Éléments de cryptologie

François Pouit

August 28, 2024

Introduction

Plan

Plan

Principe

La cryptologie repose sur le principe suivant : un message source M (" en clair ") est transformé par un algorithme f en un autre message C = f(M), dit " codé " ou " crypté ".
La différence est due aux objectifs visés, qui séparent les algorithmes en trois classes :

A. La transformation est réversible

la fonction f est injective, et la machine peut reconstituer exactement M à partir de C. à cette classe appartiennent les algorithmes de cryptage/décryptage permettant de transmettre des informations confidentielles, et les algorithmes de compression de type ZIP. La différence est que dans le premier cas la reconstitution doit être quasiment impossible sans la connaissance précise de certains éléments appelés " clés ".

B. La transformation est presque réversible

la machine peut reconstituer à partir de C un message " assez proche " de M. C'est le cas des assembleurs, qui passent du langage d'assemblage au langage machine, la transformation inverse étant effectuée par le " désassembleur " (les commentaires du programmeur ainsi que les déclarations et directives sont évidemment perdus); c'est aussi le cas des algorithmes de compression du type JPEG (Join Photograph Expert Group) qui permettent de retrouver une image ressemblant à l'image de départ.

C. La transformation n'est absolument pas réversible

on n'a pas besoin de connaître M à partir de C : f n'est pas injective. Ce mode de cryptage correspond à certains algorithmes permettant d'identifier un utilisateur au moyen d'un mot de passe : la machine garde en mémoire l'algorithme f et le code C, et, quand l'utilisateur entre son mot de passe M, elle compare directement C avec f(M).
Toutefois les algorithmes d'identification usuels utilisent plutôt les standard décrit dans la suite.

Cryptographie

La **cryptographie** vient du grec " kryptos ", caché, est l'art d'établir des communications secrètes. Elle joue un rôle économique et militaire.

Le **chiffrement** ou **cryptage** rend un message incompréhensible pour un adversaire ou intru.

Le **déchiffrement** ou **décryptage** redonne au message reçu son caractère compréhensible.

Cryptanalyse

La **cryptanalyse** est l'art du déchiffrement et du chiffrement des messages " ennemis ".

Un **algorithme cryptographique** est l'ensemble des procédures qui permettent une communication incompréhensible pour un " intrus ".

La **cryptologie** est la partie des mathématiques concernant la cryptographie et la cryptanalyse.

Cryptanalyse

la cryptanalyse a des rapports avec la traduction ou le déchiffrement des langages naturels (français, anglais, breton ...). Il est évidemment impossible de déchiffrer un message unique si l'on a pas une idée de son contenu.

Dans certains cas, il n'y a pas de déchiffrement, on se contente de valider un message crypté (mots de passe, signatures ...).
Il est possible (et la méthode est connue depuis longtemps de faire passer de façon " transparente ", un message crypté dans un message en clair (programme, image, poème...); une image de 1024×1024 peut servir par exemple à cacher un texte de 64 ko.

Il est possible de se servir d'un algorithme de compression comme algorithme de cryptage, si la technique de compression est secrète.

Plan

Algorithmes cryptographiques

Les algorithmes cryptographiques peuvent être :

à algorithme privé : ce sont soit des algorithmes cryptographiques militaires, soit des algorithmes cryptographiques illégaux.

à algorithme publics : c'est le cas des algorithmes cryptographiques utilisés sur internet, et on distingue alors :

* les algorithmes à "clés privées" : dans ce cas, les paramètres de l'algorithme de cryptage et de l'algorithme de décryptage sont connus des deux utilisateurs.

* les algorithmes à " clés publiques " : la clé de cryptage est connue de plusieurs utilisateurs, la clé de décryptage reste privée.

Cryptanalyse

L'hypothèse de base du cryptographe est que le cryptanalyste connaît l'algorithme. C'est à dire que son algorithme doit résister à une attaque même si le cryptanalyste en connaît tous les détails.

On distingue les types d'attaque suivants :

- attaque à texte chiffré connu : le cryptanalyste dispose de plusieurs textes chiffrés avec la même clé, il essaye de trouver des messages lisibles, et si possible la clé.

Cryptanalyse

- attaque exhaustive : avec un seul texte chiffré, essayer toutes les clés possibles pour trouver un message lisible.

- Attaque à texte en clair connu : le cryptanalyste dispose non seulement de plusieurs textes chiffré avec la même clé,mais aussi des textes en clair correspondants il essaye de trouver alors la clé, ou un algorithme de décryptage.

- attaque à texte en clair choisi : le cryptanalyste dispose de la machine à crypter, impossible à désassembler, et il veut décrypter les messages reçus.

Cryptanalyse

- attaque à texte chiffré choisi : inversement, le cryptanalyste dispose de la machine à décrypter, et veut trouver le moyen de crypter de nouveaux messages.

- Attaque des anniversaires : essayer de trouver deux messages en clair produisant le même texte chiffré (pour casser un système d'authentification).

Plan

Les algorithmes de classe A sont eux-mêmes divisés en deux groupes (suivant la connaissance qu'ont les utilisateurs de l'application réciproque f $^{-1}$) :

les algorithmes standard, du type DES, qui utilisent la même clé (secrète) au cryptage et au décryptage.
Si l'algorithme est public, comme DES, celui qui connaît f connaît aussi f $^{-1}$.

Les algorithmes dits " à clé publique ", qui séparent la clé e de cryptage de la clé d de décryptage.
On peut connaître f sans pour autant connaître f $^{-1}$.
Font partie de ce type, l'algorithme de Rivest, Shamir, et Adleman (RSA) et celui de McEliece et Niederreiter.

Algorithmes à permutation

Méthode de chiffrement utilisée depuis l'antiquité : la lettre C(i) est obtenue à partir de M(i) par une permutation de l'alphabet. Le système n'a pas résisté aux progrès des probabilités.

voir
http://fr.wikipedia.org/wiki/Chiffrement_par_décalage

Algorithmes du type VERNAM

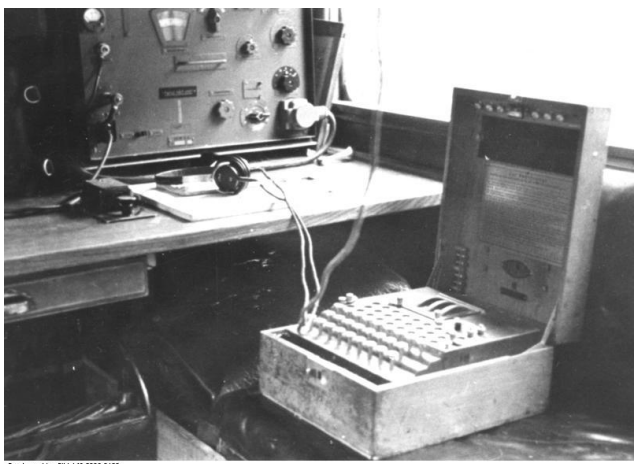
Méthode de chiffrement utilisée par les Allemands jusqu'à la seconde guerre mondiale (machine Enigma) :

voir

`https://fr.wikipedia.org/wiki/Enigma_(machine)`

$C(i) = (M(i) + e(i \bmod l(e)) \bmod 26,$

Ou l(e) est la longueur de la clé e.
Bien que simple, ce système est complètement invulnérable quand il est utilisé en " masque aléatoire jetable ".



Algorithmes du type VERNAM

Le chiffrement par la méthode du masque jetable consiste à combiner le message en clair avec une clé présentant les caractéristiques très particulières suivantes :

La clé doit être une suite de caractères au moins aussi longue que le message à chiffrer.
Les caractères composant la clé doivent être choisis de façon totalement aléatoire.
Chaque clé, (ou masque) , ne doit être utilisée qu'une seule fois (d'où le nom de masque jetable).

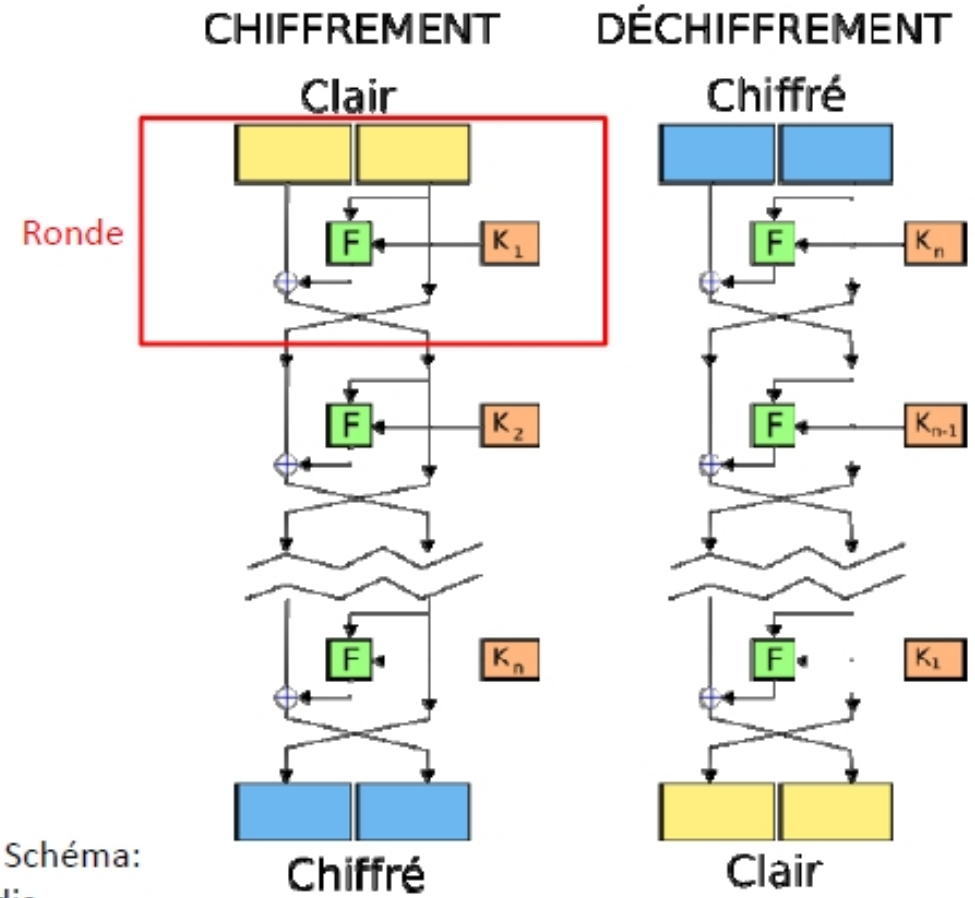
Réseau de Feistel

Proposé par Horst Feistel (1973), il est à la base de la plupart des algorithmes modernes à clés secrètes (DES en particulier).

- Système de chiffrement par blocs
- Division d'un bloc en 2 parties égales
- Chiffrement de la première partie par une fonction F avec une clé K
- Modification de la seconde partie par un XOR avec la première partie chiffrée
- Permutation des 2 parties
- On répète l'opération n fois
- Chaque itération s'appelle une ronde
- A chaque ronde, la clé va changer pour renforcer le processus

Le chiffrement et le déchiffrement s'effectuent suivant le même principe

Réseau de Feistel



Source Schéma: wikipédia

Algorithme DES

Voir :

https://fr.wikipedia.org/wiki/Data_Encryption_Standard

https://web.maths.unsw.edu.au/~lafaye/CCM/crypto/des.htm

https://en.wikipedia.org/wiki/Data_Encryption_Standard

https://en.wikipedia.org/wiki/DES_supplementary_material

Cet algorithme utilise une clé de 8 octets (dont 56 bits significatifs et 8 bits de parité).
Il y a donc 2^{56} clés possibles, soit à peu près 7.10^{16}.

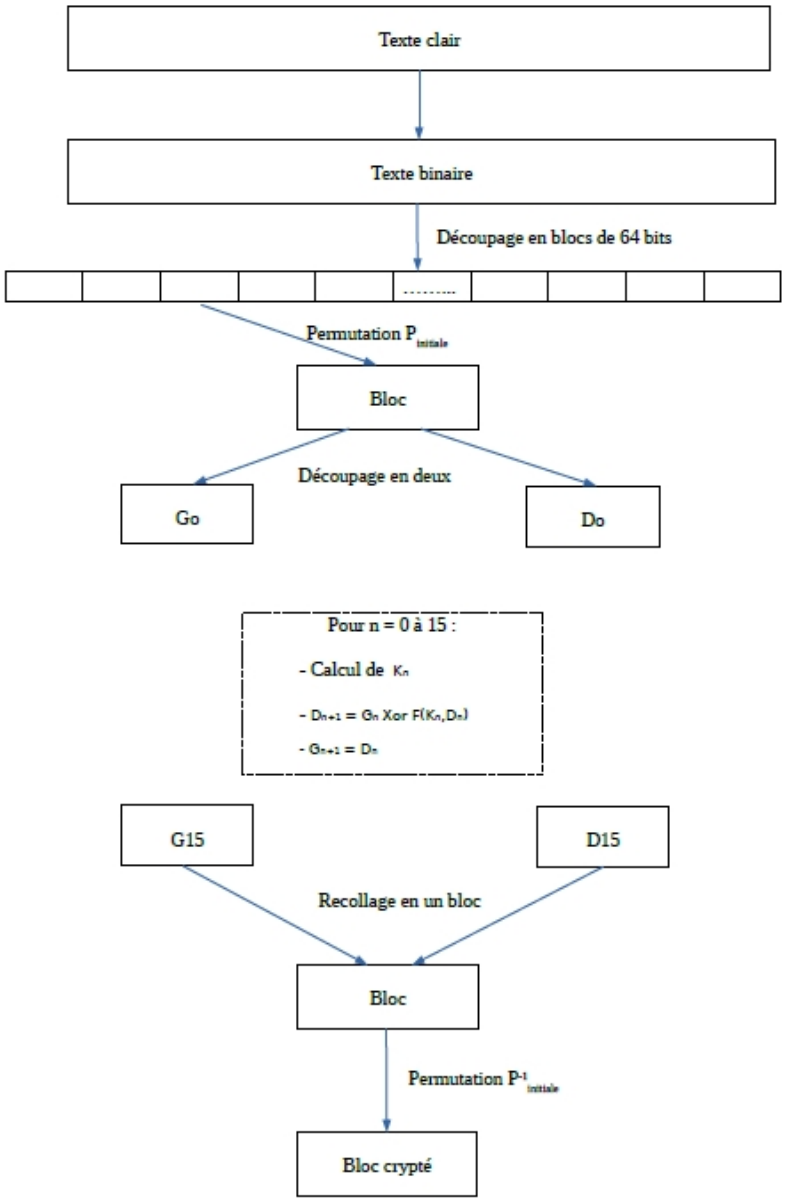
En faisant un essai tous les millionièmes de seconde il faudrait en moyenne 1000 ans pour trouver la bonne clé.
Le cryptage est réalisé par un subtil mélange au niveau du bit.

Algorithme DES : Algorithme général

1. Le texte est fractionné en blocs de 64 bits.
2. Pour chaque bloc on fait :
   - 2.1 une permutation initiale
   - 2.2 un découpage en deux parties G_0 et D_0.
   - 2.3 pour chaque paire (D_n,G_n) faire 16 fois :
     - Détermination d'une clé K_n
     - $D_{n+1} = G_n$ Xor $F(K_n,D_n)$
     - $G_{n+1} = D_n$
   - 2.4 les deux parties G_{15} et D_{15} sont recollées
   - 2.5 une permutation inverse de la première

L'algorithme est très rapide, et 10 fois plus encore s'il est implémenté en circuits logiques.

Algorithme DES : Algorithme général



Algorithme DES

étape 2.1 : La permutation initiale :

PI								
	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

étape 2.2 : on obtient ensuite G_0 et D_0 :

G_0								
	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

D_0								
	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

étape 2.3 calcul de la clé K_n

On part d'une *master key* K de 64 bits au hasard et calculée dès le départ.
Génération de clés à partir de cette *master key* :

- à partir de la clé de 64 bits on extrait à l'aide d'une permutation, une clé de 56 bits.
  puis découpage en deux blocs de 28 bits.(voir table PC-1). Les bits 8,16,24,32,40,48,56 et 64 disparaissent.
- ces blocs subissent un décalage de bits vers la gauche

Ronde	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Nbre de décalage	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

- on recolle les blocs
- on refait une permutation (voir table PC-2) : transformation en clé de 48 bits pour la ronde $= K_n$. Cette permutation supprime les bits 9, 18, 22, 25, 35, 38, 43, 54.

étape 2.3 calcul de la clé K_n

Permuted choice 1 (PC-1)

PC-1

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

Permuted choice 2 (PC-2)

PC-2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

étape 2.3 : fonction F

La fonction F :

- extension du bloc D_n de 32 bits en un bloc D'_n de 48 bits, à l'aide de la transformation E en répétant certains bits :

E						
	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

étape 2.3 : fonction F

- On peut faire un XOR entre D'_n et la clé $K_n = D^*_n$.
- le bloc D^*_n obtenu est ensuite découpé en 8 blocs de 6 bits
- on passe chaque bloc de 6 bits dans une fonction de substitution S :
  - S permet de passer de 6 bits à 4 bits
  - Pour cela les bits 1 et 6 de chaque bloc nous permet de déterminer la ligne
  - Les bits 2,3,4,5 servent à trouver la colonne
  - Avec la ligne et la colonne on peut trouver une valeur qu'il suffit de coder en binaire
- cette fonction S change à chaque ronde
- On recolle ensuite les 8 blocs de 4 bits = 32 bits puis on y applique une permutation (permutation P), ce qui nous donne $F(K_n,D_n)$

étape 2.3 : fonction F

Permutation P :

P

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

étape 2.3 : fonction F

Exemple de fonction S, avec bloc de 6 bits = (110111)

no ligne = $(11)_2 = (3)_{10}$.
no colonne = $(1011)_2 = (11)_{10}$.

si la fonction S1 est :

S_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

le bloc de sortie sera donc $(14)_{10} = (1110)_2$.

Algorithme DES

étape 2.4 : une fois les 16 rondes effectuées on réunis les deux blocs $G = G_{16}$ et $D = D_{16}$.

étape 2.5 : le bloc obtenu subit la permutation inverse de la première :

PI-1								
	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

On obtient ainsi le bloc crypté et on répète l'opération pour tous les blocs.

limites de DES

- taille assez faible
- conception assez opaque de la fonction S (tout de même validée par la cryptanalyse différentielle dans les années 90).
- non adaptation à l'évolution de la puissance de calcul (DES cassé en 1998 par la machine DeepCrack pour un coût de 200000 dollars)
- possibilité de réduire encore le temps (une trentaine d'heure pour craquer une clé DES par force brute avec 1000 pc cadencés à 1Ghz fonctionnant en parallèle)

amélioration en triple DES : à vous de chercher ...

Algorithme AES

Voir :

`https://fr.wikipedia.org/wiki/Advanced_Encryption_Standard`

C'est le nouveau standard de chiffrement pour les organisations du gouvernement des États-Unis. Il a été approuvé par la NSA (National Security Agency) dans sa suite B1 des algorithmes cryptographiques. Il est actuellement le plus utilisé et le plus sécuritaire.

L'algorithme prend en entrée des blocs de 128 bits (16 octets), la clé fait 128, 192 ou 256 bits. Les 16 octets en entrée sont permutés selon une table définie au préalable.

les limites du chiffrement symétrique

- gestion des nombreuses clés
  - une clé par canal de com entre les deux entités
  - explosion du nombre de clés
- échange sécurisé des clés
  - valise diplomatique ?
  - authentification, garantie d'intégrite ...

le chiffrement asymétrique

Principe : James Ellis (1969) et Witfield Diffie (1975)
Couple de clés :

- Clé publique que l'on peut diffuser : P
- Clé secrète que l'on ne communique jamais : p
- Deux fonctions D et C telles que :
- D(a,b) : déchiffrer le cryptogramme b avec la clé a
- C(a,b) : chiffrer le message b avec la clé a
- D(p,C(P,M)) = D(P,C(p,M)) = M

le chiffrement asymétrique

La construction du couple de clés
- Choix dune clé secrète aléatoire
- Génération de la clé publique à partir de la clé secrète

Par contre, échange des clés par un canal non sûr, espionné par Roger...

- Force de l'algorithme = relation entre les deux clés
- Faire en sorte qu'il soit impossible de retrouver la clé privée à partir de la publique
- Utilisation de fonctions unidirectionnelles munies de portes arrières

le chiffrement asymétrique

- Fonction unidirectionnelle y=f(x) telle que si on connaît y il est très difficile voire impossible de retrouver x sans la clé privée.
- Exemple : factorisation des grands nombres
- Fonction munie d'une porte arrière : il existe x = g(y, z) telle que si l'on connaît z, il est facile de calculer x à partir de y
- Toute la difficulté consiste à trouver f et g : mathématiquement complexe !

Algorithme RSA de Ron Rivest, Adi Shamir, et Len Adelman

Rivest, Shamir et Adleman (1977)

- Basé sur la factorisation de deux grands nombres entiers.
- Utilisation de nombres premiers
- Pas de limite sur la taille des clés
- Adaptation à la loi de Moore
- RSA 512 et RSA 768 cassés en 1999 et 2010
- Plus de 5000 coeurs utilisés par l'INRIA pour RSA 768
- Utilisation de clés d'au moins 1024 ou 2048 bits

Algorithme RSA : principe

Cet algorithme est fondé sur l'hypothèse que la recherche des facteurs premiers d'un nombre entier n de plus de 100 chiffres est impossible à réaliser dans un délai raisonnable (c'est à dire au moins un siècle).

Du point de vue du cryptage il est au moins 100 fois plus lent que DES. Le message à envoyer est divisé en morceaux, et chaque morceau est converti en un nombre entier positif M_i.

Algorithme RSA : principe

On choisit d'abord deux nombres premiers p et q le plus grand possible (au moins 50 chiffres). Pour que l'algorithme fonctionne correctement, il faut que chaque nombre M_i (représentant une partie du texte à coder) soit inférieur à p et q.

On pose $n = p \times q$ et $\varphi(n)$ la fonction d'Euler.
On choisit un autre entier e, entre 2 et $\varphi(n)$, et premier avec $\varphi(n)$. Le plus simple est de prendre e premier, ne divisant pas $\varphi(n)$.

Algorithme RSA : principe

On résout ensuite une identité de Bézout :
ed + k φ(n) = 1.
afin de trouver l'inverse d de e dans Z/φ(n)Z. (on peut le faire en remontant l'algorithme d'Euclide).

Les nombres n et e sont les clés publiques de cryptage.

Les nombres p, q et φ(n) peuvent être oubliés, mais ne doivent absolument pas être rendus publics.

Algorithme RSA : principe

Le cryptage :

Supposons le message (ou une partie) transformé en un entier M $<$ p et q. Le crypteur calcule le reste de la division de M^e par n : M'.

Le décryptage :

Le décrypteur reçoit le nombre M'. il calcule le reste de la division de M'^d par n, qui par le th d'Euler n'est autre que M :

Algorithme RSA : principe

Ainsi pour décrypter un message reçu crypté il faut connaître d.
comme dit plus haut la clé publique est n et e.
Les nombres p, q et ϕ(n) sont secrets.
Pour connaître d il faut connaître donc ϕ(n), c'est à dire être capable de décomposer n en produit de facteurs premiers.
Or ce problème est très coûteux à résoudre car il demande un temps calcul trop important pour des nombres n très grands.

C'est sur ce fait qu'est basé la sécurité du RSA.

voir
https://fr.wikipedia.org/wiki/Décomposition_en_produit_de_facteurs_premiers

Algorithme RSA : principe

Pour générer la clé il faut trouver des nombres premiers très grands.

Il existe des algorithmes probabilistes très rapides pour trouver des grands nombres premiers (p et q).
Ces algorithmes offrent une probabilité très forte d'obtention de nombres premiers.

C'est à l'aide de ces algorithmes que l'on génère une clé RSA.

voir
https://fr.wikipedia.org/wiki/Test_de_primalité

Algorithme RSA : décryptage

Montrons que $M'^{d} \equiv M$ [n].

En effet :

le cryptage nous donne :
$M' \equiv M^{e}$ [n], donc :

$M'^{d} \equiv (M^{e})^{d}$ [n] $\equiv M^{ed}$ [n].

or : ed $\equiv 1$ [ϕ(n)], avec ϕ(n) = (p-1)(q-1).
donc $M'^{d} \equiv M.M^{k(p-1)(q-1)}$ [n], k étant un entier.
donc $M'^{d} \equiv M.(M^{(p-1)})^{k(q-1)}$ [n].

Algorithme RSA : décryptage

donc $M'^d \equiv M.M^{k(p-1)(q-1)}$ [n], k étant un entier.
donc $M'^d \equiv M.(M^{(p-1)})^{k(q-1)}$ [n].

Or M est premier avec p, donc par le petit th de Fermat :

$M^{p-1} \equiv 1$[p]
c'est à dire : $(M^{(p-1)})^{k(q-1)} \equiv 1$[p]
donc $(M^{(p-1)})^{k(q-1)} = 1 + ap$, avec a entier.

De même M est premier avec q donc :
$(M^{(q-1)})^{k(p-1)} = (M^{(p-1)})^{k(q-1)} = 1 + bq$, avec b entier.

Algorithme RSA : décryptage

donc ap = bq, avec p et q premiers entre eux.

par le th de Gauss : a est divisible par q et b divisible par p, donc :
a = a'q et b = b'p, a' et b' étant des entiers.

On reporte cela, et on obtient :

$(M^{(p-1)})^{k(q-1)} = 1+a'pq = 1+a'n \equiv 1[n]$

et donc finalement : $M'^{d} \equiv M\ [n]$: CQFD.

Algorithme RSA : bilan

Standard dans le domaine de la cryptographie asymétrique.

- Fiable si clés de taille suffisante
- Adaptation à la loi de Moore et évolution de la clé

Algorithme d'ElGamal (1987)
- Alternative à RSA (utilisation de RSA limitée par brevet jusqu'au début des années 2000)
- El Gamal : non breveté
- Utilisé dans PGP (Pretty Good Privacy) et GPG (GNU Privacy Guard)
- Utilisé pour les signatures électroniques dans DSA (Digital Signature Algorithm)

Algorithme RSA : bilan

RSA : Communications sécurisées pour tous

- Chiffrement asymétrique basé sur RSA
- Publication de la clé publique de tous les partenaires
- Stockage des clés publiques dans des annuaires hébergés chez des tiers de confiance
- Annuaire : maillon faible de la solution
- Violation d'intégrité des clés publiques et MITM

RSA et authentification

- Chiffrement avec clé privée, déchiffrement avec clé publique associée
- Preuve de source du message
- Notion de challenge et authentification du partenaire

Symétrique VS asymétrique

le chiffrement à clé symétrique BEAUCOUP plus rapide !

Débit de chiffrement

	Matériel dédié	Logiciel
RSA Clés 1024 bits	300kb/s	21,6kb/s
DES Clé de 56 bits	300Mb/s	2,1Mb/s

donc basculement vers le chiffrement symétrique après échange de la clé secrète, protégée au moyen d'un algorithme asymétrique.

protocole d'échange Diffie-Hellmann

Comment s'assurer de la confidentialité lors de l'échange de messages.
Ce protocole d'échange a la particularité de ne pas nécessiter la rencontre préalable à la transmission du message entre l'expéditeur (Alice) et le destinataire (Bob).

protocole d'échange Diffie-Hellmann

Fonction dont le calcul direct est facile mais le calcul inverse est extrêmement complexe