3. LE THÉORÈME DU RSA

Le protocole RSA se fonde sur le théorème suivant.
Soient p et q deux nombres premiers. On pose $n = pq$. Si le nombre e est un entier premier avec le nombre $(p-1)(q-1)$, alors il existe un entier d positif, tel que $ed = 1 \bmod (p-1)(q-1)$, et, pour cet entier d et un entier A quelconque, on a : $A^{ed} = A \pmod n$.
La démonstration n'utilise que des mathématiques parfaitement connues depuis le XVIIIe siècle. Il est amusant de voir que les mathématiques du RSA attendaient depuis deux siècles qu'on leur trouve une application : les mathématiques étaient en avance. Ce n'est pas toujours le cas, et, aujourd'hui, en matière de preuve de complexité, la cryptographie attend des résultats mathématiques qui ne viennent pas (concernant les liens entre RSA et factorisation, ou concernant la difficulté de la factorisation) : les mathématiques sont en retard!

(b) Destinataire rend publics n et e, qui constituent la clef publique. Il la publie dans un annuaire ou la communique à *Émetteur*, qui la lui demande. Il ne communique surtout pas p, q ou d. Les nombres p et q peuvent être oubliés, car ils ne serviront plus à personne. Le nombre d constitue la clef secrète de *Destinataire*.

(c) Émetteur, qui veut transmettre une information secrète à *Destinataire*, transforme son information en un nombre entier A, inférieur à n (ou en plusieurs si nécessaire), en utilisant des conventions connues de tous (provenant, par exemple, des codes numériques des caractères typographiques, ou en prenant $a = 01$, $b = 02$, etc.).

(d) Émetteur calcule, *(voir la figure 4)*, grâce à la méthode d'exponentiation rapide, $B = A^e \pmod n$, envoie B à *Destinataire* par un canal qui n'a pas besoin d'être protégé (par exemple, le courrier électronique).

(e) Destinataire, pour décoder B, calcule $B^d \pmod n$, ce qui lui redonne A, car, d'après le théorème du RSA, on a $B^d = A^{ed} = A \pmod n$.

UTILISATION ET SÉCURITÉ DU RSA

Avec les techniques utilisées aujourd'hui pour programmer les systèmes RSA, les spécialistes estiment que doubler la longueur des clefs multiplie le temps de création des clefs par 16 (croissance en L^4, L étant la longueur des clefs) et multiplie, en revanche, le temps des opérations de codage, décodage, signature et authentification seulement par 4 (croissance en L^2). En se fondant sur l'hypothèse que casser le RSA demanderait des algorithmes non polynomiaux, c'est-à-dire plus longs, certains déduisent que, plus les machines seront puissantes, plus on pourra utiliser facilement de longues clefs, et que donc plus l'écart entre la puissance disponible et celle nécessaire pour casser le RSA sera grand. Autrement dit, plus le temps passe, plus RSA serait robuste et sûr. Nous verrons plus loin que cette présentation des choses est très optimiste et loin d'être mathématiquement établie.

Sur le plan pratique, les experts recommandent aujourd'hui d'utiliser des nombres n de 768 bits (232 chiffres décimaux) pour mettre en œuvre le RSA dans le cas de données pas trop importantes, mais ils conseillent 1 024 bits (309 chiffres décimaux) pour un usage commercial et 2 048 bits (617 chiffres décimaux) pour avoir une garantie se prolongeant sur une longue période de temps. Les clefs de 512 bits (155 chiffres décimaux), encore très utilisées, ne devraient plus l'être, car on a réussi, en août 1999, à factoriser un nombre n (produit de deux grands nombres premiers) de 512 bits.

La confiance dans la sécurité du RSA n'est pas due à la démonstration théorique que ce système est sûr, car une telle démonstration n'existe pas (nous allons y revenir). La confiance affichée provient de l'échec, répété depuis plus de 20 ans, de toutes les tentatives entreprises pour casser ce système, tentatives qui n'ont conduit qu'à la formulation de quelques recommandations pour le choix des paramètres p, q, e, d. D'autres systèmes concurrents du RSA sont peut-être aussi bons ou même meilleurs, mais aucun ne bénéficie de la validation par l'usage et la robustesse aux attaques dont le RSA peut se prévaloir. Le fait d'avoir été l'un des premiers lui a donné un avantage qui, puisqu'il résiste aux tentatives d'effraction, le maintient en tête.

Sur le plan théorique, la situation est décevante et le restera certainement longtemps. Celui qui sait factoriser $n = pq$ retrouve ensuite facilement d. Inversement, les mathématiques montrent que celui qui connaît n, e et d peut trouver rapidement p et q. La robustesse du RSA apparaît donc liée à la difficulté de la factorisation. Malheureusement, il n'est pas exclu que quelqu'un, sans réussir à obtenir d ni p ni q, puisse décrypter un message : autrement dit, il n'a pas été prouvé que la difficulté du RSA est équivalente à celle de la factorisation. Deux attaques théoriques du RSA sont donc envisageables. Celle passant par la factorisation : quiconque sait factoriser les nombres de la taille de $n = pq$ sait lire comme à livre ouvert tous les messages cryptés par le RSA. Celle contournant la factorisation, dont personne n'a su établir qu'elle était impossible et pour laquelle, au contraire, on a proposé récemment des arguments indiquant qu'elle devait être sérieusement crainte. Dan Boneh et Ramarathnam Venkatesan ont en effet établi en 1998 que casser le RSA, lorsqu'il est utilisé avec des exposants e trop petits, n'est pas équivalent à factoriser n.

TENTATIVES POUR BRISER LE RSA

D'autres consignes et mises en garde sont d'ailleurs formulées à destination des utilisateurs du RSA. Certains nombres premiers p et q doivent être évités : on conseille de prendre p et q de taille proche, et d'éviter que les nombres $p + 1$, $p - 1$, $q + 1$, $q - 1$ soient faciles à factoriser (car alors n risquerait de l'être aussi par l'utilisation d'algorithmes spécialisés de factorisation qui savent tirer parti des factorisations de $p + 1$, $p - 1$, $q + 1$, $q - 1$). Un conseil évident, mais à ne pas oublier, est que le nombre n ne doit pas être choisi par plus d'une entité : sinon, l'utilisation des diverses clefs publiques des utilisateurs du même nombre n risquerait de donner accès aux facteurs de n.

Plus subtil, mais très important, il ne faut jamais accepter de signer un message en apparence aléatoire soumis par un inconnu. En effet, en choisissant bien ce message, l'inconnu peut en déduire la signature d'un message qu'il aura choisi lui-même au préalable, et vous vous retrouveriez donc à devoir assumer une signature que vous n'avez pas donnée.

L'utilisation d'exposants e trop petits doit être évitée.

Il faut compléter les textes, avant de les coder, par des bouts choisis aléatoirement et surtout pas par de longues séries de blancs ou de zéros.

Une attaque astucieuse proposée par P. Kocher consiste à mesurer minutieusement le temps de calcul (ou la consommation électrique) demandé pour le codage, ce qui permet, si l'on dispose de suffisamment de données, de retrouver les clefs utilisées. Appliquée aux cartes à puce, cette méthode se révélerait catastrophique. Heureusement, se prémunir contre cette attaque est assez facile : il suffit de s'arranger pour que le temps de calcul (ou la consommation électrique)

apparent soit faussé, en opérant des calculs supplémentaires inutiles une fois le codage réalisé, avant d'envoyer l'information que le codage est terminé.

ON NE S'IMPROVISE PAS CRYPTOGRAPHE

De nombreuses erreurs dans la gestion imprudente ou maladroite des clefs sont aussi à l'origine de décryptages inopinés des messages considérés inviolables, car obtenus avec RSA. Une prudence à tous les niveaux est indispensable pour en arriver à des systèmes fiables. Dans un article paru en 1999, Dan Boneh, qui est le grand spécialiste des attaques du RSA proposait cette conclusion rassurante : «En ce moment, il apparaît que l'on peut avoir confiance dans la sécurité que procure le RSA au monde numérique.» Bruce Schneier, un autre spécialiste international de la cryptographie très préoccupé des réalisations opérationnelles, insiste sur le fait que 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.

L'idée défendue que le noyau mathématique du RSA est inviolable est seulement une conclusion expérimentale tirée de l'échec des tentatives connues d'attaques du RSA (lorsqu'on l'utilise en respectant les consignes formulées par les spécialistes… dont la liste s'allonge chaque année). Cette conclusion n'interdit pas que le risque théorique lié au RSA puisse être jugé grand, puisqu'on n'a ni réussi à montrer que la factorisation est difficile (aucun espoir n'existe à court terme qu'on y arrive), ni même réussi à montrer que le RSA est aussi difficile que la factorisation.

À l'évidence, si une agence de renseignements a découvert quelque chose concernant la factorisation ou s'appliquant directement au RSA sans passer par la factorisation, elle ne rendra pas publique sa découverte et, au contraire, fera tout pour la maintenir secrète le plus longtemps possible. Dire cela n'est pas faire preuve d'une méfiance maladive, mais du simple bon sens. La tranquille assurance exprimée par les spécialistes à propos de la sécurité du RSA apparaît incompréhensible, et parfois même assez louche.

Dix fois, dans l'histoire des services d'espionnage, un système de codage a été considéré cent pour cent sûr et inviolable par ceux qui l'utilisaient, alors qu'au même moment un service ennemi ayant percé le secret du système lisait tranquillement tous les messages qui tombaient entre ses mains. Cela s'est produit pendant la Première Guerre mondiale : les services français décodaient chaque nuit les messages destinés aux U-boot tapis au fond de la Méditerranée, ce qui permettait de modifier avant l'aube les programmes de navigation des bateaux menacés. À nouveau, pendant la Seconde Guerre mondiale, cela s'est produit avec la fameuse affaire de la machine *Enigma*, où s'illustra Alan Turing. Signalons aussi l'extraordinaire secret qui a entouré cette aventure impliquant les centaines de personnes qui ont travaillé au Bletchley Park, près de Londres, qui toutes sont restées muettes pendant trente ans : le sens patriotique est une motivation suffisante pour faire taire même le mathématicien le plus soucieux du partage de ses découvertes. Il me semble que, si un mathématicien canadien (même s'il est universitaire) découvre quelque chose d'important sur la factorisation, il le gardera pour les militaires canadiens, de même qu'un mathématicien français réservera ses théorèmes cryptographiques cruciaux pour les militaires français, etc.

Pourquoi donc, alors que des moyens considérables sont consacrés à la cryptanalyse – et donc à toutes les mathématiques et techniques qui y sont liées – par les agences spécialisées de renseignements puissantes et fortunées, ne pas envisager sérieusement, qu'ignorée des circuits de la recherche universitaire (et provenant peut-être d'un universitaire patriote), une découverte fondamentale faisant tomber le RSA ait été faite et reste cachée?

Même si l'on n'est pas paranoïaque, en l'absence de résultats mathématiques établissant la sécurité du RSA, il est sage de ne pas trop lui faire confiance. Réalise-t-on, quand on utilise la méthode de cryptage à deux étages évoquée plus haut (RSA pour communiquer une clef, puis algorithme symétrique), que si un seul des deux étages est percé c'est tout le système qui coule? Comment ne pas penser à de la désinformation quand on lit, en contradiction avec tout bon sens, que le système à deux étages est d'autant plus solide qu'il utilise deux méthodes cryptographiques !

A. M. Odlyzko a publié un bilan sur le cryptosystème à double clef de Merkle et Hellman fondé sur le problème du sac à dos (un problème combinatoire de rangement). Grâce à certaines propriétés mathématiques le reliant à d'autres problèmes assez bien compris, ce cryptosystème fut considéré un temps comme plus prometteur que le RSA. Malheureusement, il fut cassé, et les variantes qui ont été proposées aujourd'hui de ce système se sont révélées pour l'essentiel aussi fragiles que la version originale, tant et si bien que du système le plus prometteur d'il y a 15 ans il ne reste rien aujourd'hui. N'en sera-t-il pas de même pour le RSA dans 15 ans, et tout ce qui

4. LE CALCUL DE L'EXPONENTIATION RAPIDE

Dans la mise en œuvre aussi bien des algorithmes probabilistes pour engendrer des nombres premiers que pour l'implémentation du RSA, on a besoin de calculer rapidement A^e mod n. Voici une méthode (dont le principe général est emprunté à la multiplication égyptienne) qui permet ce calcul.

Calcul de A^e mod n

- partir du triplet $(A, e, 1) = (P, J, R)$ puis :
- si J est pair, passer à $(P^2 \bmod n, J/2, R \bmod n)$
- si J est impair, passer à $(P^2 \bmod n, (J-1)/2, R \times P \bmod n)$
- s'arrêter si $J = 1$, le résultat est $R = P \bmod n$

Exemple :

$A^{13} \bmod n \longrightarrow (A,13,1) \longrightarrow (A^2 \bmod n, 6, A) \longrightarrow (A^4 \bmod n, 3, A) \longrightarrow (A^8 \bmod n, 1, A^5) \longrightarrow R = A^{13} \bmod n.$ Avec $n = 5$ et $A = 2$:

$213 \bmod 5 \longrightarrow (2,13,1) \longrightarrow (4, 6, 2) \longrightarrow (1, 3, 2) \longrightarrow (1, 1, 2) \longrightarrow R = 2$

5. EXEMPLE NUMÉRIQUE SIMPLIFIÉ DE CODAGE RSA

$p = 47$, $q = 71$, $n = pq = 3337$, $(p-1)(q-1) = 3220$, $e = 79$, $d = 1019$

On vérifie que $ed = 80501 = 25 \times 3220 + 1 = 1 \bmod 3220$

Prenons un message préalablement transformé en un nombre :

$m = 6882326879666683$

On découpe m : $m_1 = 688$ $m_2 = 232$ $m_3 = 687$ $m_4 = 966$ $m_5 = 668$ $m_6 = 3$

Le premier bloc est chiffré par le nombre $c_1 = 688^{79} \bmod 3337 = 1570$;

de même : $c_2 = 2756$ $c_3 = 2091$ $c_4 = 2276$ $c_5 = 2423$ $c_6 = 158$.

Le déchiffrement du message se fait en calculant :

$c_1 = 1570^{1019} \bmod 3337 = 688$, etc.