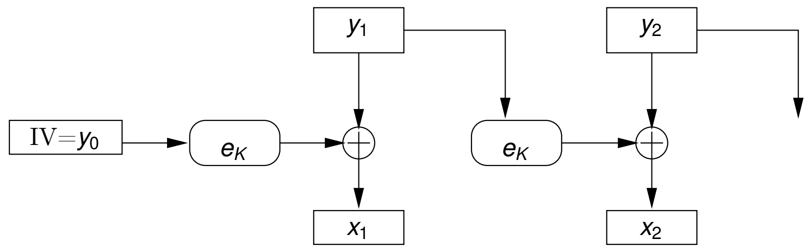


MAC-MDC

For Message Authentication Code (Modification Detection Code), or message fingerprint (MAC=MDC+IV$\neq$ 0).

Possible with CBC and CFB.
We start with IV=0. We build the ciphertext $y_1 \dots y_n$ with the key K in CBC mode. MAC is the last block y_n.
Alice sends the message $x_1 \dots x_n$ and the MAC y_n.
Upon reception of $x_1 \dots x_n$, Bob builds $y_1 \dots y_n$ by using the secret key K and verifies that y_n is the same than the received MAC.

CFB deciphering



G. Brassard.
Cryptologie contemporaine.
Logique, mathématiques, informatique. Masson, 1993.

J. Daemen and V. Rijmen.
AES proposal : Rijndael.
Technical report, Katholieke Universiteit Leuven, 1999.

J. Daemen and V. Rijmen.
The Rijndael bloc cipher.
Technical report, AES proposal, 1999.

E Dawson and L Nielsen.
Automated cryptanalysis of xor plaintext strings.
Cryptologia, XX(2) :165–181, May 1996.

D. Kahn.
La guerre des codes secrets.
InterEditions, 1980.

R.L. Rivest.
Cryptography.
In *Handbook of Theoretical Computer Science*, volume A, chapter 13. Elsevier, 1990.

W. Stallings.
Cryptography and Network Security.
Prentice-Hall, 4th. edition, 2006.

J. Stern.
La science du secret.
Odile Jacob, 1998.

D. Stinson.
Cryptographie, théorie et pratique.
International Thomson Publishing, 1995.