Cryptographie symétrique et asymétrique

Entre les deux extrêmes que nous venons de voir (code de césar d'un côté et le one-time pad). Divers mécanismes ont été développés.

Crypto système à clé secrète (cryptographie symétrique)

Une clé secrète est partagée entre toutes les personnes qui doivent communiquer.

Les systèmes historiques correspondent à cette catégorie, mais aujourd'hui, on a également des manières plus sécurisées telles que AES.

Bien que ce crypto système ait été le standard pendant plusieurs siècles, il a quelques problèmes.

Par exemple, les clés doivent être distribuées et rester sûr, si la clé est compromise, tous les messages sont compromis. Et si une clé différente est utilisée par paire d'utilisateur, le nombre de clés nécessaires pour rester sûre devient très élevé.

Crypto système à clé publique (cryptographie asymétrique)

À la place d'avoir une clé secrète partagée par tout le monde, on va avoir une clé qui ne peut faire que du chiffrement (la clé publique), et une clé de déchiffrement (la clé privée).

La clé publique, comme son nom l'indique, peut être partagée partout. De cette manière, si on veut communiquer avec 100 personnes, à la place d'avoir 100 clés, on va avoir une seule clé publique partagée partout.

N'importe qui peut chiffrer un message avec cette clé publique, mais seule la clé privée pourra permettre de la déchiffrer.

Cela règle donc les problèmes de la cryptographie asymétrique, cependant ce système a le désavantage d'être beaucoup plus lourd et de demander beaucoup de calcul (ce qui est la raison pour laquelle il est si récent).

Il reste tout de même un problème, celui de la confiance. Comment pouvons-nous être sûrs que la personne qui donne la clé privée est bien qui elle prétend être ?

Pour cela, on peut utiliser des tiers de confiance déjà connus à l'avance qui pourront certifier des clés (c'est par exemple le cas de l'entreprise Let's Encrypt qui permet de certifier les clés TLS pour le HTTPS).

RSA

> Pour apprendre et comprendre le fonctionnement de RSA, allez voir cette playlist, pour avoir des détails sur le calcul uniquement, vous pouvez consulter cette vidéo et si vous voulez utiliser quelque chose de plus simple que l'algorithme d'Euclide étendu, vous pouvez utiliser le théorème de Bachet-Bézout à la place.

Voici comment calculer les clés publiques et privées en RSA,

1. Tout d'abord, on choisit deux nombres premiers, que l'on va appeler $ p $ et $ q $. Par exemple $ p = 3 $ et $ q = 5 $.
2. Ensuite, on fait le produit de ces deux nombres, que l'on va appeler $ n $, soit $ n = pq = 3 * 5 = 15 $
3. Ensuite, on calcule la fonction phi tel que $ \Phi (n) = (p - 1)(q - 1) = (3 - 1)(5 - 1) = 2 * 4 = 8$
4. Ensuite, on choisit un entier $ e $ dont le PGCD avec $ \Phi (n) $ vaut 1; autrement dit, il faut trouver un nombre $ e $ tel que $ e $ et $ \Phi (n) $ soient premiers entre eux (aucun facteurs premiers communs)
5. Enfin, il faut trouver le nombre de déchiffrement $ d $ tel que $ ed \mod \Phi (n) = 1 $, soit $ ed - kn = 1 $.
   - On peut ici utiliser le théorème de Bachet-Bézout qui dit que si deux nombres ($ a $ et $ b $) sont premiers entre eux, alors on peut trouver des entiers $ x $ et $ y $ tel que $ ax + by = 1$. Ici, on a $ e $ et $ \Phi (n) $ qui sont premiers entre eux, par conséquent on peut appliquer l'algorithme. En considérant $ x $ comme étant $ d $ et $ y $ comme étant $ k $. Note: si la valeur de d est négative on peut faire $ d + \Phi (n) $ pour avoir une valeur positive utilisable.
6. La clé publique est $ { e, n } $ et la clé privée est $ { d, n }$.

On peut donc maintenant chiffrer un message $ m $ (qui doit être inférieur à $ n $) en faisant $ m^e \mod n = c $. Et on peut déchiffrer un message en faisant $ c^d \mod n = m $. $

De même on peut signer un message en "déchiffrant" un texte en clair, $ m^d \mod n = s $ et on peut le vérifier en "chiffrant" la signature, $ s^e \mod n = m $.

Les chapitres ci-dessous sont optionnels pour le cours, mais aident à mieux comprendre le fonctionnement de RSA.

Exponentiation modulaire

Pour pouvoir avoir une clé pour déchiffrer et une clé pour chiffrer, il faut pouvoir trouver un moyen de faire une opération facilement (chiffrement avec clé secrète) mais de rendre l'opération inverse très compliquée (déchiffrement) si on ne connait pas une valeur supplémentaire (clé privée).

Cette fonction pour RSA c'est l'exponentiation modulaire, l'idée est que si on fait $ m^e \mod n = c $, cela demande beaucoup d'essai-erreur pour pouvoir en partant de e, n et c revenir à m.

Cependant, si on a un autre exposant (d) on peut l'inverser simplement en faisant $ c^d \mod n = m $.

Si on applique e et d ne même temps, le message ne change donc pas, ainsi $ m^{ed} \mod n = m $, cela sera important pour plus tard.

Factorisation de nombres premiers

Maintenant, il faut trouver un moyen de trouver e, d et n de manière à rendre tout cela possible. Pour cela, il faut trouver une autre fonction qui simple à faire dans un sens et compliquée à faire dans l'autre.

Cette fonction dans RSA c'est la factorisation de nombres premiers (pour rappel, un nombre premier est un nombre qui ne peut être divisé entièrement que par un ou lui-même). On sait que tous les nombres ont exactement une factorisation de nombres premiers, cependant, cette factorisation de plus en plus compliquée en fonction de la grandeur du nombre.

Cette propriété fait que la factorisation est un très bon candidat, car si on utilise des nombres premiers assez grands, il sera impossible de le factoriser avec nos moyens actuels.

Ainsi, on peut trouver deux nombres premiers très grands et les multiplier ensemble. Le produit de ces deux nombres premier sera très simple à calculer, mais très difficile à inverser parce que la multiplication est simple, mais la factorisation est elle très complexe.

Maintenant, il faut trouver une fonction qui dépend de la connaissance de la factorisation de n.

Indicatrice d'Euler, fonction Phi

Cette fonction, c'est indicatrice d'Euler que l'on va ici appeler phi. Ainsi la fonction $ \Phi (n) $ donne le nombre d'entiers positifs plus petit que $ n $ qui ne partagent pas de facteurs premiers avec $ n $. Par exemple $ \Phi (8) = 4 $ car huit ne partagent pas de facteurs communs avec 1, 3, 5 et 7, mais partagent des facteurs communs avec 2, 4 et 6.

Cette fonction est donc très compliquée à calculer pour des grands nombres, mais vraiment simple à calculer pour des nombres premiers. Puisqu'un nombre premier ne peut être divisé que par 1 ou lui-même, la fonction phi revient à dire $ \Phi (n) = n - 1 $.

De même la fonction est multiplicative, donc si $ a $ et $ b $ sont premiers, $ \Phi (a*b) = (a-1)(a-1) $.

Il faut maintenant trouver un moyen de lier la fonction phi à l'exponentiation modulaire.

Théorème d'Euler

le théorème d'Euler indique que $ a^{\Phi (n)} \mod n = 1 $ si a et $ \Phi (n) $ sont premiers entre eux.

On sait qu'un exposant peut être multiplié par n'importe quel nombre car cela ne changera pas le résultat du modulo. Donc $ a^{k \Phi (n)} \mod n = 1$ est vrai également. Une autre propriété est que si on incrémente l'exposant, alors cela devient $ a^{k \Phi (n) + 1 \mod n = a $.

Cela donne donc une forme similaire à $ m^{ed} \mod n = m $. On peut donc en déduire que $ ed = k \Phi (n) + 1 $, que l'on peut reformuler sous la forme $ 1 = ed + k \Phi (n) $, ou $ 1 = ed \mod \Phi (n) $.

Il suffit alors de trouver une valeur e qui soit premier avec $ \Phi (n) $ et trouver *d* en utilisant l'algorithme d'Euclide étendu ou le théorème de Bachet-Bezou.

Revision #4
Created 6 January 2024 19:09:20 by SnowCode
Updated 8 January 2024 22:42:57 by SnowCode