Chapitre 3 : Chiffrement Symétrique & Asymétrique

3.1 Vocabulaire de base dans la cryptographie

Texte en clair : c'est le message à protéger (à chiffrer).
Texte chiffré : (cryptogramme) , c'est le résultat du chiffrement du texte en clair.
Chiffrement : c'est la méthode ou l'algorithme utilisé pour transformer un texte en clair en texte chiffré.
Déchiffrement : c'est la méthode ou l'algorithme utilisé pour transformer un texte chiffré en texte en clair.
Clé : c'est le secret partagé utilisé pour chiffrer le texte en clair en texte chiffré et pour déchiffrer le texte chiffré en texte en clair. On peut parfaitement concevoir un algorithme qui n'utilise pas de clé, dans ce cas c'est l'algorithme lui-même qui constitue la clé, et son principe ne doit donc en aucun cas être dévoilé.
Cryptosystème : algorithmes + clés
Cryptographie : cette branche regroupe l'ensemble des méthodes (algorithmes) qui permettent de chiffrer et de déchiffrer un **texte en clair** afin de le rendre incompréhensible pour quiconque n'est pas en possession de la clé à utiliser pour le déchiffrer.
Cryptanalyse : c'est l'art de révéler les textes en clair qui ont fait l'objet d'un chiffrement sans connaître la clé utilisée pour chiffrer le texte en clair.
Cryptologie : il s'agit de la science qui étudie les communications secrètes. Elle est composée de deux domaines d'étude complémentaires : la cryptographie et la cryptanalyse.

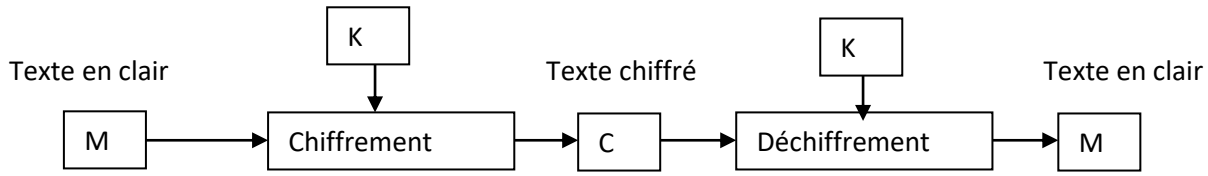


Fig 3.1 principe d'un cryptosystéme

En cryptographie, la propriété de base est que M = D(E(M)) où
– M représente le texte clair,
– C est le texte chiffré,
– K est la clé (dans le cas d'un algorithme à clé symétrique), Ek et Dk dans le cas d'algorithmes asymétriques,
– E(x) est la fonction de chiffrement, et
– D(x) est la fonction de déchiffrement.
Ainsi, avec un algorithme à clef symétrique, M = D(C) si C = E(M)

3.2 Chiffremnt symétrique

Dans la cryptographie symétrique, la clé de chiffrement est la même que la clé de déchiffrement. La clé est donc un secret partagé uniquement entre l'émetteur et le destinataire. Il existe plusieurs algorithmes de chiffrement symétrique : DES, RC4, RC5, Blowfish, IDEA, AES, ….

La cryptographie symétrique utilise deux types de chiffrement manipulant des mots binaires :

- Le chiffrement par bloc
- Le chiffrement par flot

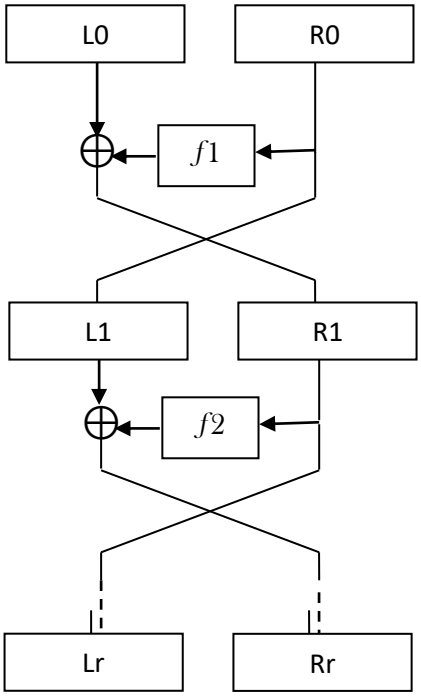
3.2.1 Le chiffremnt par bloc

Le chiffrement par bloc consiste en premier lieu à découper le message à chiffrer en bloc de taille fixe (exemple : 64 bits). Puis appliquer l'algorithme de chiffrement sur chaque bloc. On découpe le message M de n bits en s blocs de l = n/s bits (on ajuste initialement la taille du message en ajoutant des bits a 0 afin que sa taille soit un multiple de l.

3.2.1.2 Chiffrement de Feistel

Un schéma de Feistel est un chiffrement itératif par blocs transformant
$m= (L_0, R_0) \in F^m{}_2 \times F^m{}_2$ en $c= (L_{r-1}, R_{r-1}) \in F^m{}_2 \times F^m{}_2$ par une procédure de $r \geq 1$ tours de Feistel.
Chaque tour transforme (L_{i-1}, R_{i-1}) en (L_i, R_i) en utilisant une sous clés Ki, et une fonction de confusion f par : $\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus f(R_{i-1}, K_i). \end{cases}$ L'opération est inversible

$M= (L_O, R_O)$



Nous avons pour le chiffrement :

$$\begin{cases} L_1 = R_0 \\ R_1 = L_0 \oplus f(R_0) \end{cases}$$

La fonction f est appelée fonction de confusion

Pour le déchiffrement on a :

$$\begin{cases} L_{i+1} = R_i \\ R_{i+1} = L_i \oplus f(R_i) \end{cases}$$

Exemple avec le schéma de Feistel

Soit le message M= 101110
Et les fonctions $f1$, $f2$, et $f3$:

$f1$

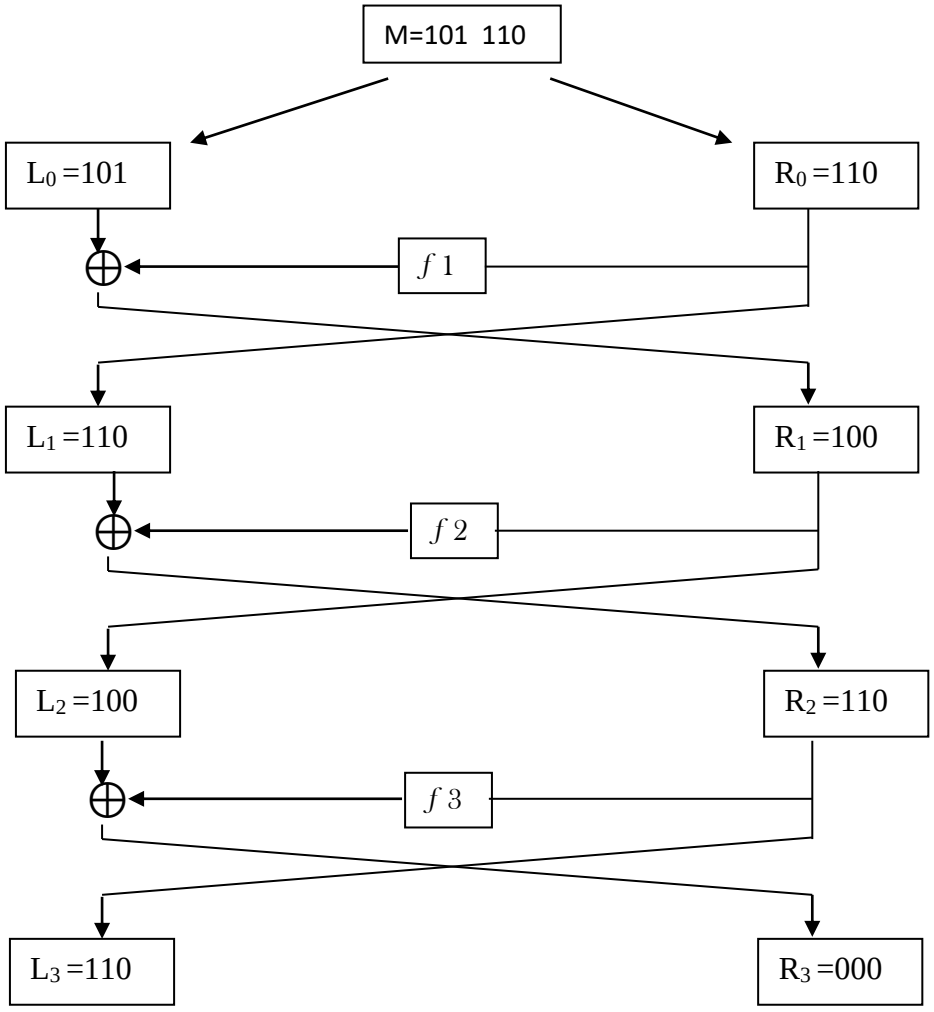
000	001	010	011	100	101	110	111
110	100	111	000	110	010	001	101

$f2$

000	001	010	011	100	101	110	111
010	011	110	111	000	101	110	110

$f2$

000	001	010	011	100	101	110	111
111	010	110	110	000	101	100	001



Le chiffré de m est donc C=110000

3.2.2.2 Modes d'opérations

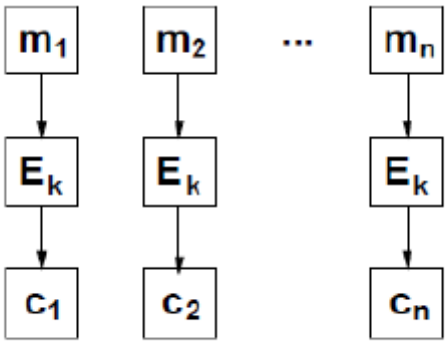
On distingue plusieurs modes d'opérations utilisés dans le chiffrement par bloc, en général ils sont basés sur les opérations suivantes :

- Permutation
- Substitution
- Opération XOR (Ou Exclusif)

Dont les plus courants sont :

- **Le mode Electronic Code Book (ECB)**

Dans ce mode, le message M est découpé en blocs ***mi*** de taille fixe.
Chaque bloc est alors chiffré séparément par une fonction ***Ek***, paramétrée par une clé ***k.*** Ainsi un bloc de message donné ***mi*** sera toujours codé de la même manière. Ce mode de chiffrement est le plus simple mais il est très vulnérable aux attaques. $C_i = E(M_i)$.
Pour le déchiffrement on inverse la fonction de codage: Dk= Ek
Déchiffrement : mi=Dk(ci)

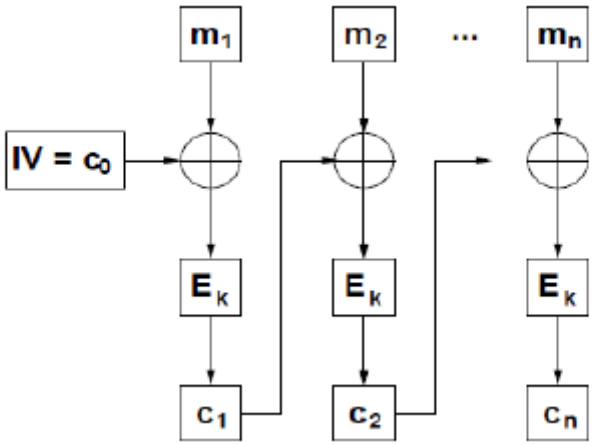


- **Le mode Cipher Block Chaining (CBC)**

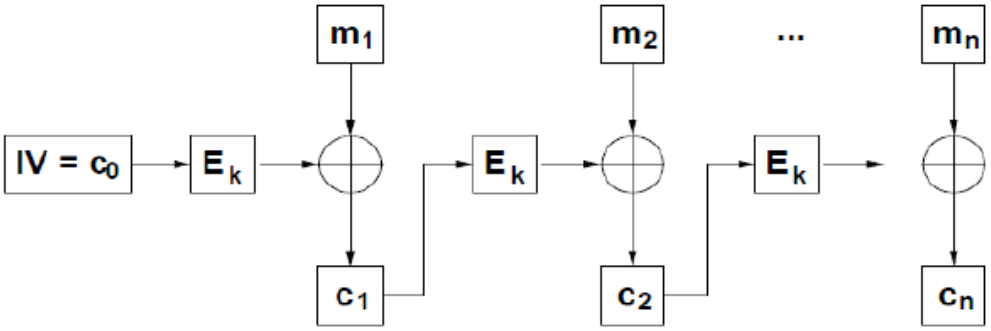
Le mode CBC a été introduit pour qu'un bloc ne soit pas codé de la même manière s'‟il est rencontré dans deux messages différents. Il faut ajouter une valeur initiale C0 aléatoire (ou IVpour « Initial Value »).
Chaque bloc est d'abord modifié par XOR avec le bloc chiffré précédent avant d'être lui même chiffré. CBC est le mode de chiffrement le plus utilisé.

Le chiffrement : Ci= $Ek(m_i \oplus C_{i-1})$
Le déchiffrement nécessite d'inverser la fonction de chiffrement : Dk= Ek

Déchiffrement : $mi = C_{i-1} \oplus Dk(C_i)$

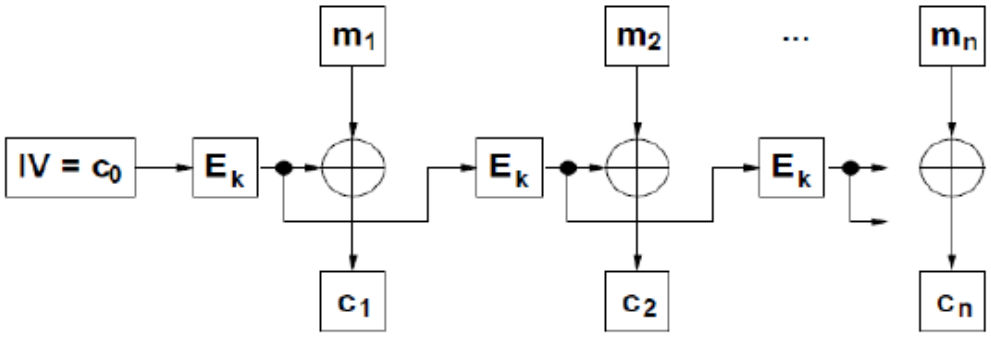


- **Le mode Cipher Feedback (CFB)**
  Dans ce mode le déchiffrement ne nécessite pas l'implémentation de la fonction inverse de la clé de chiffrement :



Le chiffrement est donné par : Ci =mi ⊕ Ek(ci-1)
Le déchiffrement est donnée par : mi =ci ⊕ Ek(ci-1)

- **Le mode Output Feedback (OFB)**



Dans ce mode le chiffrement est donné par :
r0=c0 ; r i = Ek(ri-1) ; Ci=mi ⊕ ri

Déchiffrement :
r i = Ek(ri-1) ; mi = ci ⊕ ri

3.3 Chiffrement Asymétrique ou à clé publique

- Chaque utilisateur dispose d'une paire de clé : une clé publique et une clé privée
- La clé publique est utilisée pour le cryptage.
- La clé privée est utilisée pour le décryptage.

Les algorithmes asymétriques possèdent 2 modes de fonctionnement :

- Le mode chiffrement dans lequel l'émetteur chiffre un fichier avec la clé publique du destinataire pour chiffrer. Le destinataire utilise sa clé privée pour déchiffrer le fichier. Dans ce mode, l'émetteur est sûr que seul le destinataire peut déchiffrer le fichier.

- Le mode signature dans lequel l'émetteur signe un fichier avec sa propre clé privée. Le destinataire utilise la clé publique de l'émetteur pour vérifier la signature du fichier. Dans ce mode, le destinataire est sûr que c'est bien l'émetteur qui a envoyé le fichier.

Pour résumer :

- L'émetteur chiffre avec la clé publique du destinataire, le destinataire déchiffre avec sa clé privée.
- L'émetteur signe avec sa clé privée, le destinataire vérifie la signature avec la clé publique de l'émetteur.

Quelques algorithmes de cryptographie asymétrique très utilisés:

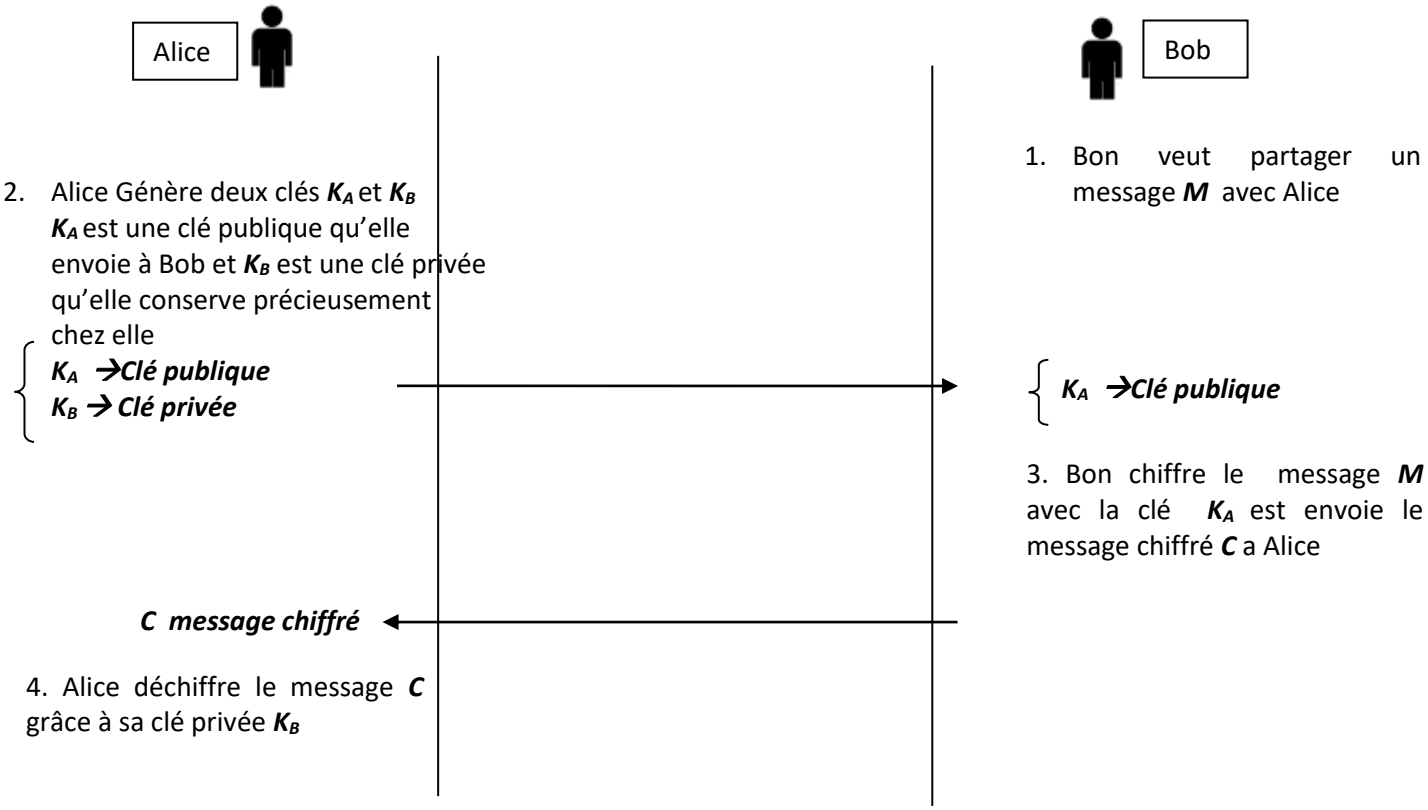
- RSA (chiffrement et signature).
- DSA (signature).
- Protocole d'échange de clés Diffie-Hellman(échange de clé)

3.4.1 L'algorithme RSA (Rivest, Shamir et Adleman, 1977)

3.4.1.1 Principe :

L'algorithme RSA utilise deux clés

1. une clé publique qui sert a chiffré
2. une clé privée qui sert a déchiffré



3.4.1.2 Génération des clés

1. Alice choisit deux nombre premier entre eux p et q
   Alice calcule { n=p q
   Φ(n)= (p-1) (q-1)
2. Alice choisit : e premier avec Φ(n) avec $0<e<\Phi(n)$
   Et d tel que $ed=1 \bmod \Phi(n)$
3. Alice génère une clé publique K_A =(e, n) qu'elle envoie a Bob
   Et une clé privée K_B = (d, n) qu'elle garde soigneusement chez elle

Pour le chiffrement on a : $C= M^e \bmod n$
Pour le déchiffrement on a : $M=C^d \bmod n$

On donne pour le chiffrement et le déchiffrement ***la fonction Modulo exponentiation*** :
En arithmétique modulaire, l'exponentiation modulaire est un type d'élévation à la puissance (exponentiation) exécutée modulo un entier. Elle est particulièrement utilisée en informatique, spécialement dans le domaine de la cryptologie.
Généralement, les problèmes d'exponentiation modulaire s'expriment sous la forme suivante :

$C= M^e \bmod n = (M *M*M*M….*M) \% n$

$= ((M*M)\% n)*M…..)\%n$

3.4.1.2 Exemple avec RSA

1. p = 7 et q = 19
2. n = 7 * 19 = 133
3. m = (p-1) * (q-1) = 6 * 18 = 108
4. Choix de e premier avec m, on a le PGCD(5,108) = 1 donc e = 5
5. Détermination de d tel que de mod m = 1, autrement dit, il existe k tel que d = (1+km)/e d=65

Donc les deux clés de notre RSA sont :

Clé publique : (n = 133 ; e = 5)
Clé privée : (n = 133 ; d = 65)
Supposons maintenant qu'on veut transmettre le message M=6
Le chiffrement $C=M^e \bmod n$

$C=6^5 \bmod 133 = 7776 \bmod 133 = 62$

Le déchiffrement $M=C^d \bmod n$

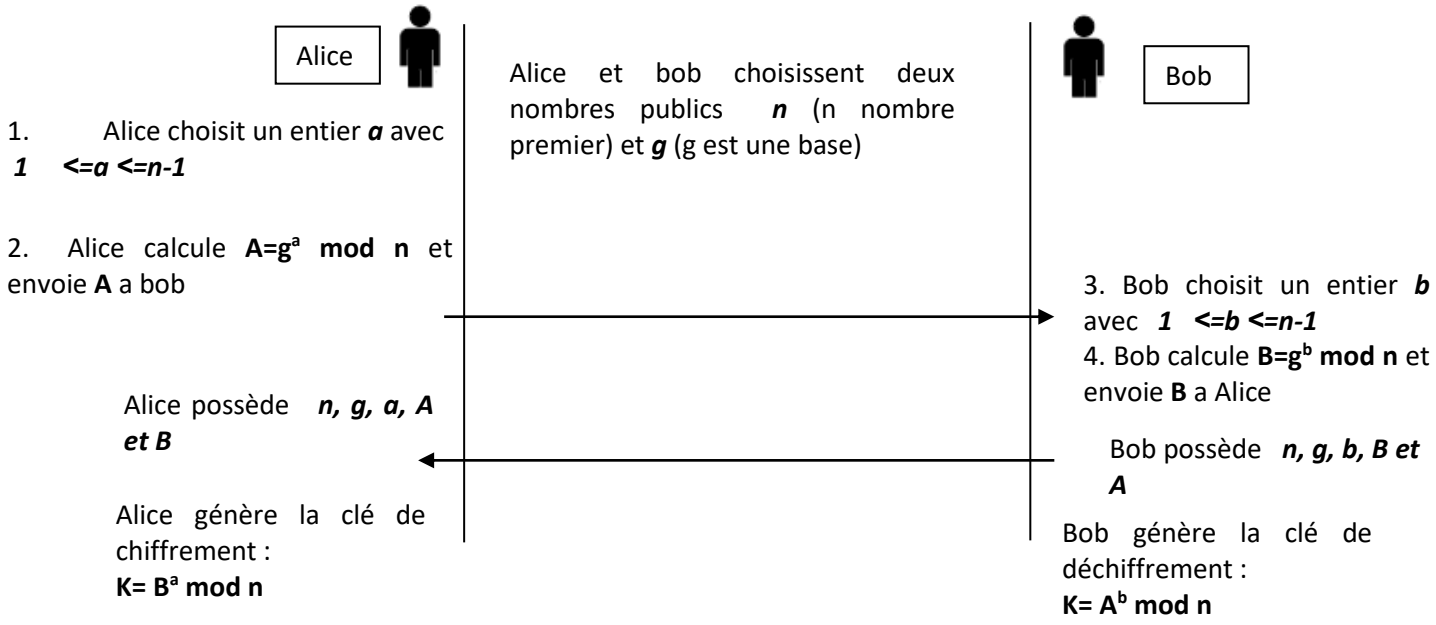
$M=62^{65} \bmod 133$

$M=62^{65} \bmod 133$
$= 62 * 62^{64} \bmod 133$
$= 62 * (62^2)^{32} \bmod 133$
$= 62 * (3884)^{32} \bmod 133$
$= 62 * (3884 \bmod 133)^{32} \bmod 133$
$= 62 * 120^{32} \bmod 133$
$= 62 * 36^{16} \bmod 133$
$= 62 * 99^8 \bmod 133$
$= 62 * 92^4 \bmod 133$
$= 62 * 85^2 \bmod 133$
$= 62 * 43 \bmod 133$
$= 2666 \bmod 133$
$= 6$

3.4.2 Protocole d'échange des clés de Diffie-Hellman

3.4.2.1 Principes :

Deux entités Alice et Bob voudraient se mettre d'accord sur un secret pour échanger des messages publiques, d'une manière confidentiel.



Protocole d'échange des clés de Diffie-Hellman exemple

1 Alice et Bob choisissent un nombre premier n=23 et une base g=3
2 Alice choisit un nombre entier au hasard a avec 1 <= a <=n-1, a=6
3 Alice calcule $A=g^a \bmod n$
$A=3^6 \bmod 23 = 16$

4. Bob choisit un entier b au hasard avec 1 <= b <=n-1, b=15
5. Bob calcule $B=g^b \bmod n$
$B=3^{15} \bmod 23 = 12$

6. Alice génère sa clé $K=B^a \bmod n = (12)^6 \bmod 23 = 9$
7. Bob génère sa clé $K=A^b \bmod n = (16)^{15} \bmod 23 = 9$

La clé d'Alice et la même clé que celle de Bob