TD de Cryptographie

Simon Abelard

Février, 2026

Exercice 1 : Un protocole (trop ?) simple (source : TD de C. Ritzenthaler)

En s'inspirant de Diffie-Hellman, on imagine le protocole suivant : pour envoyer un message x à Bob, Alice calcule $A_1 = x \oplus a$, avec a sa clé secrète. Bob envoie alors $B = A_1 \oplus b$, avec b sa clé secrète. Alice renvoie alors $A_2 = B \oplus a$ à Bob.

1) Faites un schéma représentant le protocole et justifiez pourquoi Bob est capable de retrouver x.

2) Comparez ce protocole à Diffie-Hellman en expliquant les similitudes, les différences, et en comparant les performances des deux protocoles.

3) On suppose qu'Oscar a vu passer tous les messages entre Alice et Bob (il connaît donc A_1, B et A_2). Montrez qu'Oscar arrive à calculer x. Concluez sur l'intérêt du protocole.

Exercice 2 : Signature El Gamal (adapté du TD de C. Ritzenthaler)

On rappelle le principe du schéma de signature d'El Gamal, qui devrait vous rappeler fortement le schéma ECDSA :

- Les paramètres sont g un élément primitif de $G = \mathbb{F}_p^*$ (i.e. g engendre tout le groupe G).
- Alice génère sa clé secrète en choisissant a au hasard dans l'intervalle $[1, p-2]$. Sa clé publique est $b = g^a \bmod p$.
- **Signature :** Si Alice veut signer y, elle calcule d'abord un haché $m = H(y)$, elle génère un nonce $k \in [1, p-2]$, elle calcule $r = g^k \bmod p$ puis $s = k^{-1}(m - ra) \bmod (p-1)$ et renvoie (y, r, s).
- **Vérification :** Bob vérifie que $0 \leq r \leq p-1$, calcule $m = H(y)$, $v = g^m \bmod p$ et $w = b^r r^s \bmod p$. Si $v = w$, Bob considère que la signature est authentique.

1) Pourquoi faut-il empêcher Alice de choisir $a = 0$ ou $a = p-1$? Quelle autre condition doit-on imposer sur le tirage de a ?

2) Rappelez les critères pour qu'une signature numérique soit sécurisée.

3) Vérifiez que si la signature est légitime, on a bien $v = w$.

4) Si un attaquant veut imiter la signature d'Alice, quel élément doit-il calculer ? Proposez une attaque permettant de faire ceci et évaluez sa complexité en fonction de p.

5) Donnez une taille minimale de p pour assurer un niveau de sécurité de 128 bits. En réalité, p est de l'ordre de plusieurs milliers de bits. Qu'est-ce qui peut expliquer cela ?

Le rôle du nonce k

6) Expliquez pourquoi il ne faut pas utiliser deux fois le même k pour signer deux messages distincts. Indice : regardez comment est calculé s.

7) Un ingénieur propose d'utiliser un compteur pour s'assurer de ne jamais réutiliser k. Est-ce une bonne idée ?

Pourquoi vérifier $r < p$?

On considère le scénario suivant. Oscar connaît une signature valide (y, r, s) qu'il a obtenue d'Alice. Il aimerait signer un message y' en se faisant passer pour elle.

— Il calcule $u = H(y')H(y)^{-1} \bmod (p-1)$

— Il calcule $s' = su \bmod (p-1)$

— Il résout les équations modulaires suivantes : $r' = ru \bmod (p-1), \quad r' = r \bmod p$.

8) Montrer qu'alors (y', r', s') sera bien une signature valide.

9) Comment s'y prend-on pour résoudre l'équation en r', quelle condition faut-il assurer pour qu'il y ait forcément une solution ?

10) Montrer que si $H(y) \neq H(y')$, alors le fait de vérifier $r' < p$ permet de ne pas se faire piéger.

11) Pourquoi peut-on supposer que $H(y) \neq H(y')$ en pratique ?

Quid de la fonction H

Ici, on suppose que le message m n'est pas haché et qu'il est directement utilisé dans le calcul de s.

12) On va supposer que $r = g^i b^j$, avec i et j des entiers compris entre 0 et $p-2$. Montrez que la vérification $v = w$ est vraie si et seulement si $g^{y-is} = b^{r+js} \bmod p$.

13) Un moyen d'avoir cette égalité serait simplement de faire en sorte que chacun des termes soit égal à 1. Justifiez pourquoi cela revient à demander $y - is = r + js = 0 \bmod p - 1$.

14) Donnez une condition pour garantir l'existence de i et j satisfaisant cette condition, et exprimez le couple (i, j) en fonction de (r, s, y).

15) Comparez cette attaque avec la précédente : d'après vous, laquelle des deux a l'impact le plus important ?