RSA

1. Introduction

Tout le monde s'accorde à dire que le système élaboré au MIT en 1977 par les cryptologues Ronald Rivest,Adi Shamir et Leonard Addleman figure parmis les plus sûrs. Il a depuis été nommé RSA, initiales des noms de ses concepteurs. Cette méthode de cryptage est dite à clé publique, et elle est basée sur l'utilisation de l'exponentielle modulaire, qui est une fonction mathématique aux propriétés particulières.
Les principales applications de RSA sont bien sûr la transmission de messages codés,mais aussi l'authentification de messages,fonction appelée aussi « signature électronique ».

2. Les systèmes à clé publique

En cryptologie, on ne distingue généralement que deux types d'algorithmes à clé : les algorithmes à clé secrète et les algorithmes à clé publique.
Les premiers fondent leur sécurité sur la robustesse de l'algorithme de codage, et non de celle de la clé; DES en fait partie.
Le deuxième type d'algorithme puise sa force en l'impossibilité pratique de résoudre un problème mathématique, pour lequel trouver une solution se traduit par un temps de calcul monumental. On se trouve donc dans un cas d'infaisabilité ; RSA fonctionne selon ce schéma.

Dans le cas des systèmes à clé publique, l'originalité vient du fait qu'il existe deux clés distinctes : une pour le **cryptage** des données et une pour le **décryptage**. Ces deux clés sont déterminées simultanément par une même personne, à une clé de codage correspond une seule clé de décodage et inversement. Le point le plus important est l'impossibilité de déduire une des deux clés à partir de l'autre en dans un délai raisonnable.
L'algorithme peut être connu, une des clés (celle de codage ou celle de décodage) peut être publiée, d'où l'appellation de clé publique, mais la clé nécessaire à l'opération inverse (codage ou décodage) doit être gardée secrète.

2.1. Confidentialité

La confidentialité est la première application possible des algorithmes à clé publique, elle consiste à permettre à une personne de transmettre un message à un correspondant, en garantissant que personne d'autre ne puisse en prendre connaissance.
La mise en oeuvre de ce procédé est très simple : la personne désirant recevoir les messages (destinataire), calcule un couple de clés pour l'algorithme RSA. Une fois ces clés établies, il va publier la clé de codage (clé publique) et garder secrète la clé de décodage (clé privée). Grâce aux propriétés des algorithmes à clé publique comme RSA, toute personne pourra crypter un message au moyen de la clé donnée par le destinataire, mais seul celui-ci pourra les déchiffrer avec sa clé gardée secrète.

2.2. Authenticité

L'authenticité est le deuxième type d'application possible avec RSA. Il s'agit dans ce cas de permettre à une personne recevant un message d'être sûr de sa provenance, c'est pourquoi on appelle aussi ce procédé signature électronique.
La mise en place d'un tel système est aussi facile que pour la confidentialité : cette fois c'est la personne qui désire signer ses messages (émetteur) qui calcule le couple de clés nécessaires au fonctionnement de RSA. Mais c'est la clé de déchiffrage qui est publiée, alors que la clé de chiffrage est gardée secrète. L'émetteur pourra alors crypter ses messages et les envoyer a toute personne voulant les recevoir, il suffit pour les décoder d'utiliser la clé de décryptage publiée. Comme seule la personne ayant la clé de cryptage correspondante est capable de chiffrer les messages, on peut donc être certain de leur provenance

2.3. Combinaison

Il peut être nécessaire, dans certains cas, d'assurer simultanément la confidentialité et l'authenticité des informations transmises. RSA n'interdit pas cette possibilité mais elle est un peu plus complexe puisqu'il faut alors utiliser deux couples de clés.
Le premier couple de clés est défini par l'émetteur, qui va publier sa clé de décodage et garder secrète sa clé de codage.
Le deuxième couple est déterminé par le destinataire, qui va fournir une clé de codage et garder secrète sa clé de décodage.
La procédure à suivre est alors la suivante :
- l'émetteur chiffre le message avec sa clé de signature secrète.
- l'émetteur code le fichier obtenu avec la clé de cryptage publique fournie par le destinataire
- le résultat est transmis.
- le destinataire décode le fichier reçu avec sa clé de décryptage secrète.
- le destinataire vérifie la provenance du message avec la clé de signature publiée par l'émetteur.

3. L'algorithme RSA

Petite définition pour commencer : on appelle exponentielle modulaire de base a, d'exposant b et de modulo n la fonction suivante: **f(a) = a^b mod n**

3.1. Un peu de mathématiques

L'algorithme RSA repose sur l'opération modulaire d'élévation à la puissance. Si on utilise les éléments suivants :
x : données à envoyer (vues comme un nombre),
y : données cryptées (idem)
e : clé de chiffrement
d : la clé déchiffrement
Alors tout le principe de RSA peut se résumer par les 2 relations suivantes:

y = x ^ e modulo n (cryptage)
x = y ^ d modulo n (décryptage)

- la clé publique est le couple (e,n)
- la clé privée est le couple (d,n).
- n est le nombre caractéristique de l'arithmétique modulaire. Il doit être pris très grand pour assurer une bonne protection (plusieurs centaines de chiffres) et doit être connu de la source et de la destination.

note : Le texte ou les données à coder,quelles qu'elles soient,doivent être découpées en blocs **et traduites en une suite de chiffres** pour pouvoir être cryptées par RSA.Il faut bien vérifier que les chiffres obtenus ne sont pas susceptibles d'être supérieurs à n, sinon ils seraient tronqués par le modulo

3.2. Détermination des clés

La méthode de cryptage/décryptage est en fait très simple, seule la détermination des clés est délicate car elle nécessite une puissance de calcul relativement importante au regard de la taille des chiffres manipulés.

3.2.1. Choix de n

n doit simplement être le produit de deux nombres premiers p et q :
n = p.q
on calcule alors F(n) de la façon suivante (F est la fonction d'Euler) :
F(n) = (p-1)(q-1)

3.2.2. Choix de e et d

On se base sur la formule :
e.d = K.F(n)+1 = 1 mod F(n)

Cette formule doit être vérifiée pour qu' une exponentielle modulaire de modulo n et d'exposant e ait une inverse de modulo n et d'exponentielle e
K est une constante quelconque à choisir de façon d'éviter d'avoir e ou d sous forme fractionnaire,ce qui complique le traitement.

Les nombres d et e se calculent de la façon suivante :

- Calcul de d tel que PGCD(d, F(n)) = 1 … en clair, d et F(n) doivent être premiers entre eux) et 1 < d < F(n) (d est non nul dans l'arithmétique modulo F(n)) : il suffit de prendre **par exemple** d premier > max(p,q).
- Calcul de e tel que : e.d = 1 mod F(n) (e et d sont inverses dans l'arithmétique)
- Vérifier que e et n sont premiers entre eux
- Vérifier que e est dans l'intervalle [2,F(n)-1]

Les trois paramètres e,d et n sont les seuls nécessaires au fonctionnement de RSA,toutes les autres données ayant permis de les déterminer sont a détruire.

3.2.3. Exemple

Pour illustrer cette méthode complexe de choix de clés,voici un petit exemple numérique qui a le mérite d'être vérifiable sur une calculatrice sans la saturer.

1ere étape : choix de n

n = p.q (produit de deux nombres premiers)
Prenons p=3 et q=5, on a alors n=3*5=15 et f(n)=(p-1)*(q-1)=8

2e étape : choix de e et d

e.d = K.F(n)+1 = 1 mod F(n)

d=7 (premier ,>max(3,5) et <8)
k=6
donc e=(6*8+1)/7 = 7 (7 et 15 premiers entre eux et 7 appartient à [2,7]

A ce stade nous possédons les trois éléments composant les deux clés :
la base n = 15
la clé de décodage d = 7
la clé de codage e = 7
Les autres paramètres p,q et F(n) doivent être détruits et oubliés.

4. Sécurité de RSA

Toute la sécurité de RSA repose sur une simple constatation algorithmique: il est beaucoup plus aisé de calculer un produit de nombres que de calculer la décomposition en facteurs premiers. Cette assertion prend toute sa force lorsqu'on parle de grands nombres comportant plusieurs centaines de chiffres.
Quatre type d'approches peuvent être définies pour attaquer RSA:
1) Méthode brutale (essai de toutes les clés possibles)
2) Factoriser n en facteurs premiers, ce qui permettrait de calculer F(n) = (p-1).(q-1) et donc la clé privée à partir de la relation e.d = 1 mod F(n)… car la clé publique est connue.
3) Trouver j(n) directement sans calculer p et q (même remarque)
4) Trouver d directement sans calculer j(n)

Pour se protéger contre le premier type d'attaque, il suffit classiquement d'augmenter la taille du domaine des clés. Pour le second, aucun algorithme n'a été trouvé à ce jour pour permettre de factoriser efficacement un grand nombre. Des estimations récentes des algorithmes existants évaluent à 1 an la factorisation d'un nombre à 150 chiffres par une machine dont le coût de fabrication serait de 10 millions de dollars. Ces mêmes estimations précisent qu'il faudrait 1 000 ans pour factoriser un nombre à 200 chiffres avec un coût de mise au point de plusieurs milliards de dollars. On peut remarquer qu'augmenter même très légèrement la taille des clés est tout de suite très payant. En ce qui concerne les types 3 et 4, les algorithmes pour trouver j(n) et d sont aussi coûteux qu'une factorisation proprement dite, voire plus.

En fait si l'on utilise une base n de plus de 200 chiffres décimaux l'opération s'avère informatiquement infaisable, même avec des clés de codage e et d petites. A titre d'exemple le temps moyen de décomposition d'un modulo n de 150 chiffres est presque d'un million d'années. En pratique le modulo est généralement codé sur 512 bits

5. Jugement du système

5.1. Points forts

Un des gros avantages de RSA est d'être un algorithme de chiffrement à clef publique. Il échappe ainsi aux inconvénients majeurs des algorithmes à clef secrète, notamment la transmission des clefs.
RSA est actuellement considéré comme incassable en un temps ou avec des moyens raisonnables.

5.2.Points faibles

Un premier problème réside dans le choix des deux nombres premiers p et q. Il est en effet préférable de ne pas choisir ces nombres sans précautions. Les concepteurs du système RSA ont donné un certain nombre de règles qu'il est conseillé de suivre lorsqu'il s'agit de choisir le quadruplet (p, q, d, e), car un mauvais choix de ces paramètres peut rendre le système de codage relativement vulnérable et cassable par un bon algorithme de factorisation spécialisé.

Choix de p et q :
- prendre p et q de tailles sensiblement différentes, mais pas trop.
- choisir des nombres premiers p et q "sûrs", de la forme 2x + 1, avec x premier, et tel que (x – 1) possède de grands facteurs premiers.

Choix de d et e :
- choisir d en premier, tel que d et (p -1)*(q - 1) soient premiers entre eux.
- choisir ensuite e et vérifier que e >> log2 (n).
- ne retenir e que si n et e sont premiers entre eux.

Le second problème est dû au fait que l'ensemble des nombres premiers connus est fini. Le calcul de nouveaux nombres nécessitant des factorisations dont la complexité est exponentielle, il en résulte que la découverte rapide de nouveaux entiers premiers est informatiquement difficile. Cela met en danger le RSA, dans la mesure où les nombres premiers utilisables ne sont pas si nombreux, et ne doivent pas être utilisés trop souvent, ce qui risquerait de favoriser la cryptanalyse.

Les grands organismes de recherche en cryptologie dépensent beaucoup de temps et d'argent pour vérifier l'invulnérabilité des algorithmes à clefs publiques, et donner des règles pour bien les utiliser. Mais leur préoccupation s'oriente surtout autour du point central qui assure la sécurité de RSA : à savoir qu'on ne connaît pas d'algorithme rapide pour calculer la factorisation d'un grand nombre.
Même s'il n'est pas prouvé que RSA soit aussi difficile que le problème de la factorisation, si jamais quelqu'un mettait au point un tel algorithme, le RSA serait définitivement abandonné.

6. Applications

6.1. Cryptage d'un message

Le noyau du RSA accepte comme message d'entrée un entier et renvoie un autre nombre entier correspondant au premier crypté.Or le message originel n'est pas nécessairement sous forme d'un tableau d'entier. Il est donc indispensable de transformer ce message (que l'on assimilera à une chaîne de caractères) en une série de nombres qui eux, pourront être cryptés.

Le message M à crypter doit être découpé en blocs numériques mi ayant chacun une représentation unique modulo n : par exemple, si n fait x chiffres, on peut découper M en blocs de (x-1) chiffres.
- La formule de chiffrement pour chaque bloc mi est : ci = mi^e mod n
- Pour déchiffrer un message, on prends chaque bloc ci on calcule : mi = ci^d mod n
(Le message aurait évidemment pu être chiffré avec d et déchiffré avec e).

6.1.1. Génération des clés

Prenons par exemple p = 47 et q = 71 (deux nombres premiers), alors :
n = p*q = 3337

La clef de déchiffrement d ne doit pas avoir de facteurs communs avec :
F(n) = (p - 1)*(q - 1) = 46*70 = 3220

Choisissons d=79 (premier, >max(47,71) et <3220). Dans ce cas :

e.d = K.F(n)+1 = 1 mod F(n)
e=(1+25*3220)/79 = 79^(-1) mod 3220 = 1019

Il nous reste alors à publier d et n, à oublier p et q, et à garder e secret.

6.1.2. Cryptage

Pour crypter le message : m = 688232687966683, on le découpe en blocs de trois chiffres (n fait 4 chiffres). On obtient :
m1 = 688
m2 = 232
m3 = 687
m4 = 966
m5 = 668
m6 = 3

Le premier bloc est chiffré par :
688^1019 mod 3337 = 2441

De même pour les autres blocs :
232^1019 mod 3337 = 1385
687^1019 mod 3337 = 1592

966^1019 mod 3337 = 151
668^1019 mod 3337 = 2677
3^1019 mod 3337 = 2140

donc c= 2441 1385 1592 0151 2677 2140

Le déchiffrement s'effectue de la même manière, mais en utilisant la clef de déchiffrement.
Donc :

2441^79 mod 3337 = 688
1385^79 mod 3337 = 232
1592^79 mod 3337 = 687
151^79 mod 3337 = 966
2677^79 mod 3337 = 668
2140^79 mod 3337 = 3

on retrouve bien m = 688 232 687 866 668 3

6.2. Décryptage sans connaissance de la clé de décryptage

soit le message crypté :
c= 2441 1385 1592 0151 2677 2140

nous sommes en possession de la clé publique de cryptage :
e = 1019
n = 3337

pour décrypter le message, il nous faut la clé privée de décryptage :
d = ?
n = 3337

nous savons que :
n = p.q
F(n) = (p-1)(q-1)
e.d = K.F(n)+1 = 1 mod F(n)

après moultes calculs, on trouve que :
3337 = 47 * 71
donc :
F(n) = 3220
et :
d = (K.3220+1)/1019 □ d=79 si K=25

la clé privée est donc (79,3337) et le message peut être décrypté !