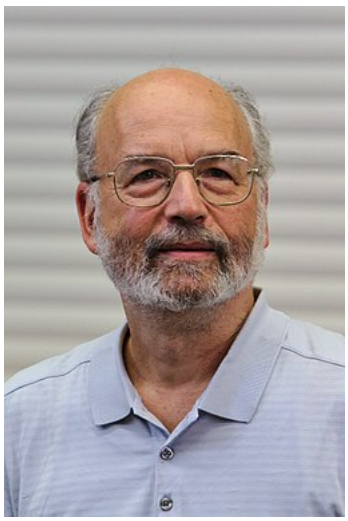
WIKIPÉDIA

Chiffrement RSA

Le **chiffrement RSA** (nommé par les initiales de ses trois inventeurs) est un algorithme de cryptographie asymétrique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet. Cet algorithme a été décrit en 1977 par Ronald Rivest, Adi Shamir et Leonard Adleman. RSA a été breveté[1] par le Massachusetts Institute of Technology (MIT) en 1983 aux États-Unis. Le brevet a expiré le 21 septembre 2000.

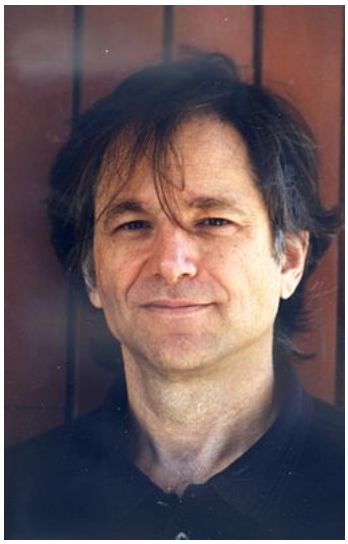
Ronald Rivest (2015).

Sommaire



Adi Shamir (2013).

Fonctionnement général


Leonard Adleman (2010).

Le chiffrement RSA est *asymétrique* : il utilise une paire de clés (des nombres entiers) composée d'une *clé publique* pour chiffrer et d'une *clé privée* pour déchiffrer des données confidentielles. Les deux clés sont créées par une personne, souvent nommée par convention *Alice*, qui souhaite que lui soient envoyées des données confidentielles. Alice rend la clé publique accessible. Cette clé est utilisée par ses correspondants (*Bob*, etc.) pour chiffrer les données qui lui sont envoyées. La clé privée est quant à elle réservée à Alice, et lui permet de déchiffrer ces données. La clé privée peut aussi être utilisée par Alice pour signer une donnée qu'elle envoie, la clé publique permettant à n'importe lequel de ses correspondants de vérifier la signature.

Une condition indispensable est qu'il soit « calculatoirement impossible » de déchiffrer à l'aide de la seule clé publique, en particulier de reconstituer la clé privée à partir de la clé publique, c'est-à-dire que les moyens de calcul disponibles et les méthodes connues au moment de l'échange (et le temps que le secret doit être conservé) ne le permettent pas.

Le chiffrement RSA est souvent utilisé pour communiquer une clé de chiffrement symétrique, qui permet alors de poursuivre l'échange de façon confidentielle : Bob envoie à Alice une clé de chiffrement symétrique qui peut ensuite être utilisée par Alice et Bob pour échanger des données.

Fonctionnement détaillé

Ronald Rivest, Adi Shamir et Leonard Adleman ont publié leur chiffrement en 1978 dans *A Method for Obtaining Digital Signatures and Public-key Cryptosystems*. Ils utilisent les congruences sur les entiers et le petit théorème de Fermat, pour obtenir des fonctions à sens unique, avec brèche secrète (ou porte dérobée).

Tous les calculs se font modulo un nombre entier n qui est le produit de deux nombres premiers. Le petit théorème de Fermat joue un rôle important dans la conception du chiffrement.

Les messages clairs et chiffrés sont des entiers inférieurs à l'entier n (tout message peut être codé par un entier). Les opérations de chiffrement et de déchiffrement consistent à élever le message à une certaine puissance modulo n (c'est l'opération d'exponentiation modulaire).

La seule description des principes mathématiques sur lesquels repose l'algorithme RSA n'est pas suffisante. Sa mise en œuvre concrète demande de tenir compte d'autres questions qui sont essentielles pour la sécurité. Par exemple le couple (clé privée, clé publique) doit être engendré par un procédé vraiment aléatoire qui, même s'il est connu, ne permet pas de reconstituer la clé privée. Les données chiffrées ne doivent pas être trop courtes, pour que le déchiffrement demande vraiment un calcul modulaire, et complétées de façon convenable (par exemple par l'Optimal Asymmetric Encryption Padding).

Création des clés

L'étape de création des clés est à la charge d'Alice. Elle n'intervient pas à chaque chiffrement car les clés peuvent

être réutilisées, la difficulté première, que ne règle pas le chiffrement, est que Bob soit bien certain que la clé publique qu'il détient est celle d'Alice. Le renouvellement des clés n'intervient que si la clé privée est compromise, ou par précaution au bout d'un certain temps (qui peut se compter en années).

1. Choisir p et q, deux nombres premiers distincts ;
2. calculer leur produit $n = pq$, appelé *module de chiffrement* ;
3. calculer $\varphi(n) = (p - 1)(q - 1)$ (c'est la valeur de l'indicatrice d'Euler en n) ;
4. choisir un entier naturel e premier avec $\varphi(n)$ et strictement inférieur à $\varphi(n)$, appelé *exposant de chiffrement* ;
5. calculer l'entier naturel d, inverse de e modulo $\varphi(n)$, et strictement inférieur à $\varphi(n)$, appelé *exposant de déchiffrement* ; d peut se calculer efficacement par l'algorithme d'Euclide étendu.

Comme e est premier avec $\varphi(n)$, d'après le théorème de Bachet-Bézout il existe deux entiers d et k tels que $ed + k\varphi(n) = 1$, c'est-à-dire que $ed \equiv 1 \pmod{\varphi(n)}$: e est bien inversible modulo $\varphi(n)$.

Le couple (n,e) est la *clé publique* du chiffrement, alors que le nombre d est sa *clé privée*[2], sachant que l'opération de déchiffrement ne demande que la clef privée d et l'entier n, connu par la clé publique (la clé privée est parfois aussi définie comme le triplet (p, q, d)[3]).

Chiffrement du message

Si M est un entier naturel strictement inférieur à n représentant un message, alors le message chiffré sera représenté par

$$C \equiv M^e \pmod{n},$$

l'entier naturel C étant choisi strictement inférieur à n.

Déchiffrement du message

Pour déchiffrer C, on utilise d, l'inverse de e modulo $(p - 1)(q - 1)$, et l'on retrouve le message clair M par

$$M \equiv C^d \pmod{n}.$$

Exemple

Un exemple avec de petits nombres premiers (en pratique il faut de très grands nombres premiers) :

1. on choisit deux nombres premiers $p = 3$, $q = 11$;
2. leur produit $n = 3 \times 11 = 33$ est le module de chiffrement ;
3. $\varphi(n) = (3 - 1) \times (11 - 1) = 2 \times 10 = 20$;
4. on choisit $e = 3$ (premier avec 20) comme exposant de chiffrement ;
5. l'exposant de déchiffrement est $d = 7$, l'inverse de 3 modulo 20 (en effet $ed = 3 \times 7 \equiv 1 \mod 20$).

La clé publique d'Alice est $(n, e) = (33, 3)$, et sa clé privée est $(n, d) = (33, 7)$. Bob transmet un message à Alice.

- Chiffrement de $M = 4$ par Bob avec la *clé publique* d'Alice : $4^3 \equiv 31 \mod 33$, le chiffré

est $C = 31$ que Bob transmet à Alice ;
- Déchiffrement de $C = 31$ par Alice avec sa *clé privée* : $31^7 \equiv 4 \bmod 33$, Alice retrouve le message initial $M = 4$.

Le mécanisme de signature par Alice, à l'aide de sa clé privée, est analogue, en échangeant les clés.

Justification

La démonstration repose sur le petit théorème de Fermat, à savoir que comme p et q sont deux nombres premiers, si M n'est pas un multiple de p on a la première égalité, et la seconde s'il n'est pas un multiple de q :

$$M^{p-1} \equiv 1 \pmod{p}, \quad M^{q-1} \equiv 1 \pmod{q}.$$

En effet

$$C^d \equiv (M^e)^d \equiv M^{ed} \pmod{n}.$$

Or

$$ed \equiv 1 \pmod{(p-1)(q-1)}$$

ce qui signifie que pour un entier k

$$ed = 1 + k(p-1)(q-1)$$

donc, si M n'est pas multiple de p d'après le petit théorème de Fermat

$$M^{ed} \equiv M^{1+k(p-1)(q-1)} \equiv M \cdot \left(M^{p-1}\right)^{k(q-1)} \equiv M \pmod{p}$$

et de même, si M n'est pas multiple de q

$$M^{ed} \equiv M \pmod{q}.$$

Les deux égalités sont en fait réalisées pour n'importe quel entier M, car si M est un multiple p, M et toutes ses puissances non nulles sont congrus à 0 modulo p. De même pour q.

L'entier $M^{ed} - M$ est donc un multiple de p et de q, qui sont premiers distincts, donc de leur produit $pq = n$ (on peut le voir comme une conséquence de l'unicité de la décomposition en facteurs premiers, ou plus directement du lemme de Gauss, sachant que p et q sont premiers entre eux, étant premiers et distincts).

Asymétrie

On constate que pour chiffrer un message, il suffit de connaître e et n. En revanche pour déchiffrer, il faut d et n.

Pour calculer d à l'aide de e et n, il faut trouver l'inverse modulaire de e modulo $(p-1)(q-1)$, ce que l'on ne sait pas faire sans connaître les entiers p et q, c'est-à-dire la décomposition de n en facteurs premiers.

Le chiffrement demande donc de pouvoir vérifier que de « très grands » nombres sont des nombres premiers, pour pouvoir trouver p et q, mais aussi que le produit de ces deux très grands nombres, ne soit pas factorisable pratiquement. En effet les algorithmes efficaces connus qui permettent de vérifier qu'un nombre n'est pas premier ne fournissent pas de factorisation.

Théorème d'Euler

La valeur $\varphi(n)$ de l'indicatrice d'Euler en n est l'ordre du groupe des éléments inversibles de l'anneau $\mathbb{Z}/n\mathbb{Z}$. Ceci permet de voir immédiatement, par le théorème d'Euler (conséquence du théorème de Lagrange), que si M est premier avec n, donc inversible (ce qui est le cas de « la plupart » des entiers naturels M strictement inférieurs à n)

$$M^{ed} \equiv M^{1+k\varphi(n)} \equiv M \pmod{n}$$

soit de justifier le chiffrement RSA (pour de tels M).

Il s'avère que quand n est un produit de nombres premiers distincts, l'égalité est vérifiée pour tout M[4] (la démonstration est essentiellement celle faite ci-dessus pour RSA, dans le cas particulier où n est un produit de deux nombres premiers).

Implémentation

Engendrer les clefs

Le couple de clefs demande de choisir deux nombres premiers de grande taille, de façon qu'il soit calculatoirement impossible de factoriser leur produit.

Pour déterminer un nombre premier de grande taille, on utilise un procédé qui fournit à la demande un entier impair aléatoire d'une taille suffisante, un test de primalité permet de déterminer s'il est ou non premier, et on s'arrête dès qu'un nombre premier est obtenu. Le théorème des nombres premiers assure que l'on trouve un nombre premier au bout d'un nombre raisonnable d'essais.

La méthode demande cependant un test de primalité très rapide. En pratique on utilise un test probabiliste, le test de primalité de Miller-Rabin[5] ou une variante[6]. Un tel test ne garantit pas exactement que le nombre soit premier, mais seulement une (très) forte probabilité qu'il le soit.

Chiffrer et déchiffrer

Le calcul de $M = c^d \mod n$ ne peut se faire en calculant d'abord c^d, puis le reste modulo n, car cela demanderait de manipuler des entiers beaucoup trop grands. Il existe des méthodes efficaces pour le calcul de l'exponentiation modulaire.

On peut conserver une forme différente de la clé privée pour permettre un déchiffrement plus rapide à l'aide du théorème des restes chinois.

Sécurité

Il faut distinguer les attaques par la force brute, qui consistent à retrouver p et q sur base de la connaissance de n uniquement, et les attaques sur base de la connaissance de n mais aussi de la manière dont p et q ont été générés, du logiciel de cryptographie utilisé, d'un ou plusieurs messages éventuellement interceptés etc.

La sécurité de l'algorithme RSA contre les attaques par la force brute repose sur deux conjectures :

1. « casser » RSA de cette manière nécessite la factorisation du nombre n en le produit initial des nombres p et q,
2. avec les algorithmes classiques, le temps que prend cette factorisation croît exponentiellement avec la longueur de la clé.

Il est possible que l'une des deux conjectures soit fausse, voire les deux. Jusqu'à présent, ce qui fait le succès du RSA est qu'il n'existe pas d'algorithme connu de la communauté scientifique pour réaliser une attaque force brute avec des ordinateurs classiques.

Le 12 décembre 2009, le plus grand nombre factorisé par ce moyen, en utilisant une méthode de calculs distribués, était long de 768 bits. Les clés RSA sont habituellement de longueur comprise entre 1 024 et 2 048 bits. Quelques experts croient possible que des clés de 1 024 bits seront cassées dans un proche avenir (bien que ce soit controversé [réf. nécessaire]), mais peu voient un moyen de casser de cette manière des clés de 4 096 bits dans un avenir prévisible [réf. nécessaire]. On peut néanmoins présumer que RSA reste sûr si la taille de la clé est suffisamment grande. On peut trouver la factorisation d'une clé de taille inférieure à 256 bits en quelques minutes sur un ordinateur individuel, en utilisant des logiciels librement disponibles[7]. Pour une taille allant jusqu'à 512 bits, et depuis 1999, il faut faire travailler conjointement plusieurs centaines d'ordinateurs. Par sûreté, il est couramment recommandé que la taille des clés RSA soit au moins de 2 048 bits.

Si une personne possède un moyen « rapide » de factoriser le nombre n, tous les algorithmes de chiffrement fondés sur ce principe seraient remis en cause ainsi que toutes les données chiffrées dans le passé à l'aide de ces algorithmes.

En 1994, un algorithme permettant de factoriser les nombres en un temps non exponentiel a été écrit pour les ordinateurs quantiques. Il s'agit de l'algorithme de Shor. Les applications des ordinateurs quantiques permettent théoriquement de casser le RSA par la force brute, ce qui a activé la recherche sur ce sujet ; mais actuellement ces ordinateurs génèrent des erreurs aléatoires qui les rendent inefficaces.

Les autres types d'attaques (voir Attaques ci-dessous), beaucoup plus efficaces, visent la manière dont les nombres premiers p et q ont été générés, comment e a été choisi, si l'on dispose de messages codés ou de toute autre information qui peut être utilisée. Une partie de la recherche sur ce sujet est publiée mais les techniques les plus récentes développées par les entreprises de cryptanalyse et les organismes de renseignement comme la NSA restent secrètes.

Il faut enfin noter que casser une clé par factorisation du nombre n ne nécessite pas d'attendre d'avoir un message chiffré à disposition. Cette opération peut débuter sur base de la connaissance de la clé publique seulement, qui est généralement libre d'accès. Dans ces conditions, si n est factorisé, la clé privée s'en déduit immédiatement. Les conséquences de cette observation sont également qu'un code peut être cassé avant même son utilisation.

Applications

Lorsque deux personnes souhaitent s'échanger des informations numériques de façon confidentielle, sur Internet

par exemple avec le commerce électronique, celles-ci doivent recourir à un mécanisme de chiffrement de ces données numériques. RSA étant un algorithme de chiffrement asymétrique, celui-ci hérite du domaine d'application de ces mécanismes de chiffrement. On citera :

- l'authentification des parties entrant en jeu dans l'échange d'informations chiffrées avec la notion de signature numérique ;
- le chiffrement des clés symétriques (nettement moins coûteuse en temps de calcul) utilisées lors du reste du processus d'échange d'informations numériques chiffrées.

Ce dernier est en fait intégré dans un mécanisme RSA. En effet, le problème des algorithmes symétriques est qu'il faut être sûr que la clé de chiffrement ne soit divulguée qu'aux personnes qui veulent partager un secret. RSA permet de communiquer cette clé symétrique de manière sûre. Pour ce faire, Alice va tout d'abord choisir une clé symétrique. Voulant échanger un secret avec Bob elle va lui transmettre cette clé symétrique en utilisant RSA. Elle va, pour cela, chiffrer la clé symétrique avec la clé publique (RSA) de Bob, ainsi elle sera sûre que seul Bob pourra déchiffrer cette clé symétrique. Une fois que Bob reçoit le message, il le déchiffre et peut alors utiliser la clé symétrique définie par Alice pour lui envoyer des messages chiffrés que seuls lui et Alice pourront alors déchiffrer.

Attaques

Plusieurs attaques ont été proposées pour casser le chiffrement RSA[8].

Attaque de Wiener

L'attaque de Wiener (1989) est exploitable si l'exposant secret d est inférieur à $\frac{1}{3}N^{\frac{1}{4}}$ [réf. nécessaire][9]. On peut retrouver dans ce cas l'exposant secret à l'aide du développement en fractions continues de $\frac{e}{N}$ [10].

Attaque de Håstad

L'attaque de Håstad, l'une des premières attaques découvertes (en 1985), repose sur la possibilité que l'exposant public e soit suffisamment petit. En interceptant le même message envoyé à plusieurs destinataires différents, il est possible de retrouver le message originel à l'aide du théorème des restes chinois[11].

Attaque par chronométrage (*timing attacks*)

Paul Kocher a décrit en 1995 une nouvelle attaque contre RSA : en supposant que l'attaquante Ève en connaisse suffisamment sur les documents d'Alice et soit capable de mesurer les temps de déchiffrement de plusieurs documents chiffrés, elle serait en mesure d'en déduire rapidement la clef de déchiffrement. Il en irait de même pour la signature.

En 2003, Boneh et Brumley ont montré une attaque plus pratique permettant de retrouver la factorisation RSA sur une connexion réseau (SSL) en s'appuyant sur les informations que laissent filtrer certaines optimisations appliquées au théorème des restes chinois. Une façon de contrecarrer ces attaques est d'assurer que l'opération de déchiffrement prend un temps constant. Cependant, cette approche peut en réduire significativement la performance. C'est pourquoi la plupart des implémentations (mises en œuvre) RSA utilisent plutôt une technique différente connue sous le nom d'« aveuglement cryptographique » (*blinding*).

L'aveuglement se sert des propriétés multiplicatives de RSA en insérant dans le calcul une valeur secrète aléatoire dont l'effet peut être annulé. Cette valeur étant différente à chaque chiffrement, le temps de déchiffrement n'est plus directement corrélé aux données à chiffrer, ce qui met en échec l'attaque par chronométrage : au lieu de calculer $c^d \pmod n$, Alice choisit d'abord une valeur aléatoire secrète r et calcule $(r^e c)^d \pmod n$. Le résultat de ce calcul est $rm \pmod n$ et donc l'effet de r peut être annulé en multipliant par son inverse.

Attaque à chiffrés choisis (*Adaptive chosen ciphertext attacks*)

Tel que décrit dans cet article, RSA est un chiffrement déterministe, et ne peut donc pas être sémantiquement sûr. Une contremesure est l'utilisation d'un schéma de remplissage probabiliste de manière telle qu'aucune valeur de message, une fois chiffré, ne donne un résultat peu sûr, par exemple si $C = M^e \leq N$, une attaque simple est le calcul direct de la racine e-ième de C, qui n'aura pas été réduite modulo N.

En 1998, Daniel Bleichenbacher décrit la première attaque pratique de type « chiffré choisi adaptable » contre des messages RSA[12]. En raison de défauts dans le schéma de remplissage PKCS #1 v1, il fut capable de récupérer des clefs de session SSL. À la suite de ce travail, les cryptographes recommandent maintenant l'utilisation de méthodes de remplissage formellement sûres [réf. nécessaire], telles que OAEP, et les laboratoires RSA ont publié de nouvelles versions de PKCS #1 résistantes à ces attaques[13].

Notes et références

1. (en) esp@cenet document view (http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=US4405829).
2. Menezes, van Oorschot et Vanstone 1996, p. 286 ; Schneier 1996, *Applied Cryptography*, p. 467.
3. Stinson 2006, *Cryptography : Theory and Practice*, p. 175.
4. Demazure 2008, proposition 2.19, p. 63.
5. Menezes, van Oorschot et Vanstone 1996, chap 4.
6. Voir les recommandations du NIST, décrites dans le standard FIPS 186-4 (http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf), p. 69-71.
7. Tutoriel sur le déchiffrement de clé privée (http://www.parlonssecurite.com/factoriser-cracke-une-cle-rsa/).
8. Voir (en) Dan Boneh, « Twenty Years of attacks on the RSA Cryptosystem », *Notices Amer. Math. Soc.*, vol. 46, nº 2, 1999, p. 203-213 (lire en ligne (http://crypto.stanford.edu/~dabo/abstracts/RSAattack-survey.html)).
9. « Techniques de cryptanalyse de RSA » (ftp://ftp.irisa.fr/local/caps/DEPOTS/BIBLIO2009/Grenier_Christophe.pdf), sur *http://irisa.fr*, 28 janvier 2009 (consulté le 21 janvier 2016)
10. http://www3.sympatico.ca/wienerfamily/Michael/MichaelPapers/ShortSecretExponents.pdf.
11. (en) Johan Håstad, « On using RSA with low exponent in a public key network », dans *Advances in Cryptology – CRYPTO'85, Lecture Notes in Computer Science*, vol. 218, Springer, p. 403-408.
12. Bleichenbacher 1998.
13. (en) RSA Laboratory, « Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography (https://tools.ietf.org/html/rfc3447) », Request for comments nº 3447.

Annexes

Bibliographie

- [Stinson 2003] Douglas Stinson, *Cryptographie, théorie et pratique*, Vuibert, 2003, 2e éd.
- [Schneier 2001] Bruce Schneier, *Cryptographie appliquée*, Vuibert, 2001, 2e éd. (ISBN 2-7117-8676-5)
- [Barthélemy *et al.* 2005] Pierre Barthélemy, Robert Rolland, Pascal Véron et Hermes Lavoisier, *Cryptographie, principes et mises en œuvre*, 2005 (ISBN 2-7462-1150-5)
- [Demazure 2008] Michel Demazure, *Cours d'algèbre. Primalité Divisibilité. Codes*, 2008 [détail de l'édition]
- [Bleichenbacher 1998] **(en)** Daniel Bleichenbacher, « Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1 », *Crypto*, 1998 (DOI 10.1007/BFb0055716 (http://dx.doi.org/10.1007%2FBFb0055716))
- [Rivest, Shamir et Adleman 1978] **(en)** Ronald Rivest, Adi Shamir et Leonard Adleman, « A method for obtaining digital signatures and public-key cryptosystems », *Communications of the ACM*, vol. 21, no 2, 1978, p. 120–126 (lire en ligne (http://people.csail.mit.edu/rivest/pubs/RSA78.pdf) **[PDF]**)
- [Menezes, van Oorschot et Vanstone 1996] **(en)** Alfred J. Menezes, Paul C. van Oorschot et Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996, 816 p. (ISBN 0-8493-8523-7, lire en ligne (http://cacr.uwaterloo.ca/hac/))

Articles connexes

- Authentification
- Signature numérique
- Digital Signature Algorithm (DSA)
- Compétition de factorisation RSA (Défi RSA)
- Nombre RSA

Liens externes

- RSA expliqué aux débutants (http://fr.openclassrooms.com/informatique/cours/la-cryptographie-asymetrique-rsa)
- « Cryptage RSA en C++ » (http://linor.fr/tutoriaux/tuto-423-cryptage-rsa-en.php)
