[CRYPTOGRAPHIE]
UN CHIFFREMENT À CLÉ PUBLIQUE :
LE PROTOCOLE D'ÉCHANGE DE CLÉS DE DIFFIE ET HELLMAN (1976)

Notions réinvesties : congruences, exponentiation modulaire

Parallèlement à leur découverte du principe de la cryptographie à clé publique, Diffie et Hellman ont proposé en 1976 un protocole d'échange de clés totalement sécurisé.
Le problème est le suivant : Alice et Bob veulent s'échanger un message crypté en utilisant un algorithme nécessitant une clé K. Ils veulent s'échanger cette clé K, mais ils ne disposent pas de canal sécurisé pour cela. Le protocole d'échange de clés de Diffie et Hellman répond à ce problème lorsque K est un entier.

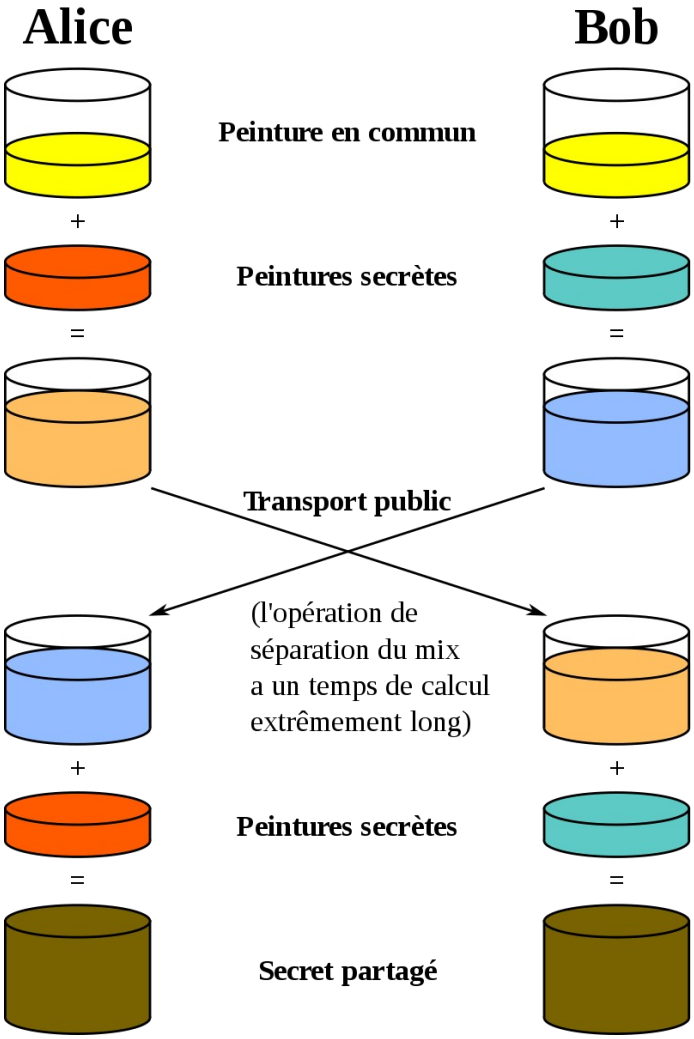

Illustration conceptuelle d'un échange de clés Diffie-Hellman (Wikipédia)

Whitfield Diffie

Martin Hellman

Ce protocole repose sur l'arithmétique modulaire et sur le postulat suivant :

Étant donné des entiers p (p premier), a (avec $1 \leqslant a \leqslant p-1$), x :
- il est facile de calculer l'entier y tel que $y \equiv a^x [p]$
- si on connaît a, p et $y \equiv a^x [p]$, il est très difficile de retrouver x, pourvu que p soit assez grand.

Retrouver x s'appelle résoudre le ***problème du logarithme discret***, pour lequel on ne dispose pas d'algorithme efficace.

Voici comment Alice et Bob font pour s'échanger la clé secrète.
Ils font des actions en parallèle, que l'on décrit dans le tableau suivant :

	Alice	Bob
Étape 1	Alice et Bob choisissent un nombre premier p et un entier a tel que $1 \leqslant a \leqslant p-1$. L'échange n'est pas sécurisé.	
Étape 2	Alice choisit secrètement un nombre x_1 .	Bob choisit secrètement un nombre x_2 .
Étape 3	Alice calcule y_1 tel que : $y_1 \equiv a^{x_1} [p]$.	Bob calcule y_2 tel que : $y_2 \equiv a^{x_2} [p]$.
Étape 4	Alice et Bob s'échangent les valeurs de y_1 et y_2 . L'échange n'est pas sécurisé.	
Étape 5	Alice calcule la clé secrète $y_2^{x_1} [p]$	Bob calcule la clé secrète $y_1^{x_2} [p]$.

1. Démontrer que $y_2^{x_1} \equiv y_1^{x_2} [p]$.

2. On souhaite appliquer ce protocole avec les clés suivantes :
- clés publiques : $p = 81\,629$ et $a = 65\,127$.
- clés privées : $x_1 = 12\,111\,985$ et $x_2 = 29\,051\,994$.

a) On cherche y_1 tel que : $y_1 \equiv 65\,127^{12\,111\,985} [81\,629]$.
Comment pourrait-on calculer y_1 (à la main) ? (on ne demande pas de faire le calcul)

b) Une méthode plus rapide existe, il s'agit de ***l'exponentiation modulaire rapide***.
La fonction Python `pow(a, e, n)` permet de calculer avec cette méthode a^e modulo n.
Déterminer alors y_1 , y_2 et la clé secrète K. Vérifier que Bob et Alice ont bien la même clé secrète.

À la fin du protocole, Alice et Bob sont donc en possession d'une même clé secrète K, qu'ils ne se sont pas échangés directement. Si quelqu'un a espionné leurs conversations, il connaît p, a, y_1 et y_2 .
Il ne peut pas retrouver K comme le font Alice ou Bob, car il lui manque toujours l'une des informations nécessaires, à savoir x_1 ou x_2 . Et il ne peut pas retrouver x_1 connaissant a, p et $y_1 \equiv a^{x_1} [p]$, puisque la résolution du logarithme discret est un problème difficile.

Il faut toutefois que p soit bien choisi et que les nombres utilisés soient suffisamment grands pour éviter une attaque par recherche exhaustive.
À l'heure actuelle, un nombre premier p de l'ordre de 300 chiffres ainsi que x_1 et x_2 de l'ordre de 100 chiffres sont tout simplement impossibles à casser même avec les meilleurs algorithmes de résolution

du logarithme discret. Cette impossibilité n'est pas théorique, mais algorithmique : ce n'est pas faisable en un temps réaliste actuellement. Si une solution pratique pour résoudre un logarithme discret venait à apparaître, d'autres systèmes cryptographiques pourraient tomber, notamment le système de ElGamal, qui repose sur le même principe.

Un inconvénient majeur de cette méthode et qu'elle ne permet pas de signer les messages…
En effet, ***ce protocole est vulnérable à « l'attaque de l'homme du milieu »***, qui implique un attaquant capable de lire et de modifier tous les messages échangés entre Alice et Bob : ils croient ainsi avoir échangé une clé secrète alors qu'en réalité ils ont chacun échangé une clé secrète avec l'attaquant, l'homme du milieu.

Le mécanisme de couplage d'appareils **Bluetooth** repose sur l'échange de clés à courbe elliptique Diffie-Hellman (ECDH) qui est un protocole d'échange de clés anonyme qui permet à deux pairs, chacun ayant un couple de clé privée/publique basé sur les courbes elliptiques (hors programme), d'établir un secret partagé à travers un canal de communication non sécurisé. Ce secret partagé peut être employé directement comme une clé de chiffrement ou être utilisé pour dériver une autre clé qui, à son tour, peut être utilisée pour chiffrer les communications. Il s'agit d'une variante du protocole d'échange de clés Diffie-Hellman, aussi utilisée dans des **messageries chiffrées** comme **Telegram**.

À ce sujet des messageries sécurisées, je vous déconseille fortement d'utiliser celles de **Messenger**, **Instagram** ou **WhatsApp** (toutes les trois propriétés de Facebook !), la moins mauvaise des trois étant tout de même WhatsApp. D'autres messageries sont à privilégier, tout aussi bien faites mais sécurisées : **Signal** par exemple (conseillée par Snowden), ses deux seuls défauts étant d'être sous juridiction américaine (d'une part, avec le CLOUD Act de 2018, l'État a accès à ce qu'il veut, d'autre part Signal pourrait être compromis soit par une porte dérobée dans ses systèmes, soit par une ordonnance gouvernementale l'obligeant à assister la NSA) et d'être centralisé (les infos chiffrées passent par un serveur intermédiaire, celui de Signal… si celui-ci venait à être piraté… c'est peu probable mais quand même).

En début d'année 2020 est sortie l'application **Olvid**, messagerie sécurisée française (bien meilleure juridiction !) et décentralisée, une grande nouveauté. À priori, c'est donc la meilleure messagerie sécurisée… mais elle n'est pas open source (même si auditée) et ne sera gratuite que pendant un temps seulement, semble-t-il, alors que Signal est gratuite (pour ce qu'on en fait). À suivre…

Si Signal vous intéresse, je vous conseille cet article, puis de l'installer.

Allez aussi faire un tour sur le site d'Olvid.

Et il existe aussi **Wire**, excellente messagerie au même niveau que Signal (auparavant un poil meilleure, car sous juridiction suisse, mais les USA sont passés par là et ont racheté une partie de l'entreprise).
Il existe aussi ce tableau comparatif des 12 messageries les plus connues : vous y verrez que Skype, Instagram etc. sont des horreurs.