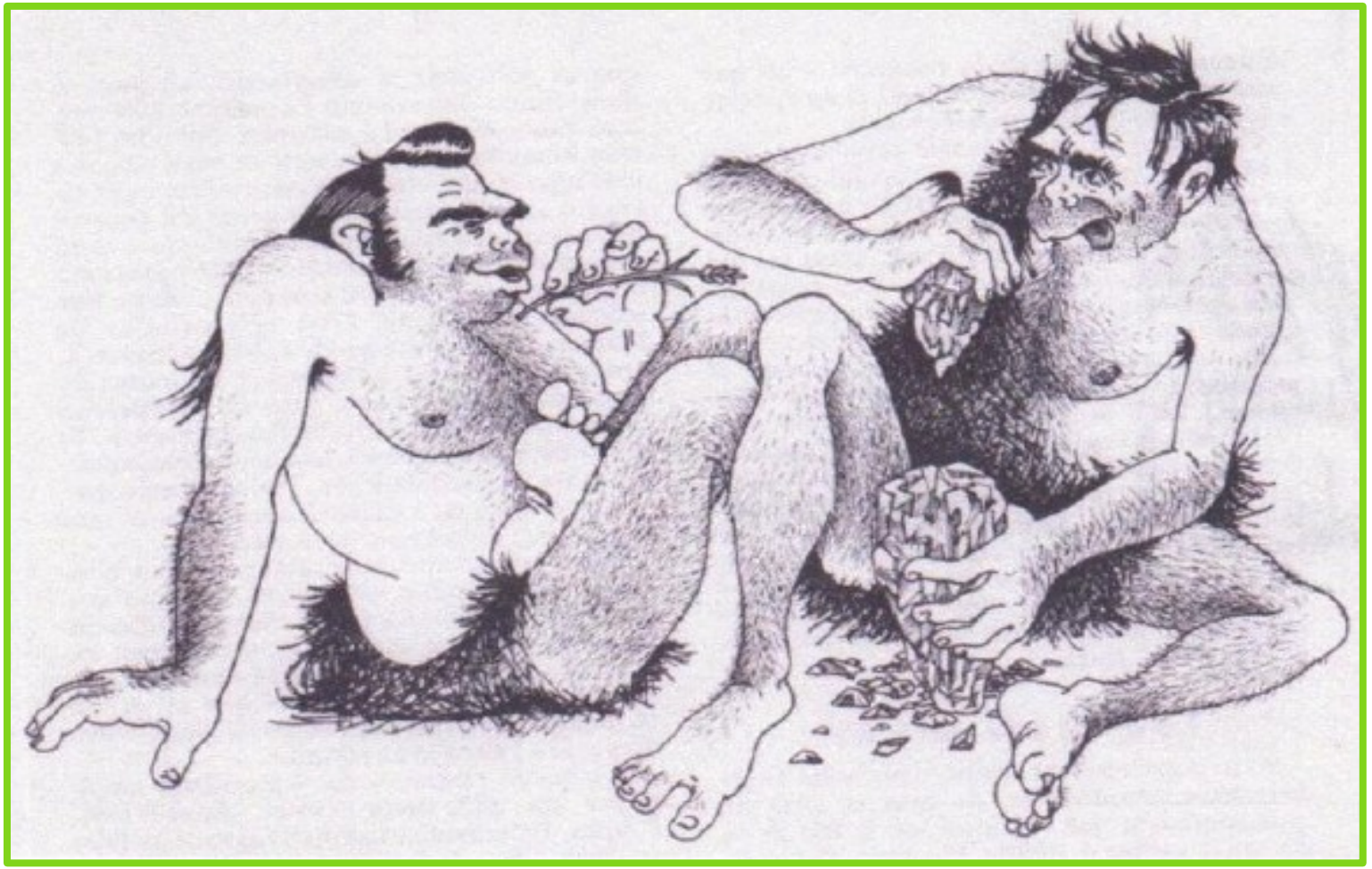
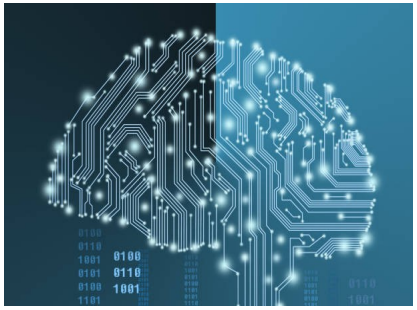
Un Brin de Cryptographie : le RSA en Action

Ou : « Cachez ce Secret que je Saurai Voir ! »

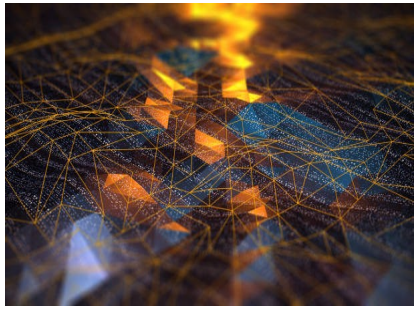


Crois-moi, Jojo, la cryptographie, c'est l'avenir !

Un Brin de Cryptographie : le RSA en Action
Ou : « Cachez ce Secret que je Saurai Voir ! »



S O M M A I R E



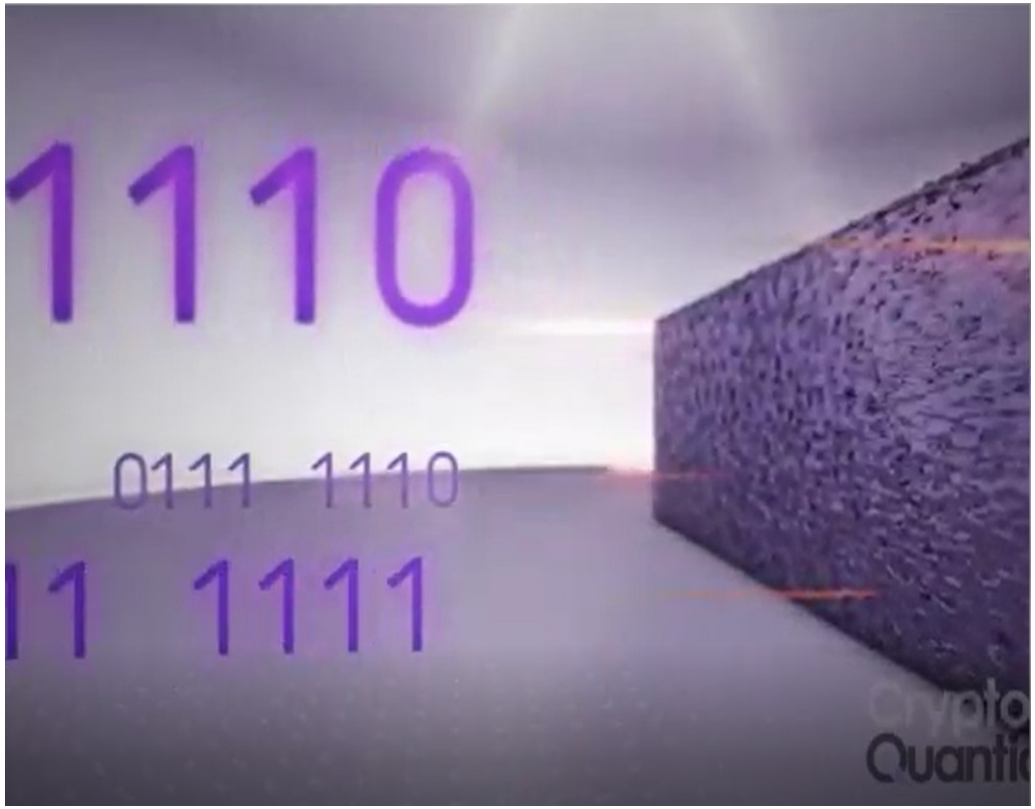
1

La CRYPTOLOGIE, Art Ancien, Science Nouvelle ...

La **cryptologie** n'est pas un domaine nouveau.

Son origine remonte à l'Antiquité gréco-romaine ...

La **cryptologie** a connu un rapide essor au XXème siècle, surtout à partir de l'arrivée des ordinateurs … Elle utilise aujourd'hui largement l'outil mathématique.



1-1

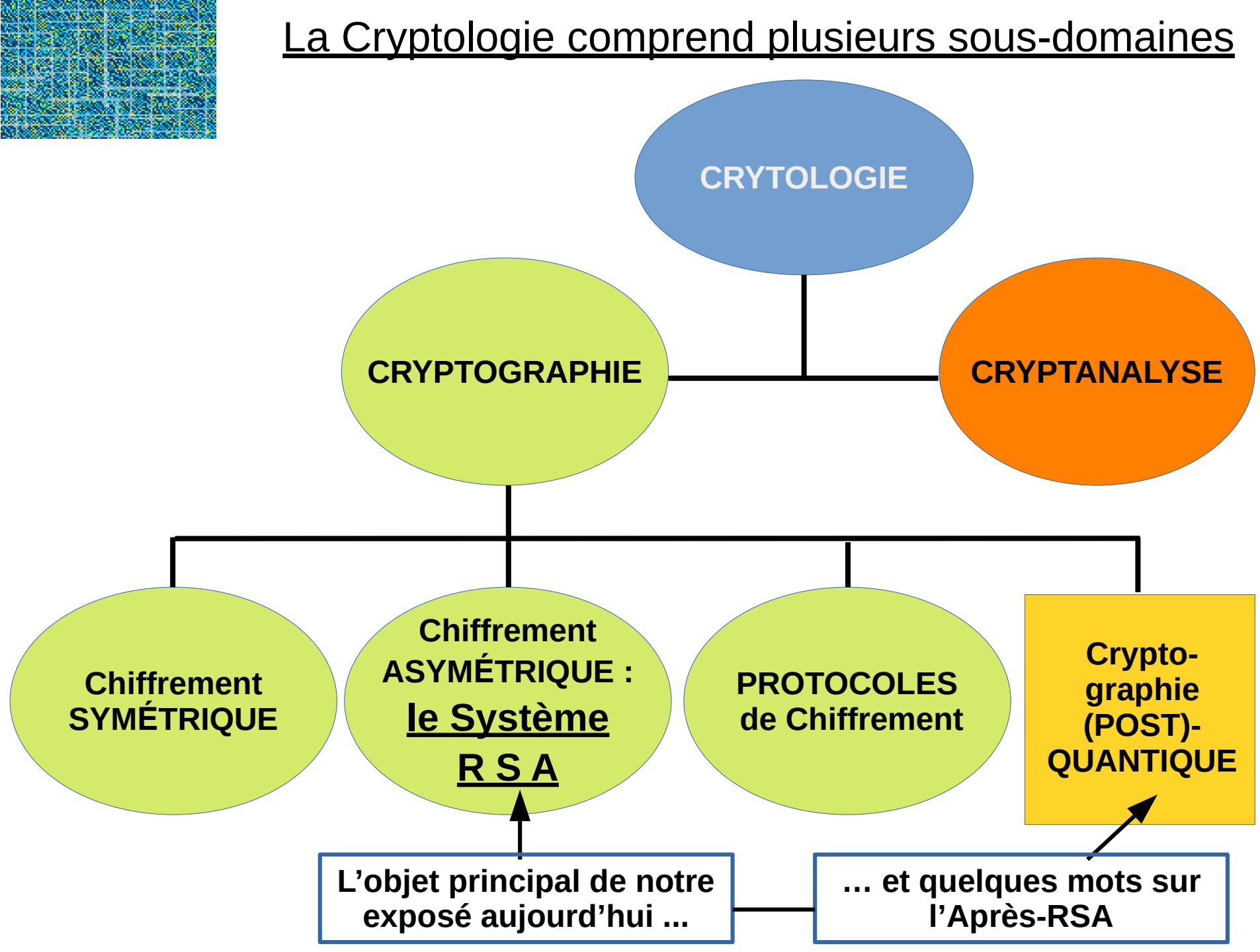
Les DOMAINES de la CRYPTOLOGIE … et la PLACE du RSA

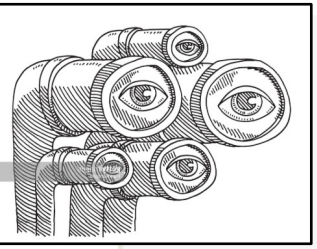
Les deux principaux domaines de la **cryptologie** :

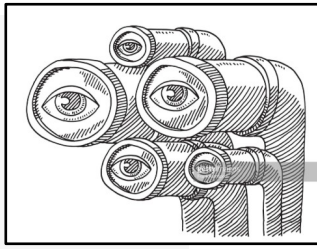
- la **cryptographie**, ou **chiffre de défense** rassemble les techniques de chiffrement (ou cryptage), donc de *défense*, contre les intrusions,

- la **cryptanalyse**, ou **chiffre d'attaque** comprend les techniques de déchiffrement (ou décryptage), donc d'*attaque*, au sens de *découverte* des informations chiffrées.

La Cryptologie comprend plusieurs sous-domaines







Les Usages de la Cryptographie, d'après la CNIL
(https://www.cnil.fr/fr/comprendre-les-grands-principes-de-la-cryptologie-et-du-chiffrement)

Le RSA, objet de notre exposé, se situe ici ...

1-2 RAPPEL des ÉTAPES de la CRYPTOGRAPHIE, de l'ANTIQUITÉ à NOS JOURS

1-2-1 La Cryptographie de l'Antiquité à nos jours

On peut diviser l'histoire de la cryptographie jusqu'à la fin du XXième siècle en trois grandes périodes :

1 - L'**ère artisanale**, de l'Antiquité à la fin du XIXième siècle, avec la **stéganographie**, le **chiffrement symétrique** par **substitution ou transposition** (Codes **CÉSAR**, de **VIGENÈRE**)

Blaise de VIGENÈRE
(1523-1596)

2 - L’**ère de la radio et de la mécanisation du codage**, jusqu’au milieu du XXième siècle, avec la découverte de la radio par Marconi, le code **ADFGVX**, la machine **ENIGMA** ...

La Machine ENIGMA

3 - L'**ère moderne**, depuis **1950**, marquée :

- par l'utilisation de l'**outil mathématique** (arithmétique modulaire, courbes elliptiques, etc),

- par l'usage de l'**outil informatique**,

- enfin par le grand évènement qu'a été l'**échange de clés Diffie-Hellmann** (**1976**), suivi du 1er système de **chiffrement asymétrique RSA** (**1978**).

1-2-2 La Cryptographie dans la Période Actuelle

Vers la fin du XXème et au début du XXIème siècles, les enjeux de la protection des données portent sur :

- L'**ordinateur** et la **cryptographie quantiques**,

- La **cryptographie post-quantique**, dont l'objectif est de *contrer* la cryptographie quantique.

2 APERÇU de QUELQUES OUTILS MATHÉMATIQUES en CRYPTOGRAPHIE à CLÉ PUBLIQUE

► A ses débuts, la cryptographie **à clef publique** a utilisé des **notions** et **résultats** d'*arithmétique* et d'*algèbre* :

- divisibilité, congruences,
- nombres premiers,
- groupes, corps finis, etc.

► Plus tard, elle utilisera des résultats de la *géométrie algébrique* (courbes elliptiques).

2-1 DIVISION EUCLIDIENNE, PGCD et CONGRUENCE

2-1-1 Division Euclidienne

La **division euclidienne** se définit à partir du résultat suivant :

Div-1 (théorème de la division dans $\mathbb{Z}$)

Quels que soient les entiers a et b, $b \neq 0$, il existe des entiers *uniques* q et r, éléments de $\mathbb{Z}$, avec $0 \leq r \leq b$, tels que l'on ait :

$$[1] \qquad \boxed{a = bq + r}$$

L'opération qui vérifie l'égalité [1] s'appelle la **division euclidienne** de a par b.

L'entier q s'appelle le **quotient** et l'entier r s'appelle le **reste** de cette division .

2-1-2 PGCD

Soient **a**,**b**$\in$Z des entiers. L'entier **d**$\in$Z, noté **pgcd(a,b)**, ou simplement **(a,b)**, est appelé

PLUS GRAND COMMUN DIVISEUR ou **PGCD**

de **a** et de **b** si, et seulement si,

(1) **d**|**a** et **d**|**b**

(2) si **c**|**a** et **c**|**b**, alors **c** ≤ **d**

2-1-3 L'Algorithme d'EUCLIDE

Cet algorithme donne le **PGCD** de deux entiers … On peut l'exécuter :

- ''à la main'', par **divisions successives** ou par **soustractions successives**,
- à l'aide d'un logiciel de calcul scientifique ...

Lors du calcul ‘’à la main’’ du PGCD, la propriété suivante s’applique :

Div-2

Si **a** = **bq**+**r**, alors le PGCD **d** est l’entier

$$PGCD(\mathbf{a},\mathbf{b}) = (\mathbf{a},\mathbf{b}) = PGCD(\mathbf{b},\mathbf{r}) = (\mathbf{b},\mathbf{r})$$

Autrement dit :

Si **b** est un diviseur de **a**, alors le PGCD de **a** et de **b** (dividende et diviseur) est aussi le PGCD de **b** et de **r** (diviseur et reste)

Exemple : Soient les entiers a = 46848 et b = 2379.

Colonne 1 : Expression de la division euclidienne

Colonne 2 : Expression de la propriété DIV-2

Ligne	$a = b \cdot q + r$	Application de $(a,b) = (b,r)$
1	$46848 = 2379 \cdot 19 + 1647$	$(46848, 2379) = (2379, 1647)$
2	$2379 = 1647 \cdot 1 + 732$	$(2379, 1647) = (1647, 732)$
3	$1647 = 732 \cdot 2 + 183$	$(1647, 732) = (732, 183)$
4	$732 = \boxed{183} \cdot 4 + 0$	$(732, 183) = (183, 0)$

PGCD(a,b) = **183** = Dernier reste NON NUL

2-1-4 Théorème de BEZOUT

La justification théorique de

l'algorithme d'Euclide est le

théorème de Bezout :

BEZOUT 1

Si **d** = (**a,b**) est le **PGCD** des entiers **a** et **b**, alors

il existe deux entiers **x** et **y**, tels que :

$$d = (a,b) = ax + by$$

Un cas particulier important du **théorème de Bezout**

est celui où l'entier **d** de l'équation

$$ax+by = d$$

est égal à **1**.

Le résultat suivant sera utilisé dans le système RSA :

BEZOUT 2 (Identité de Bezout)

Deux entiers a et b sont **PREMIERS ENTRE EUX** si, et seulement si, il existe des entiers $x, y \in \mathbb{Z}$ tels que $ax + by = 1$:

$$(a, b) = 1 \Leftrightarrow ax + by = 1, \text{ pour } x, y \in \mathbb{Z}$$

2-1-3 Une définition *simple* de la Relation de Congruence

La relation de **congruence** entre nombres entiers se définit à partir de la **division euclidienne**.

Deux entiers **x** et **y** sont **congrus modulo n** s'ils vérifient la condition suivante :

x et y ont le même RESTE dans la division euclidienne par n

L'entier **n** est appelé le **module** .

2-1-4 OPÉRATIONS dans l'Ensemble Z_n d'Entiers modulo *n*

Les opérations (modulo **n**) *usuelles* dans un ensemble Z_n sont l'addition, la multiplication et l'exponentiation :

Soient a, b, c, d, **n** ϵ $\mathbb{N}$. Si :

$$\begin{cases} \mathbf{a} \equiv \mathbf{c} \pmod{\mathbf{n}} \\ \mathbf{b} \equiv \mathbf{d} \pmod{\mathbf{n}} \end{cases}$$, alors :

- $\mathbf{a} + \mathbf{b} \equiv \mathbf{c} + \mathbf{d} \pmod{\mathbf{n}}$ (**addition**)
- $\mathbf{a} \times \mathbf{b} \equiv \mathbf{c} \times \mathbf{d} \pmod{\mathbf{n}}$ (**multiplication**)

Soient **n**, **k** ϵ $\mathbb{N}$. Si $\mathbf{a} \equiv \mathbf{b} \pmod{\mathbf{n}}$, alors :

- $\mathbf{a}^{\mathbf{k}} \equiv \mathbf{b}^{\mathbf{k}} \pmod{\mathbf{n}}$ (**exponentiation**)

2-2 NOMBRES PREMIERS et THÉORÈME de FERMAT

2-2-1 COMMENT SAVOIR SI UN ENTIER EST PREMIER

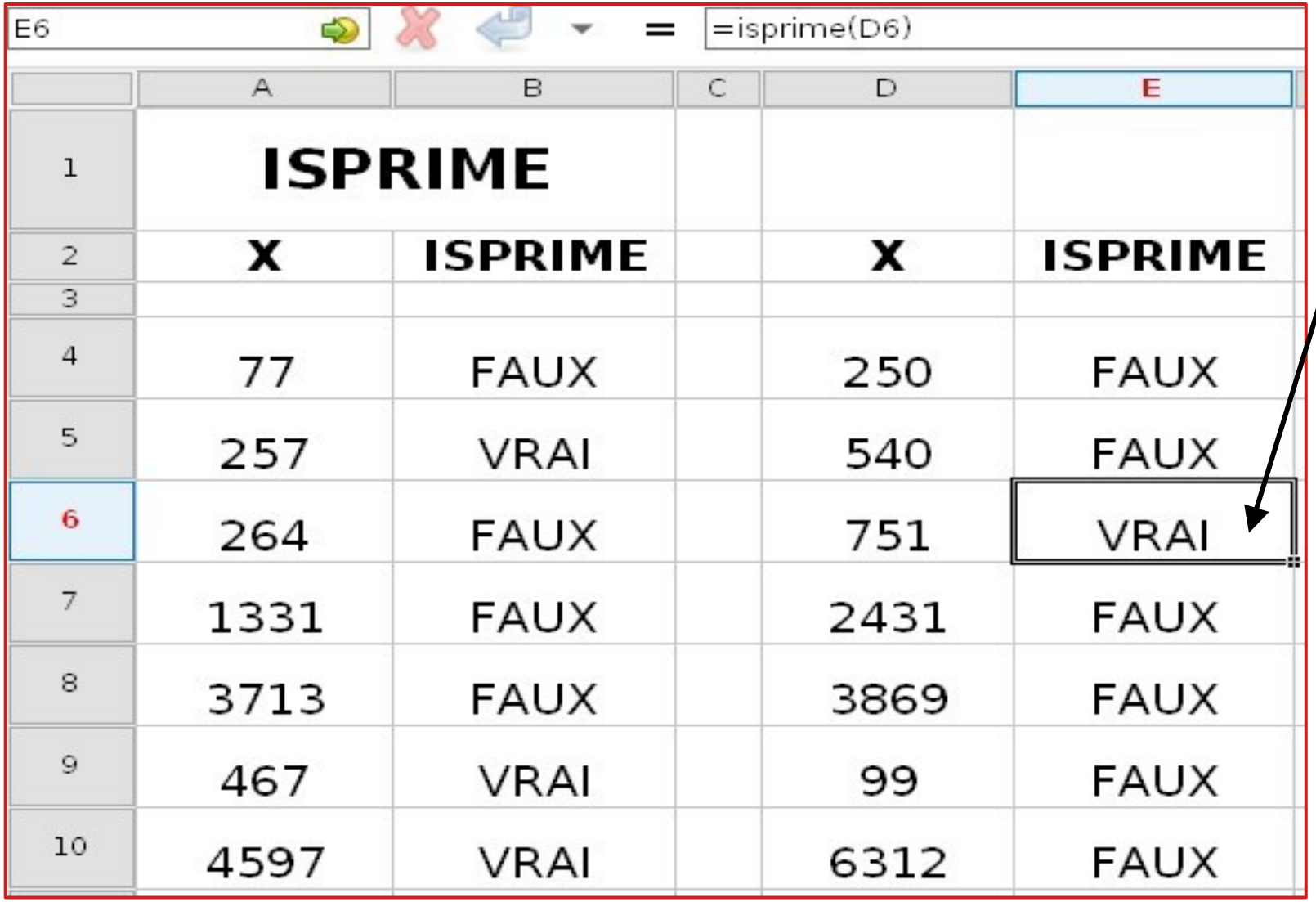
Pour déterminer si un entier **n** est *premier* ou *composé*, plusieurs moyens sont disponibles …

► On peut recourir à un tests de primalité (déterministe ou probabiliste) et l'on peut aussi utiliser, entre autres, l'un des outils suivants :

- les tableurs (libres) comme CALC de LIBREOFFICE, GNUMERIC, CLASSEUR d'ONLYOFFICE;
- les logiciels de calcul comme MAXIMA, OCTAVE, SAGE, SCILAB, XCAS ;
- les langages de programmation comme C, JULIA, PYTHON, etc.

► … Le plus simple est d'utiliser un tableur ...

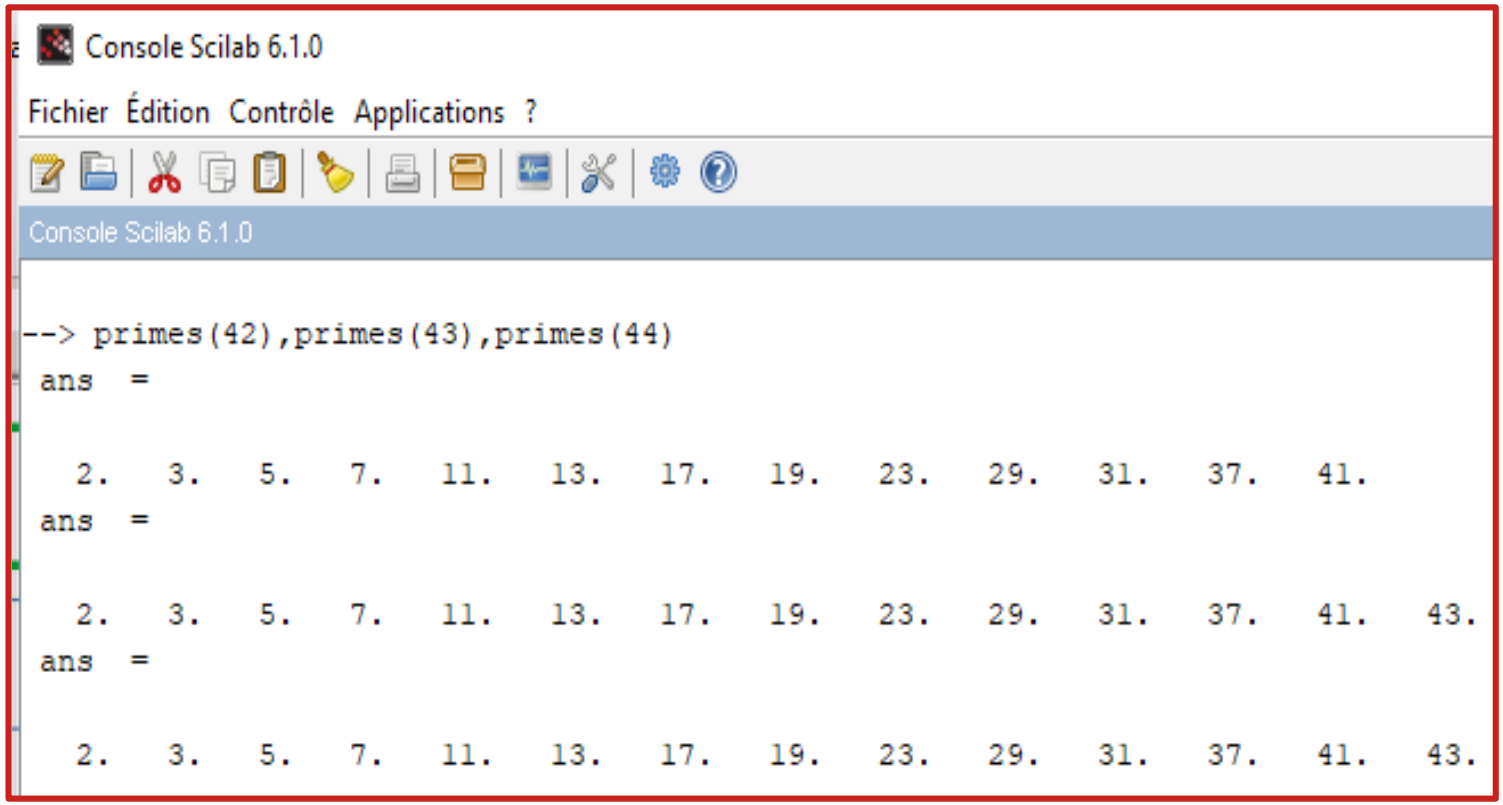
Par exemple avec le tableur **GNUMERIC** et sa fonction **isprime**, qui répond VRAI ou FAUX, suivant que **x** *est* ou *n'est pas* premier …



	A	B	C	D	E
1	**ISPRIME**				
2	**X**	**ISPRIME**		**X**	**ISPRIME**
3					
4	77	FAUX		250	FAUX
5	257	VRAI		540	FAUX
6	264	FAUX		751	VRAI
7	1331	FAUX		2431	FAUX
8	3713	FAUX		3869	FAUX
9	467	VRAI		99	FAUX
10	4597	VRAI		6312	FAUX

► … on peut aussi facilement recourir aux logiciels de calcul ...

Par exemple, avec **SCILAB** et sa commande **primes** (x), qui renvoie la liste **P** des entiers premiers $y \leq x$: **si l'entier x se trouve dans cette liste, alors x est premier**.



Les entiers x=**42** et x=**44** sont *composés*, car ils ne sont pas dans la liste P retournée par la commande **primes**.
L'entier x=**43** est un entier *premier*, car il figure dans la liste **P**.

2-2-2 – Le THÉORÈME de FERMAT

♣ Rappelons l'énoncé du (petit) **théorème de FERMAT** :

Si **p** est un entier **premier**, alors pour tout élément $\bar{a} \in \mathbb{Z}_p$, $\bar{a} \neq 0$, on a :

$$\bar{a}^{p-1} = \bar{1} \quad \text{ou} \quad a^{p-1} \equiv 1 \pmod{p}$$

C7 | =MOD((C2^A7);5)

	A	B	C	D	E
1		EXPONENTIATION dans Z⁵			
2	Éléments a →	1	2	3	4
3	Puissance a^b v				
4	1	1	2	3	4
5	2	1	4	4	1
6	3	1	3	2	4
7	4	1	1	1	1
8	5	1	2	3	4

Éléments **a** de Z_5

♣ On peut illustrer ce théorème en dressant la table d'exponentiation de $\mathbf{Z_p}$, ...

Exemple : Dans la table de $\mathbf{Z_p}=\mathbf{Z_5}$, on a :

$$1^4 = 1 \equiv 1 \pmod 5,$$
$$2^4 = 16 \equiv 1 \pmod 5,$$
$$3^4 = 81 \equiv 1 \pmod 5,$$
$$4^4 = 256 \equiv 1 \pmod 5,$$

Dans chaque cellule, on a : a^b modulo p

Pour l'exposant **b** = **p-1** = 4, le reste modulo **p** est égal à **1**.

2-3 La FONCTION φ (phi) et le THÉORÈME d'EULER

► La fonction d'EULER **φ(n)** est un *outil* indispensable dans le système RSA :

Pour un entier **n**, la **fonction φ (phi) d'Euler**, ou **fonction indicatrice d'Euler**, notée **φ(n)**, renvoie le nombre **k** des entiers **x** de l'intervalle [1,**n**[qui sont **premiers** avec **n** :

φ(n) = nombre **k** d'entiers **x** vérifiant :

$0 \leq x < n$ et **pgcd(n,x) = 1**

Leonhard Euler
1707 - 1783

► La **fonction φ (phi) d'EULER** a de nombreuses propriétés intéressantes ...

φ0 Si **p** est un entier **premier**, alors :

$$\varphi(p) = p - 1$$

La suivante joue un rôle important dans le fonctionnement du cryptosystème **RSA** :

φ1 Si **n** = **p.q** est le produit de deux entiers **premiers** distincts **p** et **q**, alors : $\varphi(n) = (p - 1)(q - 1)$

► Le **théorème d'EULER** *généralise* le théorème de Fermat :

φ2 Pour tout module **n**, si **a** est un entier premier avec **n** (un **élément résiduel** de $\mathbf{Z}_n$), alors on a :

$$a^{\varphi(n)} \equiv 1 \pmod{n}$$

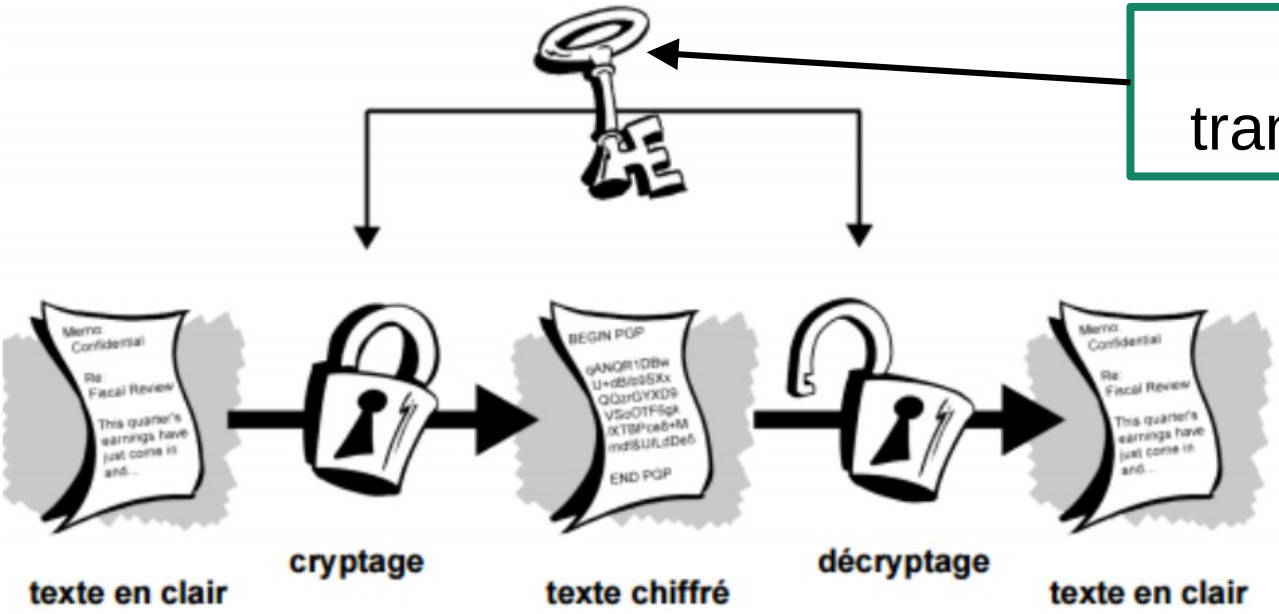
3 La RÉVOLUTION de la CRYPTOGRAPHIE à CLEF PUBLIQUE

- Tous les algorithmes de la cryptographie, depuis l'Antiquité jusqu'en 1976, reposaient sur des méthodes de chiffrement dit **asymétrique**, qui utilisent une **clé privée** (ou **clé secrète**) à la fois pour le *chiffrement* et pour le *déchiffrement*.

Ces méthodes, (3DES, SHA, etc), très rapides, sont encore très employées aujourd'hui.

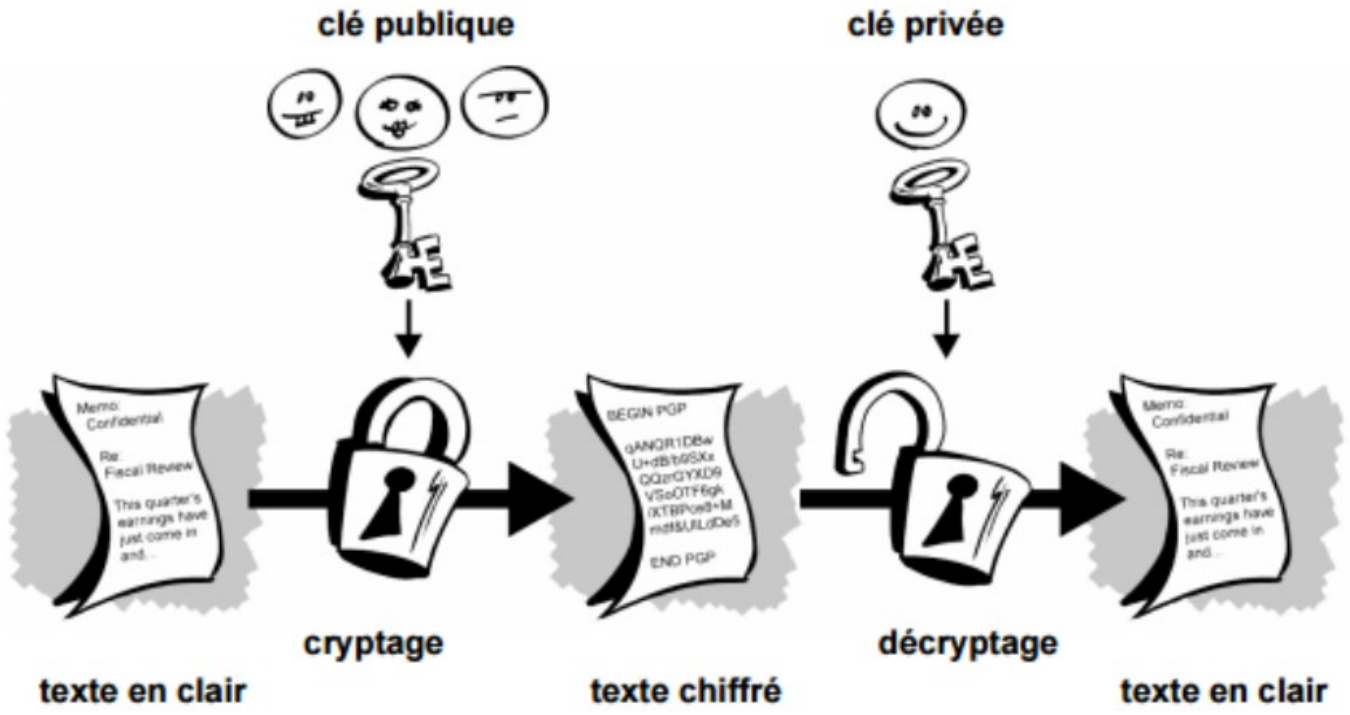
3-1 ORIGINE de la CRYPTOGRAPHIE à CLÉ PUBLIQUE

► L'inconvénient des méthodes de chiffrement à clé **privée**, c'est que la transmission de la clé peut être interceptée !



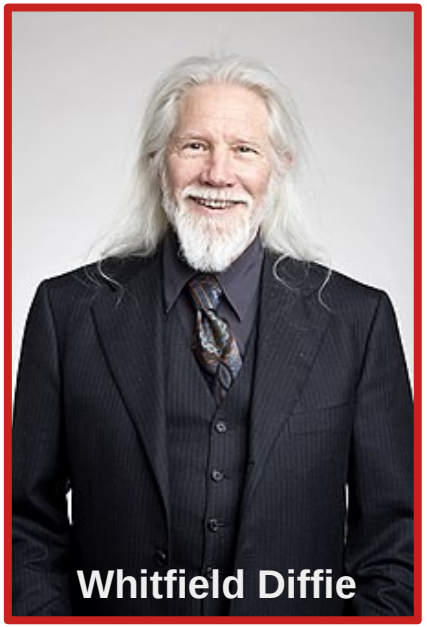
► Le problème de la cryptographie à clé privée, c'est donc avant tout **la protection de la clé elle-même** !

► A partir de 1976 sont apparus les algorithmes de chiffrement dit **asymétrique**, (Diffie-Hellman, El-Gamal, RSA, RSA elliptique, etc), qui utilisent, en plus de la clé privée, une **clé publique**.



►**1976** : Naissance de la notion de **cryptographie à clef publique**

Whitfield Diffie et **Martin Hellman**, dans un article *fondateur* (*), proposent le concept de **clef publique** et ouvrent ainsi la voie à un nouveau type de chiffrement, la cryptographie à clef publique.


Whitfield Diffie


Martin Hellmann

Cette technologie sera de fait accessible à tout émetteur d'information sur le *réseau public*, et non plus réservée aux seuls organismes ''officiels''.

(*) New directions in cryptography

3-2 Le PROTOCOLE de DIFFIE-HELLMAN

Le protocole de DIFFIE-HELLMAN est à l'origine des systèmes de

chiffrement asymétrique,

qui reposent sur deux clés :

. une **clé privée** (**secrète**),

. une **clé publique**.

Ces deux clés sont utilisées à la fois pour le *chiffrement* et pour le *déchiffrement*.

Le but du protocole de **DIFFIE-HELLMAN** est de

sécuriser le transfert de clé,

donc de résoudre

le problème de la **distribution d'une clé secrète**

servant à chiffrer/déchiffrer un message.

3-3 Schéma d'un Système de Cryptage à CLÉ PUBLIQUE

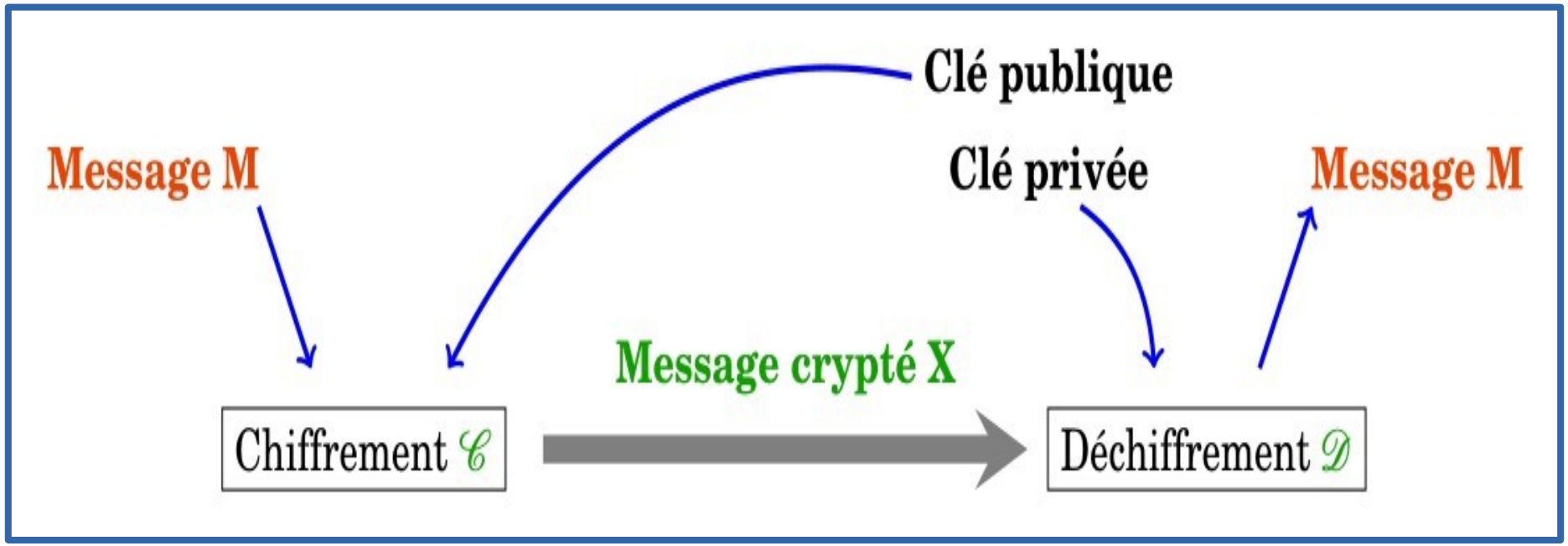
Les protagonistes d'un **cryptosystème à clé publique** sont l'*émetteur* et le *récepteur* du message (**A**lice et **B**runo) :

Alice

Alice et **B**runo utilisent :
- une **clé privée** et
- une **clé publique**

pour chiffrer et pour déchiffrer.

Bruno

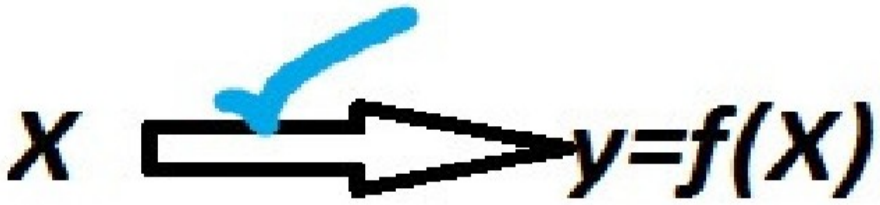


3-4 DÉROULEMENT du PROTOCOLE de DIFFIE-HELLMAN

3-4-1 FONCTION à SENS UNIQUE

Le protocole de DIFFIE-HELLMAN repose sur la notion de **fonction à sens unique**. (*)

Une fonction *qui n'est pas inversible* ou qui n'est inversible qu'au prix de très longs calculs est appelée **fonction à sens unique**.



(*) ou ''fonction difficilement inversible'' ; ang. : one-way function

► Exemples de Fonctions à Sens Unique

1) Mot de passe : trouver un mot de passe y = f(m)

Si un mot de passe **m** est stocké en machine sous la forme **y** = **f(m)**, il peut être difficile (et très long) de trouver **m** à partir de **y**.

2) Multiplication : trouver les facteurs x et y d'un produit (**FACT**)

Étant donné un produit **p** = **x.y**, il est difficile (et très long) de trouver les facteurs **x** et **y** de **p**, surtout si le produit **p** est un grand nombre et, en particulier, si **x** sont de grands nombres premiers.

3) Exponentiation : calcul du logarithme discret (**DLOG**)

Si l'élément **x** d'un ensemble G à **n** éléments s'écrit sous la forme

$$x = g^{\alpha},$$

l'élément **g** est appelé le **générateur** de G et

$$\alpha = \log_g(x)$$

s'appelle le **logarithme discret de x en base g**.

Calculer $\alpha = \log_g(x)$ revient à résoudre l'équation

$$\alpha g = x \text{ (modulo } n)$$

… ce qui est un problème réputé très difficile !

3-4-2 Les ÉTAPES du PROTOCOLE de DIFFIE-HELLMAN

- **ÉTAPE PRÉLIMINAIRE :** Alice et Bruno s'entendent d'abord sur le choix d'une **fonction à sens unique**

- Ensuite le Protocole entre Alice et Bruno compte 4 étapes :

1ÈRE ÉTAPE : Choix des ENTIERS SECRETS

2ÈME ÉTAPE : Application d'une *FONCTION à SENS UNIQUE*

3ÈME ÉTAPE : ÉCHANGE entre Alice et Bruno

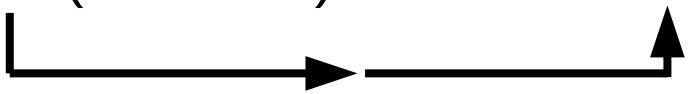
4ÈME ÉTAPE : RÉCUPÉRATION de la CLEF PRIVÉE

3-4-3

EXEMPLE d'ÉCHANGE de COUPLE (CLÉ *SECRÈTE* , CLEF *PUBLIQUE*) suivant le protocole de DIFFIE-HELLMAN

Les protagonistes de l'échange de clefs sont :

Alice (émetteur) et **B**runo (récepteur).



Alice et Bruno choisissent une **fonction à sens unique f** :

► $f(x) = G^x \ (mod\ P) = 5^x \ (mod\ 13)$

(Avec **G<P** et **P** premier ; (ici : **G=5** et **P=13**)

► La fonction **f(x)** et les entiers **G=5** et **P=13** ne sont pas secrets !

Les premières valeurs de la fonction

f(x) = G^x (*mod* P) = 5^x (*mod* 13) :

5^x	5	25	125	625	3125	15625	78125	390625	1953125	9765625
5^x (mod 13)	5	12	8	1	5	12	8	1	5	12

Il n'y a aucune régularité dans l'ensemble des *images (valeurs)* de cette fonction f(x) qui permette de déduire la valeur de l'*antécédent* x.

Exemple : **5** est l'image de **5**, de **3125** et de **1953125**.

Autrement dit :

la fonction **f(x) = G^x (*mod* P) = 5^x (*mod* 13)**

est bien une **fonction à sens unique.**

► **1ÈRE ÉTAPE du protocole de DIFFIE-HELLMAN :**

Choix des Entiers Secrets

Alice choisit un entier, par exemple : **A = 8**

Bruno choisit un entier, par exemple : **B = 11**

Les entiers **A** et **B** sont gardés **secrets** !

► 2ÈME ÉTAPE du protocole de DIFFIE-HELLMAN :

Application de la Fonction à Sens Unique

La fonction choisie est cette fois :

$f(x) = G^x \ (mod\ P) = 5^x \ (mod\ 17)$

… avec **P=17**. Elle a pour image :

Pour **A**lice (x=A=**8**) :

$f(A) = 5^A \ (mod\ 17)$

$= 5^8 \ (mod\ 17)$

$= 390\,625 \ (mod\ 17)$

$= 16$

390 625 = 22 977 x 17 + 16

Pour **B**runo (x=B=**11**) :

$f(B) = 5^B \ (mod\ 17)$

$= 5^{11} \ (mod\ 17)$

$= 48\,828\,125 \ (mod\ 17)$

$= 11$

48 828 125 = 2872242 x 17 + 11

► **<u>3ÈME ÉTAPE</u> du protocole de DIFFIE-HELLMAN :**

l'ÉCHANGE entre Alice et Bruno ...

<u>A</u>lice nomme le résultat de la 2ème étape

α = 16

et l'envoie à **B**runo.

<u>B</u>runo nomme le résultat de la 2ème étape

β = 11

et l'envoie à **A**lice.

Les résultats **α = 16** et **β = 11** sont échangés de façon <u>non protégée</u> !

C'est lors de cette étape qu'**<u>E</u>ve** (la vile espionne) a la possibilité d'*intercepter* la communication entre **A**lice et **B**runo et de connaître ainsi les nombres

α = 16 et **β = 11**.

<u>Mais c'est sans conséquence, car ces nombres *ne sont pas* la clef !</u>

4ÈME ÉTAPE du protocole de DIFFIE-HELLMAN :
Récupération (= calcul) de la CLEF PRIVÉE

Alice prend le résultat

$\beta = 11$

envoyé par **Bruno** et calcule :

$\beta^A \pmod P =$

$11^8 \pmod{17} =$

$214\,358\,881 \pmod{17} =$

16

Bruno prend le résultat

$\alpha = 16$

envoyé par **Alice** et calcule :

$\alpha^B \pmod P =$

$16^{11} \pmod{17} =$

$17\,592\,186\,044\,416 \pmod{17} =$

16

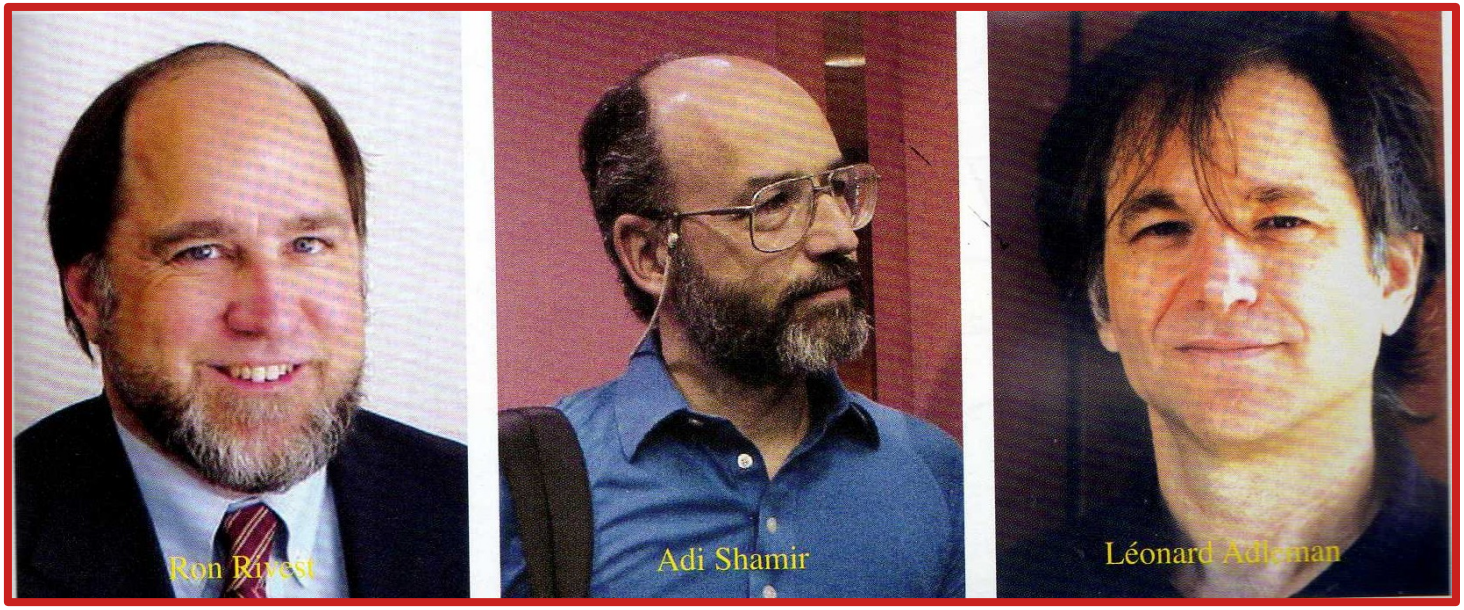
La **CLÉ PRIVÉE** est **C** = **16** !

Alice et **Bruno** ont seulement échangé l'information nécessaire à établir la clef privée (secrète), mais pas la clef elle-même !

4 Le FONCTIONNEMENT du SYSTÈME RSA

W. Diffie et M. Hellmann proposaient un *concept* et une *méthode*, mais pas de *système de chiffrement* proprement dit ...

1978 : **R**ivest, **S**hamir et **A**dleman mettent au point un *système* de **cryptographie à clef publique**, appelé **RSA**.



4-1 DÉROULEMENT *THÉORIQUE* du CHIFFREMENT RSA

► Dans le système **RSA**, le chiffrement comprend 6 étapes :

1ÈRE ÉTAPE : **Choisir la CLEF PRIVÉE de Chiffrement (p,q)**

2ÈME ÉTAPE : **Calculer la CLEF PUBLIQUE de Chiffrement (n,e)**

3ÈME ÉTAPE : **Calculer la CLEF PRIVÉE d de Déchiffrement**

4ÈME ÉTAPE : **Diffusion de la CLEF PUBLIQUE (n,e)**

5ÈME ÉTAPE : **CHIFFRAGE du Message M** par **A**lice

6ÈME ÉTAPE : **DÉCHIFFRAGE du Message C** par **B**runo

► **1ÈRE ÉTAPE du chiffrement RSA :**

Choix de la CLEF PRIVÉE de Chiffrement

Alice choisit deux entiers premiers, par exemple :

p = **17** et **q** = **11**

Les entiers premiers **p** et **q** sont gardés **secrets** !

Le couple d'entiers premiers

(p , q)

représente la **CLÉ PRIVÉE de Chiffrement**

2ÈME ÉTAPE du chiffrement RSA :

Calculer la CLEF PUBLIQUE de Chiffrement (n , e)

1) Alice commence par calculer le produit **n** :

$$n = p \times q = 17 \times 11 = 187$$

L'entier

$$n = p \times q = 187$$

Représente le 1er élément de la **CLÉ PUBLIQUE** **(n , e)**

2) **A**lice choisit ensuite un *autre* entier, **e**, par exemple l'entier

e = 7

C'est l'**exposant de chiffrement**,

et le 2ème élément de la **CLÉ PUBLIQUE** **(n , e)**.

Cet entier **e** doit être tel que :

PGCD(e , φ(n)) = 1

Autrement dit :

L'exposant de chiffrement **e** et **φ(n)**

sont des entiers premiers entre eux.

φ(n) est la **fonction d'EULER**

avec laquelle on a : **φ(n) = φ(pq) = (p - 1)(q - 1)**

► **3ÈME ÉTAPE du chiffrement RSA :**

Calcul de la CLEF PRIVÉE d de Déchiffrement

La clef privée **d** servira à *décrypter* le message.

La clef privée **d** doit vérifier l'équation modulaire :

$$e \times d \equiv 1 \pmod{\varphi(n)}$$

autrement dit, avec **e** = **7**, **p** = **17** et **q** = **11** :

$7 \times d \equiv 1 \pmod{(p-1)(q-1)}$, d'après la règle **φ1**

$7 \times d \equiv 1 \pmod{(16 \times 10)}$

$7 \times d \equiv 1 \pmod{160}$

A l'aide de l'**Algorithme d'Euclide**, on obtient la valeur de

d = **23**

► **4ÈME ÉTAPE du chiffrement RSA :**

Diffusion de la CLEF PUBLIQUE (n , e)

Alice *diffuse* la clef publique

(n , e) = (187 , 7)

La diffusion de **(n , e)** n'est pas protégée !

▶ 5ÈME ÉTAPE du chiffrement RSA :

CRYPTAGE du Message M par Alice

1) Alice commence par **convertir le message en clair M en un nombre**, à l'aide d'une fonction de conversion de son choix (code ASCII binaire ou autre)

2) Alice **crypte** ensuite le message **M** sous sa forme numérique pour obtenir le message *chiffré* **C**, suivant la formule de cryptage :

$$C = M^e \pmod{n}$$

Exemple :

Le message **M** est la lettre **X**

(Le message est ici *réduit* à une seule lettre afin de simplifier les calculs.)

Code ASCII binaire

A	100 0001	H	100 1000	O	100 1111	V	101 0110
B	100 0010	I	100 1001	P	101 0000	W	101 0111
C	100 0011	J	100 1010	Q	101 0001	X	101 1000
D	100 0100	K	100 1011	R	101 1010	Y	101 1001
E	100 0101	L	100 1100	S	101 0011	Z	101 1010
F	100 0110	M	100 1101	T	101 0100	a	110 0001
G	100 0111	N	100 1110	U	101 0101	b	110 0010

La lettre **X** est codée « **1011000** » en ASCII *binaire*, soit :

M = **88**

en système *décimal*.

La formule de cryptage :

$$C = M^e \pmod{n}$$

donne, avec **e** = **7** et **n** = **187** :

$$C = M^7 = 88^7 \pmod{187}$$

On effectue alors un calcul des puissances en arithmétique modulo **n** = **187**. Le résultat obtenu est le message chiffré :

$$C = 11$$

Pour ceux qui doutent, ... le détail du calcul des puissances :

Suivant la règle de calcul :

$$a^{p+q} = a^p \cdot a^q$$

on a :

$$88^7 \pmod{187} = [88^4 \pmod{187} \times 88^2 \pmod{187} \times 88^1 \pmod{187}]$$

$$88^1 = 88 = 88 \pmod{187}$$

$$88^2 = 7744 = 77 \pmod{187}$$

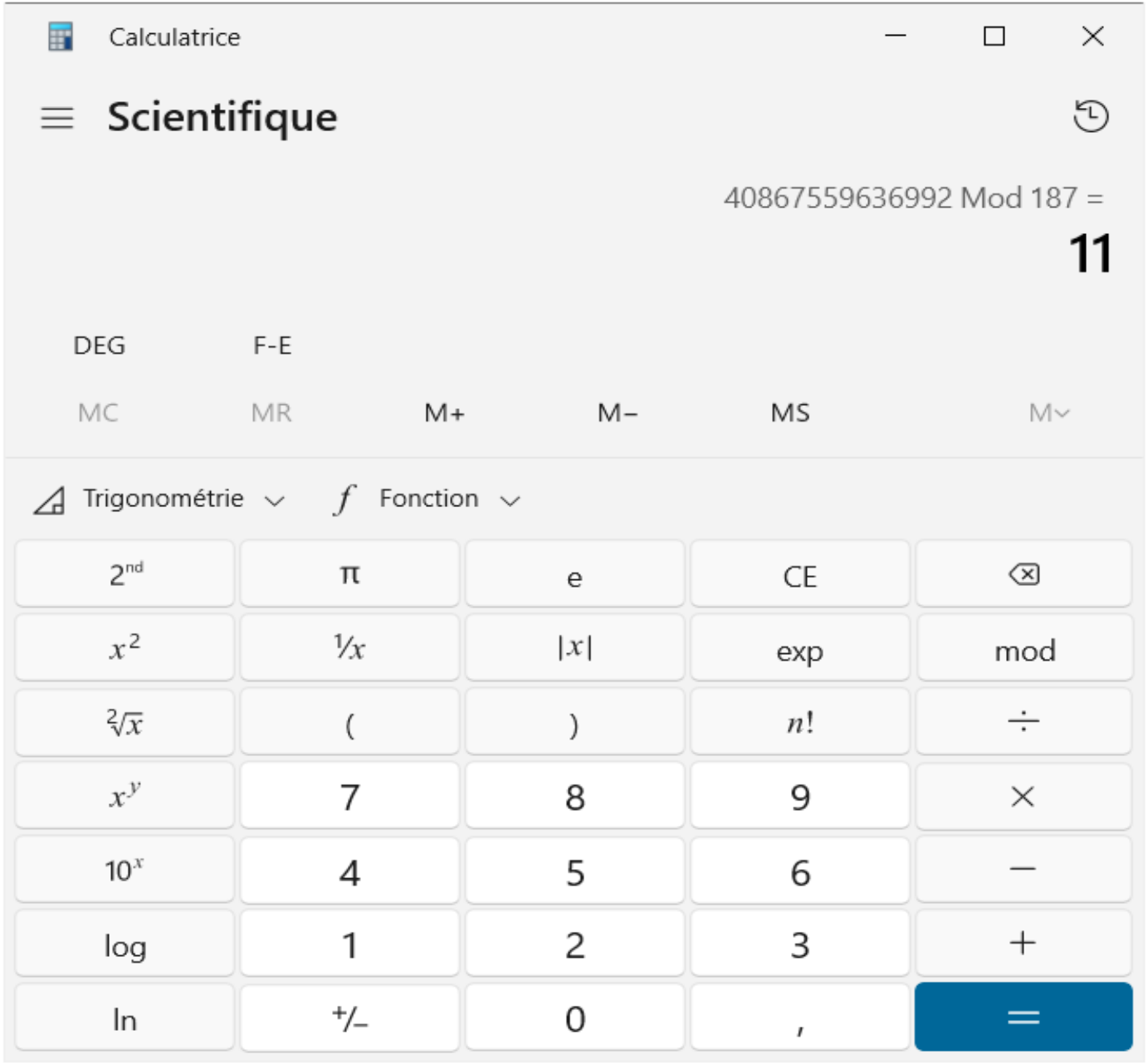
$$88^4 = 59\,969\,536 = 132 \pmod{187}$$

D'où :

$$88^7 = 88^1 \times 88^2 \times 88^4 \pmod{187} =$$

$$88 \times 77 \times 132 = 894\,432 = \mathbf{11} \pmod{187}$$

Ce qu’on vérifie aisément sur une calculatrice en mode scientifique :



6ÈME ÉTAPE du chiffrement RSA :

DÉCRYPTAGE du Message C par Bruno

1) Bruno, qui connaît $n = p \times q = 17 \times 11 = 187$ et $d = 23$ (qu'il a calculé par l'algo. d'Euclide) applique la formule de décryptage suivante : $M = C^d \pmod{187}$ avec $C = 11$

Cela donne :

$M = 11^{23} \pmod{187}$, d'où :

Décomposition de l'exposant
23 = 1+2+4+16

$$M = 11^1 \pmod{187} \times 11^2 \pmod{187} \times 11^4 \pmod{187} \times 11^{16} \pmod{187} =$$

$$M = (11 \times 121 \times 14641 \times 45\,949\,729\,863\,572\,161) \pmod{187} =$$

$$M = 895\,430\,243\,255\,237\,372\,246\,531 \pmod{187} = 88$$

Soit, en *ASCII*, le message en clair de départ : **X**

4-2 ESSAI de CHIFFREMENT RSA avec «MAPLE CLOUD»

► Sur le site **MAPLE CLOUD**, on peut réaliser en ligne le chiffrement **RSA** d'un message …

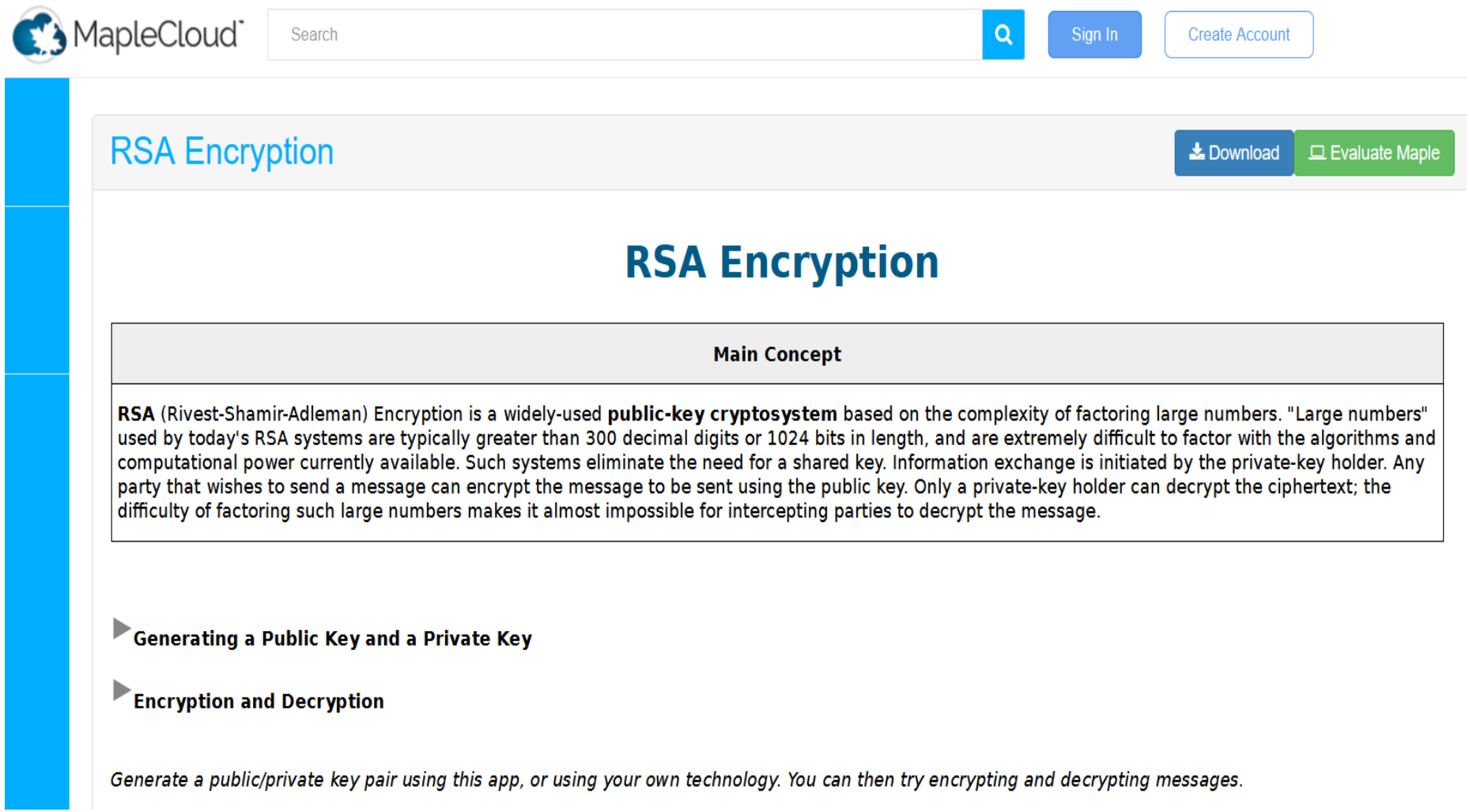
Puis son déchiffrement.

Les éléments de ces deux opérations sont affichés.

► L 'accès à MAPLE CLOUD est libre :

https://maple.cloud/app/5696938746839040

Le site se présente de la manière suivante :



Le visiteur lance les étapes du système **RSA** à partir des données choisies par lui.
Exemple ...

► 1) Clé Publique (n,e) et Clé Privée d du chiffrement RSA :

```
GENERATE TWO LARGE PRIME NUMBERS
p = 444291551481268318602213381974607823284115560860624813859372204701739780689985896452357 1
q = 441853995693207821268512323438203160744359268595513039119145191269933271182489974451014 9

*****PUBLIC KEY (n, e) :

COMPUTE PRODUCT, <n>
n = p*q
196311997274732952651657888894714968937290467575095098099737720976861692317869965488365071418000411918
761207755317325718183594493264262799466278352174072980947359507585592220 79

COMPUTE TOTIENT(n)
phi(n) = (p-1)*(q-1)
196311997274732952651657888894714968937290467575095098099737720976861692317869965488364982803445694471
147220682746784437085191645781317185680980500434475813642172259998501883 60

GENERATE INTEGER, <e> | 1 < e < TOTIENT(n)
128545851386570345183648370524173684409528501571506801365356369058425507908005089959680274356868641589
905191426174535139660695538614761023514114693413784183938617131029098054 3

*****PRIVATE KEY (d)  :

COMPUTE DECRYPTION KEY, <d>

d = e^(-1) mod phi(n)
898424741650130983516254222713106604543787233126285666725059982210073362315738987756922208468384032992
068430565208621074466827914789017103598263512045074682271580744644960448 7
```

Les entiers premiers **p** et **q**

La fonction d'Euler **φ(n) = (p-1)(q-1)**

Le 2ème élément de la clé publique **e** tel que **PGCD(e,φ(n)) = 1** (mod **φ(n)**)

La clé privée **d** = e^{-1} (mod **φ(n)**)

2) **CHIFFREMENT RSA du MESSAGE :**

Module **n** = **p** x **q**

Exposant de Chiffrement **e** tel que :
e et φ(n) sont premiers entre eux

1. Modulus (n)

```
196311997274732952651657888894714968937290467575095098099737720976861692317869965488365071418000411918761207755317325718183594493264262799466278352174072980947359507585592220 79
```

2. Encryption key (e)

```
128545851386570345183648370524173684409528501571506801365356369058425507908005089959680274356868641589905191426174535139660695538614761023514114693413784183938617131029098054 3
```

Encodage Numérique du Message

3. Plaintext

```
Ceci est un essai de chiffrement RSA
```

Encode

4. Numerical message

```
851975628914155039603678257834129961950180307585319309722106646777238 45
```

Message en clair

Message crypté

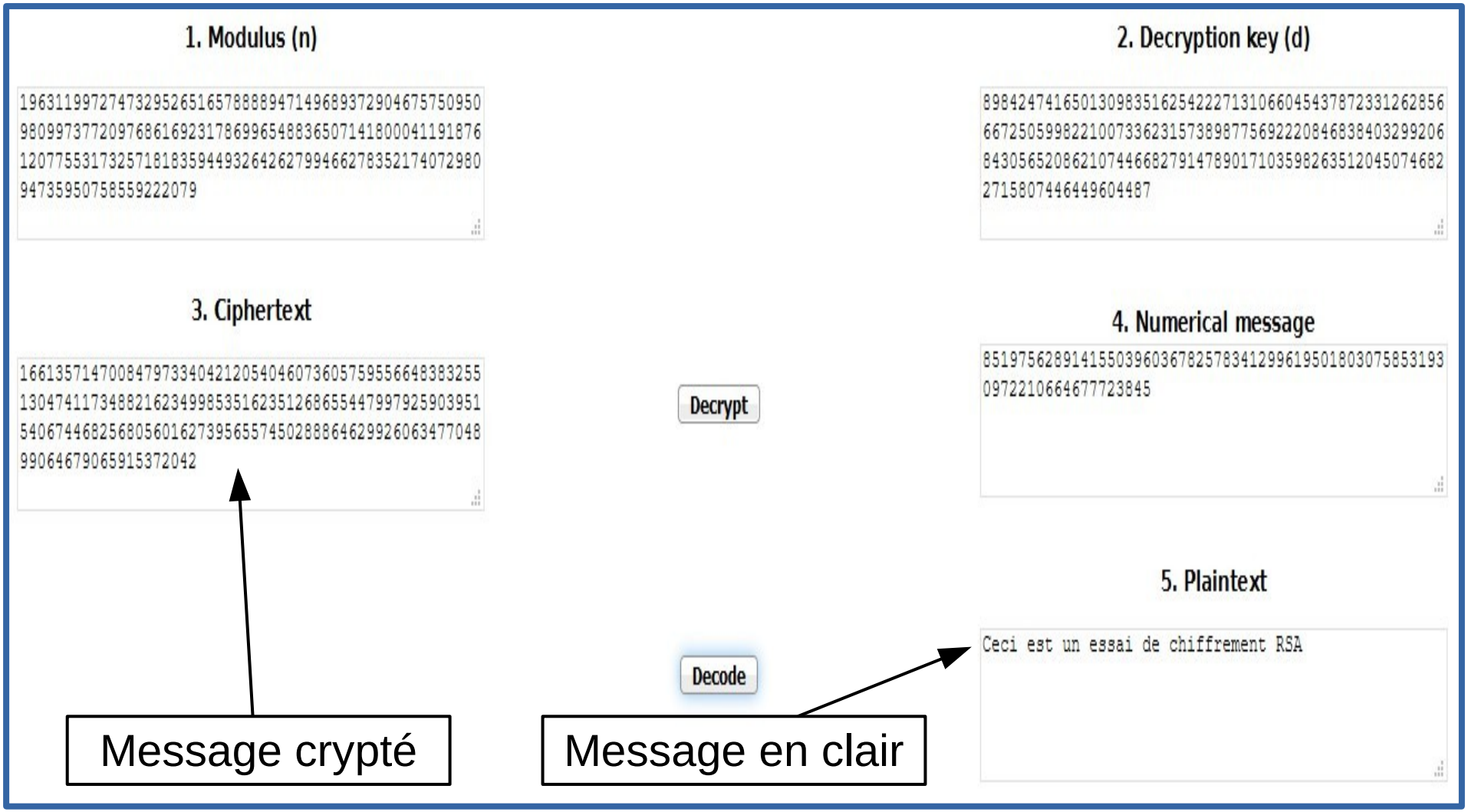
Encrypt

5. Ciphertext

```
166135714700847973340421205404607360575955664838325513047411734882162349985351623512686554479979259039515406744682568056016273956557450288864629926063477048990646790659153720 42
```

3) **DÉCHIFFREMENT RSA du MESSAGE :**

On <u>déchiffre</u> le message en utilisant la même clé publique (**n**,**e**), ce qui implique de connaître la clé privée **d** = e^{-1} (mod **φ(n)**) !



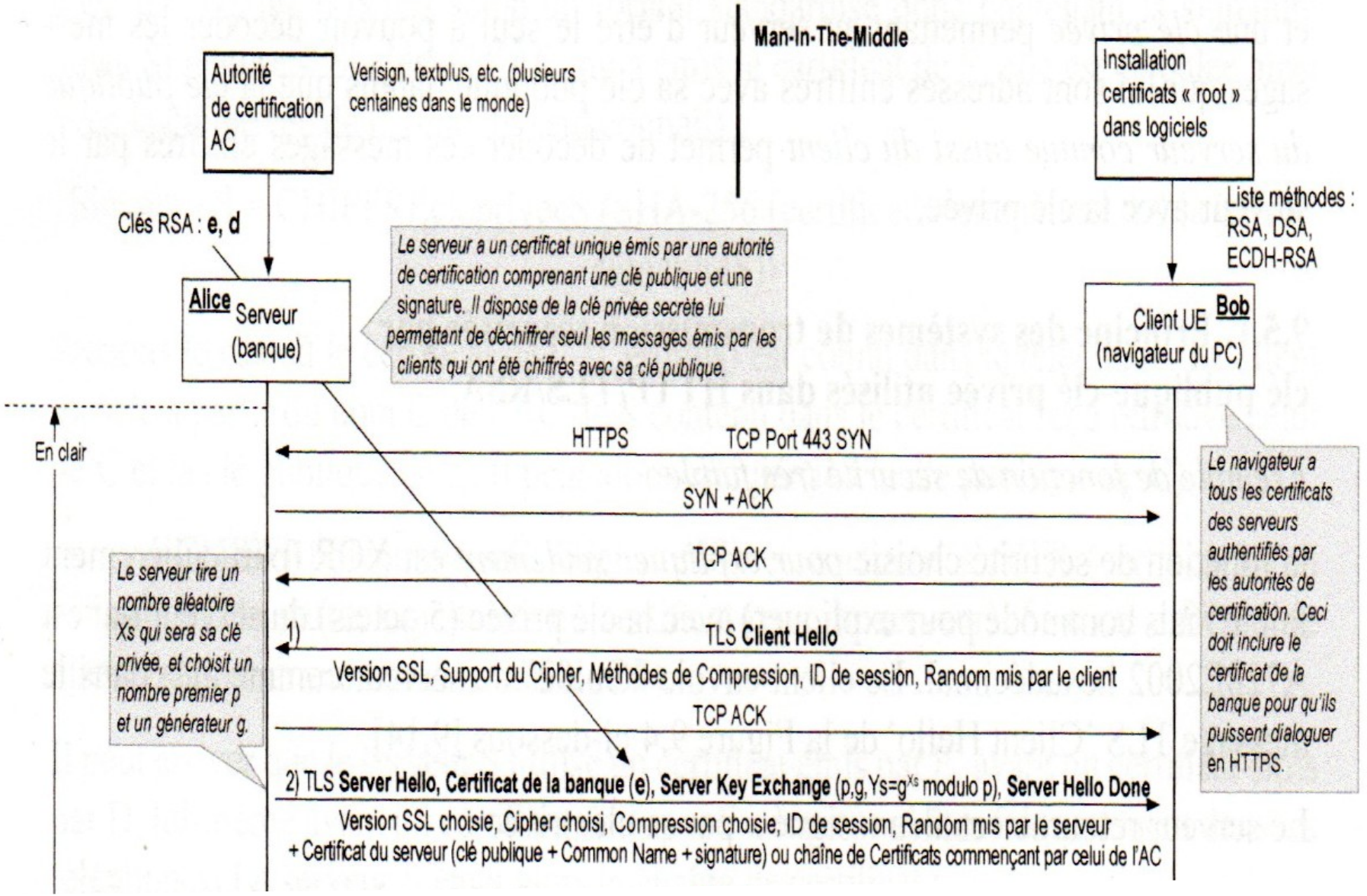
4-3 DÉROULEMENT *PRATIQUE* du CHIFFREMENT RSA

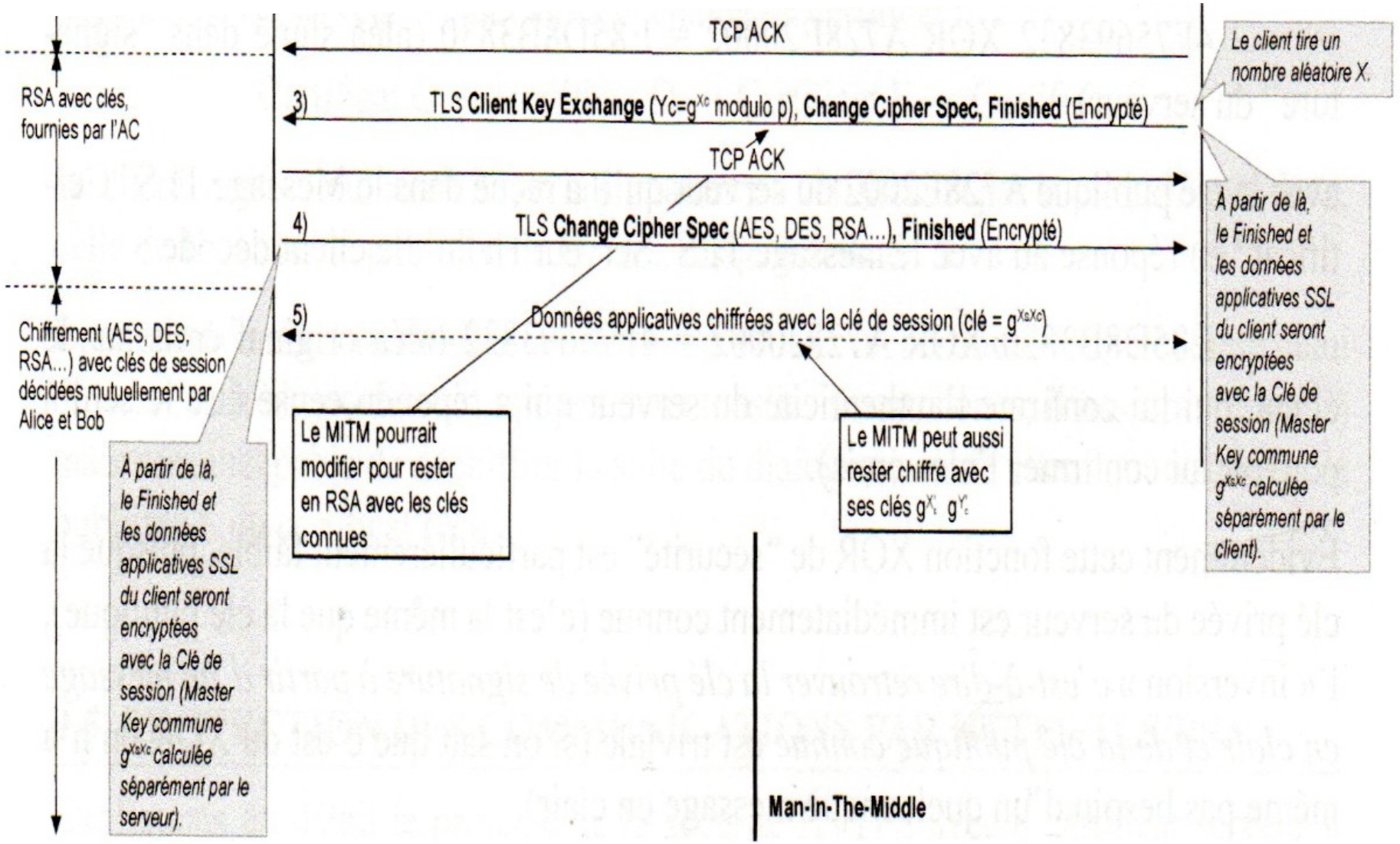
- Ce que nous venons de voir, c'est le fonctionnement *théorique* du système **RSA** de base …

- *Dans la pratique*, la mise en œuvre du système **RSA** n'est qu'un élément des protocoles de sécurisation des échanges par réseau informatique, notamment par Internet.

  Le protocole utilisé actuellement est **TLS** (Transport Layer Security) … dans lequel des clés cryptographiques différentes, publiques ou privées, sont utilisées lors d'une même session ...

Schéma d'une transaction sécurisée par **HTTPS** (Hypertext Transfer Protocol Secure), protocole de transfert sécurisé comprenant une couche de chiffrement comme TLS ...





(D'après Henry-Labordère A., ''Cryptologie classique ...'')

C'est lors de ce transfert qu'une *interception des données* peut avoir lieu, traditionnellement appelée **MITM** (Man In The Middle) …

5 MENACES sur le RSA … QUELLE SERA la RELÈVE ?

5-1 FAIBLESSES et DYSFONCTIONNEMENT du RSA ...

- La sécurité du **RSA** est liée notamment à la difficulté du **problème de la factorisation** de grands nombres (Problème **FACT**).

- Son *point faible* est qu'il implique de lourds calculs et se révèle donc bien plus lent que les systèmes de chiffrement à clé privée.

  Ces derniers (**AES**, **SHA** et autres) sont donc toujours employés.

  De plus, le **RSA** est le système *le plus attaqué*.

- Le **RSA** n'est donc pas utilisé pour chiffrer de grosses masses de données, surtout si le chiffrement doit être fait en temps réel.

- Le **RSA** sert donc le plus souvent à chiffrer des messages assez courts, en particulier les **clés** des systèmes à clé privée !

- En **2010**, on prévoyait de pouvoir factoriser un entier de 617 chiffres … en 2041 ! Ce qui signifie que ...
... la résistance de **RSA** aux attaques dépend de la taille de la clé, donc de celle des entiers premiers choisis !

- La taille des entiers premiers utilisés par **RSA** a varié. Aujourd'hui, le **RSA-2048** (bits) permet d'engendrer des clés de plus de *600* chiffres … qui ressemblent à ceci :

99999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999
9999999999999 ...

► Les succès des casseurs du système **RSA** ne sont souvent pas dus au système lui-même :

« … la plupart des systèmes de cryptographie qu'on a réussi à *casser* l'ont été, non par l'attaque réussie du noyau mathématique du système, mais à cause de faiblesses dans la mise en œuvre :

- mauvaise génération des clefs,
- mots de passe mal choisis ou mal protégés,
- contournement possible de l'algorithme mathématique,

etc. »

5-2 La RELÈVE du RSA ...

Question pour les *accros* et *mordus* de cryptographie :
Quels sont les *autres* systèmes de **chiffrement récents** ou **à venir** ?

Différents systèmes peuvent *déjà* ou devraient *bientôt* assurer la relève partielle de la cryptologie actuelle, RSA compris :

1 - la cryptographie fondée sur les **courbes elliptiques**,

2 - la cryptographie par **obfuscation de programmes**,

3 - la cryptographie **quantique**,

4 - la cryptographie dite **post-quantique** ...

1 - La Cryptographie fondée sur les COURBES ELLIPTIQUES

La cryptographie à base de **courbes elliptiques** a été introduite, voici plus de 30 ans, comme une application de la théorie des nombres.

Elle constitue une extension et une amélioration de la cryptographie à clé publique en autorisant des clés plus courtes.

C'est un chapitre assez difficile de la cryptographie qui demande une préparation mathématique ...

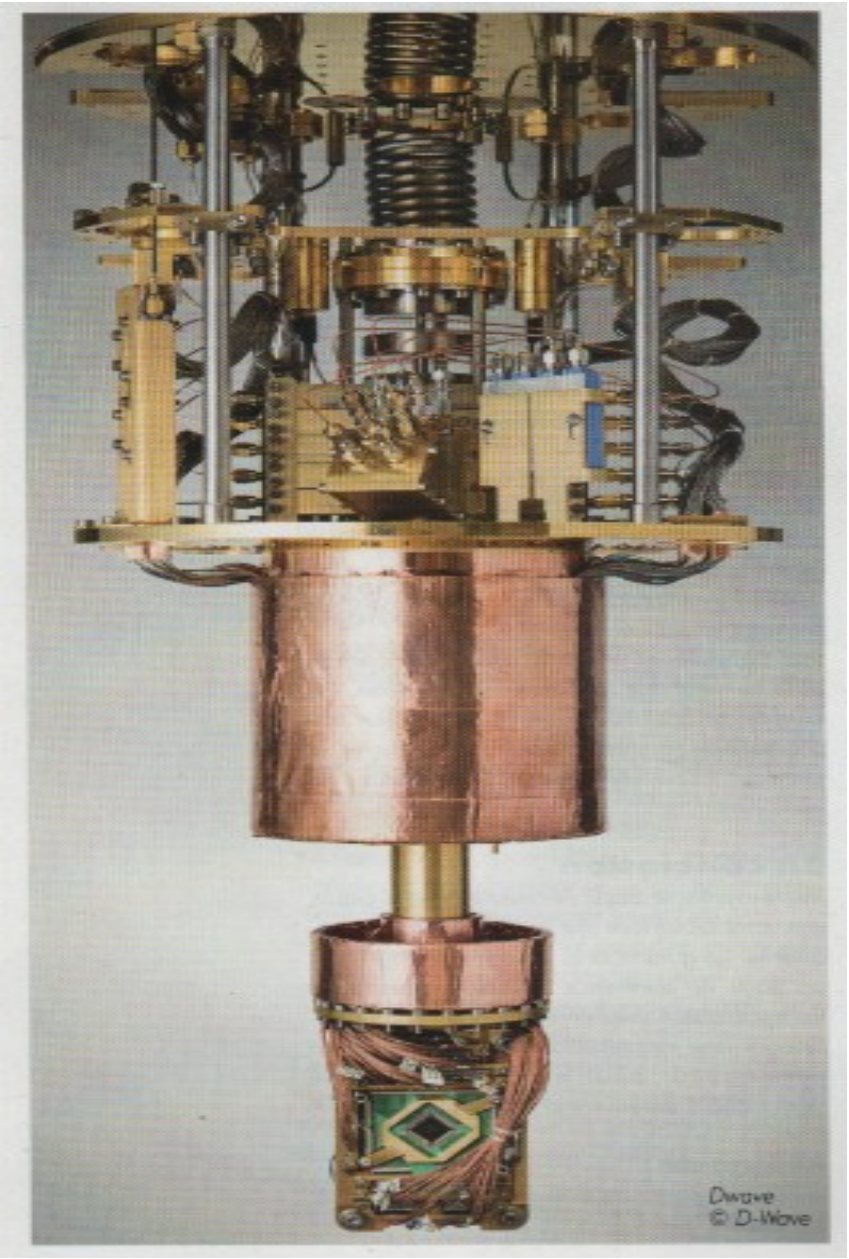
2 - Cryptographie et OBFUSCATION de PROGRAMMES

Une nouvelle technique de chiffrement est apparue dans les années 2000 :

la cryptographie par **obfuscation de programmes**.

Elle consiste à rendre (partiellement) illisible le code d'un algorithme de chiffrement sans faire obstacle à son fonctionnement.

3 - Cryptographie QUANTIQUE



L'Algorithmique quantique,

et, en particulier,

la Cryptographie quantique,

ont déjà plusieurs décennies

d'existence !

En revanche,

L'Ordinateur quantique n'en est

qu'à ses débuts …

La **cryptographie quantique** (ordinateur et algorithmique quantiques) est le nouvel enjeu de la recherche dans le domaine, mais ne remet pas en cause *pour le moment* les acquis de la cryptographie classique.

Le scénario dit de la « **cryptocalypse** » est celui où l'ordinateur quantique résoudra

FACT et **DLOG** en un temps raisonnable !

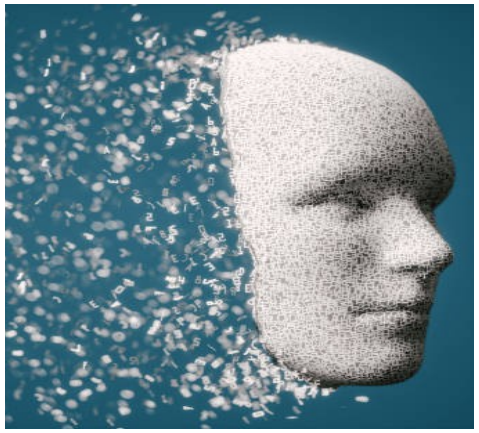
4 - La Cryptographie POST-QUANTIQUE

La **cryptographie post-quantique** est considérée comme une alternative à la cryptographie quantique. C'est pourquoi elle fait l'objet de nombreuses recherches.

L'objectif de l'**algorithmique post-quantique** est justement de *contrer* la puissance de l'ordinateur quantique.

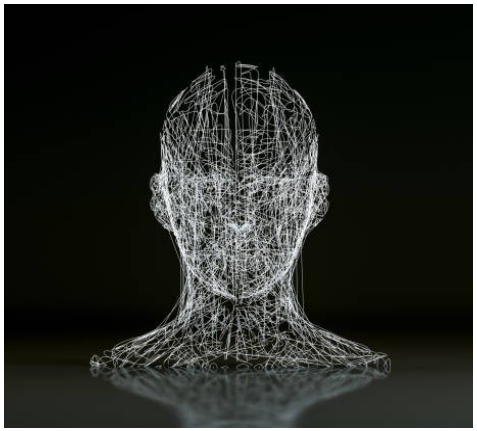
Parmi les « techniques » cryptographiques post-quantiques :

► La Cryptographie Multivariée



La **cryptographie multivariée** repose sur la difficulté du *problème PoSSo (*Polynomial Systems Solving) : trouver, s'il existe, un zéro commun d'un ensemble de polynômes non-linéaires.

► La Cryptographie sur Codes Correcteurs



La **cryptographie sur codes correcteurs**, dont le 1er exemple d'algorithme est celui de McEliece (1978) repose sur la *difficulté de décoder un code linéaire* si les équations comportent des erreurs.

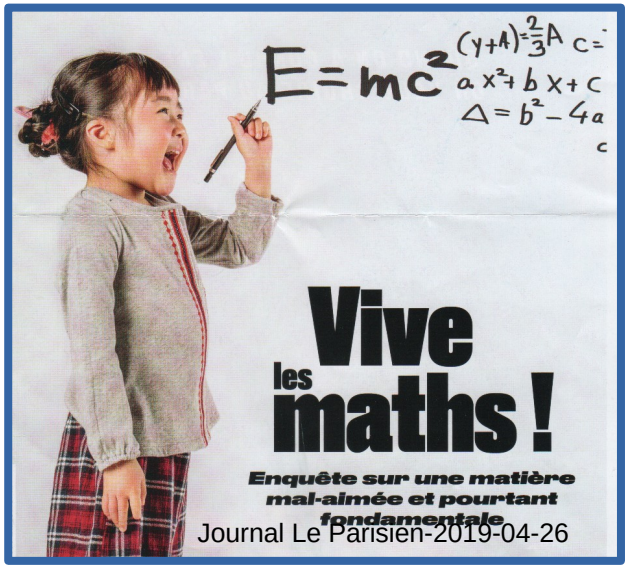
► La Cryptographie sur Réseaux Euclidiens

La **cryptographie sur réseaux euclidiens** s'appuie sur la notion de *matrice aléatoire* dont les coefficients sont des entiers modulo un nombre premier.

► La Cryptographie sur Arbre de Hachage

La sécurité de la **cryptographie sur arbre de hachage** repose sur une **fonction de hachage cryptographique**, très difficilement inversible.

CONCLUSION



Journal Le Parisien-2019-04-26

VOILA ! C'est FINI, …
pour aujourd'hui !

Si la CRYPTOGRAPHIE
est un domaine qui vous
« parle »,
il vous reste à explorer :

- les nombreuses variantes du système RSA classique,
- les courbes elliptiques … et la variante elliptique du RSA,
- les signatures et certificats (SSL, TLS, etc.),
- les algorithmes quantiques et post-quantiques,
- la cryptanalyse (dont les attaques sur le RSA),

… et bien d'autres choses encore ...

… et bien sûr à vous exercer chez vous !