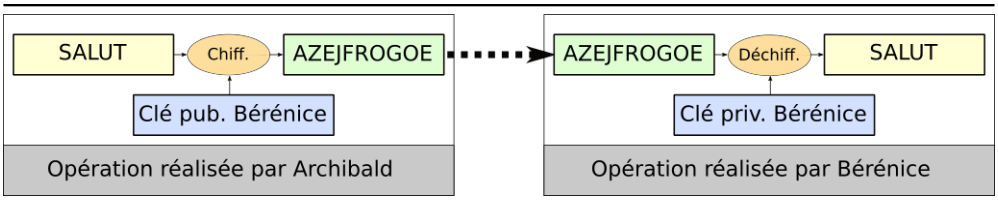
Cryptographie asymétrique

Laurent Signac – CC-BY-SA – 19-02-22 0622 f2ef32607a11ee76eb96

1 Chiffrement asymétrique

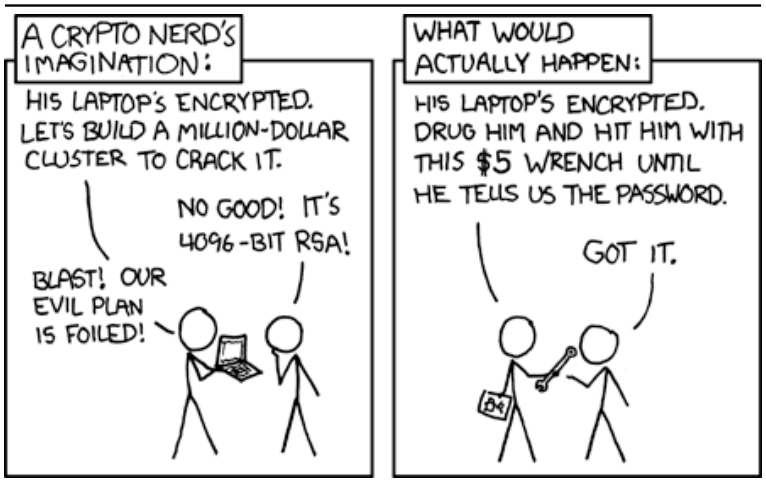
L'idée du chiffrement asymétrique a été proposée par Diffie et Hellman en 1976 (bien que probablement, d'autres aient eu l'idée avant, sans la publier). Dans ce type d'algorithme, chaque intervenant possède deux clés. La clé publique, utilisée généralement pour le chiffrement, et que tout le monde connaît, et la clé privée utilisée pour le déchiffrement, que seul son propriétaire connaît.

Bien entendu, les deux clés sont liées, mais il doit être très difficile d'obtenir la clé privée à partir de la clé publique et le chiffrement choisi doit être résistant à une attaque sur texte en clair choisi (puisque la clé de chiffrement est publique).



Pour qu'Alice envoie un message à Bob, seul Bob a besoin d'une paire de clés

Si le principe était établi, il restait à trouver un système utilisable en pratique.



Xkcd : http://xkcd.com/538/

2 RSA

L'algorithme à clé publique le plus utilisé est Rsa (du nom de ses auteurs, Ronald Rivest, Adi Shamir et Leonard Adleman qui l'ont découvert en 1977). Sa mise en œuvre sur des exemples jouets (avec de petits nombres) est très simple :

- choisir deux nombres premiers (normalement très grands) : p et q
- rechercher un nombre e qui soit premier avec $(p-1)$ et $(q-1)$ et donc avec $z = (p-1)(q-1)$.
- calculer d tel que $ed = 1 \mod z$) (possible puisque e est premier avec z)

Les nombres e et $n = pq$ constituent la clé publique, et les nombres d et n la clé privée. Nous admettrons que, connaissant e et n, on ne peut pas retrouver d autrement qu'en retrouvant les entiers p et q (pour calculer z). Or, il n'existe pas de méthodes efficace pour retrouver la décomposition en facteurs premiers de n et le calcul est impraticable si n est très grand (quelques milliers de chiffres binaires).

Pour chiffrer un nombre P (avec $P < n$), on calcule simplement $C = P^e \mod n$, ce qui est facile connaissant e et n, mais très difficile à inverser… sauf dans certains cas particuliers (on dit que la fonction utilisée est une fonction à sens unique et à brèche secrète). En effet, la fonction réciproque est simplement $P = C^d \mod n$, mais son calcul implique de connaître d, qu'on ne peut pas déduire de e et n… à moins de trouver la décomposition en facteurs premiers de n. Finalement, la sécurité de RSA repose sur cette difficulté. Actuellement, les meilleurs algorithmes de factorisation on une complexité sous-exponentielle, qui les rend impraticables dans les cas utilisés pour RSA.

Malgré les nombreux avantages du système RSA (plus de problème d'échange de clés secrètes), le système est trop lent pour être utilisé de façon complètement transparente. Il est donc essentiellement utilisé pour deux choses : la signature numérique, et l'échange de clé secrètes pour pouvoir utiliser dans la suite un chiffrement symétrique (comme AES, bien plus rapide)

2.1 Preuve du déchiffrement

Dans les conditions précédentes, montrons que $P^d = M^{ed} = M \mod n$.

□ Si M est premier avec n :
Puisque d est l'inverse de e modulo $z = \varphi(n)$, alors $\exists k, ed = 1 + k\varphi(n)$. Donc $M^{ed} = M \times (M^{k\varphi(n)}) \mod n$.
Si M est premier avec n, le théorème d'Euler[1] indique que $M^{\varphi(n)} = 1 \mod n$. Donc finalement $M^{ed} = M \mod n$

□ Si M n'est pas premier avec n :
Si M n'est premier ni avec p, ni avec q, alors M est un multiple de n et $M = 0 \mod n$. Dans ce cas, évidemment $M^{ed} = M = 0 \mod n$.
Si M est un multiple p et pas de q (le raisonnement est le même si c'est l'inverse), rappelons que $\varphi(n)$ est le nombre d'entiers strictement inférieurs à n premiers avec n. Comme p et q sont premiers et que $n = pq$, $\varphi(n) = \varphi(p)\varphi(q)$. Donc, puisque $ed = 1 \mod \varphi(n)$, alors $ed = 1 \mod \varphi(q)$. Donc, comme précédemment (théorème d'Euler), $M^{ed} = M \mod q$. Comme par ailleurs M est un multiple de p, $M = 0 \mod p$, et donc $M^{ed} = M \mod n$.

3 Certificats - Authentification des clés publiques

La cryptographie à clé publique pose un problème si une personne C mal intentionnée donne sa propre clé publique en prétendant qu'elle appartient à B. La personne A pourrait alors utiliser cette clé pour communiquer avec B, ne sachant pas que C peut intercepter les messages, les lire et les chiffrer avec la véritable clé publique de B afin que celui-ci ne se doute de rien. L'attaque qui utilise ce principe s'appelle *Man in the middle*.

Il existe donc un moyen d'« authentifier » une clé publique et de garantir le fait qu'elle appartient bien à telle personne : c'est la notion de certificat. Les certificats font appel à une organisation tierce digne de confiance. Cette organisation, en possession de la clé publique et de l'identité du détenteur lui remet un certificat. Ce certificat contient la clé publique et le nom du détenteur, signés numériquement par le tiers de confiance. La signature numérique garantit que le certificat n'est pas un faux, et a bien été délivré par l'autorité de certification. Il permet donc de vérifier que la clé publique appartient bien à telle personne. La certification des clés par un tiers de confiance est généralement un service payant.

Les formats de certificats sont par exemple X.509 ou OpenPGP.

[1] https://fr.wikipedia.org/wiki/Th%C3%A9or%C3%A8me_d%27Euler_(arithm%C3%A9tique)