1. – Le cryptographie symétrique : La clef est secrète aux 2 extrémités (la même). Niveau sécurité, il n'y a pas de preuve formelle. Très rapide niveau performance. L'avantage par rapport à l'asymétrique est sa rapidité : 10-100 Mbits/s .
   – Le cryptographie asymétrique : Seule la clé privée est secrète, grand nombre de clefs dans un réseau, garantie de l'authenticité des clefs publiques. Sa sécurité repose sur la difficulté (supposée) de problèmes mathématiques. L'inconvénient par rapport au symétrique est sa lenteur : 10-100 Kbits/s.
2. Un cryptosystème est considéré sûr si un attaquant est incapable : 1) Trouver la clé 2) Trouver le message en entier 3) Extraire de « l'information » sur le message en clair ou la clé à partir des chiffrés.

Chiffrement de flux :

3. – Une fonction dite *aléatoire* est une fonction capable de produire une séquence de nombres dont on ne peut pas « facilement » tirer des propriétés *déterministes* ou les prédire, de façon que cette séquence puisse être appelée « suite de nombres aléatoires ».

   Des méthodes pour obtenir des nombres aléatoires existent depuis très longtemps et sont utilisées dans les jeux de hasard, comme : jet de dés, roulette, tirage au sort, mélange des cartes, etc. Elles peuvent toutefois souffrir (et souffrent généralement) de biais. Les fonctions génératrices de nombres aléatoire n'existent effectivement pas dans le monde informatique. On parle plûtot de nombre pseudo aléatoire qui s'approchent plus ou moins d'un aléa parfait. Un nombre pseudo aléatoire peut être définit comme une suite de bits de taille définit dont les propriétés de génération du nombre suivant semble aléatoire.

   – Dans les méthodes de chiffrement symétriques on voudrait produire une clé de chiffrement de telle maniere qu'elle soit *imprédictible* par un attaquant. Par conséquent, si la clé est parfaitement aléatoire, alors la complexité d'une attaque par recherche exhaustive est maximisée. Les nombres aléatoires sont omniprésents dans ce domaine. Le chiffrement par flot consiste à utiliser un XOR entre les données et une suite aléatoire. Par analogie, dans la cryptographie asymétrique, il est nécessaire de produire de grands nombres aléatoires avec des contraintes supplémentaires (premier, premier entre eux, etc.). De plus, un texte chiffré doit s'approcher le plus possible d'un contenu

aléatoire pour limiter les fuites d'information.

4. La connaissance du message chiffré n'apporte *aucune information* sur le message clair même en supposant un attaquant avec des capacités calculatoires illimitées.

   Tous les systèmes utilisés dans la pratique sont théoriquement *cassables*. À ce jour, seul le One Time Pad (ou le masque jetable) remplit cette condition sous des hypothèses strictes i.e. la clef secrète doit être aussi longue que le texte clair, en plus chaque clé, ou « masque », ne doit être utilisée qu'une seule fois. Si le masque a un contenu aléatoire et indépendant d'autres masques alors tous les textes clairs sont plausibles à partir du chiffré.

   Exemple : La probabilité d'un texte clair X sachant que le texte chiffré est Y est la meme probabilité de n'importe que X. Le texte chiffré n'apporte donc aucune information sur le texte clair.

   Un algorithme de chiffrement doit s'approcher au maximum de cette probabilité de 1/2.

5. On attaque un masque jetable en réutilisant la meme clé, démontré en cours. Des clés parfaitement aléatoires ne peuvent pas être produites par un ordinateur par un simple calcul : en effet ce calcul est déterministe, dont le résultat est totalement prévisible quand on connaît l'algorithme et les données initiales. Des algorithmes appelés *générateurs de nombres pseudo-aléatoires* sont utiles dans la cryptographie, mais n'atteignent pas la condition du parfait aléa, qui seule garantit la sécurité inconditionnelle démontrée par Claude Shannon.

6. Non, car on ne peut pas prouver qu'un générateur de nombres aléatoires est parfaitement et infiniment aléatoire (estimations heuristiques seulement).

Chiffrement par blocs :

1. Les deux principes fondamentaux sont : La confusion et la diffusion. La confusion vise à cacher n'importe quelle structure algébrique dans le système pour la rendre intelligible. La diffusion permet d'éparpiller la redondance du message i.e. permettre à chaque bit de texte clair d'avoir une influence sur une grande partie du texte chiffré. Ce qui signifie que la modification d'un bit du bloc d'entrée doit entraîner la modification de nombreux bits du bloc de sortie correspondant.

2. Les S-Boxes permettent de créer le principe de la confusion, donc de casser la linéarité de la structure de chiffrement.

Chiffrement DES

3. La fonction aléatoire est assurée dans le chiffement DES à travers les permutations aléatoires dans le fonction F du réseau Feistel (Details dans le cours).
4. Théorème de Sécurité des Réseaux Feistel (ou Luby-Rackof) : Si une fonction aléatoire sûre est utilisée pour trois tours de Feistel avec trois clés indépendantes, on obtient alors une fonction pseudo aléatoire avec des permutations pseudo aléatoire.
5. Par la recherche exhaustif car 2^{56} est très faisable par un ordinateur.
6. Attention la sécurité réelle d'un chiffrement DES est 2^{112} mais la sécurité effective est de 2^{57}.

   Explication : Au lieu d'attaquer naivement 2^{112} opérations par une recherche exhaustive des 2^{112} clés possibles. On peut l'attaquer par le milieu, aussi appelée la méthode compromis temps/mémoire :
   - En faisant la recherche exhaustif sur 2^{56} opération de clés possible.
   - Et en stockant en mémoire 2^{56} couples (clairs, chiffrés) puis en les triant suivant la valeur de chiffrés

     Pour déchiffrer la 2e étape, on applique à nouveau cette recherche en n opérations. Finalement, nous avons $2.n$ opérations, soit $2{\cdot}2^{56}$ dans le cas de DES. En conclusion, 2^{57} opérations sont nécessaires pour casser un tel chiffrement.

Chiffrement AES :

1. – **AddRoundKey :** XOR avec la sous-clé
   – **SubBytes :** passage dans une S-box
   – **ShiftRows :** décalage des ligne (rotation)
   – **MixColumns :** mélange des colonnes (sauf dernier tour).

   Ceci 10 à 14 fois (tours)
2. 10 tours pour AES 128 bits
3. Dans la S-Box du DES la sortie de 4 bits est obtenue à partir de l'entrée de 6 bits S. Alors que la S-box d'AES est une fonction fixe de 8 bits vers 8 bits (fonction bijective).
4. Attaques différentielle et linéaire.