Sécurité des communications informatiques
TD#2

2024–02–02/09

Exercice 1 : Attaque *man-in-the-middle* pour DH

Le but de cet exercice est d'étudier la sécurité d'une instance quelconque du protocole d'échange de clef DH en présence d'un adversaire *actif*. Pour cela, on modélise (assez informellement) un tel adversaire en modifiant le jeu UP-EAV en UP-ACT de la façon suivante :

— à l'étape #2, l'adversaire peut modifier les messages échangés entre $\mathscr{A}$ et $\mathscr{B}$. On note $k_{\mathscr{A}}$ (resp. $k_{\mathscr{B}}$) le secret partagé obtenu par $\mathscr{A}$ (resp. $\mathscr{B}$) à l'issue de cette exécution.

— à l'étape #3, l'adversaire doit renvoyer $\hat{k}_{\mathscr{A}}$ et $\hat{k}_{\mathscr{B}}$ et gagne le jeu ssi. ces deux valeurs sont égales à $k_{\mathscr{A}}$ et $k_{\mathscr{B}}$ respectivement.

Q.1 :

1. Proposez une attaque « efficace » permettant à un adversaire de gagner le jeu UP-ACT avec probabilité 1.

Exercice 2 : Exponentiation rapide

On considère dans cet exercice un groupe cyclique $\mathbb{G}$ d'ordre N et l'un de ses générateurs g. L'objectif est de calculer le plus efficacement possible une exponentiation dans $\mathbb{G}$ (en base g) : étant donné $x \in [\![0, N-1]\!]$, on souhaite calculer g^x. On suppose pour cela que l'on dispose d'une représentation la plus compacte possible des éléments de $\mathbb{G}$ et d'une fonction `mulG` permettant de calculer le produit de deux éléments de $\mathbb{G}$ donnés dans cette représentation ; le coût de l'exponentiation se mesurera par le nombre de produits nécessaires dans son calcul.

Q.1 :

1. Spécifiez un algorithme effectuant une exponentiation dans $\mathbb{G}$ en $\Theta(N)$ produits.
2. Caractérisez le coût de cet algorithme (est-il linéaire, quadratique, exponentiel..?) en fonction de la taille de ses entrées.

Q.2 :

1. Spécifiez un algorithme effectuant une exponentiation dans $\mathbb{G}$ en $\Theta(\log(N))$ produits.
   INDICE : Utilisez l'écriture en base 2 de l'entrée x, et le fait que si $x = \sum_{i=0}^{n} x_i 2^i$, on a dans $\mathbb{G}$ l'égalité $g^x = \prod_{i=0}^{n} g^{x_i 2^i}$ (avec la convention qu'un produit vide vaut 1), ainsi que le fait que les puissances g^{2^i} peuvent se calculer facilement par mise au carré successives de g.
   EXEMPLE : $g^{257} = g \times g^{2^8} = g \times (((((((g^2)^2)^2)^2)^2)^2)^2)^2$.
2. Caractérisez le coût de cet algorithme (est-il linéaire, quadratique, exponentiel..?) en fonction de la taille de ses entrées.

Exercice 3 : DLOG, CDH, DDH

L'hypothèse du *logarithme discret* (DLOG) pour un groupe cyclique $\mathbb{G} = \langle g \rangle$ de taille N dit qu'étant donné g et $h := g^x$ avec $x \twoheadleftarrow [\![0, N-1]\!]$, il est « difficile » de calculer de le logarithme discret x de h en base g. On dit d'un adversaire capable d'effectuer ce calcul qu'il *casse* DLOG pour $\mathbb{G}$.

L'hypothèse du *problème de Diffie-Hellman calculatoire* (CDH) pour $\mathbb{G}$ dit qu'étant donné g, g^a, g^b avec $a, b \twoheadleftarrow [\![0, N-1]\!]$, il est « difficile » de calculer g^{ab}. On dit d'un adversaire capable d'effectuer ce calcul qu'il *casse* CDH pour $\mathbb{G}$.

L'hypothèse du *problème de Diffie-Hellman décisionnel* (DDH) pour $\mathbb{G}$ dit qu'il est « difficile » pour un adversaire de distinguer avec un « bon » avantage le quadruplet (g, g^a, g^b, g^{ab}) du quadruplet (g, g^a, g^b, g^c), avec $a, b, c \twoheadleftarrow [\![0, N-1]\!]$ On dit d'un adversaire capable d'effectuer ce calcul qu'il *casse* DDH pour $\mathbb{G}$.

Q.1 :

1. Montrez qu'un adversaire capable de casser DLOG pour un coût L et de calculer une exponentiation pour $\mathbb{G}$ pour un coût E est capable de casser CDH pour un coût $L + E$.
2. Montrez qu'un adversaire capable de casser CDH pour un coût C est capable de casser DDH avec avantage ≈ 1 pour un coût C.

Q.2 : Une hypothèse A est dite *plus forte* qu'une hypothèse B si casser B implique la possibilité de casser A avec un coût proche, mais que casser A n'implique pas nécessairement la possibilité de casser B avec un coût proche.

1. Ordonnez les hypothèses DDH, CDH et DLOG de la plus faible à la plus forte.
2. Expliquez le sens de cette terminologie.

Exercice 4 : *Baby-step/Giant-step*

On considère dans cet exercice un groupe cyclique $\mathbb{G}$ d'ordre N et l'un de ses générateurs g. L'objectif est de calculer un logarithme discret dans $\mathbb{G}$ (en base g) : étant donné g^x, $x \in [\![0, N-1]\!]$, on souhaite calculer x. On suppose pour cela que l'on dispose d'une fonction `mulG` et d'une fonction `expG` permettant de calculer le produit de deux éléments de $\mathbb{G}$ ainsi que l'exponentiation d'un élément de $\mathbb{G}$; le coût du calcul du logarithme discret se mesurera par la somme du nombre de produits et d'exponentiations.

Q.1 :

1. Spécifiez un algorithme calculant un logarithme discret dans $\mathbb{G}$ pour un coût $\Theta(N)$.

Q.2 :

1. Spécifiez un algorithme calculant un logarithme discret dans $\mathbb{G}$ pour un coût $\Theta(\sqrt{N})$.

   INDICE #1 : Il peut être utile dans une première phase de calculer (et stocker dans une structure de donnée appropriée) les valeurs $g^{\sqrt{N}}$, $g^{2\sqrt{N}}$, etc.

   INDICE #2 : ☼
2. Caractérisez le coût de cet algorithme (est-il linéaire, quadratique, exponentiel..?) en fonction de la taille (supposée minimale) de ses entrées.
3. Quelle taille *minimale* conseilleriez-vous de prendre pour $\mathbb{G}$, si celui-ci est utilisé dans un contexte cryptographique où le calcul du logarithme discret doit être « difficile » ?