Niveau : 2 **Orange**

Échange de clé de Diffie-Hellman

1 Principe de l'échange de clé de Diffie-Hellman

L'échange de clé de Diffie-Hellman a été développé par ces deux auteurs en 1976 et publié dans l'article : W. Diffie and M.E. Hellman, New directions in cryptography, IEEE Transactions on Information Theory 22 (1976), 644-654.

Nous rappelons ici que l'échange d'une clé secrète est fondamental en cryptographie. En effet tout chiffrement d' une grande quantité de données ne peut se faire qu'avec du chiffrement à clé secrète, surtout si cet échange a lieu en temps réel, en raison de la lenteur relative des chiffrements à clé publique.

Il s'agit donc, comme il est exigé par de nombreux protocoles, d'échanger entre deux interlocuteurs A et B une clé secrète K de taille t octets. Pour cela A et B disposent d'un groupe cyclique fini G et d'un générateur a de ce groupe (les éléments de G sont donc, si on note multiplicativement l'opération du groupe, $1, a, a^2, \cdots, a^{s-1}$ où s est l'ordre de G). Prenons par exemple pour G le groupe multiplicatif $(\mathbb{Z}/p\mathbb{Z})^*$ où p est un grand nombre premier et a un élément générateur de ce groupe (mais ça pourrait être aussi un générateur d'un grand sous-groupe de $(\mathbb{Z}/p\mathbb{Z})^*$).

Voici comment se passe (de manière schématique) l'échange. Les calculs indiqués sont faits dans le groupe G, donc dans notre exemple modulo p.

- Données publiques : le groupe $G = (\mathbb{Z}/p\mathbb{Z})^*$, un générateur a de ce groupe, un générateur de masque h.
- A tire au sort un entier n tel que $1 < n < p-1$ et le garde secret.
- A envoie a^n à B (calcul fait dans le groupe, donc ici modulo p).
- B tire au sort un entier m tel que $1 < m < p-1$ et le garde secret.
- B envoie a^m à A.
- A calcule $s = (a^m)^n$.
- B calcule $s = (a^n)^m$.
- A et B disposent maintenant même s

Rappelons que les calculs sont faits modulo p bien sûr. Le nombre s est à peu près de la taille de p et doit certainement être adapté à la taille de la clé commune convoitée. Ceci est fait grâce au générateur de masque :

$$K = h(s, t)$$

où t est la taille en octets de la clé secrète cherchée K.

2 Sur quoi repose la confidentialité de la clé construite

La robustesse du système repose à première vue sur la difficulté du problème du logarithme discret. En regardant plus en détail, on peut préciser plus. En fait l'attaquant, connaissant a^n et a^m dans le groupe G ne doit pas pouvoir reconstituer a^{mn}. Ce problème est connu sous le nom de problème de Diffie et Hellman calculatoire (CDH). Bien entendu, si on sait résoudre le problème du logarithme discret (DLP), qui consiste à retrouver n si on connaît l'élément a^n du groupe G, alors on sait aussi résoudre le problème CDH. Dans l'autre sens nous n'avons pas de résultat aussi fort. Mais il est prouvé dans une série d'articles de Ueli Maurer puis de Ueli Maurer et Stefan Wolf, que dans de nombreux cas, on dispose d'une réduction polynômiale du problème DLP au problème CDH. Par exemple dans le papier de Maurer et Wolf intitulé : The Relationship Between Breaking the Diffie-Hellman Protocol and Computing Discrette Logarithms (SIAM Journal on Computing, vol 28, n.5, pp. 1689-1721, 1999).

3 La mise en place

3.1 Cas du groupe multiplicatif $(\mathbb{Z}/p\mathbb{Z})^*$

Pour utiliser ce système on doit donc construire des groupes multiplicatifs $(\mathbb{Z}/p\mathbb{Z})^*$, c'est-à-dire des nombres premiers p ainsi que dans chaque cas un élément générateur du groupe (ou tout au moins un générateur d'un grand sous groupe. Cette construction est expliquée dans la fiche "fichecrypto_ 111" (construction de nombres premiers). Le protocole SSH par exemple utilise des nombres de Sophie Germain pour construire les éléments recherchés.

3.2 Cas du groupe des points d'une courbe elliptique

L'utilisation des courbes elliptiques sera étudiée dans une fiche ultérieure. Il suffit de savoir pour le moment que l'échange de clé de Diffie-Hellman peut aussi se réaliser avec le groupe des points d'une courbe elliptique où le problème du logarithme discret est difficile. L'intérêt est justement que pour le moment on ne connaît aucun algorithme sous-exponentiel pour résoudre le problème du logarithme discret sur un groupe bien choisi de ce type. Bien choisi veut dire que les courbes utilisées doivent répondre à un certain nombre de conditions, et en particulier ne pas être supersingulières. La taille des clés s'en ressent (par exemple 160 bits au lieu de 1024 bits dans le cas $(\mathbb{Z}/p\mathbb{Z})^*$).

4 L'attaque de l'homme au milieu

L'échange de clé de Diffie-Hellman est sensible à l'attaque de l'homme au milieu. Cette attaque permet à un attaquant actif O de s'intercaler dans la communication entre A et B et de créer avec A une clé commune, de faire de même avec B. Ainsi A et B pensent communiquer directement alors qu'en réalité, chacun communique avec O. C'est donc une attaque qui exploite le défaut d'identification de A vis à vis de B et de B vis à vis de A.

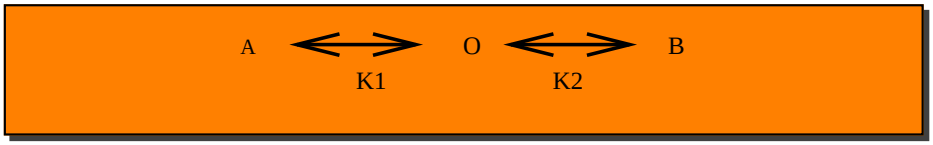

FIG. 1 – Attaque de l'homme au milieu

5 Exemples de protocoles intégrant l'échange de clé de Diffie-Hellman

5.1 Station to station protocol

Le protocole **Station to Station** (STS) consiste à combiner l'échange de clé de Diffie-Hellman avec de la signature à clé publique, assurant ainsi l'authentification mutuelle des deux interlocuteurs de manière à éviter l'attaque de l'homme au milieu.

Les deux interlocuteurs A et B disposent des fonctions cryptographiques suivantes :

- Chacun dispose d'un générateur pseudo-aléatoire.
- Ils disposent en commun d'un système de chiffrement à clé secrète dont la fonction de chiffrement est $\mathcal{E}$ et la fonction de déchiffrement est $\mathcal{D}$.
- Ils disposent en commun d'un groupe cyclique G et d'un générateur α de ce groupe (par exemple un $\mathbb{Z}/p\mathbb{Z})^*$ où p est un grand nombre premier). On suppose que dans cette configuration le problème CDH est difficile.
- L'interlocuteur A dispose d'un système de signature à clé publique constitué d'une fonction de signature $\mathcal{S}_1$ d'une fonction de vérification de la signature $\mathcal{V}_1$ ainsi que d'une clé privée d_A et d'une clé publique e_A.
- De même B dispose de $\mathcal{S}_2$, $\mathcal{V}_2$, d_B, e_B.
- Les deux intrlocuteurs disposent en commun d'un générateur de masque F.

Voici comment se passe l'échange.

- A tire au hasard un entier $0 < m < card(G)$, calcule α^m et l'envoie à B.
- B tire au hasard un entier $0 < n < card(G)$. Il calcule la clé secrète à échanger $K = F\left((\alpha^m)^n\right) = F\left(\alpha^{mn}\right)$. Puis B signe la concaténation de α^m et α^n : $s_B = \mathcal{S}_2(d_B, \alpha^m||\alpha^n)$, et enfin il envoie $(\alpha^n, \mathcal{E}(K, s_B))$ à A.
- A calcule $K = F\left(\alpha^{mn}\right)$, calcule $s_B = \mathcal{D}\left(K, \mathcal{E}(K, s)\right)$, vérifie la signature s_B avec la clé publique de B : $\mathcal{V}_2(e_B, \alpha^m||\alpha^n, s_B)$, et si la signature est correcte A authentifie B. Alors A signe $\alpha^n||\alpha^m$: $s_A = \mathcal{S}_1(d_A, \alpha^n||\alpha^m)$, et envoie : $\mathcal{E}(K, s_A)$ à B.
- B déchiffre ce dernier envoi : $s_A = \mathcal{D}\left(K, \mathcal{E}(K, s_A)\right)$, puis vérifie la signature de A : $\mathcal{V}_1(e_A, \alpha^n||\alpha^m, s_A)$, et si cette vérification est positive, la clé secrète K est échangée avec succés.

5.2 Secure Shell (SSH)

Nous verrons dans une fiche ultérieure l'utilisation de l'échange de clé de Diffie-Hellman pour créer un canal sécurisé dans le protocole SSH. Dans ce protocole il est conseillé d'utiliser des nombres de Sophie Germain, et un élément qui est soit primitif (ordre $p-1$), soit d'ordre $q = (p-1)/2$.

Auteur : Ainigmatias Cruptos
Diffusé par l'Association ACrypTA