Exercice 1

Le protocole de Diffie-Hellman

1. Alice et Bernard se mettent publiquement d'accord sur deux nombres α et p tels que $\text{PGCD}(\alpha, p) = 1$.
2. Chaqu'un de leur coté ils engendrent une clé secrète s (notée a pour Alice et b pour Bernard) et calculent
$k_s \equiv \alpha^s \bmod p$ avec $s = a$ pour Alice et $s = b$ pour Bernard
3. Alice communique publiquement k_a à Bernard qui lui communique k_b en retour.
4. Alice calcule $(k_b)^a \bmod p$ et Bernard fait de même avec $(k_a)^b \bmod p$.

1. Mettre en œuvre le protocole décrit ci-dessus avec votre voisin. Vérifier que les deux valeurs $(k_a)^b$ et $(k_b)^a$ sont bien congrue modulo n.
2. Démontrer que $(k_b)^a \equiv (k_a)^b \bmod p$ (on notera K cette valeur). En déduire que ce protocole permet la génération et l'échange sécurisé de clés secrètes (K ici).

Exercice 2

Cryptosystème RSA [DS 2010,5 pts]

On considère les nombres premiers $p = 73$, $q = 43$ et $e = 29$ qui permettent de constituer la partie publique (n, e) d'une clé RSA avec $n = p \times q$.

1. Chiffrer le message $x = 122$ avec la clé (n, e).
Indication : *Pour simplifier le calcul utiliser l'exponentiation modulaire rapide.*
2. Calculer $\Phi(n)$.
3. Calculer d, l'exposant de la clé privé (n, d).
Indication : *d est l'inverse de e modulo $\Phi(n)$, on doit don donner un résultat dans* $\{0; 1; \ldots; \Phi(n) - 1\}$
4. Déchiffrer le message $y = 121$ en utilisant la clé privé (n, d).
Indication : *Pour simplifier le calcul utiliser l'exponentiation modulaire rapide.*

Exercice 3

Attaques du cryptosystème RSA

Une entreprise utilise le protocole RSA pour des échanges sécurisés d'informations. Pour limiter le temps de calcul des clés publiques la méthode de génération suivante est utilisée :

- 50 nombres premiers (tous distincts) sont générés aléatoirement et stockés en mémoire,
- Pour chaque choix possible de 2 nombres premiers $p_i \neq p_j$ de la liste précédente
  — on calcule $n_{ij} = p_i p_j$ et $\Phi_{ij} = (p_i - 1)(p_j - 1) = \Phi(n_{ij})$
  — on génère aléatoirement un nombre e_{ij} premier avec Φ_{ij}
  — on calcule un nombre d_{ij} tel que $d_{ij} e_{ij} \equiv 1 \bmod \Phi_{ij}$

1. Combien de nombres n_{ij} (et donc de clés RSA (n_{ij}, e_{ij}, d_{ij})) a-t-on obtenu ?
2. Que vaut $\text{PGCD}(n_{ij}, n_{ik})$?
3. En déduire une attaque possible du système de chiffrement dans ce cas ?
4. Quel est l'intérêt de cette attaque par rapport à celle qui consiste à vouloir décomposer en facteurs premiers les n_{ij} indépendamment les uns des autres ?

Exercice 4

signature électronique via le RSA

Alice et Bernard communiquent via le cryptosystème RSA. Alice possède la clé secrète (n, d) et Bernard seulement la clé publique (n, e). On appellera f la fonction de chiffrement (que possède Bernard grace à (n, e)) et g la fonction de déchiffrement (que seule possède Alice grace à (n, d)).

$$\begin{array}{rl} \text{fonction de chiffrement} & f(x) = (x^e \bmod n) \\ \text{fonction de déchiffrement} & g(x) = \big(x^d \bmod n\big) \end{array}$$

Bernard veut poser une question à Alice sous forme d'un texte Q. Il envoie donc à Alice $C = f(Q) = Q^e \bmod n$. Alice décode la question en calculant $G(C) = C^d \bmod n$. Pour donner sa réponse Alice envoie à Bernard un texte en clair R et, pour que Bernard soit sûr que c'est bien Alice qui lui répond, elle lui envoie aussi le texte $g(R)$.

1. Quel calcul permet de retrouver R connaissant $g(R)$?
2. Pourquoi ce calcul permet-il à Bertrand d'authentifier l'auteur du message ?

Repères historiques : On entend par cryptosystème à clé publique un système de chiffrement/déchiffrement où le chiffrement reposerai sur une clé publique pouvant être divulguée (par exemple dans un annuaire ouvert à tout le monde) le déchiffrement reposant quant à lui sur une clé privée connue seulement de celui qui déchiffrement ! Le risque de voir la sécurité du cryptosystème mise à mal par un "transfuge" devient alors nulle. L'idée d'un cryptosystème à clé publique est apparue dans les années 60. les agences de renseignement Britannique focalisèrent leur efforts sur cette question vers 1969 (sous l'impulsion de Ellis) et la solution, trouvée en 1973 (par un certain Cocks), resta classée "secret défense". C'est en 1977 que trois Américains (Rivest, Shamir et Adleman) ont proposés la première mise en œuvre d'un cryptosystème à clé publique : Le RSA. Les Américains déposèrent un brevet et créèrent la société *R*SA security pour l'exploiter. Les Britanique ayant conservé tout le secret de leur découverte perdirent donc tout leur avantage dans l'exploitation commerciale de cette découverte technologique ...

Exercice 5

Codes correcteurs d'erreurs

On transmet des séries de données qui peuvent être représentées sous la forme :

$$x = (x_1, x_2, \ldots, x_{10}), \quad \forall i = 1; 2; \ldots; 10, \quad x_i \in \{0; 1; 2; \ldots 10\}$$

mais au cours de la transmission une ou plusieurs peuvent être altérées, c'est à dire que le récepteur reçoit le message :

$$\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{10}), \quad \forall i = 1; 2; \ldots; 10, \quad \tilde{x}_i \in \{0; 1; 2; \ldots 10\}$$

Pour déterminer quelles données sont érronées dans $\tilde{x}$ on ajoute au message des *données de contrôle* a et b :

$$(x_1, x_2, \ldots, x_{10}, a, b), \qquad \text{avec} \quad \begin{cases} a & \equiv & \sum_{i=1}^{10} x_i \bmod 11 \\ b & \equiv & \sum_{i=1}^{10} i \times x_i \bmod 11 \end{cases}$$

qui peuvent être aussi affectées d'une erreur (c'est à dire qu'on reçoit $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{10}, \tilde{a}, \tilde{b})$. On suppose que lors de la transmission une seule donnée est erronée, le but de cet exercice est de donner une méthode pour trouver la donnée erronée et la corriger. On notera

$$\Delta_i = \tilde{x}_i - x_i \quad \forall i = 1; 2; \ldots; 10,$$

1. On suppose que la donnée erronée est $\tilde{a}$ ou $\tilde{b}$
   (a) Si c'est $\tilde{a}$ montrer que $\tilde{a} \not\equiv \sum_{i=1}^{10} \tilde{x}_i \bmod 11$ *mais* $\tilde{b} \equiv \sum_{i=1}^{10} i \times \tilde{x}_i \bmod 11$
   (b) Si c'est $\tilde{b}$ montrer que $\tilde{a} \equiv \sum_{i=1}^{10} \tilde{x}_i \bmod 11$ *mais* $\tilde{b} \not\equiv \sum_{i=1}^{10} i \times \tilde{x}_i \bmod 11$
2. On suppose que la donnée erronée est un des $\tilde{x}_i$ (notée x_j)
   (a) Montrer que $\tilde{a} \not\equiv \sum_{i=1}^{10} \tilde{x}_i \bmod 11$ *et* $\tilde{b} \not\equiv \sum_{i=1}^{10} i \times \tilde{x}_i \bmod 11$
   (b) En déduire un test pour savoir si une des données $\tilde{x}_i$ transmises est erronée.
3. Dans le cas où la seule donnée erronée est $\tilde{x}_j$
   (a) Montrer que $\tilde{a} - \sum_{i=1}^{10} \tilde{x}_i \equiv \Delta_j \bmod 11$ *et* $\tilde{b} - \sum_{i=1}^{10} i \times \tilde{x}_i \equiv j \times \Delta_j \bmod 11$
   (b) En déduire que l'on corrige l'erreur en utilisant les formules :

$$\begin{cases} \Delta_j & \equiv & \tilde{a} - \sum_{i=1}^{10} \tilde{x}_i \bmod 11 \\ \Delta_j^{-1} & = & \text{inverse de } \Delta_j \text{ dans } \mathbb{Z}/11\mathbb{Z} \\ j & \equiv & \Delta_j^{-1} \times \left(\tilde{b} - \sum_{i=1}^{10} i \times \tilde{x}_i\right) \bmod 11 \\ x_j & \equiv & \tilde{x}_j + \Delta_j \bmod 11 \end{cases}$$

4. (a) On reçoit les séries de données (avec données de contrôles) suivantes :
   - $\tilde{x} = \begin{pmatrix} 5 & 4 & 2 & 6 & 4 & 10 & 0 & 5 & 2 & 4 \end{pmatrix}$, $\tilde{a} = 9, \tilde{b} = 1$
   - $\tilde{x} = \begin{pmatrix} 6 & 7 & 7 & 10 & 5 & 2 & 2 & 2 & 9 & 7 \end{pmatrix}$, $\tilde{a} = 5, \tilde{b} = 3$
   - $\tilde{x} = \begin{pmatrix} 3 & 10 & 2 & 3 & 3 & 3 & 6 & 5 & 3 & 6 \end{pmatrix}$, $\tilde{a} = 7, \tilde{b} = 1$

   retrouver les données envoyées en corrigeant si besoin les données reçues.
   (b) Peut on appliquer la même méthode avec une autre valeur que 11 ?