Le système cryptographique R.S.A.

Introduction

Le système R.S.A., provenant des initiales des noms de ses inventeurs américains en 1977 : Ronald Rivest (informaticien), Adi Shamir (informaticien) et Leonard Adleman (mathématicien), est le chiffre le plus influent de la cryptographie moderne. Il est utilisé actuellement dans plus de 85 % des échanges sécurisés mondiaux.

1 L'arithmétique du système R.S.A.

Soit p et q deux nombres premiers distincts ($p \neq 2$ et $q \neq 2$).
On pose $n = pq$ et on désigne par e un entier tel que :

$$1 < e < (p-1)(q-1) \quad \text{et } e \text{ premier avec } (p-1)(q-1)$$

1) Montrer qu'il existe d unique tel que :

$$1 \leqslant d < (p-1)(q-1) \quad \text{et} \quad ed \equiv 1 \ [(p-1)(q-1)]$$

2) Prouver que, pour tout $m \in \mathbb{N}$, $m^{ed} \equiv m \ [n]$

3) On choisit $p = 41$ et $q = 53$.
   a) Calculer n et $(p-1)(q-1)$
   b) On choisit $e = 1\,427$. Montrer que e est premier avec $(p-1)(q-1)$ à l'aide de l'algorithme d'Euclide. En remontant cet algorithme, en déduire d.

2 Le fonctionnement du système R.S.A.

Nous l'expliciterons dans le cas général avec, en parallèle, un exemple numérique « *naïf* ».
Pour permettre à A (Alice) – ou à tout autre personne – de transmettre des informations à B (Bob), le protocole est le suivant

1) Le destinataire B choisit deux entiers premiers distincts p et q ainsi qu'un nombre e (entre 1 et $(p-1)(q-1)$) premier avec $(p-1)(q-1)$.
2) B calcule l'unique entier d tel que

$$1 \leqslant d < (p-1)(q-1) \quad \text{et} \quad e \times d \equiv 1 \ [(p-1)(q-1)].$$

3) B diffuse largement les entiers $n = pq$ et e tout en gardant secret les entiers p, q et d.
4) Pour envoyer un message à B, A convertit ce message en une suite de nombres inférieurs à n.

   Pour chiffrer chacun de ces nombres m, A calcule le nombre c entre 1 et n tel que $m^e \equiv c \ [n]$, qu'elle transmet à B.

5) B déchiffre c en calculant $c^d \mod(n)$.
Et comme $c^d = m^{de}$ et que, d'après la partie A, $m^{de} \equiv m \ [n]$, B « retrouve » le nombre m.

3 Exemple de fonctionnement

On reprend les nombres : $p = 41$, $q = 53$ et $e = 1\,427$

Alice voudrait transmettre le mot « salut ».
Alice traduit chaque lettre à l'aide du code ASCII (prononcez "aski") obtenant les nombres de trois chiffres suivants :

a	b	c	d	e	f	g	h	i	j	k	l	m
97	98	99	100	101	102	103	104	105	106	107	108	109
n	o	p	q	r	s	t	u	v	w	x	y	z
110	111	112	113	114	115	116	117	118	119	120	121	122

$$m_1 = 115 \ , \ m_2 = 097 \ , \ m_3 = 108 \ , \ m_4 = 117 \ , \ m_5 = 116$$

Alice code ensuite ces nombres. Pour le premier nombre :

$$m_1^e \equiv c_1 \ [n] \ \Leftrightarrow \ 115^{1427} \equiv 1064 \ [2173] \quad \text{d'où} \quad c_1 = 1064$$

1) Proposer un programme permettant de vérifier qu'il est transmis à Bob les nombres :

$$c_1 = 1064 \ , \ c_2 = 950 \ , \ c_3 = 1305 \ , \ c_4 = 1546 \ , \ c_5 = 386$$

Bob décode alors ces nombres. Pour le premier nombre :

$$c_1^d \equiv m_1 \ [n] \ \Leftrightarrow \ 1\,067^{1083} \equiv 115 \ [2173] \quad \text{d'où} \quad m_1 = 115$$

2) Sur le même modèle proposer un programme permettant de vérifier que Bob après décodage obtient bien m_1, m_2, m_3, m_4 et m_5

3) Bob reçoit après les nombres suivant :
$1402 - 145 - 1116 - 2012 - 2013 - 119 - 950 - 1322 - 1305 - 544$
Que lui dit Alice ?

4 Commentaires

4.1 Un système à clef publique

Le couple (n, e) connu de tout un chacun permet donc à « tout public » de transmettre un message à Bob.

4.2 La sécurité du système

Si l'on sait factoriser n sous la forme $n = pq$, connaissant e, on trouve d sans problème et la lecture du courrier électronique de Bob se fait comme dans un livre ouvert.

De même, on montre que, connaissant n, e et d, on trouve rapidement p et q.

La sécurité du système tient pour l'essentiel dans :

- la construction de nombres premiers « grands » (on sait faire) ;
- la difficulté de décomposer un nombre grand (300 chiffres, par exemple) en produit de deux nombres premiers.

Mais personne ne sait si une attaque du R.S.A. qui éviterait la factorisation est impossible !

4.3 Les fonctions à sens unique

La fonction *one-way*, de Bob, f_B est la fonction de $X = \{1, 2, \ldots, n\}$ dans X définie par $f(m) = m^e \ [n]$: elle est connue de tout le monde.
En revanche, la fonction réciproque $f_B^{-1}\ m \mapsto m^d \ [n]$, n'est connue que de Bob et est pratiquement impossible à calculer, sauf si l'on dispose d'une information supplémentaire (*trapdoor information*), d'où le nom donné à une telle fonction : **fonction trappe**.

4.4 Le R.S.A. 155

En 1999, un nombre de 155 chiffres (qui servait de clef à un système R.S.A.) a été décomposé en produit de deux facteurs premiers de 78 chiffres chacun (C.W.I. Amsterdam) [1]
Le temps cumulé de calculs a été évalué à 8 000 Mips (8 000 millions d'instructions par seconde pendant un an !).
Depuis, la Société R.S.A. *Data Security* recommande des nombres de 309 chiffres, voire 617 chiffre.
⚠ C'est la taille d'un nombre mesurée en bits qui conduit à 155, 309, 617 ...

5 Pour aller plus loin

1) On garde : $p = 41$, $q = 53$ et on prend $e = 1\,071$. Montrer que e est premier avec $(p-1)(q-1)$ puis calculer d.
2) **Authentification** : Pour l'instant, rien n'assure à Bob que le message reçu vient bien D'Alice !

   Le système R.S.A. offre une parade imbattable.

   Alice dispose également d'une fonction trappe f_A (f_A est publique et f_A^{-1} n'est connue que d'Alice).

   Alice envoie à Bob un message contenant
   - ce qu'elle avait à lui dire ;
   - une double signature : A, $f_A^{-1}(A)$

   Comment Bob peut-il s'assurer que le message reçu vient bien d'Alice ?

1. À titre de curiosité !

10 941 738 641 570 527 421 809 707 322 040 357 612 003 732 945 449 205 990 913 842 131 476 349 984 288 934 784 717 997 257 891 267 332 497 625 752 899 781 833 797 076 537 244 027 146 743 531 593 354 333 897 =
102 639 592 829 741 105 772 054 196 573 991 675 900 716 567 808 038 066 803 341 933 521 790 711 307 779 × 106 603 488 380 168 454 820 927 220 360 012 878 679 207 958 575 989 291 522 270 608 237 193 062 808 643