TIPE :Quand mathématiques et informatique entrent dans le secret public...

Marie-Liane LEKPELI (L2 Mathématiques)

7 juillet 2009

Table des matières

Remerciements à François-Xavier Roblot,encadrant de ce TIPE, pour ses conseils avisés notamment concernant le choix du thème et sa disponibilité.

Remerciements également à Thierry Dumont,enseignant-chercheur à l'Université Claude Bernard pour la présentation on ne peut plus entraînante de l'outil latex.

Introduction

Le besoin de communication secrète et la mise en oeuvre de moyen pour y arriver est à peu près aussi vieux que l'écriture. Les premiers documents cryptographiques dateraient de 1900 avant JC. Cette science qui fascine et passionne, on le devine, a énormément évolué depuis ses débuts. Préservation de secret de profession ou encore démarche de stratégie militaire ou politique, les raisons qui ont motivé la course à l'ingéniosité dont elle faisait l'objet et son développement sont multiples.

De ces hiéroglyphes non conformes considérés comme premières traces documentaires de cryptographies à la machine ENIGMA de **Arthur Scherbius** en passant par les scytales grecques et les substitutions alphabétiques de César, la cryptologie nous emporte dans sa palpitante histoire. Grâcieusement nourrie par la théorie des nombres,elle prend avec l'avènement des ordinateurs et du réseau internet un tournant décisif : c'est le début de l'ère moderne. Cette ère qui verra naître la cryptographie à clé publique dont on doit l'idée à **Whitfield Diffie** et **Martin Hellman**.

"Je prends une lettre ; l'enferme dans un coffre et cache ce coffre quelque part dans New York... si je vous demande ensuite de lire la lettre, il n'est pas question de sécurité : c'est de l'obscurité. Autre exemple : je prends une lettre, l'enferme dans un coffre et vous donne le coffre avec ses spécifications de conception et une centaine d'autres coffres identiques avec leurs combinaisons, de telles manière que vous et les meilleurs perceurs de coffres-forts puissiez étudier le mécanisme de verrouillage... si vous ne pouvez toujours pas ouvrir le coffre contenant la lettre, il est alors question de sécurité".

L'enjeu principal de la cryptographie moderne est parfaitement décrit dans cette phrase de **Bruce Schneier**.

Désormais la cryptologie est appliquée à la sécurisation de l'internet et au commerce électronique entre autres. Arithmétique,algèbre, complexité algorithmique et autres se mettent à son service pour l'aider à faire face au renouvellement incessant des défis qu'elle doit relever.

Au milieu de ce perpétuel mouvement, se dresse le système RSA. Du nom de ses inventeurs **Ron Rivest, Adi Shamir** et **Leonard Adleman**, RSA s'inscrit dans la ligne de la cryptographie à clé publique.Il a réussit l'exploit de s'imposer et de se maintenir 32 ans après son invention parmi les systèmes cryptographiques les plus utilisés dans le monde.

Expliquer le fonctionnement de ce système mais aussi mettre en lumière ses armes contre le temps, voici le challenge que va tenter de relever ce TIPE.

Chapitre 1

RSA : l'algorithme

On va ici s'intéresser à la démarche à suivre pour envoyer et recevoir un message en utilisant RSA. On entend par envoyer et recevoir, coder et décrypter.

Prenons donc l'exemple d'une conférence de messagerie instantanée sur internet entre au moins trois personnes. A un moment donné pour une raison ou pour une autre, deux des antagonistes veulent se dire quelque chose en excluant les autres de la compréhension tout en restant dans la conférence. La situation est peut-être un peu confuse mais ne faisant office que de décor,elle peut s'autoriser quelques incohérences. Nommons nos deux cachotiers D1(comme destinataire 1) et E1(comme émetteur 1).

* D1 prend deux grands nombres premiers p et q et en fait le produit que l'on notera **module**. Ce produit peut être diffusé sur le canal public sans problème.

* D1 choisit ensuite un autre grand nombre **exppr** pour exposant privé premier avec le produit (p-1)(q-1).
  Nous expliquerons un peu plus tard l'obsession des **GRANDS** nombres.
  Le produit (p-1)(q-1) correspond à l'indicateur d'Euler de **module** et peut être noté $\varphi(module)$.

* A partir de **exppr**, D1 calcule **exppu**(pour exposant public) tel que **exppu** et **exppr** sont inverses dans $\mathbb{Z}/\varphi(module)\mathbb{Z}$ c'est-à-dire que $exppr * exppu \equiv 1 \pmod{\varphi(module)}$

E1 comme tous les autres participants à la conférence aura connaissance du couple (module,exxpu) qui constitue la **clé publique**. (module,exppr) est appelé **clé privée**.
Notons **clair** l'information à transmettre à D1. Cette dernière circulera sous la forme **crypte** suivante :

$$crypte \equiv clair^{exppu} \pmod{module}$$

Avec ce procédé(dans des conditions correctes d'utilisation, notamment avec un choix judicieux de nombres) seul D1 sera capable de déchiffrer le message en utilisant la formule ci-dessous :

$$clair \equiv crypte^{exppr} \pmod{module}$$

On peut très vite remarquer que tous ces calculs se font sur des entiers et que normalement les messages échangés sont constitués de mots. Cependant le passage de chaines de caractères à des tableaux d'entiers(et vice-versa) est loin d'être insurmontable. On peut penser au codage

ASCII par exemple. En pratique E1 découpe l'information à transmettre en petit morceau dont la correspondance en entier (que l'on a noté **clair**) est inférieure à p et q. Il s'en suit, comme p et q sont des nombres premiers que **clair** est premier avec p et q, donc avec **module**. Nous verrons par la suite pourquoi ceci est important.

Une dernière remarque est que les éléments peuvent ne pas être choisis dans l'ordre décrit. Ce qui compte ce sont les relations qui les lient.

Nous venons de voir de façon condensée la démarche à suivre. Revenons sur cette dernière en faisant de petites haltes mathématiques.

Chapitre 2

RSA : les fondements mathématiques de la mise en oeuvre

Commençons par le commencement... c'est-à-dire ici les nombres premiers.

2.1 La génération de nombres premiers

D1 doit choisir deux nombres premiers suffisament grands. Ceci est toujours possible car l'ensemble des nombres premiers est infini. Avant de donner un preuve de cette affirmation rappelons ce qu'est un nombre premier.

Définition *Soit n∈ $\mathbb{N}$. n est dit premier s'il admet* ***exactement*** *deux diviseurs : 1 et lui-même.*

On peut déduire de cette définition le théorème suivant qui va nous servir pour la preuve.

Théorème *Dans $\mathbb{N} \setminus \{0,1\}$ tout nombre admet au moins un diviseur premier.*

En effet, soit n $\in \mathbb{N} \setminus \{0,1\}$.n est divisible par lui-même. Deux cas sont alors possibles :
- n est premier : La condition est immédiatement vérifiée.
- n n'est pas premier et alors il admet p+2 diviseurs avec p $\in \mathbb{N} \setminus \{0\}$. Si on note ppd le plus petit de ces diviseurs, ppd est forcément premier sinon il aurait un diviseur d plus petit que lui et qui diviserait n.

Nous sommes maintenant suffisament armé pour montrer le caractère infini de l'ensemble des nombres premiers en raisonnant par l'absurde.

Supposons que cet ensemble soit fini.Notons n son cardinal et p_1, p_2, ... p_n ses éléments. Soit alors

$$N = \prod_{i=1}^{n} p_i + 1$$

N n'est divisible par aucun p_i car $p_i|N \Rightarrow p_i|1$.Ceci entraine que N est premier or N n'est pas un p_i. On en déduit que l'ensemble des nombres premiers est bien infini.

On sait maintenant que D1 trouvera toujours dans l'ensemble des nombres premiers deux éléments qui correspondent à ses critères de taille. Les nombres à choisir étant très grands, il est invraisemblable que D1 les choisissent parmi les nombres premiers qu'il connait de mémoire(son appelation peut prêter à confusion mais D1 est bien un être humain). Il faudra qu'il se serve d'un algorithme un peu plus évolué de génération de nombres premiers. Intéressons-nous ici à la **méthode de Lucas** qui en est un.

La méthode de Lucas

Commençons par énoncer le théorème de Lucas.

Théorème *Soit $N \in \mathbb{N}$. On cherche un critère de primalité de N. Il est donc inutile de s'intéresser au cas $N < 3$ que l'on sait résoudre très rapidement. On peut dnc supposer que $N \geq 3$. soit en plus $a \in \mathbb{N}$, tel que PGCD(a,N)=1. si*

$$a^{N-1} \equiv 1 \pmod{N}$$

et

$$a^{\frac{N-1}{q}} \neq 1 \pmod{N} \quad \forall q \ \ premier \ \ tel \ \ que \ \ q \mid N-1$$

alors ***N est premier.***

Nous donnerons une preuve de ce théorème lorsque nous aurons vu ce qu'est l'indicateur d'Euler. Patience c'est pour bientôt. Revenons pour l'instant à notre génération de nombres premiers.

On part d'un nombre premier connu que l'on peut noté r. On pose ensuite :

N=kr+1 avec k un petit entier

On va faire varier k dans un intervalle en fonction de la vitesse à laquelle on veut atteindre de grandes tailles(par exemple $1 \leq k \leq 100$). On teste à chaque fois la primalité de N à partir du théorème de Lucas.

- Si N est premier mais qu'il n'est pas de la taille qu'on souhaite, on recommence en remplaçant r par N.
- Si N n'est pas premier, on fait varier k et on recommence le test.

La suite de notre algorithme nous mène au produit (p-1)(q-1) que l'on dit correspondant à l'indicateur d'Euler de **module** sur lequel nous allons maintenant nous pencher.

2.2 L'indicateur d'Euler

Soit n $\in \mathbb{N}$. L'indicateur d'Euler de n noté $\varphi(n)$ est le nombre d'éléments inversibles dans $\mathbb{Z}/n\mathbb{Z}$. Ce nombre d'éléments correspond également au nombre d'entiers a inférieurs à n et tels que PGCD(a,n)=1. En effet PGCD(a,n)=1 (d'après le théorème de Bézout) $\Leftrightarrow \exists$ (b,q) $\in \mathbb{Z} * \mathbb{Z}$ tels que a*b + q*n= 1. On en déduit que $ab \equiv 1 \pmod{n}$ par définition de la congruence modulo n. Il s'en suit que PGCD(a,n)=1 $\Leftrightarrow$ a inversible dans $\mathbb{Z}/n\mathbb{Z}$

Trois propriétés de cet indicateur sont importantes à relever dans notre contexte.

- Premierement de façon très immédiate si p est un nombre premier alors

  $$\varphi(p) = p - 1$$

  En effet si p est premier tous les entiers inférieurs à p non nuls sont premiers avec lui.

- Ensuite si p est un nombre premier et α un entier naturel strictement positif alors

  $$\varphi(p^\alpha) = (p-1)p^{\alpha-1}$$

- Enfin si PCDG(m,n)= 1 alors

  $$\varphi(mn) = \varphi(m)\varphi(n)$$

.

$\varphi(n)$ est aussi l'ordre du groupe $(\mathbb{Z}/n\mathbb{Z})^*$. C'est-à-dire pour tout élément a de $\mathbb{Z}/n\mathbb{Z}$ on a :

$$a^{\varphi(n)} \equiv 1 \pmod{n}$$

C'est le théorème d'Euler-Fermat.

Revenons comme promis sur le théorème de Lucas en énonçant d'abord un théorème qui servira pour la preuve : **le théorème de Lagrange**.

Théorème *Soient G un groupe fini et H un sous-groupe de G. Alors Card(H) | Card(G)*

Avant de poursuivre démontrons ce théorème.

Considérons pour ce faire un groupe G et H un sous-groupe de G. Soit la relation $\sim$ définie sur H par :

$$\forall a, b \in G,\ a \sim b \ \ lorsque\ ab^{-1} \in H$$

Vérifions que $\sim$ est bien une relation d'équivalence c'est-à-dire qu'elle est réflexive, symétrique et transitive.

* Soit a $\in$ G. G étant un groupe, a est inversible et son inverse noté a^{-1} appartient à G. De plus $aa^{-1} = 1$. 1 est l'élement neutre et appartient donc à H qui est un sous-groupe. Ainsi

  $$a \sim a\ \forall a \in H$$

  D'où la réflexivité de $\sim$

* Soit a,b et c $\in$ G tels que a$\sim$b (1) et b$\sim$c (2).

  $$(1) \Leftrightarrow ab^{-1} \in H$$

  $$(2) \Leftrightarrow bc^{-1} \in H$$

  Par suite on a :

  $$ab^{-1}.bc^{-1} \in H$$

  Ceci équivaut à :

  $$ac^{-1} \in H$$

  D'où la transitivité de $\sim$

* Soit a et b $\in$ G tels que a$\sim$b. Ceci signifie par définition que $ab^{-1} \in H$. H étant un sous-groupe, on a par suite que

  $$(ab^{-1})^{-1} = ba^{-1} \in H$$

  On a ainsi que a$\sim$b $\Rightarrow$ b$\sim$c. D'où la transivité de $\sim$

Nous venons de vérifier que $\sim$ est bien une relation d'équivalence.

On va maintenant montrer qu'il existe une bijection entre chaque classe d'équivalence pour la relation $\sim$ et le sous-groupe H.

Soit a $\in$ G. Notons $\dot{a}$ sa classe d'équivalence.pour la relation $\sim$. Posons l'application f définie sur H comme suit :

$$f : H \to \dot{a}$$

$$h \mapsto ha$$

ha est bien un élément de $\dot{a}$. En effet on a

$$a(ha)^{-1} = aa^{-1}h^{-1} = h^{-1}$$

H étant un sous-groupe, $\forall h \in H$, $h^{-1} \in H$.

Il reste à vérifier que f est bien une bijection. Pour cela montrons que chaque élément de $\dot{a}$ admet un et un seul antécédent par f. Soit donc b $\in \dot{a}$. h$\in H$ est antécédant de b si et seulement si

$$ah = b$$

Ceci entraine que

$$h = ba^{-1}$$

b étant un élément de *dota* on sait que $a \sim b$ et donc que $b \sim a$(par symétrie). Ceci signifie que $ba^{-1} \in H$. b a donc un antécédent unique qui est ba^{-1}.

On sait donc qu'il existe une bijection entre chaque classe d'équivalence pour la relation $\sim$ et H. Ceci entraîne que les classes d'équivalence pour la relation $\sim$ ont toutes le même nombre d'éléments qui est celui de H.

Pour conclure, il faut dire que l'ensemble des classes d'équivalence pour $\sim$ que l'on note $G/\sim$ consitue une partition de G. Chaque partie de cette partition contient Card(H) éléments. On en déduit que le nombre d'éléments de G est un multiple de celui de H.

Il peut être utile de rappeler que **cardinal d'un groupe** et **ordre d'un groupe** désignent le même concept.

Revenons à notre théorème de Lucas.

On suppose maintenant que l'on dispose d'un N et d'un a qui répondent aux conditions du théorème de Lucas. Dans ce cas l'ordre de a par définition de l'ordre d'un élément est (n-1). Le **théorème de Lagrange** permet de poursuivre en disant que dans ce cas :

$$(n-1) \mid \varphi(n)$$

De plus, on sait par définition que

$$\varphi(n) \leq n-1$$

Cette inégalité combinée à la condition de divisibilité donne l'égalité suivante :

$$\varphi(n) = n-1$$

De là on déduit la primalité de n. En effet tous les entiers inférieurs à n sont premiers avec lui. n n'est donc divisible que par 1 et lui-même.

L'équivalence entre l'inversibilité d'un élément a dans $\mathbb{Z}/n\mathbb{Z}$ et PGCD(a,n)= 1 justifie l'existence du **exppu** de la démarche calculé à partir de **exppr**.Ce calcul d'inverse se fait à partir de l'algortithme d'Euclide étendu.

Il ne reste plus qu'à se pencher sur l'égalité du décryptage. A savoir

$$clair \equiv crypte^{exppr} \pmod{module}$$

2.3 L'égalité du décryptage

On affirme que :

$$\text{Si PGDC}(\varphi(module),\text{exppr})=1 \text{ et } exppu * exppr \equiv 1 \pmod{\varphi(module)} \quad (1)$$

$$\text{Alors } crypte \equiv clair^{exppu} \pmod{module} \ (2) \Leftrightarrow clair \equiv crypte^{exppr} \pmod{module}$$

Pour montrer cette implication , on va d'abord appliquer le petit théorème de Fermat à **clair** et p puis à **clair** et q. On peut se le permettre car **clair** est premier avec p et q(voir chapitre 1) On obtient :

$$clair^{p-1} \equiv 1 \pmod{p} \ \ et \ \ clair^{q-1} \equiv 1 \pmod{q}$$

On rappelle que $\varphi(module) = (p-1)(q-1)$. De plus on a la propriété suivante pour la congruence :

$$a \equiv b \pmod{c} \Rightarrow a^n \equiv b^n \pmod{c}$$

Ceci entraine que :

$$clair^{\varphi(module)} \equiv 1 \pmod{p} \ \ et \ \ clair^{\varphi(module)} \equiv 1 \pmod{q}$$

Ainsi

$$(clair^{\varphi(module)} - 1) \mid p \ \ et \ \ (clair^{\varphi(module)} - 1) \mid q$$

On en déduit que :

$$(clair^{\varphi(module)} - 1) \mid pq \ \Rightarrow clair^{\varphi(module)} \equiv clair \pmod{pq} \ (3)$$

(1) $\Leftrightarrow exppu * exppr = k\varphi(module) + 1$ avec k $\in \mathbb{Z}$ (2) $\Leftrightarrow crypte^{exppr} \equiv clair^{exppu*exppr}$ (mod $module$) Les deux équivalences précendentes et (3) entrainent que :

$$crypte^{exppr} \equiv clair \pmod{module}$$

On rappelle que $clair < p$. Ceci entraîne que $clair < module$. **clair** est donc l'unique solution de l'équation d'inconnue x :

$$crypte^{exppr} \equiv x \pmod{module}$$

qui vérifie

$$x < module$$

L'on peut à ce stade prétendre savoir tout(ou presque tout) ce qui se passe mathématiquement parlant à chaque étape de la démarche de D1 et E1.

On rappelle que la communication secrète se fait sur un canal public. De plus D1 pousse l'audace jusqu'à laisser à la connaissance de tous le couple **(exppu,module)**. Notons que au vu de comment l'envoi de message fonctionne D1 n'avait pas trop le choix s'il voulait que E1 puisse communiquer avec lui. Mais ceci est un détail. Naturellement on se demande sur quoi s'appuie nos deux acteurs pour prendre un tel "risque".

Chapitre 3

RSA : Les clés de la sureté

Admettons ici que nous sommes dans des conditions optimales d'utilisation de RSA. C'est-à-dire que D1 et E1 connaissent toutes les précautions à prendre et les prennent effectivement. Nous reviendrons un peu plus tard sur cet ensemble de précautions nécessaires.

Le système RSA joue sur une limite technique qui est la factorisation des grands nombres.Ce problème occupe depuis des siècles les mathématiciens.Si il a pu être considéré comme un jeu de matheux inutile, la cryptographie moderne (avec RSA notamment) lui donne un statut bien différent. Essayons dans ce TIPE de le traiter avec tout le respect que nous sommes en mesure de lui donner au vu de nos dispositions mathématiques.

3.1 Le casse-tête de la factorisation des grands nombres

Factoriser un nombre consiste à trouver ses facteurs premiers.Ces facteurs premiers existent toujours et bien plus il caractérise le nombre.Ceci vient du fait que tout entier(en dehors de 0 et 1) se décompose de façon unique en produit de facteurs premiers.Nous allons le remontrer

Soit n $\in \mathbb{N}$ privé de 0 et de 1. Attaquons nous d'abord à l'existence de ladite décomposition.

- Si n est premier la décomposition est toute faite. n=n.
- Sinon d'après d'après le théorème précédent n admet au moins un facteur premier que l'on note p_1. Si l'on note q_1 le quotient de la division euclidienne de n par p_1 on a

$$n = p_1.q_1$$

avec $q_1 < n$ On renouvelle le procédé de façon récurrente jusqu'à obtenir un quotient premier ce qui arrivera toujours (car tout nombre admet au moins un diviseur premier).

Nous savons que la décomposition en facteurs premiers existe montrons maintenant qu'elle est unique. On suppose qu'elle ne l'est pas. Soit

$$n = \prod p_i^{\alpha_i}$$

une décomposition en facteurs premiers de n et

$$n = \prod q_i^{\beta_i}$$

une autre décomposition de n avec $q_i \neq p_i \ \forall i$.

Il est immédiat que l'on peut exclure le cas $p_i = q_i \ \ \forall i$ avec les exposants non tous égaux. En effet l'on aurait pour au moins un i $p_i^{\alpha_i} = p_i^{\beta_i}$ avec $\alpha_i \neq \beta_i$ Ce qui est impossible car les p_i sont premiers donc différents de 0 et 1.

On sait que :

$$q_i \mid n$$

Ceci signifie que

$$\exists j \in \mathbb{N}\backslash\{0,1\} \quad tel \quad que \; q_i \mid p_j$$

or les p_i et q_i sont premiers donc par définition d'un nombre premier on a

$$q_i = p_j$$

A partir de cette égalité, on voit que les deux décompositions sont identiques.

Prouver que l'on peut toujours factoriser un nombre est une chose, le factoriser en est une autre. En théorie, la preuve de l'existence de la décomposition, indique comment factoriser tout entier.La factorisation n'est donc pas en elle-même un exercice impossible. Cependant il faut tenir compte du facteur temps.Il est déjà assez agaçant et fastidieux pour certains d'effectuer à la main seulement 10 divisions successives.Bien sûr on peut compter sur cet outil extraordinaire qu'est l'ordinateur. Mais là encore il y a des limites.

Pour avoir une idée de ces limites, on peut relever qu'en 2007,le record de factorisation était détenu par une équipe internationale de recherches avec la factorisation de $2^{1039}-1$ qui comporte 307 chiffres. La performance étant le fruit de **ONZE MOIS** de travail.

Il faut noter de plus que $2^{1039}-1$ a une forme spéciale : il est proche d'une puissance de deux.Il a été conjecturé que cette information a aidé dans l'établissement de ce record.

Dans le cas de RSA, le nombre que l'on cherche à factoriser(c'est-à-dire **module**) est le produit d'exactement deux nombres premiers.Ce qui n'est pas le cas de $2^{1039}-1$. **module** est un nombre dit RSA et le record de factorisation pour cette catégorie est celle d'un nombre de 200 chiffres.

Plusieurs méthodes de factorisation se sont succédées, les anciennes s'effaçant devant l'efficacité supérieure des nouvelles.Elles progressent très rapidement et face à cette rapide progression, l'obsession des GRANDS nombres déjà évoquée prend tout son sens ; la contrainte de complexité d'algorithmes restant la seule barrière contre une cryptanalyse par factorisation de **module**.

On ne saurait finir cette partie sans s'arrêter sur une méthode de factorisation.Nous choisirons la méthode ρ de Pollard qui allie arithmétique, théorie des groupes et probabilités.

La méthode ρ de Pollard

On cherche à factoriser **module**= pq. $\forall x \in \mathbb{N} \quad PGCD(x, module) \in \{1, p, q, module\}$

Le principe des tiroirs appliqué à la congruence modulo p nous affirme que dans une suite de p+1 entiers choisi aléatoirement il y en a forcément deux qui sont congrus modulo p. Réduisons un peu nos attentes. Plutôt qu'une probabilité de 1/1 que cela arrive cherchons juste une probabilité supérieure à 1/2.

Intuitivement, on se dit que pour avoir deux nombres congrus modulo avec cette probabilité une suite de moins de p+1 entiers pourrait faire l'affaire. Il s'agit maintenant de détermnier de combien d'entiers doit être constituée cette suite.

Supposons pour se faire que l'on prend k entiers dans Z/pZ (avec $k < p$) et on les note x_i avec $1 \leq i \leq k$. Supposons également que l'on veuille que les p_i soit tous différents.

On a p possiblités pour choisir x_1, p-1 pour choisir x_2 et ainsi de suite jusqu'à x_k. On obtient une probabilité de $A = \frac{p(p-1)\ldots(p-k+1)}{p^k}$ que les x_i soit tous différents.

Travaillons un peu sur ce rapport A.

$$A = \frac{(p-1)(p-2)\cdots(p-k+1)}{p^{k-1}}$$

On peut vérifier rapidement que l'on a également

$$A = (1 - \tfrac{0}{p})(1 - \tfrac{1}{p}) \cdots (1 - \tfrac{k-1}{p})$$

Ce qui équivaut à :

$$A = \prod_{i=0}^{k-1} (1 - \frac{i}{p})$$

Rappelons le développement limité de l'exponentielle au voisinage de 0 à l'ordre 1 :

$$\mathrm{e}^x = 1 + x + o(x)$$

Pour $p \gg k$ on a $\dfrac{i}{p}$ proche de 0 $\forall i \in [0; k-1]$

Permettons nous donc d'utiliser le développement limité pour approximer A. On obtient :

$$A \approx \prod_{i=0}^{k-1} \mathrm{e}^{\frac{-i}{p}}$$

Grâce aux propriétés de l'exponentielle on peut écrire :

$$A \approx \mathrm{e}^{-\sum_{i=0}^{k-1} \frac{i}{p}}$$

Pour simplifier encore plus on rappelle que :

$$\sum_{i=0}^{k-1} \frac{i}{p} = \frac{\frac{(k-1)k}{2}}{p}$$

On a donc :

$$A \approx \mathrm{e}^{\frac{-k(k-1)}{2p}}$$

Pour que l'on ait deux nombres(au moins) congrus modulo p avec une probabilité supérieure à $\dfrac{1}{2}$, il faut que :

$$A < \frac{1}{2}$$

Par suite avec la dernière approximation de A on a :

$$\frac{-k(k-1)}{2p} < \ln \frac{1}{2}$$

Puis

$$-k(k-1) + 2p \ln 2 < 0$$

En considérant le polynôme de l'inégalité précédente comme un polynôme en k, on trouve qu'elle est vrai pour :

$$k > \frac{1 + \sqrt{1 + 8p \ln 2}}{2}$$

Si on note $\lceil x \rceil$ le plus petit entier supérieur strictement à x, on utilisera en pratique :

$$k > \lceil 0,5 + \sqrt{0,25 + 2p \ln 2} \rceil$$

Affirmons sans le vérifier que l'ordre de $\sqrt{p}$ est supérieur à $\lceil 0,5 + \sqrt{0,25 + 2p \ln 2} \rceil$. Cette vérification est une question de calcul que nous nous permettons de nous épargner.

Ainsi si l'on prend aléatoirement k entiers que l'on note x_i avec k de l'ordre de $\sqrt{p}$ alors
$\exists$ i et $j \in \mathbb{N}$ avec $i \neq j$ tels que $x_i \equiv x_j \pmod{p}$ (avec une probabilité supérieure à $\frac{1}{2}$)

Par suite on aura PGCD($x_i - x_j$,n)>1. On pourra ainsi en déduire une factorisation de **module**.

En pratique, les x_i ne sont pas choisis aléatoirement mais à partir d'une fonction.

On imagine bien que ce serait trop beau si à partir de cette méthode on pouvait casser RSA en temps raisonnable.Pour ôter tout doute,on précise qu'elle est bien trop coûteuse même avec des améloriations. En effet pour une factorisation d'un entier de seulement 6 chiffres, il faut environ une soixantaine de calculs.

A défaut de factoriser **module**, on pourrait avoir envie pour percer le système de calculer $\varphi(module)$.Nous allons maintenant voir que ceci ne s'avère pas plus aisé.

3.2 Détermination de $\varphi(module)$

Si l'on connait $\varphi(module)$ en plus de **module**, trouver les facteurs premiers de ce dernier relève de la résolution d'une simple équation. En effet on rappelle que :

$$\varphi(module) = (p-1)(q-1)$$

En développant, on obtient :

$$\varphi(module) = pq - (p+q) + 1$$

On peut donc à partir de là extraire (p+q). Et comme on connait déjà pq= module, on sait que p et q sont les solutions de l'équation suivante :

$$x^2 + (p+q)x - pq = 0$$

Calculer l'indicateur d'Euler d'un entier n'est pas en soit un exercice compliqué.

Soient $n \in \mathbb{N}$ et $n = \prod_i p_i^{\alpha_i}$ sa décomposition en facteurs premiers avec les α_i des entiers naturels. L'indicateur d'Euler de n répond aux formules suivantes dont la première est un conséquence immédiate des propritétés de l'indicateur mises en exergue dans le chapitre 2.2 :

$$\varphi(n) = \prod_i (p_i - 1) p_i^{\alpha_i - 1}$$

et

$$\varphi(n) = n \prod_i (1 - \frac{1}{p_i})$$

Seulement, il y a un problème. Il s'agit ici de calculer l'indicateur d'Euler de **module** sans connaître sa factorisation. Autant dire que ceci est une autre paire de manche. Au vu des formules de calcul connues pour l'indicateur d'Euler, on imagine bien que trouver $\varphi(module)$ est au moins aussi difficile que de factoriser **module**. Nous venons de pointer du doigt une deuxième "limite technique" qu'exploite le système RSA.

Nous connaissons maintenant ces "petites choses" qui font de RSA un système sur, ou peut-être pas. Il est important d'émettre une petite réserve sur cette liste de "clés de sureté" de RSA. En effet, il a été **CONJECTURE** que la sécurité de RSA dépendait du problème de factorisation(auquel nous avons associé le calcul de l'indicateur d'Euler). Cependant il n'a pas été prouvé mathématiquement que la factorisation de **module** était la seule voie vers la découverte de **clair**. On précise quand même ici que nous sommes dans un contexte d'utilisation avec toutes les précautions connues et prescrites.

Ainsi donc, on ne peut pas omettre l'hypothèse qu'un autre moyen d'arriver de calculer **clair** soit découvert ;moyen qui ne passe pas par la factorisation de **module**.

Mais rassurons-nous, pour le moment ce moyen n'est pas disponible(encore une fois dans le cadre d'une utilisation très précautionneuse).

Voici enfin venu le moment d'éclaicir(ou du moins de tenter d'éclaicir) le point sur les fameuses conditions optimales d'utilisation de RSA.

Chapitre 4

RSA : les précautions d'utilisation

Plusieurs décisions lourdes et importantes pèsent sur les épaules de D1. Elles concernent la sélection de certains nombres et pas d'autres. Sans surprise, la moindre maladresse est potentiellement un risque de mise à nu du message en clair.Passons ici en revue quelques faux pas liés au choix des clés et "faciles" à éviter.

4.1 Méthode de génération de p et q

Le chapitre précédent a fait un arrêt sur la méthode de Lucas pour la génération de grands nombres premiers. Pour bien faire les choses avec RSA, il est en fait recommandé d'éviter ce genre de méthodes constructives. En effet si D1 utilise une telle méthode et qu'elle est connu d'un intrus, il est possible à ce dernier d'arriver à retrouver p et q en essayant de déterminer la fonction qui remplace le choix aléatoire et l'ordre de grandeur de **module**.

A la place, il est préférable de vraiment choisir un nombre au hasard et de tester sa primalité. Il a été montré que si l'on prend un nombre de taille n au hasard, la probabilité qu'il soit premier est de l'ordre $\log 2^n$.

On ne peut résister à la tentation d'aborder (dans la mesure du possible) dans cette partie, un autre test de primalité qui est fortement conseillé car très efficace : **le test de Miller-Rabin**

Test de Miller-Rabin

Ce test se base sur deux propriétés. La première est donnée par le petit théorème de Fermat que nous rappelons :

Théorème *Soit n un nombre premier et a un entier tel que PCDG(a,n)= 1.Alors on a*

$$a^{n-1} \equiv 1 \pmod{n}$$

Soit donc $(n, a) \in \mathbb{N}^2$ tel que $1 < a < n$. On déduit du théorème que si : $a^{n-1} \neq 1 \pmod{n}$ alors n est nécessairement composé, c'est-à-dire que n n'est pas un nombre premier.

La réciproque de cette propriété est cependant fausse. En effet il existe une infinité de nombres composés n qui vérifient

$$a^{n-1} \equiv 1 \pmod{n} \; \forall a \in \mathbb{N} \; tel \; que 1 < a < n$$

. Ces nombres sont appelés **nombres de Carmichael**. Leur existence justifie l'utilisation d'une seconde propriété pour le test de Miller-Rabin.

Voici donc la seconde base de notre test de primalité.

Propriété *Soit $n \in \mathbb{N}$. Si n est premier alors les racines carrées de 1 modulo n sont 1 et -1(c'est-à-dire n-1).*

Essayons de le montrer.
Soit donc (n,r) $\in \mathbb{N}^2$ tel que $1 < r < n$ et n premier. On suppose que

$$r^2 \equiv 1 \pmod{n}$$

Ceci entraine que :

$$r^2 - 1 \equiv 0 \pmod{n}$$

On a donc que :

$$n \mid (r+1)(r-1)$$

n étant premier en on déduit que :

- $n \mid (r+1)$ (1)
  ou(non exclusif)
- $n \mid (r-1)$ (2)

Ceci nous permet de conclure. En effet :

$$(1) \Rightarrow r \equiv -1 \pmod{n}$$

et

$$(2) \Rightarrow r \equiv 1 \pmod{n}$$

Le principe du test de Miller-Rabin est le suivant. On cherche à savoir si n est un nombre premier. On tire au hasard un entier a $\in [2, n-1]$ et on calcule $a^{n-1} \pmod{n}$ en utilisant un algorithme de calcul qui procède par carrés successifs. Ceci permet de vérifier si l'on a une racine carré non triviale(c'est-à-dire différente de 1 et de -1) de l'unité.Si tel est le cas on sait que n est composé.De même si l'on trouve que $a^{n-1} \neq 1 \pmod{n}$ on peut conclure que n est composé. Lorsque n est déclaré composé, il est inutile de continuer.Si à après cette première étape n est supposé premier, on recommence avec un autre a tiré aléatoirement, et ainsi de suite.

Le nombre de test que l'on fait subir à n dépend bien sûr de son statut à la fin de chaque test(supposé premier ou déclaré composé)mais aussi de la probabilité avec laquelle on veut qu'il soit supposé premier.

4.2 Ordre de grandeur de $|p - q|$

Il a été répété bien des fois que p et q doivent être **suffisament grands**. On juge s'ils le sont ou pas en fonction de l'importance de la communication et des moyens dont disposent les potentiels violeurs du secret.Juste pour le rappel dans notre cas il s'agit des autres participants à la conférence de messagerie instantanée. Ainsi **suffisament grands** n'aura pas le même sens dans notre cas et dans le cas de communication entre membres de services secrets sur canal à haut risque par exemple.

Cela ne nous dit toujours pas quelle doit être la taille de p par rapport à celle de q.

On peut déjà rayer une option qui est de prendre l'un des deux **sensiblement** plus grand que l'autre. En effet si X1 procède ainsi,la factorisation de n est menacée de mise à nu par tout algorithme dont la complexité dépend de la taille du plus petit facteur premier(par exemple les divisions successives).

Bien sûr cela fait quand même un bon paquet de calculs mais qui se réduisent très vite quand le degré de la maladresse(c'est-à-dire l'écart de taille entre p et q) s'agrandit. En bref, la difficulté de factorisation de **module** se retrouve ébranlée par un tel choix de ses deux facteurs.

Toutefois, une fois cette erreur évitée, il ne faut pas non plus tomber dans le piège inverse à savoir prendre p et q trop proches. Supposons que D1 fasse cette seconde erreur. On a alors :

$$p \approx q \approx \sqrt{module}$$

Une telle configuration peut être très dangereuse car un attaquant qui tente une factorisation en recherchant des facteurs premiers autour de *sqrtmodule* peut en venir à bout dans un temps disons raisonnable.

De façon plus formel,il a été montré que D1 doit dans son choix de grands nombres premiers respecter la condition suivante :

$$|p - q| > \sqrt[4]{module}$$

Le choix de p et q est également soumis à d'autres précautions liées à la menace que représente certains algorithmes de factorisation dont nous n'avons pas parlé. Nous n'entrerons donc pas dans le détail pour ce qui est des motivations de ces précautions supplémentaires.

4.3 "Force" de p et q

Il est recommandé que p et q soient des **nombres premiers forts**. Mais qu'est-ce qu'un nombre premier fort ?

Définition *Soit **prime** un nombre premier.**prime** est un **nombre premier fort** s'il vérifie :*

- *prime -1 a un grand facteur premier que l'on note r*
- *prime +1 a un grand facteur premier*
- *r a un grand facteur premier*

Une fois les deux grands nombres premiers choisis dans les règles de l'art, D1 doit encore choisir deux exposants dont l'un sera diffusé(**exppu**) et l'autre tenu secret(**exppr**). C'est la taille de ce dernier qui peut éventuellement poser un problème dont nous allons parler maintenant sans entrer dans les détails ; la théorie sous-jacente dépassant le niveau du TIPE.

4.4 Ordre de grandeur de l'exposant secret

On se base en fait sur le théorème suivant que nous ne démontrerons pas :

Théorème *Soient N,e,d $\in \mathbb{N}$ et p et q deux nombres premiers. Si les conditions suivantes sont vérifiées :*

- $N = pq$ *avec* $q < p < 2q$
- $e < \varphi(N)$
- $ed \equiv 1 \pmod{\varphi(N)}$
- $d < \frac{1}{3}\sqrt[4]{N}$

Alors on peut calculer d et factoriser N.

Lorsque les conditions du théorème sont remplies avec N= **module**, e=**exppu** et d=**expr** on peut factoriser **module**. On parle alors **d'attaque de Wiener**.

Concrètement, cette attaque consiste à retrouver l'exposant secret **exppr** à partir du développement en fractions continues de $\frac{exppu}{module}$.

Il est important de noter toutefois que un choix aléatoire **exppu** et **exppr** est en quelque sorte une protection contre l'attaque de Wiener. Ceci découle du fait qu'en mode aléatoire, les exposants choisis remplissent rarement les conditions du théorème.

Chapitre 5

Cryptanalyse de RSA

Il s'agit dans cette partie en réalité de pointer du doigt d'autres erreurs à ne pas commettre dans une utilisation de RSA. Cependant contrairement à celles dont il a été question dans le chapitre 4, celles-ci ne sont pas liées, ou du moins pas directement aux clés en elle-même(p,q,module, exppu, exppr).

Admettons que D1 participe à plusieurs conférences de messagerie instantanées dont l'intersection des ensembles de participants est différente du singleton D1 ;Disons pour simplifier que D1 participent à deux conférences. Dans chacune d'elles, il a un interlocuteur avec lequel il souhaite avoir une communication codée ; disons E1 dans la première conférence et E2 dans la deuxième. On admet aussi que les deux communications codées portent sur le thème(on n'en a pas vraiment besoin). Ceci pour se permettre de supposer par la suite que E1 et E2 souhaitent envoyer le même message à D1.

D1 pour une raison quelconque peut être tenté d'uliser le même **module** pour E1 et E2. Nous allons voir pourquoi ceci ne serait pas très prudent.

5.1 La faille du module commun

Notons **exppu1** l'exposant public relatif à la communication entre D1 et E1 et **exppu2** celui relatif à l'échange entre D1 et E2. On rappelle que ces deux exposants on été choisis aléatoirement à partir du même **module** en suivant la démarche que nous connaissons. Il est important de noter que le caractère aléatoire du choix entraîne que la probabilité que l'on ait PCDG(exppu1,exppu2)= 1 est assez forte pour se permettre de le supposer. Souffrez que cette affirmation ne soit pas suivi d'une preuve. On considère donc que **exppu1** et **exppu2**. Le théorème de Bézout nous dit que $\exists (r,s) \in \mathbb{Z}^2$ tels que :

$$r * exppu1 + s * exppu2 = 1 \qquad (a)$$

Dans cette égalité l'un des deux termes de la somme est forcément négatif sinon cela voudrait dire que les deux termes de la somme sont inférieurs à 1. On prend arbitrairement $r * exppu1 < 0$. Ceci entraîne que $r < 0$ car $exppu1 \in \mathbb{N}$

Jusque là, on ne voit toujours pas où est le problème. E1 et E2 envoient à D1 respectivement **crypte1** et **crypte2**. Tout s'assombrit si **crypte1** et **crypte2** correspondent au même **clair**. C'est-à-dire si :

$$crypte1 \equiv clair^{exppu1} \pmod{module} \qquad (b)$$

et

$$crypte2 \equiv clair^{exppu2} \pmod{module} \qquad (c)$$

Avant de montrer où se situe le problème, procédons à une petite manipulation de signe.

On a que $r < 0$. On rappelle que **crypte1** est premier avec **module**. On en déduit que **crypte1** est inversible dans $\mathbb{Z}/module\mathbb{Z}$. Si on note $crypte1^{-1}$ son inverse,alors on a l'égalité suivante modulo **module** : $crypte1^{-1*(-r)} = crypte1^r$

(b) et (c) entraînent que :

$$crypte1^r \equiv clair^{exppu1*r} \pmod{modulo}$$

et

$$crypte2^s \equiv clair^{exppu2*s} \pmod{modulo}$$

On a par suite que :

$$crypte1^r * crypte2^s \equiv clair^{exppu1*r+exppu2*s} \pmod{modulo} \qquad (d)$$

(a) et (d) entrainent que :

$$crypte1^r * crypte2^s \equiv clair \pmod{modulo}$$

Prenons maintenant E1 comme acteur principal. E1 doit envoyer un même message **clair** à D1,D2 et D3 qui ont tous suivi la démarche décrite au chapitre 1 pour choisir clé publique et clé privée.Soit i $\in \{1, 2, 3\}$, on note (modulei,exppui) la clé publique relative à Di et **cryptei** le message chiffré relatif à Di. On suppose que

$$exppu1 = exppu2 = exppu3 = exppu$$

On suppose en plus que **exppu** est très petit. Nous verrons que dans de telles conditions le message en clair peut être retrouvé.

5.2 Attaque de Hastad

Avant d'aller plus loin, rajoutons une condition à savoir que les **modulei** sont premiers entre eux deux à deux.

La situation décrite quelques lignes plus haut nous donne :

$$cryptei \equiv clair^{exppu} \pmod{modulei}$$

De là, la symétrie de la relation de congruence permet d'écrire le système suivant :

$$\begin{cases} clair^{exppu} \equiv crypte1 \pmod{module1} \\ clair^{exppu} \equiv crypte2 \pmod{module2} \\ clair^{exppu} \equiv crypte3 \pmod{module3} \end{cases}$$

Le théorème des restes chinois nous dit qu'il existe des solutions x à ce système. ces solutions(et donc $clair^{exppu}$) vérifient :

$$x \equiv \prod_{i=1}^{3} cryptei \pmod{\prod_{i=1}^{3} modulei}$$

exppu étant très petit, on peut supposer que $clair^{exppu}$ est l'unique solution qui vérifie

$$x < \prod_{i=1}^{3} modulei$$

De là, on peut retrouver **clair** avec d'un calcul de racine cubique.Autant dire que ceci est loin d'être un casse-tête.

Il a été vu dans le chapitre 3 que dans de bonnes conditions d'utilisation, il n'était pour l'instant pas "raisonnable" d'envisager une attaque par factorisation de **module**. Rassurez-vous, les choses n'ont pas encore changé depuis lors. Nous allons juste maintenant voir de façon très brève(parce qu'il n'en faut pas plus pour le dire) comment à partir de la factorisation de **module**, on pouvait casser le système.

5.3 Attaque par factorisation

Reprenons notre tout premier exemple de communication entre E1 et D1. On admet donc qu'un intrus réussisse à factoriser **module**. Il a donc connaissance de p et q. De façon immédiate, il obtient $\varphi module$. On rappelle maintenant que **exppu** et **exppr** sont inverses dans $\mathbb{Z}/\varphi(module)\mathbb{Z}$. L'attaquant à partir de **exppu** peut donc retrouver **exppr** très aisément en utilisant l'algorithme d'Euclide étendu.

Avant de mettre fin à notre TIPE, on peut faire une dernière remarque concernant cette partie. Certains messages dits invariants pourraient poser un problème. Ce sont des **clair** tels que :

$$clair^{exppu} \equiv clair \pmod{module}$$

Et on imagine très facilement pourquoi ils sont pourraient être gênants. Cependant en pratique dans une communication, ils sont noyés dans la masse des informations qui circulent.

Conclusion

Nous arrivons au terme de notre TIPE qui s'était donné pour mission de mettre en lumière le fonctionnement du système RSA mais aussi d'expliquer les éléments qui en faisaient un outil sûr. Nous espérons avoir mené à bien cette mission qui s'est avérée tout à fait passionnante à exécuter.

Il est plutôt aisé de comprendre le fonctionnement du système RSA. C'est peut-être l'une des raisons de sa célébrité. Ne dit-on pas que les meilleures choses sont toujours les plus simples.Toutefois, derrière cette simplicité d'exécution, et c'est çà le plus beau, se trouve un arsenal solide qui permet au monde de RSA de tourner correctement. On devine cependant que comme toute chose, RSA

est loin de la perfection. Nous n'avons pas parlé de ses défauts,ce n'était pas notre objectif. Mais il pourrait être intéressant de relever que il est loin d'être l'algorithme de chiffrement le plus rapide.

A côté de cela, il est important de poser une réserve sur la fiabilité de RSA car l'histoire a déjà prouvé que les exploits d'hier peuvent vite être éclipsés par les avancés de demain et qu'il ne faut jamais s'arrêter aux acquis d'un système. On peut penser à ENIGMA qui était à son époque un géant incassable et qui a fini avec le statut de belle archive de la cryptographie. Bien sûr, avec le temps, les progrès attendus sont de plus en plus complexes mais l'on peut faire confiance à la science(et à travers elle à tous les chercheurs et futurs chercheurs) pour poursuivre le chemin. Déjà on parle depuis un certain temps de cryptographie quantique. Cette méthode basée sur la polarisation de la lumière pourrait vraisemblablement, une fois aboutie, porter un coup à bien des sytèmes déclarés aujourd'hui fiables, RSA compris.

Tout laisse penser que la grande histoire de la cryptologie n'a pas fini de nous étonner et connaîtra bien de RSA. Dans tous les cas, ce système restera ce qu'il est aujourd'hui malgré ses défauts : un incroyable produit de l'ingéniosité humaine.

Références

1. Gilles Dubertret, *Initiation à la cryptographie*, 3eme édition, Vuibert
2. Bruce Schneier, *Cryptologie appliquée : algorithme, protocoles et codes source en C*, 2eme édition, Vuibert

Il faut également noter les cours de

1. François-Xavier Roblot *Factorisation des entiers*
2. Pierre Lavaurs *Groupe*
3. Paul Zimmermann *Introduction à la cryptologie*
4. Pierre Rouchon *Arithmétique et test de primalité*