Sécurité des SI

TD (cryptographie)

1) Les fonctions de hachage permettent de construire des clés hierarchiques. Essayez de généraliser le schéma lorsque les niveaux de sécurité admettent un ordre partiel (certains niveaux ne peuvent pas être comparés).
2) Alice part en conférence à l'étranger et veut se connecter à l'INRIA pendant son séjour. L'INRIA exige qu'Alice utilise un nouveau password à chaque nouvelle connexion. Avant de partir, Alice choisit un nombre aléatoire w et hache cette valeur t+1 fois afin d'obtenir une valeur qu'elle donne au système. Elle garde tous les hachés sur une clé USB qu'elle emporte à la conférence avec elle. Sur le lieu de la conférence, Elle décide de se connecter, proposez un protocole qui lui permette de se connecter t fois avec à chaque fois un nouveau password.
3) quels sont les avantages et inconvénients du chiffrement asymétrique par rapport au chiffrement symétrique?
4) Vous utilisez un chiffrement symétrique avec des clés de 128 bits, quelle longueur de clé devez-vous choisir pour le chiffrement asymétrique (non elliptique)? Quelle fonction de hachage correspond à cette même sécurité?
5) Pourquoi le chiffrement doit-il être probabiliste? Donnez un exemple.
6) A quoi sert un certificat?
7) Quel est le principal intérêt de la cryptographie elliptique?
8) Alice et Bob partagent une clé secrète de longueur 300 bits et veulent s'envoyer des message chiffrés. Décrivez un protocole de chiffrement (en particulier le choix de l'algorithme de chiffrement et du mode de chiffrement).
9) Secret sharing scheme : Alice a un secret qu'elle ne veut pas garder sur elle. Elle décide de le diviser en fragments de secrets et distribue un fragment à chacun de ses quatre plus proches amis. Elle veut pouvoir reconstituer le secret avec trois fragments. Pour cela, elle a juste à contacter trois amis parmi les quatre et à partir des trois fragments reçus, utiliser l'interpolation de Lagrange pour obtenir le secret.
   Alice choisit p=7. Elle distribue un point à chacun de ses quatre amis.
   Alice distribue (1,4), (2,1), (3,2), (6,1)
   On rappelle que le secret est le terme constant du polynôme construit par Alice.
   Quel est le secret d'Alice ?
   Rappel : Si les points sont notés (xi, yi) le secret se retrouve par la formule

$$d_0 = \sum_{i=0}^{k} \left(\prod_{\substack{j=1 \\ j \neq i}}^{k} \frac{-x_j}{x_i - x_j} \right) y_i \pmod p$$

10) L'algorithme ElGamal utilise un groupe Z*p et un générateur g du groupe.
    1) Détaillez l'algorithme
    2) Comment trouver le générateur g facilement ?
11) Fault attack on RSA. L'algo de signature utilise souvent CRT pour aller plus vite (environ 4 fois plus rapide). Il s'agit de calculer Sp=c^d dans Zp et Sq=c^d dans Zq et d'en déduire S=c^d dans Zn (n=p.q). Un attaquant peut provoquer une faute lors du calcul de Sq (par exemple avec un laser). Comment, à partir de la valeur Sp et de la valeur fausse Sq, un attaquant peut-il en déduire la factorisation de n ?
12) Fonction d'accumulation. Un dictionnaire authentifié est une structure de données qui supporte les requêtes authentifiées d'appartenance. Par exemple, si Alice émet une

requête pour obtenir un document, elle le recevra (s'il existe) et obtiendra une preuve d'authenticité du document. Le dictionnaire pourrait signer tous les documents comme preuve d'authenticité. Mais cette solution est lourde si le nombre de documents est grand et que ceux-ci sont mis à jour très souvent. Une autre solution consiste à prouver que le document reçu fait bien partie du dictionnaire. Pour cela on peut utiliser une fonction d'accumulation. Soit n=p.q et g un élément de Z*n. Chaque document a un identifiant i et son haché est hi. L'entité gérant le dictionnaire va signer g^A, où A est le produit de tous les hachés. Lorsqu'un client souhaite obtenir le document j, il le reçoit accompagné de la valeur Pj=g^B, où B est le produit de tous les hachés (sauf sauf celui de j). Le client hache le document reçu calcule (Pj)^hj et compare la valeur à g^A.
Soit n=323, g^A=234. Le client demande le document doc1 de valeur de haché 47. Il le reçoit accompagné de la valeur 157.
1) L'authentification est-elle correcte ?
2) Quels sont les élément publics, secrets dans ce système pour obtenir une bonne sécurité ?
Cette solution est arithmétique et donc un peu lourde.
3) Existe-t-il une autre solution utilisant uniquement des fonctions de hachages et permettant de vérifier que le document reçu appartient bien au dictionnaire ?

Gestion de clés :

13) Donner les principales hierarchies de clés
14) La distribution de clés publiques doit-elle assurer l'intégrité ou la confidentialité?
15) Pour quelles raisons une clé de session doit-elle être éphémère?
16) Dans un protocole d'authentification, quelles sont les outils qui permettent d'éviter le rejeu?
17) Proposer un protocole de distribution de clé entre A et B sachant que A et B ont tous les deux une paire de clés publique/privée
18) proposer un protocole de distribution de clé entre A et B, par l'intermédiaire d'un TTP
19) Proposer une méthode pour renouveler une clé
20) Que faire si une clé est compromise?
21) A quoi sert un certificat? Existe-t-il des alternatives?

Authentification :
Citez quatre moyens d'authentifier l'identité d'un utilisateur
Citez et décrivez brièvement les principales menaces qui pèsent sur la confidentialité des passwords
Citez une technique pour protéger le fichier de passwords
Citez quatre techniques pour sélectionner ou assigner un password
Quel est le défaut d'un password aléatoire ?
Pourquoi utiliser du sel ?
Quel est l'intérêt que la taille du sel soit grande ? comment doit-on fixer cette taille ?
Dans le système Unix, comment faire pour retrouver un password perdu ?
Expliquez la différence entre une carte à mémoire et une smart-card
Quelles sont les principales méthodes biométriques pour l'identification ?
Quelles sont les différentes phases pour une identification biométrique ?

Kerberos :
Quelle sorte de cryptographie utilise Kerberos (sym/asym)? Pourquoi?
Comment l'authentification du client se fait-elle?
Expliquez les différentes étapes qui permettent d'accéder à un service

Le FIPS 186 décrit le standard de signature DSS.
Ecrire les algorithmes *Key generation* de génération de clef pour DSA, *Signature generation* et *vérification*.

Exemple de signature avec de (très) petits paramètres (notations de FIPS 186). Le résultat de la fonction de hachage H est donné (il ne s'agit pas ici de SHA).
Génération de clef : p=124540019, q=17389, h=110217528,
g=10083255, x= 12496, y=119946265, k=9557.
H(m)= 5246.
Calculer la signature de m.
Vérifier la signature.
Supposons que la signature soit (r=34, s=13049). Que donne la vérification ?