La cryptologie moderne

Anne Canteaut
INRIA
Projet CODES
BP 105
78153 Le Chesnay Cedex
Anne.Canteaut@inria.fr
http://www-rocq.inria.fr/~canteaut/

Françoise Lévy-dit-Véhel
École Nationale Supérieure
des Techniques Avancées
32, boulevard Victor
75739 Paris Cedex 15
levy@ensta.fr
http://www.ensta.fr/uer/uma/crypto/fldv.html

La cryptographie est une discipline ancienne. Déjà dans l'antiquité, les Grecs avaient inventé des méthodes pour chiffrer les messages. L'une d'entre elles, datant du VIème siècle avant J.C., consistait à enrouler une bande de papier autour d'un cylindre, puis à écrire le message sur la bande. Une fois déroulé, le papier était envoyé au destinataire qui, dès lors qu'il possédait le diamètre du cylindre, pouvait déchiffrer le message.

Pendant de nombreuses années, la cryptographie était exclusivement réservée au domaine militaire et diplomatique. La littérature sur le sujet était donc très peu abondante. La première publication fondamentale dans ce domaine a été l'article de Claude Shannon de 1949 - *"The communication theory of secrecy systems"* [Sha]- dans lequel il jette les bases mathématiques d'un système de communication chiffrée, à partir de la définition d'un nouveau modèle : la théorie de l'information. Une contribution importante a ensuite été celle de Feistel, avec la publication, au début des années 1970, de ses travaux sur les schémas de chiffrement itératifs par blocs [Fei1, Fei2], qui ont conduit en 1977 à la proposition de l'algorithme DES comme standard de chiffrement à clef secrète pour des applications non classifiées. L'accroissement de la puissance des ordinateurs ayant remis en cause la sécurité du DES, il a été remplacé en octobre 2000 par un nouveau standard appelé AES. Cet algorithme est l'aboutissement de recherches récentes notamment dans le domaine de la cryptanalyse.

Mais l'avancée majeure en cryptographie a incontestablement été la publication, en 1976, de l'article *"New directions in cryptography"* [Dif], de Whitfield Diffie et Martin Hellman. Cet article introduit le concept révolutionnaire de cryptographie *à clef publique*. Même si les auteurs ne donnent pas de réalisation pratique d'un système à clef publique, les propriétés d'un tel système sont clairement énoncées. En outre, ils présentent un protocole par lequel deux entités peuvent convenir d'une clef secrète à partir de la connaissance préalable de seules données publiques. La première réalisation d'un système à clef publique est due à Ronald Rivest, Adi Shamir et Leonard Adleman, en 1978 : c'est le RSA [Riv]. Depuis lors, la littérature sur ce sujet n'a cessé de se développer.

Plus récemment, pour faire face aux nouvelles menaces induites par le développement des réseaux et la numérisation massive des documents, la cryptographie a dû offrir de nouvelles fonctionnalités : garantie de l'authenticité des messages (provenance et contenu), réalisée par des algorithmes de signature numérique, certification de l'identité d'une personne (techniques d'identification), en sont les principaux exemples.

Cet article introductif présente d'abord les deux grandes catégories de procédés cryptographiques les plus utilisées : les *algorithmes de chiffrement*, qui servent à protéger la confidentialité des données, et les *algorithmes de signature* qui, comme les signatures manuscrites, garantissent la provenance et l'intégrité des messages. Il détaille ensuite quelques aspects de

l'implantation pratique de tels procédés. Il est à noter qu'il est indispensable, pour une application précise, de répertorier les fonctionnalités souhaitées avant de rechercher une solution cryptographique adéquate.

1 Le chiffrement

Un algorithme de chiffrement transforme un message, appelé texte clair, en un texte chiffré qui ne sera lisible que par son destinataire légitime. Cette transformation est effectuée par une fonction de chiffrement paramétrée par une clef de chiffrement. Un interlocuteur privilégié peut alors déchiffrer le message en utilisant la fonction de déchiffrement s'il connaît la clef de déchiffrement correspondant. Un tel système n'est sûr que s'il est impossible à un intrus de déduire le texte clair du message chiffré, et *a fortiori* de retrouver la clef de déchiffrement.

Cette formalisation a maintenant un peu plus d'un siècle. A cette époque, les cryptographes ont pris conscience qu'il n'était pas réaliste de faire reposer la sécurité d'un système de chiffrement sur l'hypothèse qu'un attaquant n'a pas connaissance de la méthode utilisée. La publication récente sur Internet des spécifications d'algorithmes propriétaires, tel celui utilisé dans le système GSM, nous a encore montré qu'il est impossible de conserver un algorithme secret à long terme. En conséquence, la sécurité d'un algorithme de chiffrement doit uniquement reposer sur le secret de la clef de déchiffrement. Par ailleurs, le fait de rendre publiques les méthodes de chiffrement et de déchiffrement offre une certaine garantie sur la sécurité d'un système, dans la mesure où tout nouvel algorithme cryptographique est immédiatement confronté à la sagacité de la communauté scientifique.

On distingue deux grands types d'algorithmes de chiffrement, les algorithmes à clef secrète et les algorithmes à clef publique. Chacune de ces deux classes possède ses propres avantages et inconvénients. Les systèmes à clef secrète nécessitent le partage d'un secret entre les interlocuteurs. La découverte en 1976 des systèmes à clef publique a permis de s'affranchir de cette contrainte, mais elle n'a pas pour autant apporté de solution parfaite, dans la mesure où tous les algorithmes de chiffrement à clef publique, de par leur lenteur, ne permettent pas le chiffrement en ligne. Dans la plupart des applications actuelles, la meilleure solution consiste à utiliser un système hybride, qui combine les deux types d'algorithmes.

1.1 Le chiffrement à clef secrète

Les *algorithmes de chiffrement à clef secrète* (ou *symétriques* ou encore *conventionnels*) sont ceux pour lesquels émetteur et destinataire partagent une même clef secrète — autrement dit, les clefs de chiffrement et de déchiffrement sont identiques. L'emploi d'un algorithme à clef secrète lors d'une communication nécessite donc l'échange préalable d'un secret entre les deux protagonistes à travers un canal sécurisé ou au moyen d'autres techniques cryptographiques.

Un paramètre essentiel pour la sécurité d'un système à clef secrète est la taille de l'espace des clefs. En effet, il est toujours possible de mener sur un algorithme de chiffrement une attaque dite *exhaustive* pour retrouver la clef. Cette attaque consiste simplement à énumérer toutes les clefs possibles du système et à essayer chacune d'entre elles pour décrypter un message chiffré. Si l'espace des clefs correspond à l'ensemble des mots de k bits, le nombre moyen d'appels à la fonction de déchiffrement requis dans une attaque exhaustive est égal à 2^{k-1}. Une telle attaque devient donc hors de portée dès que l'espace des clefs est suffisamment grand. Au vu de la puissance actuelle des ordinateurs, on considère qu'une clef secrète doit comporter au minimum 64 bits (ce qui nécessite en moyenne 2^{63} essais pour une attaque

exhaustive). Notons que cette limite évolue avec la technologie. Pour donner un ordre de grandeur, une attaque exhaustive du système de chiffrement DES, qui utilise une clef secrète de 56 bits, a été réalisée en janvier 1998 en 39 jours sur 10 000 Pentium en parallèle, puis en 56 heures en juillet 1998 à l'aide d'une machine dédiée comportant 1500 composants DES [1]. Le temps de calcul nécessaire à une attaque exhaustive est évidemment exponentiel en la taille de la clef secrète. Il est 2^{64} fois, c'est-à-dire 18446744073709551616 fois plus dur de casser un système possédant une clef de 128 bits que de casser un système avec une clef de 64 bits (ce qui est déjà très difficile).

Il existe d'autres types d'attaques sur les systèmes de chiffrement à clef secrète. La plupart consistent à exploiter certaines structures particulières de l'algorithme ou certains biais statistiques dans la distribution des couples clairs-chiffrés. Les plus connues sont la cryptanalyse différentielle, inventée par les Israëliens Biham et Shamir en 1991, et la cryptanalyse linéaire proposée par le Japonais Matsui en 1993. On considère généralement qu'un chiffrement à clef secrète présente une bonne sécurité s'il n'existe pas d'attaque dont la complexité soit inférieure à celle de la recherche exhaustive. A l'heure actuelle, la sécurité des systèmes à clef secrète repose uniquement sur la constatation empirique qu'ils sont difficiles à cryptanalyser. On peut démontrer qu'un algorithme de chiffrement résiste aux attaques classiques, mais on ne peut pas exclure l'apparition de nouvelles attaques efficaces.

Seules les techniques dites *de chiffrement par blocs* sont envisagées ici. Un système de chiffrement est dit par blocs s'il divise le texte clair en blocs de taille fixe et chiffre un bloc à la fois. La taille des blocs est généralement de 64 ou de 128 bits.

DES Jusqu'à très récemment, le système de chiffrement à clef secrète le plus célèbre et le plus utilisé était le DES (Data Encryption Standard). Il a été adopté comme standard américain en 1977 (standard FIPS 46 [2]) pour les communications commerciales, puis par l'ANSI en 1991. Le DES opère sur des blocs de 64 bits et utilise une clef secrète de 56 bits. Il est donc désormais vulnérable aux attaques exhaustives.

C'est pourquoi la plupart des applications l'utilisent maintenant sous la forme d'un triple DES à deux clefs, constitué de trois chiffrements DES successifs avec deux clefs secrètes. Cette technique permet de doubler la taille de la clef secrète (112 bits). Plus précisément, pour chiffrer avec le triple DES, on effectue d'abord un chiffrement DES paramétré par une première clef de 56 bits, puis un déchiffrement DES paramétré par une seconde clef, et à nouveau un chiffrement DES avec la première clef. Seules deux clefs sont utilisées dans la mesure où l'emploi de trois clefs secrètes différentes ne permet pas d'accroître la sécurité de l'algorithme. Le triple DES à deux clefs a notamment été adopté dans les standards ANSI X9.17 et ISO 8732. Il est extrêmement utilisé pour les applications bancaires.

D'autres applications, comme le système PGP, lui préfère l'algorithme de chiffrement IDEA (International Data Encryption Algorithm) conçu par Lai et Massey en 1992. Cet algorithme opère sur des blocs de 64 bits, mais utilise une clef de 128 bits.

AES L'AES (Advanced Encryption Standard) est le nouveau standard de chiffrement à clef secrète. Il a été choisi en octobre 2000 parmi les 15 systèmes proposés en réponse à l'appel d'offre lancé par le NIST (National Institute of Standards and Technology). Cet algorithme, initialement appelé RIJNDAEL, a été conçu par deux cryptographes belges, V. Rijmen et

1. `http://www.eff.org/descracker.html`
2. `http://csrc.nist.gov/cryptval/des.htm`

J. Daemen. Il opère sur des blocs de message de 128 bits et est disponible pour trois tailles de clef différentes : 128, 192 et 256 bits. Les spécifications de ces trois versions ainsi que plusieurs implémentations sont disponibles sur la page Web du NIST [3].

Comme pour la plupart des algorithmes par blocs, le processus de chiffrement de l'AES consiste à itérer une permutation paramétrée par une valeur secrète, appelée sous-clef, qui change à chaque itération. Les différentes sous-clefs sont dérivées de la clef secrète par un algorithme de cadencement de clef. Pour une clef de 128 bits, l'AES effectue 10 itérations de la fonction décrite à la figure 1, chacune des sous-clefs comportant également 128 bits. La première itération est précédée d'un ou exclusif bit-à-bit entre le message clair et la sous-clef numéro 0 ; de même, la dernière itération est légèrement différente des itérations précédentes.

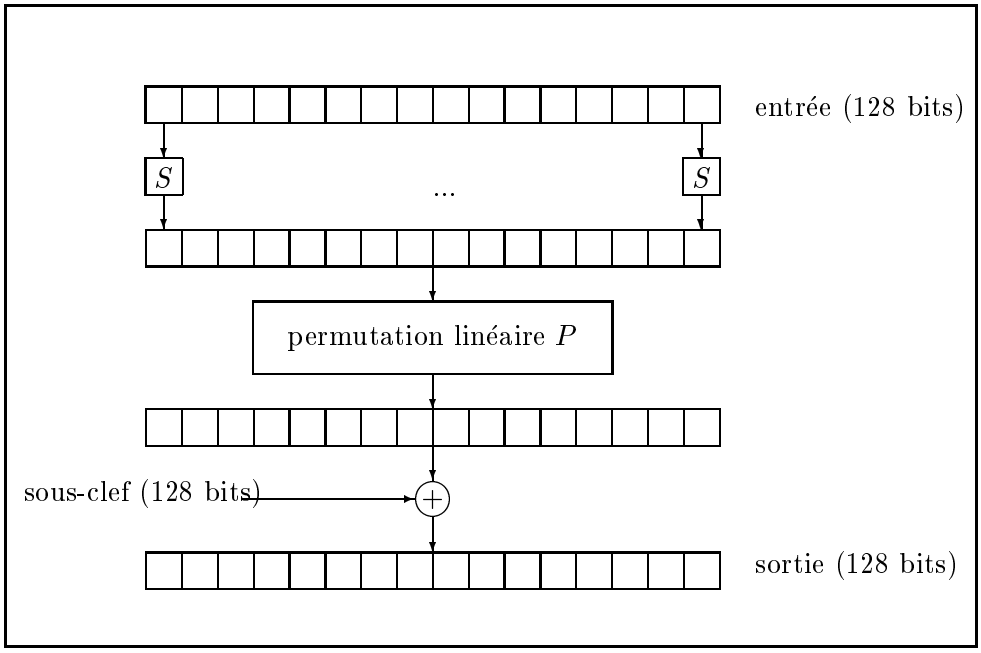

FIG. 1 – *Une itération de l'AES*

La fonction itérée se décompose elle-même en trois étapes, conformément aux principes fondamentaux de confusion et de diffusion énoncés par Shannon. La première étape, dite de confusion, consiste à appliquer à chacun des 16 octets de l'entrée une même permutation S. Cette fonction correspond à la fonction inverse dans le corps fini à 2^8 éléments (dans la pratique, elle est mise en table) ; elle assure la résistance de l'algorithme aux attaques classiques (cryptanalyse différentielle, cryptanalyse linéaire ...). Ensuite, lors de la phase de diffusion, on permute les bits du mot obtenu suivant une fonction P qui est également composée d'opérations simples sur le corps à 2^8 éléments. Enfin, on effectue un ou exclusif bit-à-bit entre le résultat et la sous-clef de l'itération.

Les sous-clefs de 128 bits, numérotées de 0 à 10, sont dérivées de la clef secrète de la manière suivante : la sous-clef numéro 0 correspond à la clef secrète ; ensuite, la sous-clef numéro i (utilisée à la $i$ème itération) est obtenue à partir de la sous-clef numéro $(i-1)$ grâce à l'algorithme décrit à la figure 2.

3. `http://csrc.nist.gov/encryption/aes/rijndael/`

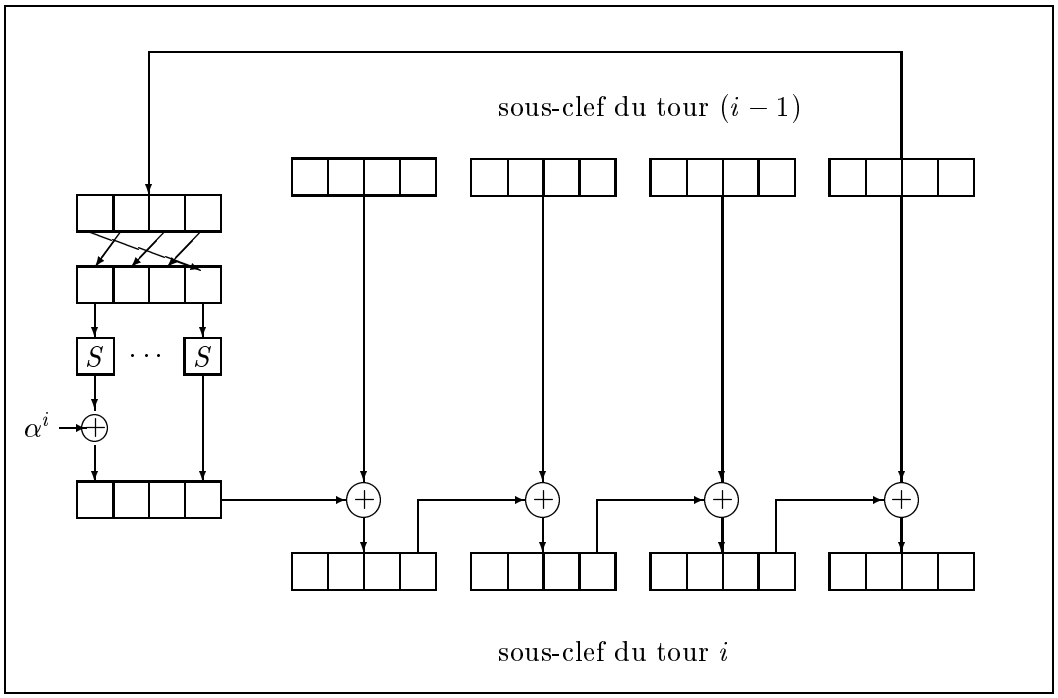

FIG. 2 – *Algorithme de cadencement de clef de l'AES*

On permute les quatre derniers octets de la clef numéro $(i-1)$, puis on leur applique la fonction S. Après avoir ajouté une constante (dépendant de i) au premier octet, on effectue un ou exclusif bit-à-bit entre les quatre octets ainsi obtenus et les quatre premiers octets de la sous-clef précédente. Les trois autres blocs de quatre octets de la clef numéro i sont ensuite simplement le résultat d'un ou exclusif entre le bloc correspondant de la sous-clef $(i-1)$ et le bloc précédent de la sous-clef i.

Le fait que l'AES soit uniquement composé d'opérations simples sur les octets le rend extrêmement rapide, à la fois pour les processeurs 8 bits utilisés dans les cartes à puce et pour les implémentations logicielles. Il atteint par exemple un débit de chiffrement de 70 Mbits/s pour une implémentation en C++ sur un Pentium à 200 MHz[4].

1.2 Le chiffrement à clef publique

La *cryptographie à clef publique* (ou *asymétrique*) évite le partage d'un secret entre les deux interlocuteurs. Dans un système de chiffrement à clef publique, chaque utilisateur dispose d'un couple de clefs, une *clef publique* qu'il met en général à disposition de tous dans un annuaire, et une *clef secrète* connue de lui seul. Pour envoyer un message confidentiel à Bob, Alice chiffre donc le message clair à l'aide de la clef publique de Bob. Ce dernier, à l'aide de la clef secrète correspondante, est seul en mesure de déchiffrer le message reçu.

Si l'on devait comparer les deux types de chiffrement à des moyens physiques d'échange de messages confidentiels, un système à clef secrète serait un coffre-fort, alors qu'un système à clef publique correspondrait à une boîte aux lettres. Imaginons qu'Alice veuille communiquer un document à Bob en le déposant dans un coffre-fort. Dans ce cas, Alice et Bob doivent partager

4. `http://fp.gladman.plus.com/cryptography_technology/rijndael/`

un secret, la combinaison du coffre, qui permet à la fois à Alice de déposer les documents et à Bob de les récupérer. Une tierce personne ne peut pas se servir du même coffre pour communiquer avec Bob, sauf si elle connaît elle aussi la combinaison secrète. Mais, cette dernière solution lui fournit également la possibilité de lire tous les documents placés dans le coffre, même ceux qui ne lui sont pas destinés. Supposons maintenant que Bob demande à ses interlocuteurs de lui transmettre les documents confidentiels en les déposant dans sa boîte aux lettres (qui ferme à clef). Toute personne connaissant l'adresse de Bob (qui est publique) peut lui déposer des messages. Par contre, seul Bob possède la clef qui lui permet d'ouvrir la boîte aux lettres et de lire les messages qui lui sont destinés.

La notion essentielle sur laquelle repose le chiffrement à clef publique est celle de *fonction à sens unique avec trappe*. Une fonction est appelée *à sens unique* si elle est facile à calculer mais impossible à inverser. Impossible signifie ici infaisable en un temps réaliste avec une puissance de calcul raisonnable. On considère comme étant impossible, par exemple, un calcul qui, réparti sur un milliard de processeurs en parallèle, nécessiterait un milliard d'années. Une telle fonction est dite *à trappe* si le calcul de l'inverse devient facile dès que l'on possède une information supplémentaire (la trappe).

Il est très simple de construire un système de chiffrement à clef publique à partir d'une fonction à sens unique avec trappe. La procédure de chiffrement consiste simplement à appliquer la fonction au message clair. La fonction étant à sens unique, il est très difficile de l'inverser, c'est-à-dire de déterminer le message clair à partir du message chiffré, sauf si on connaît la trappe, qui correspond à la clef secrète du destinataire. Toute la difficulté réside donc dans la recherche de ces fonctions très particulières. Leur construction s'appuie généralement sur des problèmes mathématiques réputés difficiles; le plus célèbre est celui de la factorisation de grands nombres entiers, qui est à la base du système RSA.

RSA C'est le système à clef publique le plus utilisé. RSA n'est pas à proprement parler un standard mais son utilisation est décrite et recommandée dans un grand nombre de standards officiels, en particulier pour les applications bancaires, par exemple le standard français ETEBAC 5, le standard américain ANSI X9.31.

Son fonctionnement repose sur des résultats classiques d'arithmétique. Dans toute la suite, pour deux entiers a et n, la notation $a \bmod n$ désigne le reste de la division euclidienne de a par n. Considérons un entier n formé par le produit de deux nombres premiers p et q. D'après le théorème d'Euler, si a est un entier tel que $a \bmod (p-1)(q-1) = 1$, alors, pour tout entier non nul $x < n$, on a :

$$x^a \bmod n = x \ .$$

Le principe de RSA est alors le suivant : la clef publique d'un utilisateur est formée d'un nombre n produit de deux nombres premiers p et q, et d'un entier e premier avec $(p-1)(q-1)$. Les valeurs de n et e sont publiées dans un annuaire.

La clef secrète correspondant est un entier d qui vérifie $ed \bmod (p-1)(q-1) = 1$. Il est très facile de trouver un tel nombre d à partir de e, p et q. En effet, par hypothèse, l'entier e est premier avec $(p-1)(q-1)$. D'après le théorème de Bezout, il existe donc deux entiers A et B non nuls tels que

$$A(p-1)(q-1) + Be = \mathrm{pgcd}((p-1)(q-1), e) = 1 \ .$$

La clef secrète d est donc l'entier positif correspondant au reste de B modulo $(p-1)(q-1)$.

Dans RSA, les blocs de message sont représentés par des entiers compris entre 0 et $n-1$. Pour envoyer un message m à Bob, Alice va donc chercher la clé publique de Bob et elle calcule le message chiffré c correspondant par : $c = m^e \bmod n$.

Lorsqu'il reçoit le chiffré c, Bob retrouve le texte clair en calculant $c^d \bmod n = m$. En effet, par définition, e et d sont tels que $ed \equiv 1 \bmod (p-1)(q-1)$. On a donc :

$$c^d \bmod n = (m^e)^d \bmod n = m^{ed} \bmod n = m.$$

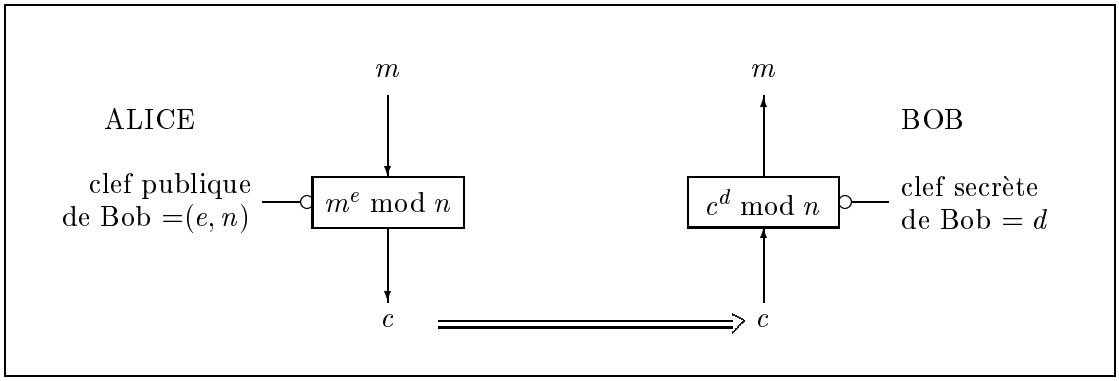

FIG. 3 – *Le chiffrement à clef publique RSA*

Attaquer le système RSA consiste donc à retrouver le texte clair m à partir de la connaissance du chiffré $c = m^e \bmod n$ et de la clef publique (e, n). Aucun algorithme efficace n'est connu à ce jour pour résoudre ce problème. La seule attaque générale connue pour décrypter RSA consiste à retrouver la clef secrète d à partir des valeurs publiques (e, n). On peut démontrer que résoudre ce problème est équivalent à factoriser l'entier n. Il n'existe actuellement pas d'algorithme de factorisation rapide. Le plus grand nombre ordinaire factorisé à ce jour est un nombre de 512 bits (155 chiffres décimaux). Ce record a été établi en 1999 par la collaboration de onze équipes scientifiques [5]. La factorisation a nécessité deux mois et demi de calculs répartis sur 300 ordinateurs et 224 heures sur un Cray-C916. Ces résultats récents montrent que, pour être à l'abri des algorithmes de factorisation, il est impératif d'utiliser pour RSA des entiers p et q qui soient tels que leur produit comporte au moins 768 bits, et on utilise en général des premiers dont le produit est de 1024 bits.

Cryptosystèmes sur courbes elliptiques Outre la factorisation entière, un autre problème largement utilisé en cryptographie est l'extraction de logarithmes discrets, qui peut s'énoncer comme suit : étant donné un groupe fini G, noté multiplicativement, un générateur g de G, et un élément β dans G, trouver x dans [6] $\{0, \ldots, |G|-1\}$, tel que $\beta = g^x$. Ce problème est à l'origine du protocole d'échange de clefs de Diffie-Hellman (78), du cryptosystème d'El Gamal (84), et de plusieurs schémas de signature (cf. section 2). Depuis une quinzaine d'années, son adaptation à d'autres groupes a été envisagée, et a donné lieu à ce que l'on appelle les cryptosystèmes sur courbes elliptiques. L'idée, dûe à N. Koblitz et V. Miller en 85, est d'utiliser comme groupe G le groupe additif des points d'une courbe elliptique sur un corps

5. `http://ultralix.polytechnique.fr/Labo/Francois.Morain/rsa155.html`
6. $|G|$ désigne ici le cardinal de G.

fini **F**. Sans préciser davantage les notions en jeu, citons les deux principales raisons motivant une telle approche :

- On peut générer de cette manière un grand nombre de groupes sans changer le corps **F** ; ainsi, on peut concevoir un processeur arithmétique optimisé pour calculer spécifiquement dans **F**, que l'on pourra utiliser pour mettre en œuvre diverses instances du même cryptosystème (ou de différents cryptosystèmes basés sur le même corps).
- Il n'est pas connu d'algorithme sous-exponentiel qui résolve le problème du logarithme discret dans ce contexte[7], contrairement au problème du logarithme discret dans le groupe multiplicatif G d'un corps fini, dont le meilleur algorithme de résolution a pour complexité $\exp(O((\lg q)^{1/3}(\lg\lg q)^{2/3}))$, q étant la taille du corps.

Cette dernière observation a pour conséquence importante de permettre l'utilisation de clefs de taille significativement moindre, comparée à celles nécessaires aux cryptosystèmes basés sur le logarithme discret dans les groupes classiques, ou celles de RSA[8]. Par exemple, 170 bits de clefs suffisent pour assurer le même niveau de sécurité qu'un chiffrement RSA à 1024 bits. La mémoire nécessaire au stockage de ces clefs, ainsi que la taille des données en jeu, s'en trouvent donc réduites. Notons qu'actuellement, le record d'extraction de logarithmes discrets sur courbes elliptiques concerne une clef de 108 bits. Il a été établi par Robert Harley de l'INRIA, au prix d'un effort de calcul similaire à celui nécessaire à casser une clef RSA de 512 bits[9].

Quelques remarques sur la taille des clefs On constate ici que la taille de la clef n'a pas du tout la même signification que pour les systèmes à clef secrète. Si la meilleure attaque possible pour casser un bon système à clef secrète est la recherche exhaustive, cela n'est pas le cas pour un système à clef publique. Pour trouver une clef secrète RSA de 512 bits, il serait absurde de passer en revue les 2^{512} nombres de 512 bits, ce qui est évidemment hors de portée. Il est beaucoup plus rapide d'utiliser une technique de factorisation. Par ailleurs, les attaques sur les systèmes à clef publique ne sont généralement pas exponentielles en la taille de la clef[10]. Pour un système à clef secrète, il est deux fois plus dur de casser un algorithme ayant une clef de 65 bits qu'un algorithme avec une clef de 64 bits. Cette propriété n'est pas vraie pour RSA. Par conséquent, si un système à clef secrète utilisant des clefs de 128 bits est considéré comme sûr, un système à clef publique avec la même longueur de clef est extrêmement faible.

2 La signature numérique

Dans de nombreuses communications, la confidentialité des données importe peu mais il est nécessaire de s'assurer de leur provenance et de leur intégrité, c'est-à-dire de vérifier qu'elles n'ont pas été modifiées lors de la transmission.

7. Hormis pour certaines classes de courbes, bien identifiées.
8. Les tailles de clefs dans ces deux types de systèmes sont effectivement comparables, la difficulté des deux problèmes sous-jacents étant, semble-t-il, du même ordre.
9. `http://cristal.inria.fr/~harley/ecdl7/`
10. Sauf pour les cryptosystèmes à base de logarithme discret sur courbes elliptiques génériques, sous réserve d'avancéee algorithmique dans ce domaine.

2.1 Principe de la signature

Un procédé de *signature numérique* consiste à adjoindre au texte clair un petit nombre de bits qui dépendent simultanément du message et de son auteur. Pour obtenir les mêmes fonctionnalités que la signature que l'on appose au bas d'un texte à support papier, il faut que chacun puisse vérifier une signature mais que personne ne puisse l'imiter.

Un schéma de signature est donc composé d'une fonction de signature et d'une fonction de vérification. La fonction de signature est paramétrée par une clef secrète propre au signataire ; elle associe à tout message clair une signature. La fonction de vérification, elle, ne nécessite la connaissance d'aucun secret. Elle permet à partir du message clair et de la signature de vérifier l'authenticité de cette dernière.

Un schéma de signature doit donc posséder un certain nombre de propriétés. En particulier, il doit être en pratique impossible de contrefaire une signature : seul le détenteur de la clef secrète peut signer en son nom. La signature ne doit plus être valide si le message clair a été modifié ; il doit être impossible de réutiliser une signature. Enfin, le signataire ne doit pas pouvoir nier avoir signé un message.

Un schéma de signature garantit donc :

- l'identité de la personne émettant le message ;
- l'intégrité des données reçues, c'est-à-dire l'assurance que le message n'a pas été modifié lors de sa transmission ;
- la non-répudiation du message, ce qui signifie que l'émetteur du message ne pourra pas nier en être l'auteur.

C'est pourquoi les procédés de signature numérique constituent une preuve au même titre que la signature manuscrite. Leur valeur juridique est désormais reconnue par la loi (loi 2000-230 du 13 mars 2000).

2.2 Principaux schémas de signature

Signature RSA Certains systèmes de chiffrement à clef publique, dits réversibles, peuvent être utilisés pour construire des schémas de signature. La fonction de signature correspond alors à la fonction de déchiffrement paramétrée par la clef secrète de l'utilisateur, et la fonction de vérification est dérivée de la fonction de chiffrement.

Ainsi, dans le schéma de signature RSA par exemple, un utilisateur signe un message m en lui appliquant la fonction de déchiffrement RSA avec sa clef secrète d. Pour vérifier la signature, il suffit de lui appliquer la fonction de chiffrement RSA paramétrée par la clef publique (e, n) associée, et de vérifier que le résultat de ce calcul correspond bien au message clair envoyé. Les conditions imposées sur la taille des entiers p et q sont les mêmes dans le contexte de la signature que dans celui du chiffrement.

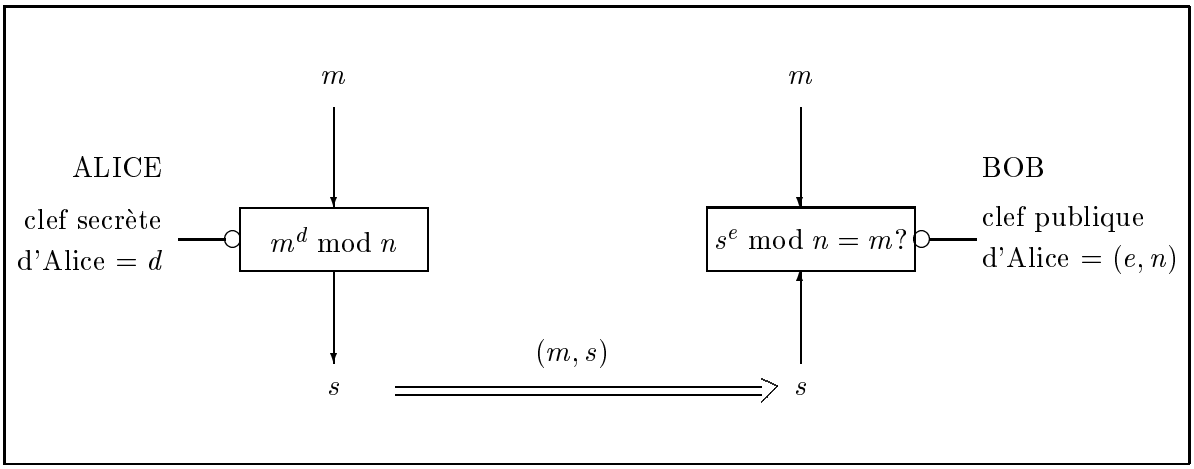

FIG. 4 – *La signature RSA*

DSA Le schéma DSA[11] fait partie d'une famille d'algorithmes de signature basés sur le logarithme discret. Il est devenu en 94 le standard américain de signature numérique pour la protection des informations non classifiées. Les performances de ce schéma sont comparables à celles de RSA pour l'opération de signature, et notablement plus coûteuses pour la phase de vérification. En revanche - et c'est son intérêt majeur - il produit des signatures courtes[12] (par rapport aux signatures RSA, typiquement 1024 bits), à savoir 320 bits, tout en offrant un niveau de sécurité analogue. Le niveau de sécurité mesure ici la difficulté de contrefaire[13] une signature, c'est-à-dire de fabriquer une signature valide sans posséder la clef secrète.
Il est également possible de construire des schémas de signature basés sur le logarithme discret sur courbes elliptiques: ECDSA, adaptation de DSA à ce contexte, en est un exemple.

3 La cryptographie en pratique

Jusqu'ici, nous avons présenté les primitives cryptographiques de base. A présent, regardons comment celles-ci peuvent être utilisées en pratique.

Chiffrement hybride Lorsque l'on souhaite assurer la confidentialité des messages échangés, on n'a en général pas recours à un seul type de chiffrement. En effet, la complexité des opérations en jeu dans les systèmes à clef publique rend le chiffrement extrêmement lent par rapport à un chiffrement à clef secrète (par exemple en hardware, RSA est de l'ordre de 1000 fois plus lent que le DES). D'un autre côté, seul un schéma à clef publique permet un échange sécurisé d'une donnée secrète sans secret préalable commun. Ainsi, on utilisera de préférence un algorithme à clef publique pour échanger une clef secrète, clef qui servira ensuite à chiffrer les échanges à l'aide d'un algorithme symétrique. Cette combinaison des deux techniques permet à la fois d'obtenir la rapidité des chiffrements à clef secrète et de résoudre le problème de

11. Digital Signature Algorithm
12. Cette propriété est très souhaitable, en particulier lors de l'utilisation des signatures numériques pour des certificats (cf. section 3), leur taille importante rendant tout le mécanisme d'autant plus lourd à gérer.
13. Le terme utilisé dans le jargon cryptographique est *forger*.

l'échange de la clef secrète entre les deux interlocuteurs. C'est notamment la solution utilisée pour le chiffrement dans le logiciel PGP (Pretty Good Privacy)[14]. Plus généralement, les systèmes à clef publique sont en pratique uniquement utilisés pour chiffrer des messages très courts.

Signature et fonctions de hachage Pour signer des messages, on a recours, avant d'appliquer un des algorithmes mentionnés en 2.2, à des fonctions de hachage cryptographiques. Une telle fonction, dont la description est entièrement publique, transforme une chaîne binaire de longueur quelconque en une chaîne binaire de longueur fixée (généralement 128 ou 160 bits), appelée *condensé* ou *haché*. Deux raisons motivent l'utilisation de ces fonctions : la première, on l'a vu, est la lenteur des systèmes à clef publique : les messages à signer étant relativement longs, il serait trop coûteux de les signer in-extenso ; on leur applique donc d'abord une fonction de hachage, et on signe le haché du message, et non le message lui-même.

La deuxième raison, liée à la sécurité, est d'empêcher certains types d'attaques sur les schémas de signature. Par exemple dans le cas de RSA, la signature du produit de deux messages[15] est le produit des signatures de chacun d'eux. Cette particularité de RSA le rend vulnérable à la forge de messages (certes le plus souvent inintelligibles), mais pouvant néanmoins représenter une menace, surtout lorsque le mécanisme de signature est utilisé à des fins d'identification. Le hachage du message avant signature permet de pallier cette faiblesse.

Pour pouvoir être utilisée dans des applications cryptographiques, une fonction de hachage doit cependant satisfaire la contrainte suivante : il doit être impossible en pratique de trouver une *collision*, c'est-à-dire deux messages qui aient le même haché. Pour que la recherche de collisions nécessite au moins 2^{64} essais, le haché doit avoir une longueur d'au moins 128 bits.

La plupart des fonctions de hachage utilisées actuellement sont des améliorations de la fonction MD4. Cette dernière était fréquemment utilisée avant sa cryptanalyse en 1996. Parmi les principales fonctions de hachage, on peut citer la fonction MD5 (Message Digest 5). Cette fonction produit un condensé de 128 bits. Certaines faiblesses dans sa construction ont été décelées récemment mais elles ne mettent pas directement en cause sa sécurité. Toutefois, certains préfèrent éviter son utilisation. On peut donc lui préférer le standard américain SHA-1 (Secure Hash Algorithm), utilisé dans le schéma de signature DSA ou la fonction RIPEMD-160, qui a été conçue dans le cadre d'un projet européen[16]. Ces deux fonctions ont également l'avantage de produire des condensés de 160 bits.

Certification de clefs publiques En évitant le partage d'un secret entre les protagonistes, la cryptographie à clef publique est confrontée à un autre problème extrêmement difficile : comment garantir la validité des clefs publiques ? Dès qu'un utilisateur veut chiffrer un message à l'aide d'un algorithme à clef publique ou vérifier une signature, il doit se procurer la clef publique de son interlocuteur ou celle du signataire. Si les clefs publiques sont stockées dans des annuaires non sécurisés, elles risquent d'être interceptées et remplacées par d'autres clefs.

Il est donc possible de fabriquer de fausses signatures simplement en substituant la clef publique d'un utilisateur. Ainsi, si Charlie a remplacé la clef publique d'Alice dans l'annuaire par sa propre clef publique, toute personne recevant un message signé de Charlie pensera que

14. `http://www.pgp.net/pgpnet/pgp-faq/`
15. Un message est vu comme un entier strictement inférieur au module n.
16. `http://www.esat.kuleuven.ac.be/~bosselae/ripemd160.html`

ce message comporte la signature authentique d'Alice. Avant de vérifier une signature, un utilisateur doit donc s'assurer de la validité de la clef publique du signataire.

Ce problème crucial pour toute la cryptographie à clef publique peut être résolu en introduisant une tierce partie, appelée autorité de certification, qui valide le lien entre l'identité des utilisateurs et leurs clefs publiques. Formellement, un *certificat de clef publique* est composé d'un texte clair et d'une signature. Le texte clair contient en particulier une clef publique et une chaîne de caractères identifiant le propriétaire de cette clef. La signature correspond à la signature numérique par l'autorité de certification du texte clair précédent. Si cette signature est authentique, elle prouve que l'autorité de certification valide le lien entre l'identité d'un utilisateur et sa clef publique.

Dès qu'un système possède un grand nombre d'utilisateurs, il faut donc mettre en œuvre une infrastructure de gestion de clefs publiques. Il existe des systèmes très hiérarchisés, tel celui décrit dans la norme ISO X509-v3, dans lesquels la clef d'un utilisateur est certifiée par une autorité dont la clef est à son tour certifiée par une autorité supérieure... Le système PGP utilise au contraire un système sans autorité de certification qui repose sur la confiance. On accepte la clef publique d'un utilisateur parce qu'elle est signée par une personne dont la clef est elle-même signée par quelqu'un que l'on connaît et en qui on a confiance. Toutes ces techniques restent très lourdes à mettre en œuvre. Mais elles sont fondamentales car la sécurité d'un système utilisant un algorithme à clef publique repose en grande partie sur la gestion des clefs publiques.

Quelques mots sur la carte bancaire Ces notions fondamentales de cryptographie suffisent pour comprendre les attaques menées récemment sur les cartes bancaires et pour décrypter la polémique qu'elles déclenchèrent. Lorsqu'on utilise une carte bancaire pour payer une petite somme chez un commerçant, l'opération est effectuée hors ligne, sans échange d'informations avec la banque (ces informations sont regroupées et communiquées en fin de journée). L'unique contrôle au moment du paiement (en plus du code confidentiel) consiste alors à vérifier que la carte utilisée est valide, c'est-à-dire qu'elle a bien été émise par un organisme bancaire. Cette procédure est effectuée au moyen d'une signature RSA : chaque carte possède un identifiant qui a été signé par la banque. C'est cette signature, inscrite dans la puce, qui est vérifiée à chaque transaction par le terminal du commerçant. Toute carte comportant une signature valide est donc considérée comme authentique puisque seule l'autorité bancaire possède la clef secrète RSA qui permet de signer.

L'unique faille de ce système, qui a pu être exploitée pour fabriquer de fausses cartes bancaires, était la taille des clefs RSA utilisées. La signature était produite à l'aide d'une clef de 320 bits alors qu'un nombre de cette taille se factorise aisément en quelques jours avec la puissance actuelle d'un ordinateur grand public (cette opération demandait par contre une grande puissance de calcul il y a une dizaine d'années). Pour fabriquer de fausses cartes, il suffisait donc d'obtenir la clef publique de la banque (le nombre n, produit de deux nombres premiers), ce qui est relativement facile puisque cette donnée est présente dans tous les terminaux de paiement. La factorisation de ce nombre fournit alors directement la clef secrète correspondant qui peut être utilisée pour "imiter" la signature de la banque. Il ne restait plus qu'à choisir arbitrairement un identifiant et à inscrire dans la puce d'une carte vierge la signature associée. Une telle carte sera acceptée par le terminal. Comme il ne s'agit pas d'une vraie carte bancaire modifiée mais d'un leurre, la protection par code confidentiel n'existe plus.

Ni la sécurité de la carte à puce, ni celle de l'algorithme de signature RSA ne sont remises en cause par cette attaque. Le seul problème était la taille de clef utilisée, qui est heureusement de 792 bits dans les cartes les plus récentes. Dans tout système cryptographique digne de ce nom, la taille des clefs doit évidemment évoluer en même temps que la puissance des ordinateurs.

Références

[Dif] W. Diffie, M.E. Hellman: *New directions in cryptography*, IEEE Transactions on Information Theory, 22 (1976), pp. 644-654.

[Fei1] H. Feistel: *Cryptography and computer privacy*, Scientific American, 228, mai 1973, pp. 15-23.

[Fei2] H. Feistel: *Block cipher cryptographic system*, U.S. patent 3,798,359, 19 mars 1974.

[Riv] R.L. Rivest, A. Shamir, L.M. Adleman: *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM, 21 (1978), pp. 120-126.

[Sha] C.E. Shannon: *Communication theory of secrecy systems*, Bell System Technical Journal, 28 (1949), pp. 656-715.

Citons également :

- S. Singh: *Histoire des codes secrets.* Jean-Claude Lattès, 1999.
- A.J. Menezes, P.C. van Oorschot, et S.A. Vanstone: *Handbook of Applied Cryptography.* CRC Press, 1997. La plupart des chapitres de cet ouvrage peuvent être téléchargés gratuitement à partir du serveur `http://cacr.math.uwaterloo.ca/hac/`.
- B. Schneier: *Applied Cryptography.* Wiley Inc., 1996.