Cryptographie

Saiida LAZAAR

Département Mathématiques & Informatique
ENSA de Tanger
Université AbdelMalek Essaadi
s.lazaar2013@gmail.com

Avant propos

Notes de cours niveau Master et Cycle Ingénieur; certains approfondissements sont reservés aux étudiants du master CyberSécurité et cyberCriminalité (C.S.C)

Prérequis

Arithmétique

Introduction

C'est quoi la cryptologie?

La cryptologie est la science du secret en grec.
Elle s'est transformée au 21ème siècle en une science dynamique à l'intersection des mathématiques, de l'informatique et de la micro-électronique.
Elle assure confidentialité, authenticité & intégrité.
Elle englobe la **cryptographie** et la **cryptanalyse**.

La cryptologie est un art ancien et une science nouvelle: Thème de recherche scientifique depuis les années 1970.

Introduction

- Cryptographie traditionnelle: Algorithmes simples, clés longues

- Cryptographie moderne: Algorithmes complexes, clés courtes

Cette discipline est liée à:

Arithmétique, Théorie des nombres, Algèbre, Complexité, Théorie de l'information, Codes correcteurs d'erreurs.

Cryptographie et modèle OSI

Normalisation ISO et entités de communication :

Trois entités interviennent dans les échanges de données:

Emetteur, Récepteur et Réseau de transport.

Chacune de ces entités doit offrir des garanties aux deux autres.

Chiffrement voie par voie dans le réseau

Il peut être réalisé au niveau de chacune des trois premières couches de l'OSI:
- Au niveau Physique: Mise en place de boîtes noires sur les lignes, qui laissent les données en clair au niveau des hôtes et nœuds de réseau.

- Au niveau Liaison: toutes les trames échangées sont chiffrées. Dans ce cas, un texte complet, en-têtes et informations d'acheminement, peut être chiffré sur la ligne.

- Au niveau Réseau, on peut chiffrer les informations échangées sur un circuit virtuel, indépendamment des autres.

Chiffrement de bout en bout

- Il laisse en clair les informations de routage.
- Seules les données constituant l'information transmise sont chiffrées.
- Dans un réseau multi nœuds, le message traverse plusieurs nœuds et garde le même chiffrement depuis son émetteur jusqu'à son destinataire final.
- Le chiffrement de bout en bout appartient à la couche Présentation.
- Les données restent brouillées dans les nœuds, et la distribution des clés est plus aisée.

Intégrité

- L'intégrité d'une unité de données ou d'un champ spécifique de données se fait par les codes de contrôle cryptographique.
- Le mécanisme est identique à celui des signatures numériques.
- L'intégrité d'un flot de données peut être assurée par le même mécanisme de cryptographie + des codes de détection d'erreurs.

Cryptographie ancienne

La technique des Hébreux

À partir du V siècle av. J-C., l'une des premières techniques de chiffrement est utilisée dans les textes religieux par les Hébreux.

La plus connue appelée Atbash est une méthode de substitution alphabétique inversée. Son nom est formé par les initiales des premières et dernières lettres de l'alphabet hébreux aleph, tau, beth, shin.

Elle consiste à remplacer chaque lettre du texte en clair par une autre lettre de l'alphabet choisie de la manière suivante: A devient Z, B devient Y, etc.

Code de César

Le code de César est la méthode cryptographique par substitution mono-alphabétique, la plus ancienne (1er siècle av J.-C).

Cette méthode a été utilisée par l'armée romaine mais elle était beaucoup moins robuste que la technique Atbash.

La faible alphabétisation de la population la rendait suffisamment efficace.

Code de César

Jule César utilisait le système suivant pour communiquer:

Chaque lettre de l'alphabet était décalé de 3 unités:

a donnait d, b donnait e, etc., x donnait a.

Ajoutons des caractères de ponctuation: espace, virgule, point, ?, et :

On disposera d'un alphabet de 31 caractères (? devient b, etc.).

Congruence modulo 31:
On remplace a par 1, b par 2... Et z par 26. . . espace par 27, virgule par 28, point par 29, ? par 30 et : par 0.

Le système de César consiste à effectuer une addition de 3 mod 31: a est remplacé par 1, on ajoute 3 pour obtenir 4 et 4 correspond à d.

Exemple

Le texte que nous souhaitons coder étant le suivant: *decaler les lettres de l'alphabet*

Le texte codé est: ghfdohu ohv ohwwuhv gh o'doskdehw

Ce système est très peu sûr, puisqu'il n'y a que 26 lettres dans l'alphabet donc seulement 26 façons de chiffrer un message avec le code de César.

Le chiffre de Vignère

En 1586, Blaise de Vigenère présente une technique de chiffrement par substitution polyalphabétique. Ce chiffrement sera décrypté en 1854.

Le chiffrement utilise une clé littérale dont chaque lettre indique le décalage alphabétique à appliquer sur le texte en clair.

- On reporte les lettres sur une grille de $26x26$ cases,(voir schéma).

- Le chiffré est l'intersection de la ligne qui commence par la lettre à coder, avec la colonne qui commence par la première lettre du mot de passe.

- Dès que l'on atteint la fin du mot de passe, on recommence à la première lettre.

Pour décoder, il suffit de faire la même chose dans l'autre sens.

Points forts

Cet algorithme de cryptographie comporte beaucoup de points forts.

Il est très facile d'utilisation, et le déchiffrement est tout aussi facile si on connaît la clé.

Un autre avantage réside dans le fait que l'on peut produire une infinité de clés. Il a fallut attendre près de 4 siècles pour qu'il soit cryptanalysé au milieu du $XIX^{ème}$siècle.

Illustration

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

Méthode de pliage

Elle consiste à écrire le message dans une matrice comportant autant de colonnes que la clé de caractères.

Le cryptogramme est obtenu en écrivant en ligne le clair dans la matrice puis en lisant cette matrice en colonnes dans l'ordre alphabétique des lettres de la clé.

Décryptage: A partir d'une matrice à n colonnes où n est la longueur de la clé, on écrit chaque colonne dans le cryptogramme dans l'ordre alphabétique de la clé.

Méthode de pliage: Exercice

Exercice: Soit le texte en clair: ENSA de TANGER Et soit la Clé: GSRT. Trouver le cryptogramme.

Solution: La clé est de longueur 4. On découpe le cryptogramme en 4 blocs de n caractères, puis on associe à chaque bloc une lettre dans l'ordre alphabétique.

Il suffit d'inscrire chaque bloc verticalement dans un tableau puis de le lire hortizontalement.

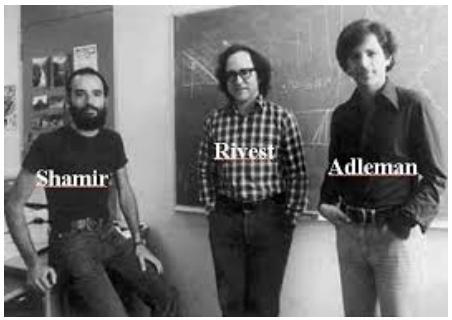
Cryptographie à clé publique ou cryptographie asymétrique

Elle repose sur ce principe: On dispose d'une fonction P clé publique, qui possède un inverse S.
On suppose qu'on peut fabriquer (P, S), mais que connaissant uniquement P, il est impossible de retrouver S.

Si Bob veut vous envoyer un message, il vous transmet $P(message)$.
Vous décodez le message en calculant:
$S(P(message)) = message$.

La connaissance de P par un tiers ne compromet pas la sécurité de l'envoi des messages codés, puisqu'elle ne permet pas de retrouver S.

RSA



Une solution possible, la cryptographie RSA.

Problème posé: Soit un nombre entier $n = pq$, il est très difficile de retrouver p et q premiers.

- La clé privée est définie à partir de p et q.

Un peu d'histoire

RSA a été décrit en 1977 par Ronald Rivest, Adi Shamir et Leonard Adelman.
RSA a été breveté par le Massachusetts Institute of Technology (MIT) en 1983 aux États-Unis.

- Ronald Linn Rivest (born May 6, 1947) is a cryptographer. He is member of MIT's Department of Electrical Engineering and Computer Science and member of MIT's Computer Science and Artificial Intelligence Laboratory.

Un peu d'histoire

RSA a été décrit en 1977 par Ronald Rivest, Adi Shamir et Leonard Adelman.

RSA a été breveté par le Massachusetts Institute of Technology (MIT) en 1983 aux États-Unis.

- Ronald Linn Rivest (born May 6, 1947) is a cryptographer. He is member of MIT's Department of Electrical Engineering and Computer Science and member of MIT's Computer Science and Artificial Intelligence Laboratory.
- Adi Shamir (born July 6, 1952) is an Israeli cryptographer. He is one of the inventors of differential cryptanalysis.

Un peu d'histoire

RSA a été décrit en 1977 par Ronald Rivest, Adi Shamir et Leonard Adelman.

RSA a été breveté par le Massachusetts Institute of Technology (MIT) en 1983 aux États-Unis.

- Ronald Linn Rivest (born May 6, 1947) is a cryptographer. He is member of MIT's Department of Electrical Engineering and Computer Science and member of MIT's Computer Science and Artificial Intelligence Laboratory.
- Adi Shamir (born July 6, 1952) is an Israeli cryptographer. He is one of the inventors of differential cryptanalysis.
- Leonard Adleman (born December 31, 1945) is an American computer scientist.

RSA

Les inconvénients de ce mode de chiffrement est la génération et le transport des clés.

Cette clé doit être parfaitement aléatoire.

A partir de cette clé et du texte à coder qu'on découpe en blocs, on effectue des transformations toujours réversibles, mais suffisamment compliquées pour que le texte obtenu donne l'impression d'être aléatoire.

Même si une clé de 128 bits peut paraître petite, elle offre une protection suffisante pour une attaque exhaustive.

La signature électronique et cryptographie asymétrique

Comment s'assurer de l'authenticité de l'envoi?

Alice veut envoyer un message crypté à Bob, mais Bob veut s'assurer que ce message provient bien d'Alice.

Alice posséde le couple clé publique/clé privée (P_A, S_A), et Bob le couple (P_B, S_B). Alice veut envoyer M.

Phase d'envoi : Alice calcule $S_A(M)$, puis $P_B(S_A(M))$.

Phase de réception : A l'aide de sa clé privée, Bob calcule $S_B(P_B(S_A(M))) = S_A(M)$.
Puis il calcule $P_A(S_A(M)) = M$.

Retour à l'Algorithme RSA

Description

RSA est le plus célébre algorithme asymétrique, en voici les grandes lignes:

- L'utilisateur choisit deux grands nombres premiers p et q et calcule $n = pq$.
- Il choisit ensuite un nombre $e < n$ et premier avec $(p-1)(q-1)$.
- Il publie la clé publique (e, n) et garde secrets p et q.
- Il calcule sa clé secrète en cherchant le nombre d:
  $\forall x, 0 \leq x \leq n, x^{ed} = x^{de} = x \mod n, ed = 1 \mod (p-1)(q-1)$.

Chiffrement et déchiffrement

Les fonctions de chiffrement et de déchiffrement sont définies par:

$E(x) = x^e \mod n$ et $D(x) = x^d \mod n$.

- L'envoi d'un message m se fait en calculant : $m^e \mod n$.

- La lecture du message se fait en calculant :

$$(m^e \mod n)^d \mod n$$

- La signature d'un message se fait en envoyant $m^d \mod n$.

Le RSA est il sûr?

- La sécurité du RSA repose sur la difficulté de factoriser de grands entiers.

- Le record établi en 1999, avec l'algorithme le plus performant et des moyens matériels considérables, est la factorisation d'un entier à 155 chiffres.

- Il faut, pour garantir une certaine sécurité, choisir des clés plus grandes.

- Les experts recommandent des clés de 768 bits pour un usage privé, et des clés de 1024, voire 2048 bits, pour un usage sensible.

Le RSA est il sûr?

- Si l'on admet que la puissance des ordinateurs double tous les 18 mois (loi de Moore), une clé de 2048 bits devrait tenir jusque 2079.
- Un modèle d'ordinateur quantique a été réalisé, il permettrait de factoriser très rapidement des entiers.
- Les ordinateurs quantiques n'en sont encore qu'à leurs prémices, et leur record (automne 2001) est la factorisation de 15=35!

Illustration

1094173864157052742180970732204035761200373294544920599091384213147634998428893478471799725789126733249762575289978183379707653724402714674353159335433389 7=

1026395928297411057720541965739916759007165678080380668033419335217907113077 79 × 1066034883801684548209272203600128786792079585759892915222706082371930628086 43

Factorisation d'un entier à 155 chiffres.

Fabrication de grands nombres premiers

L'algorithme RSA nécessite la fabrication de très grands nombres premiers (500 chiffres par ex.).

Les nombres de Mersenne sont $M_p = 2p - 1$, où p est premier.

Ils ne sont pas tous premiers, on dispose du test de Lucas-Lehmer pour tester s'ils le sont:

On construit une suite S_n en posant $S_1 = 4$, et $S_n = (S_{n-1})^2 - 2$.

Pour $p > 2$, on peut prouver que M_p est premier si et seulement si M_p divise S_p.

Test de primalité

Pour le RSA, il n'y a rien de mieux qu'un premier aléatoire, sur lequel on n'a à priori aucune information.

L'idée est de prendre un entier de 500 chiffres au hasard, et de tester s'il est premier :

Si c'est le cas, on le garde, sinon on choisit un autre au hasard, jusqu'à finir par tomber sur un premier.
Il faut que l'on puisse trouver un premier au bout d'un nombre raisonnable de tirages.

Test de primalité

Si l'on choisit un nombre de 500 chiffres au hasard, on a environ un chance sur $ln(10^{500})$ de tomber sur un premier.

i.e environ une chance sur 1150. Ce qui est raisonnable!

D'autre part, étant donné un entier n, il faut qu'on puisse rapidement déterminer s'il est premier ou non.

Nécessité de tests de primalité efficaces.

Test de primalité

Le plus rudimentaire est de prendre tous les nombres entre 2 et $\sqrt{n}$, et de vérifier s'ils divisent ou non n.

Mais c'est trop naïf, car il nécessite 10^{250} calculs environ pour un nombre de 500 chiffres. Impossible!

On a recours à d'autres tests, les tests de Solovay-Strassen et Miller-Rabin. Leur particularité est d'être des algorithmes probabilistes.

En général, en cryptographie, on se contente de nombres dont on sait qu'ils sont premiers avec une probabilité supérieure à $1 - \frac{1}{2^{100}}$.

Exemples

Un utilisateur A a rendu sa clé (e, n) publique $(11, 11023)$.
Il garde sa clé secrète : d $= 5891$, sachant que $n = 73x151$.

Un utilisateur B veut envoyer un message x à A et souhaite protéger son message. Soit x $= 2814$.
Le chiffrement du message x se fait avec la clé publique de A, soit:
$y = E(x) = 2814^{11} \mod 11023 = 1473$.

L'utilisateur A (et lui seul) peut lire le message y en appliquant sa clé secrète: $x = D(y) = 1473^{5891} \mod 11023 = 2814$.

Echange des clés de Diffie-Helmann

- Alice et Bob choisissent publiquement un nombre premier p, et un entier $1 < a < p$ qui est racine première de p.
- Alice choisit secrètement x_1, et Bob choisit secrètement x_2.
- Alice envoie à Bob a^{x_1}, et Bob calcule $K = a^{x_1 x_2}[p]$.
- Bob envoie à Alice a^{x_2}, et Alice calcule $K = a^{x_2 x_1} = a^{x_1 x_2}[p]$.

Alice et Bob sont en possession d'une même clé secrète K, non échangée directement.
Si quelqu'un a piraté la conversation, il aura p, a, a^{x_1} et a^{x_2}. Il aura à résoudre l'équation du logarithme discret $y = a^x[p]$.

Définition du logarithme discret

- On définit une racine première d'un nombre premier p comme celle dont les puissances produisent tous les entiers de 1 à $p-1$:

Si a est une racine première du nombre premier p alors les nombres: $a \mod p, a^2 \mod p, ..., a^{p-1} \mod p$ sont distincts et consistent en les entiers de 1 à $p-1$.

Definition

$\forall b \in \mathbb{N}$ et pour toute racine première d'un nombre premier p, on peut trouver un exposant unique i tel que:

$$b = a^i \mod p, 0 \leq i \leq p-1$$

L'exposant i est appelé logarithme discret.

Un peu d'arithmétique modulaire:

$$K = (a^{x_2} \mod p)^{x_1} \mod p = (a^{x_2})^{x_1} \mod p$$
$$= a^{x_2 x_1} \mod p$$
$$= (a^{x_1} \mod p)^{x_2} \mod p$$
$$= a^{x_1 x_2} \mod p$$

Exemple

Supposons que l'échange des clés est basé sur l'utilisation d'un nombre premier $q = 71$ et une racine première de 71.
Dans ce cas, $a = 7$.
Bob et Alice choisissent des clés privées $x_1 = 5$ et $x_2 = 12$ respectivement. Chacun calcule sa clé publique:

$$K_{alice} = 7^5 \mod 71 = 51, K_{bob} = 7^{12} \mod 71 = 4$$

Après avoir échangé les clés publiques, chacun de son côté calcule la clé secrète commune:

$K = 4^5 \mod 71 = 30$ et $K = 51^{12} \mod 71 = 30$

Problème du logarithme discret

Soient p un nombre premier, g un élément du groupe multiplicatif $\mathbb{Z}/p\mathbb{Z}$, et n l'ordre de g.
Comme le groupe multiplicatif est d'ordre $p-1$, on aura: $n|p-1$.
On choisira g de telle manière que n soit premier.
$\forall l \in \mathbb{N}$, l'élément $h = g^l \mod p$ peut être calculé rapidement.
En effet, au lieu d'effectuer l multiplications, on peut remarquer que:

Si l est pair, $g^l = (g^{l/2})^2$. Sinon, $g^l = g(g^{(l-1)/2})^2$.

On est donc ramené au même problème, mais pour une valeur de l deux fois plus petite.

Une implémentation récursive résoud le problème initial en $O(\log l)$ opérations.

Problème du logarithme discret

Réciproquement, on suppose que l'on se donne un élément h dans le sous-groupe engendré par g.
Le problème du logarithme discret est le suivant:

Retrouver un entier l entre 0 et $n-1$ tel que: $h = g^l \mod p$.
Il n'y a pas d'algorithme pour résoudre ce problème en temps polynomial en $\log n$.
La méthode naïve qui consiste à essayer successivement toutes les valeurs de l conduit à une complexité de $O(n)$ multiplications.

(Il existe des algorithmes qui permettent d'abaisser la complexité à $O((n)^1/2)$ opérations.)

Exercices

1. Dans un système RSA, la clé publique d'un utilisateur est $e = 31, n = 3599$. Quelle est la clé privée de cet utilisateur?

2. Considérer le dispositif suivant:

2.1. Choisir un nombre impair E.

2.2. Choisir deux nombres premiers P et Q où $(P-1)(Q-1)-1$ est divisible par E.

2.3. Multiplier P et Q pour obtenir N.

2.4. Calculer $D = \frac{(P-1)(Q-1)(E-1)+1}{E}$.

Ce dispositif est il équivalent à RSA? Justifier votre réponse.

Exercices

3. Dans un système à clé publique utilisant RSA, on intercepte le texte chiffré suivant: $C = 10$ envoyé à un utilisateur dont la clé publique est $e = 5, n = 35$. Quel est le texte en clair M?

Algorithme par blocs

Le Data Encryption Standard, mis au point dans les laboratoires d'IBM, a été adopté par le National Bureau of Standard américain en 1977.

Sa normalisation au sein de l'ISO est proposée sous le nom de Data Encipherment Algorithm.
C'est un algorithme très répandu dans le monde industriel et bancaire.

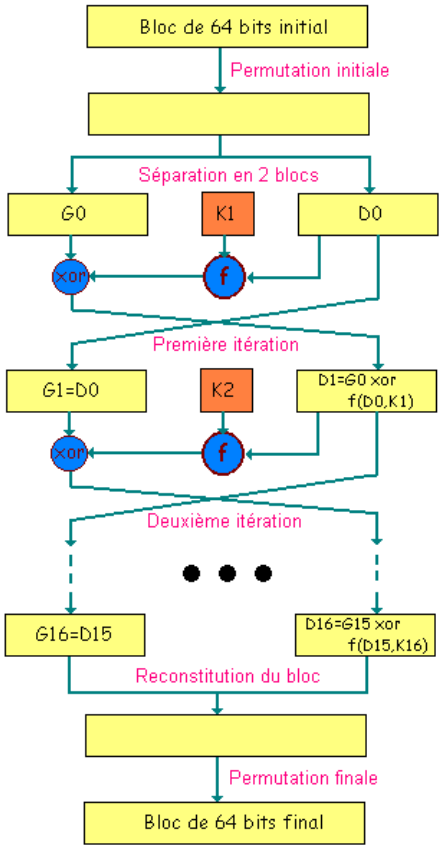
Principe: Il consiste à découper un message M en blocs de **64 bits** ; chacun d'eux sera crypté à l'émission et décrypté à la réception indépendamment des autres blocs grâce à une clé K de **56 bits**.

Algorithme

Les grandes lignes de l'algorithme sont les suivantes:

- Fractionnement du texte en blocs de 64 bits;
- Permutation initiale des blocs;
- Découpage des blocs en deux parties: gauche et droite, nommées G et D;
- Étapes de permutation et de substitution répétées 16 fois (appelées rondes);
- Recollement des parties gauche et droite puis permutation initiale inverse.

Organigramme de DES



Permutation initiale de DES

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

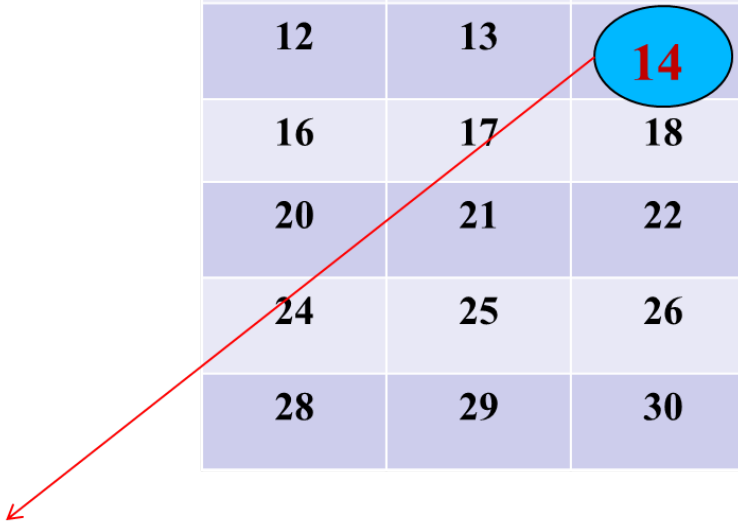
$58^{ème}$ bit du bloc de texte de 64 bits se retrouve en première position

Permutation finale de DES

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Expansion de 32 bits à 48 bits

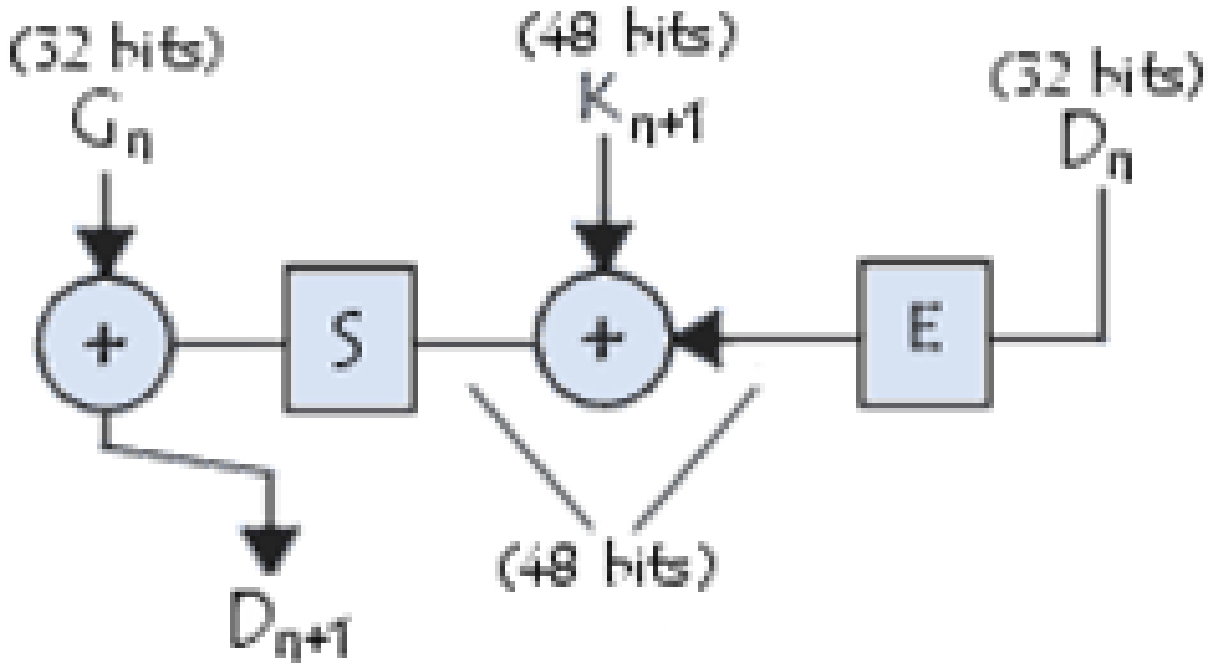
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1



Fonctions de sélection ou S-box

- Après l'expansion, on scinde en 8 blocs de 6 bits. Chacun de ces blocs passe par des fonctions de sélection, notées S_i.
- Les premiers et derniers bits de chaque blocs de 6 détermine (en binaire) la ligne de la fonction de sélection, les autres bits déterminent la colonne.
- Les 6 premiers bits passent par la boite S_1 pour former 4 bits en sorties, les 6 suivants passent par S_2.
- Les 32 bits en sorties de S subissent une nouvelle permutation; le résultat est additionné (mod 2) à G_{i-1}.

Les rondes ou transformations itératives



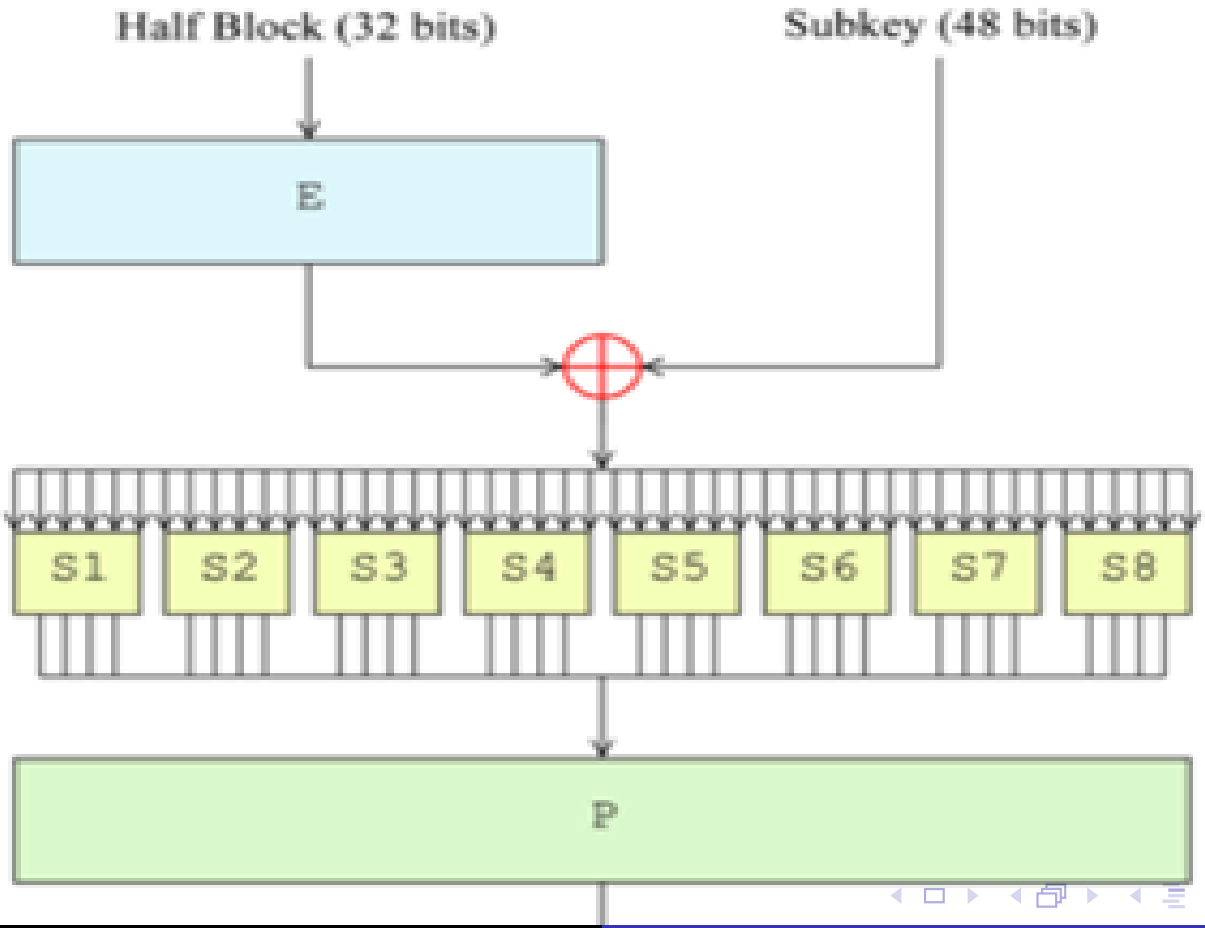
Fonctions de sélection

On applique, en parallèle, 8 boîtes de 6 bits vers 4 bits

Ceci réduit l'état interne de:
8 x 6 = 48 bits à 8 x 4 = 32 bits

Chaque boîte S est codée comme un tableau avec $2^6 = 64$ entrées

Fonctionnement de la machine S-Box



Description de la S-Box 1

Supposons qu'on a six bits à l'entrée 110101:

Le premier et le dernier bit soit 11 indique le numéro de la ligne (3) puisque 11 binaire=3 décimal.

Les quatre bits restant 1010 (lu comme 10 décimal)indique le numéro de la colonne.

A l'intersection (voir table de S1), on trouve 3 qui correspond à 0011: 4 bits en sortie

Table de la S-Box 1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Triple DES

Le tripleDES (3DES) est en fait l'algorithmeDES appliqué 3 fois sur les données.
Il a été conçu par Whitfield Diffie, Martin Hellman et Walt Tuchmann en 1978.
L'algorithme utilise une taille de clé comprise entre 128 bits et 192 bits.
La taille des blocs est de 8octets.
Le tripleDES a été approuvé FIPS (Federal Information Processing Standards) et peut être utilisé par les organisations gouvernementales.

A.E.S (Advanced Ecryption Standard)

AES a été défini en octobre 2000 suite au concours lancé en 1997 par le NIST et devint le nouveau standard de chiffrement pour les organisations du gouvernement des États-Unis. Il a été approuvé par la NSA (National Security Agency).

Les auteurs sont les belges Joan Daemen et Vincent Rijmen.

Rijndael a été conçu de manière à rendre des attaques linéaire ou différentielle très difficiles.

Description

L'algorithme prend en entrée un bloc de 128 bits qui sont mélangés selon une table donnée.

Ces octets sont placés dans une matrice de 4x4 éléments et les lignes subissent une rotation vers la droite.

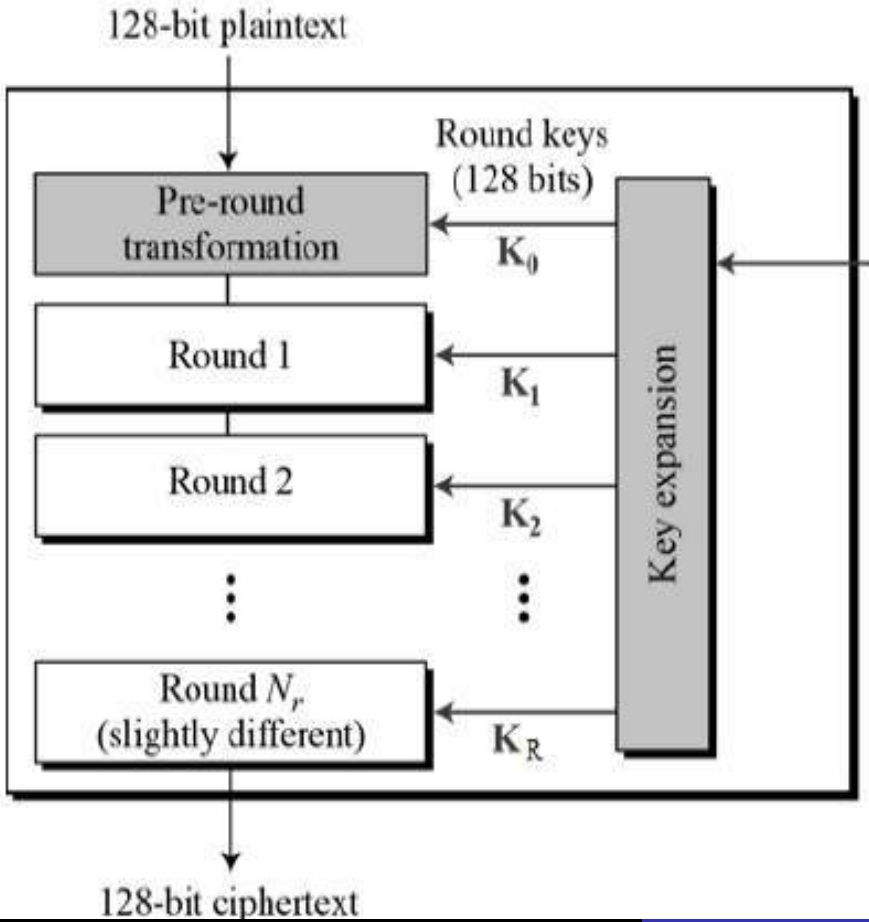
L'incrément pour la rotation varie selon le numéro de la ligne.

Une transformation est ensuite appliquée sur la matrice par un Xor avec une matrice clé.

Un Xor entre la matrice et une autre matrice permet d'otenir une matrice intermédiaire.

Cest opérations sont répétées plusieurs fois et définissent un tour.

Organigramme AES

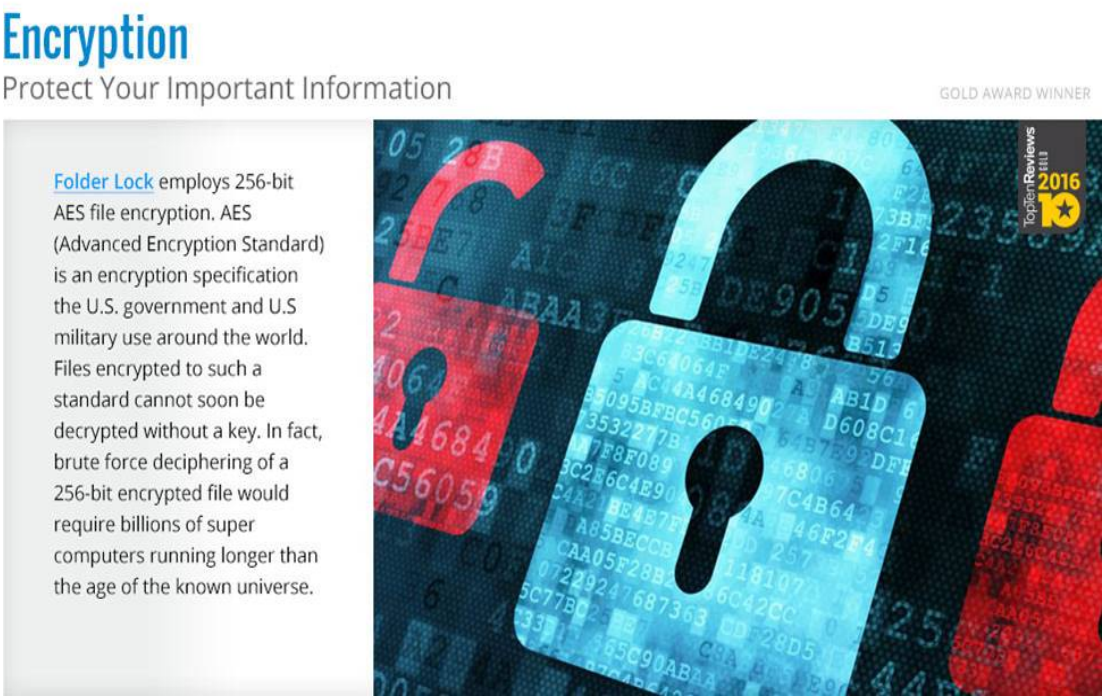


R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

Best 2016

http://encryption-software-review.toptenreviews.com/



Introduction

Les courbes elliptiques sont utilisées dans la cryptographie à clé asymétrique et notamment par l'algorithme ECDSA (Elliptic Curve Digital Signature Algorithm).

Comparaison avec RSA

RSA utilise les multiplications de nombres premiers tandis que ECDSA offre un meilleur niveau de sécurité à taille de clé équivalente.

Les opérations de signature sont beaucoup plus rapides dans le cas de ECDSA.

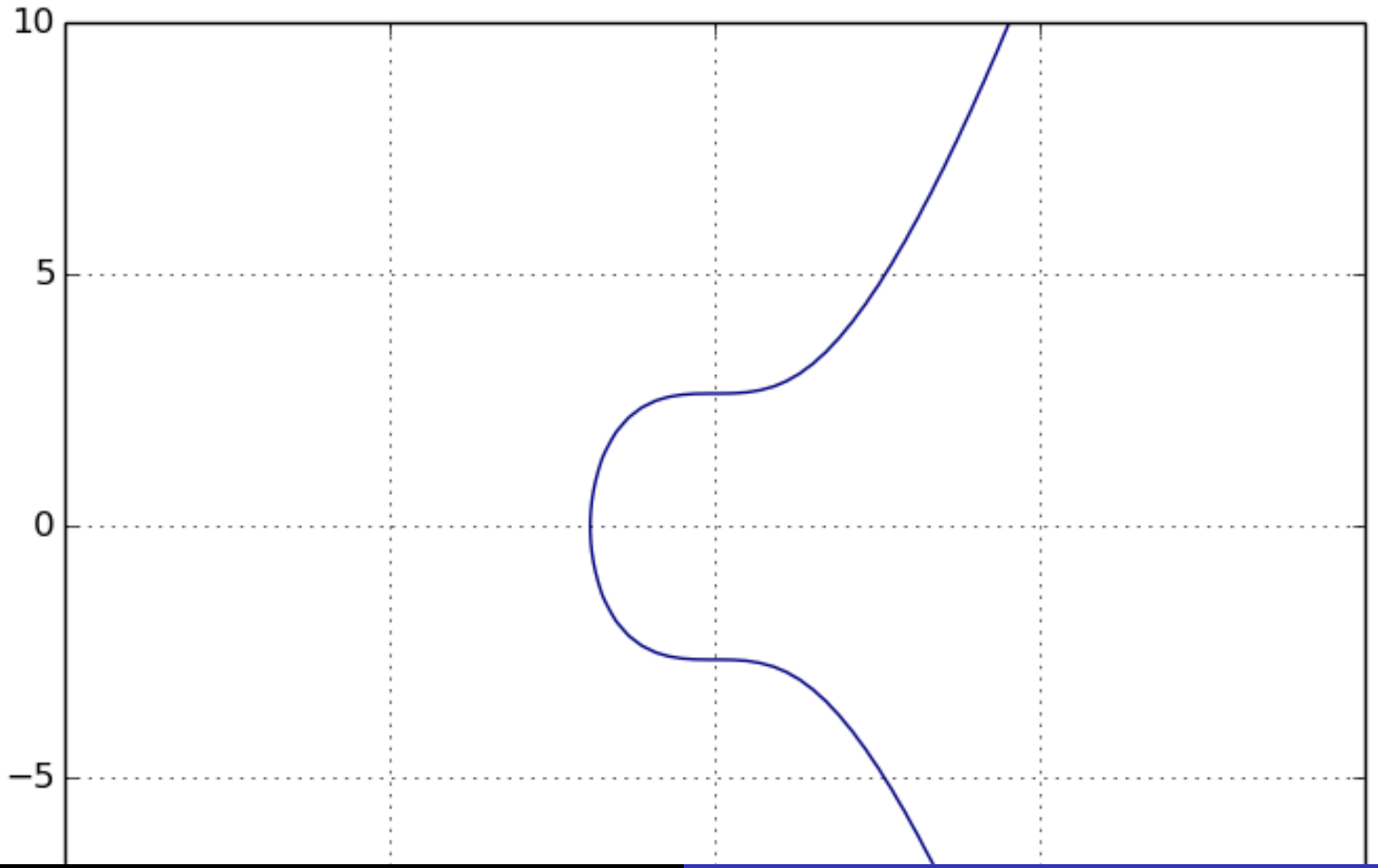
C'est quoi une Courbe elliptique ?

Une courbe elliptique n'est rien d'autre qu'une représentation d'une équation de la forme :

$$y^2 = x^3 + ax + b$$

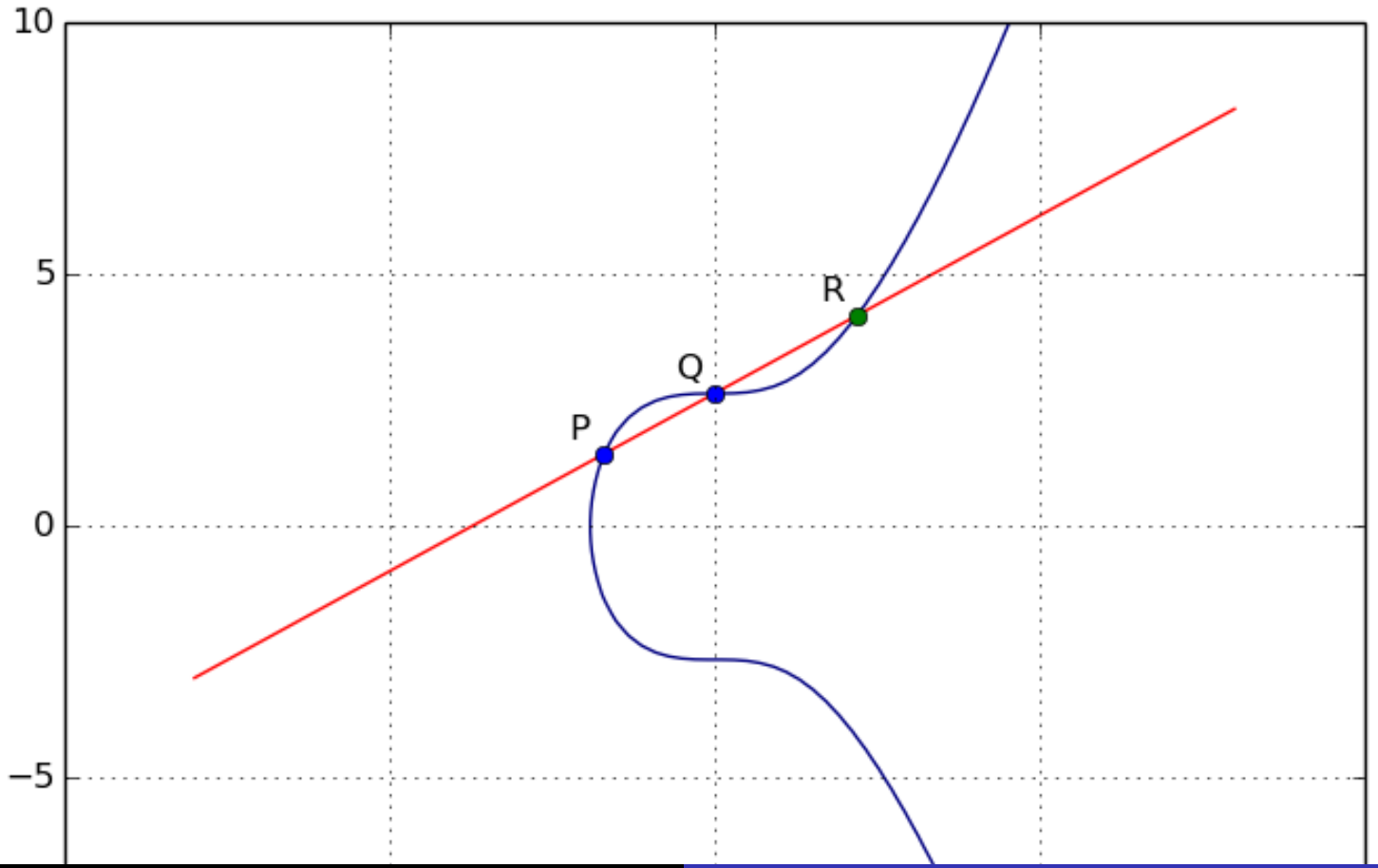
où a et b sont des nombres réels.

Exemple



Une propriété importante est qu'une droite qui coupe la courbe en deux points passe par un troisième point sauf exceptions.

Illustration

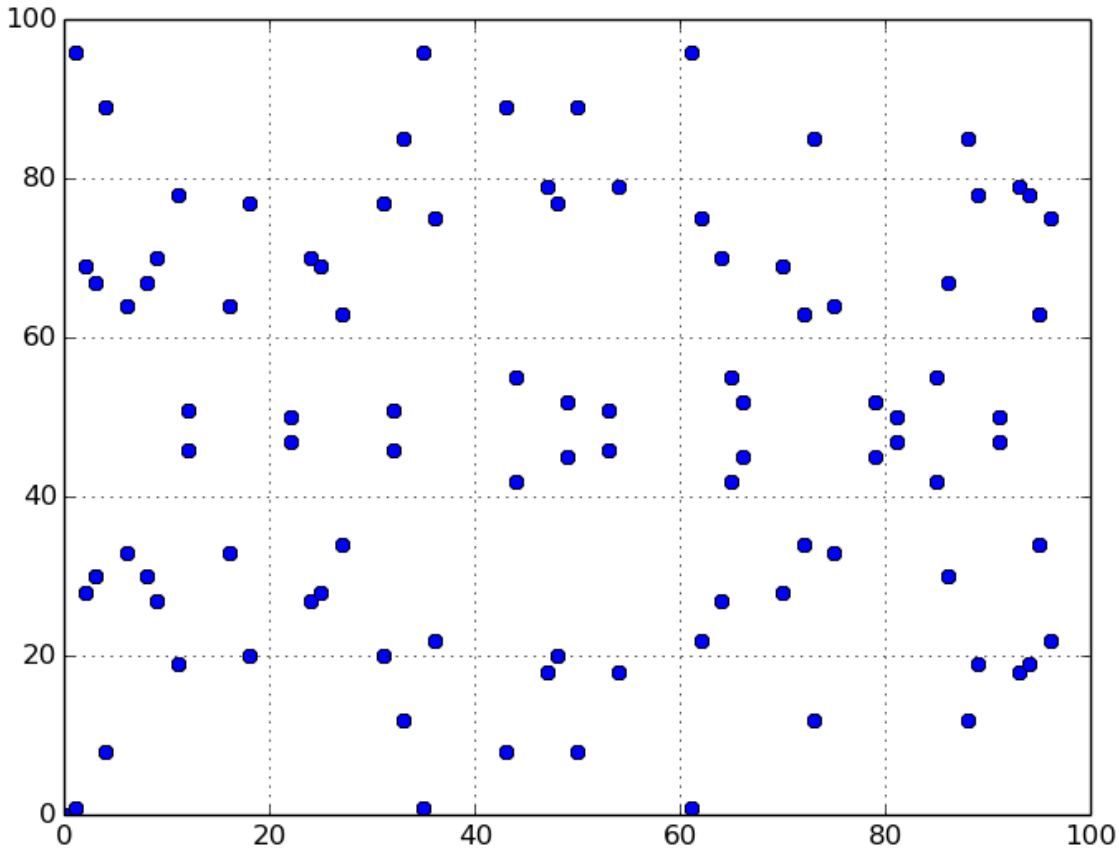


Courbes elliptiques et arithmétique

Pour pouvoir utiliser les courbes elliptiques en cryptographie, il faut travailler avec des nombres entiers naturels.

- On garde seulement les points dont l'abscisse et l'ordonnée sont des entiers.
- On utilise l'arithmétique modulaire pour limiter l'abscisse et l'ordonnée maximale de chaque point.
- A une courbe, on ajoute un nouveau paramètre P, et chaque point sera exprimé modulo P.

Courbe modulo 97



Point à l'infini

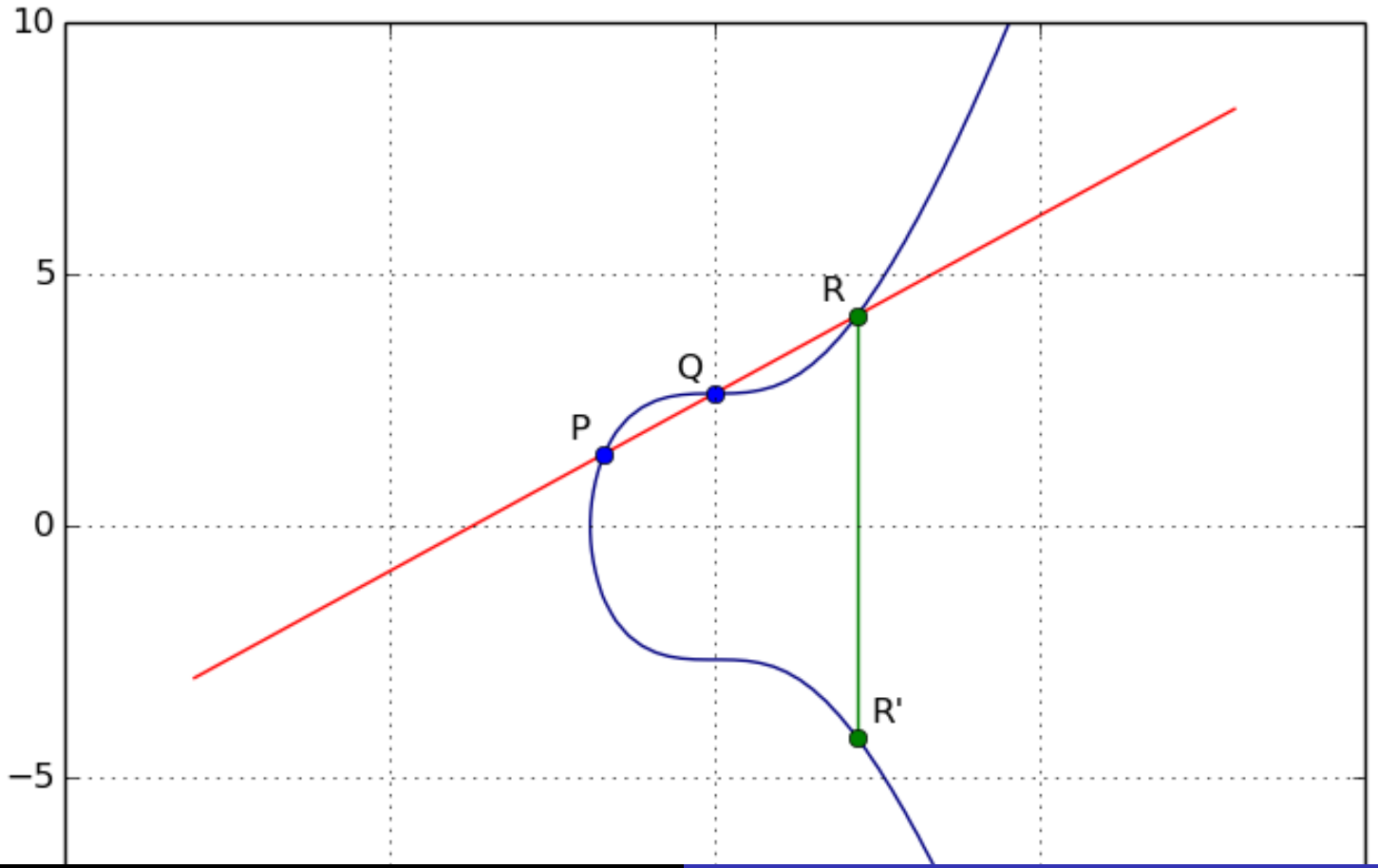
On prend tous les points de la courbe, pour chaque point, on trace une droite verticale : notre 0 se trouve au point de croisement de toutes ces droites.

Addition et multiplication

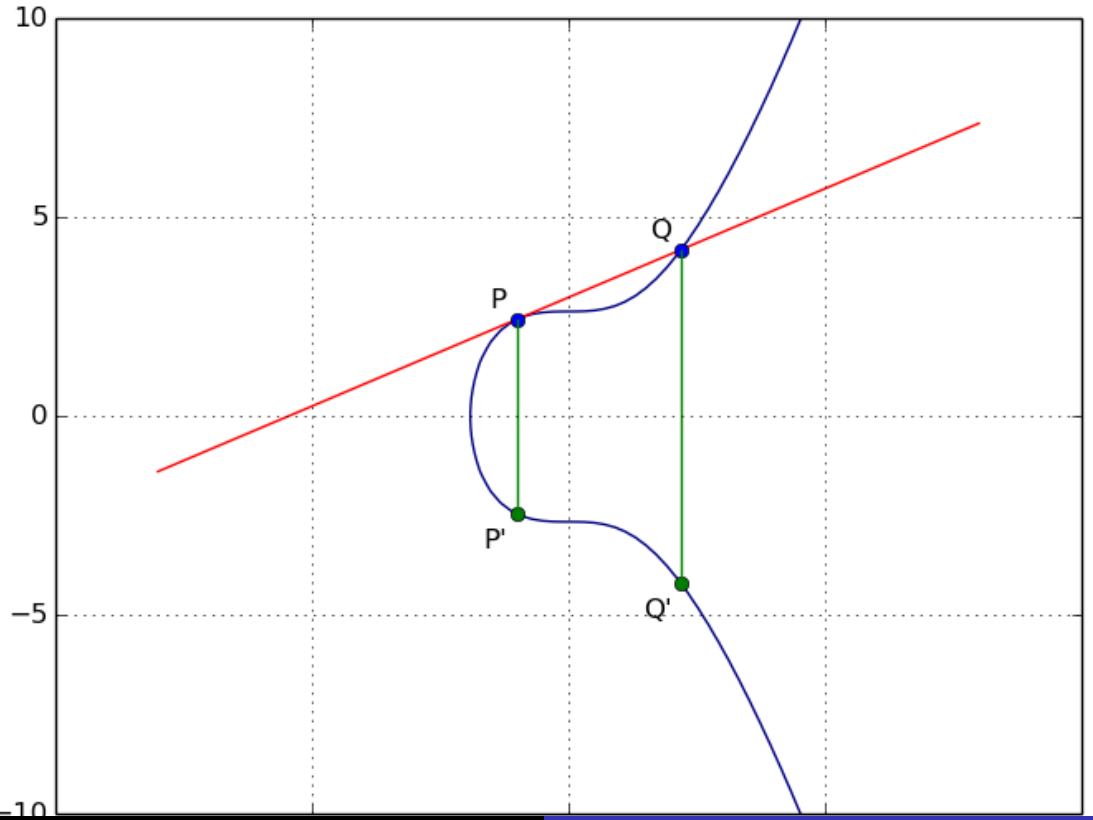
Soient deux points P et Q sur une courbe C.
- Tracez la droite D qui passe par P et Q.
- Soit R le troisième point d'intersection entre D et C.
- Soit R' le symétrique de R par rapport à l'axe des abscisse :

$$P + Q = R'$$

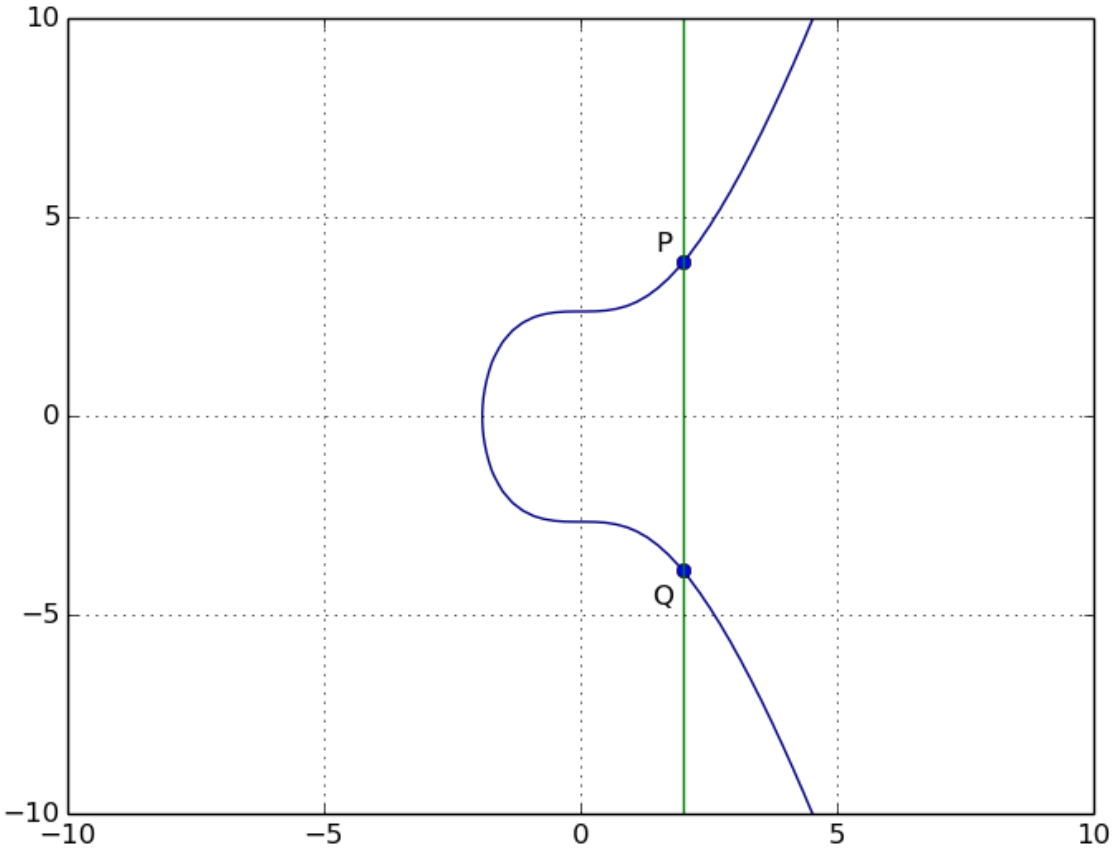
Illustration



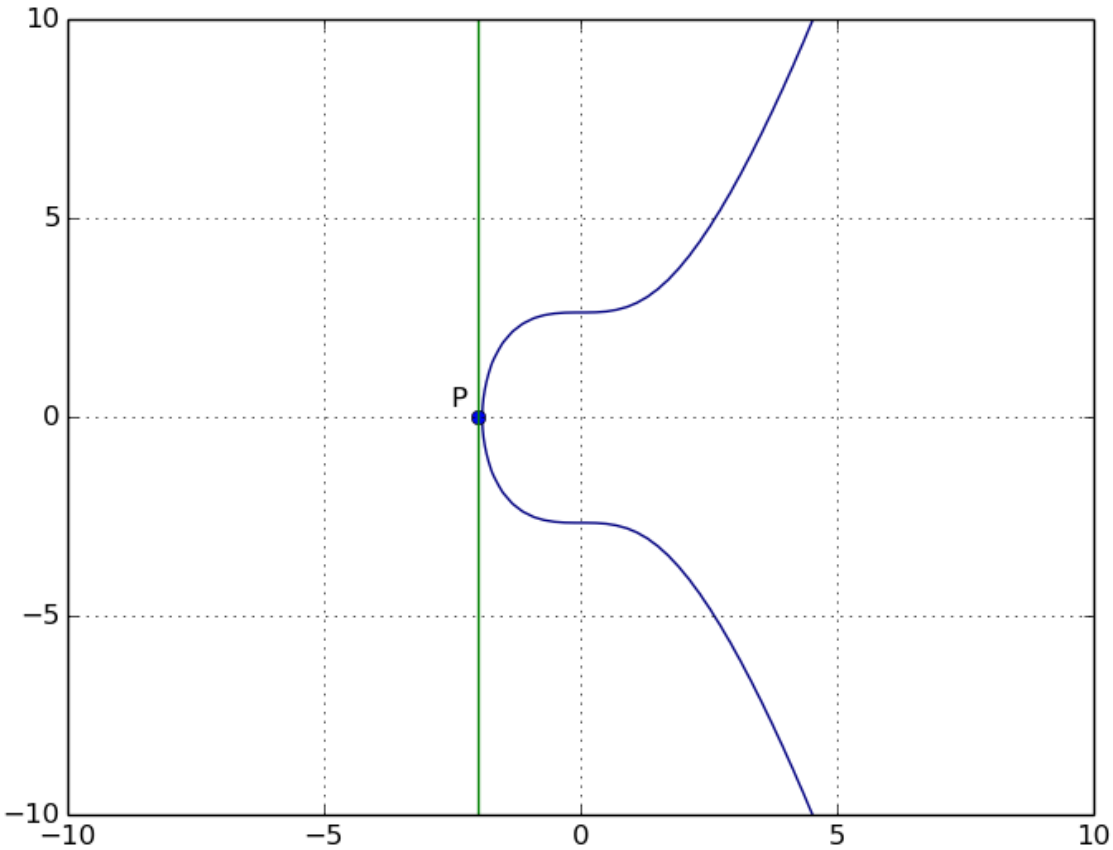
Si la droite ne passe que par deux points de la courbe parce qu'elle lui est tangente, on considère le point de tangente comme deux points distincts, on a: $P + P = Q'$, $P + Q = P'$.



Si la droite qui passe par P et Q est verticale, alors le troisième point se trouve à l'infini. Il s'ensuit que $P + Q = 0$.



P=Q



La multiplication elliptique

Comment calculer n * P ?

L'approche naïve serait d'opérer une succession d'addition.

$P + P + \ldots + P$(n fois).

L'algorithme classique s'appelle double and add .

- Convertissons n dans sa représentation binaire.

Par exemple: $n = 19 \rightarrow 19 = 2^4 + 2^1 + 2^0$

Notons que $19P = 2^4P + 2^1P + 2^0P$

Donc :

$$19P = 2(2(2(2(P)))) + 2(P) + P$$

.

$\Rightarrow$ On calcule ainsi $19P$ en 2 additions et 5 doubling au lieu de 18 additions.

Calcul de clé secrète par les courbes elliptiques

Il s'agit d'un échange similaire à celui de Diffie-Hellman.

- Alice et Bob choisissent publiquement une courbe elliptique $E(a, b, K)$, K est un corps fini (par ex.,$\mathbb{Z}/p\mathbb{Z}$) et une courbe elliptique de la forme: $y^2 = x^3 + ax + b$.
- Ils choisissent publiquement, un point P de la courbe.
- Alice choisit secrètement un entier k_A et Bob choisit un entier k_B
- Alice envoie à Bob le point de la courbe elliptique $k_A P$, et Bob envoie à Alice le point $k_B P$.
- Chacun de son côté calcule $k_A(k_B P) = k_B(k_A P) = (k_A k_B)P$. Le point $k_A k_B P$ de la courbe elliptique constitue la clé secrète commune.

Difficulté d'une cryptanalyse

- Si un pirate a espionné les échages publiques, il sera en possession de $E(a, b, K), P, k_A P, k_B P$

- Pour retrouver la clé secrète, il faut calculer k_A connaissant P et $k_A P$ et, k_B connaissant P et $k_B P$.

- C'est le problème du logarithme discret sur la courbe elliptique.

Le logarithme discret est difficile à résoudre dans les groupes $\mathbb{Z}/p\mathbb{Z}$ et il est très compliqué pour les groupes issus des courbes elliptiques.

Chiffrement d'un message

Supposons qu'Alice veut envoyer à Bob un message M.

- Alice choisit secrètement et aléatoirement un entier n.
- Alice calcule nP et $M + nk_BP$ et envoie ces deux points à Bob.

Déchiffrement

- Avec sa clé secrète k_B, Bob calcule $nk_BP = k_BnP$ à partir de nP.
- Il calcule ensuite $M = (M + nk_BP) - nk_BP$.

Bibliographie

- GILLES LACHAUD. Revue La Recherche, n 346, Octobre 2001, pp. 24-30.
- William Stallings. Sécurité des réseaux. Applications et standards. Vuibert 2002.
- Saiida LAZAAR. Sécurité des réseaux et cryptographie. Edilivre ISBN 2334085693, Mars 2016.
- https://fr.wikipedia.org/
- https://www.miximum.fr/