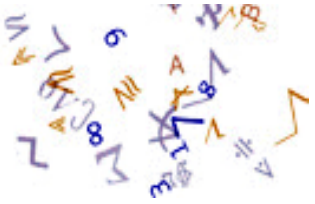


Introduction to Cryptography

Philippe Jaillon, ISI Department
H. Fayol Institute - ENSM-SE
philippe.jaillon@emse.fr

Cryptology: the science of secrecy

- cryptography: Secret writing
  - Methods to protect our secrets.
  - We have the keys
- cryptanalysis: Analysis
  - Look for the weaknesses,
  - Obtain the keys
  - Read the message without having the keys,

Objectives of cryptography

Ensure the confidentiality of communications

+ New requirements
  - Authentication: sign messages
  - Non-repudiation: Obtain proof of actions
  - Integrity: Seal information

Usable, Efficient and Secure

Kerckhoffs's principle (1883)

- The system must be practically, if not mathematically, indecipherable;
- It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience;
- Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents;
- It must be applicable to telegraphic correspondence;
- It must be portable, and its usage and function must not require the concourse of several people;
- Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe.

Auguste Kerckhoffs, "La cryptographie militaire", Journal des sciences militaires, vol. IX, pp. 5-38, Janvier 1883, pp. 161-191, Férier 1883.

Quick history of cryptography

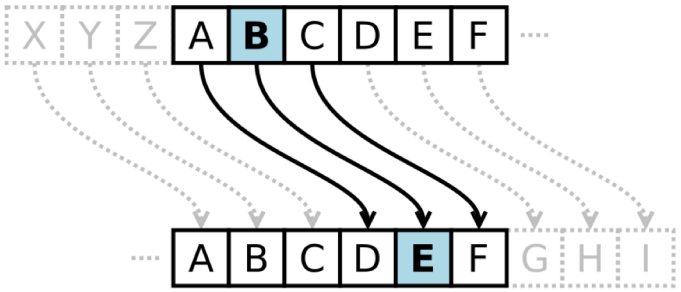
- Two great times
  - Before computers
    - Classical methods
    - Mechanization at the beginning of the 20th century
  - From computers
    - Transposition of classical methods
    - Automating
    - New possibilities

Quick history of cryptography

- Two clearly differentiated areas:
  - secret key cryptography
    - ➢Symmetrical / classical
  - public key cryptography
    - ➢Asymmetric / modern.

Shift cipher

- Simply shift message's letters by fixed positition in the alphabet

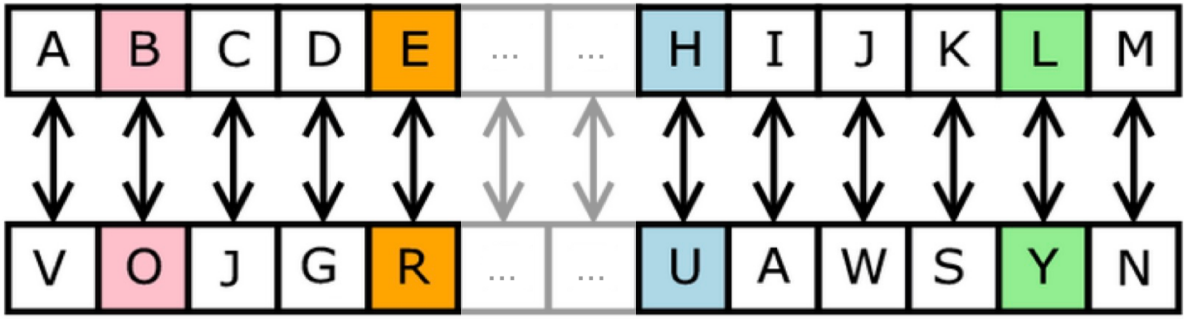


$e_K(x)=x+K \bmod 26$ et $d_K(y) = y - K \bmod 26$

- Key is the shift value (26 differents keys)
  - K=3 : Cesar code
- Cryptanalysis :
  - Brute force approach (trying all possible keys)

Substitution cipher

- Keys: one of the letter permutations of the alphabet (26! different keys)
- Each letter is replaced by its value in the permutation table



- Cryptanalysis
  - Al-Kindi, 9th century
  - Using the letter distribution of the message.

26! = 403 291 461 126 605 635 584 000 000

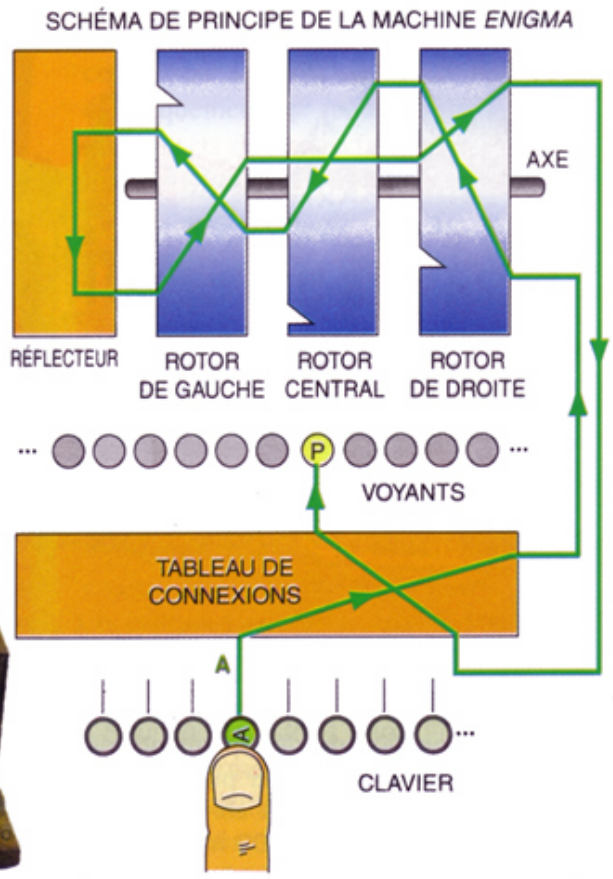
Vigenère cipher

- The message is divided into blocks of the size of the key. The key is added (modulo 26) to each block.

```
clef clef clef cle
leme ssag esec ret
OQRK VEFM HEJI UQY
```

- Polyalphabetic process: each character can take *m* different values
- Number of different keys is 26^m

Enigma



Alan Turing participate to enigma cryptanalysis

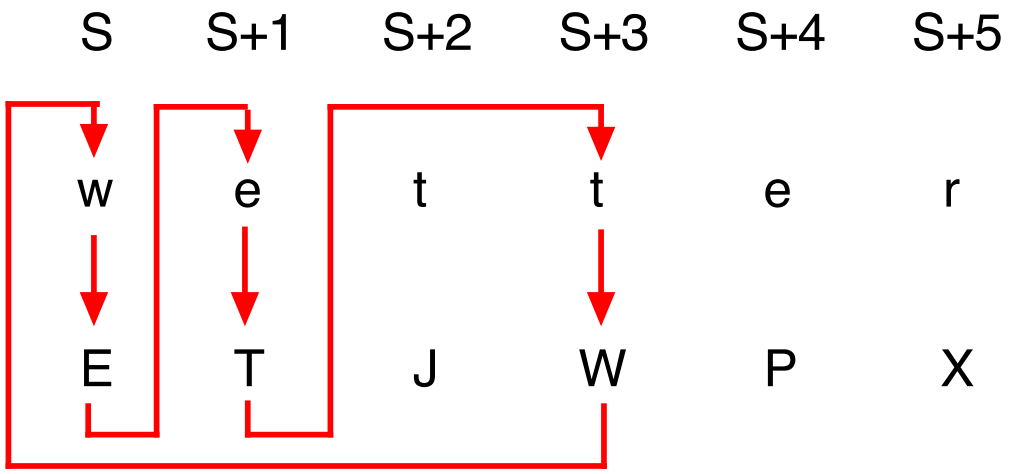
Enigma cryptanalisis

1,05869E+16 possible combinatisons

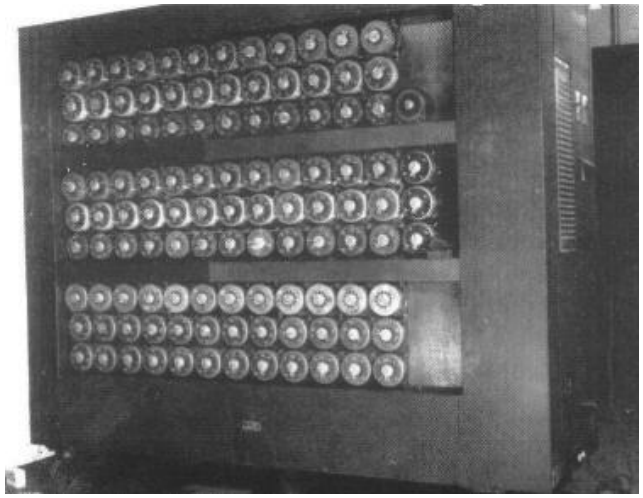
- Two sub-problems
  - 105456 positions for rotors:
    - Rejewski « Chains » method
  - 1,00392E+11 permutations
    - Statistical analysis

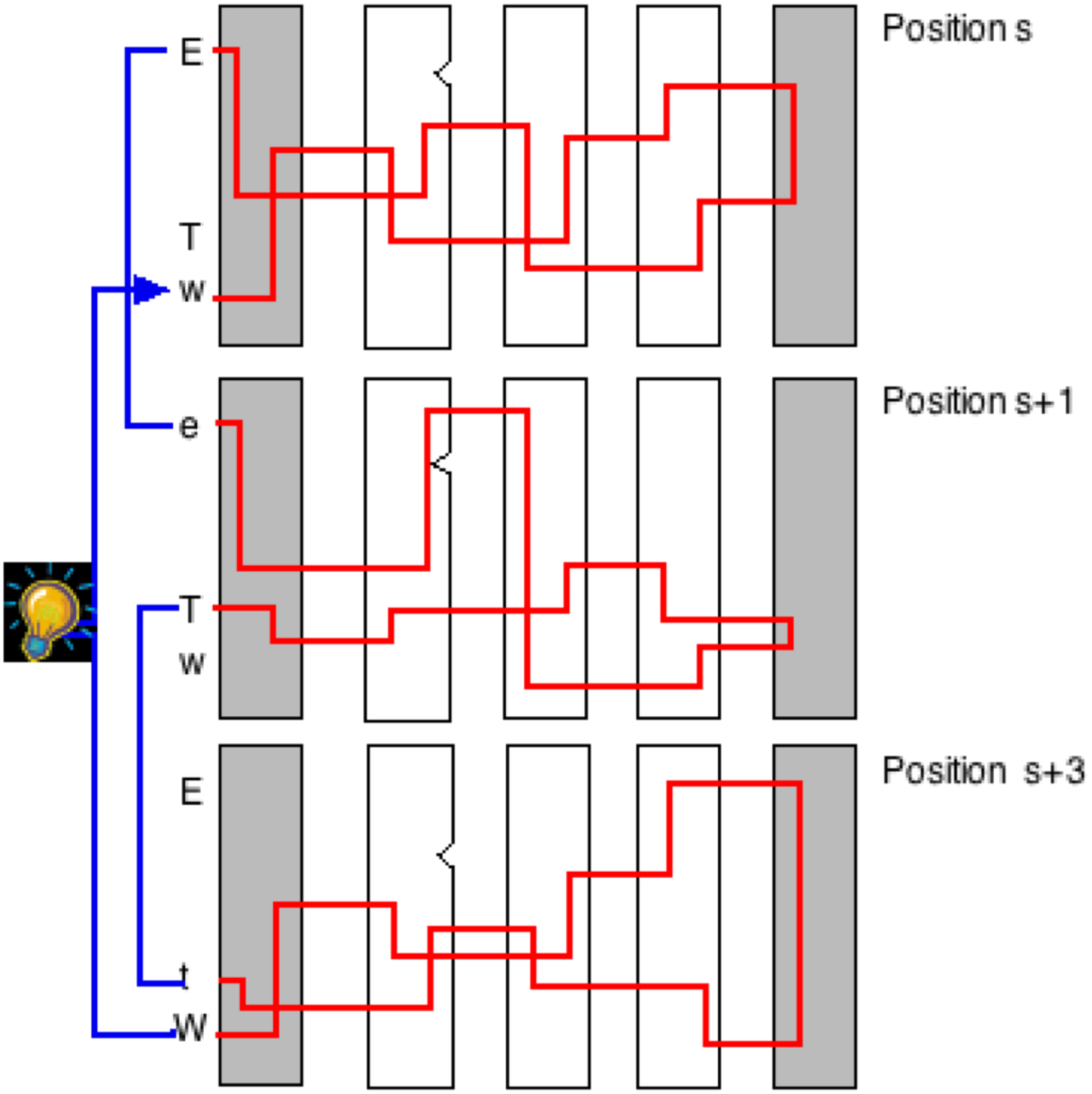
The Alan Turing method

- In its 1945 version, enigma used 3 rotors choosed from 6 (1 054 560 positions).
- Use of probable terms (the *cribs*)



Bombs



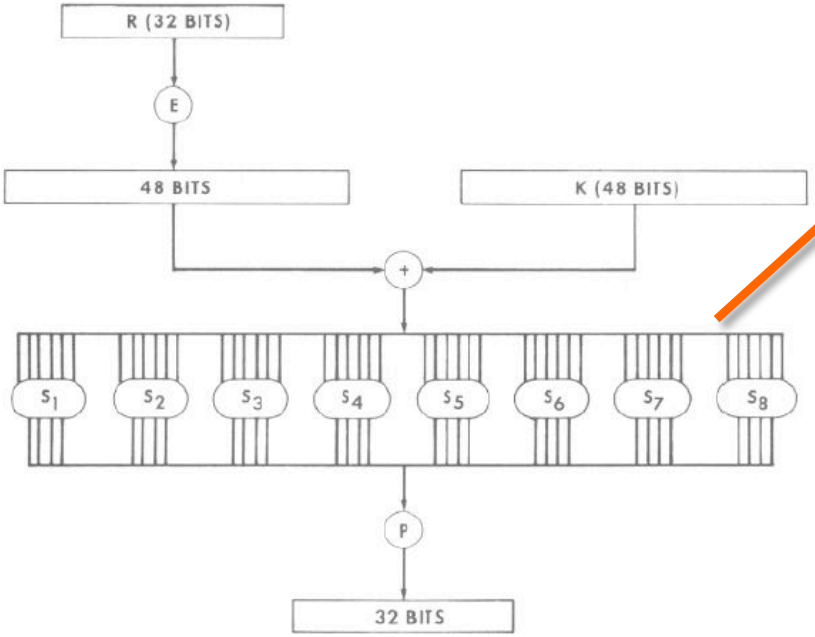


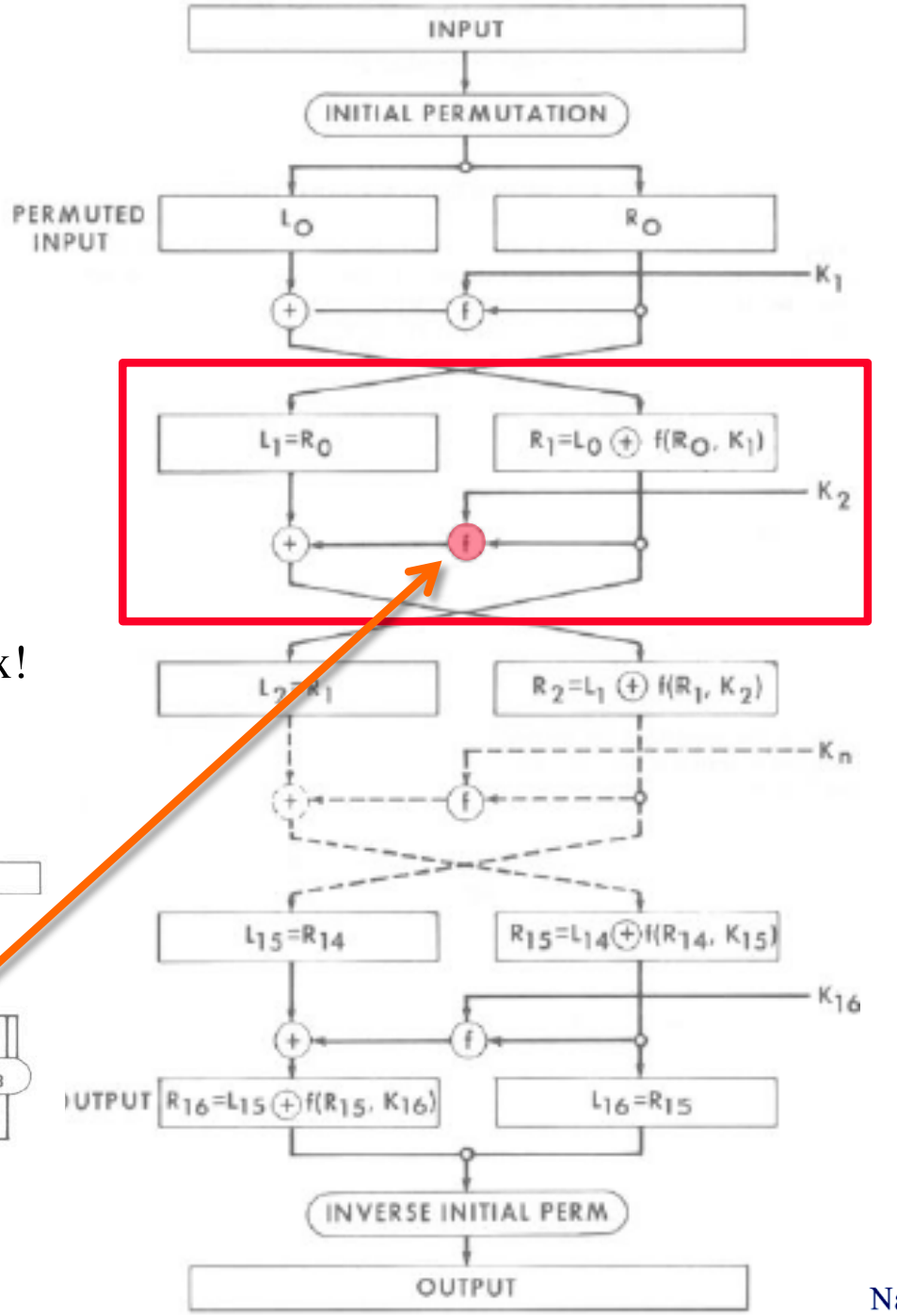
The breakthrough

- Computers and classical cryptography
  - DES (1973) .. AES (2001)
- Public-key cryptography
  - Diffie-Hellman (1976)
  - RSA (1977)

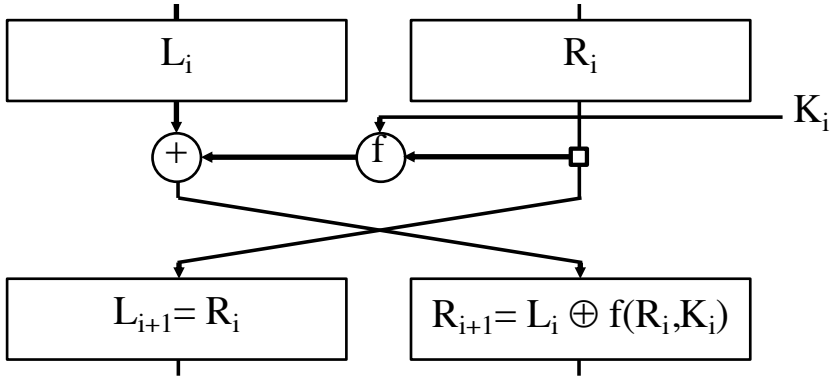
DES

- Ciphering a 64 bits block with a 56 bits key
- 16 rounds
- The S Box
  - Unknown criteria to build S box!





Feistel method

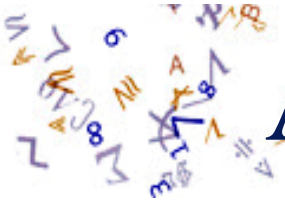


- How to build a « pseudo-random » bijection
  - G and D are the two part of a message M of 2n bits and f_1 a functions (not necessarily bijective)
  - We can build Ψ a bijective function

    $\Psi(L,D) = (S,T)$ with $S = R$ et $T = L \oplus f(R)$

    Ψ is bijective : $R = S$ et $L = T \oplus f(S)$
  - We can apply *n* times the Feistel schema

$$\Psi(L,R) = (L \oplus f_1(R), R \oplus f_2(L \oplus f_1(R)))$$



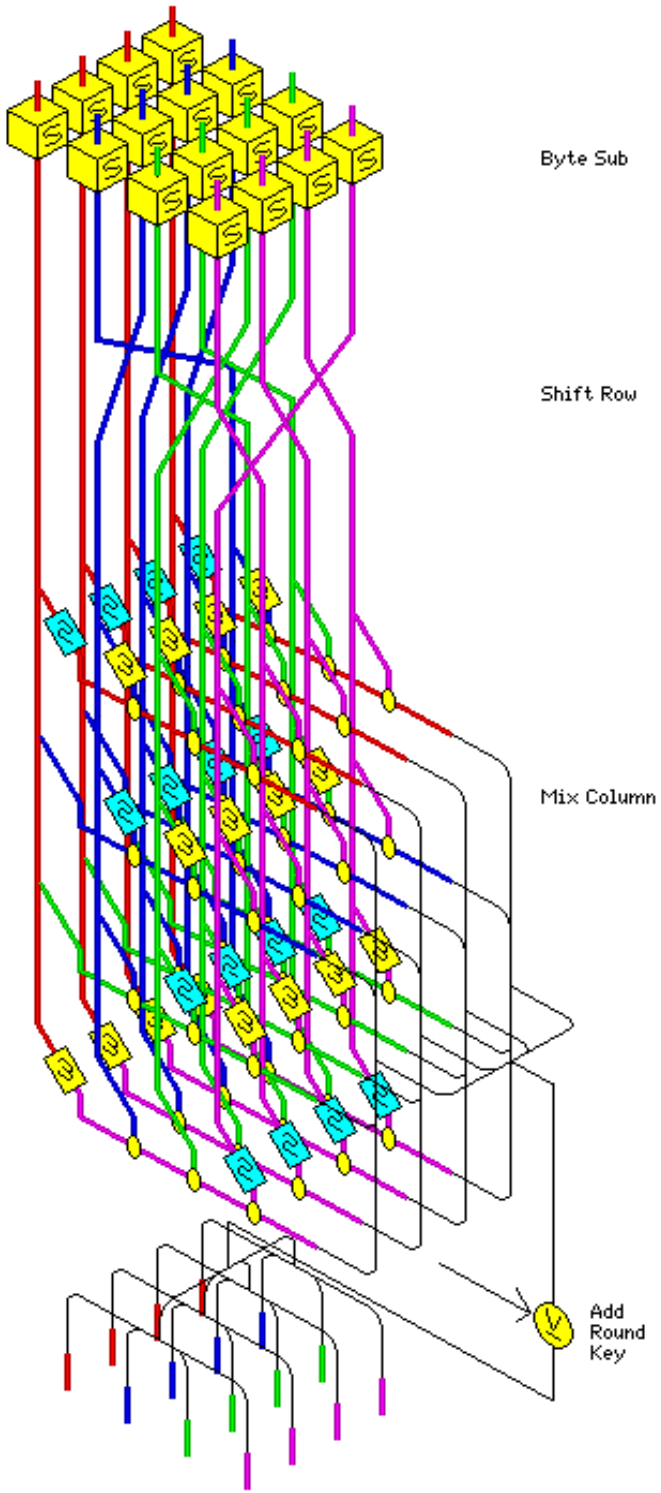
AES

Advanced Encryption Standard

- 1998 : paper explain how to break DES with « only » 250000$
- International contest launch by NIST
  - All algorithms are evaluated by internationnal community
  - Winner: Rijndael (J. Daemen & V. Rijmen, Belgium)
- Specifications
  - Key length 128, 192 et 256 bits
  - Fast as possible
  - Easy to implement (software/hardware)
  - Simple to understand
  - Copyright free

AES

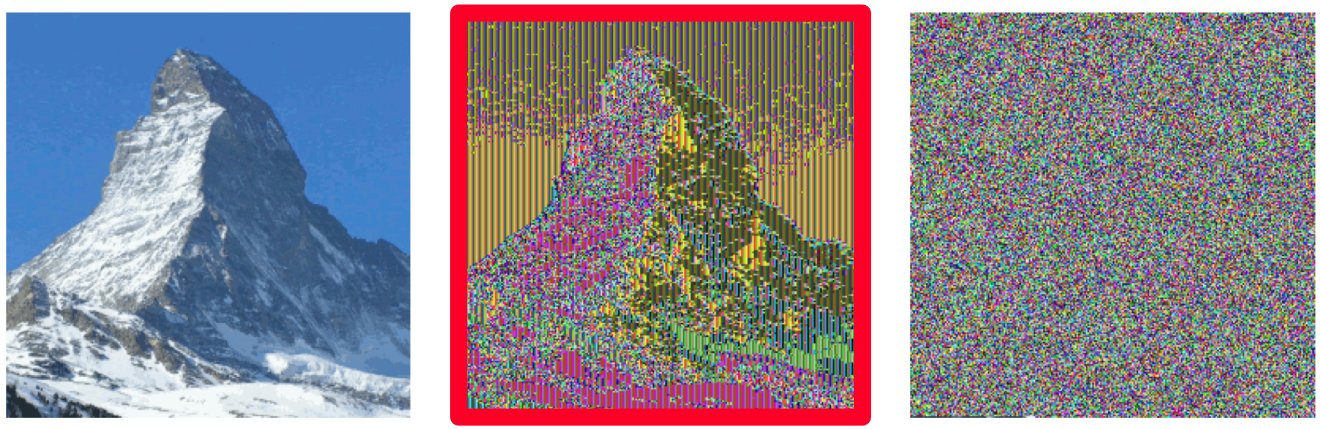
- 128 bits key
- 10 rounds
  1. Non linear transformation
  2. Shift rows
  3. Mix columns
  4. Add the round key

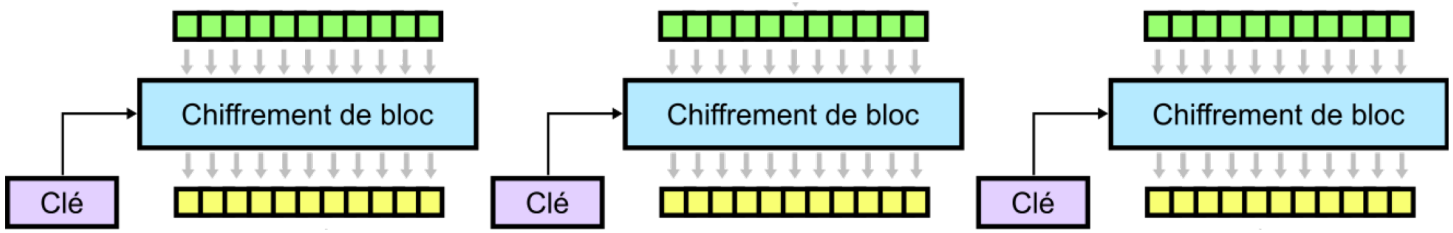


AES

```
Cipher(M, K[])
        AddRoundKey(M, K[0])
        for round = 1 step 1 to 9
                SubBytes(M_round)
                ShiftRows(M_round)
                MixColumns(M_round)
                AddRoundKey(M_round,K[round])
        end for
        SubBytes(M_9)
        ShiftRows(M_9)
        AddRoundKey(M_9,K[10])
End
```

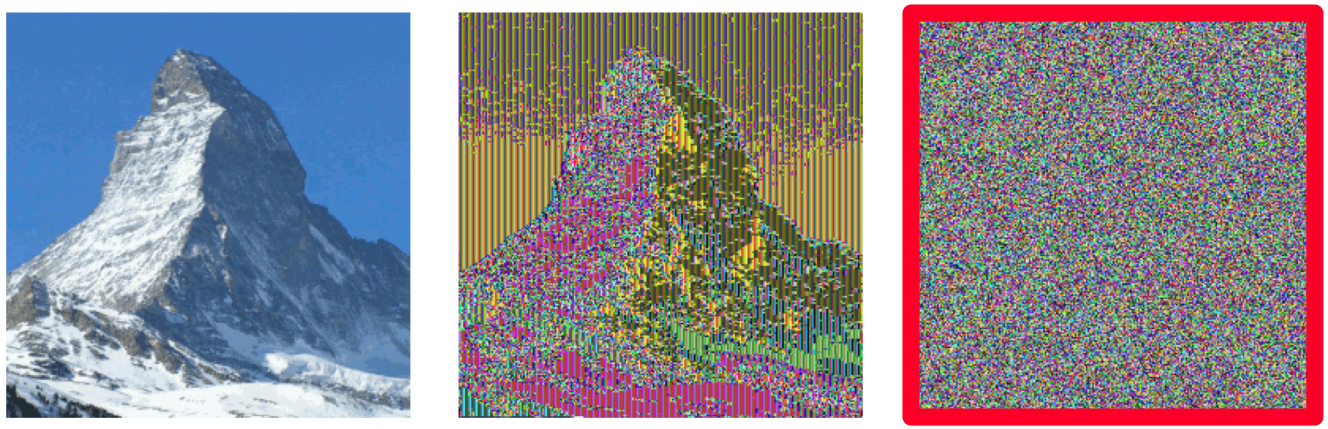
Using block cipher

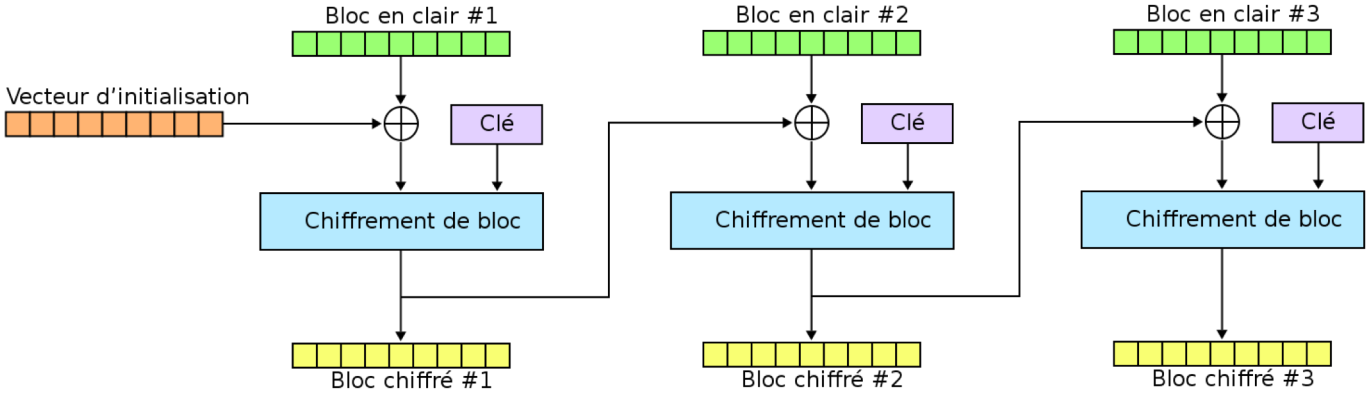




- ECB : Electronic Code Book

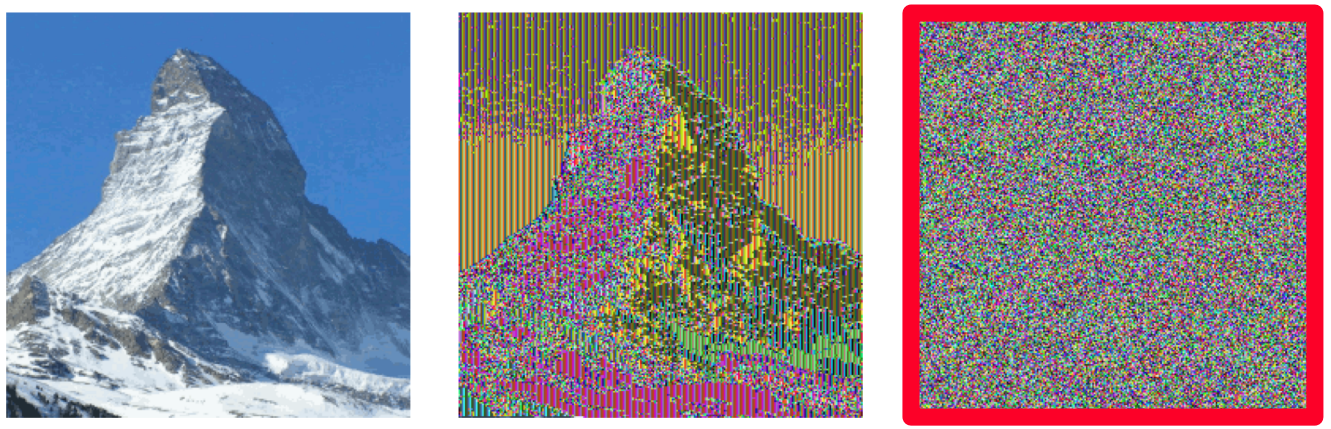
Using block cipher

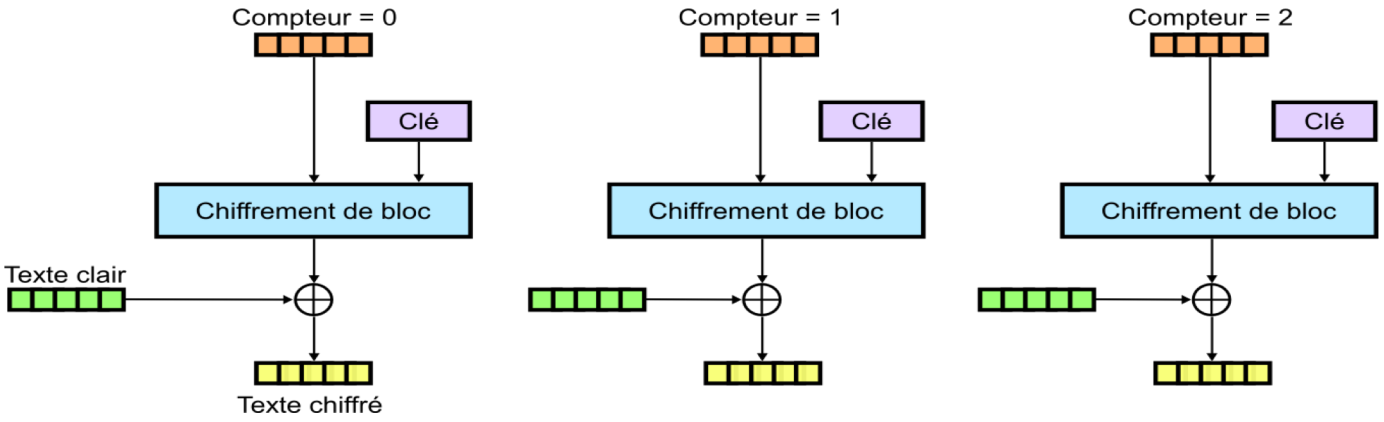




- CBC : Cypher Bloc Chaining

Using block cipher





- CTR : CounTeR

About cryptanalysis

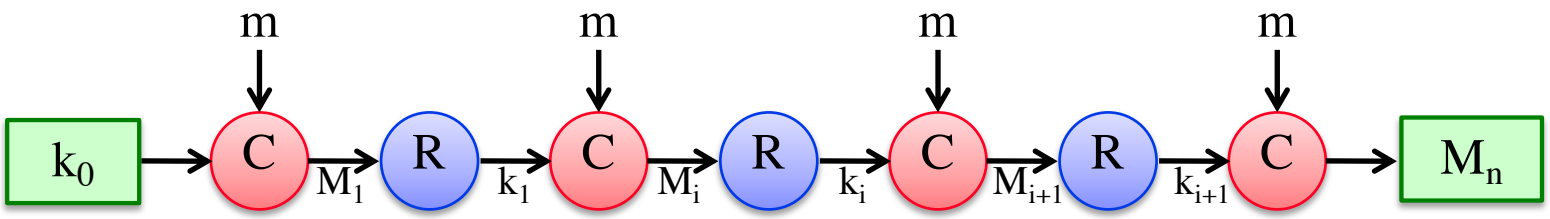
A crypto-system is good if there is not known attack better than trying all the possible keys

Brute force attack

- All cryptographic algorithms are breakable using brute force attack (if you have enough time and energy)
  - Needs exploring all the key space
    - takes very long time, to much memory…
  - Reducing the key space
    - In the general case, the full key space isn't used
    - Ad-hoc statistical properties, dictionary… can help
    - Reduce search time by pre-compute some data, build dictionaries
    - …

Space–time tradeoff

- Variant of brute force attacks
  - Time to compute all keys.
  - Space required to store all keys.



$$M_{i+1} = E(\, m, k_i = R(M_i))$$

k_0 = start of chain, M_n=end of chain

  - Build chains where we only retain extremities.

Martin Hellman (1980), Philippe Oechslin (1993)

Differential Cryptanalysis

- Comparing outputs of the algorithm when input messages have a fixed difference.

$$E(m) <=> E(m \oplus \delta)$$

- DES resist to differential analysis
  - It's evidence that authors knew the technique and were forewarned.

Eli Biham et Adi Shamir (Crypto 1990)

Linear Cryptanalysis

- Finding a linear approximation between output bits, input bits and bits of the key.
  - $M(m_1, m_2), K(k_1, k_2)$
  - $c_1+c_2 = a_1m_1 + a_2m_2 + a_3k_1 + a_4k_2$
- If for some $(a_1, a_2, a_3 \ldots a_n)$ a probability the occurrence of results $\neq 1/2$, then there exists a correlation between inputs and outputs.
- With this method, we can break DES in "only" 2^{39} operations.

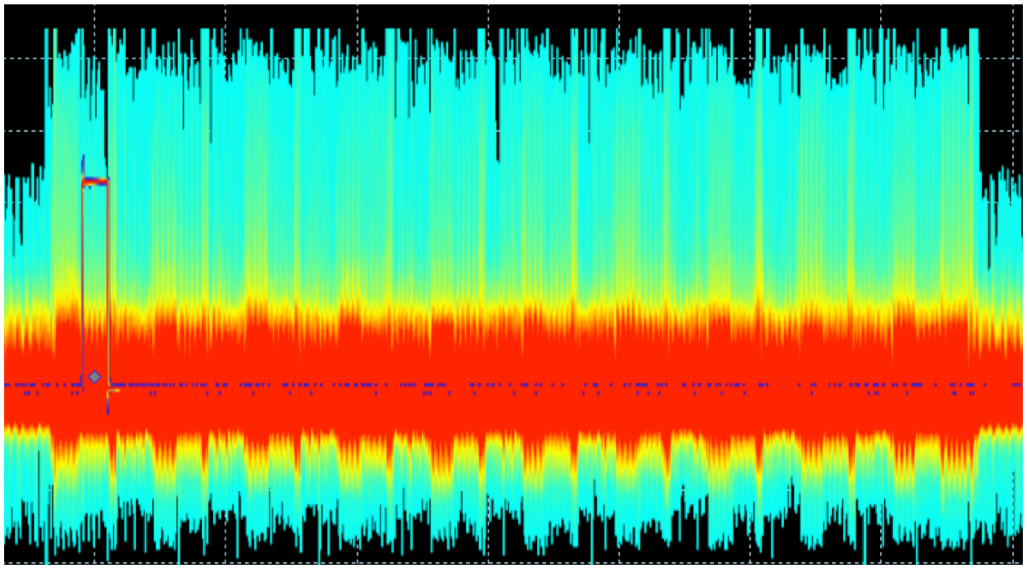
Mitsuru Matsui (1993), Henri Gilbert

About cryptanalysis (more)

Remember: security level of a system is equal to the lowest level of all its components.

- Some time it's better to find a "weakest link"
  - Hardware is a good candidate
  - Side channel attacks
  - Fault attacks

Differential Power Attack (DPA)



- The power consumption, time, electromagnetic radiation (Side Channels) of the chip are **strongly correlated** to the data (and so to the secret) which are manipulated by the circuit.

Correlation Power Attack (CPA)

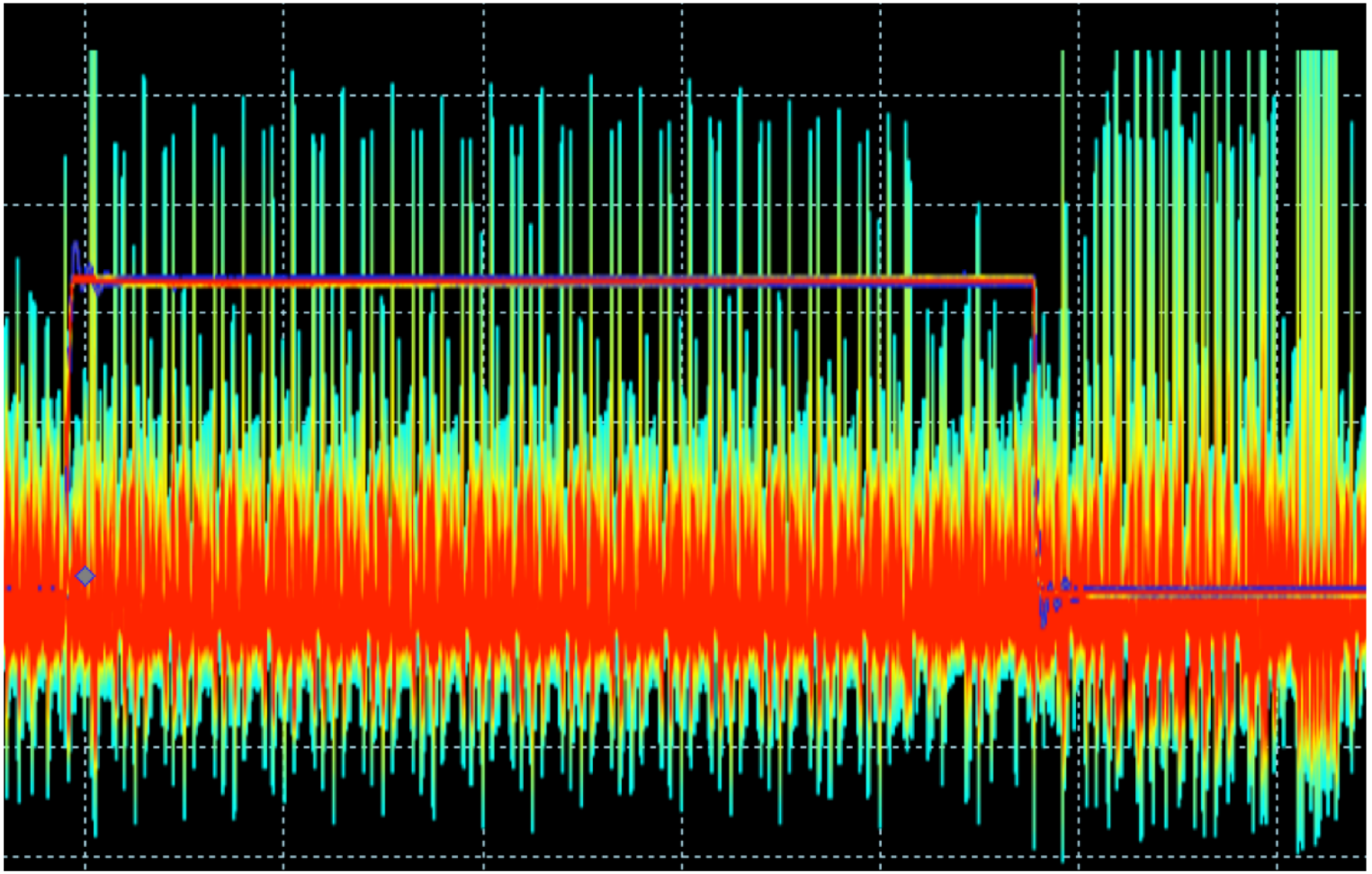
- Predictive attack on chosen text

1. Choose where to attack
   - ➢The first sub byte in AES is a good candidate
2. Build a consumption model
   - ➢Use for example Hamming weight
3. Compute correlation between the model and the acquired data
4. Enjoy …

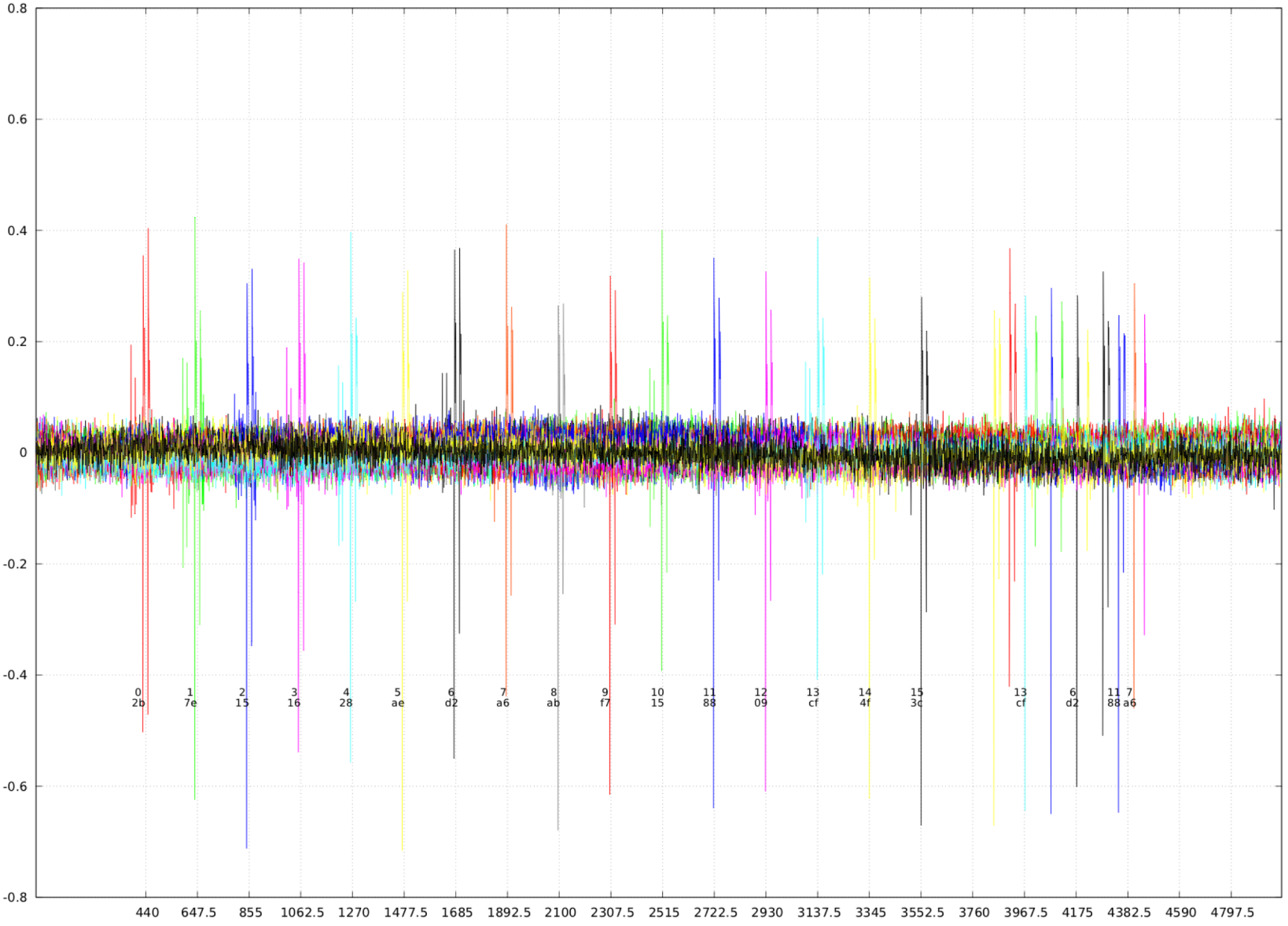
AES

```
Cipher(M, K[])
        AddRoundKey(M, K[0])
        for round = 1 step 1 to 9
                SubBytes(M_round)
                ShiftRows(M_round)
                MixColumns(M_round)
                AddRoundKey(M_round,K[round])
        end for
        SubBytes(M_9)
        ShiftRows(M_9)
        AddRoundKey(M_9,K[10])
End
```

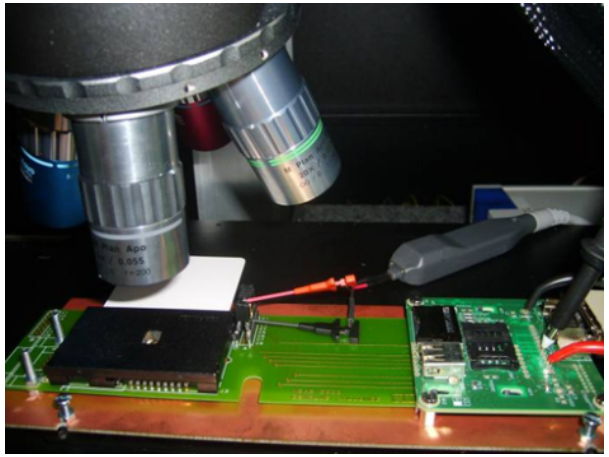
AES SubBytes



Picoscope / 200Mhz -2μs par division


0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0 2b
1 7e
2 15
3 16
4 28
5 ae
6 d2
7 a6
8 ab
9 f7
10 15
11 88
12 09
13 cf
14 4f
15 3c
13 cf
6 d2
11 88
7 a6
440
647.5
855
1062.5
1270
1477.5
1685
1892.5
2100
2307.5
2515
2722.5
2930
3137.5
3345
3552.5
3760
3967.5
4175
4382.5
4590
4797.5

Differential Fault Attack (DFA)



- Errors may be induced during the computation of a cryptographic algorithm.
- It can be used to recover secrets
  - D. Boneh, R. A. Demillo and R. J. Lipton, On the Importance of Checking Cryptographic Protocols for Faults, EUROCRYPT'97
  - E. Biham and A. Shamir, Differential Fault Analysis of Secret Key Cryptosystems, CRYPTO'97

Differential Fault Attack of AES

```
Cipher(M, K[])
        AddRoundKey(M, K[0])
        for round = 1 step 1 to 9
                SubBytes(M_round)
                ShiftRows(M_round)
                MixColumns(M_round)
                AddRoundKey(M_round,K[round])
        end for
        SubBytes(M9)
        ShiftRows(M9)
        AddRoundKey(M9,K[10])
End
```

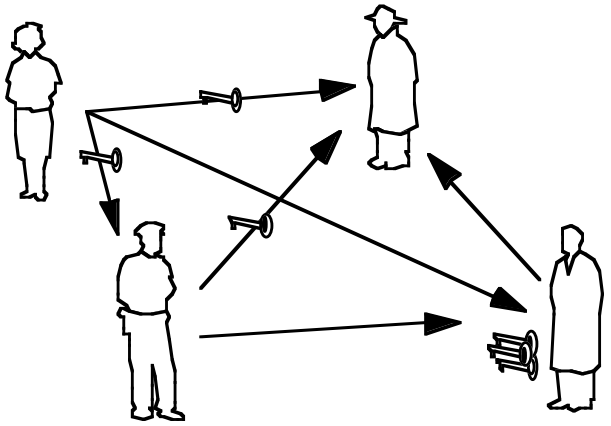
G(F(M^K[0],K))^K[10]

- If we can introduce errors in M_9 before `AddRoundKey` step:
  - `C=shift(sub(M9))^K[10]` ← without error
  - `D=shift(sub(M9^e))^K[10]` ← with error
  - `C^D = shift(sub(M9))^shift(sub(M9^e))`
- If we can find M_9 we can recover `K[10]`, and `K`

How-to sign ?

- Use of a trusted third party
  - Ivan is a powerful and respected referee
  - It shares the *ka* secret key with Alice (and *kb* with Bob)
  - Alice encrypts her message to Bob with *ka* and sends result to Ivan
  - Ivan decrypts the message with *ka*
  - Ivan added a notice to the message that he received it from Alice then encrypts everything with *kb*
  - Bob decrypts all with *kb*
- Difficult to implement, the third party must keep track of transactions to avoid unauthorized use
- Everything depends on the third party integrity

Limits



- Keys exchange with a secured channel
- Compromises keys problems
- We need one key for each users pair
  - Each user must have one key for each peers
  - Number of keys used in the system : n(n-1)/2

Is the method secure?

- Concrete security (complexity)
  - The best known algorithm needs N operation to break cipher and N is too large.
  - Security of the system is equal to the difficulty to solve a well known problem (NP-Hard problem for example).
- Unconditional security (probabilities)
  - Can't be break, even when we have unlimited computing power (ex. Vernam 1917).

$$Pr(x|y) = Pr(x)$$

Ciphertext provides no information about plaintext
without knowledge of the key

The Vernam cipher

- Unbreakable if we never reuse the same key.
  - Proved by Shannon in the thirty
- Simple encryption algorithm
  - $e_K(x) = (x_1+K_1, x_2+K_2, \ldots, x_n+K_n) \bmod 2$
  - $d_K(y) = (y_1+K_1, y_2+K_2, \ldots, y_n+K_n) \bmod 2$
- Difficult to use
  - How to transmit K as long as the message
  - K can be used only once

One-way hash function

- A one-way hash function *H(M)* acts on a message of any size and provides a hash value of a fixed length.

$$h = H(m)$$

  - For a given *m*, it's easy to compute *h*
  - For a given *h*, it's difficult to compute *m*
  - For a given *m*, it's difficult to find another message *m'* such that *H(m) = H(m')*.

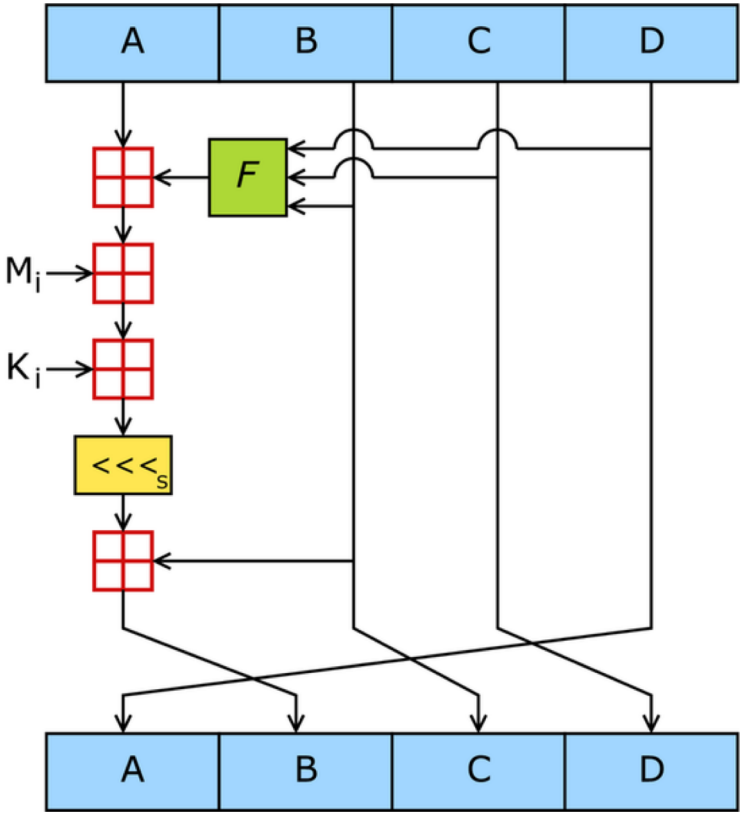
- Used for the integrity check, signature…

MD5

Message Digest 5

- R. Rivest, MIT & RSA Data Security, Inc. April 1992 (rfc 1321)
  - 128 bits digest
  - 64 round for each 512 bits block
- Collisions
  - Exhaustive search: 2^{64} operations.
  - 2004 : Complete collision of 2 messages Xiaoyun Wang, Dengguo Feng, Xuejia Lai.
  - 2007 : Colliding X.509 Certificates for Different Identities (M. Stevens, A. Lenstra, and B. de Weger - EuroCrypt).

$$MD5(m_1|b_1) == MD5(m_2|b_2)$$

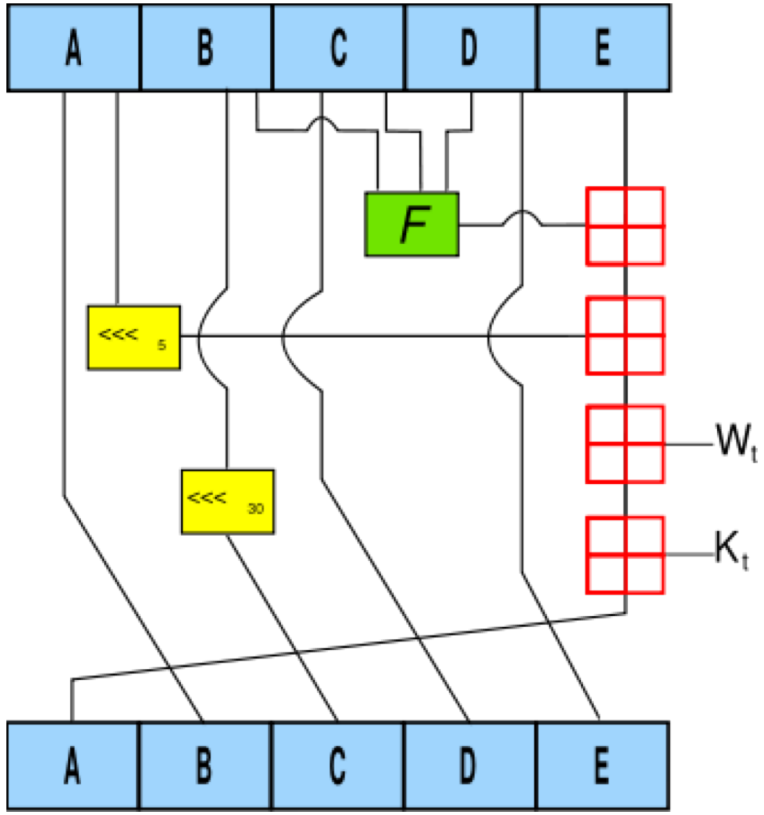


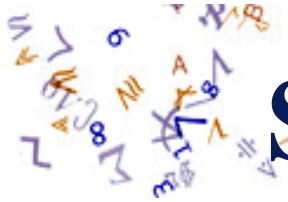
Therefore, no longer use MD5

SHA-1

Secure Hash Algorithm

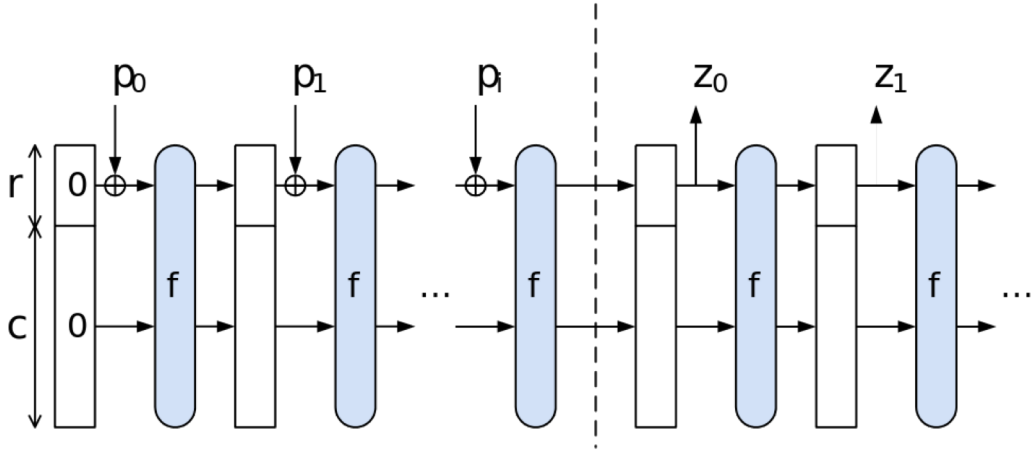
- NSA 1993, published by the NIST
  - FIPS 180-3 (SHS), rfc 3174
- 160 bits digest
  - 80 round for each 512 bits block
- Collisions
  - Birthday paradox : 2^{80} operations
  - february2005 : Wang, Yin et Yu. 2^{69} operations
  - Crypto 2005 : 2^{63} operations.
  - Eurocrypt 2009 : 2^{52} operations.
- Preferably use SHA extensions
  - SHA-256, SHA-384, SHA-512





SHA3
Keccak and the *Sponge Functions*

- NIST launched a contest for the new SHA in November 2007
- Winner declared in September 2012:
  - G.Bertoni, **J. Daemen**, M. Peeters and G.Van Assche (STMicroelectronics, NXP Semiconductors)



http://sponge.noekeon.org/

Z/nZ

- Euler indicator
  - φ(n) is the number of invertible elements of $Z/_nZ$.
    - φ(n) = n-1 if n is prime,
    - si PGCD(m,n)=1 φ(mn)=φ(m)φ(n)
- Euler theorem
  - If GCD(*a*,n) = 1 then $a^{\varphi(n)} = 1$.

  → inv(a) = $a^{\varphi(n)-1}$
- Small Fermat theorem
  - If *n* is prime and if *n* doesn't divide *a* then $a^{n-1}=1 \bmod n$

  → If *n* is prime and *a* integer, $a^n=a \bmod n$

Prime numbers

- There is about 10^{151} 512 bits prime numbers
- How to test number primality:
  - Factorization: very difficult
  - Prove that a number is composite: more easy
    - Positive Monte-Carlo algorithms
  - Remove first naive composite numbers
    - All the even numbers: remove 50% of candidates
    - Composite with 5 and 7: remove 54% odd numbers
    - Composite with prime numbers < 100, 76% ; < 256, 80%

Chances to be prime

- Solovay-Strassen (1978)
  - Repeated *t* times this test, have *log* n -2 / *(log* n-2 + 2^{t+1}*)* chances to not be prime
- Rabin-Miller (1980)
  - Repeated *t* times, 1-1/4^t chances to be prime
- Lehmann (1982)
  - Repeated *t* times, 1-1/2^t chances to be prime

Random numbers

- A computer is not a random machine
- It is very difficult to produce good random numbers
- Pseudo Random Number Generator
  - Linear Congruential Generator
  - Linear Feedback Shift Registers
  - Blum Blum Shub

Linear congruential generator

$$X_{n+1} = (aX_n + b) \bmod m$$

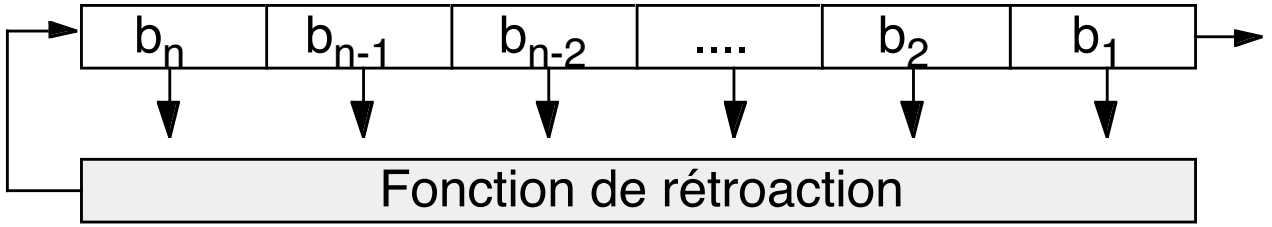
- Seed (initial value): X_0
- Maximal period is at most m
- Extremely sensitive to the choice of a, b and m

Ex. : a=8121, b=28441 and m=134456

Do not use for cryptographic purpose

Linear Feedback Shift Registers

- Hardware easy to built
- Must be combined to be not trivially predictable
- Maximal period is 2^n-1

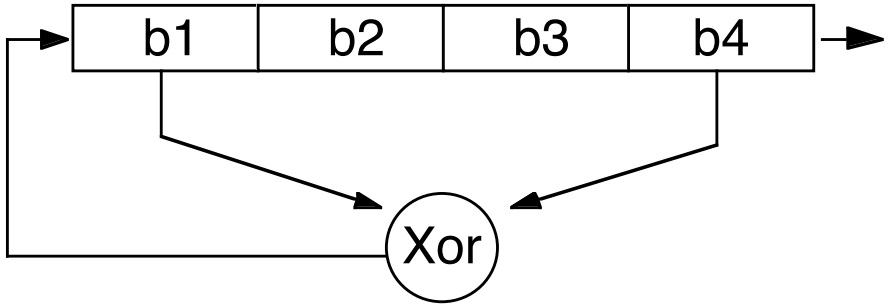


Used in GSM cell phones, Bluetooth… but craked!

Linear Feedback Shift Registers

- Example for a 4bits LFSR

```
1111 (seed)
0111
1011
0101
1010
1101
0110
0011
1001
0100
0010
0001
1000
```



Blum Blum Shub generator

- p and q are two large prime numbers, both congruent to 3 (mod 4)
- n = p.q is a Blum integer
- Choose a seed $1 < x_0 < n$
- $x_i = x_{i-1}^2 \bmod n$
- b_i is the less significant bit in x_i
- Computing directly b_i:
  - $x_i = x_0^{(2\hat{}i) \bmod ((p-1)(q-1))} \bmod n$

Blum Blum Shub generator

- Interesting properties:
  - It is impossible to predict the next bit without factoring n
  - The generator is "unpredictable" at left but also at right
  - We could use the less significant $\log_2\log_2 n$ bits of x_i (9 bits if $n \cong 2^{512}$)

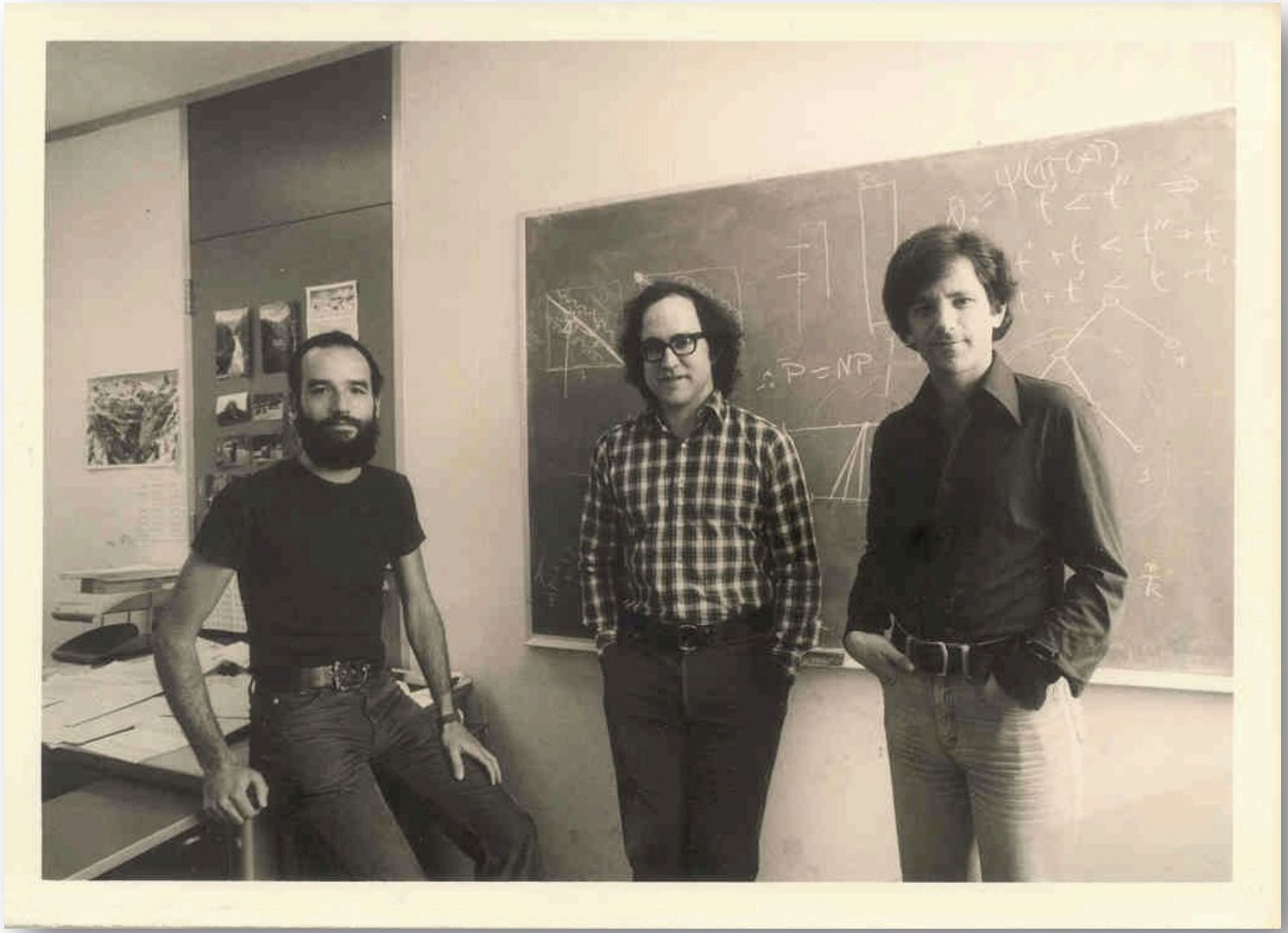
Diffie-Hellman

- First public key algorithm: 1976
- Security is based on the difficulty of computing discrete logarithms.
- Given $X = g^x \pmod n$, it is very difficult to calculate x.
- Used as a secret key exchange protocol (session key).

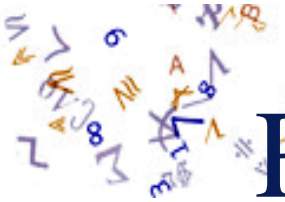
Diffie-Hellman

- Alice and Bob choose n and g, two large integers, g and n are relatively prime.
- Alice chooses a random integer x and sends to Bob
  - $X = g^x \ (mod\ n)$
- Bob chooses a random integer y and sends to Alice
  - $Y = g^y \ (mod\ n)$
- Alice computes $k = Y^x \ (mod\ n) = g^{xy} \ (mod\ n)$
- Bob computes $k = X^y \ (mod\ n) = g^{xy} \ (mod\ n)$
- k can be used as a secret key for the session

RSA - MIT 1977



Ronald Rivest, Adi Shamir et Leonard Adleman



RSA
R. Rivest, L. Shamir et L. Adelman

"The public key algorithm"

- Get 2 large prime numbers p and q ; compute $n = pq$
- Chose a random encryption key e as:
  $pgcd(e,(p-1)(q-1))= 1$.
- Choose the decryption key d as:
  $d = e^{-1} \ mod((p-1)(q-1))$ (Euclide)
- (e,n) are the public key and (d,n) are the private key.
- Encryption E : $c = m^e \ (mod\ n)$
- Decryption D : $m = c^d \ (mod\ n)$

RSA : Why it works

- Rappels
  - if p prime, $\varphi(p) = p-1$
  - $m^{\varphi(p)} = 1 \pmod p$ if m prime with p
  - $\varphi(p.q) = \varphi(p)\varphi(q)$, $m^{\varphi(p.q)} = m^{\varphi(p)\varphi(q)} = 1 \pmod{p.q}$
  - If $a \equiv 1 \pmod p$, $a = kp + 1$
- $m = m^{ed} \pmod n$
  - $ed \equiv 1 \bmod (p-1)(q-1)$, $(p-1)(q-1) = \varphi(p)\,\varphi(q) = \varphi(n)$
  - $ed = k\varphi(n) + 1$
  - $m = m^{k\varphi(n)+1} = m \,.\, m^{k\varphi(n)} \pmod n = m \,.\, 1 \pmod n$

RSA

- Operations can be switched
  - $D(E(m)) = (E(m))^d = (m^e)^d = m^{ed} \pmod n$
  - $E(D(m)) = (D(m))^e = (m^d)^e = m^{ed} \pmod n$
- Used to sign and ensure non-repudiation

Using RSA

- Encryption with the public key of your correspondent
  - Only your correspondent can read the message.
- Encryption with your private key
  - The message can be read by everyone, but it is certain that you are the author.
- Signature: mix of the two
  - Encrypt the message with your private key, and encrypt the result with your correspondent's public key.
  - Only your correspondent can red the message and he is certain you are the author.

RSA

- The security of the algorithm is based on the difficulty of large numbers factorization.
- Be careful to poor implementations
  - $a^e \ mod \ n = a^e$ if a or e are small in front of n
  - Computing a^x de manière simpliste

    `If x is even then` $a^x = (a^2)^{x/2}$

    `Else` $a^x = a.(a^2)^{(x-1)/2}$
  - *For efficiency purpose, we use small public exponents (3 or $2^{16}+1$ in X509), in this case* ***e*** *and* ***d*** *don't play the same role and can't be exchanged.*

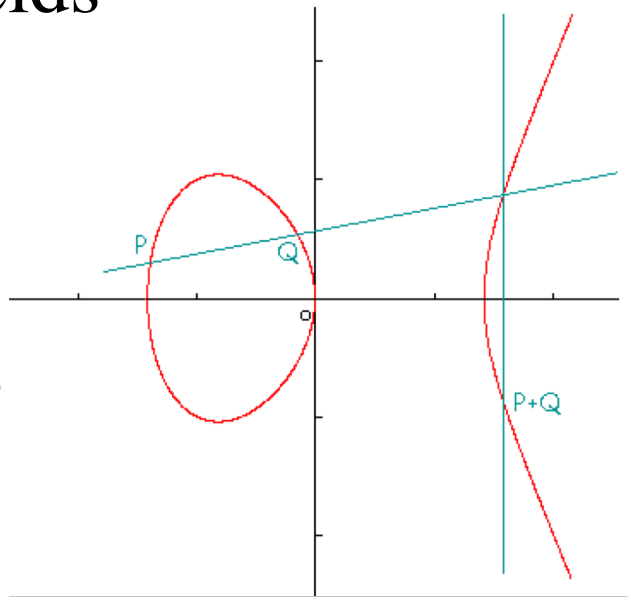
RSA

- Advantage
  - RSA is safe
  - It can be used to sign documents
  - It can authenticate the sender
- Disadvantages
  - Authenticity of public keys
  - RSA is slow (1000 times slower than DES or AES)

- RSA can be used to exchange session keys.

Elliptic curves

- E : $y^2=x^3+ax+b$ (avec $4a^3+27b^2 \neq 0$)
  - It's a commutative group (+, O)
- They can be defined on finite fields
  - $Z/_pZ$ with *p* prime
  - For any points G of E,
    {0, G, G+G, G+G+G,… }
    is a cyclic group
  - Can be used as a multiplicative group for the power of integers modulo p
    $\{g^0, g^1, g^2, g^3, \ldots\}$



Elliptic curves

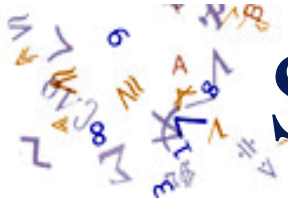
- Use of elliptic curves give a new life to Diffie-Hellman.
  - Shorter keys in Z / pZ (200 bits instead of 1024 for RSA)
  - Attractive for small devices (computation time is proportional to key length)

➢ DH : k = x(yG) = y(xG) = xyG (G is a point of the EC)

Standard intenational : Elliptic Curve Integrated Encryption Scheme

Session keys

- Private key cryptography: sharing a secret
- Public key cryptography: slow
- If Alice wants to communicate efficiently with Bob
  - Alice chooses a secret key k, she signs it with its private key and encrypts it with the Bob's public key. She sends this message.
  - Bob can extract the secret key k and it is sure of its origin.
  - The following communications can now be efficiently encrypted using the key k and a “private key” algorithm.
- Diffie-Hellman protocol can be used to exchange session keys, but it does not provide any authentication.

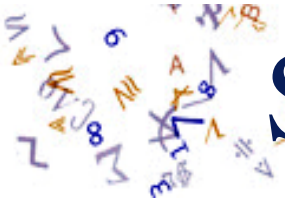


Security of cryptographic protocols

- Mutual authentication
  - Needham-Schroeder (1978)

$$A \rightarrow B : \{A, N_a\}K_B$$
$$B \rightarrow A : \{N_a , N_b\}K_A$$
$$A \rightarrow B : \{N_b\}K_B$$

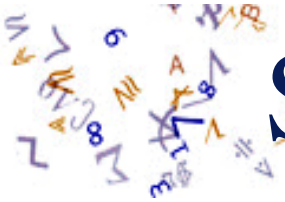
BAN notation (Burrows, Abadi, Needham)



Security of cryptographic protocols

- Man in the Middle Attack

$A \rightarrow \{A, N_a\}K_I$	$\rightarrow I \rightarrow$	$\{A, N_a\}K_B$	$\rightarrow B$
$A \leftarrow \{N_a, N_b\}K_A$	$\leftarrow I \leftarrow$	$\{N_a, N_b\}K_A$	$\leftarrow B$
$A \rightarrow \{N_b\}K_I$	$\rightarrow I \rightarrow$	$\{N_b\}K_B$	$\rightarrow B$



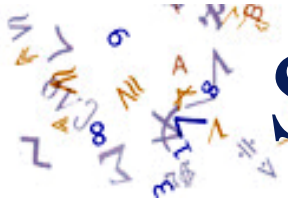
Security of cryptographic protocols

- First proposed revision

$$A \rightarrow B : \{A, N_a\}K_B$$
$$B \rightarrow A : \{N_a, N_b, B\}K_A$$
$$A \rightarrow B : \{N_b\}K_B$$

- Data type problem

			$I \rightarrow$	$\{A, I\}K_B$	$\rightarrow B$
$A \leftarrow$	$\{I,(N_b,B)\}K_A$	$\leftarrow$	$I \leftarrow$	$\{I, N_b, B\}K_A$	$\leftarrow B$
$A \rightarrow$	$\{(N_b,B), N_a,A\}K_I$	$\rightarrow$	$I \rightarrow$	$\{N_b\}K_B$	$\rightarrow B$
$A \leftarrow$	$\{N_a\}K_A$	$\leftarrow$	I		



Security of cryptographic protocols

- Right proposal

$$A \rightarrow B : \{A, N_a\}K_B$$
$$B \rightarrow A : \{B, N_a, N_b\}K_A$$
$$A \rightarrow B : \{N_b\}K_B$$

$$\begin{array}{llllll} & & & I \rightarrow & \{A, I\}K_B & \rightarrow B \\ A \leftarrow & \{B,(I, N_b)\}K_A & \leftarrow & I \leftarrow & \{B, I, N_b\}K_A & \leftarrow B \\ A \rightarrow & \{A,(I, N_b), N_a\}K_B & \rightarrow & I \rightarrow & ??? & \rightarrow B \\ A \leftarrow & ??? & \leftarrow & I & & \end{array}$$

Implementation

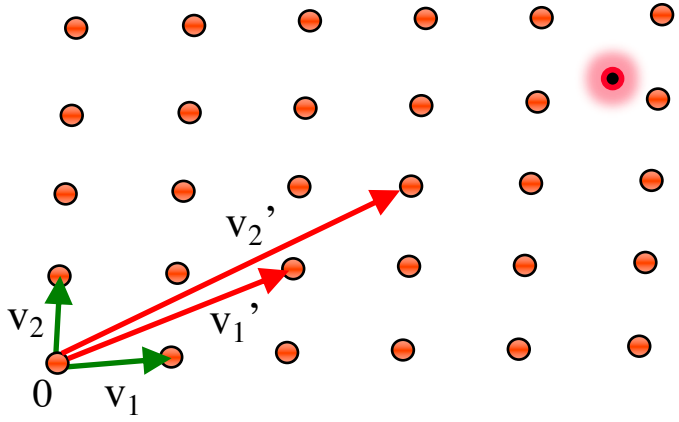
- How is encrypted your password
  - A one-way hash function
  - Using secure NIS, Kerberos ...
- Sending email safely
  - PGP (Pretty Good Privacy)
  - X509 and certification authorities
- Web Security: SSL (Secure Socket Layer)
  - Ephemeral session keys with Diffie-Hellman
  - Authentication mechanisms described by PKI (Public Key Infrastructure).
- Network Security: IPSec
  - Session keys with Diffie-Hellman
  - Sealing data with MD5, SHA1 ...
  - Data encryption with 3DES, AES, Blowfish …

RSA is dead!

- Peter Shor propose in 1994, a quantum algorithm for integer factorization.
  - Complexity : $O((\log N)^3)$
  - In 2001, IBM, factor 15 into 3×5 on a quantum computer with 7 qubits

Imply a *post-quantum* cryptography

Lattice encryption

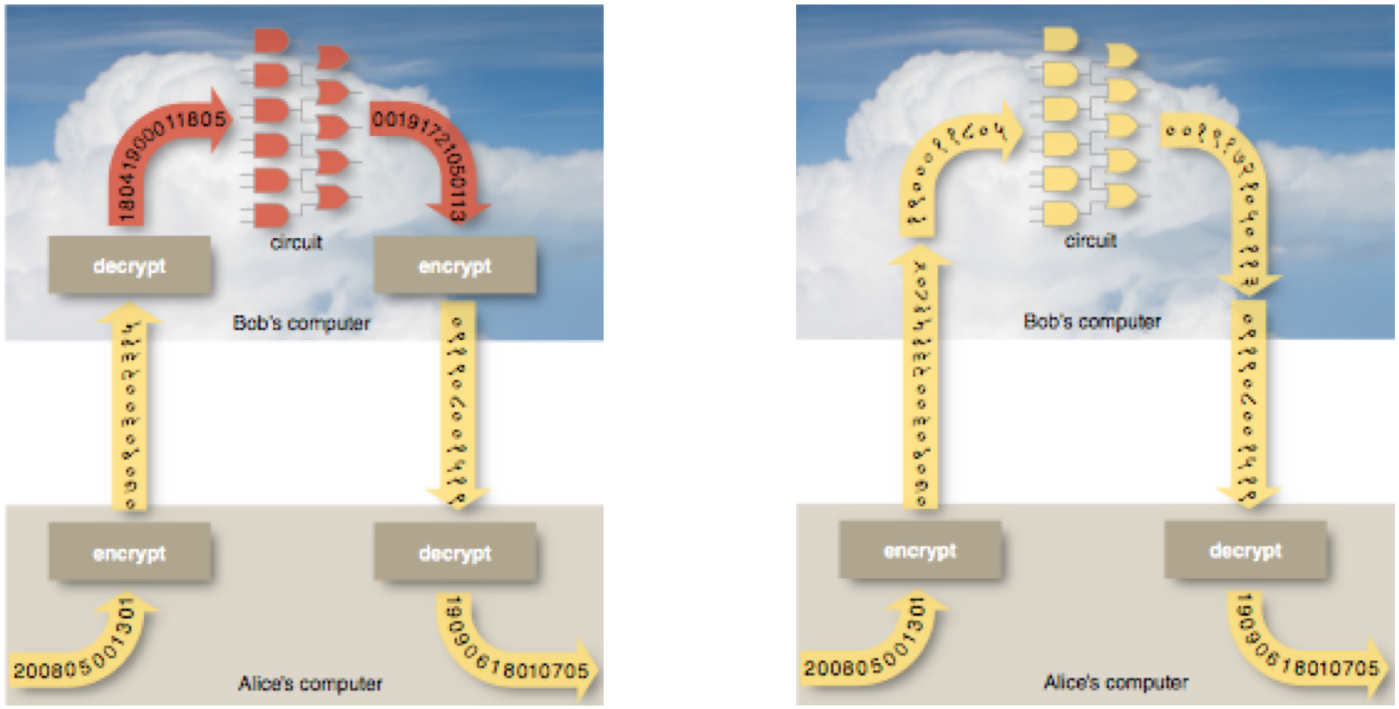


- The main idea:
  - Public key corresponds to a “bad” basis.
  - Private key is a “good” basis for this same lattice.
  - The plaintext is encoded as a point in the space R^N.
  - Encryption translates that point into the bad (public) basis.
  - Decryption translates the ciphertext point into the good (private) basis.

The security relies partly on the fact that it is generally difficult to find a good, nearly orthogonal basis for a given lattice.

Lattice encryption is quantum computing resistant

Homomorphic encryption

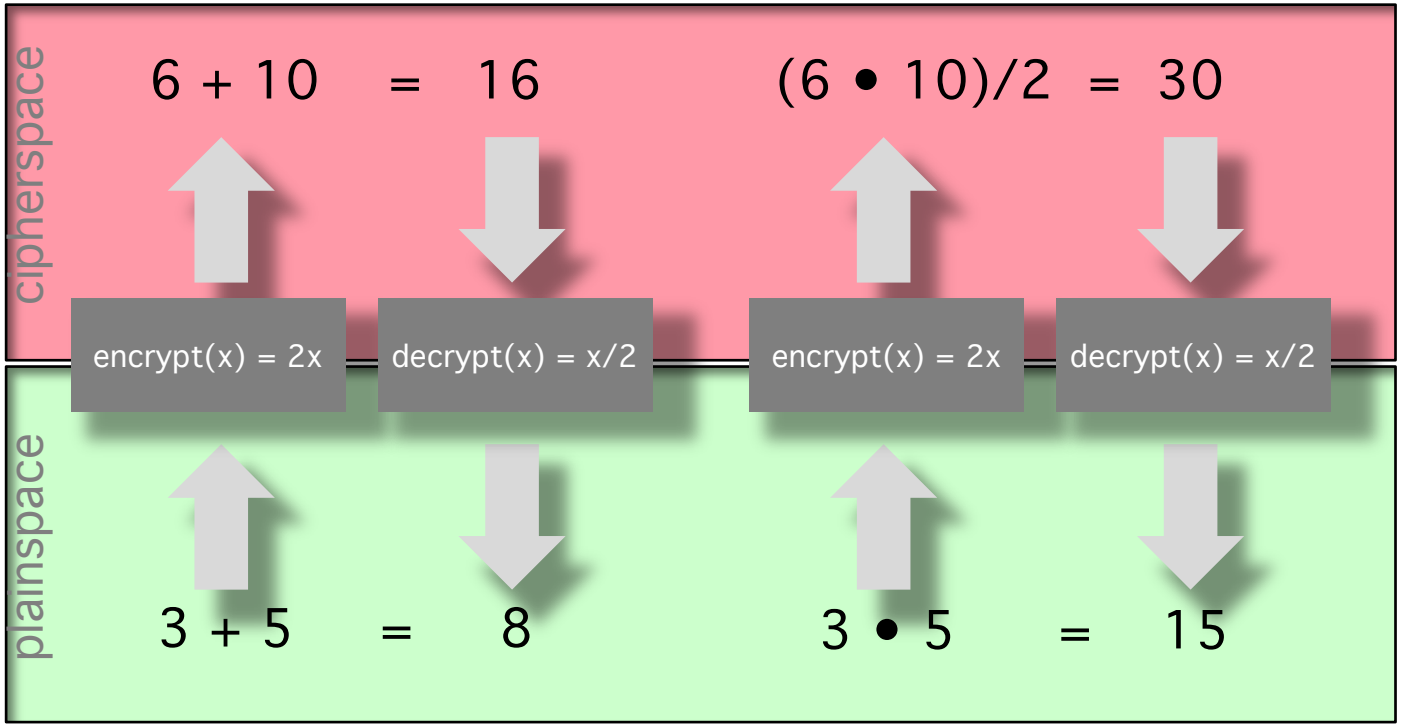


$$\{ a \}_k \; op' \; \{ b \}_k == \{ a \; op \; b \}_k$$

A dream for cloud computing !

Homomorphic encryption

- Toy exemple



Please, never use it for security purpose ☺

Partial homomorphic schemes

- RSA (and el Gamal)
  - E : $c = m^e \ (mod\ n)$

$$c_1 \bullet c_2 = m_1^e \ (mod\ n) \bullet m_2^e \ (mod\ n)$$
$$= (m_1^e \bullet m_2^e) \ (mod\ n)$$
$$= (m_1 \bullet m_2)^e \ (mod\ n)$$
$$c_1 + c_2 = ???$$
$$s$$

- Paillier cryptosystem
  - E : $c = g^m.r^n \ (mod\ n^2)$
  - D : $m = L(c^\lambda \ mod\ n2).\ \mu \ mod\ n$

g=n+1

$\mu=\varphi(n)^{-1} \bmod n$

$\lambda=\varphi(n)$, $L(u)=(u-1)/n$

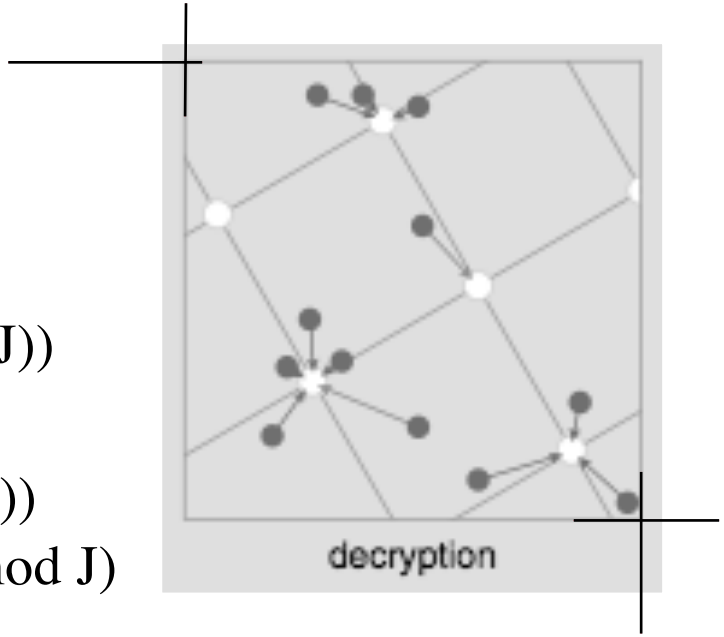
$$D(E(m_1,r_1) \bullet E(m_2,r_2) \bmod n^2) = m_1 + m_2 \bmod n$$
$$D(E(m_1,r_1)^{m_2} \bmod n^2) = m_1 \bullet m_2 \bmod n$$

Craig Gentry, 2009
Full Homomorphic encryption

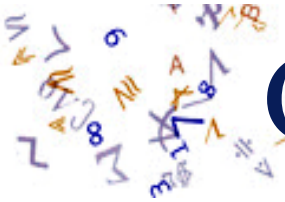
- Based on lattice encryption
  - E: $c=m+i \pmod J$

$$c_1+c_2 = (m_1+i_1 \pmod J)+(m_2+i_2 \pmod J)$$
$$= (m_1 + m_2) + (i_1 + i_2) \pmod J$$

$$c_1 \cdot c_2 = (m_1+i_1 \pmod J)\cdot(m_2+i_2 \pmod J)$$
$$= (m_1 \cdot m_2)+(i_1 \cdot i_2+i_1 \cdot m_2+i_2 \cdot m_1) \pmod J$$



- Error increase at each operation and we will not be able to decrypt message.
- Not a really fully homomorphic scheme !



Craig Gentry, 2009
Full Homomorphic encryption

- A solution: refesh data
  - Decrypt the resulting data, and encrypt it back to remove noise.
  - Can't be done in an insecure context!

- The solution
  - When decrypt is run in *cipherspace* with enciphered key, the result is not plaintext but a new encryption of the ciphertext, with reduced noise.

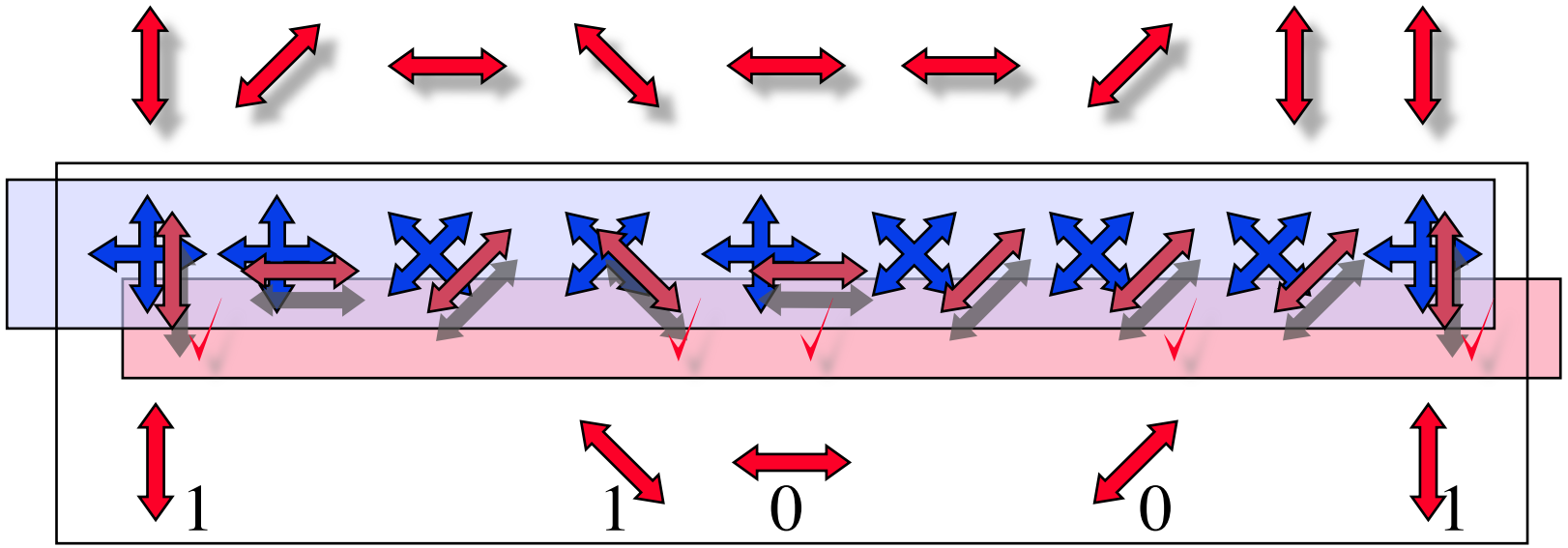
- Remaining problems:
  - decrypt routine must not accumulate excessive noise
    - Imply usage of very large keys
    - Implementation is very inefficient

Quantum Cryptography

- Goal:
  - Check that nobody is aware of communication.
  - Quantum mechanics tells us that the acquisition of information on the system inevitably disturbs it.
- Protocols
  - Exchange of « secret » data
  - Remove compromised or uncertain data..
- Methods
  - Photons polarization
  - Temporal encoding

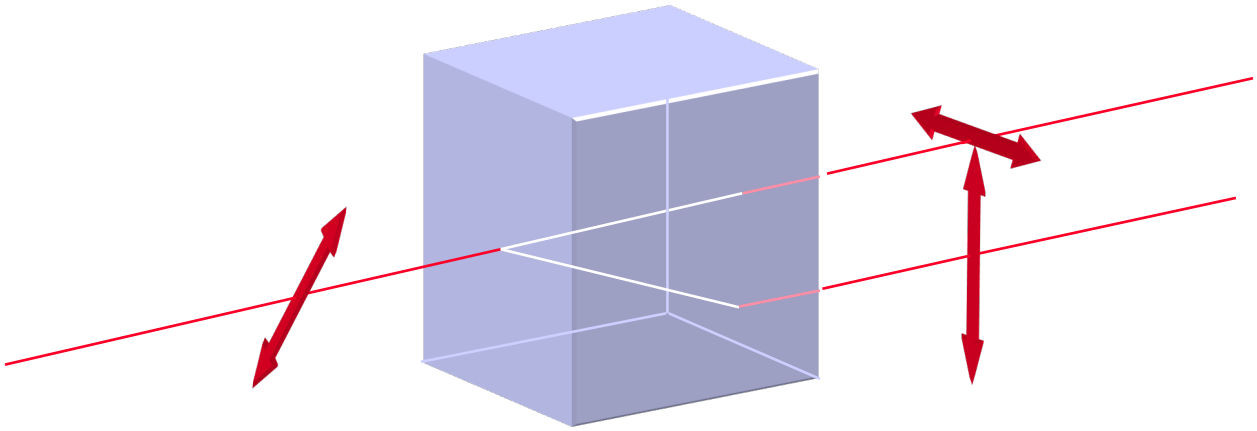
BB84 (Bennett & Brassard)

- Quantique channel
  - Polarised light à 0° , 45° , 90° ou 135°
- Classical channel



BB84 (Bennett & Brassard)

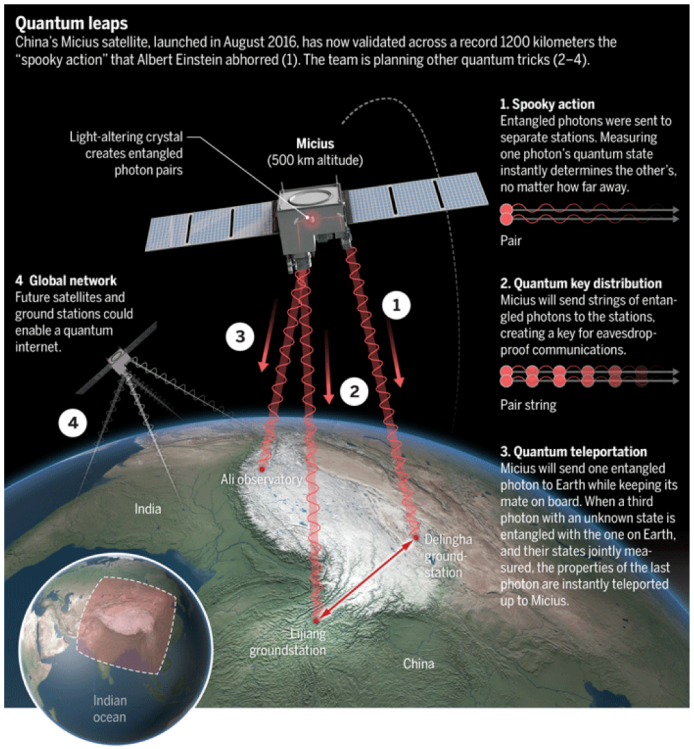
- Photons, in an intermediate polarization, are in a superposition of the two polarizations.



Calcite cristal

E91 (Artur Ekert, 1991)

- Entangled pairs of photons.
  - EEntangled states are perfectly correlated
  - Any attempt at eavesdropping destroys these correlations and can be detected.
- Protocol
  1. Measure each received photon using one basis (+ , x)
  2. keep photons measured using the same basis.
  3. Eavesdropping if violation of Bell's Theorem.



https://www.sciencemag.org/news/2017/06/china-s-quantum-satellite-achieves-spooky-action-record-distance

- If the protocol is successful, use results to generate keys.

References

- Livres
  - Histoire des codes secrets - Simon Singh - J.C. Lates - 1999.
  - Cryptographie Appliquée - Bruce Schneier - Thomson publising - 1996.
  - Cryptographie, Théorie et pratique - Douglas Stinson - Thomson publising - 1996.
  - Initiation à la Cryptographie - Gilles Dubertret - Vuibert - 1998.
  - Cours de cryptographie - Gilles Zémor - Cassini - 2000.
- Revues
  - Dossier pour la science : l'art du secret. Juillet/oct 2002.
- Web
  - http://csrc.nist.gov/

This is the end